Training Adaptive Reconstruction Networks for Blind Inverse Problems

Alban Gossard, Pierre Weiss

Training Adaptive Reconstruction Networks for Blind Inverse Problems

Alban Gossard[1,2] alban.paul.gossard@gmail.com

Pierre Weiss[1,2] pierre.armand.weiss@gmail.com

[1] Institut de Mathématiques de Toulouse; UMR5219; Université de Toulouse; CNRS

[2] UPS, F-31062 Toulouse Cedex 9, France

October 13, 2023

Abstract

Neural networks allow solving many ill-posed inverse problems with unprecedented performance. Physics informed approaches already progressively replace carefully hand-crafted reconstruction algorithms in real applications. However, these networks suffer from a major defect: when trained on a given forward operator, they do not generalize well to a different one. The aim of this paper is twofold. First, we show through various applications that training the network with a family of forward operators allows solving the adaptivity problem without compromising the reconstruction quality significantly. Second, we illustrate that this training procedure allows tackling challenging blind inverse problems. Our experiments include partial Fourier sampling problems arising in magnetic resonance imaging (MRI) with sensitivity estimation and off-resonance effects, computerized tomography (CT) with a tilted geometry and image deblurring with Fresnel diffraction kernels.

Index terms— Blind inverse problems, self-calibration, adaptivity, model-based reconstruction, convolutional neural network, unrolled networks, MRI reconstruction, computerized tomography, blind deblurring

1 Introduction

The primary contribution of this paper is the design of model-based neural networks to solve *families* of blind inverse problems. Many sensing devices like cameras, Magnetic Resonance Imaging (MRI) or Computerized Tomography (CT) systems measure a signal $x \in \mathbb{K}^N$ through a linear operator $A(\theta) \in \mathbb{K}^{M\times N}$ with $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. The parameter $\theta \in \mathbb{R}^P$ characterizes the sensing operator. For instance, it can encode the point spread function in image deblurring, the projection angles in CT or the Fourier sampling locations and coil sensitivities in MRI. This leads to measurements of the form:

$$y = \mathcal{P}(A(\theta)x), \tag{1}$$

where $\mathcal{P} : \mathbb{K}^M \to \mathbb{K}^M$ is a perturbation (e.g. additive Gaussian noise, quantization). A model-based inverse problem solver constructs an estimate $\hat{x}$ of x from y and $A(\theta)$. If the parameter θ is unknown, then we speak of a blind inverse problem. In this paper, we focus on neural network based reconstructions. We consider mappings of the form:

$$\begin{array}{rccc} \mathcal{N}: & \mathbb{R}^D \times \mathbb{K}^{M\times N} \times \mathbb{K}^M & \to & \mathbb{R}^N \\ & (w, A, y) & \mapsto & \mathcal{N}(w, A, y). \end{array} \tag{2}$$

Given a weight $w \in \mathbb{R}^D$, a forward operator A and a measurement vector y, the network $\mathcal{N}$ outputs an estimate $\hat{x} = \mathcal{N}(w, A, y)$. The network depends on the operator A since it typically consists in alternating an inversion of A followed by a regularization with a neural network. For a fixed forward operator $A(\theta_0)$, the traditional procedure to train the network consists in optimizing the weights w by minimizing the risk:

$$\inf_{w\in\mathbb{R}^D} R(w) \quad \text{with} \quad R(w) \overset{\text{def}}{=} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\mathcal{N}(w, A(\theta_0), \mathbf{y}) - \mathbf{x}\|_2^2 \right]. \tag{3}$$

In this equation, $\mathbb{E}_{\mathbf{x},\mathbf{y}}$ indicates the expectation with respect to the random vector $(\mathbf{x}, \mathbf{y})$. The random vector $\mathbf{y}$ is generated using the forward model (1). Ideally $\mathbf{x}$ should be a random vector describing

the distribution of images to reconstruct. Unfortunately, it is usually unknown and approximated by a discrete probability measure of the form $\frac{1}{I}\sum_{i=1}^{I}\delta_{x_i}$, where $(x_i)_{1\leq i\leq I}$ is a collection of training images. We then speak of empirical risk minimization. In words, we wish the reconstruction mapping $\mathcal{N}(w,\cdot,\cdot)$ to output images close in average to the true underlying signals. In this paper, we explore a seemingly minor variation of this principle by solving:

$$\inf_{w\in\mathbb{R}^D} E(w) \quad \text{with} \quad E(w) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\theta}}\left[\|\mathcal{N}(w,A(\boldsymbol{\theta}),\mathbf{y})-\mathbf{x}\|_2^2\right], \tag{4}$$

where the expectation is also taken with respect to the parameter $\boldsymbol{\theta}$ considered as a random vector. That is, we train our reconstruction mapping on a *distribution of operators*. While this idea is quite natural and most likely implemented already in a few methods, we believe that this paper is the first to address a systematic study of its performance. The main motivation for this modification is twofold. First, we want to address a lack of adaptivity for the standard training procedure. Second, we want to use the resulting reconstruction mapping to solve blind inverse problems. Let us discuss these two points in more depth.

Training mismatch issue While model-based reconstruction networks provide state-of-the art results in a large panel of applications, it is now well established that they suffer from a *lack of adaptivity*. This means that a network trained for a specific operator $A(\theta_0)$ may have a significant performance drop if used for another operator $A(\theta_1)$. This drop can be evaluated as follows. Let $\theta_0\neq\theta_1$ denote two different operator parametrizations. Let $y_0=\mathcal{P}(A(\theta_0)x)$ and $y_1=\mathcal{P}(A(\theta_1)x)$. Assume that $w_0^\star$ and $w_1^\star$ are the weights of a reconstruction network optimized for $A(\theta_0)$ and $A(\theta_1)$ respectively. We compare the quality of $\mathcal{N}(w_0^\star,A(\theta_0),y_0)$ and $\mathcal{N}(w_1^\star,A(\theta_0),y_0)$ in the third and fourth rows of Fig. 1. Observe the significant performance difference.

To avoid this pitfall, we propose to train the network by minimizing (4) instead of (3). After providing some theoretical hints on why a favorable behavior may occur, we will carefully evaluate the performance of the resulting networks in Section 6 for MR image reconstruction from under-sampled data, CT imaging and image deblurring. We conclude that this learning approach yields a reconstruction network which is significantly more stable to variations of the forward operator. In addition, the performance of an unrolled network trained on a restricted family is only marginally worse than that of a network that would be trained and used for a single operator. It therefore provides a satisfactory answer to the adaptivity issue. We also address several questions raised by our methodology. Can the unrolled network trained on a family extrapolate to unseen operators? How to sample the space of admissible operators? What is the gain of this approach in comparison to more "universal approaches" such as plug-and-play (P&P) priors?

Model mismatch issue Assume that we observe $y_1=\mathcal{P}(A(\theta_1)x)$. Unfortunately, we only have access to an approximate knowledge $A(\theta_0)$ of the true forward model $A(\theta_1)$. This can be due to an imprecise calibration of the sensing device or to the motion of a patient in a scanner for instance. We then face a blind inverse problem. A problem solved with such a model mismatch (i.e. with the operator $A(\theta_0)$ in place of $A(\theta_1)$), can lead to catastrophic reconstruction results, as illustrated in the last row of Fig. 1.

The second contribution of this work is to propose a systematic approach called "deep unrolled prior" to recover an estimate $\hat{\theta}_1$ of θ_1 from the observation y_1. We show that unrolled networks trained on a family of forward models provide a powerful tool to solve several blind inverse problems. The idea is simply to minimize the data consistency error

$$\hat{\theta}\in\arg\min_{\theta\in\Theta}\frac{1}{2}\|A(\theta)\mathcal{N}\left(w,A(\theta),y\right)-y\|_2^2. \tag{5}$$

The reconstructed image $\hat{x}\stackrel{\text{def}}{=}\mathcal{N}\left(w,A(\hat{\theta}),y\right)$ is defined as the output of the unrolled neural network. This consistency principle is spread massively in the literature of blind inverse problems and usually appears when constructing maximum a posteriori estimates. The main contribution here is to plug it with a specific training procedure on a family of forward operators.

2 Related works

Regularization theory From a historical perspective, the first inverse problem solvers were based on simple inverses or approximate inverses of $A(\theta)$. This approach provides low quality results when the

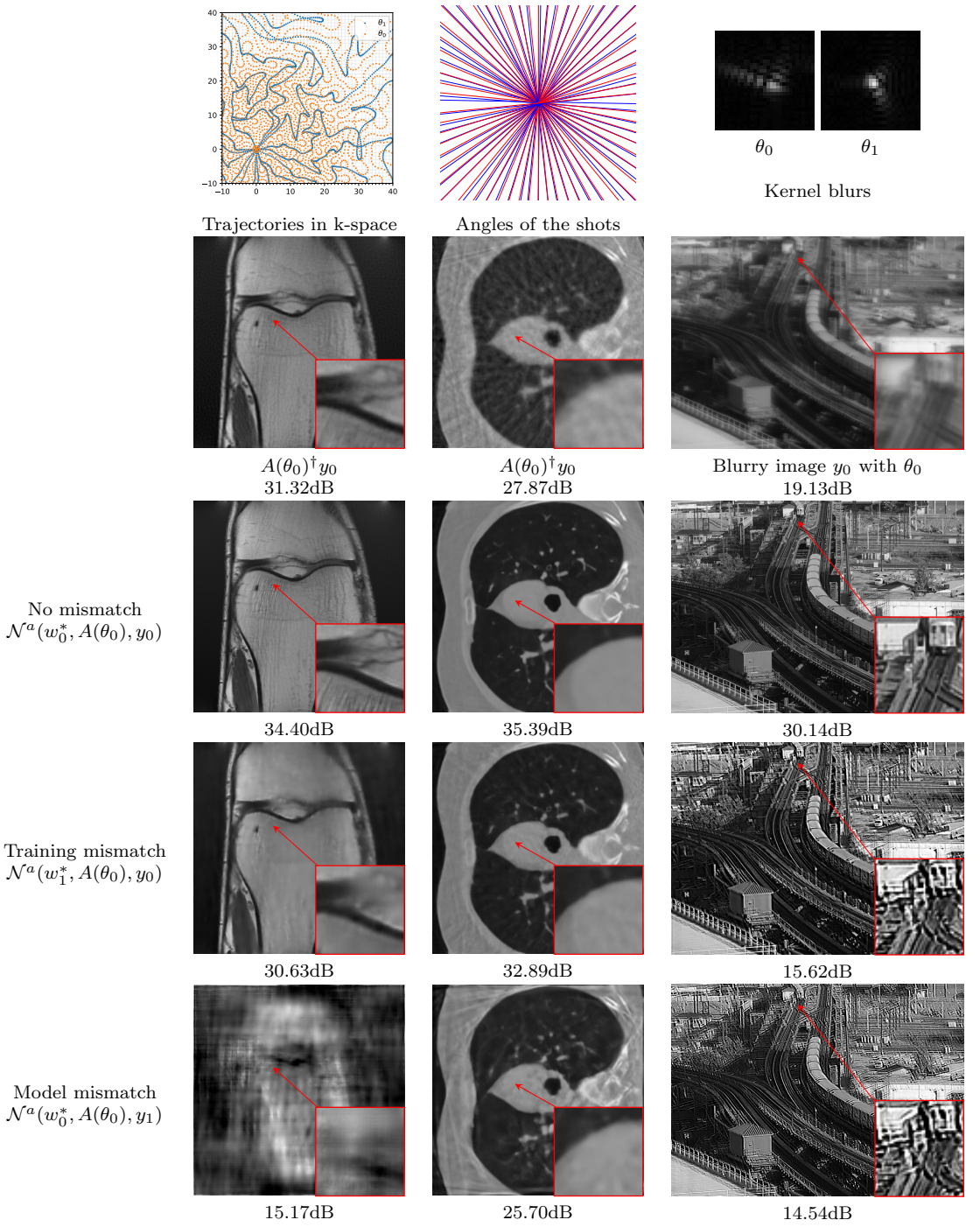


Figure 1: Examples of the issues addressed in this paper. *1st row:* description of the forward operators parameterized by θ_0 and θ_1. *2nd row:* pseudo-inverse reconstruction of $y_0 = \mathcal{P}(A(\theta_0)x)$ for MRI and CT and the blurry image y_0 for deblurring. *3rd row:* reconstruction with no model or training mismatch. *4th row:* reconstruction with a training mismatch. *Last row:* reconstruction with a model mismatch (blind). All the models are an unrolled ADMM trained on $A(\theta_0)$. The reconstruction PSNR is provided below each image.

matrix $A(\theta)$ has a non trivial kernel or when the conditioning number of $A^*(\theta)$ is high. In those cases, it is critical to use regularization terms. For long ($\sim$ 1960-2000), simple quadratic terms (Tikhonov) dominated the scientific landscape. Around 1990, a second research trend appeared with convex, nonlinear regularizers such as total variation [77]. This area culminated with the development of the compressed

sensing theory [18, 63].

