

Inverse Problems with Learned Forward Operators

Simon Arridge*, Andreas Hauptmann† and Yury Korolev‡

Abstract: Solving inverse problems requires the knowledge of the forward operator, but accurate models can be computationally expensive and hence cheaper variants that do not compromise the reconstruction quality are desired. This chapter reviews reconstruction methods in inverse problems with learned forward operators that follow two different paradigms. The first one is completely agnostic to the forward operator and learns its restriction to the subspace spanned by the training data. The framework of regularisation by projection is then used to find a reconstruction. The second one uses a simplified model of the physics of the measurement process and only relies on the training data to learn a model correction. We present the theory of these two approaches and compare them numerically. A common theme emerges: both methods require, or at least benefit from, training data not only for the forward operator, but also for its adjoint.

Keywords: operator correction, operator learning, regularisation by projection, photo-acoustic tomography

MSC 2020: 65J22, 47A52, 35R30, 74J25

1 Introduction

The quality of solutions to an inverse problem depends crucially on the availability of a reliable forward model allowing one to make accurate predictions that can be compared with measured data. Such models do not always exist due to the complexity of the phenomena involved and even when accurate models exist they may be computationally too expensive for practical use in time-critical applications. Consequently, there is a need for efficient models that allow for fast computations without sacrificing reconstruction quality.

More efficient models can be obtained by introducing simplifying assumptions, such as neglecting scattering in X-ray imaging [1, 2], which can only be used in certain idealised scenarios. Another possibility for obtaining more efficient models is to consider coarser discretisations, for instance of the finite element mesh in PDE based models, but this may lead to a considerable loss of accuracy and hence a compensation is needed to retain sufficient reconstruction quality [3, 4]. Finally, in some applications the model and solutions can be constrained to a subspace allowing for a reduced order representation of the model [5–7].

In recent years the interest in data-driven methods has also sparked new interest in designing techniques that combine analytical and learned components in the forward model. We will start with a brief overview of some data-driven methods.

Data projection methods. The idea of building a low-dimensional representation of data sets and operators between them is an established technique in statistics and forms the basis of several classical machine learning methods as well as more recent deep learning based approaches. Linear methods based on principle components analysis (PCA) and robust-PCA construct spaces such that the residual error from projection onto them is small with respect to a specified tolerance in a 2-norm or 1-norm respectively. Applying these techniques to the range and domain of an operator provides a so-called reduced order model (ROM), which can as well be applied to a PDE based operator

*Department of Computer Science, University College London, 90 High Holborn, London, WC1V 6LJ, UK. Email: Simon.Arridge@cs.ucl.ac.uk

†Research Unit of Mathematical Sciences, University of Oulu, Pentti Kaiteran katu 1, Linnanmaa, Finland. Email: Andreas.Hauptmann@oulu.fi

‡Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, UK. Email: ymk30@bath.ac.uk

and its inverse (the Green’s operator) [7–9]. Kernel-PCA provides an extension by specifying a distance between data samples through a kernel function that allows for a non-linear separation criterion between components [10]. Independent component analysis (ICA) is another classical non-linear factorisation of data that is based on maximising the statistical independence of the estimated components [11]. Finally, recent developments in deep learning assume that appropriate training data lie on a manifold in an abstract latent space that is obtained by learning simultaneously an encoder to, and a decoder from, the manifold such that the composite operator (an “autoencoder”) minimises an appropriate loss function [12].

Let us also mention that projections are the basis of some classical regularisation methods [13–18].

Operator learning. There exists a growing body of literature on learning mappings between infinite-dimensional spaces, referred to as “operator learning”. One prominent example is Neural Operators [19] that have a multi-layer structure similar to a conventional neural network but whose layers are infinite-dimensional operators. Examples include Fourier Neural Operators [20] and Deep Operator Networks [21]. Although neural operators are infinite-dimensional objects, they need to be discretised in practice, resulting in a conventional neural network. However, in order to ensure consistency of this discretisation with the infinite-dimensional limit, down- and upsampling operators need to be included in the architecture [22].

While training a neural operator, just like training a finite-dimensional neural network, can be computationally expensive, random feature models (also known as kernel methods) combine optimisation with randomisation and result in a significantly simpler convex problem. Random feature models also make sense in infinite dimensions and admit efficient (Monte-Carlo) convergence rates in terms of the number of features [23, 24]. In some sense, random feature models turn the problem of learning a nonlinear operator over the input space into that of learning a linear operator over the parameter space (this is sometimes referred to as the “kernel trick”). Learning a linear operator as an inverse problem was considered in [25, 26] where convergence rates have also been obtained. Approximation rates can also be obtained for infinite-dimensional holomorphic operators, which has been done in [27] based on earlier work [28]. Barron operators are another class of infinite-dimensional operators for which efficient approximation rates have been obtained [29].

In the context of inverse problems, [30] proposes to learn the forward operator of an inverse problem (or its regularised inversion) based on invertible residual networks. Learned PDE operators have also been used in parameter estimation problems [31].

For more details on operator learning we refer the reader to the recent review [32].

Our contribution. In this chapter we will review two fundamentally different paradigms for solving inverse problems with the aid of training data based either on learning a data-driven representation of the forward model following the paper [33] or learning a correction operator to a given cheaper approximation of the forward model [34].

The first approach relies on the important observation that if the forward operator is linear then its restriction to the span of the training data can be computed *without any access to the forward operator*. The method proposed in [33] relies on orthogonalising the training set using a Gram-Schmidt process (see also [35] for generalisations). While this is costly, it has to be done only once and “offline”, i.e. before solving the actual inverse problem, and can be reused for problems with the same operator but different measurements. In some sense, this is similar to using a neural network, where training is costly but applying a trained network is cheap.

The second approach follows the classical model correction paradigm and assumes that a computationally inexpensive simplified model is given, such as a coarser discretisation [3] or an analytic approximation [36], which in itself is not sufficiently accurate to produce good reconstructions when used in a variational reconstruction framework. Recent work in [34, 36] considers learning a data-driven correction given by a neural network trained on suitable training data. We will discuss two approaches to learning such a correction either as part of a learned reconstruction operator, such as an unrolled iterative scheme [36], or as a separate explicit correction network for the forward operator [34].

We start with a discussion of data-driven *regularisation by projection* in Section 2. Most of

the material is taken from [33], but we also add a new discussion of the connections to iterative reconstruction methods in Section 2.3.2. Then we move on, in Section 3, to the case where an approximate model is used alongside with a learned correction. We discuss implicit and explicit corrections and draw connections to the previous section on regularisation by projection. Then we present numerical experiments with these approaches in Section 4 for the problem of limited-view photoacoustic tomography. We briefly present ongoing work on optimisation on learned manifolds in Section 4.3.1 and discuss possible directions for future research in Section 5.

We also identify a common theme: it turns out that in order to obtain good reconstructions, both methods require (or at least benefit from) training data not only for the forward operator but also for its adjoint.

1.1 Mathematical setting

Within this chapter we will consider a linear inverse problem

$$Ax = y, \quad (1.1)$$

where $A: \mathcal{X} \rightarrow \mathcal{Y}$ is a linear bounded operator acting between separable Hilbert spaces \mathcal{X} and \mathcal{Y} and $A^*: \mathcal{Y} \rightarrow \mathcal{X}$ is its adjoint. Often, the exact right-hand side in (1.1) is not available and we only have access to an approximation y^δ such that $\|y - y^\delta\| \leq \delta$ for some $\delta > 0$. We will describe methods for solving (1.1) that do not require access to the exact operator A during the solution phase, but rely on training pairs/triples

$$x^i \in \mathcal{X}, \quad y^i = Ax^i \in \mathcal{Y}, \quad \text{and} \quad z^i = A^*Ax^i \in \mathcal{X}, \quad i = 1, \dots, n, \quad (1.2)$$

together with (in some cases) a simplified approximate model and its adjoint $\tilde{A}: \mathcal{X} \rightarrow \mathcal{Y}$, $\tilde{A}^*: \mathcal{Y} \rightarrow \mathcal{X}$. Borrowing terminology from tomography, we will call x^i 's images, y^i 's measurements and z^i 's backprojections. The collection $\{x^i, y^i, z^i\}_{i=1, \dots, n}$ from (1.2) will be referred to as *training data*.

2 Data-driven regularisation by projection

2.1 Setting and main assumptions

We start with a simple but important observation that was made in [33]: the training data (1.2) completely describe the forward and the normal operators on the span of the training images $\text{span}\{x^i\}_{i=1, \dots, n}$. The restriction of A and A^*A to this subspace can be computed using Gram-Schmidt orthogonalisation, which needs to be done only once and can be done offline, prior to solving the inverse problem. In this section we will show how such learned operators can be used for regularised inversion of (1.1).

Let us first fix some notation and state our main assumptions. The exact solution of (1.1) (with exact forward model A and noise-free measurement) will be denoted by x_{true} .

The spans of the training images, measurements and backprojections will be denoted by

$$\mathcal{X}_n := \text{span}\{x^i\}_{i=1, \dots, n}, \quad \mathcal{Y}_n := \text{span}\{y^i\}_{i=1, \dots, n}, \quad \mathcal{Z}_n := \text{span}\{z^i\}_{i=1, \dots, n}.$$

Orthogonal projection operators onto \mathcal{X}_n , \mathcal{Y}_n and \mathcal{Z}_n are denoted by $P_{\mathcal{X}_n}$, $P_{\mathcal{Y}_n}$ and $P_{\mathcal{Z}_n}$, respectively.

Assumption 1 (Independence, uniform boundedness, sequentiality, density).

Linear independence: For every $n \in \mathbb{N}$ the images $\{x^i\}_{i=1, \dots, n}$ are linearly independent.

Uniform boundedness: There exist constants $c_u, C_u > 0$ such that $c_u \leq \|x^i\| \leq C_u$ for all $i \in \mathbb{N}$. Hence with no loss of generality we will assume that $\|x^i\| = 1$ for all $i \in \mathbb{N}$.

Sequentiality: The families of training triples are nested, i.e. for every $n \in \mathbb{N}$

$$\{x^i, y^i, z^i\}_{i=1, \dots, n+1} = \{x^i, y^i, z^i\}_{i=1, \dots, n} \cup \{x^{n+1}, y^{n+1}, z^{n+1}\}.$$

Consequently, the subspaces \mathcal{X}_n , \mathcal{Y}_n and \mathcal{Z}_n are nested, that is

$$\mathcal{X}_n \subset \mathcal{X}_{n+1}, \quad \mathcal{Y}_n \subset \mathcal{Y}_{n+1}, \quad \mathcal{Z}_n \subset \mathcal{Z}_{n+1} \quad \text{for all } n.$$

Density: the subspaces spanned by the images $\{x^i\}_{i \in \mathbb{N}}$ are dense in \mathcal{X} , that is

$$\overline{\bigcup_{n \in \mathbb{N}} \mathcal{X}_n} = \mathcal{X}.$$

Consequently, the subspaces spanned by the training measurements $\{y^i\}_{i \in \mathbb{N}}$ and backprojections $\{z^i\}_{i \in \mathbb{N}}$ are dense in the closures of the ranges $\overline{\mathcal{R}(A)}$ and $\overline{\mathcal{R}(A^*)}$, respectively. (The last statement follows from the fact that $\overline{\mathcal{R}(A^*A)} = \overline{\mathcal{R}(A^*)}$, which is easily checked.)

2.2 The injective case: regularisation by projection

Let $y \in \mathcal{R}(A)$ be the exact, noise-free right-hand side in (1.1) and consider the following projected problem

$$AP_{\mathcal{X}_n}x = y. \quad (2.1)$$

Its minimum-norm solution is given by

$$x_n^{\mathcal{X}} = (AP_{\mathcal{X}_n})^\dagger y, \quad (2.2)$$

where $(AP_{\mathcal{X}_n})^\dagger$ denotes the Moore-Penrose inverse of $AP_{\mathcal{X}_n}$. The superscript \mathcal{X} in $x_n^{\mathcal{X}}$ reflects the fact that the projection in (2.1) takes place in \mathcal{X} .

The following result shows that in the injective case, a simple formula for $(AP_{\mathcal{X}_n})^\dagger$ exists.

Theorem 1 ([33, Thm. 4]). *Let A be injective. Then the Moore-Penrose inverse of $AP_{\mathcal{X}_n}$ is given by*

$$(AP_{\mathcal{X}_n})^\dagger = A^{-1}P_{\mathcal{Y}_n}.$$

Combining this with (2.2), we get a simple reconstruction formula

$$x_n^{\mathcal{X}} = A^{-1}P_{\mathcal{Y}_n}y. \quad (2.3)$$

Since A is injective, the restriction of its inverse to \mathcal{Y}_n can be computed using only the training data (1.2). Gram-Schmidt orthogonalisation can be used for this purpose; we refer to [33, Sect. 3.1] for details.

