Deep Neural Networks for Inverse Problems with Pseudodifferential Operators: An Application to Limited-Angle Tomography

DEEP NEURAL NETWORKS FOR INVERSE PROBLEMS WITH PSEUDODIFFERENTIAL OPERATORS: AN APPLICATION TO LIMITED-ANGLE TOMOGRAPHY *

TATIANA A. BUBBA†, MATHILDE GALINIER‡, MATTI LASSAS† , MARCO PRATO‡ , LUCA RATTI† , AND SAMULI SILTANEN†

Abstract. We propose a novel convolutional neural network (CNN), called ΨDONet, designed for learning pseudodifferential operators (ΨDOs) in the context of linear inverse problems. Our starting point is the Iterative Soft Thresholding Algorithm (ISTA), a well-known algorithm to solve sparsity-promoting minimization problems. We show that, under rather general assumptions on the forward operator, the unfolded iterations of ISTA can be interpreted as the successive layers of a CNN, which in turn provides fairly general network architectures that, for a specific choice of the parameters involved, allow to reproduce ISTA, or a perturbation of ISTA for which we can bound the coefficients of the filters. Our case study is the limited-angle X-ray transform and its application to limited-angle computed tomography (LA-CT). In particular, we prove that, in the case of LA-CT, the operations of upscaling, downscaling and convolution, which characterize our ΨDONet and most deep learning schemes, can be exactly determined by combining the convolutional nature of the limited angle X-ray transform and basic properties defining an orthogonal wavelet system. We test two different implementations of ΨDONet on simulated data from limited angle geometry, generated from the ellipse data set. Both implementations provide equally good and noteworthy preliminary results, showing the potential of the approach we propose and paving the way to applying the same idea to other convolutional operators which are ΨDOs or Fourier integral operators.

Key words. X-ray transform, limited angle tomography, deep neural networks, convolutional neural networks, wavelets, sparse regularization, Fourier integral operators, pseudodifferential operators, microlocal analysis

AMS subject classifications. 44A12, 68T07, 35S30, 58J40, 92C55

1. Introduction.

In the context of microlocal analysis, the theory of pseudodifferential operators (ΨDOs), introduced by Kohn and Nirenberg in 1965, and Fourier integral operators (FIOs), defined by Hörmander in 1971, finds remarkable applications in many fields of Mathematics, from spectral theory to general relativity, from the study of the behavior of chaotic systems to scattering theory and inverse problems [26]. A prominent example in the inverse problem field is given by the X-ray transform or, in the two-dimensional case, Radon transform:

$$R(u)(s,\omega) = \int_{-\infty}^{\infty} u(s\omega^{\perp} + t\omega)\,\mathrm{d}t \qquad s \in \mathbb{R},\ \omega, \omega^{\perp} \in S^1 \tag{1.1}$$

where $\omega^{\perp}$ denotes the vector in the unit sphere S^1 obtained by rotating ω counterclockwise by 90° [38]. It is possible to show (see, *e.g.*, [43]) that the normal operator R^*R of the Radon transform R is an elliptic ΨDO of order -1 and a convolutional

*Corresponding author: Luca Ratti (luca.ratti@helsinki.fi).

Funding: TAB, ML, LR and SS were supported by the Finnish Centre of Excellence in Inverse Modelling and Imaging, 2018-2025, decision number 312339, Academy of Finland grants 284715, 312110, 310822 and Faculty of Science ATMATH project. MG and MP acknowledge support by INdAM-GNCS (Research Projects 2018). MG is also supported by INdAM Doctoral Programme in Mathematics and/or Applications Cofunded by Marie Sklodowska-Curie Actions (INdAM-DP-COFUND-2015), grant number 713485. MP is partially funded by the ECSEL JU programme under the PRYSTINE Project Grant Agreement No. 783190.

†Dept. of Maths and Stats, University of Helsinki, Helsinki, FI-00014 (tatiana.bubba@helsinki.fi, luca.ratti@helsinki.fi, matti.lassas@helsinki.fi, samuli.siltanen@helsinki.fi).

‡Dept. of Physics, Comp. Sci. and Maths, University of Modena and Reggio Emilia, Modena, IT-41125 (mathildeemmanuelle.galinier@unimore.it, marco.prato@unimore.it)

operator associated with the Calderón-Zygmund kernel $K(x,y) = \frac{1}{|x-y|}$ for $x \neq y$. When the direction vector ω is restricted within a limited angular range $[-\Gamma, \Gamma]$, the normal operator $R_\Gamma^* R_\Gamma$ of the limited angle Radon transform R_Γ is a convolutional operator associated with the kernel

$$K(x,y) = \frac{1}{|x-y|}\,\chi_\Gamma(x-y) \qquad \text{for } \; x \neq y, \tag{1.2}$$

where χ_Γ denotes the indicator function of the cone in $\mathbb{R}^2$ between the angles $-\Gamma$ and Γ. The operator $R_\Gamma^* R_\Gamma$ is no longer a ΨDO, but it belongs to a wider class of FIOs, which includes operators associated with a kernel showing some discontinuities along lines [26].

The inverse problem arising from the limited angle Radon transform, *i.e.*, limited-angle computed tomography (LA-CT), appears frequently in practical applications, such as dental tomography [31], damage detection in concrete structures [24], breast tomosynthesis [55] or electron tomography [15].

Microlocal analysis has been widely applied on the Radon transform, especially in the case of incomplete data, with the purpose to characterize its behavior with respect to singularities in the images (see, *e.g.*, [20, 44, 45, 29, 28, 16]). In particular, some recent works are focused on the treatment of artifacts appearing in reconstructions from limited-angle data (see [39, 40, 7]). In this framework, microlocal analysis can be used to predict which singularities can be reconstructed in a stable way from limited angle measurements [7, 18, 32, 42]. In practice, thanks to microlocal analysis, we are able to identify the part of the wavefront set of the target corresponding to the missing wedge from the measurement geometry.

Even with this fundamental information, the task of robustly recovering the unknown quantity of interest from such partial indirect measurement is a challenging one, due to the ill-posedness of the CT problem, which is even more severe because of the limited angular range [13]. As a result, classical methods, such as the filtered backprojection (FBP) [38], yield poor performances. Traditional inversion methods of the form (2.3)-(2.5), based on complementing the insufficient measurements by imposing *a priori* information on the solution, define effective regularization methods which generally allow for accurate reconstructions from fewer tomographic measurements than usually required by standard methods like FBP. In more recent years, machine learning approaches, in particular, deep learning, with convolutional neural networks (CNNs) being the most prominent design in the context of imaging, are increasingly impacting the field of inverse problems [4], and (LA-)CT is no exception (see, in particular, [4, section 4] for an overview of learning approaches from a functional analytic regularization perspective and [4, section 7] for their applicability to prototypical examples of inverse problems, including CT). The majority of recent data-driven approaches for LA-CT focuses on recovering or inpainting the missing part of the wavefront set from the measured data (see, *e.g.*, [9, 48] and the references therein for a thorough review of model-based and data-driven approaches based on sparsifying transforms and edge-preserving regularizers in the context of LA-CT).

In this paper, we are *not* interested in designing an(other) approach for inferring the missing wedge in LA-CT, but rather we aim at investigating neural networks inspired by FIOs and ΨDOs, for which LA-CT is a case study. Our starting point is the traditional sparsity-based minimization problem of the form (2.3)-(2.5). A well-known technique for its solution is the Iterative Soft Thresholding Algorithm (ISTA), introduced in 2004 in the seminal paper by Daubechies, Defrise and De Mol [12]. The convergence result in the paper relies on the assumption that the sparsifying system

forms an orthogonal basis, as it is the case for many families of wavelets [35]. ISTA iteratively creates the sequence $\{w^{(n)}\}_{n=1}^{N}$ as follows:

$$w^{(n)} = \mathcal{S}_{\lambda/L}\left(w^{(n-1)} - \frac{1}{L}K^{(n)}w^{(n-1)} + \frac{1}{L}b^{(n)}\right), \tag{1.3}$$

where, in our case, $K^{(n)} = WR_\Gamma^* R_\Gamma W^*$, with W wavelet transform associated with an orthogonal family, $b^{(n)} = WR_\Gamma^* m$ and $S_\beta(w)$ is the (component-wise) soft-thresholding operator (see equations (2.7) and (3.1) for all the details). It is well-known that the unrolled iterations of ISTA can be considered as the layers of a neural network. Learned ISTA (LISTA), introduced in [22], and ISTA-Net, introduced in [53], are examples of neural networks obtained by laying out the operations of ISTA for a few iterations. The major difference with our approach is that LISTA and ISTA-Net are not CNNs. Unrolled schemes coming from proximal primal-dual optimization method are also proposed in [2, 3], where the proximal operators are replaced with CNNs. While in [2, 3] the goal is to learn a proximal operator, in the approach we propose the regularization term is fixed and we learn a correction of the normal operator $R_\Gamma^* R_\Gamma$. Deep unfolded schemes for problems other than CT are introduced, for instance, in [25, 54]. In [34] the authors propose an unsupervised approach, combined with unrolled schemes, to learn adversarial regularizers and apply it to the case of full-angle CT. In [27] the authors investigate the relationship between CNNs and iterative optimization methods, including ISTA, for the case of normal operators associated with a forward model which is a convolution. However, the resulting U-net, FBPConvNet, does not aim at imitating an unrolled version of an iterative method, which makes it fundamentally different in spirit to the methodology we propose. Indeed, the goal of our work is to show that, under some assumptions on the operator $R_\Gamma W^*$, it is possible to interpret the operations in (1.3) as a layer of a CNN, which in turn provides fairly general network architectures that allow to recover standard ISTA for a specific choice of the parameters involved.

Motivated by this, we propose a new CNN, which we name **ΨDONet**, aimed at learning convolutional FIOs and ΨDOs. The key feature of ΨDONet is that we split the convolutional kernel into $K = K_0 + K_1$ where K_0 is the known part of the model (in the limited angle case, $K_0 = R_\Gamma^* R_\Gamma$) and K_1 is an unknown ΨDO to be determined or, better, to be learned. Basically, in K_1 lays the potential to add information in the reconstruction process with respect to the known part of the model K_0. ΨDONet takes advantage of the possibility to use small filters encoding a combination of upscaling, downscaling and convolution operations, as it is common practice in deep learning. Remarkably, we prove that such operations can be exactly determined combining the convolutional nature of the limited angle Radon transform and basic properties defining an orthogonal wavelet system. While this might seem contrary to the machine learning philosophy which finds its strength in avoiding any predefined structure for neural networks, our recipe gives insight into understanding and interpreting the results of the proposed CNN, combining results from FIOs, ΨDOs and classical variational regularization theory. At the same time, the possibility to deploy such operations allows for a significant reduction of the parameters involved, especially when compared to the standard interpretation of ISTA as a recurrent neural network: this is fundamental when it comes to a practical numerical implementation of the proposed CNN. Overall, ΨDONet is able to reproduce ISTA, or a perturbation of ISTA for which we can bound the coefficients of the filters, and has the potential to learn ΨDO-like structures which are intrinsic to the problem at hand.

As a proof of concept, we test ΨDONet on simulated data from limited angle geometry, generated from the ellipse data set. We provide two different implementations of ΨDONet: Filter-Based ΨDONet (ΨDONet-F), where the backprojection opererator is approximated by its filter-equivalent, and Operator-Based ΨDONet (ΨDONet-O), where the backprojection opererator encoded in K_0 is not approximated but explicitly computed. Both implementations provide equally good and noteworthy preliminary results, the main difference being a greater computational efficiency for ΨDONet-O. The improvement provided by our results, compared to standard ISTA (and FBP), bodes well for further numerical testing which we leave to future work.

Finally, we stress that the contribution of our paper is mainly theoretical and is in line with current research in data-driven inversion, which combines knowledge from traditional inverse problems theory with data-driven techniques. While in our paper we derived the result contingently to the case of limited angle Radon transform, our approach is actually very general and can be extended to any convolutional operator which is a FIO or ΨDO. This is the case, for instance, of the geodesic X-ray transform [49], and its applications in seismic imaging, or synthetic-aperture radar (SAR) [41]. Finally, our paper paves the way to theoretical generalization results, in light of recent contributions like [14].

The remainder of this paper is organized as follows: section 2 is devoted to reviewing the theoretical background of sparsity promoting regularization and the wavelet transform. In section 3, we detail the key idea of our approach, namely we give a convolutional interpretation of ISTA using the wavelet transform. The neural network architecture we propose, ΨDONet, is introduced in section 4, where we also prove our main theoretical result. Two different implementations of ΨDONet, which we call Filter-Based ΨDONet and Operator-Based ΨDONet, are described in section 5. Finally, we demonstrate the performance of our network by a series of numerical experiments (see section 6). Concluding remarks and future prospects are briefly summarized in section 7. The appendices collect proofs of some of the results presented in section 2.

2. Theoretical background. In this section, we collect some theoretical results which are preliminary to the main discussion of the paper.

2.1. Sparsity-promoting regularization via ISTA. Consider the inverse problem of determining $u^\dagger \in X$ from the measurements $m = Au^\dagger + \epsilon$, being $A : X \to Y$ a linear bounded operator between the Hilbert spaces X and Y. The perturbation $\epsilon \in Y$ is such that $\|\epsilon\|_Y \leq \delta$.
The main application we have in mind is the limited-angle Radon transform R_Γ, which is a continuous linear operator, *e.g.*, from $X = L^2(\Omega)$ (being $\Omega \subset \mathbb{R}^2$) to $Y = L^2([-\Gamma, \Gamma] \times [-S, S])$ (see [38, Theorem 2.10]).
Introduce an orthonormal basis $\{\psi_I\}_{I \in \mathbb{N}}$ in X. For later purposes, we will assume that such basis is a wavelet system. Define $W : X \to \ell^2(\mathbb{N})$ the operator associating to any $u \in X$ the sequence of its component with respect to the wavelet basis: $(Wu)_I = (u, \psi_I)_X$, where $(\cdot, \cdot)_X$ denotes the inner product in X. We assume to know *a priori* that the exact solution $u^\dagger$ is sparse with respect to the wavelet basis ψ_I :

$$Wu^\dagger = w^\dagger \in \ell^0(\mathbb{N}). \tag{2.1}$$

The reconstruction of $u^\dagger$ (or, equivalently, $w^\dagger$) from the noisy measurements m is in general an ill-posed problem, hence we introduce the following regularized problem:

$$\min_{w \in \ell^1(\mathbb{N})} \|AW^* w - m\|_Y^2 + \lambda \|w\|_{\ell^1}, \tag{2.2}$$

being $\lambda > 0$. The requirement $w \in \ell^1(\mathbb{N})$ is in general not satisfied by any $w = Wu$, $u \in X$; hence, we define $Z \subset X$, $Z = \{u \in X : Wu \in \ell^1(\mathbb{N})\}$. In particular, in the tomography application, it is possible to show that the ℓ^1 norm of the components of the wavelet representation of a $L^2(\Omega)$ function is equivalent to the Besov norm $B^1_{1,1}(\Omega)$ (see, *e.g.*, [12, formula (A3)]). Hence, the minimization problem (2.2) is equivalent to

$$\min_{u \in Z} \|Au - m\|_Y^2 + \lambda \|u\|_Z. \tag{2.3}$$

It is well known that the regularization term involving the ℓ^1 norm is a good choice to encode the *a priori* information regarding the sparsity of $w^\dagger$. In particular if the noise level tends to 0, there exists a suitable choice of $\lambda = \lambda(\delta)$ ensuring the convergence of w^δ_λ to $w^\dagger$, being w^δ_λ the solution of (2.2). We report a result from [17] which also shows that such convergence occurs with linear rate. In particular, [17, Corollary 2] does not require $w^\dagger$ to satisfy a classical source condition, but relies on the sparsity assumption (2.1) and on the injectivity of the operator A. Such property can be restrictive in some applications, and as a consequence many alternative results involve some weaker assumptions (as the well known Restricted Isometry Property); nevertheless, in our tomographic application, we can rely on the injectivity of the Radon transform, even in the limited angle case.

PROPOSITION 2.1. *Let $w^\dagger$ satisfy (2.1), and suppose $A : X \to Y$ is injective. Define w^λ_δ a solution of problem (2.2) associated with a regularization parameter λ and a noise level δ. For sufficiently small δ, provided that λ is chosen such that $\lambda = c_0\delta$, then there exists a positive constant $c_1 = c_1(c_0, A, \|w^\dagger\|_{\ell^0})$ such that*

$$\|w^\dagger - w^\lambda_\delta\|_{\ell^1} \leq c_1 \delta. \tag{2.4}$$

This proposition is an immediate consequence of [17, Corollary 2], relying on [17, Lemma 2] to ensure that A is weak*-to-weak continuous. From now on, we suppose that λ is chosen as a linear function of δ and denote u^λ_δ as u_δ and w^λ_δ as w_δ.