Learned reconstruction In the 2010's learned regularizers such as variational networks emerged [44, 46, 39, 5]. They can be convex or nonconvex and can come with nice theoretical guarantees (e.g. robustness and stability) developed in the frame of compressed sensing. They apply seamlessly to a large variety of inverse problems.

Starting from 2015, impressive performance gains have occurred with the advent of neural networks. They seem able to replace the initial methods in a growing number of technologies [89]. There are two main approaches to attack reconstruction problems using machine learning [8]. A first solution is *end-to-end networks* where the neural network is agnostic to the operator $A(\theta)$. It gets trained through pairs (y_i, x_i) generated with the model (1). A popular example is AUTOMAP [99]. In this algorithm, the network needs to infer the forward model from the training data. This usually requires a huge amount of training data for large M and N.

The other possibility is *model-based* reconstruction networks that are defined as mappings of the form (2). They are often praised for the fact that they require less training data and benefit from a higher interpretability. Two popular approaches among this class are:

- *Denoising nets*: The reconstruction network performs a rough inversion followed by a denoising network such as a U-Net, to remove the remaining artifacts, see e.g. [47].

- *Unrolled nets*: Many efficient iterative methods have been developed to solve convex optimization problems (proximal gradient descent, Douglas-Rachford, ADMM, Primal-Dual, ...) [24]. They have the general form:

$$x_{k+1} = \text{prox}_R \left(M(A(\theta), y, x_k) \right), \tag{6}$$

  for $k = 1$ to $K \in \mathbb{N}$. The mapping M can be interpreted as a crude way to invert the operator, in the sense that $A(\theta)M(A(\theta), y, x_k) \simeq y$. The term prox_R can be interpreted as a way to regularize (denoise) the remaining artifacts. The so-called P&P priors [88] fit in this category.

  The unrolled networks draw their inspiration from (6). They consist in replacing the handcrafted or learned proximal operator prox_R by a sequence of neural networks $(\mathcal{D}_k(w_k, \cdot))_{1 \leq k \leq K}$ promoting an output x_K similar to the training images. The difference with the P&P priors is that the weights w_k are trained specifically for a given operator $A(\theta)$. Examples of approaches in this category include [84, 26, 1, 96, 28, 2, 3, 45, 59]. These algorithms are currently among the most efficient for MRI reconstruction [66].

For completeness, let us mention that a popular alternative consists in synthesizing the images x with generative models [12, 9]. Compared to the approaches mentioned above, it typically suffers from a higher computational cost. Indeed, a gradient descent in the latent space needs to be performed. In addition, specific care must be taken to handle images living outside the range of the generator. Hence, we will not consider this approach further in this work.

Adaptivity Neural network reconstructions can suffer from severe instabilities. This issue was notably discussed in [6]. The authors show that well chosen additive noise (an adversarial attack) or modifications of the forward operator can lead to disastrous hallucinations for some specific architectures. This problem was also studied experimentally in [32]. There, the authors have shown that careful training procedures could fix many issues and yield robust and state-of-the-art reconstruction results, with a stability on par with handcrafted methods. Yet, it should be noted [38] that there is a fundamental bottleneck in the resolution of severely ill-posed inverse problems. In fact, any attempt to solve some of their instances stably and accurately is doomed, since multiple plausible signals may live simultaneously in the kernel of the forward operator.

A paper closely related to our work is [33]. The authors study the same robustness issue to model mismatches. They propose two distinct algorithmic approaches to address it. The first one is called parametrize & perturb by the authors. It suffers from an important drawback, which is the need to re-optimize the network weights for every new operator. It can therefore be slow at run time and we do not compare it in this paper. The other approach is called Reuse & Regularize ($R\&R$). It consists in training a network for a given operator $A(\theta_0)$, and then use this network as a regularizer for another operator $A(\theta_1)$. This is done in an iterative procedure, accounting for the data consistency term $\|A(\theta_1)x - y\|_2^2$. The approach we propose in this paper is significantly lighter at run time: we just train the network once with a family of operators and use it for multiple operators.

An older and popular alternative consists in replacing the proximal operator in (6) by a denoiser. This approach is often called a plug-and-play (P&P) prior [88]. It was first used with hand-crafted priors [43] and a significant performance boost occurred with the use of pre-trained neural networks among which we can cite [78, 97]. In addition, let us mention that [97] trains the denoiser with various noise levels (instead of forward operators). This makes it possible to fine-tune the regularization in the P&P method. This approach has the huge asset of adapting painlessly to arbitrary inverse problems. We propose some comparisons and discuss the pros and cons of each approach in Section 6.

Finally, let us point out that the idea of training solvers on families of operators is probably implemented already on a variety of neural networks. For instance, the SFTMD network in [42] is trained on a family as well. Our main contribution here is a systematic empirical study of this methodology.

Blind inverse problems Blind inverse problems are spread massively in applications. The review papers [52, 17] provide a good idea of the wealth of results for the sole field of blind deconvolution and super-resolution.

A possibility is to design a two-step method. First an estimate of the forward operator is built. Second, this estimate is used in conjunction with the methods from the previous section. In some cases, it is possible to exploit some redundancy in the data to estimate the operator parameters. This is the case in parallel MRI, where the coil sensitivity maps can be estimated using only the low frequencies [81, 73, 41]. When no redundancy is available, estimating the operator can be achieved by minimizing the discrepancy between the statistics of the acquired measurement and the statistics of the measurements generated by applying an operator to a "natural" signal. A good example in blind deblurring is the Goldstein-Fattal approach [34], which analyzes the power spectrum of the blurry image. A few authors proposed to build an identification network that learns to identify the blur kernel [79] or a blur parametrization [83, 94, 19, 25] from the blurry-noisy image. While this approach is cheap computationally, it requires an application specific design.

One of the most popular alternatives consists in minimizing a combination of a data fidelity term and a regularizing prior. This can be addressed through an alternate minimization between the image and the operator parametrization. Most of the literature suggests the use of hand-crafted priors on the unknown operator or on the image to recover (see e.g. [20, 30, 50, 51, 93, 4, 69, 65, 70, 74, 10, 62, 22, 98] for blind deblurring, or [75, 91, 64] in CT imaging).

While these approaches can provide excellent results, they are likely to be outperformed by neural network based approaches in a near future. Indeed, impressive performance has already been reached recently thanks to neural network based regularizers. Different strategies have been suggested, going from untrained networks (see [14] for an application in optics), generative models (see [9] for an application in blind deblurring), or unrolled networks (see [56, 57] for an application to super-resolution from an image sequence).

The method advocated in our paper is close in spirit to the works in this latest category. It differs in the way the training is performed. In our work, we first train an unrolled network on a family of forward operators, which allows fixing the weights once for all. We then minimize (5) in the space of parameters of the forward model. This methodology has various advantages:

- Compared to untrained networks [14], the method does not optimize the network weights to solve the problem, which is typically quite computationally heavy. It is therefore faster at evaluation time. In addition, it is adapted to a clearly defined image dataset.
- Methods based on generative models [12, 9] may suffer from a significant drawback: the produced images necessary live in the range of the generator. To avoid this issue, a possibility is to add hand-crafted regularization terms such as total variation that allow extending the span of possible images [9].
- In [56, 57], the neural network weights are trained directly to solve the blind inverse problem. This significantly limit the number of weights and iterations within the iterative procedure. In this paper, we propose to train the network beforehand, allowing to use arbitrary solvers and as many iterations as desired to find the parameter θ.

3 Preliminaries

In this paper, we consider forward models of the form

$$y = A(\theta)x + b \tag{7}$$

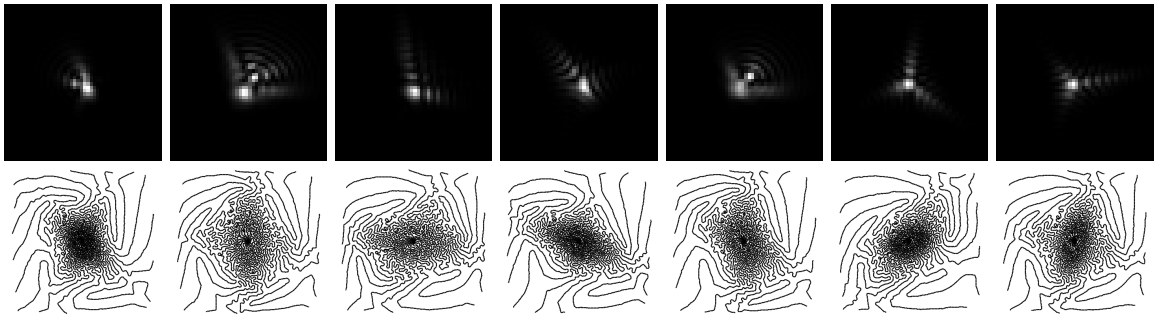


Figure 2: Top: examples of point spread functions generated with Fresnel diffraction theory. The pupil function is defined through a linear combination of 7 Zernike functions with random coefficients. The dependency on the coefficients is highly nonlinear. All PSFs are realistic (e.g. non-negative and bandlimited). Bottom: examples of sampling schemes in parallel MRI used in this work. All sampling schemes are realistic and can be implemented on an actual scanner. They include realistic physical constraints of speed and acceleration (maximum gradient amplitude and slew rate).

where $A(\theta) \in \mathbb{K}^{M \times N}$ is a linear mapping either real ($\mathbb{K} = \mathbb{R}$) or complex ($\mathbb{K} = \mathbb{C}$). In all our experiments, we define b as additive white Gaussian noise (complex for MRI) $b \sim \mathcal{N}(0, \sigma^2 \mathrm{Id})$. The dependency of A with respect to its parameter θ can be linear or nonlinear. We let $N \in \mathbb{N}$ denote the number of pixels of the image x with $N = N_x \times N_y$ for 2D images and M is the number of measurements.

3.1 Forward models

To illustrate our problem, we consider three important biomedical applications: parallel magnetic resonance imaging, computerized tomography and microscopy/astronomy. We provide a quick overview of the models below and a more precise mathematical description is given in Appendix A.

Parallel Magnetic Resonance Imaging In this application, $A(\theta)$ is the product of a partial non uniform Fourier transform with a set of diagonal matrices encoding the "sensitivities" of reception coils around the object to image. The samples in the Fourier domain, also denoted k-space, are located along a smooth trajectory. The reconstruction network should adapt to:

- different sampling trajectories,
- different sensitivity maps (which are smoothly varying multipliers),
- the effect of imperfect gradient coils/ off-resonance effects that deteriorate the trajectory.

In our experiments, we consider a subsampling ratio of 4 and 10, meaning that $M = N/4$ or $M = N/10$ respectively. The parameter θ encodes all the parameters above. The sensitivity maps and the trajectory perturbations are usually unknown, making MRI reconstruction a blind inverse problem. Examples of realistic sampling trajectories used in this work are displayed in Fig. 2, bottom.

Computerized tomography We consider parallel beam X-ray computerized tomography. It consists in probing line integrals of an object along a set of parallel lines that may be rotated and shifted. In this application the parameter θ represents the angles and shift at origin of the lines. The problem becomes blind if the object to image moves during the scan.

Deblurring in optics The most common way to parametrize the Point Spread Function (PSF) of an optical system in optics is by using Fresnel diffraction theory [13]. In this theory, the PSF is entirely determined by the pupil function, which is a complex function defined over the objective aperture. For a circular aperture, the pupil function can be expanded with Zernike polynomials, which are orthogonal polynomials over the disk [68, 53]. The parameter θ coincides with the coefficients of this linear decomposition. The problem is blind whenever the PSF is unknown. In our experiments, we consider a linear combination of 7 Zernike polynomials. Examples of random PSFs generated through this model are displayed in Fig. 2, top.