The approach (2.1) is known as *regularisation by projection* [37]. In the model-based setting, projections are taken onto subspaces spanned by a certain number of basis functions, for example, finite elements. If the basis of singular vectors of the forward operator A is used, the method reduces to the truncated singular value decomposition [37]. In our case, projections are taken onto subspaces given by training data.

2.2.1 Convergence analysis

Examples of non-convergence exist [38] that show that minimum-norm solutions (2.2) can diverge as $n \rightarrow \infty$ even for a noiseless measurement y . Therefore, without additional assumptions on the subspaces \mathcal{X}_n , we cannot expect convergence of the reconstructions (2.3). In [33] sufficient conditions have been obtained that rely on the interplay between the training images $\{x^i\}_{i \in \mathbb{N}}$, the exact solution x_{true} and the forward operator A .

To state these conditions, let us apply Gram-Schmidt orthogonalisation to the sequence $\{x^i\}_{i \in \mathbb{N}}$, obtaining an orthonormal basis $\{\bar{x}^i\}_{i \in \mathbb{N}}$ of \mathcal{X} . Transforming accordingly the training measurements $\{y^i\}_{i \in \mathbb{N}}$, we obtain a corresponding sequence $\{\bar{y}^i\}_{i \in \mathbb{N}}$ such that $A\bar{x}^i = \bar{y}^i$. Expanding the exact solution in the basis $\{\bar{x}^i\}_{i \in \mathbb{N}}$, we get

$$x_{\text{true}} = \sum_{i=1}^{\infty} \langle x_{\text{true}}, \bar{x}^i \rangle \bar{x}^i. \quad (2.4)$$

We are now ready to state our assumptions.

Assumption 2. *Coefficients of the expansion (2.4) are in ℓ^1 , i.e.*

$$\sum_{i=1}^{\infty} |\langle x_{\text{true}}, \bar{x}^i \rangle| < \infty.$$

Summability conditions such as Assumption 2 are common in machine learning and are used to define so-called *variation norm spaces* [39, 40].

Remark 1. *In [30], the authors introduce the so-called local approximation property (Theorem 3.1) which requires that the neural network achieves certain approximation rates in the vicinity of the ground truth, but not globally. This turns out to be sufficient for the convergence of regularised solutions. On the conceptual level, this assumption is similar to our Assumption 2 which only requires a certain approximation rate by the subspaces spanned by the training images for the ground truth, but not globally on the whole space.*

Assumption 3. *For every $n \in \mathbb{N}$ and any $i \geq n+1$ consider the following expansion of $P_{\mathcal{Y}_n} \bar{y}^i \in \mathcal{Y}_n$*

$$P_{\mathcal{Y}_n} \bar{y}^i = \sum_{j=1}^n \beta_j^{i,n} \bar{y}^j. \quad (2.5)$$

We assume that for every $n \in \mathbb{N}$

$$\sum_{j=1}^n (\beta_j^{i,n})^2 \leq C, \quad \text{for every } i \geq n+1, \quad (2.6)$$

where $C > 0$ is a constant independent of i and n .

We emphasise that the expansion coefficients $\beta_j^{i,n}$ change with n because $\{\bar{y}^i\}_{i=1,\dots,n}$ is not an orthogonal basis.

Assumption 3 is far less interpretable than Assumption 2. It depends on the interplay between the training images $\{x^i\}_{i \in \mathbb{N}}$ and the forward operator A . As discussed in [33, Sect. 6.1.3], checking this assumption numerically is also problematic because computing the coefficients $\beta_j^{i,n}$ would involve inverting an ill-conditioned matrix. (Note, however, that this inversion is not required for finding the solution (2.3).) In the non-convergence example from [38], Assumption 3 can be checked analytically (and is valid).

The above two assumptions allow us to prove uniform boundedness of minimum-norm solutions (2.3).

Theorem 2 ([33, Thm. 11]). *Let Assumptions 2 and 3 be satisfied. Then $x_n^{\mathcal{X}}$ as defined in (2.3) is uniformly bounded with respect to n . Consequently, we have that $x_n^{\mathcal{X}} \rightharpoonup x_{\text{true}}$ weakly along a subsequence.*

Under additional assumptions it is possible to prove strong convergence [33, Thm. 15].

So far, we have considered the case of a noise-free measurement y in (1.1). If the measurement is noisy (y_δ such that $\|y - y_\delta\| \leq \delta$ for some known noise level $\delta > 0$) then the parameter n in the projected equation (2.1) becomes a regularisation parameter [37] that needs to be chosen as a function of the measurement noise, the larger the noise level δ the smaller n . Details about our specific setting can be found in [33, Thm. 17].

It may seem counter-intuitive that increasing the size of the training set should lead to instabilities, but in our case the parameter n also controls model complexity, i.e. the number of components in the solution. By the nature of the reconstruction formula (2.3), we are in the regime where the number of parameters matches the number of data (i.e., we are not in the overparametrised regime) and hence the complexity of a model has to be controlled by the noise in the data. This is in line with classical results on training neural networks from noisy data [41].

2.2.2 Dual least squares

Although projecting the equation (1.1) in the space \mathcal{X} as in (2.1) does not yield a convergent solution in general, it is known that projecting (1.1) in the space \mathcal{Y} yields convergent solutions. This method is also referred to as *dual least squares* [37].

The dual least squares method consists in finding the minimum norm solution of the following problem

$$P_{\mathcal{Y}_n} Ax = P_{\mathcal{Y}_n} y, \quad (2.7)$$

We denote the minimum norm solution of (2.7) by $x_n^{\mathcal{Y}}$, where the superscript \mathcal{Y} emphasises the fact that the projection in (2.7) takes place in \mathcal{Y} .

The following classical result shows that $x_n^{\mathcal{Y}}$ converges strongly to the exact solution x_{true} as $n \rightarrow \infty$.

Theorem 3 ([37, Thm. 3.24]). *Let y be the exact data in (1.1). Then the minimum norm solution of (2.7) is given by*

$$x_n^{\mathcal{Y}} = P_{A^* \mathcal{Y}_n} x_{\text{true}}, \quad (2.8)$$

where $P_{A^* \mathcal{Y}_n}$ is the orthogonal projector onto the subspace $A^* \mathcal{Y}_n$. Consequently,

$$x_n^{\mathcal{Y}} \rightarrow x_{\text{true}} \quad \text{as } n \rightarrow \infty.$$

The following result gives a simple characterisation of the Moore-Penrose inverse of $P_{\mathcal{Y}_n} A$, similarly to Theorem 1.

Theorem 4 ([33, Thm. 19]). *Let A have a dense range. Then the Moore-Penrose inverse of $P_{\mathcal{Y}_n} A$ is given by*

$$(P_{\mathcal{Y}_n} A)^\dagger = P_{A^* \mathcal{Y}_n} A^{-1}.$$

Hence, the minimum norm solution $x_n^{\mathcal{Y}}$ of (2.7) is given by

$$x_n^{\mathcal{Y}} = P_{A^* \mathcal{Y}_n} A^{-1} P_{\mathcal{Y}_n} y = P_{A^* \mathcal{Y}_n} x_n^{\mathcal{X}}, \quad (2.9)$$

where $x_n^{\mathcal{X}}$ is the minimum norm solution of (2.1) as defined in (2.3).

The space $A^* \mathcal{Y}_n$ is, in fact, nothing but the span of the training backprojections $\{z^i\}_{i=1,\dots,n}$

$$A^* \mathcal{Y}_n = \mathcal{Z}_n.$$

Therefore, in order to compute the stable reconstruction (2.9), having training data for the forward operator A is not sufficient, one also needs training data for the adjoint A^* . The need for training data for the adjoint is a topic that we will also encounter later in Section 3 when we will discuss data-driven model corrections.

As in Section 2.2.1, if the measurement in (1.1) is noisy, the model complexity (that is, the dimension of the space n) has to be controlled by the amount of noise in the measurement. Convergence analysis of the dual least squares method can be found in [37, Thm. 3.26].

For numerical experiments with data-driven regularisation by projection (2.1) and dual least squares (2.7) we refer to [33, Sec. 6].

2.3 The non-injective case: variational regularisation

If the forward operator A is not injective, the results of Section 2.2 do not apply because (2.3) requires us to be able to apply the inverse A^{-1} to elements in the span of the training measurements $\mathcal{Y}_n = \text{span}\{y^i\}_{i=1,\dots,n}$. However, the training data (1.2) still allow us to evaluate the forward operator on the span of the training images $\mathcal{X}_n = \text{span}\{x^i\}_{i=1,\dots,n}$. Indeed, using the orthonormalised system $\{\bar{x}^i\}_{i=1,\dots,n}$ and the corresponding transformed measurements $\{\bar{y}^i\}_{i=1,\dots,n}$, we get that for any $x \in \mathcal{X}_n$

$$x = \sum_{i=1}^n \langle x, \bar{x}^i \rangle \bar{x}^i \quad \text{and} \quad Ax = \sum_{i=1}^n \langle x, \bar{x}^i \rangle A\bar{x}^i = \sum_{i=1}^n \langle x, \bar{x}^i \rangle \bar{y}^i, \quad x \in \mathcal{X}_n.$$

For an arbitrary $x \in \mathcal{X}$, therefore, we can evaluate the restriction $AP_{\mathcal{X}_n}$ without having numerical access to A :

$$AP_{\mathcal{X}_n}x = \sum_{i=1}^n \langle x, \bar{x}^i \rangle \bar{y}^i, \quad x \in \mathcal{X}. \quad (2.10)$$

The operators $AP_{\mathcal{X}_n}$ approximate A pointwise as $n \rightarrow \infty$; if A is compact then approximation also holds in the operator norm [42]. Hence, we are in the framework of inverse problems with operator errors (e.g., [13, 16, 43]).

In this section we will study the following variational regularisation problem

$$\min_{x \in \mathcal{X}} \frac{1}{2} \|AP_{\mathcal{X}_n}x - y^\delta\|^2 + \alpha \mathcal{J}(x), \quad (2.11)$$

where $\mathcal{J}: \mathcal{X} \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ is a regulariser and $\alpha > 0$ a regularisation parameter. Before we proceed with the analysis, let us say how our setting differs from the literature.

Firstly, although the forward operator in (2.11) is evaluated on a finite-dimensional subspace, the solution will not be finite-dimensional, in general. This is different from the setting of discretised variational regularisation [13, 16], where the solution is constrained to lie in an a priori prescribed finite-dimensional space. This is also in contrast with Section 2.2, where the reconstructions (2.3) and (2.9) are linear combinations of a finite number of training points.

Secondly, classical theory of regularisation under operator errors deals with bounds in the operator norm h_n such that $\|A - AP_{\mathcal{X}_n}\| \leq h_n$. This is a global estimate that depends on how well the subspaces \mathcal{X}_n agree with the operator A (the ideal choice would be, obviously, the eigenspaces of A corresponding to n largest eigenvalues). From the data-driven point of view, we would like to work with a local error estimate such as $\|(A - AP_{\mathcal{X}_n})x_J^\dagger\|$ or $\|(I - P_{\mathcal{X}_n})x_J^\dagger\|$, where x_J^\dagger is the \mathcal{J} -minimising solution of (1.1). Even if the global approximation error $\|A - AP_{\mathcal{X}_n}\|$ is large, convergence can still be fast if the training data (1.2) are chosen well for a particular solution x_J^\dagger . This will be formalised in Theorem 6 below. Such local approximation conditions appear in other contexts as well, such as regularisation by invertible residual networks [30], as discussed earlier.

2.3.1 Convergence analysis

We will make the following standard assumptions.

Assumption 4. *The regularisation functional $\mathcal{J}: \mathcal{X} \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ is proper, convex, lower-semicontinuous, and absolutely p -homogeneous ($p \geq 1$).*

Denote by $\mathcal{N}(\mathcal{J})$ the kernel (zero-level set) of the regulariser \mathcal{J} , which is a linear subspace because \mathcal{J} is convex and absolutely p -homogeneous.