We now introduce a finite-dimensional approximation of the regularized problem (2.2). Consider the subspace $X_p \subset X$, $X_p = \text{span}\{\psi_I\}_{I=1}^p$, mapped by W into the space $W_p = \{w \in \mathbb{R}^{\mathbb{N}} : \ w_I = 0 \ \forall I > p\}$ (which is isomorphic to $\mathbb{R}^p$). Denote by $\mathbb{P}_p$ the orthogonal projection of $\ell^2(\mathbb{N})$ onto W_p and by $\widetilde{\mathbb{P}}_p = W^* \mathbb{P}_p W$ the orthogonal projection of X onto X_p. Moreover, we introduce an orthogonal basis $\{\varphi_j\}_{j=1}^\infty$ on Y and define $Y_q = \text{span}\{\varphi_j\}_{j=1}^q$ and the projection $\mathbb{P}_q : Y \to Y_q$. For any choice of $p, q > 0$, let $A_{p,q}$ be the representation of the operator A in the subspaces X_p, Y_q, namely $A_{p,q} = \mathbb{P}_q A \widetilde{\mathbb{P}}^*_p$. Consider the following minimization problem:

$$\min_{w \in W_p} \|A_{p,q} W^* w - \mathbb{P}_q m\|_Y^2 + \lambda \|w\|_{\ell^1}. \tag{2.5}$$

Denote by $w_{\delta,p,q}$ a solution of (2.5). We can prove the following convergence result:

PROPOSITION 2.2. *Let $w^\dagger$ satisfy (2.1) and A be an injective operator. Suppose moreover that for a suitable choice of p, q it is possible to ensure that $\|w^\dagger - \mathbb{P}_p w^\dagger\|_{\ell^2} \leq c_p\delta$ and $\|(I - \mathbb{P}_q)A\|_{X \to Y} \leq c_q\delta$. Then, provided that λ is chosen as $\lambda = c_0\delta$, there exists a positive constant c_2 (depending on $\|A\|, \|w^\dagger\|_{\ell^1}$, on the choice of $\{\psi_I\}, \{\varphi_j\}$ and on the constants c_0, c_1, c_p, c_q) such that:*

$$\|w_{\delta,p,q} - w^\dagger\|_{\ell^1} \leq c_2 \delta. \tag{2.6}$$

The proof, which follows by an application of the variational source condition reported in [17, Section 3], is reported in Appendix A.

Remark 2.3. Upper bounds of the kind $\|w^\dagger - \mathbb{P}_p w^\dagger\|_{\ell^2} \leq f(p)$ can be explicitly computed under some particular assumptions on $w^\dagger$. If, for example, we suppose that $u^\dagger$ is a cartoon-like image (*i.e.*, $u^\dagger$ is a C^2-smooth functions apart from jump discontinuity along a finite set of C^2-curves) and $\{\psi_I\}$ is the Haar wavelets basis, it is well known that $\|w^\dagger - \mathbb{P}_p w^\dagger\|_{\ell^2} \leq p^{-1}$ (see, *e.g.*, [35, Chapter 9]).
On the other hand, an estimate for the term $\|(I - \mathbb{P}_q)A\|_{X\to Y}$ can be obtained by standard results of finite-rank approximation of compact operators. For example, suppose that the operator A is a compact operator. Define $\{s_j\}$ its singular values (i.e., let $\{(s_j, e_j)\}$ be the eigenvalues and eigenfunctions of $(A^*A)^{\frac{1}{2}}$) and suppose the sequence s_j is non-increasingly converging to 0. A sufficient condition for this is that A is a Schatten operator of any class p. If we select the basis $\{\varphi_j\}$ such that $\varphi_j = Ue_j$, where U is the partial isometry in the polar decomposition $A = U(A^*A)^{\frac{1}{2}}$, then it holds $\|(I - \mathbb{P}_q)A\|_{X\to Y} \leq s_{q+1}$. In the case of the Radon transform in 2D, according to [37, Section IV.3], $s_j = c_R j^{-\frac{1}{2}}$, hence to get $\|(I - \mathbb{P}_q)A\|_{X\to Y} \leq c_R\delta$ it is enough to consider $q \geq \frac{1}{\delta^2} - 1$

A well-know technique for the solution of the minimization problem (2.5) is the Iterative Soft Thresholding Algorithm (introduced in [12]), which consists in selecting an initial guess $w^{(0)} \in \mathbb{R}^p (\cong W_p)$ and in iteratively creating the sequence $\{w^{(n)}\}_{n=1}^N$ as follows:

$$w^{(n)} = \mathcal{T}(w^{(n-1)}) = \mathcal{S}_{\lambda/L}\left(w^{(n-1)} - \frac{1}{L}WA^*_{p,q}A_{p,q}W^*w^{(n-1)} + \frac{1}{L}WA^*_{p,q}m\right), \tag{2.7}$$

where $\frac{1}{L} > 0$ is interpreted as a (fictitious) time step and, for $\beta > 0$, $S_\beta(w)$ is the (component-wise) soft-thresholding operator:

$$[\mathcal{S}_\beta(w)]_I = S_\beta(w_I); \qquad S_\beta(w_I) = \begin{cases} w_I + \beta & \text{if } w_I < -\beta \\ 0 & \text{if } |w_I| \leq \beta \\ w_I - \beta & \text{if } w_I > \beta \end{cases}.$$

The convergence of $\{w^{(N)}\}$ to a minimizer $w_{\delta,p,q}$ of (2.5) is analyzed, in an infinite dimensional context, in [8]. The following result for the discrete problem under consideration is instead a direct consequence of [6, Theorem 25]:

Proposition 2.4. *If L is chosen such that $L \geq \|WA^*_{p,q}A_{p,q}W^*\|/2$ then the sequence $\{w^{(N)}\}$ generated via (2.7) by any $w^{(0)} \in \mathbb{R}^p$ converges in ℓ^2 to the solution $w_{\delta,p,q}$ of (2.5). Moreover, there exist $c_3 > 0$ and $0 \leq a < 1$ (both depending on $A_{p,q}$, L and $\|w^\dagger\|_{\ell^2}$) such that*

$$\|w^{(N)} - w_{\delta,p,q}\|_{\ell^2} \leq c_3 a^N. \tag{2.8}$$

2.2. A modification of ISTA. We now consider a perturbation of ISTA (2.7). Let $Z : \ell^2(\mathbb{N}) \to \ell^2(\mathbb{N})$ satisfy

$$\|WA^*_{p,q}A_{p,q}W^* - Z\|_{\ell^2\to\ell^2} \leq \rho. \tag{2.9}$$

Then, we substitute Z in place of the matrix $WA^*_{p,q}A_{p,q}W^*$ in the expression of ISTA. To remark the dependency on the perturbation amplitude ρ, we denote by $\{w^{(n)}_\rho\}$ the sequence obtained by selecting $w^{(0)}_\rho \in \mathbb{R}^p$ and iterating

$$w^{(n)}_\rho = \mathcal{T}_\mathcal{Z}(w^{(n-1)}_\rho) = \mathcal{S}_{\lambda/L}\left(w^{(n-1)}_\rho - \frac{1}{L}Zw^{(n-1)}_\rho + \frac{1}{L}WA^*_{p,q}m\right). \tag{2.10}$$

The following result shows a connection between the convergence of the sequence $\{w_\rho^{(n)}\}$ to the minimizer $w_{\delta,p,q}$ and the magnitude of the perturbation ρ.

PROPOSITION 2.5. *Let $w^{(0)} = w_\rho^{(0)}$, $L \geq \|WA_{p,q}^* A_{p,q} W^*\|$ and consider $N_0, \eta_0 > 0$. Then there exists a constant $\widetilde{c}_4$, depending on $L, A, w^{(0)}, \|w^\dagger\|_{\ell^2}$ and on N_0, η_0, such that if $N \geq N_0$ and $\rho N \leq \eta_0$ then*

$$\|w_\rho^{(N)} - w_{\delta,p,q}\|_{\ell^2} \leq c_3 a^N + \widetilde{c}_4 \rho N. \tag{2.11}$$

If, moreover, N, ρ are chosen as $N > \frac{\ln(\delta^{-1})}{\ln(a^{-1})}$ and $\rho < \frac{\delta}{N}$, then (for $c_4 = c_3 + \widetilde{c}_4$)

$$\|w_\rho^{(N)} - w_{\delta,p,q}\|_{\ell^2} \leq c_4 \delta. \tag{2.12}$$

The proof of this proposition follows by the nonexpansivity of the soft-thresholding operator and is reported in Appendix B.

We collect the results obtained in Proposition 2.2 and Proposition 2.5 in the following final convergence estimate:

THEOREM 2.6. *Let $w^\dagger$ satisfy (2.1) and let A be injective. For sufficiently small δ, select a regularization parameter $\lambda = c_0\delta$. Select p, q s.t. $\|w^\dagger - \mathbb{P}_p w^\dagger\| \leq c_p \delta$ and $\|(I - \mathbb{P}_q)A\|_{X\to Y} \leq c_q \delta$. Let $L \geq \|WA_{p,q}^* A_{p,q} W^*\|$ and consider the perturbed ISTA iterations (2.10), where the operator Z satisfies (2.9), $N = \log_a \delta$ and $\rho = \frac{\delta}{N}$. Then, there exists a positive constant c_5 (depending on the previously introduced constants $c_0, c_1, c_2, c_3, c_4, c_p, c_q$) such that, for sufficiently small δ,*

$$\|w_\rho^{(N)} - w^\dagger\|_{\ell^2} \leq c_5 \delta. \tag{2.13}$$

2.3. Wavelets in 2D. In order to derive the main results of the paper, we need to assume that the orthogonal basis $\{\psi_I\}_{I=1}^\infty$ is a wavelet basis in $X = L^2(\Omega)$. Although our approach is sufficiently general to handle higher-dimensional spaces, we are going to focus on the two-dimensional case, *i.e.*, $\Omega \subset \mathbb{R}^2$ (*e.g.*, $\Omega = [0,1]^2$). Before moving to the representation of the operator A^*A with respect to such basis, we need to describe in more details its structure.

A common way to define a wavelet basis in $\mathbb{R}^2$ is to rely on two real functions ψ and φ, respectively defined as mother wavelet and scaling function, whose support is in $[0,1]$. We identify an element ψ_I of the basis by its scale j, its translation $k \in \mathbb{N}_0^2$ and its type $(t) \in \{(v), (h), (d), (f)\}$ (respectively, vertical, horizontal, diagonal and low-pass filter). We denote $\psi_I(x)$ as $\psi_{j,k}^{(t)}(x) = 2^j \psi^{(t)}(2^j x - k)$, $x \in [0,1]^2$, where we have:

$$\psi^{(v)}(x_1, x_2) = \phi(x_1)\psi(x_2) \qquad \psi^{(h)}(x_1, x_2) = \psi(x_1)\phi(x_2)$$
$$\psi^{(d)}(x_1, x_2) = \psi(x_1)\psi(x_2) \qquad \psi^{(f)}(x_1, x_2) = \phi(x_1)\phi(x_2)$$

When selecting a maximum scale J (and $J_0 < J$ as coarsest scale), we can define a wavelet basis of $p = 2^{2J}$ elements as follows: take $j \in \{J_0, \dots, J_1 = J - 1\}$; for each $j \neq J_0$, consider wavelets of the types (v), (h) and (d), whereas for $j = J_0$ include also the type (f). For each level j and type (t), consider offsets $k = (k_1, k_2)$, $k_1 = 0, \dots, 2^j - 1$, $k_2 = 0, \dots, 2^j - 1$.

We group the wavelet basis functions in subbands, each of which is identified by a scale j and a type (t), obtaining $3(J - J_0) + 1$ subsets.

3. ISTA and Convolutional Neural Networks. It is already well known that the unrolled iterations of ISTA can be considered as the layers of a neural network (see, *e.g.*, [22]). Indeed, the n-th iteration of ISTA can be written as

$$w^{(n)} = \mathcal{S}_{\lambda/L}\left(w^{(n-1)} - \frac{1}{L}K^{(n)}w^{(n-1)} + \frac{1}{L}b^{(n)}\right), \tag{3.1}$$

being $K^{(n)} = WA^*_{p,q}A_{p,q}W^*$ and $b^{(n)} = WA^*_{p,q}m$, independently of n. At the same time, (3.1) can be seen as the n-th layer of a recurrent Neural Network, where $K^{(n)}$ is the matrix of the weight coefficients and $b^{(n)}$ is the bias vector. Notice that the resulting architecture is the one of a recurrent neural network although, due to its theoretical deduction, it does not present any advanced residual block (such as skip connections) which are common features in the related literature. We also point out that formula (3.1) enforces a specific choice of the nonlinear activation function, namely the soft-thresholding operator, instead of the more widely-used ReLU or sigmoid functions. Additionally, the soft thresholding operator S_α can be written in terms of rectified linear units as follows:

$$S_\alpha(x) = \max(0, x-\alpha) - \max(0, -x-\alpha) = \mathrm{ReLU}(x-\alpha) - \mathrm{ReLU}(-x-\alpha) \tag{3.2}$$

for $x \in \mathbb{R}$; for vectors, it is applied componentwise.

When considering only the first N iterations of ISTA, we can collect the parameters appearing in the layers in a vector $\theta \in \Theta$. Together with the entries of the matrices $K^{(n)}$, we may consider as parameters the steplength L as well as the regularization parameter λ: see subsection 5.3 for more details. Conversely, the bias vectors $b^{(n)}$ are not to be considered as parameters: they are fixed and equal to $WA^*_{p,q}m$ in each layer. We then introduce the map $f_\theta : Y \to \ell^1(\mathbb{N})$, parameterized by $\theta \in \Theta$, which takes as an input $m \in Y_q$ and computes N iterations like (3.1), where, for each n, $K^{(n)} \in \mathbb{R}^{p\times p}$ is specified in θ and $b^{(n)} = WA^*_{p,q}m$. For any selected value of p, q, N, λ, L, we know that there exists a particular choice θ_0 which corresponds to the ISTA iterations associated to the measurements m.

In this section we show that, under some assumptions on the operator A, it is possible to interpret the operations in (3.1) as a layer of a CNN. We therefore provide a fairly general network architecture which allows to recover the standard ISTA iterations (or a perturbation of the kind described by (2.9)), for a specific choice of the parameters.

From now on, we focus on the case $X = L^2(\Omega)$, and consider a wavelet basis $\{\psi_I\}$ of the kind described in subsection 2.3.

3.1. A convolutional interpretation of ISTA. We first show, under additional assumptions on operator A, how to translate the Neural Network encoded by the operator f_θ above into a CNN, allowing for a significant reduction of the number of parameters involved. Suppose that A^*A is a convolutional kernel operator, *i.e.*,

$$\begin{aligned} \mathbf{K}_{I,I'} = (A^*A\psi_I, \psi_{I'})_X = \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} K(x,x')\psi_I(x)\psi_{I'}(x')dxdx', \\ K(x,x') = K(x-x'). \end{aligned} \tag{3.3}$$

According to the description in subsection 2.3, the wavelet basis can be naturally split in subbands, each of which is identified by a couple j,(t). This implies that the matrix $\mathbf{K}$ representing A^*A can be seen as a block matrix. We now aim at describing the application of each block $\mathbf{K}^{(t)\to(t')}_{j\to j'}w^{(t)}_j$ by means of the following operations:

1. Discrete convolution. Let $B \in \mathbb{R}^{b\times b}$, $C \in \mathbb{R}^{(2b-1)\times(2b-1)}$, and denote the elements of C with indices i, j , being $i = -b+1, \dots, 0, \dots, b-1$, $j = -b+1, \dots, 0, \dots, b-1$. Then, $C * B \in \mathbb{R}^{b\times b}$:

$$(C * B)_{k,l} = \sum_{i=0}^{b-1}\sum_{j=0}^{b-1} C_{k-i,l-j} B_{i,j} \tag{3.4}$$

2. Upsampling. Let $B \in \mathbb{R}^{b\times b}$; then, $\mathscr{U}(B) \in \mathbb{R}^{2b\times 2b}$ satisfies:

$$\mathscr{U}(B)[2k : 2k+1, 2l : 2l+1] = \begin{bmatrix} B_{k,l} & 0 \\ 0 & 0 \end{bmatrix} \quad \forall k, l = 0, \dots, b-1, \tag{3.5}$$

   where the notation $\mathscr{U}(B)[2k : 2k+1, 2l : 2l+1]$ is used to denote a submatrix of $\mathscr{U}(B)$ containing the rows from $2k$ to $2k+1$ and all the columns from $2l$ to $2l+1$. We denote by $\mathscr{U}^\eta$ the iterated application of $\mathscr{U}$: $\mathscr{U}^\eta = \mathscr{U} \circ \dots \circ \mathscr{U}$ (η times).
3. Downsampling. Let $B \in \mathbb{R}^{2b\times 2b}$; then, $\mathscr{D}(B) \in \mathbb{R}^{b\times b}$ satisfies:

$$\mathscr{D}(B)_{k,l} = B_{2k,2l} \quad \forall k, l = 0, \dots, b-1. \tag{3.6}$$

We denote by $\mathscr{D}^\eta$ the iterated application of $\mathscr{D}$: $\mathscr{D}^\eta = \mathscr{D} \circ \dots \circ \mathscr{D}$ (η times). The following crucial result provides a full description of the convolutional interpretation of the matrix representing A^*A in the wavelet domain. Such result can be compared to the ones already known in literature, see *e.g.* [11, Formula (4.2)], although the more complicated structure of the wavelet basis entails some significant differences.