3.2 Model-based reconstruction networks

In this paragraph, we detail the neural network architectures considered in this work for the numerical experiments. In all the following, $\mathcal{D} : \mathbb{R}^D \times \mathbb{K}^N \to \mathbb{K}^N$ denotes a neural network with weights $w \in \mathbb{R}^D$ and input signal $x \in \mathbb{R}^N$. The letter $\mathcal{D}$ stands for denoising, since the goal of this network is to remove artifacts on x remaining after inversion of the forward model.

3.2.1 Inversion + denoising network

Possibly the simplest way to construct an operator-aware reconstruction network is to consider a mapping $\mathcal{N}^d$ of the form:

$$\mathcal{N}^d(w, A(\theta), y) \stackrel{\text{def}}{=} \mathcal{D}\left(w, A(\theta)^\dagger y\right), \tag{8}$$

where $A(\theta)^\dagger$ is the pseudo-inverse of $A(\theta)$. The idea is simply to roughly invert the model and train a single denoising network $\mathcal{D}$ to remove the artifacts [47]. This type of network was one of the earliest ones. We will consider this architecture only for a single MRI experiment due to its overall poor performance.

3.2.2 Unrolled ADMM

The unrolled ADMM network is an efficient architecture providing results close to the state-of-the-art in a number of applications. It takes the form (see e.g. [67]):

$$\begin{aligned}
x_0 &= 0 \quad \text{and} \quad \mu_0 = 0 \\
z_{k+1} &= \left[A^*A + \beta_k \mathrm{Id}\right]^{-1} \left(A^*y + \beta_k x_k - \mu_k\right) \\
x_{k+1} &= \mathcal{D}\left(w_k, z_{k+1} + \frac{\mu_k}{\beta_k}\right) \\
\mu_{k+1} &= \mu_k + \beta_k \left(z_{k+1} - x_{k+1}\right).
\end{aligned}$$

This sequence runs for K iterations and the result is denoted $\mathcal{N}^a : (w, A, y) \mapsto x_K$. The number K will be set equal to 5 when the weights w need to be trained. This is mostly due to memory and computing time limitations. This ADMM based architecture can also be used as a P&P algorithm, in which case, a higher number of iterations can be considered to obtain the best possible signal-to-noise ratio. The parameter β_k is a penalty parameter, which can vary from one iteration to the next. We use the update rule proposed by [97] for the P&P algorithms in all experiments. The weights w to be trained are $w = [w_0, \ldots, w_{K-1}]$. They differ at each iteration for the unrolled networks and are identical in the P&P networks.

3.2.3 Denoising network architecture

All our experiments are achieved with a *fixed denoising architecture* $\mathcal{D}$. We choose the so-called DRUNet network [97] (for Denoising Residual U-Network). This network is the current state-of-the-art when used within P&P algorithms. One of its important assets is its ability to accommodate for different noise levels. The idea is to set one of the input channels as a constant image with a value equal to the standard deviation of the noise. This is an important feature for P&P algorithms, which depend on a parameter describing the noise level. The same advantage applies to unrolled networks. The noise level is a user-defined parameter that can be changed to vary the regularization level depending on the application. We decided to use a single denoising architecture in our experiments for the following reasons:

- The network is currently the state-of-the-art for the field of P&P methods. It therefore makes the comparison with the P&P methods more relevant.
- Compared to others architectures we have tried, the training stage was easier and the performance higher.
- We want to simplify the message by avoiding too many experiments and by comparing the different methods using only a single architecture.
- One single training of each network is already about a week of computation on an A100 Nvidia graphics card and we want to reduce the overall computing time for this paper.

4 Training with an operator distribution

In most existing approaches, networks are trained by minimizing the empirical risk for a given forward model $A(\theta_0)$ as in (3). Instead, we propose to minimize the risk over an operator distribution as in (4). In this section, we explain a few differences between both approaches.

To begin with, notice that the pre-image by $A(\theta)$ of the measurement vector $y = A(\theta)x + b$ is given by

$$A(\theta)^{-1}(y) = \left\{x + k + w, k \in \ker(A(\theta)), w = A(\theta)^\dagger b \in \ker(A(\theta))^\perp\right\}. \tag{9}$$

The vector w is a correlated noise with a correlation that depends on $A(\theta)$. Hence, the reconstruction network $\mathcal{N}$ should serve two purposes:

1. Recover the missing data k in the kernel of $A(\theta)$.
2. Remove the correlated noise w.

Each of these two tasks is clearly highly dependent on $A(\theta)$. We explain below that training a network on a single operator may result in some overfitting for the operator $A(\theta_0)$ and to a lack of generalization to other operators. This problem is strongly mitigated by training the network on a family.

4.1 Large operator families are better in an ideal world

We take a Bayesian viewpoint and assume that $\mathbf{x}$ is a random vector in $\mathbb{K}^N$ with probability distribution measure μ_x. We also see $\mathbf{A}$ as a random operator in $\mathbb{K}^{M\times N}$ with distribution μ_A. Finally, we consider the random measurement vector $\mathbf{y}$ generated through the forward model 1. Alternatively, we can see the parameter $\boldsymbol{\theta} \in \mathbb{R}^P$ as a random vector with probability distribution measure μ_θ and we construct the random operator $A(\boldsymbol{\theta})$, i.e. pushforward $\boldsymbol{\theta}$ through the mapping $A(\cdot)$. For instance, the traditional training procedure (3) for a given operator $A(\theta_0)$ consists in assuming that $\mu_A = \delta_{A(\theta_0)}$ or equivalently $\mu_\theta = \delta_{\theta_0}$.

MMSE estimator In this framework, we may want to construct the Minimum Mean Square Error (MMSE) estimator. The Mean Square Error (MSE) of an estimator $\hat{x} : \mathbb{K}^{M\times N} \times \mathbb{K}^M$ is a measure of performance defined by:

$$\mathrm{MSE}(\hat{x}) \overset{\text{def}}{=} \mathbb{E}_{\mathbf{x},\mathbf{A},\mathbf{y}}\left[\|\hat{x}(\mathbf{A},\mathbf{y}) - \mathbf{x}\|_2^2\right]. \tag{10}$$

The conditional MSE is defined by:

$$\mathrm{MSE}(\hat{x}|A,y) \overset{\text{def}}{=} \mathbb{E}\left[\|\hat{x}(\mathbf{A},\mathbf{y}) - \mathbf{x}\|_2^2|\mathbf{A} = A, \mathbf{y} = y\right]. \tag{11}$$

Any estimator that achieves the minimum MSE is called an MMSE estimator. It is defined for (almost) all pairs $(A, y) \in \mathbb{K}^{M\times N} \times \mathbb{K}^M$ by:

$$\hat{x}_{\mathrm{MMSE}}(A,y) \overset{\text{def}}{=} \operatorname*{arg\,min}_{\hat{x}\in\mathbb{K}^N} \mathrm{MSE}(\hat{x}|A,y).$$

An important property of this estimator is that it can be expressed as the following conditional expectation [48]:

$$\hat{x}_{\mathrm{MMSE}}(A,y) = \mathbb{E}\left[\mathbf{x}|\mathbf{A} = A, \mathbf{y} = y\right].$$

By construction, this estimator is the best we can hope for, in average for a given distribution of triplet $(\mathbf{x}, \mathbf{A}, \mathbf{y})$.

Perfectly trained neural networks are MMSE estimators Notice that the risk E defined in (4) coincides with the MSE:

$$E(w) = \mathrm{MSE}(\mathcal{N}(w,\cdot,\cdot)). \tag{12}$$

Therefore, training a neural network amounts to finding the MMSE estimator among the family of estimators

$$\mathcal{F} \overset{\text{def}}{=} \left\{\mathcal{N}(w,\cdot,\cdot), w \in \mathbb{R}^D\right\}.$$

To better understand the difference between training a network on a single operator or on a distribution we may make the following simplifying assumption.

Assumption 1 (Zero approximation and optimization errors)**.** *We assume that:*

- *the family $\mathcal{F}$ contains the MMSE estimator $\hat{x}_{\text{MMSE}}$ for the distributions μ_{θ_0} and $\mu_{\mathcal{A}}$.*
- *the optimizer returns a global minimizer of the risk in* (4)*.*

In the language of [15], this means that the approximation and optimization errors vanish. Obviously, this is not realistic in general, but many recent experiments show that it can be considered approximately correct for large overparameterized networks (see e.g. [11]). Under those assumptions, the following straightforward result shows that it can only be beneficial to train a network on a distribution of operators with a large support.

Proposition 1. *Let w_0 denote the weights optimized using a single operator $A(\theta_0)$. Let w_{μ_A} denote the weights optimized using the distribution of operators μ_A. Let $\mathcal{A} \overset{\text{def}}{=} \text{supp}(\mu_A)$ denote the family of operators that was used for training. Under Assumption 1, we get:*

$$\mathcal{N}(w_{\mu_A}, A, y) = \hat{x}_{\text{MMSE}}(A, y) \quad \textit{for almost all} \quad A \in \mathcal{A}, y \in \mathbb{K}^M.$$
$$\mathcal{N}(w_0, A(\theta_0), y) = \hat{x}_{\text{MMSE}}(A(\theta_0), y) \quad \textit{for almost all} \quad y \in \mathbb{K}^M.$$

However, $\mathcal{N}(w_0, A, y)$ may differ from $\hat{x}_{\text{MMSE}}(A, y)$ whenever $A \neq A(\theta_0)$.

The above proposition is straightforward. It is a simple consequence of the fact that the weights are optimized to minimize the MSE. It tells us that the neural network $\mathcal{N}(w_{\mathcal{A}}, \cdot, \cdot)$ trained on an operator distribution coincides with the MMSE for almost every operator on the support. Hence, under Assumption 1, it is as good as can be for every operator seen during training. In particular it implies that $\mathcal{N}(w_{\mathcal{A}}, A(\theta_0), y) = \mathcal{N}(w_0, A(\theta_0), y)$ if $A(\theta_0) \in \mathcal{A}$. This means that there is no disadvantage to train the network on a family, even for the specific operator $A(\theta_0)$. This phenomenon will be (nearly) confirmed later in the numerical experiments.

On the other hand, nothing can be said for a network trained on a single operator $A(\theta_0)$ when applied to another operator $A \neq A(\theta_0)$. There, we need to rely on the generalization capacity of the network. This capacity looks really arbitrary since a single operator was seen during training. This is the most likely explanation for the lack of adaptivity that was observed in Fig. 1. To sum up, *under the idealist hypothesis 1, it can only be beneficial to train the network on the largest possible family of operators.*

Finally, let us mention that under Assumption 1, we could learn the prior μ_x exactly. This would make it possible to sample the posterior distribution $\mu_{x|(A,y)}$ directly [54, 82, 23]. Hence, the MMSE estimator could be accessed for *every* operator $A \in \mathbb{K}^{M \times N}$ (using, e.g. Langevin dynamics). However, this would come at the price of a significantly increased computational time to solve each inverse problem instance.

4.2 The possible downsides

In the previous section, we made the following unrealistic assumptions:

i) The family $\mathcal{F}$ is so large that it contains the MMSE estimator. In practice we use structured convolutional neural network which cannot approximate arbitrary functions and should account for approximation errors.

ii) The expectation with respect to μ_x can be evaluated. In most applications, we can only resort to a finite size dataset and to the minimization of the empirical risk.

iii) The optimization routine returns the global minimizer. In most cases, the stochastic gradient descents used for training only return approximate critical points.

Each of the above points makes the above analysis imprecise. In particular, if we only assume points ii) and iii) to hold, we get:

$$\text{MSE}(\mathcal{N}(w_0, \mathbf{A}, \mathbf{y})|A_0, y) = \inf_{w \in \mathbb{R}^D} \text{MSE}(\mathcal{N}(w, \mathbf{A}, \mathbf{y})|A_0, y) \leq \text{MSE}(\mathcal{N}(w_{\mathcal{A}}, \mathbf{A}, \mathbf{y})|A_0, y).$$

In general, the inequality above is strict since it is much easier to approximate the mapping $\hat{x}_{\text{MMSE}}$ pointwise on the domain $\{(A_0, y), y \in \mathbb{K}^M\}$ than on the whole domain $\{(A, y), A \in \mathcal{A}, y \in \mathbb{K}^M\}$. Hence, the lack of expressiveness of the network $\mathcal{N}$ will – in general – result in a performance decrease for the specific operator A_0. We will see empirically that this decay is moderate for the 3 applications considered

in this paper. Providing bounds on this decay is an intricate issue that is left open for future works. This simple observation however reveals a potential pitfall of the distribution training approach: the larger the family $\mathcal{A}$, the worst the performance for specific operators in $\mathcal{A}$. This shows that there is an adaptivity/performance trade-off, especially if the family $\mathcal{F}$ lacks expressiveness.