Assumption 5. *The kernel $\mathcal{N}(\mathcal{J})$ satisfies $\dim(\mathcal{N}(\mathcal{J})) < +\infty$ and \mathcal{J} is coercive on the quotient space $\mathcal{X}/\mathcal{N}(\mathcal{J})$. Furthermore, for all $n \in \mathbb{N}$ we have*

$$\mathcal{N}(AP_{\mathcal{X}_n}) \cap \mathcal{N}(\mathcal{J}) = \{0\}.$$

If \mathcal{J} is the Total Variation regulariser [44], Assumptions 4 and 5 are satisfied if $AP_{\mathcal{X}_n}: L^2 \rightarrow L^2$ does not annihilate constant functions.

Existence of minimisers in (2.11) follows from standard arguments. Convergence as $\delta \rightarrow 0$ can be ensured under the usual parameter choice rule $\alpha = \alpha(\delta, n)$.

Theorem 5 ([33, Thm. 23], slightly modified). *Suppose that Assumptions 4 and 5 are satisfied and the regularisation parameter $\alpha = \alpha(\delta, n)$ is chosen such that*

$$\alpha \rightarrow 0 \quad \text{and} \quad \frac{(\delta + \|A(I - P_{\mathcal{X}_n})x_J^\dagger\|)^2}{\alpha} \rightarrow 0 \quad \text{as } \delta \rightarrow 0 \text{ and } n \rightarrow \infty.$$

Then every sequence of minimisers $x_J^{n,\delta}$ of (2.11) has a weakly convergent subsequence

$$x_J^{n,\delta} \rightharpoonup x_J^\dagger.$$

Convergence rates can also be obtained under additional assumptions. Such rates are usually stated in terms of a (generalised) Bregman distance induced by the regularisation functional [45]. We recall the definition for readers' convenience.

Definition 1 (generalised Bregman distance). *For a proper convex functional \mathcal{J} the generalised Bregman distance between $x', x \in \mathcal{X}$ corresponding to the subgradient $q \in \partial\mathcal{J}(x)$ is defined as follows*

$$D_{\mathcal{J}}^q(x', x) := \mathcal{J}(x') - \mathcal{J}(x) - \langle q, x' - x \rangle.$$

Here $\partial\mathcal{J}(x)$ denotes the subdifferential of \mathcal{J} at $x \in \mathcal{X}$.

The additional assumption required for obtaining a convergence rate is the *source condition*.

Theorem 6 ([33, Thm. 23], slightly modified). *Suppose that Assumptions 4, 5 are satisfied and that $x_{\mathcal{J}}^\dagger$ satisfies a source condition, i.e. there exists an element $q^\dagger \in \mathcal{Y}$ such that*

$$A^*q^\dagger \in \partial\mathcal{J}(x_{\mathcal{J}}^\dagger).$$

Then the following estimate holds for the Bregman distance between $x_{\mathcal{J}}^{n,\delta}$ and $x_{\mathcal{J}}^\dagger$

$$\begin{aligned} D_{\mathcal{J}}^{A^*q^\dagger}(x_{\mathcal{J}}^{n,\delta}, x_{\mathcal{J}}^\dagger) &\leq \frac{1}{2\alpha} \left(\delta + \|A(I - P_{\mathcal{X}_n})x_{\mathcal{J}}^\dagger\|^2 \right)^2 \\ &+ \frac{\alpha}{2} \|q^\dagger\|^2 + \left(\delta \|q^\dagger\| + C \|(I - P_{\mathcal{X}_n})A^*q^\dagger\| \right) \end{aligned}$$

for some constant $C > 0$.

If the regularisation parameter $\alpha = \alpha(\delta, n)$ is chosen as in Theorem 5 then

$$D_{\mathcal{J}}^{A^*q^\dagger}(x_{\mathcal{J}}^{n,\delta}, x_{\mathcal{J}}^\dagger) \rightarrow 0 \quad \text{as } \delta \rightarrow 0 \text{ and } n \rightarrow \infty.$$

For the particular choice

$$\alpha \sim \left(\delta + \max \left\{ \|A(I - P_{\mathcal{X}_n})x_{\mathcal{J}}^\dagger\|, \|(I - P_{\mathcal{X}_n})A^*q^\dagger\| \right\} \right) \quad (2.12)$$

we obtain the following estimate

$$D_{\mathcal{J}}^{A^*q^\dagger}(x_{\mathcal{J}}^{n,\delta}, x_{\mathcal{J}}^\dagger) \sim \alpha.$$

The proofs of both theorems are similar to [13, 16], although in those papers minimisers of the functional $x \in \mathcal{X} \rightarrow \|Ax - y^\delta\|^2 + \alpha\mathcal{J}(x)$ are approximated by minimisers of the same functional over \mathcal{X}_n , while we solve the problem on the whole infinite-dimensional space.

We note that the convergence rate in Theorem 6 depends not only on how well the training images $\{x^i\}_{i=1,\dots,n}$ approximate the \mathcal{J} -minimising solution $x_{\mathcal{J}}^\dagger$, which is not surprising, but also on how well they approximate the subgradient A^*q^\dagger from the source condition. This is another instance where training data for the adjoint operator A^* may be advantageous.

We would also like to emphasise the different roles that the amount of training data n plays in regularisation by projection (Section 2.2) and variational regularisation. In regularisation by projection the solution is a linear combination of n elements of the training set and therefore the size of this set n controls the model complexity. The number of parameters in this case is the same as the number of training points. Furthermore, this number has to be controlled by the level of noise in the measurement y^δ . In variational regularisation the solution is infinite-dimensional. Therefore, in some sense, we are in an overparametrised regime where the number of parameters (degrees of freedom in the solution) is infinite while the number of training points is finite. The parameter n controls the approximation quality of the forward operator and can be chosen independently of the amount of noise in y^δ .

2.3.2 Iterative reconstruction methods and the role of the adjoint

The method (2.11) has demonstrated good numerical performance learning and inverting the Radon transform [33, Sec. 6.4]. These experiments used conic solvers provided by the CVX package [46]. Due to memory requirements, such solvers are not suited for large-scale applications, and iterative solvers are used instead. In this section we briefly discuss the application of such methods.

Perhaps the simplest iterative method, gradient descent, consists in taking the following updates for solving (2.11)

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \tau_k((AP_{\mathcal{X}_n})^*(AP_{\mathcal{X}_n})x^{(k)} - (AP_{\mathcal{X}_n})^*y^\delta + \alpha q_k) \\ &= x^{(k)} - \tau_k(P_{\mathcal{X}_n}(A^*AP_{\mathcal{X}_n}x^{(k)} - A^*y^\delta) + \alpha q_k), \quad q_k \in \partial \mathcal{J}(x^{(k)}), \end{aligned} \quad (2.13)$$

where $\partial \mathcal{J}(x^{(k)})$ is the subdifferential of \mathcal{J} at the iterate $x^{(k)}$ and $\tau_k > 0$ is the step size. We see that the iteration requires computing the operator $(AP_{\mathcal{X}_n})^* = P_{\mathcal{X}_n}A^*$. It is an easy calculation to show that it can be evaluated without numerical access to A^* :

$$P_{\mathcal{X}_n}A^*y = (AP_{\mathcal{X}_n})^*y = \sum_{i=1}^n \langle y, \bar{y}^i \rangle \bar{x}^i, \quad y \in \mathcal{Y}. \quad (2.14)$$

Compare (2.13) to the gradient descent step for the corresponding problem with the exact operator A ,

$$x^{(k+1)} = x^{(k)} - \tau_k(A^*Ax^{(k)} - A^*y^\delta + \alpha q_k). \quad (2.15)$$

In (2.13) a projection $P_{\mathcal{X}_n}$ is applied after the action of the restriction of the normal operator $A^*AP_{\mathcal{X}_n}$. Depending on the problem, this may be a curse or a blessing. The range of the operator $A^*AP_{\mathcal{X}_n}$ is the span of the training backprojections $\{z^i\}_{i=1,\dots,n}$,

$$\mathcal{R}(A^*AP_{\mathcal{X}_n}) = \mathcal{Z}_n = \text{span}\{z^i\}_{i=1,\dots,n},$$

while the projection $P_{\mathcal{X}_n}$ will force the updates into the span of the training images $\{x^i\}_{i=1,\dots,n}$,

$$\mathcal{R}(P_{\mathcal{X}_n}A^*AP_{\mathcal{X}_n}) \subseteq \mathcal{X}_n = \text{span}\{x^i\}_{i=1,\dots,n}.$$

Depending on which subspaces, \mathcal{X}_n or \mathcal{Z}_n , can better approximate the \mathcal{J} -minimising solution $x_{\mathcal{J}}^\dagger$, the projection $P_{\mathcal{X}_n}$ in (2.13) may or may not be beneficial. We also note that if the forward operator is smoothing, then elements of \mathcal{Z}_n will be smoother than those of \mathcal{X}_n .

The (outer) projection $P_{\mathcal{X}_n}$ in (2.13) can be avoided if we have access to training data for the normal operator $\{z^i\}_{i=1,\dots,n}$, see (1.2). In this case we can directly approximate the normal operator in (2.15) with its restriction to \mathcal{X}_n and obtain

$$x^{(k+1)} = x^{(k)} - \tau_k(A^*AP_{\mathcal{X}_n}x^{(k)} - A^*P_{\mathcal{Y}_n}y^\delta + \alpha q_k). \quad (2.16)$$

The operator $A^*AP_{\mathcal{X}_n}$ can be evaluated without numerical access to either A or A^* using training data (1.2). Indeed, applying A^* to both sides of (2.10), we get

$$A^*AP_{\mathcal{X}_n}x = \sum_{i=1}^n \langle x, \bar{x}^i \rangle \bar{z}^i,$$

where $\{\bar{x}^i\}_{i=1,\dots,n}$ are the orthonormalised training images and $\{\bar{z}^i\}_{i=1,\dots,n}$ the corresponding transformed backprojections. We need to apply a projection $P_{\mathcal{Y}_n}$ to the measurement y^δ before applying A^* to match the range of the learned normal operator, $\mathcal{R}(A^*AP_{\mathcal{X}_n}) = \mathcal{Z}_n = \mathcal{R}(A^*P_{\mathcal{Y}_n})$. The restricted operator $A^*P_{\mathcal{Y}_n}$ can also be evaluated without numerical access to A^* using training data (1.2).

We will present numerical experiments with these approaches in Section 4.2.

3 Data-driven model correction

In this section we will consider the case when an expensive forward model can be approximated by a computationally more efficient one. For instance, in applications where the forward model is given by the solution of a partial differential equation, model reduction techniques are often used to reduce computational cost, e.g., by reduced order models or coarser discretisations. When the accurate model is replaced by a reduced one, this will lead to approximation errors, which may corrupt the reconstructed image depending on the severity of the approximation error. In the following, we will discuss how such model errors can be corrected with data-driven methods and used to solve the inverse problem in a variational setting.

We recall that we consider linear inverse problems where we denote by $x \in \mathcal{X}$ the unknown quantity of interest that we aim to reconstruct from measurements $y \in \mathcal{Y}$ and x and y fulfil the relation

$$Ax = y, \quad (3.1)$$

given bounded linear $A : \mathcal{X} \rightarrow \mathcal{Y}$ (the accurate forward operator) acting between separable Hilbert spaces \mathcal{X} and \mathcal{Y} .

We assume that the evaluation of the accurate operator A is computationally expensive and we rather want to use a cheaper approximate model $\tilde{A} : \mathcal{X} \rightarrow \mathcal{Y}$ with

$$\tilde{A}x = \tilde{y}, \quad (3.2)$$

leading to a systematic model error $\varepsilon = y - \tilde{y}$. In the following we will discuss different approaches to taking this systematic approximation error into account. First, we discuss a classical statistical correction, introduced as the approximation error method in [3, 47] and used in a specific application for linear corrections to nonlinear models in the variational setting [48]. We will then discuss two approaches, implicit and explicit corrections, in the framework of learned image reconstruction and the possibility to establish convergence guarantees for the explicit case [34].