PROPOSITION 3.1. *Let $\mathbf{K} \in \mathbb{R}^{p\times p}$ be the matrix representing an operator A^*A satisfying (3.3) in a 2D wavelet basis $\{\psi_I\}_{I=1}^p$. For a vector $w \in \mathbb{R}^p$, let $w_j^{(t)}$ be the vector of the wavelet components related to basis functions of scale j and type (t). Let $\mathbf{K}_{j\to j'}^{(t)\to(t')}$ denote the block of $\mathbf{K}$ corresponding to the $j,(t)$ subset of the column indices and the $j',(t')$ subset of the row indices. Then*

$$\mathbf{K}_{j\to j'}^{(t)\to(t')} w_j^{(t)} = \begin{cases} \mathscr{D}^\delta(\widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')} * W_j^{(t)}) & \text{if} \quad j > j' \\ \widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')} * W_j^{(t)} & \text{if} \quad j = j' \\ \widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')} * \mathscr{U}^\delta(W_j^{(t)}) & \text{if} \quad j < j' \end{cases} \tag{3.7}$$

being $\delta = |j' - j|$, and $\widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')} \in \mathbb{R}^{(2^{\widehat{j}+1}-1)\times(2^{\widehat{j}+1}-1)}$ (where $\widehat{j} = \max(j, j')$):

$$\begin{aligned} \left[\widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')}\right]_d &= \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} K(x - x' - 2^{-\widehat{j}}d)\ \psi_{j',0}^{(t')}(x')\ \psi_{j,0}^{(t)}(x)dxdx' \\ d &= (d_1, d_2); \quad d_1, d_2 = \{-2^{\widehat{j}}+1, \dots, 0, \dots, 2^{\widehat{j}}-1\}. \end{aligned} \tag{3.8}$$

The matrix $W_j^{(t)} \in \mathbb{R}^{2^j\times 2^j}$ is obtained by reshaping the vector $w_j^{(t)} \in \mathbb{R}^{2^{2j}}$ so that $[W_j^{(t)}]_d$ is the component w_I whose index is identified by $(j,(t),d)$.

Proof. Let I, I' be identified by $(j,(t),k)$ and $(j',(t'),k')$, respectively. Then,

$$\begin{aligned}
[\mathbf{K}]_{I',I} &= \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} K(x-x')\ \psi^{(t')}_{j,k'}(x')\ \psi^{(t)}_{j,k}(x)dxdx' \\
&= \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} K(x-x')\ \psi^{(t')}_{j,0}(x'-2^{-j'}k')\ \psi^{(t)}_{j,0}(x-2^{-j}k)dxdx' \\
&= \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} K(x+2^{-j}k-x'-2^{-j'}k')\psi^{(t')}_{j,0}(x)\psi^{(t)}_{j,0}(x)dxdx' \\
&= \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} K(x-x'-2^{-\widehat{j}}(2^{\delta^-}k'-2^{\delta^+}k))\psi^{(t')}_{j,0}(x)\psi^{(t)}_{j,0}(x)dxdx' = \left[\widetilde{\mathbf{K}}^{(t)\to(t')}_{j\to j'}\right]_d,
\end{aligned}$$

where $\delta^+ = \max(0, j-j')$, $\delta^- = \max(0, j'-j)$, and $d = 2^{\delta^-}k' - 2^{\delta^+}k$. For the sake of ease, we use $\mathbf{K}$ instead of $\mathbf{K}^{(t)\to(t')}_{j\to j'}$, $\widetilde{\mathbf{K}}$ instead of $\widetilde{\mathbf{K}}^{(t)\to(t')}_{j\to j'}$, w instead of $w^{(t)}_j$, W instead of $W^{(t)}_j$. Moreover, we denote by $\mathcal{I}$ the set of indices $\mathcal{I} \subset \{1,\dots,p\}$ belonging to the wavelet scale j and type (t).

Consider first the case $j = j'$. Then $\delta = \delta^+ = \delta^- = 0$, and it holds

$$[\mathbf{K}]_{I',I} = \left[\widetilde{\mathbf{K}}\right]_d, \qquad d = k'-k.$$

Therefore,

$$\begin{aligned}
[\mathbf{K}w]_{I'} &= \sum_{I\in\mathcal{I}} [\mathbf{K}]_{I',I}\, w_I = \sum_{I\in\mathcal{I}} \left[\widetilde{\mathbf{K}}\right]_{k'-k(I)} w_I \\
&= \sum_{k_1=-2^j}^{2^j}\sum_{k_2=-2^j}^{2^j} \left[\widetilde{\mathbf{K}}\right]_{k'_1-k_1,k'_2-k_2} W_{k_1,k_2} = [\mathbf{K} * W]_{I'}.
\end{aligned}$$

Let now $j < j'$. Then $\delta = \delta^+ > 0$, $\delta^- = 0$, and

$$\begin{aligned}
[\mathbf{K}w]_{I'} &= \sum_{I\in\mathcal{I}} [\mathbf{K}]_{I',I}\, w_I = \sum_{I\in\mathcal{I}} \left[\widetilde{\mathbf{K}}\right]_{k'-2^{\delta^+}k(I)} w_I \\
&= \sum_{k_1=-2^{j'}}^{2^{j'}}\sum_{k_2=-2^{j'}}^{2^{j'}} \left[\widetilde{\mathbf{K}}\right]_{k'_1-2^{\delta^+}k_1,k'_2-2^{\delta^+}k_2} \mathscr{U}^{\delta^+}(W)_{2^{\delta^+}k_1,2^{\delta^+}k_2} = [\mathbf{K} * \mathscr{U}^{\delta} W]_{I'}.
\end{aligned}$$

Finally, let $j > j'$. Then $\delta^+ = 0$, $\delta = \delta^- > 0$, and

$$\begin{aligned}
[\mathbf{K}w]_{I'} &= \sum_{I\in\mathcal{I}} [\mathbf{K}]_{I',I}\, w_I = \sum_{I\in\mathcal{I}} \left[\widetilde{\mathbf{K}}\right]_{2^{\delta^-}k'-k(I)} w_I \\
&= \sum_{k_1=-2^j}^{2^j}\sum_{k_2=-2^j}^{2^j} \left[\widetilde{\mathbf{K}}\right]_{2^{\delta^-}k'_1-k_1,2^{\delta^-}k'_2-k_2} W_{k_1,k_2} = [\mathscr{D}^{\delta}(\mathbf{K} * W)]_{I'}.
\end{aligned}$$

∎

Remark 3.2. The most relevant consequence of Proposition 3.1 is a significant reduction of the coefficients required to describe the application of A^*A as a function from $\mathbb{R}^p$ to $\mathbb{R}^p$. The standard representation, obtained by a matrix in $\mathbb{R}^{p\times p}$, involves indeed $p^2 = 2^{4J}$ parameters, whereas the representation via the convolutional filters $\widetilde{\mathcal{K}}^{(t)\to(t')}_{j\to j'}$ involves only $O(p)$ elements.

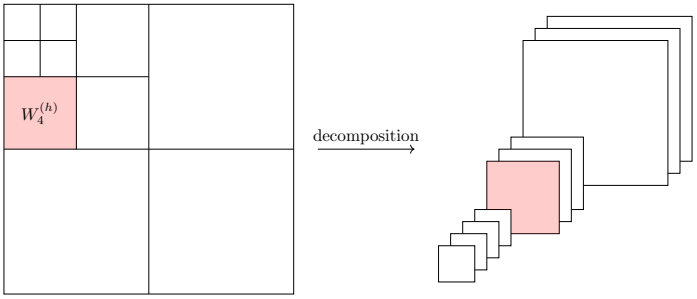


Fig. 1: Interpretation of (3.7). Step 1: decompose the wavelet transform in subbands.

This convolutional interpretation also reflects on the Neural Network architecture proposed in (3.1): if we substitute the multiplication $K^{(n)}w^{(n-1)}$ by the operations encoded by (3.7) (decomposition of $w^{(n-1)}$ in wavelet subbands, upscaling, application of convolutional filters, downscaling), the parameters θ involved in the description of $K^{(n)}$ are reduced. The representation of the linear operators $K^{(n)}$ through convolutions, upscaling and downscaling is a typical feature of CNNs: thus, by designing a CNN which reproduces exactly the operations reported in (3.7) and (3.1), we can ensure that such a network is completely equivalent, for a suitable choice θ_0 of the parameters, to the application of ISTA.

3.2. A working example. In order to better visualize the convolutional representation of ISTA reported in (3.8), we now provide a small example. Consider the case of 64×64 images, thus associated to $J = 6$ and $p = 2^{12}$. Create a wavelet basis consisting of three scales of wavelets, from $J_0 = 3$ to $J_1 = 5$. The resulting basis $\{\psi_I\}_{I=1}^p$ can be therefore split into 10 subbands: 4 associated to the scale $j = 3$ (types: (h), (v), (d) and (f)), 3 associated to the scales $j = 4$ (types: (h), (v), (d)) and 3 with $j = 5$. Each subband consists of 2^{2j} elements.
The operator A^*A is represented in the wavelet basis $\{\psi_I\}$ by a matrix $\mathbf{K} \in \mathbb{R}^{p\times p}$. According to subsection 3.1, the following procedure is equivalent to applying the matrix $\mathbf{K}$ on a vector $w \in \mathbb{R}^p$ (representing the wavelet transform of an image):

1. First, split the vector w into its 10 wavelet subbands, each of which identified by a scale j and a type (t). This operation is depicted in Figure 1. The vector $w_j^{(t)} \in \mathbb{R}^{2j}$ can also be interpreted as a matrix $W_j^{(t)} \in \mathbb{R}^{j\times j}$. The element $[W_j^{(t)}]_d = [W_j^{(t)}]_{(d_1,d_2)}$ is the component associated to the basis function $\psi_{j,d}^{(t)}(x) = 2^j\psi^{(t)}(2^jx_1 - d_1, 2^jx_2 - d_2)$.
2. Secondly, for each subband $j,(t)$, compute the 10 vectors $\mathbf{K}_{j\to j'}^{(t)\to(t')}w_j^{(t)}$, the contributions of $w_j^{(t)}$ on the subband $j',(t')$ of the vector $\mathbf{K}w$. Each matrix $\mathbf{K}_{j\to j'}^{(t)\to(t')}$ is a $2^{2j'} \times 2^{2j}$ block composing the matrix $\mathbf{K}$. According to (3.7), this can be done by means of usampling, downsampling and convolution. Consider the case $j = J_0 = 3$:
   - if $j' = 3$, then $\widehat{j} = 3$ and $\delta = 0$. Thus, if we compute the convolution of the 15×15 filter $\widetilde{\mathbf{K}}_{3\to3}^{(t)\to(t')}$ with the matrix $W_3^{(t)} \in \mathbb{R}^{8\times8}$, we get a 8×8 matrix representing the vector $\mathbf{K}_{3\to3}^{(t)\to(t')}w_3^{(t)} \in \mathbb{R}^{64}$.
   - if $j' = 4$, then we shall use the third variant in formula (3.7) with $\delta = 1$

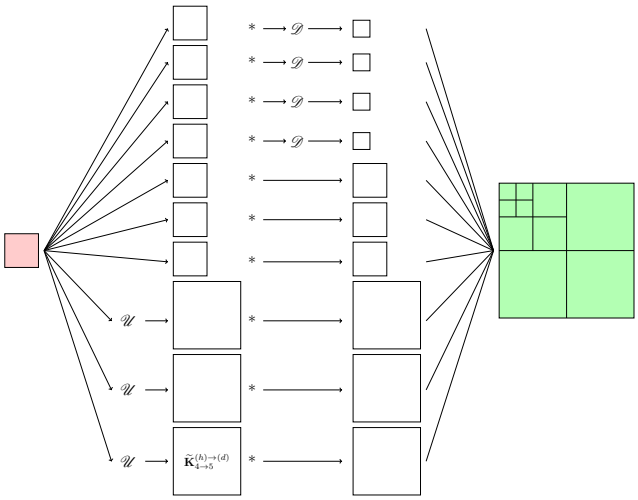


Fig. 2: Interpretation of (3.7). Step 2: convolution, upsampling and downsampling.

(whereas in (3.8) we have $\widehat{j} = 4$). To compute the 16×16 matrix associated to $\mathbf{K}_{3\to4}^{(t)\to(t')} w_3^{(t)}$, we must first upsample the matrix $W_3^{(t)}$ and then convolve it with the 31×31 filter $\widetilde{\mathbf{K}}_{3\to4}^{(t)\to(t')}$.

- if $j' = 5$, then we use again the third variant of (3.7), with $\delta = 2$; hence the matrix $W_3^{(t)}$ must be upsampled twice before being convolved with the 63×63 filter $\widetilde{\mathbf{K}}_{3\to5}^{(t)\to(t')}$.

Consider instead the case $j = 4$:

- if $j' = 3$, then we need to use the first variant in (3.7) with $\delta = 1$ (and (3.8) with $\widehat{j} = 4$), which means we first compute the convolution between the 31×31 filter $\widetilde{\mathbf{K}}_{4\to3}^{(t)\to(t')}$ and the matrix $W_4^{(t)} \in \mathbb{R}^{16\times16}$ and then to downscale it to recover the 8×8 matrix describing $\mathbf{K}_{4\to3}^{(t)\to(t')} w_4^{(t)}$.
- the case $j' = 4$ is analogous to $3 \to 3$, using 31×31 filters $\widetilde{\mathbf{K}}_{4\to4}^{(t)\to(t')}$.
- the case $j' = 5$ is analogous to $3 \to 4$: we first perform upsampling and then convolution.

Finally, for $j = J_1 = 5$,

- if $j' = 3$, then we first compute the convolution between $\widetilde{\mathbf{K}}_{5\to3}^{(t)\to(t')} \in \mathbb{R}^{63\times63}$ and $W_5^{(t)} \in \mathbb{R}^{32\times32}$ and then downsample twice.
- if $j' = 4$, we only downsample once, as in the case $4 \to 3$.
- if $j' = 5$, we only do convolution, as in the cases $3 \to 3$ and $4 \to 4$, but with 63×63 filters.

A graphical visualization of these operations is provided by Figure 2.

3. The last step consists of collecting, for each subband $j', (t')$, all the contributions coming from the vectors $w_j^{(t)}$. Thanks to the previous step, among the 100 computed matrices, all the ones associated to those contributions have dimensions $2^{j'} \times 2^{j'}$. By adding them up we recover the $j', (t')$ subband of the vector $\mathbf{K}w$ (see Figure 3).

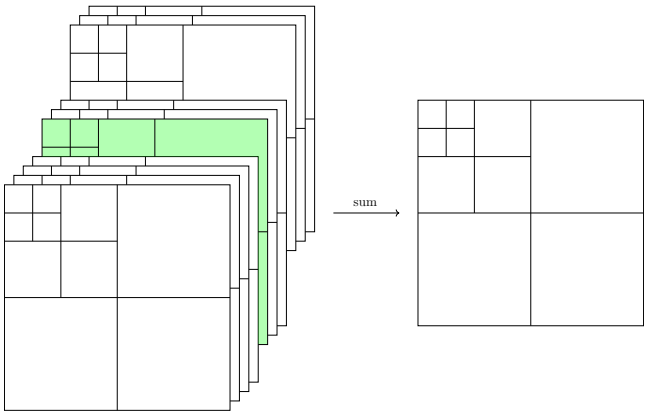


Fig. 3: Interpretation of (3.7). Step 3: reassembling each wavelet subband.

3.3. On the possibility to use smaller filters. When designing a CNN, it is common practice to employ a large numbers of convolutional filters of small size. In the architecture determined by (3.7) and (3.8), the required number of filters is exactly $(3(J-J_0)+1)^2$, and each part of the vector $w^{(n-1)}$ interacts only with $(3(J-J_0)+1)$ of them. Moreover, the size of each filter must be equal to $(2^{j'+1}-1)(2^{j+1}-1)$. We now consider the effect of substituting such large filters with smaller ones.

We would like to use filters of size $\tau \times \tau$, being $\tau = (2\xi+1)$ and $\xi > 1$, obtained by extracting the central elements of the large filters $\widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')}$. In particular, we define $\widetilde{\mathbf{K}}^{\tau} = (\widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')})_{\tau}$, being $\tau = 2\xi+1$, as

$$\left[\widetilde{\mathbf{K}}^{\tau}\right]_d = \begin{cases} \left[\widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')}\right]_d & if\ \|d\|_\infty \le \xi, \\ 0 & if\ \|d\|_\infty > \xi. \end{cases} \tag{3.9}$$

We claim that this modification is equivalent to performing a perturbation of ISTA of the type treated in Proposition 2.5, where the parameter ρ is a suitable function of τ. Although providing a detailed proof of this would entail cumbersome computation, we prove the most important result which is required to accomplish this task: we exhibit a bound on the coefficients of the filters which are discarded due to (3.9).

Such an estimate can be obtained by assuming further hypotheses on the operator A. In particular, suppose that A^*A is a convolutional operator of kernel K (as in (3.3)) and, in addition, that for $x \neq x'$ the kernel $K(x,x')$ is smooth and such that

$$K(x,x') \le \frac{C}{|x-x'|} \qquad |\nabla_x K(x,x')| + |\nabla_{x'} K(x,x')| \le \frac{C}{|x-x'|^2}. \tag{3.10}$$

It is easy to verify that (3.10) is satisfied whenever A^*A is a ΨDO of order -1 with constant coefficients, that is

$$A^*Af = \mathcal{F}^{-1}\left\{a(\xi)\mathcal{F}\left\{f\right\}(\xi)\right\}, \qquad a(\xi) \sim \frac{1}{|\xi|}\ as\ \xi \to 0.$$

We also assume first-order vanishing moment property for wavelet basis functions:

$$\int_{\mathbb{R}^2} \psi_I(x)dx = 0. \tag{3.11}$$

Such property is verified even by 2D Haar wavelets, apart from the type (f).