4.3 Choosing distributions of operators

Choosing a proper family and distribution of operators obviously depends on each application. Ideally, this distribution should reflect the real distribution of the underlying imaging system. Unfortunately, this is often hard to characterize. As mentioned in the previous section, the main feature to consider is the family of operators $\mathcal{A}$ seen during training, i.e. the support of the distribution μ_A. This family should be sufficiently large to reflect any operator that could arise in practice. Once this family is characterized, it is possible to sample it as uniformly as possible. Overall, the choice of an operator distribution is nontrivial and should rely on an expert knowledge of the imaging system. We detail how we addressed this question for the three applications below.

4.3.1 Magnetic Resonance Imaging

In this modality, the family of forward operators is constructed by considering different sampling schemes and sensitivity maps.

Sampling schemes We propose to generate random sampling schemes $\boldsymbol{\xi}$ following the ideas from [16, 21, 55]. The principle it to design a scheme that fits a target probability measure $\rho : \mathbb{R}^2 \to \mathbb{R}_+$. To this end, we define

$$\boldsymbol{\xi}(\rho) \stackrel{\text{def}}{=} \operatorname*{arg\,min}_{\boldsymbol{\xi} \in \Xi} \operatorname{dist}\left(\frac{1}{M} \sum_{m=1}^{M} \delta_{\boldsymbol{\xi}_m}, \rho \right), \tag{13}$$

where dist is a discrepancy between probability measures and $\Xi \subseteq \mathbb{R}^{2\times M}$ is a set that describes the admissible trajectories from the scanner.

Following [36], we generate random target densities ρ as anisotropic power decaying distributions. They are parameterized by a random vector $\boldsymbol{\lambda}$ that encodes the density at origin, the anisotropy and the power decay law. To avoid solving (13) at training time, we have pre-computed 1000 sampling patterns. The corresponding vectors $\boldsymbol{\lambda}$ have been generated by using a max-min sampling (see [72, 25]) of a set of an admissible set of parameters Λ. We refer to [36] for more details. Examples of densities and sampling patterns $\boldsymbol{\xi}(\rho(\boldsymbol{\lambda}))$ without constraints are displayed in Fig. 3a-3e. Notice that we did not include trajectory constraints for generating this figure. They are taken into account for the blind inverse problem part.

Sensitivity maps As for the sensitivity maps, we used real estimates generated using the fastMRI database [95]. We first estimate them using a standard approach [41] and then project the estimates onto the span of a parametrization composed of thin plate splines. At training time, they are associated to the corresponding training pairs.

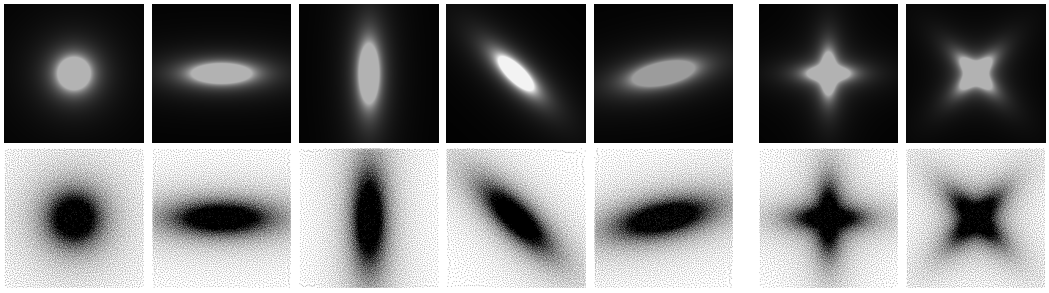
Trajectories filtering We did not include the trajectory perturbation effect (convolution with $\mathbf{h}(\boldsymbol{\omega})$) at training time.

4.3.2 Computerized tomography

In this modality, we assume that the distribution of projection angles follows a uniform distribution centered on a vector of regularly spaced angles $\alpha_0 = (-\pi/2, -\pi/2 + \pi/J, \ldots, \pi/2)$ (see the red lines in first row of Fig. 1) and shift at origin $s_0 = 0$. Hence, we have $\alpha = \alpha_0 + \boldsymbol{\alpha}_\delta$ with $\boldsymbol{\alpha}_\delta \sim \mathcal{U}\left([-1.37^\circ, 1.37^\circ]^J\right)$ and the random shifts are $\mathbf{s} \sim \mathcal{U}\left([-2, 2]^J\right)$. These perturbations may reflect movements of the patient inside the scanner during the scan.

4.3.3 Deblurring

In this application, we vary the blur kernel (the PSF) by changing only the 4-th to the 10-th Zernike polynomials. We set $\theta_1 = \theta_2 = \theta_3 = 0$ in (21). In the Noll nomenclature, θ_1 coincides with the piston, which does not change the PSF, θ_1 and θ_2 are tilts, which just shift the PSF. We want to discard those



(a) ◯ (b) − (c) | (d) (e) (f) + (g) ×

Figure 3: Densities (top) and corresponding sampling schemes (bottom). Fig. 3a, 3b, 3c, 3d and 3e belong to the family $\mathcal{A}$. Fig. 3f and 3g (crosses) do not. Notice that the sampling patterns are diverse with significant differences from one to the other.

coefficients to avoid the usual translation ambiguity in blind deconvolution. We let the coefficients θ_4 to θ_{10} follow a uniform distribution in $[-0.15, 0.15]$.

5 Deep unrolled prior

Assume that y is generated according to the forward model $y = \mathcal{P}(A(\bar{\theta})\bar{x})$, *with an unknown parameter* $\bar{\theta}$. In that case, we need to estimate both $\bar{x}$ and $\bar{\theta}$, or alternatively the operator $A(\bar{\theta})$.

5.1 The proposed principle

After training a network $\mathcal{N}$ on a family $\mathcal{A}$, we get a weight vector $w_{\mathcal{A}}$. For any pair $(A, y) \in \mathbb{K}^{M\times N} \times \mathbb{K}^M$, we are therefore able to build an estimate $\hat{x}(A, y) = \mathcal{N}(w_{\mathcal{A}}, A, y)$ of $\bar{x}$. We propose to estimate $\bar{\theta}$ by solving the optimization problem (5):

$$\hat{\theta} \in \arg\min_{\theta\in\Theta} \frac{1}{2}\|A(\theta)\mathcal{N}(w, A(\theta), y) - y\|_2^2. \tag{14}$$

In this formulation, we wish the measurements y to be consistent with the recovered signal, i.e. $A(\theta)\hat{x}(A(\theta), y) \approx y$. This approach could be called *deep unrolled prior*, since we use an unrolled network as a prior to solve a blind inverse problem. Contrarily to the popular unsupervised method called *deep image prior* [86] though, our unrolled network is trained in a supervised way. It is then used without supervision to find the parameter θ only. This approach can be related to a Bayesian approach.

Relationship with a MAP approach One of the most popular estimators for blind inverse problems is the Maximum A Posteriori (MAP). To make a link with this approach, let us assume that the forward model reads:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}, \tag{15}$$

where $\mathbf{b} \sim \mathcal{N}(0, \sigma^2\mathrm{Id})$. The random operator $\mathbf{A}$ is drawn according to a distribution $\mu_A \propto \exp(-R_A)$, where $R_A : \mathbb{K}^{M\times N} \to \mathbb{R} \cup \{+\infty\}$ is a regularizer on the operator domain. Similarly, the random image $\mathbf{x}$ is drawn according to a distribution $\mu_x \propto \exp(-R_x)$, where $R_x : \mathbb{K}^N \to \mathbb{R} \cup \{+\infty\}$ is a regularizer on the image domain. We also assume independence of $\mathbf{A}$, $\mathbf{x}$ and $\mathbf{b}$.

In that case, it is tempting to solve the MAP problem:

$$\begin{aligned}
&\arg\max_{\substack{A\in\mathbb{K}^{M\times N}\\ x\in\mathbb{K}^N}} \mathbb{P}(\mathbf{A} = A, \mathbf{x} = x|\mathbf{y} = y) \\
&= \arg\max_{\substack{A\in\mathbb{K}^{M\times N}\\ x\in\mathbb{K}^N}} \mathbb{P}(\mathbf{y} = y|\mathbf{A} = A, \mathbf{x} = x) \cdot \exp(-R_x(x)) \cdot \exp(-R_A(A)) \\
&= \arg\min_{\substack{A\in\mathbb{K}^{M\times N}\\ x\in\mathbb{K}^N}} \frac{1}{2\sigma^2}\|Ax - y\|_2^2 + R_A(A) + R_x(x) = J(A, x)
\end{aligned} \tag{16}$$

We used the Bayes rule to go from the first to the second line and applied the function $-\log$ to get to the third. This formulation often appears in the literature and is at the basis of the most successful handcrafted approaches, see e.g. the review paper [17].

Let $\hat{x}(A, y)$ denote the minimizer of (16) with A fixed. Injecting this into the cost function, we see that finding the optimal operator A in the problem above is equivalent to:

$$\underset{A \in \mathbb{K}^{M \times N}}{\arg\min} \frac{1}{2\sigma^2} \|A\hat{x}(A, y) - y\|_2^2 + R_A(A) + R_x(\hat{x}(A, y)).$$

Assuming that the distributions μ_x and μ_A are uniform over compact sets, the functions R_A and R_x are constant. Hence, the MAP approach finally simplifies to:

$$\min_{\theta \in \Theta} \frac{1}{2\sigma^2} \|A(\theta)\hat{x}(A(\theta), y) - y\|_2^2,$$

which coincides with the proposed approach.

A type of P&P Let us mention that the proposed approach can also be seen as a type of P&P prior. Instead of training a denoising network, as is the case in the initial P&P approach, we train an inverse problem solver. We then use it as a prior to infer an operator, instead of a signal. Making deeper links with this approach is out of the scope of this paper.

5.2 Numerical resolution

As the function in (5) is deterministic over a small, to moderate dimension (between 5 and 1000 for the considered applications), we can opt for many different 0-th or 1-st order optimization routines.

Applying 1-st order methods is highly non trivial without using automatic differentiation. Indeed, we need to compute the Jacobian of $\mathcal{N}(w^\star, A(\theta), y)$ with respect to the parameter θ. This in particular requires evaluating the derivative of $A(\theta)$ with respect to the parameters θ and of the neural network $\mathcal{N}(w^\star, A(\theta), y)$ with respect to its second variable. In all our experiments, we used the automatic differentiation techniques available in PyTorch. This required to implement the Jacobian of the mapping A with respect to θ. To actually solve the problem, we considered the following optimization routines:

- The L-BFGS optimizer [61]. This quasi-Newton method estimates the Hessian of the function using first order information only and is known to converge rapidly when initialized close to a (local) minimizer. It therefore seems particularly adapted when the user has a good knowledge of the true parameter $\bar{\theta}$. We used this approach for the MRI experiments.

- The RMSProp or ADAM optimizer [85, 49]. For the computerized tomography experiments, we observed issues with a convergence to bad local minimizers using L-BFGS. To avoid this phenomenon, a possibility is to resort to inertial methods, which are known to escape narrow basins of attraction. In our experiments, we used the RMSProp optimizer with a parameter $\beta = 0.9$ (Adam with $\beta_1 = 0$). This procedure turned out to provide satisfactory results consistently.

- Bayesian optimization [31]. In some cases, it can be helpul to resort to 0-th order methods. This is the case if the mapping $A(\theta)$ is not differentiable with respect to the parameter θ. This is also the case if the cost function is too chaotic in which case, the gradient of the objective function does not provide a meaningful information on the location of the global minimizer. We can then resort to Bayesian optimization techniques. They typically work reliably for moderate dimensions $1 - 20$. We used this approach for the deblurring experiments, since it only involves 7 parameters and that we wanted to secure finding a good approximation of the global minimizer.