3.1 The Approximation Error Method (AEM)

The well-established Bayesian Approximation Error Method [3, 47] is an early data-driven approach to estimating a statistical model error. Recall that in Bayesian inversion we want to determine the posterior distribution of the unknown x given y and using Bayes' formula we obtain

$$p(x|y) = p(y|x) \frac{p(x)}{p(y)}. \quad (3.3)$$

Thus, the posterior distribution is characterised by the likelihood $p(y|x)$ and the chosen prior $p(x)$ on the unknown. Typically, the likelihood $p(y|x)$ is modelled using accurate knowledge of the forward operator $A : \mathcal{X} \rightarrow \mathcal{Y}$ as well as the noise model on the data y . The underlying idea of the approximation error method is to adjust the likelihood by examining the difference between the (accurate) forward operator A and its approximation \tilde{A} (3.1)–(3.2) as

$$\varepsilon = Ax - \tilde{A}x. \quad (3.4)$$

Including an additive model for the measurement noise e , this leads to the modified observation model

$$y = \tilde{A}x + \varepsilon + e. \quad (3.5)$$

Here, both errors are (usually) assumed to be Gaussian. That is, first we model the measurement noise e independent of x as $e \sim \mathcal{N}(\eta_e, \Gamma_e)$, where η_e and Γ_e are the mean and covariance. Further, the model error ε is approximated as Gaussian $\varepsilon \sim \mathcal{N}(\eta_\varepsilon, \Gamma_\varepsilon)$ and is assumed independent of the noise e and the unknown x leading to a Gaussian distributed total error $\xi = \varepsilon + e$, $\xi \sim \mathcal{N}(\eta_\xi, \Gamma_\xi)$, where η_ε and η_ξ are means and Γ_ε and Γ_ξ are the covariance matrices of model error and total errors, respectively. We note here that the assumption of independence is a simplification and in practice

one often observes dependence of the error on the signal. Combining these leads to the so-called enhanced error model for the inverse problem [47] with a likelihood distribution of the form

$$p(y|x) \sim \exp\left(-\frac{1}{2}\|L_\xi(\tilde{A}x - y + \eta_\xi)\|_{\mathcal{Y}}^2\right)$$

where L_ξ such that $L_\xi^T L_\xi = \Gamma_\xi^{-1}$ is a matrix square root such as the Cholesky decomposition of the inverse covariance matrix of the total error. If the measurement noise e is Gaussian white noise with zero mean and constant standard deviation σ , the above formula can be written as

$$p(y|x) \sim \exp\left(-\frac{1}{2\sigma}\|L_\varepsilon(\tilde{A}x - y + \eta_\varepsilon)\|_{\mathcal{Y}}^2\right)$$

where $L_\varepsilon^T L_\varepsilon = \Gamma_\varepsilon^{-1}$. This motivates writing the variational problem, or the maximum a posteriori (MAP) estimator, for (3.3) in the form

$$x^* = \arg \min_{x \in X} \frac{1}{2}\|L_\varepsilon(\tilde{A}x - y + \eta_\varepsilon)\|_{\mathcal{Y}}^2 + \lambda \mathcal{J}(x), \quad (3.6)$$

where $\mathcal{J}(x)$ is the regularisation functional corresponding to the prior $p(x)$ and $\lambda > 0$ is the regularisation parameter.

In order to compute solutions, the unknown distribution of the model error needs to be approximated. This can be obtained for example by simulations [3, 49] as follows. Let $\{x^i\}_{i=1,\dots,n}$ be the training set. The corresponding values of the model error are

$$\varepsilon^i = Ax^i - \tilde{A}x^i \quad (3.7)$$

and the mean and covariance of the model error can be estimated from the samples as

$$\eta_\varepsilon = \frac{1}{n} \sum_{i=1}^n \varepsilon^i \quad \text{and} \quad \Gamma_\varepsilon = \frac{1}{n-1} \sum_{i=1}^n \varepsilon^i \otimes \varepsilon^i - \eta_\varepsilon \otimes \eta_\varepsilon, \quad (3.8)$$

where \otimes denotes the tensor product.

The approximation error method has found many applications in inverse problems, partly due to its simplicity yet high effectiveness in compensating for model errors. Examples of situations where it has been used successfully include model mismatch [50], uncertainty in sensor locations [51], compensating for unknown boundary shapes [52], and even recent applications in wireless communication [53].

Despite these successes, it has been recently observed that the assumption of Gaussian distributed model errors as well as the independence assumption can be too restrictive, especially in nonlinear inverse problems [48, 54, 55]. This motivated the recent development of data-driven approaches for estimating non-Gaussian and non-independently distributed model errors as discussed in the following sections.

3.1.1 Sequential model correction

Let us first discuss briefly the work [48], which further examines the non-Gaussianity of model errors in the case where the accurate forward model A is nonlinear and the approximation \tilde{A} is given by a linear model. This leads to a successive linearised and convexified problem that can be solved in a sequential manner as we will outline below.

In this case, we write (3.1) again in terms of \tilde{A} , which yields the observation model

$$y = \tilde{A}x + A(x) - \tilde{A}x + e = \tilde{A}x + \varepsilon(x) + e. \quad (3.9)$$

We note that here the approximation creates a nonlinear approximation error, denoted by $\varepsilon(x)$, in contrast to the constant (i.e., independent of x) error in (3.5). Consequently, this formulation of the

model is still nonlinear as we have just moved the nonlinearity into $\varepsilon(x)$. Let us now assume that we have access to some initial reconstruction $x_0 \in \mathcal{X}$. We can then approximate the model error by its value at x_0

$$y \approx \tilde{A}x + \varepsilon(x_0) + e. \quad (3.10)$$

This leads to a convex variational formulation, given a convex regulariser \mathcal{J} ,

$$x^* = \arg \min_{x \in \mathcal{X}} \left\{ \|\tilde{A}x - (y - \varepsilon(x_0))\|_{\mathcal{Y}} + \lambda \mathcal{J}(x) \right\}, \quad (3.11)$$

which provides a local reconstruction depending on x_0 . From here it is natural to expand this construction into a sequence

$$x^{(k+1)} = S(x^{(k)}) = \arg \min_{x \in \mathcal{X}} \left\{ \|\tilde{A}x - (y - \varepsilon(x^{(k)}))\|_{\mathcal{Y}} + \lambda \mathcal{J}(x) \right\}. \quad (3.12)$$

We emphasise here that updating the sequence, i.e., solving the linearised and thus convex optimisation problem can be done efficiently with first-order optimisation methods.

This approach is useful if a computationally effective linear model \tilde{A} is available which makes (3.12) tractable. The update for the sequence then only requires one evaluation of the the accurate nonlinear forward model and the solution of the linearised problem, which is cheaper then computing the Fréchet derivative of the nonlinear model A . The publication [48] shows that a fixed linear approximation \tilde{A} performs well when compared to a scheme that updates the approximations between each sequential step. We also note that if one does not have access to the accurate model, or its evaluation even once is too expensive, a successive estimation of the model error could be computed or estimated [54]. Such a sequential update of the approximation error is left for future research.

3.2 Learned image reconstructions and implicit model corrections

Let us now move to data-driven approaches in the context of learned image reconstruction. Here, broadly speaking, we aim to find a parameterised reconstruction operator $\mathcal{R}_\theta : \mathcal{Y} \rightarrow \mathcal{X}$ whose parameters are learned from a suitable training set. This is most often achieved by utilising neural networks to parameterise the reconstruction operator; we refer to [56] for an overview of relevant methods.

In what follows, we are interested in the framework of learned iterative reconstructions [57, 58]. That is, we aim to find a network Λ_Θ which is designed to mimic a gradient descent scheme. In particular, we train the network to perform an iterative update of the following form

$$x^{(k+1)} = \Lambda_\Theta \left(\nabla_x \frac{1}{2} \|Ax^{(k)} - y\|_{\mathcal{Y}}^2, x^{(k)} \right), \quad (3.13)$$

where $\nabla_x \frac{1}{2} \|Ax^{(k)} - y\|_{\mathcal{Y}}^2 = A^*(Ax^{(k)} - y)$. If the accurate model is expensive to evaluate, computing the updates in (3.13) is expensive, which is especially problematic when training the networks. If the model is included in the training this quickly becomes intractable. Thus, one could use an approximate model \tilde{A} instead of the accurate model and compute an approximate gradient as $\tilde{A}^*(\tilde{A}x^{(k)} - y)$ for the update in (3.13), as proposed in [36]. The network Λ_Θ is then expected to *implicitly* correct the introduced model error to produce a new reliable iterate.

That means that correction and regularisation are trained simultaneously with the update in (3.13). Such approaches are typically trained by using a loss function, such as the L^2 -loss, to measure the distance between reconstruction and a ground truth phantom. This way a substantial speed-up, compared to classical variational approaches, can be achieved with improved reconstruction quality. In a recent paper [59] the implicit correction has been extended to a model corrected learned primal dual method, where separate updating operators are learned in primal and dual space, offering further improvements of reconstruction quality.

Nevertheless, such implicit corrections within a learned reconstruction operator offer limited insights into how approximate models are corrected for and so far have only limited convergence guarantees [60]. Thus, we will consider in the following an *explicit* correction that can be subsequently used in a variational framework.

3.3 Explicit model correction and a convergence result

Let us now consider corrections for the approximation error caused by the approximate model \tilde{A} via a parameterisable nonlinear mapping $F_\Theta : \mathcal{Y} \rightarrow \mathcal{Y}$, applied directly as correction to \tilde{A} as proposed in [34]. This mapping could be given by a (convolutional) neural network, but other options can be considered. This leads to a corrected operator A_Θ of the form

$$A_\Theta = F_\Theta \circ \tilde{A}. \quad (3.14)$$

We aim to choose the correction F_Θ such that ideally $A_\Theta(x) \approx Ax$ for those $x \in \mathcal{X}$ that we are interested in. The primary question that we aim to answer is, whether such corrected models (3.14) can be subsequently used in variational regularisation approaches. Thus, it is natural to require that the obtained solutions involving the corrected operator A_Θ and the accurate operator A , are close, that is

$$\arg \min_{x \in \mathcal{X}} \frac{1}{2} \|A_\Theta(x) - y\|_{\mathcal{Y}}^2 + \lambda \mathcal{J}(x) \approx \arg \min_{x \in \mathcal{X}} \frac{1}{2} \|Ax - y\|_{\mathcal{Y}}^2 + \lambda \mathcal{J}(x). \quad (3.15)$$

Solutions are then usually computed by an iterative algorithm. Here we consider first order methods to draw connections to learned iterative schemes as in (3.13). In particular, we consider a classical gradient descent scheme, assuming a differentiable \mathcal{J} . Given an initial guess x_0 , we compute a solution by the iterative process

$$x^{(k+1)} = x^{(k)} - \gamma_k \nabla_x \left(\frac{1}{2} \|A_\Theta(x^{(k)}) - y\|_{\mathcal{Y}}^2 + \lambda \mathcal{J}(x^{(k)}) \right), \quad (3.16)$$

with an appropriately chosen step size $\gamma_k > 0$. When using (3.16) for the corrected operator it seems natural to ask for a *gradient consistency* of the approximate gradient $\nabla_x \|A_\Theta(x) - y\|_{\mathcal{Y}}^2 \approx \nabla_x \|Ax - y\|_{\mathcal{Y}}^2$. We recall that the correction F_Θ in (3.14) is given by a nonlinear neural network and with the chain rule we obtain

$$\frac{1}{2} \nabla_x \|A_\Theta(x) - y\|_{\mathcal{Y}}^2 = \tilde{A}^* \left[DF_\Theta(\tilde{A}x) \right]^* \left(F_\Theta(\tilde{A}x) - y \right). \quad (3.17)$$

Here, we denote by $DF_\Theta(y)$ the Fréchet derivative of F_Θ at y , which is a linear operator $\mathcal{Y} \rightarrow \mathcal{Y}$. That means, to satisfy the gradient consistency condition, we would need

$$\tilde{A}^* \left[DF_\Theta(\tilde{A}x) \right]^* \left(F_\Theta(\tilde{A}x) - y \right) \approx A^*(Ax - y). \quad (3.18)$$

This reveals a problem: the range of the corrected fidelity term's gradient (3.17) is limited by the range of the approximate adjoint, $\mathcal{R}(\tilde{A}^*)$. Thus, we identify the key difficulty here in the differences of the range of the accurate and the approximate adjoints rather than the differences in the forward operators themselves. A correction of the forward operator via composition with a parametrised model F_Θ in measurement space is not able to produce gradients close to the gradients of the accurate data term if $\mathcal{R}(\tilde{A}^*)$ and $\mathcal{R}(A^*)$ are too different, see also Theorem 3.1 in [34].