PROPOSITION 3.3. *Let the operator A satisfy (3.3) and (3.10). Let the indices I, I' denote two wavelets of scales j, j', type $(t), (t')$ and offsets k, k'. Let ψ_I and $\psi_{I'}$ satisfy (3.11) and let $d_{I,I'}$ be the distance between the supports of ψ_I and $\psi_{I'}$. Whenever $d_{I,I'} > 0$, it holds:*

$$\mathbf{K}_{I,I'} = (A^*A\psi_I, \psi_{I'})_X \leq c\frac{2^{-2(j+j')}}{d_{I,I'}^3} \tag{3.12}$$

We remark that the decay reported in (3.12) closely resembles formula (9.22) in [10] (according to the choice $n = 2$, $\widetilde{d} = 1$, $r = 2t = -1$) and with minor changes also formula (4.26) in [5] (with $M = 2$).

Proof. According to (3.10), and to (3.11), for any choice of $x_0 \in \text{supp } \psi_I$, $x_0' \in \text{supp } \psi_{I'}$ there exists two points ξ, ξ' in the same supports such that

$$\begin{aligned}
\mathbf{K}_{I,I'} &= \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} (K(x,x') - K(x,x_0'))\psi_I(x)\psi_{I'}(x')dxdx' \\
&\leq \int_{\mathbb{R}^2}\int_{\mathbb{R}^2} |\nabla_x K(x,\xi')||x' - x_0'|\psi_I(x)\psi_{I'}(x')dxdx' \\
&\leq C\int_{\mathbb{R}^2}\int_{\mathbb{R}^2} \frac{|x'-x_0'|}{|x-\xi'|^2}\psi_I(x)\psi_{I'}(x')dxdx' \leq C\int_{\mathbb{R}^2}\int_{\mathbb{R}^2} \frac{|x-x_0||x'-x_0'|}{|\xi-\xi'|^3}\psi_I(x)\psi_{I'}(x')dxdx'.
\end{aligned}$$

The quantity $|\xi - \xi'|$ is bounded from below by $d_{I,I'}$ by definition. Moreover, $|x - x_0| \leq \text{diam (supp } \psi_I) = c2^{-j}$, and finally $\int_{\mathbb{R}^2} \psi_I(x) \leq 2^j|\text{supp } \psi_I| = 2^{-j}$ (analogous arguments hold on I'). ∎

In view of (3.12) and of (3.8), we can easily obtain a bound on the elements of the convolutional filters:

$$\left[\widetilde{\mathbf{K}}_{j\to j'}^{(t)\to(t')}\right]_d \leq c\frac{2^{-\widehat{j}}}{(\|d\|_\infty - 1)^3},$$

provided that $\|d\|_\infty > 1$. This result, together with (3.7), allows to obtain an explicit bound (in the form of (2.12)) on the perturbation induced by the thresholding (3.9).

4. ΨDONet: formulation and theoretical results.

In this section we introduce a reconstruction algorithm for sparsity-promoting regularization based on CNNs, which leads to a novel network architecture defined ΨDONet. We report the general idea inspiring such a technique, taking advantage of the theoretical results obtained in section 3 and providing a comprehensive interpretation. Eventually, we provide a theoretical result ensuring the convergence of the proposed algorithm.

4.1. ΨDONet: a network to learn pseudodifferential operators.

Inspired by the results of the section 3, if the operator A^*A is of convolutional type, we define a reconstruction algorithm by designing a CNN of N layers, each of which is described by (3.1). In particular, the bias vectors appearing in (3.1) are $b^{(n)} = WA_{p,q}^*m$ for each n, whereas the linear operators $K^{(n)}$ are interpreted as a combination of upscaling, downscaling and convolution as described in (3.7). As shown in Proposition 3.1, if the entries of the convolutional filters are selected as is (3.8), this procedure is equivalent to performing N iterations of ISTA. Instead, the key idea of the proposed algorithm is to split the convolutional filters into two parts: a central $\tau \times \tau$ submatrix (where τ is a predefined hyper-parameter of the algorithm) and the outer frame. For each

one of the $3(J - J_0) + 1$ filters required by each layer, we suppose that the entries in the external frame are specified according to (3.8), whereas the central entries are considered as parameters, to be learned throughout the training process. Such parameters are collected in a vector θ_n (related to the n-th layer) and ultimately stored in the vector θ, possibly together with other learnable parameters. The obtained network is denoted as f_θ^τ: the aim of a CNN-based algorithm is to find a parameter θ such that the network is a good approximation of the solution map of our inverse problem, taking as an input the measurements m and giving as an output the solution $w^\dagger = Wu^\dagger$.

It is evident that, among the possible choices of the optimal parameter, the network could select the vector θ_0 which exactly replicates the ISTA iterations (it is the one for which, in every layer, also the central entries of each filter are specified by (3.8)). Nonetheless, if the optimal choice of θ differs from θ_0, it means that the network is learning something more than the ISTA iterations associated to the operator A^*A. This can be meaningfully interpreted as follows: in each layer, the network f_θ^τ applies the filters associated to an operator whose kernel is $K_0 + K_1$, where K_0 is the kernel of A^*A and K_1 is the kernel of another, *learned*, operator. Since the difference will only occur in the central elements of the convolutional filters, according to the analysis of subsection 3.3, we can argue that the learned operator is indeed a suitable approximation of a *pseudodifferential* operator. This finally allows to motivate the name we propose for this novel CNN-based reconstruction algorithm: **ΨDONet**.

There are several reasons for which the learning process could attain a better result than the one provided by ISTA. Indeed, a better choice of the parameters allows to reduce numerical errors induced by the discrete representation of A^*A, which might have a significant effect due to the error propagation among the iterations. Moreover, we might also mitigate model errors in the definition of the operator A itself. Finally, this perturbation could provide a representation of A^*A with respect to a slightly different basis, which allows to better satisfies the sparsity assumption on the solutions. For such reasons, the use of ΨDONet is specifically recommended whenever the original operator A^*A is a ΨDO itself. Indeed, its kernel representation by means of convolutional filters might benefit from learned corrections in all its most important entries: namely, the central ones.

We will show that ΨDONet is also highly recommended for FIOs: in this case, the largest entries of the convolutional filters representing A^*A are located in the center and along some lines, possibly stretching away from the center. This is the case of the limited-angle Radon transform (deeply analyzed in the following sections), which is associated with the kernel

$$K(x, y) = \frac{1}{|x - y|}\chi_\Gamma(x - y),$$

being χ_Γ the indicator function of the cone in $\mathbb{R}^2$ between the angles $-\Gamma$ and Γ. As reported in section 5, the convolutional filters related to this operator show large values only in the central elements and along two lines having the same slope of the ones delimiting the cone. This provides a curious shape for the filters, which resemble a *bowtie*. We will show that the application of ΨDONet on this operator, providing learned corrections only to the center of the bowties, is still extremely effective.

In addition to the numerical verification of the previous statements (depicted in section 6), we provide here a theoretical argument to explain why ΨDONet is expected to outperform ISTA even when A^*A is a FIO.

4.2. A theoretical justification for ΨDONet using microlocal analysis. If a ΨDO correction is learned for the normal operator, the modified ISTA iterations read as

$$w^{(n)} = \mathcal{S}_{\lambda/L}\left(w^{(n-1)} - \frac{1}{L}W(A^*_{p,q}A_{p,q} + \Lambda_p)W^* w^{(n-1)} + \frac{1}{L}WA^*_{p,q}m\right),$$

being Λ_p the discrete representation of a ΨDO. This is equivalent to performing the standard ISTA iterations on a modified version of a minimization problem (2.5), namely

$$\min_{w\in\ell^1(\mathbb{N})} \|(A^*_{p,q}A_{p,q} + \Lambda_p)^{1/2}W^*w - (A^*_{p,q}A_{p,q} + \Lambda_p)^{-1/2}A^*_{p,q}m\|^2_X + \lambda\|w\|_{\ell^1}.$$

This minimization problem is a discretised version of the continuous minimization problem

$$\min_{u\in Z} \|(A^*A + \Lambda)^{1/2}u - (A^*A + \Lambda)^{-1/2}A^*m\|^2_Y + \lambda\|u\|_Z.$$

Eventually, this accounts to finding a regularized solution, with a regularization penalty promoting solutions for which $w = Wu \in \ell^1(\mathbb{N})$ is sparse, of the problem

$$(A^*A + \Lambda)^{1/2}u = (A^*A + \Lambda)^{-1/2}A^*m$$

or equivalently,

$$(A^*A + \Lambda)u = A^*m \tag{4.1}$$

Assume next that A^*A is a FIO, that defines a bounded map between Sobolev spaces in a ball $B(R)$ of radius R and that there is $r \in \mathbb{R}$ such that for all $s \in \mathbb{R}$, $A^*A : H^s_0(B(R)) \to H^{s+r}(B(R))$. Let Λ be a (possibly unbounded) selfadjoint, positive definite, and invertible operator $\Lambda : L^2(B(R)) \to L^2(B(R))$. Moreover, assume that Λ is given by an elliptic ΨDO of order h and $h > r$. Then the operator $A^*A\Lambda^{-1}$ is an operator smoothing Sobolev spaces by order $r + h$, that is, $A^*A\Lambda^{-1} : H^s(B(R)) \to H^{s+r+h}(B(R))$ for all $s \in \mathbb{R}$. Moreover, the operator $(A^*A + \Lambda)^{-1}$ can be written as

$$(A^*A + \Lambda)^{-1} = \Lambda^{-1}(I + A^*A\Lambda^{-1})^{-1} = \Lambda^{-1}\left(\sum_{k=0}^{\overline{k}}(-A^*A\Lambda^{-1})^k + R_{\overline{k}}\right)$$

where $R_{\overline{k}} : H^s \to H^{s+\overline{k}(r+h)}$ is bounded for all $s \in \mathbb{R}$. Thus the solution u_Λ of (4.1) can be written as

$$u_\Lambda = \sum_{k=0}^{\overline{k}} u_k + \Lambda^{-1}R_{\overline{k}}A^*m, \quad u_k = -\Lambda^{-1}(A^*A\Lambda^{-1})^k = (-\Lambda^{-1}A^*A)^k\Lambda^{-1}A^*m.$$

Note that here the operator $\Lambda^{-1}A^*A$ is a Fourier integral operator whose canonical relation is determined by A and whose symbol is determined by both A and Λ. The training of the neural network can be considered as optimizing Λ so that for a given datum m the solution u_Λ of (4.1) is close to $u^\dagger$. Roughly speaking, this means to optimize Λ so that the imaging artifacts in terms u_k, caused by iteration of the operator $\Lambda^{-1}A^*A$, are minimized. Note that the remainder term $\Lambda^{-1}R_{\overline{k}}A^*m$ becomes smoother when $\overline{k}$ grows.

We are interested in applying this argument in limited-angle tomography. As a first step, we start by considering the case when X-rays are measured only from finitely many directions $\omega_j \in S^1$, $j = 1, \dots, J$. As it can be seen, the normal operator obtained via back-projection is associated to the kernel

$$K(x, y) = \sum_{j=1}^{J} \delta((x - y) \cdot \omega_j^{\perp}), \tag{4.2}$$

where $\delta \in \mathcal{D}'(\mathbb{R})$ is the Dirac delta distribution. In this case, $K = A^*A \in I^{\mu}(C'_K)$ is a FIO of order $\mu = -\frac{1}{2}$ and its canonical relation $C_K = \bigcup_{j=1}^{J} C_j$, where

$$C_j = \{(x, \xi; y, \eta) \in (T^*\mathbb{R}^2 \setminus 0) \times (T^*\mathbb{R}^2 \setminus 0) \mid \xi = \eta,\ \eta \perp \omega_j,\ (x - y) \cdot \omega_j = 0\},$$

and $K : H_0^s(B(\mathbb{R})) \to H^{s+1/2}(B(\mathbb{R}))$ for all $s \in R$. This entails that if (y, η) is in the wavefront set of u, then the elements (x, ξ) in the wave front set of Ku satisfy $(x, \xi; y, \eta) \in C_K$, that is, the operator K moves singularities along C_K. This provides a theoretical justification for the appearance of the well-known streaking artifacts in sparse tomography, see [29, 42]. The presence of a ΨDO Λ might affect the symbol of the operator, but not the canonical relation of the operator $\Lambda^{-1}A^*A$. This accounts to say that the strength of the singularities of $u_k = (-\Lambda^{-1}A^*A)^k\Lambda^{-1}A^*m$ can be reduced by Λ, without transporting them.

We finally consider the limited-angle tomography problem, in which the the operator K is defined as in (1.2). It is possible to show that such operator belongs to a class of generalized Fourier Integral operator whose properties are studied *e.g.* in [21, 20]. In addition, K can be treated as a FIO if considered far from the diagonal: namely, if a smooth truncation function ϕ is introduced such that $\phi \in C_0^{\infty}(\mathbb{R}^2)$ and ϕ vanishes near zero, the kernel $\phi(x - y)K(x, y)$, where K is given in (1.2), defines a FIO. We can thus apply the strategy proposed in subsection 4.1 by supposing to split the kernel $K = K_0 + K_1$, where now K_0 is the kernel of A^*A away from the diagonal and K_1 is the kernel of Λ, a learned correction concentrated on the diagonal. Analogously as above, we see that the operator Λ changes the strength of the artifacts appearing in $u_k = (-\Lambda^{-1}A^*A)^k\Lambda^{-1}A^*m$ but not their locations. Therefore, even in the limited-angle problem, the ΨDO correction can be seen as a regularization technique, from a microlocal analysis standpoint.

4.3. A convergence result. We now provide a theoretical result which holds true for the ΨDONet algorithm, regardless of its specific implementation. In analogy with the approach of [14], we introduce the following probabilistic approach. Let $\mathcal{B} = \{u \in X_p :\ Wu \in \ell^1(\mathbb{N}); \|Wu\|_{\ell^1} \leq C_{\mathcal{B}}\}$ and u be a random variable having a probability distribution μ on the space $\mathcal{B}$. We can consider μ as some prior information on the solution of the inverse problem. Moreover, let ϵ be a random variable in Y_q with distribution ν, which models the error on the measurements. Assume that u and ϵ are independent: hence, the measurement $m = A_{p,q}u + \epsilon$ is a random variable on the product space $X_p \times Y_q$ with density $A_*\mu \otimes \nu$, where $A_*\mu$ denotes the pushforward of μ to Y via the linear map A. In order to measure the performance of the network f_θ^τ, we introduce the loss function associated to the network f_θ^τ as:

$$\mathcal{L}(\theta; \mu, \nu) = \mathbb{E}_{u \sim \mu, \epsilon \sim \nu} \left[\|f_\theta^\tau(A_{p,q}u + \epsilon) - Wu\|_{\ell^2}^2 \right] \tag{4.3}$$

We define the optimal Neural Network as the one associated to θ^* satisfying:

$$\theta^* = \arg\min_{\theta \in \Theta} \mathcal{L}(\theta; \mu, \nu). \tag{4.4}$$

Before focusing on the properties of the optimal network $f^\tau_{\theta^*}$, it is convenient to recall that, for a specific choice of parameters θ_0, the network $f^\tau_{\theta_0}$ is equivalent to performing N iterations of (modified) ISTA. The following rough estimate will be useful:

LEMMA 4.1. *There exist two constants $k_1, k_2 > 0$ (depending on $C_{\mathcal{B}}, L, \rho$, $\|A_{p,q}\|$, $w^{(0)}$,N) such that, for all $u \in \mathcal{B}$ and $\epsilon \in Y_q$,*

$$\|f^\tau_{\theta_0}(A_{p,q}u + \varepsilon) - Wu\|_{\ell^2} \leq k_1 + k_2\|\epsilon\|_{Y_q} \tag{4.5}$$

Proof. According to (3.1), defining $\kappa = 1 + \frac{\|A^*_{p,q}A_{p,q}\|+\rho}{L}$, we get

$$\begin{aligned}\|f^\tau_{\theta_0}(A_{p,q}u + \varepsilon) - Wu\|_{\ell^2} &\leq \|f^\tau_{\theta_0}(A_{p,q}u + \varepsilon)\|_{\ell^2} + \|u\|_{X_p} \\ &\leq \kappa^N\|w^{(0)}\| + \left(1+\kappa+\ldots+\kappa^{N-1}\right)\|A_{p,q}u + \epsilon\|_{Y_q} + C_{\mathcal{B}} \\ &\leq \kappa^N\|w^{(0)}\| + C_{\mathcal{B}} + \frac{\kappa^N - 1}{\kappa - 1}(\|A_{p,q}\|C_{\mathcal{B}} + \|\epsilon\|_{Y_q})\end{aligned}$$

∎

We now focus on the case in which ϵ is a Gaussian random vector, i.e., $\nu = N(0, \sigma^2 I_q)$, being I_q the identity matrix in $\mathbb{R}^{q\times q}$. In this case, it is useful to recall that

$$\mathbb{E}[\|\varepsilon\|^2_{Y_q}] = q\sigma^2, \qquad \mathbb{E}[\|\varepsilon\|^4_{Y_q}] \leq 3q^2\sigma^4. \tag{4.6}$$