6 Numerical experiments

The numerical experiments are divided in two sections. In the first section 6.2 we compare the benefits and drawbacks of training model-based networks on a family of operators.

In the second section 6.3, we illustrate that training model-based networks on a family of operators allows solving blind inverse problems. The experiments are carefully conducted on the three applications: MRI, CT and image deblurring with an unrolled ADMM [84].

6.1 Training setting

All the models were trained using the Adam optimizer in PyTorch with the default parameters except the learning rate which was tuned for each experiment. We observed that depending on the imaging modality, the default learning rate could lead to a divergent sequence. In those situations, we divided it by 10 until we observed empirical convergence. The basic idea is to obtain a sufficient decay rate at the first epoch, without diverging.

Denoising For the P&P experiments, we trained the DRUNet model of [97] for grayscale images using the ImageNet database. We trained it for 100 epochs. We used a number of channels equal to 32, 64, 128, 256 for the 4 different layers. This results in a lighter network than the initial one, where the number of channels was doubled for each layer: 64, 128, 256, 512. The reason for this choice is to get a lighter model compatible with the unrolled architectures.

Magnetic Resonance Imaging The training database is the fastMRI knee training dataset [95]. It contains $34,742$ images of size 320×320. All evaluations were performed on the validation set of the fastMRI knee database containing $7,135$ 2D slices. We used the efficient cuFINUFFT transform [80], which is the fastest available library in our experiments (see `https://github.com/albangossard/Bindings-NUFFT-pytorch` for comparisons).

For the experiments illustrating the advantages of training on a family of operators, we set $M = N/4$, i.e. a 4x downsampling rate. We used a single reception coil ($J = 1$) with a known sensitivity map $s = 1$. The denoising network $\mathcal{N}^d$ was trained on 30 epochs with a learning rate of 10^{-3} and an exponential step decay of 0.95 after each epoch. The unrolled network $\mathcal{N}^a$ uses $K = 10$ iterations and it was trained on 14 epochs with a learning rate of 10^{-4} and an exponential step decay of 0.95 after each epoch. Both training took about 24h on an Nvidia V100, resulting in a total energy consumption of $\sim$ 70kWh.

The blind reconstruction experiments are conducted with $M = N/10$ measurements and $J = 15$ reception coils. The networks are trained for 8 epochs with a learning rate of 10^{-4} and with an exponential step decay of 0.95 after each epoch.

These subsampling rates are used frequently in MRI experiments.

Computerized Tomography We trained the network using $K = 5$ iterations using the ImageNet database. We initially used the Lung Image Database Consortium [7] database, but realized that it contains many improper slices (high noise, streaking artifacts, little contents...). We evaluated the algorithm on a curated version called LoPoDaP [58], containing less artifacts. The test dataset contains 4096 images.

As the blind inverse problem (5) requires differentiating the operator $A(\theta)$ with respect to its parameters θ, we cannot use standard GPU-based libraries to compute the Radon transform [76]. We thus resorted to an homemade implementation that relies on a NUFT through the Fourier slice theorem. In order to reduce the important numerical cost and energy consumption of the experiments with CT reconstruction, we downsized the images to 256×256.

Deblurring The image deblurring experiments were carried out with the MS COCO dataset [60] ($118,287/5,000$ images for training/validation). During training we randomly cropped patches of size 400×400 to speed-up the computation.

6.2 Benefits of training on a family

6.2.1 Training on fixed operators

In this section, we highlight the limits of training a reconstruction network on a single operator, as is currently the dominant practice. Let us detail the training procedure for each application.

For MRI reconstruction, we considered measurements coming from a single reception coil, to reduce the computational complexity. We used both the denoising network $\mathcal{N}^d$ and the unrolled proximal gradient descent $\mathcal{N}^p$ on 5 different schemes: a radial one ($\bigcirc$, Fig. 3a), a horizontal one ($-$, Fig. 3b) and a vertical one ($|$, Fig. 3c). In addition, we used two crosses, which do not belong to the training family $\mathcal{A}$. The first one is aligned with the axes ($+$, Fig. 3f) and the other one with the diagonals ($\times$, Fig. 3g).

For CT reconstruction, we considered measurements coming from randomly perturbed versions $\theta_1, \theta_2, \theta_3$ of the equiangular pattern θ_0. The network is an unrolled ADMM $\mathcal{N}^a$ ran for 4 iterations.

The perturbations θ_1, θ_2 belong to the family $\mathcal{A}$ used for the family training. The perturbation θ_3 is twice larger than what was observed during the training phase and does not belong to $\mathcal{A}$.

For the deblurring problem, we considered random convolution kernels generated using the model (21). The network is also an unrolled ADMM $\mathcal{N}^a$ ran for $K = 5$ iterations. The perturbations $\theta_0, \theta_1, \theta_2$ belong to the family $\mathcal{A}$ used for the family training, while θ_3, θ_4 do not and have a larger spatial spread.

In Table 1, we report the average peak signal-to-noise ratio on the validation set. Table 1 illustrates various observations listed below.

Lack of adaptivity The values on the diagonal are higher than the off-diagonal terms, except for the CT experiment where the family trained network is slightly better in average. This just reflects the fact that the best way to reconstruct images for a given application is to train the network for this specific application.

The drop of peak signal-to-noise ratio (PSNR) when using a network trained with the wrong operator can be as high as 9dB for the denoising net on the MRI experiment (see trained on $|$, applied on $-$). This drop is more moderate, but yet really significant (MRI: 5dB, CT: 2.4dB, Blur: 22dB) for the unrolled net. This is a striking illustration of the strong dependency of a reconstruction network to the operator used at the training stage. We illustrate the artifacts that can appear when the operator is trained on a different operator for the MRI application in Fig. 4. We can clearly see horizontal stripes oscillating at a high frequency, suggesting that the network did not properly learn to reconstruct the corresponding Fourier coefficients.

Peculiar case of deblurring nets The deblurring application has an important peculiarity: the basic block of the convolutional neural network is identical to the forward operator. Hence, when an unrolled network is trained, we can expect the networks $\mathcal{D}(w_k, \cdot)$ to not only act as "denoisers", but also as deconvolution mappings. This fact might explain the catastrophic lack of adaptivity in Table 1d. For instance with the network trained on θ_2, we obtain an average performance of 27.5dB without training mismatch and less than 10dB with a mismatch. This also confirms the conclusions of the introductory example in Fig. 1.

A similar, yet less obvious phenomenon seems to occur with the CT experiment. When training a network on the equiangular pattern θ_0, the lack of adaptivity is particularly striking. We believe that this may as well be due to a particular "algebraic compatibility" between convolution operators and the regularly spaced Radon transform. Further investigations should be conducted to further strengthen this hypothesis.

Superiority of unrolled nets The unrolled networks provide better reconstruction results than the denoising net in the MRI experiment. The overall gain on the diagonal varies between 1.4dB and 1.7dB for this particular application, which is significant. This is in accordance with recent comparisons of both strategies [66]. Hence, we only consider unrolled nets for the forthcoming experiments.

Optimal acquisition schemes Looking at the diagonal of the tables in Table 1 reveals that some acquisition schemes are better than others when the networks are trained properly. In the MRI experiment for instance, we see that the $+$ sampling scheme, yields a PSNR of 38.0dB in average while it drops to 37.1dB for the $-$ sampling scheme. This is in accordance with recent results [90, 92, 36]. For CT, the best sampling scheme is the standard equispaced one, which may not come as a surprise. In deblurring, it seems that the blur related to θ_3 is particularly hard to invert. This is probably due to a larger spatial spread.

6.2.2 Training on an operator family

Let us now study what happens, when training the reconstruction networks by varying the forward operators, as in (4). In what follows, we let ID denote the *"ideal" denoising* network and IU denote the *"ideal" unrolled* network. By ideal, we mean that the networks have been trained and tested with the same operator. They serve as a benchmark that cannot be outperformed on the training dataset for a given architecture. We let FD and FU denote the *"family" denoising* and *family unrolled* networks, which have been trained over a complete family. We also tested the P&P approach. We used an unrolled ADMM for different numbers of iterations K. The state-of-the-art DRUNet [97] network was used as an embedded denoiser and it was carefully trained on the FastMRI dataset for MRI, on MS COCO for deblurring and on ImageNet for the CT experiment (since the LIDC [7] and LoDoBap [58] datasets

Train

Evaluation	○	–	|	+	×	family
○	36.30 ±4.30	31.54 ±2.62	33.15 ±2.91	35.52 ±3.71	35.19 ±3.58	36.03 ±4.10
–	32.93 ±2.79	35.43 ±3.96	28.27 ±2.54	33.45 ±3.01	31.32 ±2.62	35.06 ±3.74
|	32.68 ±2.84	26.71 ±2.50	36.20 ±4.20	34.31 ±3.20	30.15 ±2.62	35.89 ±3.99
+	35.36 ±3.65	29.76 ±2.57	32.10 ±2.94	36.34 ±4.28	32.83 ±2.74	35.35 ±3.71
×	35.21 ±3.63	32.77 ±2.72	33.01 ±2.75	33.77 ±2.95	36.04 ±4.19	35.18 ±3.59

(a) MRI: denoising net

Train

Evaluation	○	–	|	+	×	family
○	38.04 ±5.13	37.17 ±4.60	36.72 ±4.48	37.91 ±5.05	37.87 ±5.02	38.00 ±5.09
–	35.93 ±4.10	37.09 ±4.64	32.97 ±2.60	35.65 ±4.03	35.38 ±3.64	36.97 ±4.57
|	35.37 ±4.09	32.13 ±3.06	37.61 ±4.82	37.09 ±4.53	36.50 ±4.18	37.52 ±4.77
+	37.77 ±4.92	36.09 ±4.09	36.41 ±4.02	37.98 ±5.03	37.32 ±4.60	37.88 ±4.96
×	37.56 ±4.90	36.96 ±4.47	35.74 ±4.17	36.96 ±4.50	37.74 ±4.98	37.66 ±4.92

(b) MRI: unrolled net

Model

Evaluation	θ_0	θ_1	θ_2	θ_3	family
θ_0	37.13 ±2.30	37.00 ±2.29	37.06 ±2.30	36.82 ±2.28	37.24 ±2.33
θ_1	34.96 ±2.37	36.95 ±2.27	36.63 ±2.30	36.73 ±2.28	37.12 ±2.30
θ_2	35.17 ±2.30	36.28 ±2.36	36.96 ±2.27	36.83 ±2.26	37.09 ±2.29
θ_3	34.25 ±2.35	35.38 ±2.32	35.81 ±2.25	36.63 ±2.21	36.58 ±2.21

(c) CT: unrolled net

Train

Evaluation	θ_0	θ_1	θ_2	θ_3	θ_4	family
θ_0	28.20 ±3.68	18.72 ±3.07	10.01 ±1.62	9.41 ±1.94	7.26 ±1.95	28.09 ±3.69
θ_1	23.52 ±3.25	28.49 ±3.79	12.01 ±1.15	9.97 ±1.56	6.13 ±1.79	28.34 ±3.77
θ_2	26.63 ±3.80	26.14 ±3.77	27.49 ±3.77	12.54 ±1.47	12.77 ±2.00	27.42 ±3.77
θ_3	24.20 ±3.67	24.08 ±3.58	24.18 ±3.59	26.03 ±3.66	24.94 ±3.64	25.82 ±3.66
θ_4	25.37 ±3.80	25.08 ±3.75	25.80 ±3.73	17.77 ±2.41	26.77 ±3.76	26.69 ±3.75

(d) Deblurring: unrolled net

Table 1: The lack of adaptivity. In these tables, we measure the performance of various solvers for non blind inverse problems. We train a network for a given operator and test it on others. The results for a network trained on a family of forward operators is also given in the last column of each table. The average PSNR and its standard deviation are evaluated on the different validation dataset. This means on about 7 000 images in MRI, 4 096 in CT and 5 000 in deblurring.