3.3.1 Obtaining a Forward-Adjoint Correction

To achieve a gradient-consistent model correction two networks can be considered instead. That is, we learn a network F_Θ that corrects the forward model and another network G_Φ that corrects the adjoint, such that we have

$$A_\Theta := F_\Theta \circ \tilde{A}, \quad A_\Phi^* := G_\Phi \circ \tilde{A}^* \quad (3.19)$$

These corrections can then be obtained as follows. Given a set of training samples $\{x^i, y^i = Ax^i\}_{i=1,\dots,n}$, we train the forward correction F_Θ acting in measurement space \mathcal{Y} , for the adjoint we train the network G_Φ acting on image space \mathcal{X} using two losses

$$\min_{\Theta} \sum_i \|F_\Theta(\tilde{A}x^i) - Ax^i\|_{\mathcal{Y}} \quad \text{and} \quad \min_{\Phi} \sum_i \|G_\Phi(\tilde{A}^*r^i) - A^*r^i\|_{\mathcal{X}}, \quad (3.20)$$

where $r^i = F_\Theta(\tilde{A}x^i) - y^i$. This ensures that the adjoint correction is in fact trained in directions relevant when solving the variational problem. We can then use both corrections to compute approximate gradients of the data fidelity term $\|Ax - y\|_{\mathcal{Y}}^2$ as

$$A^*(Ax - y) \approx \left(G_\Phi \circ \tilde{A}^*\right) \left(F_\Theta(\tilde{A}x) - y\right). \quad (3.21)$$

A convergence result can be established by considering the two functionals corresponding to accurate and corrected operator as

$$\mathcal{L}(x) := \frac{1}{2}\|Ax - y\|_{\mathcal{Y}}^2 + \lambda\mathcal{J}(x), \quad \mathcal{L}_\Theta(x) := \frac{1}{2}\|A_\Theta(x) - y\|_{\mathcal{Y}}^2 + \lambda\mathcal{J}(x) \quad (3.22)$$

and using the forward-adjoint correction in the minimisation. We can then obtain, under suitable conditions outlined in [34], the following theorem.

Theorem 7 (Convergence to a neighbourhood of the accurate solution [34]). *Let $\epsilon > 0$ and suitable constant $C > 0$ (controlling the subdifferential of \mathcal{L}_Θ). Assume both adjoint and forward operator are fit up to a $C/4$ -margin, i.e.*

$$\|A\|_{\mathcal{X} \rightarrow \mathcal{Y}}\|(A - A_\Theta)(x^{(k)})\|_{\mathcal{Y}} < C/4, \quad \|(A^* - A_\Phi^*)(A_\Theta(x^{(k)}) - y)\|_{\mathcal{X}} < C/4 \quad (3.23)$$

for all y and $x^{(k)}$ obtained during gradient descent over \mathcal{L}_Θ . Then eventually the gradient descent dynamics over \mathcal{L}_Θ will reach an ϵ neighbourhood of the solution of the problem corresponding to the exact operator.

The proof of Theorem 7 relies on ensuring that the gradients (3.21) are pointing in the same direction to yield a descent direction with respect to the accurate functional. That is, we want to ensure that the angle between the approximate and the exact gradients is positive,

$$\cos \Phi_v(x) := \frac{\langle \nabla \mathcal{L}(x), \nabla^\dagger \mathcal{L}_\Theta(x) \rangle}{\|\nabla \mathcal{L}(x)\|^2} > 0, \quad (3.24)$$

where ∇^\dagger is used to indicate that we compute a corrected gradient of $\mathcal{L}_\Theta(x)$ using the forward-adjoint correction given by the right-hand side of (3.21). The experiments in [34] show that when this alignment is ensured during the minimisation then indeed one can observe convergence to the same neighbourhood as with the accurate model, while if a positive alignment is not ensured then the optimisation procedure will diverge.

In [34] it was proposed to use the so-called *recursive* training to ensure this positive alignment. It requires running the iterations for every training image x^i and adding the values of the forward operator and its adjoint along the trajectory to the training set. This can be very expensive computationally.

3.3.2 Limitations and extensions

The first, minor difficulty with this approach is that it requires training two networks, one for the forward operator A and one for its adjoint A^* . This can be avoided by using training data for the normal operator A^*A as in Section 2.3.2 and learning a single network $N_\Theta: \mathcal{X} \rightarrow \mathcal{X}$ to satisfy $N_\Theta(\tilde{A}^*\tilde{A}x^i) \approx A^*Ax^i$. This will be sufficient to approximate the gradient of the data fidelity term $\|Ax - y\|_{\mathcal{Y}}^2$, see (3.21).

A much more serious difficulty is that Theorem 7 requires that the trained networks approximate the exact forward and adjoint operators for all iterates $x^{(k)}$ obtained during gradient descent (and not just in the vicinity of training samples $\{x^i\}_{i=1,\dots,n}$). This requires very cumbersome and computationally expensive recursive training discussed above.

A possible remedy for this is to use iterative algorithms that stay in the vicinity of the training images $\{x^i\}_{i=1,\dots,n}$, such as the projected gradient scheme in [61] where the authors consider a constrained variational problem over a manifold. A difficulty is, however, that in our setting the manifold is given implicitly via training samples $\{x^i\}_{i=1,\dots,n}$ and needs to be estimated “on the fly” as the iterations proceed. This is currently work in progress.

3.3.3 Connections to regularisation by projection

The simplified model \tilde{A} can also be used in the context of regularisation by projection (see Section 2.3). While the learned linear model $AP_{\mathcal{X}_n}$ is exact on the subspace \mathcal{X}_n spanned by the training images $\{x^i\}_{i=1,\dots,n}$, on the complement of this subspace the learned model is zero, and it may be beneficial to use the simplified model \tilde{A} instead. This will lead to the following approximation of the forward operator

$$A \approx AP_{\mathcal{X}_n} + \tilde{A}(Id - P_{\mathcal{X}_n})$$

The corresponding variational problem will then read as follows (cf. (2.11))

$$\min_{x \in \mathcal{X}} \frac{1}{2} \left\| \left[AP_{\mathcal{X}_n} + \tilde{A}(Id - P_{\mathcal{X}_n}) \right] x - y^\delta \right\|^2 + \alpha \mathcal{J}(x) \quad (3.25)$$

and the gradient descent iteration will become

$$\begin{aligned} x^{(k+1)} = x^{(k)} & - \tau_k \left(\left[P_{\mathcal{X}_n} A^* AP_{\mathcal{X}_n} + (Id - P_{\mathcal{X}_n}) \tilde{A}^* \tilde{A} (Id - P_{\mathcal{X}_n}) \right] x^{(k)} \right. \\ & \left. - \left(P_{\mathcal{X}_n} A^* + (Id - P_{\mathcal{X}_n}) \tilde{A}^* \right) y^\delta + \alpha q_k \right), \quad q_k \in \partial \mathcal{J}(x^{(k)}). \end{aligned} \quad (3.26)$$

All operators here can be computed without numerical access to the exact model A , relying only on the training pairs $\{x^i, y^i = Ax^i\}_{i=1,\dots,n}$ and the simplified model \tilde{A} . (Recall that the operator $P_{\mathcal{X}_n} A^* = (AP_{\mathcal{X}_n})^*$ can be computed using these training pairs via (2.14)).

Alternatively, the iteration (2.16), based on learning the normal operator on \mathcal{X}_n , can be augmented with an approximate component on \mathcal{X}_n^\perp

$$\begin{aligned} x^{(k+1)} = x^{(k)} & - \tau_k \left(\left[A^* AP_{\mathcal{X}_n} + \tilde{A}^* \tilde{A} (Id - P_{\mathcal{X}_n}) \right] x^{(k)} \right. \\ & \left. - \left(A^* P_{\mathcal{Y}_n} + \tilde{A}^* (Id - P_{\mathcal{Y}_n}) \right) y^\delta + \alpha q_k \right). \end{aligned} \quad (3.27)$$

This iteration only requires the training data $\{x^i, z^i = A^* Ax^i\}_{i=1,\dots,n}$ for the normal operator and the simplified model \tilde{A} along with its adjoint \tilde{A}^* .

Numerical experiments with these approaches will be presented in Section 4.2.

4 Photoacoustic Tomography (PAT)

In this section we will apply the presented approaches to Photoacoustic Tomography (PAT). To do that, let us first briefly discuss the PAT forward problem and then introduce the analytic approximate model for the context of model corrections.

To create the measured signal in PAT, biological tissue is exposed to a sufficiently short near-infrared light pulse that is then absorbed by chromophores. This results in a spatially-varying pressure increase, which initiates an Ultrasound (US) pulse, that then propagates to the tissue surface. The measurement consequently consists of the detected waves in space-time at the boundary of the tissue. This time evolution of the photoacoustic wave can be modelled using the equations of linear acoustics [62, 63], and can be described as an initial value problem for the wave equation with spatial coordinates $\zeta \in \mathbb{R}^2$ and time $t \geq 0$

$$(\partial_{tt} - c^2 \Delta) p(\zeta, t) = 0, \quad (4.1a)$$

$$p(\zeta, t = 0) = p_0(\zeta), \quad (4.1b)$$

$$\partial_t p(\zeta, t = 0) = 0, \quad (4.1c)$$

where c is the speed of sound. The measurement of the time series is modelled as a linear operator \mathcal{M} acting on the pressure field $p(\zeta, t)$ restricted to the boundary Γ of the computational domain and a finite time window:

$$y = \mathcal{M} p|_{\Gamma \times (0, T)}. \quad (4.2)$$

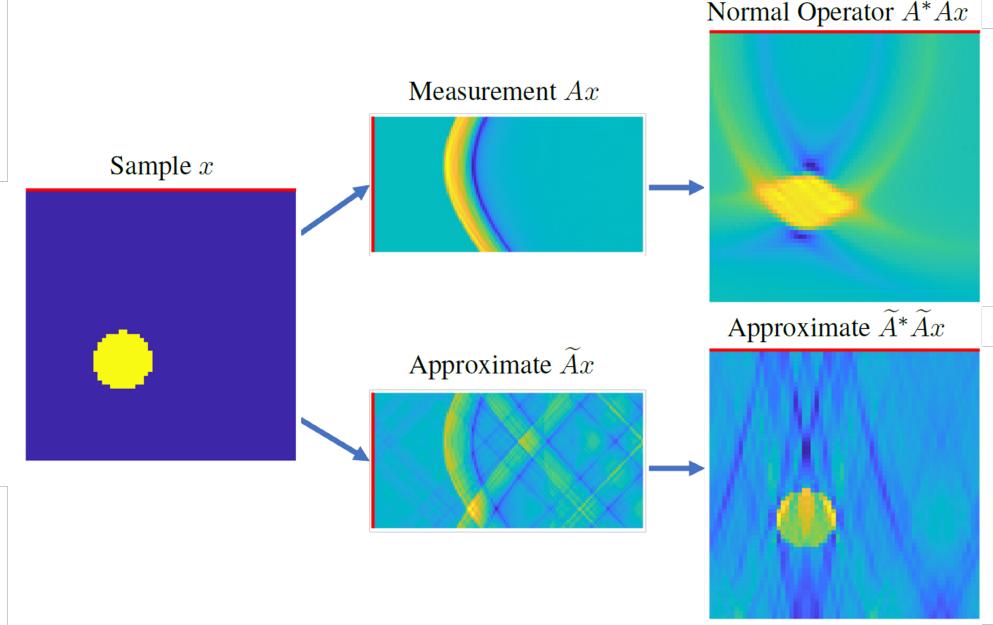


Figure 1: Illustration of the forward operator A and the approximate model \tilde{A} , as well as corresponding normal operators. The sensor is indicated with a red line (left/right images), in the measurements (middle) the red line corresponds to initial time $t = 0$.

Together, equations (4.1a) and (4.2) define the linear forward model A that maps the initial pressure $x = p_0(\zeta)$ to the measured time series y . This forward model can be accurately simulated by a pseudo-spectral time-stepping model as outlined in [63, 64]. While providing an efficient implementation, time-stepping can take a considerable amount of time depending on a possibly fine time discretisation.

Thus, we can consider a model that eliminates the time stepping and replaces it with one (Fast) Fourier transform. First, we can consider the case where measurement points lie on a line ($\zeta_2 = 0$) outside the support of x . Then under the assumption of constant speed-of-sound, the pressure on the sensor can be related to x as follows [62, 65]

$$p(\zeta_1, t) = \frac{1}{c^2} \mathcal{F}_{k_1}^{-1} \{ \mathcal{C}_\omega \{ B(k_1, \omega) \tilde{x}(k_1, \omega) \} \}, \quad (4.3)$$

where $\tilde{x}(k_1, \omega)$ is obtained from the Fourier transform $\hat{x}(k) = \mathcal{F}_\zeta \{ x(\zeta) \}$ via the dispersion relation $(\omega/c)^2 = k_1^2 + k_2^2$, \mathcal{C}_ω is a cosine transform from ω to t , and $\mathcal{F}_{k_1}^{-1}$ is the 1D inverse Fourier Transform from k_1 to ζ_1 on the detector. The weighting factor,

$$B(k_1, \omega) = \omega / \left(\text{sgn}(\omega) \sqrt{(\omega/c)^2 - k_1^2} \right), \quad (4.4)$$

contains an integrable singularity which means that if (4.3) is evaluated by discretisation on a rectangular grid (enabling the application of FFT for efficient calculation), then aliasing will appear in the measured data $p(\zeta_1, t)$. Consequently, evaluating (4.3) using FFT leads to a *fast but approximate* forward model. We can control the degree of aliasing by avoiding waves that arrive close to parallel at the sensor. This could be included in the model as an angular thresholding to control the degree of aliasing, we refer to [36] to a more detailed discussion. Either way, with or without angular thresholding in the weighting factor B , the relation (4.3) defines the approximate model \tilde{A} used in what follows. The difference between the accurate model A and the approximate model \tilde{A} is shown in Figure 1. The aliasing artefacts are clearly visible in the data space and the resulting normal operator does carry wrong information for the reconstruction.