In addition to Lemma 4.1, we can rely on the results reported in section 2 (and in particular on Theorem 2.6) to provide a more refined estimate. Indeed, we observe that the convergence result reported in (2.13) is independent of the choice of $\epsilon = m - Au^\dagger$, as long as $\|\epsilon\| \leq \delta$. Moreover, the constant c_5 appearing in (2.13) can depend on $u^\dagger$, but only through an upper bound on $\|w^\dagger\|_{\ell^1}$ (see, in particular, [17, Theorem 1] and to [6, Theorem 25] for the constant derived from Proposition 2.2 and Proposition 2.5, respectively). This allows us to conclude that

LEMMA 4.2. *Suppose $\epsilon \sim N(0, \sigma^2 I_q)$ and let $\delta = \sigma^{1/\eta}$, being $\eta > 1$. There exists $\sigma_0 > 0$ such that, for $\sigma < \sigma_0$, then for every $u \in \mathcal{B}$*

$$\mathbb{E}_{\varepsilon\sim\nu}\left[\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^2_{\ell^2}\right] \leq c_5^2\delta^2 + 2\sqrt{2}k_1^2\delta^{\eta-1} + 2\sqrt{6}k_2^2 q\delta^{3\eta-1}.$$

If, moreover, $\eta = 3$ and $\sigma < \min\{\sigma_0, q^{-1/2}\}$, then there exists a constant c^* (depending on c_5, k_1, k_2) such that

$$\mathbb{E}_{\varepsilon\sim\nu}\left[\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^2_{\ell^2}\right] \leq c^*\delta^2. \tag{4.7}$$

Proof. We start by considering that

$$\begin{aligned}\mathbb{E}_{\varepsilon\sim\nu}\left[\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^2_{\ell^2}\right] &= \int_{Y_q}\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^2_{\ell^2}d_\nu(\epsilon) \\ &= \int_{\|\varepsilon\|<\delta}\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^2_{\ell^2}d_\nu(\epsilon) + \int_{\|\varepsilon\|>\delta}\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^2_{\ell^2}d_\nu(\epsilon).\end{aligned}$$

We now employ (2.13) on the first term and Hölder inequality on the second term. Moreover, in view of Chebyshev's inequality, $\nu(\{\|\varepsilon\| > \delta\}) \leq \frac{\sigma^2}{\delta^2}$. Therefore,

$$\begin{aligned}\mathbb{E}_{\varepsilon\sim\nu}\left[\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^2_{\ell^2}\right] &\leq c_5^2\delta^2\left(1 - \frac{\sigma^2}{\delta^2}\right) + \frac{\sigma}{\delta}\left(\mathbb{E}_{\varepsilon\sim\nu}\left[\|f^\tau_{\theta_0}(Au + \epsilon) - Wu\|^4_{\ell^2}\right]\right)^{\frac{1}{2}} \\ &\leq c_5^2\delta^2 + \frac{\sigma}{\delta}\left(8k_1^4 + 8k_2^4\mathbb{E}\left[\|\epsilon\|^4_{Y_q}\right]\right)^{\frac{1}{2}}.\end{aligned}$$

By (4.6) and by $\sigma = \delta^\eta$ we immediately verify the first thesis, and imposing $\eta = 3$ and $\delta^{2\eta}q < 1$ we get (4.7) with $c^* = c_5^2 + 2\sqrt{2}k_1^2 + 2\sqrt{6}k_2^2$. ∎

In view of Lemma 4.2, we can easily prove the following convergence result regarding the the optimal network $f^{\tau}_{\theta^*}$:

PROPOSITION 4.3. *Consider* $\varepsilon \sim N(0, \sigma^2 I_q)$ *with* $\delta = \sigma^{1/3}$ *and let* θ^* *satisfy* (4.4). *There exists* $\sigma_1 > 0$ *such that, for* $\sigma \leq \min\{\sigma_1, q^{1/2}\}$, *it holds*

$$\mathcal{L}(\theta^*; \mu, \nu) \leq c^* \delta^2. \tag{4.8}$$

This also accounts to say that the random variable $f^{\tau}_{\theta^*}(A_{p,q}u + \epsilon)$ *converges to* Wu *in mean as* $\delta \to 0$.

Proof. By definition of θ^* and by Lemma 4.2,

$$\begin{aligned}\mathcal{L}(\theta^*; \mu, \nu) \leq \mathcal{L}(\theta_0; \mu, \nu) &= \int_{\mathcal{B}} \int_{Y_q} \|f^{\tau}_{\theta_0}(A_{p,q}u + \epsilon) - Wu\|^2_{\ell^2} d_\nu(\epsilon) d_\mu(u) \\ &\leq \int_{\mathcal{B}} c^* \delta^2 d_\mu(u) = c^* \delta^2.\end{aligned}$$

∎

Although the optimal network $f^{\tau}_{\theta^*}$ allows for a precise approximation of the solution map of the inverse problems, it is impractical to solve the minimization problem stated in (4.4). Instead, Neural Network algorithms require to draw a sample from the random variables U and E and to find the parameter θ which allows for the best reconstruction on such sample (training process). This task is addressed by minimizing a discretized loss functional, as reported in subsection 5.4, and results in the definition of the *trained* Neural Network. The quality of the trained network can be verified by analyzing its *generalization*, namely, its ability to provide good predictions even when tested on data outside the training sample. Such an analysis has been performed in detail (although with some different assumptions with respect to the ones in this work) in [14], and can be extended also to the problem in consideration.

5. In practice: the particular case of CT. In this section, we focus on the practical aspects of the reconstruction algorithm introduced in subsection 4.3, in the particular case of limited-angle computed tomography (LA-CT) with the discrete setting. In the remainder of the article, the discrete counterpart of the operator $A_{p,q}$ representing the LA-CT will be denoted by R_Γ. We first define the regularized minimization problem, and then propose an effective method for the computation of the convolutional kernel filters approximating the backprojection operator in the wavelet domain. Thirdly, we present and discuss the general reconstruction workflow and finally give more details on the two CNN architectures we propose in this paper.

5.1. The CT minimization problem. After suitable discretization, we are given the measurements (*i.e.*, the so-called sinogram or observed image) $\mathbf{m} \in \mathbb{R}^q$ such that:

$$\mathbf{m} = \mathbf{R}_{\mathbf{\Gamma}} \mathbf{u}^\dagger + \epsilon, \tag{5.1}$$

where $\mathbf{u}^\dagger \in \mathbb{R}^p$ denotes the (unknown) discrete and vectorized image, $\mathbf{R}_{\mathbf{\Gamma}} \in \mathbb{R}^{q \times p}$ describes a discretized version of the Radon transform where the angles are limited in the arc specified by Γ and $\epsilon \in \mathbb{R}^q$ models the measurement noise. We call $\mathbf{w}^\dagger$ the $\mathbb{R}^p$-vector such that $\mathbf{W}\mathbf{u}^\dagger = \mathbf{w}^\dagger$ where $\mathbf{W} \in \mathbb{R}^{p \times p}$ represents a discretization of the wavelet transform. Thus, the regularized minimization problem is given by:

$$\min_{\mathbf{w} \in \mathbb{R}^p} \|\mathbf{R}_{\mathbf{\Gamma}} \mathbf{W}^* \mathbf{w} - \mathbf{m}\|_2^2 + \lambda \|\mathbf{w}\|_1 \tag{5.2}$$

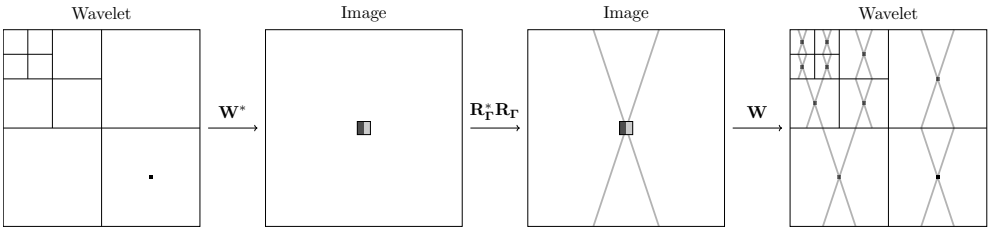


Fig. 4: Illustration of the proposed way to compute the filters of the convolutional kernel operator K in the LA-CT case. The initial object (on the left) is created such that all its pixels but one are set to zero. The only non-zero pixel is located at the center of one of its wavelet subbands. By applying the operator $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{R}_\Gamma\mathbf{W}^*$ to this initial object, one obtains a new object in the wavelet domain, whose subbands present a bowtie-shaped structure. The set of 'bowtie' subbands thus computed from all the possible initial objects constitute the filters of the convolutional kernel operator. Here we have represented three levels of decomposition in the wavelet domain, meaning that the total number of convolutional filters amounts to $(3^2+1)^2 = 100$.

Our recovery algorithm for finding a reconstruction $\mathbf{u}$ of $\mathbf{u}^\dagger$ involves convolutional architectures incorporated in the iterative structure of standard ISTA, as described in section 3. In the next paragraphs, we detail the implementation of such an algorithm.

5.2. Convolutional kernel operator for limited-angle CT. Building a convolutional algorithm that reproduces the behavior of standard ISTA first requires to identify the various blocks of the matrix $\mathbf{K}$ representing the backprojection operator in the wavelet domain. In other words, the very first step in the development of our method is to establish the convolutional filters of $\mathbf{K}$ which, once applied as defined in (3.7), provide a reliable approximation of the operator $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{R}_\Gamma\mathbf{W}^*$.

One way to compute such convolutional filters that proves to be a numerically advantageous alternative to (3.8), is represented in Figure 4. Let us consider an object whose representation in the wavelet domain has only one nonzero pixel, located at the center of one of its wavelet subbands. Applying the operator $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{R}_\Gamma\mathbf{W}^*$ to this initial object leads to a new object whose subbands present a bowtie-shaped structure. Those 'bowtie' subbands constitute a first set of convolutional filters. By reiterating this operation until the central pixel of all the wavelet subbands in the initial object has been visited, one obtains the entire collection of convolutional filters necessary for the approximation of $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{R}_\Gamma\mathbf{W}^*$. A numerical example of convolutional filter is shown in Figure 5.

In order to imitate the behavior of the operator $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{R}_\Gamma\mathbf{W}^*$, the convolutional filters so computed are then to be applied to the wavelet subbands of an object as illustrated in Figure 6. First, each wavelet subband of the object of interest is replicated $3(J-J_0)+1$ times. Those replicas are either upsampled, or downsampled, or kept with the same dimensions, depending on the scale of the filter they are to be convolved with. The set of convolutional filters used on the replicas of a particular wavelet subband of scale j and type (t) is the set of filters beforehand generated by applying the operator $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{R}_\Gamma\mathbf{W}^*$ to an object whose only non-zero pixel is located at the center of its wavelet subband of exact same scale j and type (t). Once the convolutions between the replicas and the filters have been performed, the resulting subbands are reassembled to form the wavelet representation of a new object. This

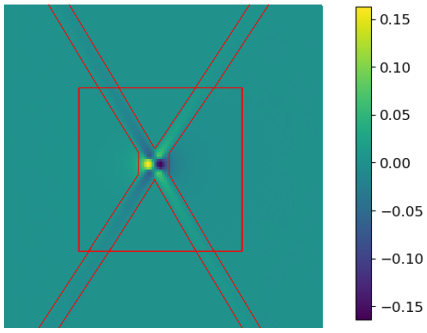


Fig. 5: Example of a 'bowtie' subband that can be used as a convolutional filter of the kernel operator K. It was generated from a 256×256 initial object, according to the procedure detailed in subsection 5.2 and illustrated in Figure 4. Theory suggests that the pixels with highest intensities are spread according to a bowtie-shaped structure. In practice, they are even more condensed: most of the energy is concentrated along two diagonal lines that intersect in the center and whose inclination is defined by the limited angle: 95.8% of the ℓ_2-norm of the filter is concentrated along those two lines, from which 94.8% are inside the central red square.

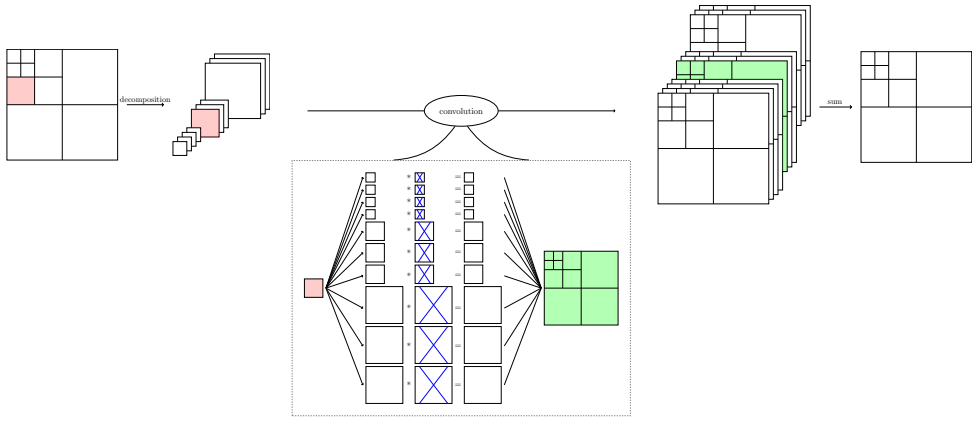
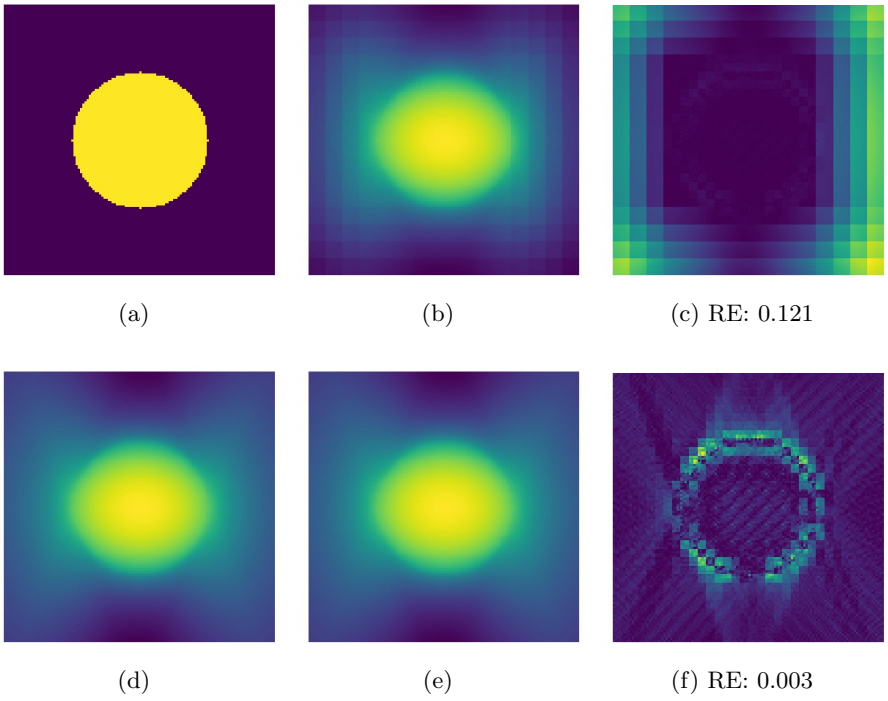


Fig. 6: Illustration of the way the filters of the convolutional kernel operator are applied to each wavelet subband of the initial object, after up- and down- sampling operations, in order to approximate the operator $\mathbf{W}\mathbf{R}_{\mathbf{\Gamma}}^{*}\mathbf{R}_{\mathbf{\Gamma}}\mathbf{W}^{*}$.

process is reiterated for all the subbands of the initial object and ultimately, the $3(J - J_0) + 1$ resulting items are summed. The final outcome is an approximation of the wavelet representation of the operator $\mathbf{W}\mathbf{R}_{\mathbf{\Gamma}}^{*}\mathbf{R}_{\mathbf{\Gamma}}\mathbf{W}^{*}$ applied to the initial object (cf Figure 7).

Two remarks are worth mentioning regarding the creation and use of the above-defined convolutional filters. First, our practical implementation very slightly differs from the theory presented in (3.7) as far as the downsampling is concerned. In our codes, downsampling is indeed applied before computing the convolution between the filter and the wavelet subband replica, and not after as it is presented in the theory.



(a) (b) (c) RE: 0.121

(d) (e) (f) RE: 0.003

Fig. 7: Illustration of the effect of the standard backprojection operator and of its approximations based on convolutional filters of different sizes. (a) shows the ground truth $\mathbf{u}^\dagger$ of interest and (d) its standard backprojection $\mathbf{R}_{\mathbf{\Gamma}}^\star \mathbf{R}_{\mathbf{\Gamma}} \mathbf{u}^\dagger$, computed with the basic functions of the Python package `scikit-image`. (b) (resp. (e)) represents the approximation $\mathbf{K}\mathbf{u}^\dagger$ obtained with convolutional filters beforehand generated from $2^J \times 2^J$ (resp. $2^{J+1} \times 2^{J+1}$) only-one-non-zero-pixel object. (c) and (f) show the absolute differences between the approximation of the backprojection operator and the expected value (d). The dynamic range of the plot is modified for better contrast.