Model

Evaluation	ID	FD	IU	FU	P&P 10	P&P 20	P&P 50	P&P 100	R&R
○	36.30 ±4.30	36.03 ±4.10	38.04 ±5.13	38.00 ±5.09	35.23 ±3.58	35.67 ±3.79	36.29 ±4.26	35.60 ±4.21	35.20 ±3.56
−	35.43 ±3.96	35.06 ±3.74	37.09 ±4.64	36.97 ±4.57	34.04 ±3.28	34.52 ±3.49	35.06 ±3.90	34.86 ±3.94	32.88 ±2.92
|	36.20 ±4.20	35.89 ±3.99	37.61 ±4.82	37.52 ±4.77	35.29 ±3.59	35.61 ±3.73	36.13 ±4.11	35.45 ±4.30	34.21 ±3.23
+	36.34 ±4.28	35.35 ±3.71	37.98 ±5.03	37.88 ±4.96	35.37 ±3.61	35.76 ±3.80	36.37 ±4.22	35.75 ±4.21	34.13 ±3.17
×	36.04 ±4.19	35.18 ±3.59	37.74 ±4.98	37.66 ±4.92	34.79 ±3.45	35.26 ±3.67	35.86 ±4.14	35.31 ±4.13	33.57 ±3.04

(a) MRI – denoising (ID, FD) and unrolled (IU, FU)

Model

Evaluation	IU	FU	P&P 4	P&P 10	P&P 20
θ_0	37.13 ±2.30	37.24 ±2.33	34.63 ±2.66	36.50 ±2.13	36.29 ±2.10
θ_1	36.95 ±2.27	37.12 ±2.30	34.51 ±2.71	36.28 ±2.10	36.01 ±2.05
θ_2	36.96 ±2.27	37.09 ±2.29	34.51 ±2.71	36.24 ±2.11	35.92 ±2.07
θ_3	36.64 ±2.21	36.58 ±2.21	33.88 ±2.74	35.67 ±2.11	35.39 ±2.07

(b) CT – unrolled

Model

Evaluation	IU	FU	P&P 5	P&P 10	P&P 20
θ_0	28.20 ±3.68	28.09 ±3.69	27.03 ±3.76	27.16 ±3.85	27.13 ±3.89
θ_1	28.49 ±3.79	28.34 ±3.77	27.31 ±3.80	27.45 ±3.91	27.45 ±3.96
θ_2	27.49 ±3.77	27.42 ±3.77	26.31 ±3.88	26.45 ±3.93	26.44 ±3.94
θ_3	26.03 ±3.66	25.82 ±3.66	24.48 ±3.64	24.72 ±3.67	24.80 ±3.67
θ_4	26.77 ±3.76	26.69 ±3.75	25.58 ±3.86	25.74 ±3.89	25.76 ±3.89

(c) Deblurring – unrolled

Table 2: The average PSNR and standard deviation in dB, for various reconstruction approaches and operators. The evaluation dataset contains about 7 000 images in MRI, 4 096 in CT and 5 000 in deblurring. ID, IU ("ideal"): the network is trained on the same operator it is applied on. FD, FU ("family"): the network is trained on a family of operators, as advocated in this paper. P&P: plug-and-play ADMM network with different numbers of iterations $K \in \{4, 5, 10, 20, 50, 100\}$. Notice that the schemes $+, \times$ for MRI and the parameters θ_3, θ_4 for CT and image deblurring do not belong to the training family and therefore allow assessing the generalization capability of the reconstruction networks.

contain images with many artifacts). It was trained specifically to denoise the images with various levels of white Gaussian noise. Finally, we implemented the reuse & regularize network (R&R) [33] composed of $K = 10$ iterations. The embedded inversion network consists of a pseudo-inverse, followed by a DRUNet network trained for the ◯ sampling scheme. The hyperparameters in the method (see [33]) were tuned to produce the best results. Table 2 shows the performance of the different architectures. The following conclusions can be drawn.

Price of adaptivity By comparing the columns FU and IU in Table 2, we see that training on a family leads to really moderate drops of performance compared to a training on a single operator. Surprisingly, it even outperforms the model trained on a single operator for the CT experiment. This might be caused by a better capacity to escape spurious minimizers at the training stage, or by a discrepancy between the testing and training datasets. Some differences are still significant for the denoising network, but they only become marginal for the unrolled networks. This has to be compared to the huge gain of adaptivity of the method: a single network is now able to tackle a vast family of different problems within a class.

For operators in the training family, the performance drop of unrolled networks is of at most 0.12dB in MRI (−), and 0.15dB in deblurring (θ_1) and there is no drop for CT. Compared to values reaching more that 10dB in the previous section, this feature is really remarkable. This perfectly illustrates one of the take home message of our paper: training unrolled networks on a family seems to not degrade the performance significantly while providing a huge boost of adaptivity.

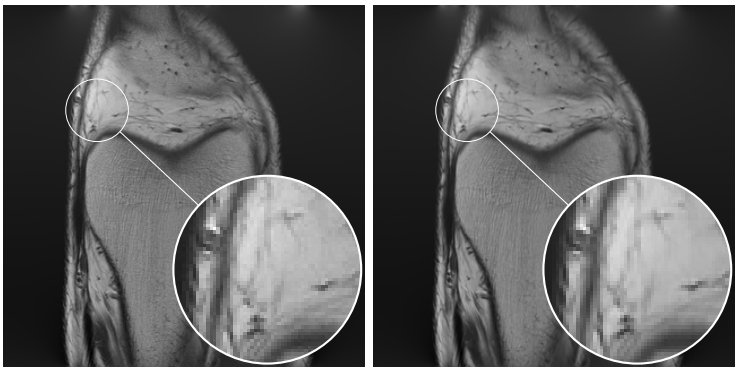
How does it generalize? It is informative to look at the last rows of the different tables in Table 2. There, we apply the unrolled networks to operators that were not encountered at the training stage. Hence, comparing IU to FU allows us to assess the generalization ability of the networks. We observe

a performance drop of 0.1dB at most in MRI, 0.06dB in CT and 0.21dB in deblurring. It can also be counterbalanced by the fact that the operators differ significantly from what was observed at the training stage: we amplified the perturbations by a factor 2 for the CT and deblurring experiments. Overall, this experiment suggests that a training stage on a family provides some generalization capabilities.

Plug & Play (P&P) When looking at Table 2, we see that the P&P approach is outperformed uniformly by both unrolled networks trained on a family and on a fixed operator. The drop lies between 1 and 2dB for the MRI experiments, about 1dB for the deblurring experiments and less than 1dB for the CT experiments. This suggests that for a given reconstruction architecture, it is beneficial to train the proximal networks for a specific task rather than using a *universal* denoiser, as is the case in P&P. This observation should be carefully examined with recent progress in diffusion models [40]. Notice however, that compared to the off-diagonal elements of Table 1 which correspond to a network trained on a fixed operator and evaluated with a different one, the P&P approach is still really competitive and likely preferable.

We also want to mention that FU does not seem to extrapolate well to problems completely different from the ones it was trained for. Indeed, we trained FU for an MRI reconstruction problem and tested it for a deblurring application. In this application, which is not reported in this paper, the P&P approach was considerably more consistent. In a sense, we can see the proposed training as an intermediate step between the P&P approach (adaptable to all inverse problems) and the traditional training of reconstruction networks (perfectly adapted to a single operator).

Reuse & Regularize (R&R) Finally, the R&R approach applied in MRI can improve the results for some problems compared to a model trained on a single operator. However, it seems that our simpler training approach provides significantly better results. Hence, we did not consider this alternative for the CT and deblurring problems.



(a) Original image (b) Tr.| Ev.|, 38.9dB

(c) Tr.| Ev.−, 33.7dB (d) FU Ev.−, 38.0dB

Figure 4: Examples of reconstructions using the unrolled network. We trained it on | (4b, 4c) and on a family (4d). We tested it on | (4b) and − (4c, 4d).

Test	Recon. PSNR with θ_0 (dB)	Recon. PSNR with estimated $\hat{\theta}_1$ (dB)	Error traj. $\|\xi_1 - \hat{\xi}_1\|_\infty$	PSNR $\hat{s}_1$ (dB)
1	17.26	33.04	0.039	44.15
2	13.09	29.42	0.045	36.30
3	13.85	37.69	0.032	47.91
4	18.50	34.95	0.026	40.26
5	15.46	33.65	0.050	44.35
6	20.57	31.43	0.008	58.20
7	20.90	33.92	0.021	44.89
8	18.60	34.29	0.031	49.30
9	10.88	33.05	0.016	41.56
Avg	16.57	33.49	0.030	45.21

Table 3: Additional experiments for self-calibrated MRI with different images. The initial error $\|\xi_0 - \xi_1\|_\infty$ on the trajectories is 5 pixels for all test cases. We recall that $\hat{s}_1$ coincides with the estimated sensitivity maps and ξ_1 with the sampling trajectory.

6.3 Blind inverse problems

In this section, we illustrate how training on a family of operators helps solving different blind inverse problems. We assume that

$$y = A(\theta_1)x + b, \tag{17}$$

for some unknown parameter θ_1 describing the forward model. We then solve (5) using the methods described in Section 5.2, resulting in an estimate $\hat{\theta}_1$ of θ_1. Fig. 5, 6, 7 show the performance of the solver for various applications. Let us analyze these results.

6.3.1 Magnetic Resonance Imaging

This application provides surprisingly good results for various reasons:

- To the best of our knowledge, no one yet attempted to estimate the sensitivity maps and trajectory errors jointly. Estimating divergence in trajectories might look hopeless at first sight, which may explain this fact. Indeed, looking at the differences between ξ_1 and ξ_0 (see top-right and the zoom on the right-most column of Fig. 5) we see that the frequency shifts are huge (up to 5 pixels).
- The total number of parameters to estimate is large. Indeed, it consists in the 104×15 parameters describing the sensitivity maps and the 32 parameters describing the convolution kernel that perturbs the trajectories, i.e. 1592 parameters.

If solved without any correction, the reconstruction results are disastrous (see the 2nd column). Solving the consistency problem (5) provides near perfect estimates of $\hat{\theta}$ for all reconstruction mappings. For instance, the green $\hat{\xi}_1$ and orange ξ_1 trajectories cannot be distinguished on the right column. This may come as a surprise, and seems to suggest that this particular blind inverse problem is not as hard as it may seem at first sight. This might be due to some redundancy in the data: the 15 reception coils associated to a slight oversampling of the k-space center (all the trajectories start exactly from the center) seem to ensure the identifiability of the problem. A nice research perspective is to explain this phenomenon from a theoretical viewpoint.

The reconstruction result obtained with the neural network trained on a family is significantly better than the two other ones (more than +1.3dB compared to the one trained on θ_0 and to the P&P approach). In particular, the bone texture is reconstructed with the proposed approach, while it is not for the two others.

To further validate the method, we tested the methodology on 9 additional images. The results are reported in Table 3. As can be seen, the method recovers good estimates of the sensitivity maps s_1 and trajectories ξ_1 in all cases. This results in a huge PSNR increase, since the forward model is essentially correct after estimation.

6.3.2 Computerized tomography

In this application, a model mismatch might occur due to the motion of a patient in the scanner. Correcting this mismatch is essential. Not accounting for it, can result in severe artifacts including some details loss and blur as can be seen in Fig. 6.

Test	Recon. PSNR with θ_0 (dB)	Recon. PSNR with estimated $\hat{\theta}_1$ (dB)	Shift err. $\|s_1 - \hat{s}_1\|_\infty$	Angle err. $\|\alpha_1 - \hat{\alpha}_1\|_\infty$
1	28.16	33.91	0.52	0.25
2	30.63	38.17	0.66	0.31
3	27.10	32.47	0.72	0.33
4	26.93	34.09	0.36	0.28
5	26.81	34.26	0.42	0.22
6	27.52	37.89	0.62	0.12
7	29.68	37.21	0.57	0.34
8	27.44	37.16	0.68	0.40
9	26.89	35.84	0.92	0.38
Avg	27.91	35.67	0.61	0.29

Table 4: Additional experiments for self-calibrated CT with different images and operators. For all test cases, the initial angle error is $\|\alpha_0 - \alpha_1\|_\infty = 1.3°$ and $\|s_0 - s_1\|_\infty = 1$ pixel. We see that the "deep unrolled prior" method provides good estimates of the true parameters $\theta_1 = (\alpha_1, s_1)$ in all test cases.