4.1 Training data

We will consider a computational domain $(\zeta_1, \zeta_2) \in \Omega = [0, 1] \times [0, 1]$ with a rectangular discretisation of 64×64 pixels. The measurements are taken at the top of the domain. The background value is set to zero and we sample a number of indicator functions of discs located randomly in the domain with parameters uniformly distributed as follows: centre $(\zeta_1, \zeta_2) \in [0.25, 0.75] \times [0.25, 0.75]$, radius $r \in [0.1, 0.2]$. Each disc has a random initial pressure value $p_0 \in [0.5, 1]$.

4.2 Experiments with projected variational regularisation

In this section we present numerical experiments with the method described in Section 2.3. As the regulariser \mathcal{J} we take the Total Variation (TV), which we define as a functional on $L^2(\Omega)$ extending it with the value $+\infty$ on $L^2(\Omega) \setminus BV(\Omega)$. This is a common setting in imaging [66].

To orthonormalise the training images $\{x^i\}_{i=1,\dots,n}$, we use the modified Gram-Schmidt algorithm [67]. Due to its numerical instability, we restrict ourselves to $n = 1500$ training samples.

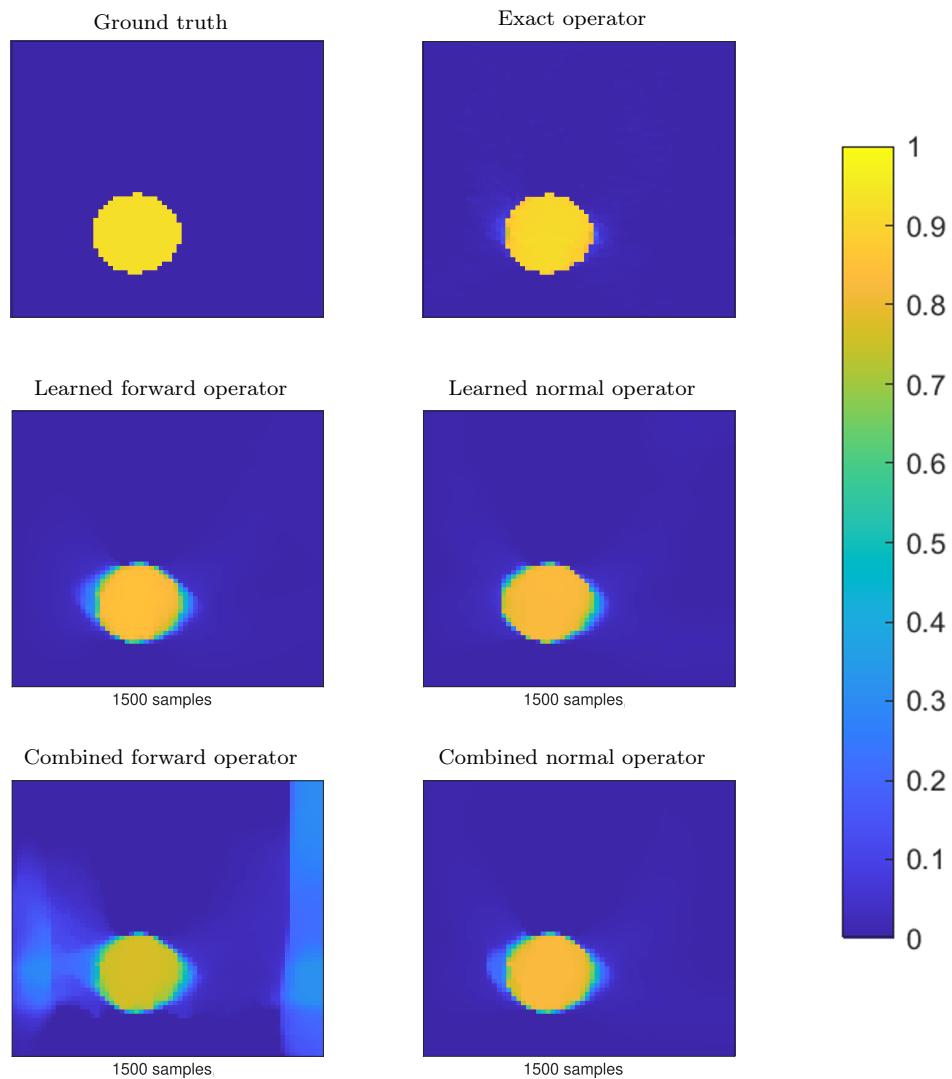


Figure 2: Projected variational regularisation. Experiments with $n = 1500$ training samples. Both learned forward operator and learned normal operator perform well, the reconstructions with the learned normal operator being perhaps a bit sharper. Surprisingly, combining the learned forward model with an approximate one \tilde{A} decreases the reconstruction quality.

We use the iterations (2.13) – learned forward operator, (2.16) – learned normal operator, (3.26) – learned forward operator combined with approximate model, and (3.27) – learned normal operator

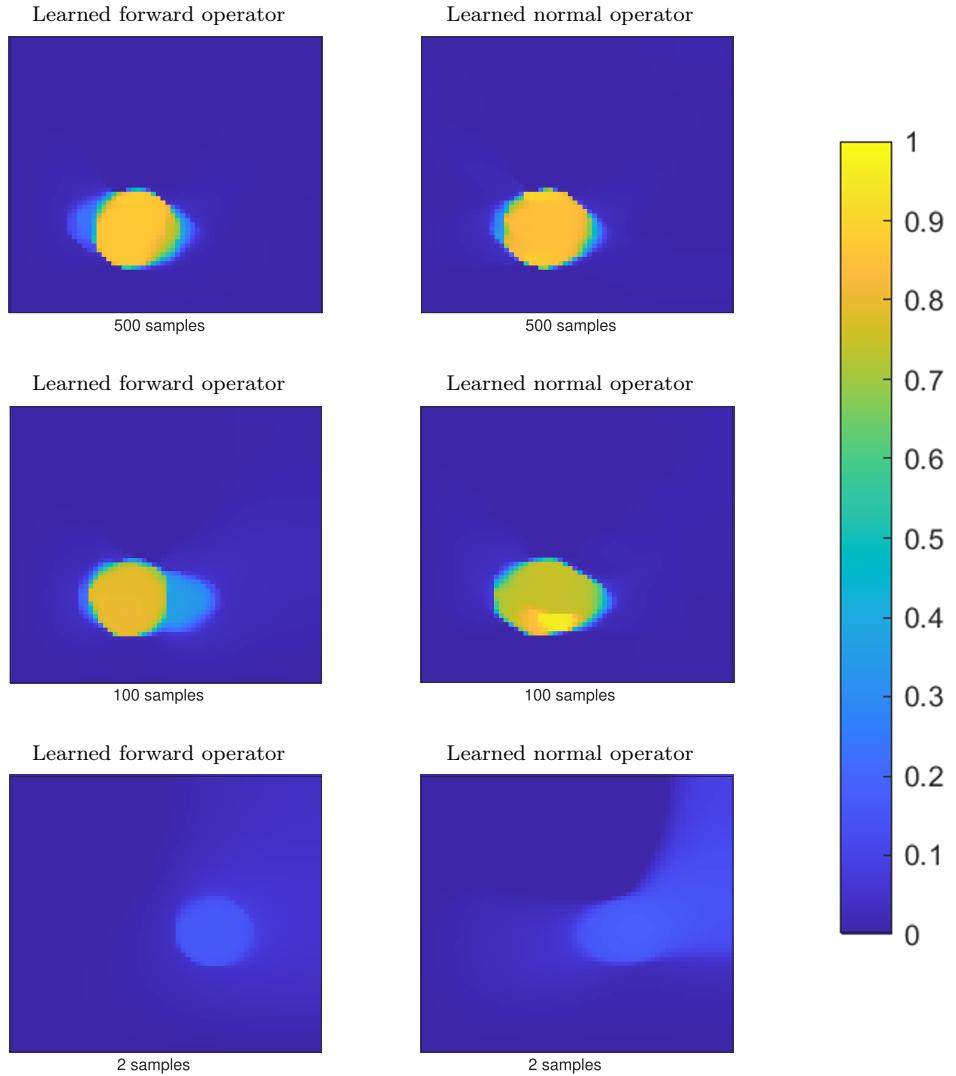


Figure 3: Projected variational regularisation. The effect of the number of training samples. The method is surprisingly robust with respect to the number of samples. Reconstructions are still reasonable for as few as 500 samples (top row), and even for 100 (middle row) samples the location of the disc is identified reasonably well. If we decrease the number of samples further so that the support of the samples does not overlap with the support of the ground truth then the location of the disc is identified incorrectly (bottom row).

combined with approximate model. The results are shown in Figure 2. The top row shows the ground truth and the reconstruction with the exact operator, which is our golden standard in these experiments. Performing a basic parameter search, we find a reasonable value of the regularisation parameter $\alpha = 2 \cdot 10^{-4}$ which yields a relative reconstruction error of 6%.

With a learned forward (middle row left) and a learned normal operator (middle row right) we obtain a relative reconstruction error of 23 – 24%, also after performing a basic parameter search and finding a reasonable value of $\alpha = 2 \cdot 10^{-2}$. The value of the parameter is higher, as expected in a problem with operator errors. The reconstruction obtained with a learned normal operator looks sharper than the one obtained with a learned forward operator, but overall the reconstruction quality is comparable.

Using a combined model as in (3.26) and (3.27), surprisingly, yields inferior performance. For the combined forward operator (bottom row left) we see artefacts outside of the support of the discs in the training set; this is the region where only the approximate model is available. The learned normal operator (bottom row right) is affected less, possibly because the support of the

backprojections $\{z^i\}_{i=1,\dots,n}$ is larger than that of the training images $\{x^i\}_{i=1,\dots,n}$. However, the reconstruction still looks blurrier than without the approximate model (middle row right).

In Figure 3 we investigate the influence of the size of the training set n on the reconstruction quality. We find that the method is very robust. For as few as $n = 500$ samples we still obtain a reasonable reconstruction (top row), the learned normal operator (right) performing slightly better than the learned forward operator (left). Even for $n = 100$ the location of the disc is identified reasonably well, even if the shape is not well reconstructed (middle row; the regularisation parameter has to be chosen larger in this case to ensure stability). The reason for such robustness seems to be that the learned model “sees” the area where the ground truth disc is located because there are other discs nearby in the training set. If we further deflate the training set to an extremely low size of $n = 2$ so that there is no overlap between the training discs and the exact solution, then the support of the reconstruction is completely off (bottom row).

4.3 Experiments with learned model correction

We continue with experiments with the second approach considered in this chapter, the learned model correction. As we have seen in Figure 1, the approximate model introduces artefacts that will cause the gradients to be incorrect, and hence a correction needs to be applied. As discussed in Section 3, there are several ways to achieve such a model correction, classified into implicit and explicit approaches. Here, we will only discuss the explicit corrections.

The experiments are also conducted on the disc data set and we compare learning a correction for the forward operator only with corrections for both the forward operator and its adjoint. Additionally, we discuss the influence of training only on the data manifold of ground-truth samples $\{x^i\}_{i=1,\dots,n}$ or along the trajectory (recursive training) as discussed in Section 3.3.1. Experiments presented here were first reported in [34]. The reconstructions are shown in Figure 4 and the convergence plots in Figure 5.

In Figure 4 we see that the accurate model with total variation regularisation—our reference solution—produces a reasonably good reconstruction with an average error of 12% over the 64 test samples, but due to strong limited view artefacts we still have a slight smearing visible. The approximate and uncorrected forward operator \tilde{A} is not able to produce a sufficiently good reconstruction and causes strong artefacts in the background medium. This is also reflected in the average reconstruction error of 55%. The classical approximation error method is indeed able to correct the strong artefacts and reaches a relative error of 32%, but results in a loss of contrast and thus wrong quantitative values. It can be also seen in Figure 5 that the convergence is much slower.