This choice is motivated by the reduction in terms of storage needs and running time such a change allows while preserving the accuracy of the approximation. Secondly, both the theoretical analysis and the experimental tests showed that the dimensions of the convolutional filters used for the approximation of $\mathbf{W}\mathbf{R}_{\mathbf{\Gamma}}^* \mathbf{R}_{\mathbf{\Gamma}} \mathbf{W}^*$ do affect the accuracy of the results. Initially, we assumed that the convolutional filters should be generated from only-one-non-zero-pixel objects with the same dimensions $2^J \times 2^J$ as the image of interest (recall that $p = 2^{2J}$). However, we reached the conclusion that they actually have to be generated from twice bigger objects, that is of dimensions $2^{J+1} \times 2^{J+1}$, in order to get an accurate approximation of the operator $\mathbf{W}\mathbf{R}_{\mathbf{\Gamma}}^* \mathbf{R}_{\mathbf{\Gamma}} \mathbf{W}^*$. An illustration of the effects of the size of the filters can be seen on Figure 7.

5.3. Our CNN architectures. The above described method for generating and applying the filters of the kernel operator $\mathbf{K}$ makes the concrete implementation of a convolutional algorithm that imitates the behavior of standard ISTA possible. Thus, the convolutional implementation of ISTA, result-wise equivalent to the standard one,

could be written as:

$$\mathbf{w}^{(n+1)} = \mathcal{S}_{\frac{\lambda}{L}}\left(\mathbf{w}^{(n)} + \frac{1}{L}\left(\mathbf{W}\mathbf{R}_{\mathbf{\Gamma}}^{*}\mathbf{m} - \mathbf{K}\mathbf{w}^{(n)}\right)\right), \qquad n = \{0, \dots, N\} \tag{5.3}$$

This algorithm offers the merits of the iterative model-based method ISTA, while allowing the incorporation of data-driven approaches, such as machine learning and deep neural network techniques. The implementation of $\mathbf{K}$ indeed involves operations that are all perfectly adaptable to the framework of CNNs. Our goal is precisely to take full advantage of this compatibility and profit from the remarkable potentials of deep neural networks by converting the hitherto fixed operator $\mathbf{K}$ into a partially trainable CNN. Thus, the center of the convolutional filters so far precomputed with the deterministic method presented in subsection 5.2 can henceforth be considered as parameters to be learned from data. The choice of learning only the central part of the convolutional filters of $\mathbf{K}$ rather than the whole filters is motivated by the need to reduce the model complexity which, in the latter case, makes the training of the model burdensome if not impractical.

In order to further improve reconstruction performance, we also propose to learn the soft-thresholding parameter as well as the step-length so far set at $1/L$. The so-defined convolutional architecture results in our proposed algorithm ΨDONet, whose convergence results are detailed in subsection 4.3. In subsection 5.3.1 and subsection 5.3.2, we propose two different implementations of ΨDONet, that proves to be result-wise similar as it can be observed in subsection 6.2.

5.3.1. ΨDONet-F. The most natural way to implement ΨDONet consists in partitioning the convolutional operator $\mathbf{K}$ into two operators: a fixed one, $\breve{\mathbf{K}}^{\tau}$, and a trainable (single-layer) CNN referred to as Λ_{ζ}^{τ}, where τ is a tunable hyperparameter. The two operators have the exact same architecture as $\mathbf{K}$ and their sum, before any training, is strictly equivalent to $\mathbf{K}$. The first operator $\breve{\mathbf{K}}^{\tau}$ is non trainable and its filters are a copy of the filters of $\mathbf{K}$ with the exception that the $\tau \times \tau$ central weights of each filter are set to zero. The second operator Λ_{ζ}^{τ}, on the contrary, is composed by $\tau \times \tau$-trainable filters that are initialized with the $\tau \times \tau$ central part of the filters of $\mathbf{K}$ (cf Figure 8). This first implementation of ΨDONet, referred to as Filter-Based ΨDONet or ΨDONet-F, is formulated as:

$$\mathbf{w}^{(n+1)} = \mathcal{S}_{\gamma_n}\left(\mathbf{w}^{(n)} + \alpha_n\left(\mathbf{W}\mathbf{R}_{\mathbf{\Gamma}}^{*}\mathbf{m} - \beta_n\left(\breve{\mathbf{K}}^{\tau}\mathbf{w}^{(n)} + \Lambda_{\zeta_n}^{\tau}\mathbf{w}^{(n)}\right)\right)\right) \tag{5.4}$$

where $n = \{0, \dots, N\}$ and the parameters to be learned are $\{\gamma_0, \alpha_0, \beta_0, \zeta_0, \dots, \gamma_N, \alpha_N, \beta_N, \zeta_N\}$. The parameters $\{\beta_0, \dots, \beta_N\}$ have been added in such a way that the influence of the fixed operator $\breve{\mathbf{K}}^{\tau}$ with respect to the constant term $\mathbf{W}\mathbf{R}_{\mathbf{\Gamma}}^{*}\mathbf{m}$ can be adjusted in order to maximize the accuracy of the results. It is worth mentioning that for the particular choice of $\gamma_n = \frac{\lambda}{L}, \alpha_n = \frac{1}{L}, \beta_n = 1$, for any $n = \{0, \dots, N\}$, this model before any training is exactly equivalent to standard ISTA.

The trade-off between the number of parameters that can be improved through the learning process and the trainability of the model is controlled by τ. For a sound choice of such a hyperparameter, the complexity of the model is sufficiently reduced to allow for the convergence of the learning algorithm while enabling the enhancement of a significant number of weights in the filters.

This implementation has the advantage of offering a clear interpretation of the role and meaning of the convolutional filters belonging to $\breve{\mathbf{K}}^{\tau}$ and Λ_{ζ}^{τ}. Those filters

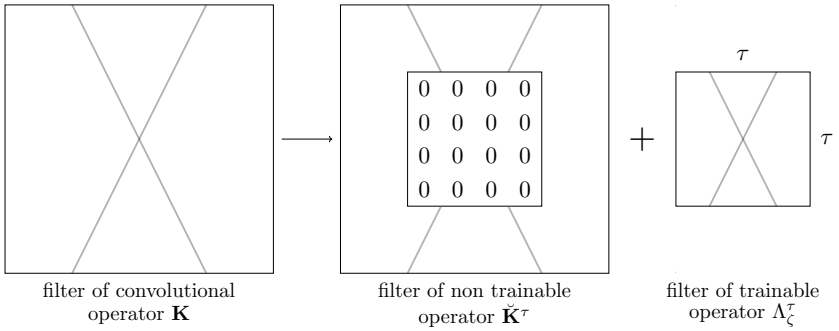


Fig. 8: Illustration of the way the convolutional filters of the two operators in ΨDONet-O are computed based on the filters of operator $\mathbf{K}$. Each filter of $\mathbf{K}$ is partitioned into two filters, which sum is equivalent to the initial one. The filter of $\breve{\mathbf{K}}^\tau$ is a copy of the filter of $\mathbf{K}$ with the exception that the $\tau \times \tau$ central weights are set to zero. The filter of Λ^τ_ζ has dimensions $\tau \times \tau$ and is initialized with the central $\tau \times \tau$ central weights of the filter of $\mathbf{K}$.

are indeed initialized with the filters of the operator $\mathbf{K}$ that imitates the behavior of $\mathbf{W}\mathbf{R}^*_{\mathbf{\Gamma}}\mathbf{R}_{\mathbf{\Gamma}}\mathbf{W}^*$. Thus, modifying their weights through the learning process can be thought of as a direct improvement of the back-projection operator.

ΨDONet-F has led to very satisfactory preliminary results, presented in section 6. However, training such a model on big images may quickly become extremely onerous in terms of running time and storage requirements. Such problems may arise while training ΨDONet-F on images of dimensions greater or equal to 256×256. Unlike typical CNNs that usually make use of small-sized convolutional filters, the filters of $\breve{\mathbf{K}}^\tau$ in our proposed algorithm are much bigger than the wavelet subbands they are convolved with. This uncommon procedure, that *inter alia* implies the padding, *i.e.*, the addition of many extra pixels to the edge of each wavelet subband, brings about a severe speed reduction in the training process as well as the necessity of a substantial memory space. The alternative implementation of ΨDONet, described in subsection 5.3.2 , addresses these shortcomings.

5.3.2. ΨDONet-O. The main flaw of ΨDONet-F rests upon the use of operator $\breve{\mathbf{K}}^\tau$ which implies numerous burdensome convolutions. This issue is worked around in ΨDONet-O (5.5), as $\breve{\mathbf{K}}^\tau$ is not involved anymore. Here, the backprojection operator is not approximated, meaning that $\mathbf{W}\mathbf{R}^*_{\mathbf{\Gamma}}\mathbf{R}_{\mathbf{\Gamma}}\mathbf{W}^*$ is indeed implemented as the succession of the inverse wavelet, Radon, inverse Radon and direct wavelet transforms applied to the iterate $\mathbf{w}^{(n)}$. This second implementation of ΨDONet, named Operator-Based ΨDONet or ΨDONet-O, reads as:

$$\mathbf{w}^{(n+1)} = \mathcal{S}_{\gamma_n}\left(\mathbf{w}^{(n)} + \alpha_n\left(\mathbf{W}\mathbf{R}^*_{\mathbf{\Gamma}}\mathbf{m} - \mathbf{W}\mathbf{R}^*_{\mathbf{\Gamma}}\mathbf{R}_{\mathbf{\Gamma}}\mathbf{W}^*\mathbf{w}^{(n)}\right) + \beta_n\Lambda^\tau_{\zeta_n}\mathbf{w}^{(n)}\right) \tag{5.5}$$

where $n = \{0, \dots, N\}$, the parameters to be learned are $\{\gamma_0, \alpha_0, \beta_0, \zeta_0, \dots, \gamma_N, \alpha_N, \beta_N, \zeta_N\}$, and $\Lambda^\tau_{\zeta_n}$ has the same architecture as the operator $\mathbf{K}$. The block diagram of the method is represented in Figure 9. For the special choice of $\gamma_n = \frac{\lambda}{L}, \alpha_n = \frac{1}{L}, \beta_n = 0$, for any $n = \{0, \dots, N\}$, this model is exactly equivalent to standard ISTA. The only convolutions involved in this alternative implementation are the ones composing

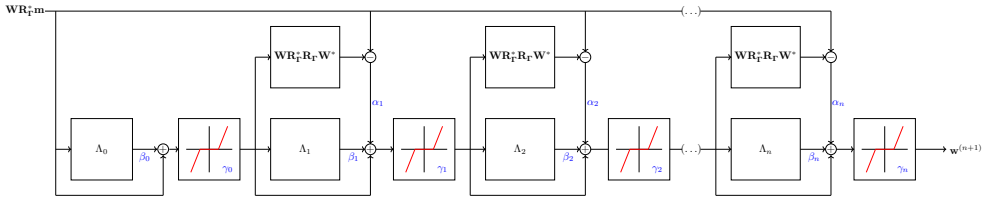


Fig. 9: Block diagram of the proposed model (5.5). Notice that the soft thresholding operator acts as a nonlinear activation function.

the CNN $\Lambda^{\tau}_{\zeta_n}$, whose filters are chosen to be small enough to avoid any running time or storage issue. In that sense, ΨDONet-O offers an implementation numerically preferable to ΨDONet-F, while retaining the same properties on a theoretical level. Furthermore, such a model keeps offering a clear interpretation of its post-processing abilities since $\Lambda^{\tau}_{\zeta_n}$, on account of its architecture, can still be seen as an adjunct for improving the back-projection operator.

5.3.3. Note on soft-thresholding parameters. From a theoretical point of view, the parameters $\gamma_0, \dots, \gamma_N$ in ΨDONet-F and ΨDONet-O have to be non-negative, as they represent the soft-thresholding parameters. In order to stick to the operator originally involved in standard ISTA, it is possible to enforce the positivity of the coefficient by replacing each γ_n by $10^{\widetilde{\gamma}_n}$, where $\widetilde{\gamma}_n$ becomes the actual trainable parameter. However, in order to allow for a greater degree of freedom in the learning process, we decided to implement the operator $\mathcal{S}_{\gamma_n}$ in such a way that it is also interpretable for negative values of its parameter γ_n. In such a case, we define the operator $\mathcal{S}_{\gamma_n<0}$ as the symmetric of the soft-thresholding curve with respect to $y = x$, while for non-negative values of γ_n, $\mathcal{S}_{\gamma_n}$ is exactly equivalent to the soft-thresholding operator. Formally, $\mathcal{S}_{\gamma_n}$ becomes:

$$\text{For } \gamma_n \geq 0: \quad \mathcal{S}_{\gamma_n}(x) = \begin{cases} x - \gamma_n, & \text{if } x \geq \gamma_n \\ 0, & \text{if } |x| < \gamma_n \\ x + \gamma_n, & \text{if } x \leq -\gamma_n \end{cases}$$

$$\text{For } \gamma_n < 0: \quad \mathcal{S}_{\gamma_n}(x) = \begin{cases} x - \gamma_n, & \text{if } x \geq 0 \\ x + \gamma_n, & \text{if } x < 0 \end{cases}$$

The two implementations ΨDONet-F and ΨDONet-O are tested with and without the positivity constraint on γ (cf results in subsection 6.2).

5.4. Supervised Learning. If we denominate f^{τ}_{θ} the N-layer CNN that, given $\mathbf{m}$, computes the final output $\mathbf{w}^{(N+1)}$ according to one of the two proposed architectures, we aim at learning the optimal high-dimensional vector $\theta = \{\gamma_0, \alpha_0, \beta_0, \zeta_0, \dots, \gamma_N, \alpha_N, \beta_N, \zeta_N\}$ that ideally satisfies the relation:

$$f^{\tau}_{\theta}(\mathbf{m}) \approx \mathbf{W}\mathbf{u}^{\dagger} \tag{5.6}$$

For a mathematical formalization, we regard the tuple $\left(\mathbf{m}, \mathbf{u}^{\dagger}\right) \in \mathbb{R}^q \times \mathbb{R}^p$ as a random variable with a joint probability distribution Ξ, as detailed in subsection 4.3. Ideally, we would like to find a parameter vector θ^* minimizing the expected risk:

$$\min_{\theta} \left(\mathbb{E}_{(\mathbf{m},\mathbf{u}^{\dagger})\sim\Xi} \| f^{\tau}_{\theta}(\mathbf{m}) - \mathbf{W}\mathbf{u}^{\dagger} \|_2^2 \right) \tag{5.7}$$

Other loss functions, such as the weighted l_2-norm, where the wavelet coefficients are weighted depending on their scale, have been tested and lead to results similar to the non-weighted l_2-norm. For the sake of brevity, we will stick to the basic form of (5.7).

In practice, computing the expectation with respect to Ξ is not possible. Instead, we are typically given a finite set of independent drawings $(\mathbf{m}_1, \mathbf{u}_1^\dagger), \dots, (\mathbf{m}_S, \mathbf{u}_S^\dagger)$ and consider the minimization of the empirical risk:

$$\min_\theta \frac{1}{S} \sum_{i=1}^{S} \| f_\theta^\tau(\mathbf{m}_i) - \mathbf{W}\mathbf{u}_i^\dagger \|_2^2 \tag{5.8}$$

Depending on the properties of f_θ^τ, the optimization problem is in general non-convex. In the case of neural networks, typically some form of gradient descent is used, where the gradients are calculated via backpropagation [47]. Computing the gradient over the entire training set in (5.8) is often not feasible for large-scale problems due to memory limitations. To circumvent this problem, stochastic of minibatch gradient descent is used, in which the gradient is approximated over smaller, randomly selected batches of training examples [19, Chapter 8]. The final performance (*i.e.*, the generalization) of the trained map f_θ^τ is evaluated on a separate set of independent drawing s, the *test set*, that were not previously used in the optimization of θ in (5.8).

6. Experiments and results. In this section, we evaluate the performance of the proposed reconstruction schemes by comparing with standard ISTA.

6.1. Preliminaries. Let us begin by describing the considered experimental scenario, the implementation of the used operators and the training procedure.

6.1.1. Data set. The data set consists of 10700 synthetic images of ellipses, where the number, locations, sizes and intensity gradients of the ellipses are chosen randomly. Using Matlab's `radon`, we simulate measurements for a missing wedge of 60° with Gaussian noise. To avoid inverse crime [36] the measurements are simulated at a higher resolution and then downsampled for an image resolution of 128×128. 10000 images are used for training, 200 images for validation and 500 for testing.

6.1.2. Operators. For the implementation of the discrete limited angle operator $\mathbf{R}_\Gamma$ we use the `radon` routine of the Python package `scikit-image` [51], or the 2D parallel beam geometry of the Operator Discretization Library (ODL) library [1], which is based on the Astra toolbox [50]. The former is employed for generating the backprojections $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{m}$ provided as inputs to ΨDONet-F and ΨDONet-O, while the latter is used for the implementation of $\mathbf{W}\mathbf{R}_\Gamma^*\mathbf{R}_\Gamma\mathbf{W}^*$ in ΨDONet-O. The direct and inverse Radon transform operators are multiplied by a constant so that their norm is equal to one. Regarding the wavelet transform, we make use of the Python package `pywt` [33] or a rectified version of the package `tf-wavelets` [23]. In all our experiments, we consider the case $J = 7$ and $J_0 = 4$, implying that the wavelet decomposition $\mathbf{W}\mathbf{u}$ has 10 subbands. For the two ΨDONet-F and ΨDONet-O, we choose to set τ to 32. Note that according to theory, τ is supposed to be odd, however, in practice we prefer it to be even. This very slight modification has no effect on the results.