To identify the forward model, we ran the Adam optimizer on the parameters $\theta = (\alpha, s)$ for 2000 iterations. In this application, α represents the angle of the parallel shots and s their shift at origin. All the reconstruction methods are able to significantly reduce the model mismatch, passing from maximal angles shifts of 7 degrees to less that 1 degree. Similarly, the shifts at origin are reduced from more than a pixel to about 0.3 pixel. The reconstruction performance is significantly improved after estimating the forward model with PSNR increases of 4dB and more. The neural network trained on a family provides the best reconstruction results on this example.

Similarly to blind MRI, Table 4 shows that the "deep unrolled prior" method consistently provides good estimates of the forward model and significantly improves the reconstruction quality for the CT experiments.

6.3.3 Blind deblurring

Finally, we present some results of the "deep unrolled prior" methodology in Fig. 7. In this experiment, we simply used 3 Zernike polynomials and optimized them globally using Bayesian optimization. Hence, the recovered kernels can be safely considered as the (near) global minimizers of the functional 5. It appears that in every case, the method returns the same kernel, which is the one with the smallest possible extent in the family. It coincides with all Zernike coefficients being 0, i.e. a Airy pattern.

Hence, for this specific application, the deep unrolled prior methodology is not able to correctly identify the blur kernel. The reconstruction network still improves the image quality in average, but we cannot recommend this method for this application. Understanding the observed behavior requires further work, but shows that the methodology proposed does not work universally.

7 Conclusion

In this work we proposed a training procedure to address the adaptivity and robustness issues in model-based unrolled neural network for inverse problems. We showed that a careful training leads to networks able to adapt to different forward operators without compromising the image quality. We also showed that minimizing a consistency term with the proposed networks makes it possible to solve challenging blind inverse problems in magnetic resonance imaging and computerized tomography. In particular, we were able to correct trajectory errors and evaluate sensitivity maps convincingly for the first time. The method can be seen as a new type of P&P method to recover operators in blind inverse problems. It however does not work for blind deblurring. This experiment shows that a theoretical analysis to better understand when and why the method works is needed.

This work opens new interesting perspectives for computational imaging. A recent trend consists in optimizing the forward model and the reconstruction algorithm jointly (see e.g. [92, 37, 90] for examples in MRI). With a reconstruction method capable of adapting to a vast family of operators, it becomes possible to restrict the attention to the optimization of the forward model only [36].

Extending the method to other applications seems relevant as well. An interesting perspective would be to add motion correction for MRI. A motion in the image domain translates to a phase modulation in the Fourier domain. This is a critical issue in practice. The disconcerting ease with which we solved

the estimation of trajectory shifts and sensitivity maps, sparks good hopes to solve this long resisting problem.

A Detailed description of the forward models

Parallel Magnetic Resonance Imaging Our aim here is to reconstruct images from under-sampled Fourier samples with unknown sensitivity maps associated to $J \in \mathbb{N}$ reception coils, and with inaccurate trajectories. The parameter θ can be decomposed as $\theta = (\tau, \omega)$, where τ is a parameter describing the sensitivity maps and ω describes a perturbation of the sampling locations. To the best of our knowledge, these two problems have not been treated jointly in the literature yet.

Let $\mathcal{F}(\xi)$ denote the non-uniform Fourier transform (NUFT) [71] at frequencies $\xi = (\xi_1, \dots, \xi_M)$, defined by

$$[\mathcal{F}(\xi)]_{m,n} = e^{-i\langle p_n, \xi_m \rangle}$$

where $(p_n)_{1 \leq n \leq N}$ is a set of 2D positions on a grid. We construct a family of forward operators $\mathcal{A} = \{A(\xi, \theta), \xi \in \Xi, \theta \in \Theta\}$, where $\Xi \subset \mathbb{R}^{2 \times M}$ is a set of 2D sampling schemes with M sampling points. The parameter space $\Theta = \mathcal{T} \times \Omega$ describes the set of admissible parameters for the sensitivity maps $\mathcal{T}$ and for the perturbation Ω. The measured signal $y = (y^{(1)}, \dots, y^{(J)})$ is acquired through J coils. The m-th measurement acquired by the j-th coil is defined by

$$y_m^{(j)} = [A(\xi, \theta)x]_{m,j} + b_{m,j} = \left[\mathcal{F}(h(\omega) \star \xi) \left(x \odot s(\tau^{(j)})\right)\right]_m + b_{m,j}. \tag{18}$$

The mapping $s : \tau^{(j)} \in \mathbb{R}^T \mapsto s(\tau^{(j)}) \in \mathbb{C}^N$ parametrizes the coil sensitivity maps. Since the sensitivity maps are smooth, we use a parametrization based on thin plate splines [29]. The total number of parameters that encode the sensitivity map is $T = 104$. It consists of the splines coefficients using 7×7 regularly spaced control points, plus the coefficients of a first degree polynomial. This has to be multiplied by two for the real and imaginary parts.

Following [87, 27], we assume that the trajectory ξ is perturbed by a convolution with an impulse response $h(\omega)$. The symbol $\star$ in (18) corresponds to a discrete convolution. Evaluating the convolution filter $h(\omega)$ is known as a challenging problem that can be addressed with expensive field cameras [27]. Here, in the spirit of [87], we will rather treat it as a blind inverse problem. We parametrize h as a linear combination of the form $h(\omega) = \sum_{o=1}^{O} \omega_o h_o$, where $(h_o)_{1 \leq o \leq O}$ is an orthogonal basis. In practice, we simply use compactly supported filters of size $O = 32$ and $(h_o)_{1 \leq o \leq O}$ corresponds to the first 32 elements of the canonical basis.

Computerized Tomography Our aim is to reconstruct images from parallel beam computerized tomography. The parameter θ describes the projection angles and shift at origin (allowing to model the patient motion).

We assume that the CT scan uses parallel beams and that it performs J acquisitions with a receptor that has M sensors, resulting in $J \times M$ measurements. In this application, the parameter $\theta = (\alpha, s)$ represents the angles $\alpha \in \mathbb{R}^J$ and the shifts at the origin $s \in \mathbb{R}^J$ that describe the beams trajectories. If m corresponds to the m-th pixel of the receptor and if we index the acquisitions by $1 \leq j \leq J$, we get

$$y_m^{(j)} = \iint_\Omega x(u_x, u_y)\, \delta_{u_x \cos(\alpha_j) + u_y \sin(\alpha_j) = p_m + s_j}\, \mathrm{d}u_x \mathrm{d}u_y + b_m^{(j)}, \tag{19}$$

with $\Omega = [-N_x/2, N_x/2] \times [-N_y/2, N_y/2]$ and $p \in [\![-M/2, \dots, M/2 - 1]\!]$. A perfect model would correspond to α being equispaced angles and $s = 0$. The forward model can be computed using the Fourier slice theorem. This corresponds to performing a 2D NUFT and we resort to the same library used for MRI (see `https://github.com/albangossard/Bindings-NUFFT-pytorch`).

Deblurring in optics In this application, we wish to solve problems appearing in diffraction limited systems. The parameter θ describes the point spread function through the theory of diffraction. The acquisition model in this application simply reads

$$y = h(\theta) \star x + b, \tag{20}$$

where $h(\theta)$ is the blur kernel. We consider blurs generated by Fresnel diffraction theory [35]. This theory is the most commonly adopted in optics since the works of Zernike (see e.g. [68, 53]). It is widely used in microscopy or astronomy.

The blur kernel is parameterized by a vector $\theta \in \mathbb{R}^7$ and the convolution kernel is expressed as

$$h(\theta) = c \left| \int_{\|w\|_2 \leq f_c} \exp\left(2i\pi \left[\sum_{k=1}^{K} \theta_k Z_k + \langle u, w \rangle \right] \right) \mathrm{d}w \right|^2 . \tag{21}$$

In this expression, f_c is a cutoff frequency and c is a scaling parameter such that $\|h\|_1 = 1$. The expansion $\sum_{k=1}^{K} \theta_k Z_k$ describes the pupil function of an objective with a circular aperture. The functions Z_k are Zernike polynomials and the vector θ parametrizes the so-called pupil function.

B Details on the pseudo-inverse and the resolvent

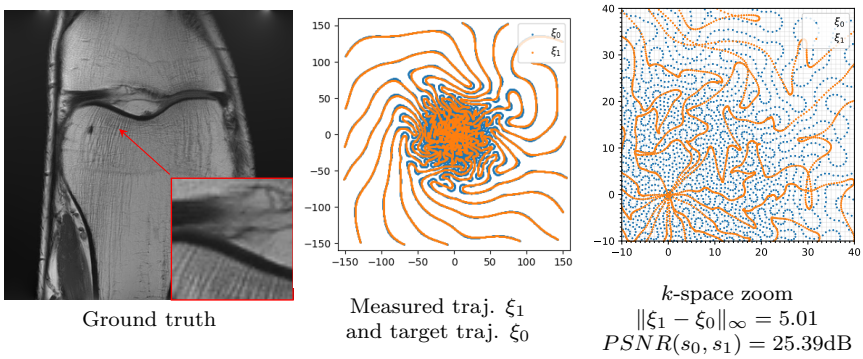
The reconstruction networks all rely on the pseudo-inverse $A(\theta)^\dagger$ or on the resolvent $(A(\theta)^* A(\theta) + \lambda \mathrm{Id})^{-1}$. In all our experiments, we implemented them using a conjugate gradient algorithm run for a fixed number of iterations. In all cases, we set this number to ensure a relative residue below a threshold of 10^{-4}.

The pseudo-inverse $x \mapsto A(\theta)^\dagger x$ in the reconstruction networks is approximated by solving the symmetric positive definite system $(A^* A + \epsilon \mathrm{Id})x = y$. The parameter ϵ is set to a small value that was tuned manually for each application.

For larger λ, the linear system is better conditioned and can be solved using less iterations with the same conjugate gradient iteration.

Acknowledgment

This work was supported by the ANR Micro-Blind and by the ANR LabEx CIMI (grant ANR-11-LABX-0040) within the French State Programme "Investissements d'Avenir". P. Weiss acknowledges the support of AI Interdisciplinary Institute ANITI funding, through the French "Investing for the Future— PIA3" program under the Grant Agreement ANR-19-PI3A-0004. This work was performed using HPC resources from GENCI-IDRIS (Grant 2021-AD011012210R1). Parts of the code can be made available on demand. The authors wish to thank Emmanuel Soubies, Valentin Debarnot, Nathanaël Munier, Frédéric de Gournay and the anonymous reviewers for their comments and advice which helped us improving the paper.