Let us now examine the learned model corrections using a neural network, that is $A_\Theta := F_\Theta \circ \tilde{A}$ for the forward correction and, if used, $A_\Phi^* := G_\Phi \circ \tilde{A}^*$ for the adjoint correction. We can see a clear difference between training along the trajectory compared to training on the data manifold only. For both corrections, forward and forward-adjoint, the training on the simple data manifold is not sufficient and leads to strong artefacts, resulting in an average relative error of 53% and 41% respectively. This is due to the correction not being valid near the point of convergence and hence the conditions of Theorem 7 are violated (recall that the theorem requires that the correction is valid for all iterates $x^{(k)}$). If we make sure that the correction is valid for all iterates by training the networks with (3.20) not only on the data manifold of ground truth samples, but also for all $x^{(k)}$ that arise during the optimisation procedure, then we can ensure convergence to a neighbourhood of the accurate minimiser as stated by Theorem 7 for the forward-adjoint correction. This is clearly seen in the reconstruction that is visually close to the accurate model and also reflected in the average relative error of 14% closest to the accurate model.

The effect of gradient alignment can also be observed in the convergence plots in Figure 5. Recursive training, both for the forward and forward-adjoint corrections, ensures positive alignment between the true and approximate gradients and results in convergence of the data misfit (measured using the accurate model). Without recursive training, both forward and forward-adjoint corrections result in non-convergence of the data misfit.

Unfortunately, this improvement in reconstruction quality requires an expensive training proce-

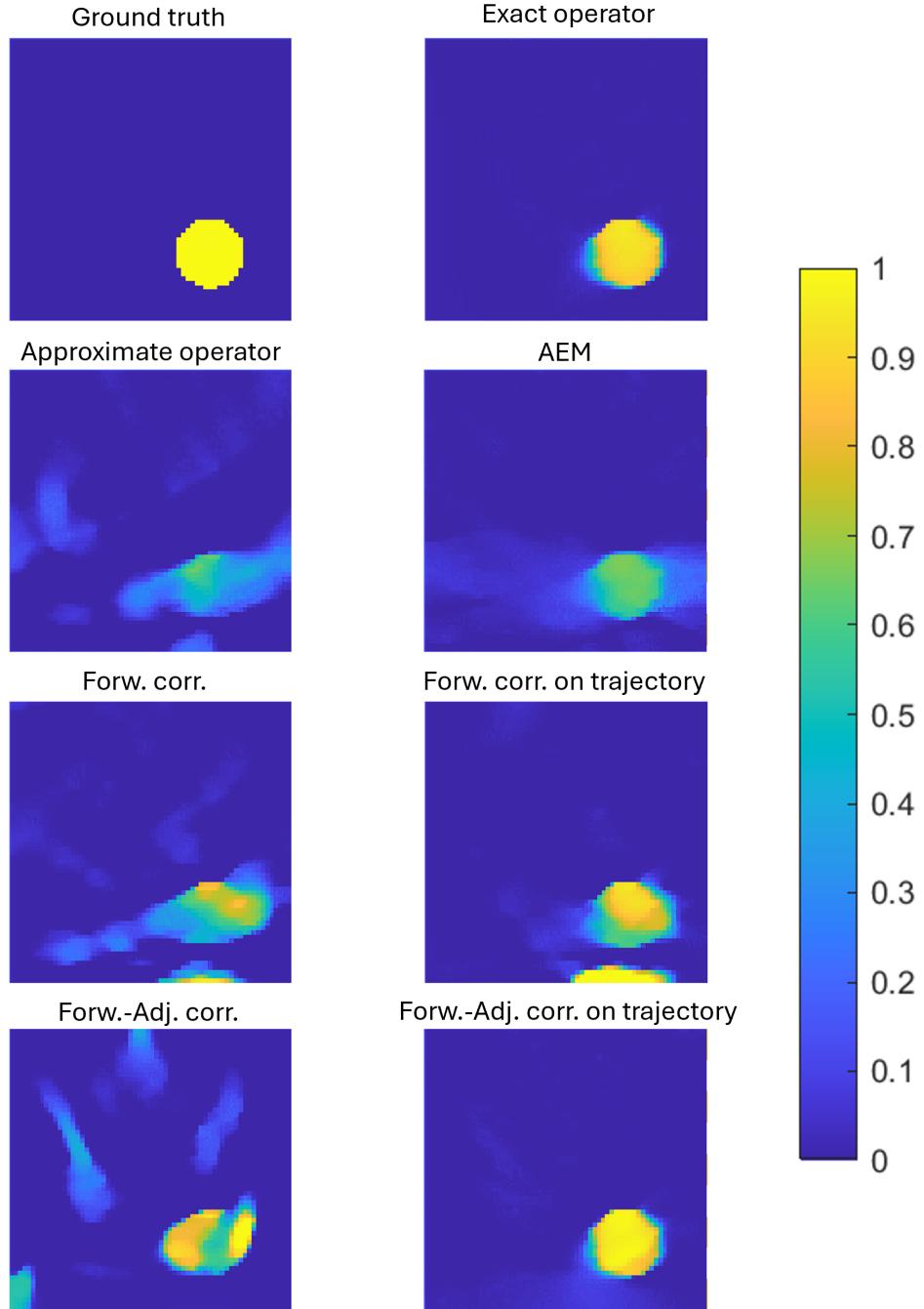


Figure 4: Comparison of reconstructions using the approximate model and learned corrections under various training schemes. (Top row) Ground truth and reconstruction with the accurate model, the sensor is located at the top. (Second row) Reconstruction using the approximate operator \tilde{A} and reconstruction using the approximation error method. (Third row) Learned correction of the forward model trained on disks only (left) and recursively on the trajectory (right). (Bottom row) Forward-adjoint correction trained on disks only (left) and recursively on the trajectory (right).

dure that takes about 4 days in total to train the networks carefully, whereas training on the data manifold of discs and corresponding measurements takes only a few hours for the forward-adjoint correction. This indicates that there is a need to improve training strategies as we discuss next.

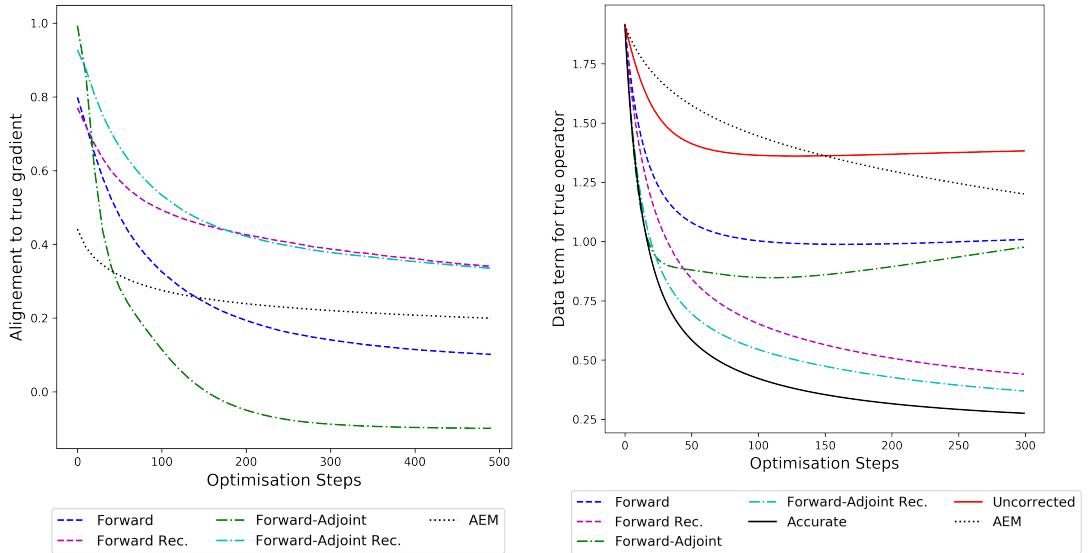


Figure 5: (Left) Alignment (3.24) of the approximate gradient and the gradient of the accurate data term $A^*(Ax^{(k)} - y)$ for different training approaches. (Right) True data term $\|Ax^{(k)} - y\|_{\mathcal{Y}}$ for different methods, tracked throughout the gradient descent scheme.

4.3.1 Training without trajectory

We have seen in the previous section that training along a trajectory is crucial for the success of the explicit learned model correction when solving the variational problem, which is in accordance with Theorem 7 requiring a valid approximation for all iterates $x^{(k)}$ obtained during minimisation. The high computational cost of such recursive training makes it necessary to think about alternatives. One possibility is to restrict the minimisation of the variational problem

$$\arg \min_{x \in \mathcal{X}} \frac{1}{2} \|A_\Theta(x) - y\|_{\mathcal{Y}}^2 + \lambda \mathcal{J}(x) \quad (4.5)$$

to a suitable set, such as the manifold representing the data distribution $\{x^i\}_{i=1,\dots,n}$. This way, the correction can be trained just on the manifold itself and constraining the trajectory to the vicinity of the manifold will ensure that the correction is valid for all $x^{(k)}$.

We present, in Figure 6, a proof-of-concept result for the hypothesis that training and optimisation over the data manifold can eliminate the need for trajectory training. We see that solving the variational problem on the manifold works well for the accurate model, whereas the approximate model suffers loss of contrast and sharpness. If we use the corrected model trained on the data manifold, but optimisation is performed freely in the full space, we introduce strong artefacts, as the correction is not valid for all iterates. However, when the optimisation path is restricted to the manifold we obtain a result close to that of the accurate model. Training of the model correction on the data manifold only takes roughly 90 minutes, compared to the full trajectory training that requires 4 days. This is a promising solution to the trajectory training problem and is currently work in progress.

5 Summary and conclusions

Inverse problems involving high dimensions and/or computationally expensive forward operators necessitate the use of computationally cheaper approximate models. This applies, e.g., to settings when imaging is performed in time-critical scenarios or when the forward operator is called many times within a larger pipeline of a learning framework.

In this chapter we have discussed two data-driven approaches, learning an approximation of the forward or the inverse operator using data-driven projections [33], and a data-driven correction to

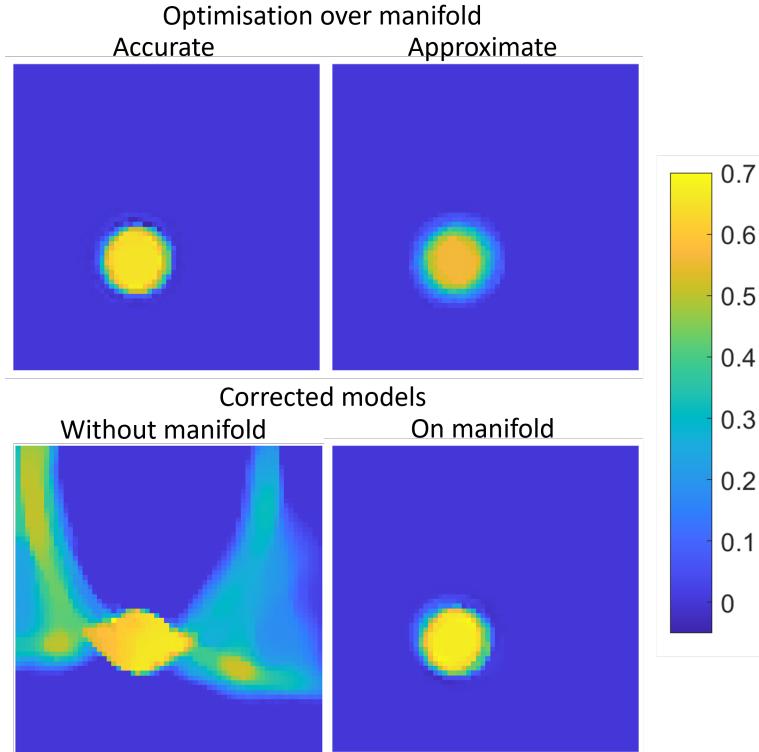


Figure 6: Proof-of-concept results for training and optimisation over the data manifold instead of the trajectory. (Top) optimisation is carried out on the data manifold. This already improves reconstructions with the approximate model, but results in loss of contrast. (Bottom) Corrected models trained on data manifold only. On the left optimisation is carried out over all \mathcal{X} and on the right only on the data manifold.

an analytic approximation [34]. A common theme has emerged, which is that in addition to learning an approximation of the forward operator, one often needs to learn a separate approximation of the adjoint.