6.1.3. Network structure and training. For the implementation of the two ΨDONet-F and ΨDONet-O, we fix the number of unrolled blocks N to 120. In order to reduce the number of parameters to be learned, we choose to use only 40 different sets of trainable parameters $\{\zeta_n, \gamma_n, \alpha_n, \beta_n\}$, each of which being used over 3 consecutive blocks, instead of the theoretically expected 120 sets. Implementing and training our

Method	RE	PSNR	SSIM	HaarPSI
$\mathbf{u}_{\text{ista}}$	0.44	22.84	0.36	0.37
$\mathbf{u}_{\text{FBP}}$	0.64	19.49	0.20	0.30
$\mathbf{u}^{+}_{\Psi\text{do-F}}$	0.29	26.63	0.59	0.47
$\mathbf{u}_{\Psi\text{do-F}}$	0.25	27.63	0.78	0.54
$\mathbf{u}^{+}_{\Psi\text{do-O}}$	0.28	26.76	0.60	0.48
$\mathbf{u}_{\Psi\text{do-O}}$	**0.23**	**28.43**	**0.81**	**0.58**

Table 1: Comparison of reconstruction methods. The similarity values are averaged over the 500 images of the test set.

algorithms has been performed using Tensorflow with an Adam optimizer [30] and a learning rate (step size) of 10^{-3}. The number of epochs was chosen to be 3, and the batch size was set to 25. The training, run on a NVIDIA Quadro P6000 GPU, roughly takes 20 hours. Our codes are available at https://github.com/megalinier/PsiDONet.

6.1.4. Compared Methods. We compare the preliminary results of the architectures we propose with the reconstructions provided by standard ISTA. In the implementation of the latter, we make use of the formula introduced in [12]. The regularization parameter λ and the constant L are respectively set to 2.10^{-6} and 5. The number of iterations for ISTA is determined by the stopping criterion:

$$\|\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)}\|_2^2 / \|\mathbf{u}^{(n)}\|_2^2 < tol \tag{6.1}$$

where *tol* is chosen to be 2.10^{-4}. Hereinbelow, we give a list of the abbreviations henceforth used for the different recovery methods.

- $\mathbf{u}_{\mathbf{ista}}$ Standard ISTA reconstruction.
- $\mathbf{u}_{\mathbf{FBP}}$ Standard filtered backprojection with the 'ramp' filter of `skicit-image`.
- $\mathbf{u}^{+}_{\Psi\mathbf{do\text{-}F}}$ Solution provided by ΨDONet-F with positivity-constraint on the soft-thresholding parameter ($\gamma_n = 10^{\widetilde{\gamma}_n}, \forall n$).
- $\mathbf{u}_{\Psi\mathbf{do\text{-}F}}$ Solution provided by ΨDONet-F without positivity-constraint on the soft-thresholding parameter.
- $\mathbf{u}^{+}_{\Psi\mathbf{do\text{-}O}}$ Solution provided by ΨDONet-O with positivity-constraint on the soft-thresholding parameter ($\gamma_n = 10^{\widetilde{\gamma}_n}, \forall n$).
- $\mathbf{u}_{\Psi\mathbf{do\text{-}O}}$ Solution provided by ΨDONet-O without positivity-constraint on the soft-thresholding parameter.

6.1.5. Similarity Measures. For an assessment of image quality, we are using several quantitative measures, such as the relative error (RE) given by $\|\mathbf{u}^\dagger - \mathbf{u}\|_2 / \|\mathbf{u}^\dagger\|_2$, where $u^\dagger$ denotes the reference image and u its reconstruction. Furthermore, we consider the peak signal-to-noise ratio (PSNR) and the structured similarity index (SSIM) [52] provided by Tensorflow. Finally, we are reporting the Haar wavelet-based perceptual similarity index (HaarPSI) that was recently proposed in [46].

6.2. Results. In the following, we will report and discuss the results of our numerical experiments. The average image quality measures of the 500 test images are reported in Table 1. Furthermore, a visualization of the reconstruction quality for two of the test images is given in Figure 10 and Figure 11. Due to the large missing angle of 60°, the FBP images in Figure 10c and Figure 11c are contaminated with streaking artifacts and contrast changes. The standard ISTA offers reconstructions

of higher quality (cf Figure 10d and Figure 11d), however, the streaking artifacts are still noticeable as well as the impurities due to the noise in the measurements. Besides, the ISTA reconstructions are toned down, meaning that for the most part, the intensity of the pixels remain significantly lower than the expected values. With our two models, ΨDONet-F and ΨDONet-O, whether with positivity constraint on the soft-thresholding parameter or without, it is possible to substantially reduce those artifacts and contrast issues. As it can be seen in Figure 10 and Figure 11, our proposed methods lead to undeniably enhanced reconstructions, with a meaningful diminution of the relative error. In particular, ΨDONet-O provides slightly better similarity values than ΨDONet-F, although both implementations produces comparable results.

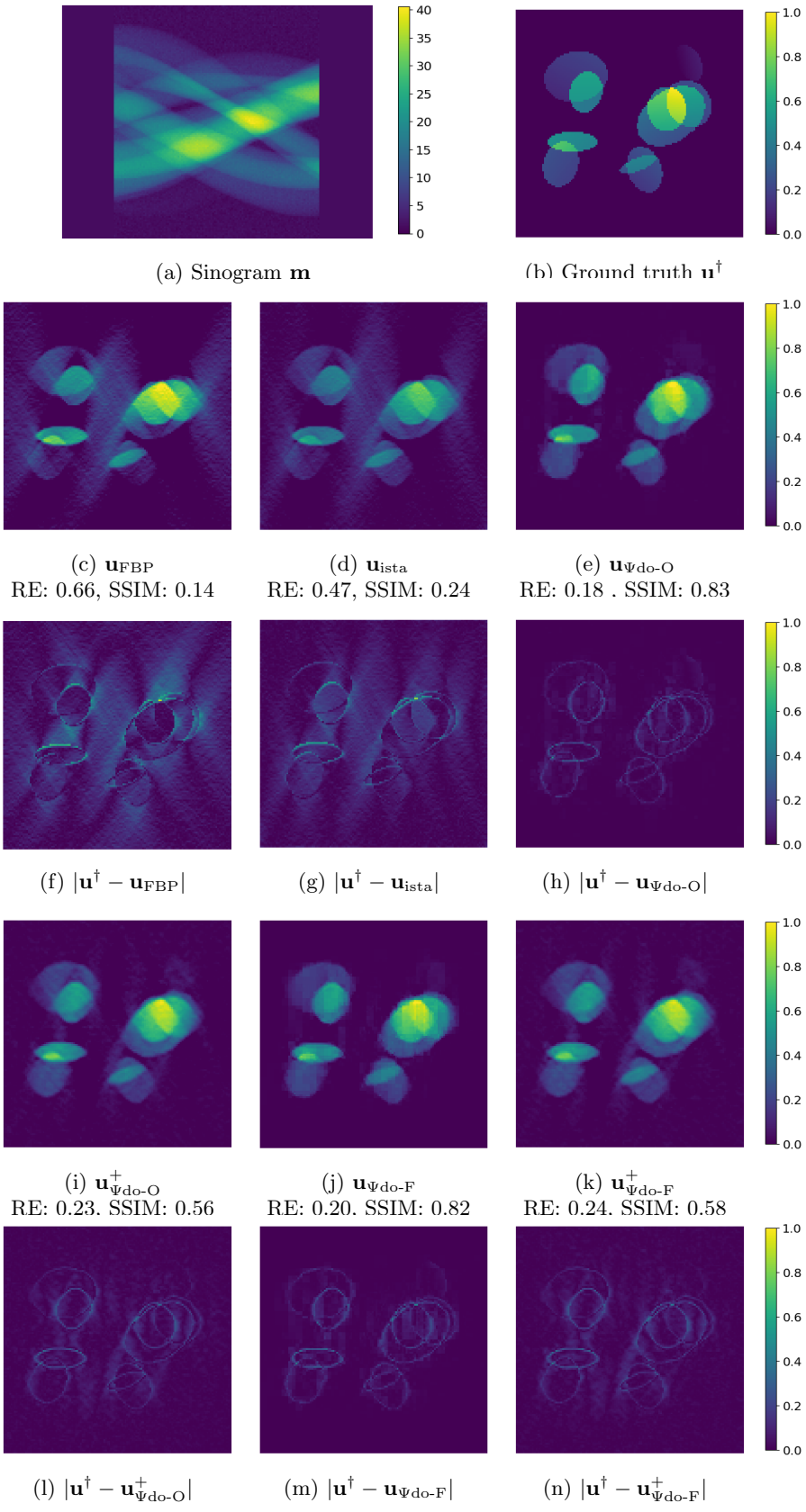
In the case where the positivity of the soft-thresholding parameter is enforced, that is for $\mathbf{u}^{+}_{\Psi\text{do-F}}$ and $\mathbf{u}^{+}_{\Psi\text{do-O}}$, one can notice that the streaking artifacts, although greatly lessened when compared with the ISTA images, are still present on the reconstructions (cf Figure 10i, Figure 10k, Figure 11i and Figure 11k). In fact, the SSIM measures are greater than in the ISTA case, but still clearly below the SSIM values obtained with the non-constrained version of the two models. The latter ($\mathbf{u}_{\Psi\text{do-F}}$ and $\mathbf{u}_{\Psi\text{do-O}}$) do a noteworthy job in removing the artifacts and sharpening the edges (cf Figure 10e, Figure 10j, Figure 11e and Figure 11j).

Overall, ΨDONet-O without any constraint on the soft-thresholding parameters offers the best results among the compared methods and allows for an optimized implementation of ΨDONet.

7. Conclusions. In the present paper, we introduced a novel CNN, named ΨDONet, inspired by the well-known ISTA and the convolutional nature of certain FIOs and ΨDOs, like the limited angle Radon transform. We proved that the unrolled iterations of ISTA can be interpreted as layers of a CNN, where the downsampling, upsampling and convolution operations, typically defining a CNN, can be exactly specified by combining the convolutional nature of the limited angle Radon transform and basic properties defining an orthogonal wavelet system. In addition, we proved that, for a specific choice of the parameters involved, ΨDONet recovers standard ISTA or a perturbation of ISTA, up to a bound on the filters coefficients which we estimated in the case of limited angle Radon transform.

The key feature of the proposed architecture is its potential to learn ΨDO-like structures, which makes it suitable to be extended to any convolutional operator which is a FIO or ΨDO. Moreover, the analysis carried out in paper allows to gain understanding and interpretability of the results, which gives insight into a whole class of inverse problems arising from FIO or ΨDO and opens up for fundamental theoretical generalization results.

As a proof of concept, we tested two different implementations of ΨDONet on simulated data from limited angle geometry, generated from the ellipse data set. The improvement, compared to standard ISTA (and classical FBP) is notable and it is promising for further numerical testing which we leave to future work. Additional directions for future numerical testing include larger sizes for images, smaller and sparser visible wedges, additional regularization for the loss function and reconstructions from real data. Also, it may be beneficial for the reconstruction to introduce more advanced features in the ΨDONet architecture, such as skip connections or other residual blocks. Incorporating such elements, while preserving full interpretability of the network, is left for future work.



(a) Sinogram $\mathbf{m}$ (b) Ground truth $\mathbf{u}^\dagger$

(c) $\mathbf{u}_{\text{FBP}}$
RE: 0.66, SSIM: 0.14

(d) $\mathbf{u}_{\text{ista}}$
RE: 0.47, SSIM: 0.24

(e) $\mathbf{u}_{\Psi\text{do-O}}$
RE: 0.18 . SSIM: 0.83

(f) $|\mathbf{u}^\dagger - \mathbf{u}_{\text{FBP}}|$ (g) $|\mathbf{u}^\dagger - \mathbf{u}_{\text{ista}}|$ (h) $|\mathbf{u}^\dagger - \mathbf{u}_{\Psi\text{do-O}}|$

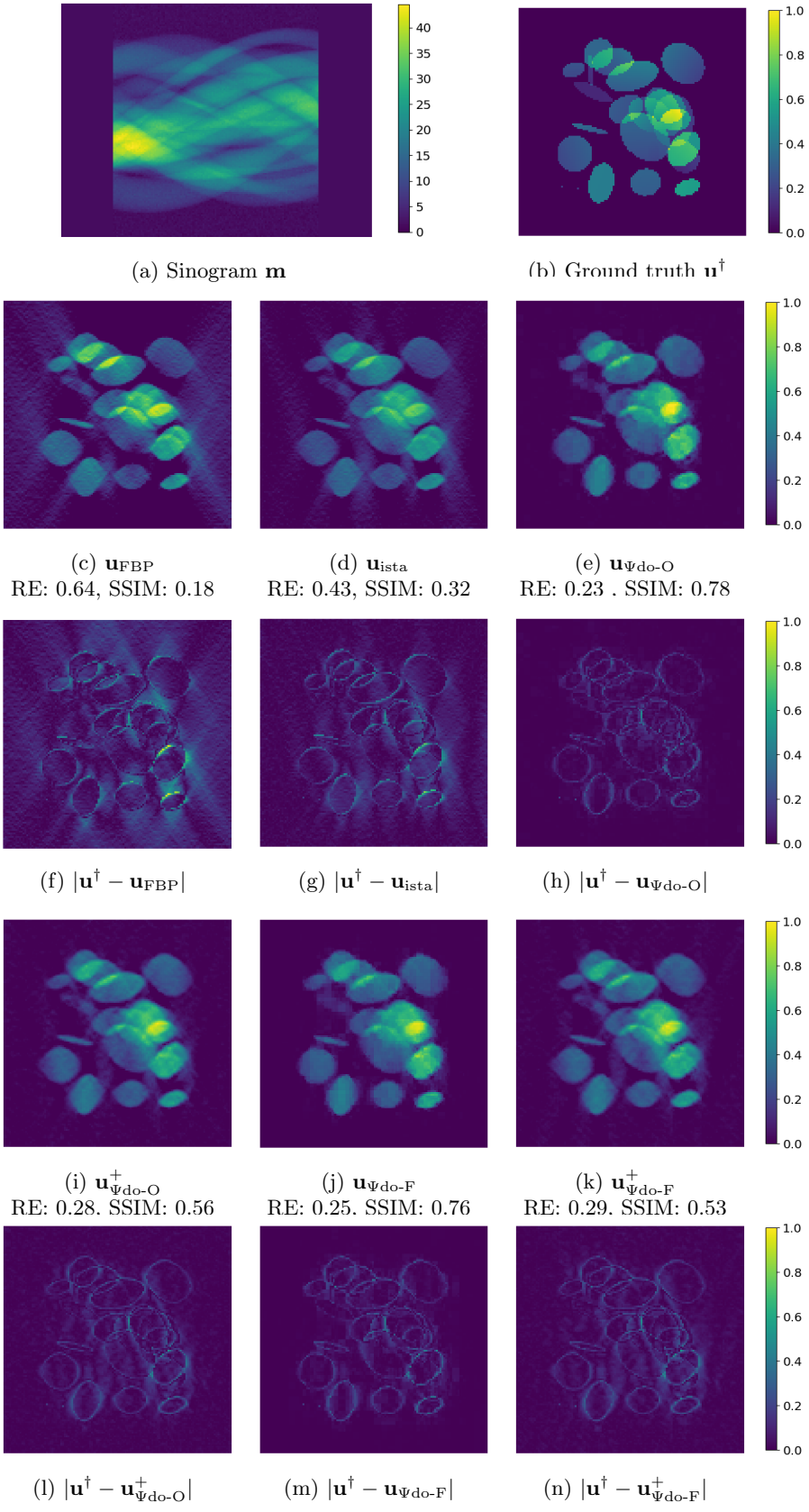
(i) $\mathbf{u}^+_{\Psi\text{do-O}}$
RE: 0.23. SSIM: 0.56

(j) $\mathbf{u}_{\Psi\text{do-F}}$
RE: 0.20. SSIM: 0.82

(k) $\mathbf{u}^+_{\Psi\text{do-F}}$
RE: 0.24. SSIM: 0.58

(l) $|\mathbf{u}^\dagger - \mathbf{u}^+_{\Psi\text{do-O}}|$ (m) $|\mathbf{u}^\dagger - \mathbf{u}_{\Psi\text{do-F}}|$ (n) $|\mathbf{u}^\dagger - \mathbf{u}^+_{\Psi\text{do-F}}|$

Fig. 10: Visualization of the sinogram (or observed image) and corresponding results for one test image.



(a) Sinogram $\mathbf{m}$ (b) Ground truth $\mathbf{u}^\dagger$

(c) $\mathbf{u}_{\text{FBP}}$
RE: 0.64, SSIM: 0.18

(d) $\mathbf{u}_{\text{ista}}$
RE: 0.43, SSIM: 0.32

(e) $\mathbf{u}_{\Psi\text{do-O}}$
RE: 0.23 . SSIM: 0.78

(f) $|\mathbf{u}^\dagger - \mathbf{u}_{\text{FBP}}|$ (g) $|\mathbf{u}^\dagger - \mathbf{u}_{\text{ista}}|$ (h) $|\mathbf{u}^\dagger - \mathbf{u}_{\Psi\text{do-O}}|$

(i) $\mathbf{u}^+_{\Psi\text{do-O}}$
RE: 0.28. SSIM: 0.56

(j) $\mathbf{u}_{\Psi\text{do-F}}$
RE: 0.25. SSIM: 0.76

(k) $\mathbf{u}^+_{\Psi\text{do-F}}$
RE: 0.29. SSIM: 0.53

(l) $|\mathbf{u}^\dagger - \mathbf{u}^+_{\Psi\text{do-O}}|$ (m) $|\mathbf{u}^\dagger - \mathbf{u}_{\Psi\text{do-F}}|$ (n) $|\mathbf{u}^\dagger - \mathbf{u}^+_{\Psi\text{do-F}}|$

Fig. 11: Visualization of the sinogram (or observed image) and corresponding results for one test image.

Appendix A. Proof of Proposition 2.2.

Proof. According to [17, Section 3], the following variational source condition is satisfied by every $w \in \ell^1(\mathbb{N})$:

$$\beta\|w - w^\dagger\|_{\ell^1} \le \|w\|_{\ell^1} - \|w^\dagger\|_{\ell^1} + C\|AW^*w - AW^*w^\dagger\|_Y. \tag{A.1}$$

We aim at applying it to $w = w_{\delta,p,q} \in W_p \subset \ell^1(\mathbb{N})$. First consider the term $\|w\|_{\ell^1} - \|w^\dagger\|_{\ell^1}$ in the right-hand side. Since $w_{\delta,p,q}$ is a solution of (2.5),

$$\begin{aligned}\lambda\|w_{\delta,p,q}\|_{\ell^1} &= \left(\|A_{p,q}W^*w_{\delta,p,q} - \mathbb{P}_q m\|_Y^2 + \lambda\|w_{\delta,p,q}\|_{\ell^1}\right) - \|A_{p,q}W^*w_{\delta,p,q} - \mathbb{P}_q m\|_Y^2 \\ &\le \|A_{p,q}W^*\mathbb{P}_p w^\dagger - \mathbb{P}_q m\|_Y^2 + \lambda\|\mathbb{P}_p w^\dagger\|_{\ell^1} - \|A_{p,q}W^*w_{\delta,p,q} - \mathbb{P}_q m\|_Y^2,\end{aligned}$$

whence

$$\|w_{\delta,p,q}\|_{\ell^1} - \|w^\dagger\|_{\ell^1} \le \frac{1}{\lambda}\|A_{p,q}W^*\mathbb{P}_p w^\dagger - \mathbb{P}_q m\|_Y^2 - \frac{1}{\lambda}\|A_{p,q}W^*w_{\delta,p,q} - \mathbb{P}_q m\|_Y^2.$$

We can easily check that $A_{p,q}W^*\mathbb{P}_p = \mathbb{P}_q AW^*\mathbb{P}_p$; then, since $\|\mathbb{P}_q\|_{Y\to Y} \le 1$, denoting by $Q = \|A_{p,q}W^*w_{\delta,p,q} - \mathbb{P}_q m\|_Y$, we have

$$\begin{aligned}\|w_{\delta,p,q}\|_{\ell^1} - \|w^\dagger\|_{\ell^1} &\le \frac{1}{\lambda}\|AW^*\mathbb{P}_p w^\dagger - m\|_Y^2 - \frac{1}{\lambda}Q^2 \\ &\le \frac{1}{\lambda}\|AW^*(\mathbb{P}_p w^\dagger - w^\dagger)\|_Y^2 + \frac{1}{\lambda}\|AW^*w^\dagger - m\|_Y^2 - \frac{1}{\lambda}Q^2.\end{aligned}$$

In conclusion,

$$\|w_{\delta,p,q}\|_{\ell^1} - \|w^\dagger\|_{\ell^1} \le \frac{1}{\lambda}\|A\|^2\|w^\dagger - \mathbb{P}_p w^\dagger\|_{\ell^2}^2 + \frac{1}{\lambda}\delta^2 - \frac{1}{\lambda}Q^2. \tag{A.2}$$

The second term in the right-hand side of (A.1), instead, can be bounded as follows:

$$\begin{aligned}\|AW^*(w_{\delta,p,q} - w^\dagger)\|_Y &= \|\mathcal{P}_q AW^*(w_{\delta,p,q} - w^\dagger)\|_Y + \|(I - \mathcal{P}_q)AW^*(w_{\delta,p,q} - w^\dagger)\|_Y \\ &\le \|A_{p,q}W^*w_{\delta,p,q} - \mathbb{P}_q m\|_Y + \delta + \|(I - \mathbb{P}_q)A\|_{X\to Y}\|w_{\delta,p,q} - w^\dagger\|_{\ell^1} + \delta \\ &\le Q + M\|(I - \mathbb{P}_q)A\|_{X\to Y} + \delta,\end{aligned} \tag{A.3}$$

where the positive constant M depends on $\|w^\dagger\|_{\ell^2}$. In order to get an estimate for $Q = \|A_{p,q}W^*w_{\delta,p,q} - \mathbb{P}_q m\|_Y$, we use (A.1): since $0 \le \beta\|w_{\delta,p,q} - w^\dagger\|_{\ell^1}$, using (A.2) and (A.3) we have

$$0 \le \frac{1}{\lambda}\|A\|\|w^\dagger - \mathbb{P}_p w^\dagger\|_{\ell^2}^2 + \frac{1}{\lambda}\delta^2 - \frac{1}{\lambda}Q^2 + Q + M\|(I - \mathbb{P}_q)A\|_{X\to Y} + \delta.$$

By solving this second-order inequality we get

$$\begin{aligned}Q &\le \frac{\lambda}{2} + \frac{\lambda}{2}\left(1 + \frac{4}{\lambda^2}\|A\|^2\|w^\dagger - \mathbb{P}_p w^\dagger\|_{\ell^2}^2\frac{4}{\lambda}\delta^2 + \frac{4M}{\lambda}\|(I - \mathbb{P}_q)A\|_{X\to Y}\frac{4}{\lambda}\delta\right)^{\frac{1}{2}} \\ &\le \lambda + \delta + \|A\|\|w^\dagger - \mathbb{P}_p w^\dagger\|_{\ell^2} + M\|(I - \mathbb{P}_q)A\|_{X\to Y}\end{aligned} \tag{A.4}$$

Combining (A.1), (A.2), (A.3), and (A.4) we easily conclude the proof. ∎

Appendix B. Proof of Proposition 2.5.

Proof. Consider the sequence $e_n = \|w_\rho^{(n+1)} - w^{(n+1)}\|_{\ell^2}$. Thanks to the nonexpansivity of the operator $S_{\lambda/L}$, it holds that

(B.1)
$$e_0 = \|\mathcal{T}_{\mathcal{Z}}(w^{(0)}) - \mathcal{T}(w^{(0)})\|_{\ell^2} \leq \frac{1}{L}\|WA_{p,q}^* A_{p,q} W^* - Z\|\|w^{(0)}\|_{\ell^2} \leq \frac{1}{L}\rho\|w^{(0)}\|_{\ell^2}.$$

Analogously, for $n \geq 1$,

(B.2)
$$\begin{aligned} e_n &\leq \|I - \frac{1}{L}Z\|\|w^{(n)} - w_\rho^{(n)}\|_{\ell^2} + \frac{1}{L}\|WA_{p,q}^* A_{p,q} W^* - Z\|\|w^{(n)}\|_{\ell^2} \\ &\leq \|I - \frac{1}{L}Z\|e_{n-1} + \frac{1}{L}\rho\|w^{(n)}\|_{\ell^2}. \end{aligned}$$

Since $L \geq \|WA_{p,q}^* A_{p,q} W^*\|$, then

$$\|I - \frac{1}{L}Z\| \leq \|I - \frac{1}{L}WA_{p,q}^* A_{p,q} W^*\| + \frac{1}{L}\|WA_{p,q}^* A_{p,q} W^* - Z\| \leq 1 + \frac{1}{L}\rho.$$

Moreover, since the sequence $\{w^{(n)}\}$ is convergent, then it is also bounded: let, *e.g.*, $\|w^{(n)}\|_{\ell^2} \leq M$. As a consequence of (B.1),(B.2),

$$e_N \leq \sum_{n=0}^{N} \left(1 + \frac{1}{L}\rho\right)^{N-n} \frac{1}{L}\rho\|w^{(n)}\|_{\ell^2} \leq M\left(\left(1 + \frac{1}{L}\rho\right)^{N+1} - 1\right)$$

Let now $N \geq N_0$ and $\rho N \leq \eta_0$: then, with a constant $c = c(N_0, \eta_0)$, it holds:

$$\|w_\rho^{(N)} - w^{(N)}\|_{\ell^2} = e_N \leq M(e^{\frac{1}{L}\rho N} - 1) \leq c\frac{M}{L}\rho N.$$

Combining this result with (2.8), we can guarantee that

$$\|w_\rho^{(N)} - w_{\delta,p,q}\|_{\ell^2} \leq \|w_\rho^{(N)} - w^{(N)}\|_{\ell^2} + \|w^{(N)} - w_{\delta,p,q}\|_{\ell^2} \leq c_3 a^N + \widetilde{c}_4 \rho N,$$

which proves (2.11). To obtain (2.12), simply substitute $N = \log_a \delta$ and $\rho = \frac{\delta}{N}$ and consider $c_4 = c_3 + \widetilde{c}_4$. ∎

REFERENCES

[1] J. Adler, H. Kohr, and O. Öktem, *Operator discretization library (ODL)*, 2017.

[2] J. Adler and O. Öktem, *Solving ill-posed inverse problems using iterative deep neural networks*, Inverse Probl., 33 (2017), p. 124007.

[3] J. Adler and O. Öktem, *Learned primal-dual reconstruction*, IEEE Trans. Med. Imaging, 37 (2018), pp. 1322–1332.

[4] S. Arridge, P. Maass, O. Öktem, and C. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numer., 28 (2019), pp. 1–174.

[5] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms i*, Commun. Pure Appl. Math., 44 (1991), pp. 141–183.

[6] J. Bolte, T. Nguyen, J. Peypouquet, and B. Suter, *From error bounds to the complexity of first-order descent methods for convex functions*, Math. Program., 165 (2017), pp. 471–507.

[7] L. Borg, J. Frikel, J. Jorgensen, and E. Quinto, *Analyzing Reconstruction Artifacts from Arbitrary Incomplete X-ray CT Data*, SIAM J. Imaging Sci., 11 (2018), pp. 2786–2814.

[8] K. Bredies and D. Lorenz, *Linear convergence of iterative soft-thresholding*, J. Fourier Anal. Appl., 14 (2008), pp. 813–837.

[9] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen, and V. Srinivasan, *Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography*, Inverse Probl., 35 (2019), p. 064002.

[10] W. Dahmen, *Wavelet and multiscale methods for operator equations*, Acta Numer., 6 (1997), pp. 55–228.
[11] W. Dahmen, S. Prössdorf, and R. Schneider, *Wavelet approximation methods for pseudo-differential equations: I stability and convergence*, Math. Z., 215 (1994), pp. 583–620.
[12] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Commun. Pure Appl. Math., 57 (2004), pp. 1413–1457.
[13] M. E. Davison, *The Ill-Conditioned Nature of the Limited Angle Tomography Problem*, SIAM J. Appl. Math., 43 (1983), pp. 428–448.
[14] M. de Hoop, M. Lassas, and C. Wong, *Deep learning architectures for nonlinear operator functions and nonlinear inverse problems*, arXiv:1912.11090, (2019).
[15] D. Fanelli and O. Öktem, *Electron tomography: a short overview with an emphasis on the absorption potential model for the forward problem*, Inverse Probl., 24 (2008), p. 013001.
[16] D. Finch, I.-R. Lan, and G. Uhlmann, *Microlocal analysis of the x-ray transform with sources on a curve*, in Inside out: inverse problems and applications, G. Uhlmann, ed., MSRI Publications, MSRI, 2003, pp. 193–218.
[17] J. Flemming and D. Gerth, *Injectivity and weak*-to-weak continuity suffice for convergence rates in ℓ_1-regularization*, J. Inverse Ill-Pose. P., 26 (2018), pp. 85–94.
[18] J. Frikel and E. Quinto, *Characterization and reduction of artifacts in limited angle tomography*, Inverse Probl., 29 (2013), p. 125007.
[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[20] A. Greenleaf and G. Uhlmann, *Estimates for singular radon transforms and pseudodifferential operators with singular symbols*, J. Funct. Anal., 89 (1990), pp. 202–232.
[21] A. Greenleaf, G. Uhlmann, et al., *Nonlocal inversion formulas for the x-ray transform*, Duke mathematical journal, 58 (1989), pp. 205–240.
[22] K. Gregor and Y. LeCun, *Learning fast approximations of sparse coding*, in Proc. 27th ICML, 2010, pp. 399–406.
[23] K. Haug and M. Lohne, *TF-Wavelets*, 2019. Link: https://github.com/UiO-CS/tf-wavelets.
[24] K. Heiskanen, H. Rhim, and P. Monteiro, *Computer simulations of limited angle tomography of reinforced concrete*, Cement Concrete Res., 21 (1991), pp. 625–634.
[25] J. R. Hershey, J. L. Roux, and F. Weninger, *Deep unfolding: Model-based inspiration of novel deep architectures*, arXiv preprint arXiv:1409.2574, (2014).
[26] L. Hörmander, *The analysis of linear partial differential operators, vols. I-IV*, Springer-Verlag, 1983-1985.
[27] K. Jin, M. McCann, E. Froustey, and M. Unser, *Deep convolutional neural network for inverse problems in imaging*, IEEE T. Image Process, 26 (2017), pp. 4509–4522.
[28] A. Katsevich, *Local tomography for the generalized radon transform*, SIAM J. Appl. Math., 57 (1997), pp. 1128–1162.
[29] A. Katsevich, *Local tomography for the limited-angle problem*, J. Math. Anal. Appl., 213 (1997), pp. 160 – 182.
[30] D. Kingma and J. Ba, *Adam: A method for stochastic optimization*, CoRR, (2015).
[31] V. Kolehmainen, S. Siltanen, S. Järvenpää, J. P. Kaipio, P. Koistinen, M. Lassas, J. Pirttilä, and E. Somersalo, *Statistical inversion for medical x-ray tomography with few radiographs: II. Application to dental radiology*, Phys. Med. Biol., 48 (2003), p. 1465.
[32] V. Krishnan and E. Quinto, *Microlocal Analysis in Tomography*, Springer, 2015, pp. 847–902.
[33] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, *PyWavelets: A Python package for wavelet analysis*, J. Open Source Softw., 4 (2019), p. 1237.
[34] S. Lunz, O. Öktem, and C.-B. Schönlieb, *Adversarial regularizers in inverse problems*, in Proc. 32nd NIPS, 2018, pp. 8507–8516.
[35] S. Mallat, *A wavelet tour of signal processing*, Elsevier, 1999.
[36] J. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*, SIAM, 2012.
[37] F. Natterer, *The mathematics of computerized tomography*, SIAM, 2001.
[38] F. Natterer and F. Wübbeling, *Mathematical methods in image reconstruction*, SIAM, 2001.
[39] L. V. Nguyen, *How strong are streak artifacts in limited angle computed tomography?*, Inverse Probl., 31 (2015), p. 055003.
[40] L. V. Nguyen, *On the strength of streak artifacts in filtered back-projection reconstructions for limited angle weighted X-ray transform*, J. Fourier Anal. Appl., 23 (2017), pp. 712–728.
[41] C. Oliver, *Synthetic-aperture radar imaging*, J. Phys. D Appl. Phys., 22 (1989), pp. 871–890.
[42] E. Quinto, *Singularities of the X-Ray transform and limited data tomography in $\mathbb{R}^2$ and $\mathbb{R}^3$*, SIAM J. Math. Anal., 24 (1993), pp. 1215–1225.
[43] E. Quinto, *An introduction to X-ray tomography and Radon transforms*, Proc. Sym-

pos. Appl. Math., 63 (2006), pp. 1–23.
[44] A. Ramm and A. Katsevich, *Inversion of incomplete radon transform*, Appl. Math. Lett., 5 (1992), pp. 41–45.
[45] A. G. Ramm and A. I. Katsevich, *The Radon transform and local tomography*, CRC press, 1996.
[46] R. Reisenhofer, S. Bosse, G. Kutyniok, and T. Wiegand, *A Haar wavelet-based perceptual similarity index for image quality assessment*, Signal Process. Image Commun., 61 (2018), pp. 33–43.
[47] D. Rumelhart, G. Hinton, and R. Williams, *Learning representations by back-propagating errors*, vol. 323, Nature, 1986.
[48] R. Tovey, M. Benning, C. Brune, M. J. Lagerwerf, S. M. Collins, R. K. Leary, P. A. Midgley, and C.-B. Schönlieb, *Directional sinogram inpainting for limited angle tomography*, Inverse Probl., 35 (2019), p. 024004.
[49] G. Uhlmann and A. Vasy, *The inverse problem for the local geodesic ray transform*, Invent. Math., 205 (2016), pp. 83–120.
[50] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. D. Beenhouwer, J. Batenburg, and J. Sijbers, *Fast and flexible X-ray tomography using the ASTRA toolbox*, Opt. Express, 24 (2016), pp. 25129–25147.
[51] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, *scikit-image: Image processing in Python*, 2014.
[52] Z. Wang, A. Bovik, H. Sheikh, and E. P. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE T. Image Process, 13 (2004), pp. 600–612.
[53] J. Zhang and B. Ghanem, *ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing*, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1828–1837.
[54] K. Zhang, L. V. Gool, and R. Timofte, *Deep unfolding network for image super-resolution*, in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3217–3226.
[55] Y. Zhang, H. P. Chan, B. Sahiner, J. Wei, M. Goodsitt, L. M. Hadjiiski, J. Ge, and C. Zhou, *A comparative study of limited–angle cone-beam reconstruction methods for breast tomosynthesis*, Med. Phys., 33 (2006), pp. 3781–3795.