While both approaches provide a possible solution to the problem of computationally expensive models in inverse problems, they also come with some drawbacks. The data-driven projection method requires a good quality *linear* approximation of the ground truth by training images, hence, for example, it is sensitive to shifting the image. If the forward operator is shift-equivariant (e.g., a convolution), it could be possible to incorporate a non-linear ‘‘projection’’ onto the training set by finding the shift that minimises the distance between the image and the span of the training data (a similar approach can be applied, e.g., to rotations). However, this is a research direction not yet taken.

In the case of learned model corrections we have discussed that while solutions are faster to compute, the computational burden moves to an expensive training phase that needs to ensure validity of the correction for all iterates of the optimisation trajectory to ensure convergence. Finally, we presented a proof-of-concept solution that may overcome this problem by limiting training and optimisation to the data manifold, which is an interesting direction for future studies.

Acknowledgements

SA and YK acknowledge EPSRC grant EP/V026259/1. YK also acknowledges the EPSRC Fellowship EP/V003615/2 and the support of the National Physical Laboratory. AH acknowledges support by the Research Council of Finland: Academy Research Fellowship (Project 338408), the Centre of Excellence of Inverse Modelling and Imaging project (Project 353093), and the Flagship of Advanced Mathematics for Sensing Imaging and Modelling grant (Project 359186).

The authors would also like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme Mathematics of deep learning (supported by EPSRC grant EP/R014604/1).

References

- [1] A. C. Kak and M. Slaney. *Principles of computerized tomographic imaging*. SIAM, 2001.
- [2] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction*. Society for Industrial and Applied Mathematics, 2001.
- [3] S. R. Arridge et al. “Approximation errors and model reduction with an application in optical diffusion tomography”. In: *Inverse Problems* 22.1 (2006), p. 175.
- [4] V. Kolehmainen et al. “Approximation errors and model reduction in three-dimensional diffuse optical tomography”. In: *JOSA A* 26.10 (2009), pp. 2257–2268.
- [5] P. Benner et al. *Model reduction and approximation: theory and algorithms*. SIAM, 2017.
- [6] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*. Vol. 92. Springer, 2015.
- [7] J. Dölz, H. Egger, and M. Schlottbom. “A model reduction approach for inverse problems with operator valued data”. In: *Numerische Mathematik* 148 (2021), pp. 889–917.
- [8] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations*. Springer, Cham, 2016.
- [9] J. Feliu-Fabà, Y. Fan, and L. Ying. “Meta-learning pseudo-differential operators with deep neural networks”. In: *Journal of Computational Physics* 408 (2020), p. 109309.
- [10] R. Vidal, Y. Ma, and S. Sastry. “Generalized principal component analysis (GPCA)”. In: *IEEE transactions on pattern analysis and machine intelligence* 27.12 (2005), pp. 1945–1949.
- [11] S. D. You and M.-J. Hung. “Reducing Dimensionality of Spectro-Temporal Data by Independent Component Analysis”. In: *2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*. 2020, pp. 93–97.
- [12] K. Lee and K. T. Carlberg. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”. In: *Journal of Computational Physics* 404 (2020), p. 108973.
- [13] A. Neubauer and O. Scherzer. “Finite-dimensional approximation of Tikhonov regularized solutions of nonlinear ill-posed problems”. In: *Inverse Problems* 11.1-2 (1990), pp. 85–99.
- [14] R. Plato and G. Vainikko. “On the regularization of projection methods for solving III-posed problems”. In: *Numerische Mathematik* 57 (1990), pp. 63–79.
- [15] B. Kaltenbacher. “Regularization by projection with a posteriori discretization level choice for linear and nonlinear ill-posed problems”. In: *Inverse Problems* 16.5 (2000), p. 1523.
- [16] C. Pöschl, E. Resmerita, and O. Scherzer. “Discretization of variational regularization in Banach spaces”. In: *Inverse Problems* 26.10 (2010), p. 105017.
- [17] U. Hämarik et al. “Regularization by discretization in Banach spaces”. In: *Inverse Problems* 32.3 (2016), p. 035004.
- [18] K. Bredies, B. Kaltenbacher, and E. Resmerita. “The least error method for sparse solution reconstruction”. In: *Inverse Problems* 32.9 (2016), p. 094001.
- [19] N. Kovachki et al. “Neural operator: Learning maps between function spaces with applications to PDEs”. In: *Journal of Machine Learning Research* 24.89 (2023), pp. 1–97.
- [20] Z. Li et al. “Fourier neural operator for parametric partial differential equations”. In: *arXiv preprint arXiv:2010.08895* (2020).

- [21] L. Lu et al. “Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators”. In: *Nature machine intelligence* 3.3 (2021), pp. 218–229.
- [22] F. Bartolucci et al. “Representation Equivalent Neural Operators: a Framework for Alias-free Operator Learning”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [23] N. H. Nelsen and A. M. Stuart. “The random feature model for input-output maps between banach spaces”. In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3212–A3243.
- [24] S. Lanthaler and N. H. Nelsen. “Error bounds for learning with vector-valued random features”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [25] M. V. de Hoop et al. “Convergence Rates for Learning Linear Operators from Noisy Data”. In: *SIAM/ASA Journal on Uncertainty Quantification* 11.2 (2023), pp. 480–513.
- [26] M. Mollenhauer, N. Mücke, and T. Sullivan. “Learning linear operators: Infinite-dimensional regression as a well-behaved non-compact inverse problem”. In: *arXiv preprint arXiv:2211.08875* (2022).
- [27] L. Herrmann, C. Schwab, and J. Zech. “Neural and gpc operator surrogates: construction and expression rate bounds”. In: *arXiv preprint arXiv:2207.04950* (2022).
- [28] A. Cohen, R. DeVore, and C. Schwab. “Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs”. In: *Foundations of Computational Mathematics* 10.6 (2010), pp. 615–646.
- [29] Y. Korolev. “Two-layer neural networks with values in a Banach space”. In: *SIAM Journal on Mathematical Analysis* 54.6 (2022), pp. 6358–6389.
- [30] C. Arndt et al. “Invertible residual networks in the context of regularization theory for linear inverse problems”. In: *arXiv preprint arXiv:2306.01335* (2023).
- [31] D. N. Tanyu et al. “Deep learning methods for partial differential equations and related parameter identification problems”. In: *Inverse Problems* 39.10 (2023), p. 103001.
- [32] N. B. Kovachki, S. Lanthaler, and A. M. Stuart. “Operator Learning: Algorithms and Analysis”. In: *arXiv preprint arXiv:2402.15715* (2024).
- [33] A. Aspri, Y. Korolev, and O. Scherzer. “Data driven regularization by projection”. In: *Inverse Problems* 36.12 (2020), p. 125009.
- [34] S. Lunz et al. “On Learned Operator Correction in Inverse Problems”. In: *SIAM Journal on Imaging Sciences* 14.1 (2021), pp. 92–127.
- [35] A. Aspri et al. “Data Driven Reconstruction Using Frames and Riesz Bases”. In: *Deterministic and Stochastic Optimal Control and Inverse Problems*. CRC Press, 2021, pp. 303–318.
- [36] A. Hauptmann et al. “Approximate k-space models and deep learning for fast photoacoustic reconstruction”. In: *Machine Learning for Medical Image Reconstruction: First International Workshop, MLMIR 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings* 1. Springer, 2018, pp. 103–111.
- [37] H. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Springer, 1996.
- [38] T. I. Seidman. “Nonconvergence Results for the Application of Least-Squares Estimation to Ill-Posed Problems”. In: *Journal of Optimization Theory and Applications* 30.4 (1980), pp. 535–547.
- [39] A. R. Barron et al. “Approximation and learning by greedy algorithms”. In: *The Annals of Statistics* 36.1 (2008), pp. 64–94.
- [40] F. Bach. “Breaking the Curse of Dimensionality with Convex Neural Networks”. In: *Journal of Machine Learning Research* 18.19 (2017), pp. 1–53.
- [41] M. Burger and H. W. Engl. “Training neural networks with noisy data as an ill-posed problem”. In: *Advances in Computational Mathematics* 13.4 (2000), pp. 335–354.

- [42] J. B. Conway. *A Course in Functional Analysis*. Springer, 1985.
- [43] A. N. Tikhonov et al. *Numerical Methods for the Solution of Ill-Posed Problems*. Dordrecht: Kluwer, 1995.
- [44] L. I. Rudin, S. Osher, and E. Fatemi. “Nonlinear total variation based noise removal algorithms”. In: *Physica D: Nonlinear Phenomena* 60.1 (1992), pp. 259–268.
- [45] M. Benning and M. Burger. “Modern Regularization Methods for Inverse Problems”. In: *Acta Numerica* 27 (2018), pp. 1–111.
- [46] M. Grant and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. <http://cvxr.com/cvx>. 2014.
- [47] J. Kaipio. *Statistical and computational inverse problems*. Vol. 160. Springer Science & Business Media, 2006.
- [48] A. Arjas, M. J. Sillanpää, and A. Hauptmann. “Sequential model correction for nonlinear inverse problems”. In: *SIAM Journal on Imaging Sciences* ((to appear)).
- [49] T. Tarvainen et al. “Bayesian image reconstruction in quantitative photoacoustic tomography”. In: *IEEE transactions on medical imaging* 32.12 (2013), pp. 2287–2298.
- [50] T. Tarvainen et al. “An approximation error approach for compensating for modelling errors between the radiative transfer equation and the diffusion approximation in diffuse optical tomography”. In: *Inverse Problems* 26.1 (2009), p. 015005.
- [51] T. Sahlström et al. “Modeling of errors due to uncertainties in ultrasound sensor locations in photoacoustic tomography”. In: *IEEE Transactions on Medical Imaging* 39.6 (2020), pp. 2140–2150.
- [52] “Approximation error method for imaging the human head by electrical impedance tomography”. In: *Inverse Problems* 37.12 (2021), p. 125008.
- [53] L. Marata et al. “Joint Activity Detection and Channel Estimation for Clustered Massive Machine Type Communications”. In: *IEEE Transactions on Wireless Communications* (2023).
- [54] D. Smyl et al. “Learning and correcting non-Gaussian model errors”. In: *Journal of Computational Physics* 432 (2021), p. 110152.
- [55] R. Nicholson et al. “On global normal linear approximations for nonlinear Bayesian inverse problems”. In: *Inverse Problems* 39.5 (2023), p. 054001.
- [56] S. R. Arridge et al. “Solving inverse problems using data-driven models”. In: *Acta Numerica* 28 (2019), pp. 1–174.
- [57] J. Adler and O. Öktem. “Solving ill-posed inverse problems using iterative deep neural networks”. In: *Inverse Problems* 33.12 (2017), p. 124007.
- [58] A. Hauptmann et al. “Model-based learning for accelerated, limited-view 3-D photoacoustic tomography”. In: *IEEE transactions on medical imaging* 37.6 (2018), pp. 1382–1393.
- [59] A. Hauptmann and J. Poimala. “Model-corrected learned primal-dual models for fast limited-view photoacoustic tomography”. In: *arXiv preprint arXiv:2304.01963* (2023).
- [60] S. Mukherjee et al. “Learned reconstruction methods with convergence guarantees: a survey of concepts and applications”. In: *IEEE Signal Processing Magazine* 40.1 (2023), pp. 164–182.
- [61] A. Hauswirth et al. “Projected gradient descent on Riemannian manifolds with applications to online power system optimization”. In: *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2016, pp. 225–232.
- [62] B. T. Cox and P. C. Beard. “Fast calculation of pulsed photoacoustic fields in fluids using k-space methods”. In: *The Journal of the Acoustical Society of America* 117.6 (2005), pp. 3616–3627.

- [63] B. E. Treeby and B. T. Cox. “k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields”. In: *Journal of biomedical optics* 15.2 (2010), p. 021314.
- [64] B. E. Treeby et al. “Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using ak-space pseudospectral method”. In: *The Journal of the Acoustical Society of America* 131.6 (2012), pp. 4324–4336.
- [65] K. P. Köstli et al. “Temporal backward projection of optoacoustic pressure transients using Fourier transform methods”. In: *Physics in Medicine & Biology* 46.7 (2001), p. 1863.
- [66] A. Chambolle and P.-L. Lions. “Image recovery via total variation minimization and related problems”. In: *Numerische Mathematik* 76.2 (1997), pp. 167–188.
- [67] L. N. Trefethen and D. Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.