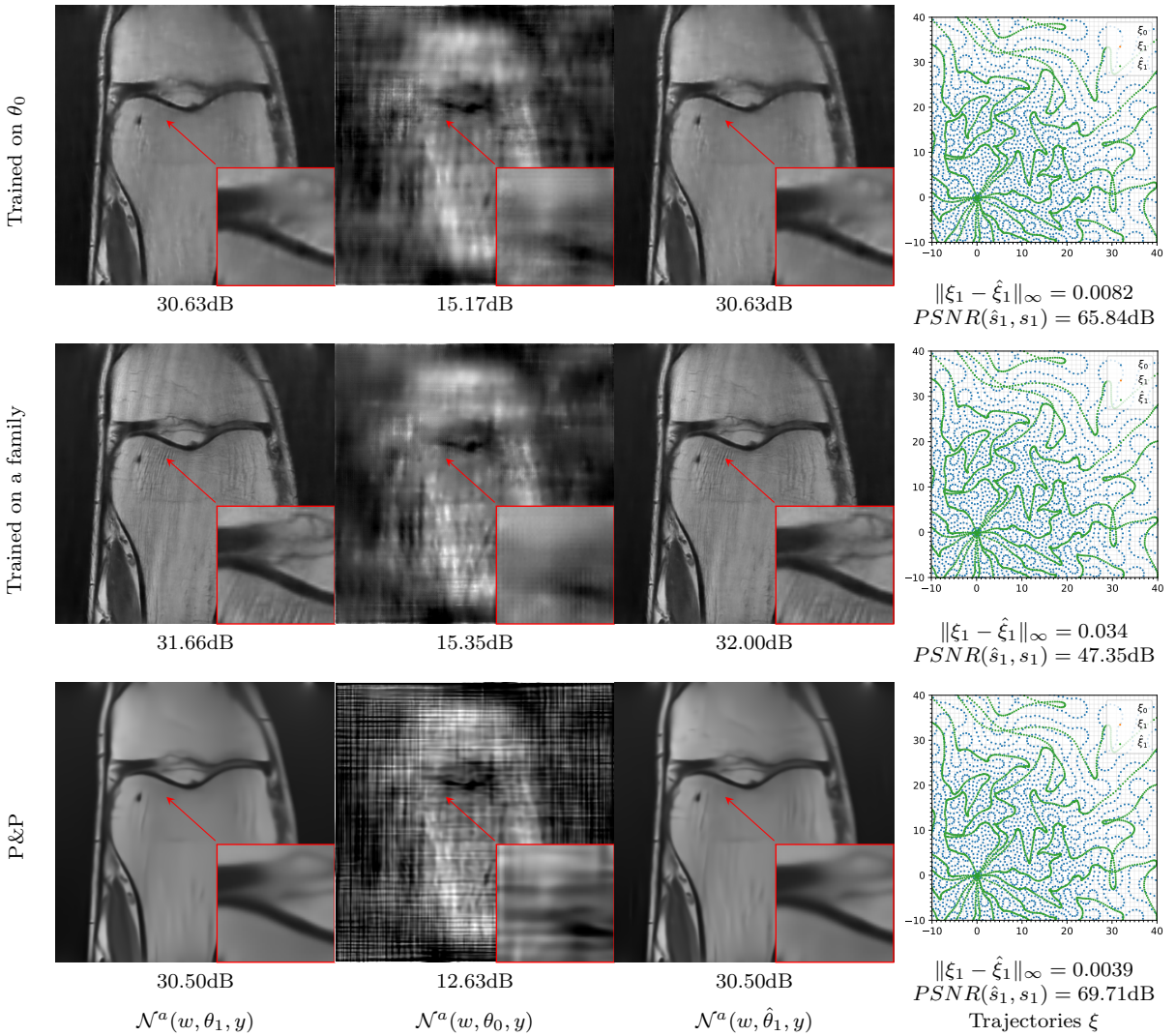


Figure 5: Self-calibrated MRI. *1st column:* reconstruction with a perfect knowledge of the forward model θ_1. *2nd column:* reconstruction assuming the wrong forward model θ_0. *3rd column:* reconstruction using the estimated forward model $\hat{\theta}_1$. *4th column:* estimate of the operator. We display the maximal distance between sampling points $\|\xi_1 - \hat{\xi}_1\|_\infty$ as well as the PSNR of the estimated sensitivity maps $\hat{s}_1$. From top to bottom: different training strategies are compared. *2nd row:* trained on θ_0. *3rd row:* trained on a family of operators. *4th row:* using a P&P prior. The PSNR is indicated below each image. The noise level given to the P&P denoiser has been tuned so as to yield the best PSNR on the non-blind problem. Other models do not require tuning at evaluation.

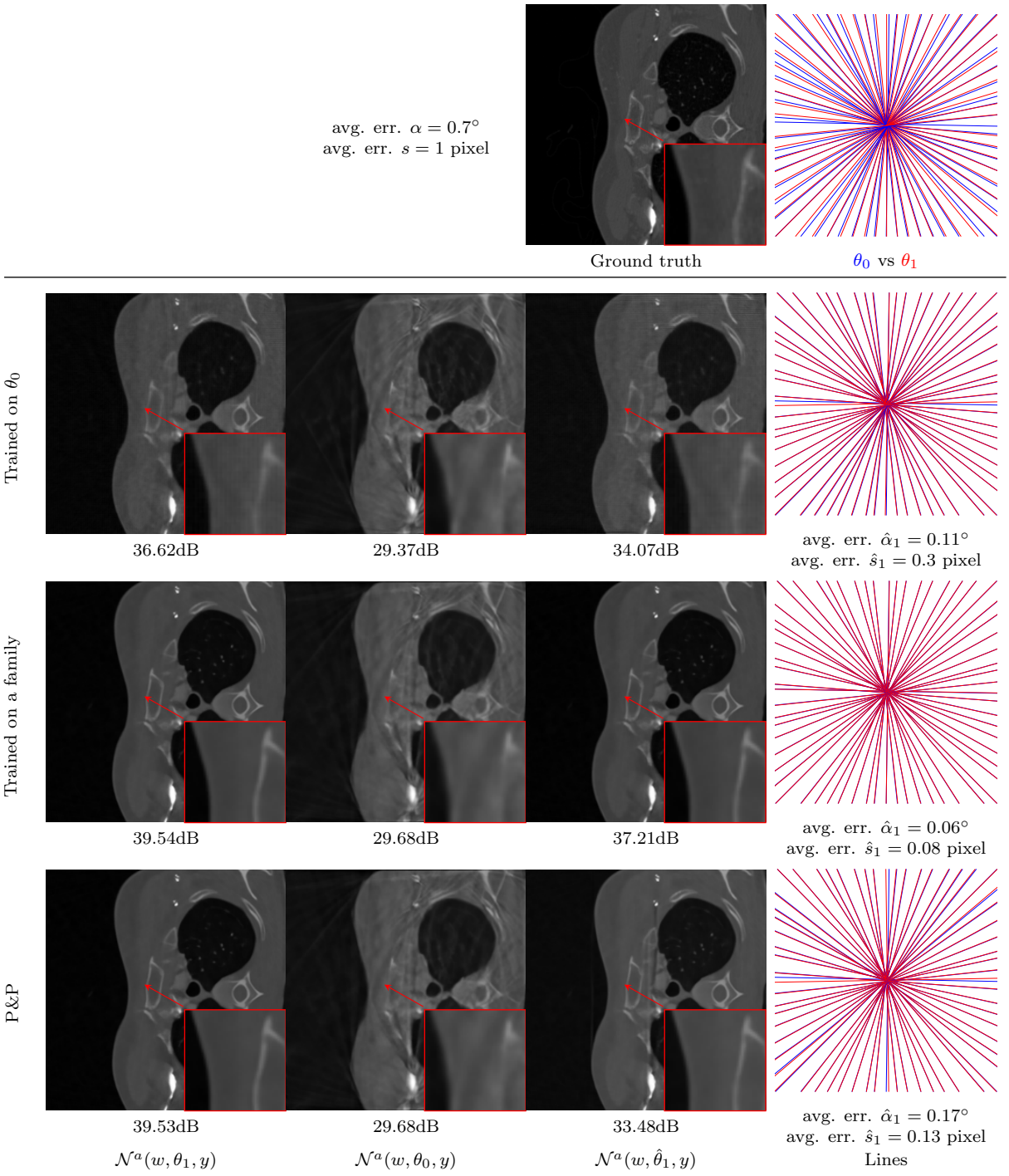


Figure 6: Self-calibrated computerized tomography. *1st column:* reconstruction with a perfect knowledge of the forward model $\theta_1 = (\alpha_1, s_1)$. *2nd column:* reconstruction assuming the wrong forward model θ_0. *3rd column:* reconstruction using the estimated forward model $\hat{\theta}_1$. *4th column:* true θ_1 (blue) and estimated $\hat{\theta}_1$ parameters (red) of the forward model. We display the average angle error and the average shift error. *2nd row:* trained on θ_0. *3rd row:* trained on a family of operators. *4th row:* using a P&P prior. The PSNR of the reconstructed image are indicated below each image. The noise level given to the P&P denoiser has been tuned as to yield the best PSNR on the non-blind problem. Other models do not require tuning at evaluation.

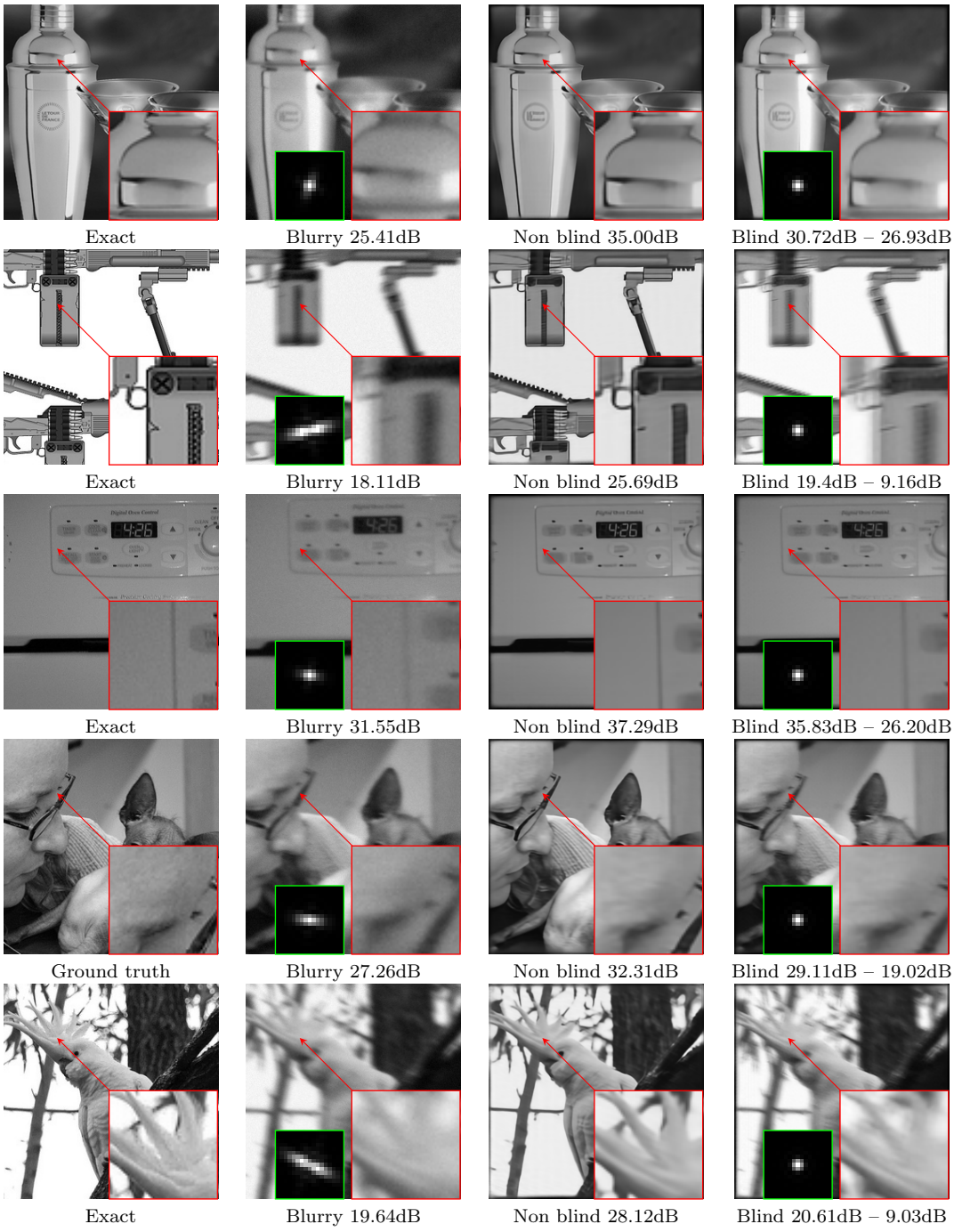


Figure 7: The failure of deep unrolled prior for blind deblurring. *1st column:* Ground truth image. *2nd column:* Blurry image and the corresponding blur kernel. *3rd column:* Non blind reconstruction using the network trained on a family. *4th column:* Blind reconstruction using the deep unrolled prior. The green box indicates the recovered kernel. The PSNR of the reconstructed image – blur kernel are indicated below each image. From top to bottom: different images/blur kernels. As can be seen, the method always returns the same kernel, which is the smallest possible (an Airy pattern) in the family of Fresnel diffraction blurs. It therefore fails to estimate the operator.

References

[1] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[2] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[3] Hemant K Aggarwal, Merry P Mani, and Mathews Jacob. Modl: Model-based deep learning architecture for inverse problems. *IEEE transactions on medical imaging*, 38(2):394–405, 2018.

[4] Ali Ahmed, Benjamin Recht, and Justin Romberg. Blind deconvolution using convex programming. *IEEE Transactions on Information Theory*, 60(3):1711–1732, 2013.

[5] Fabian Altekrüger, Alexander Denker, Paul Hagemann, Johannes Hertrich, Peter Maass, and Gabriele Steidl. Patchnr: Learning from small data by patch normalizing flow regularization. *arXiv preprint arXiv:2205.12021*, 2022.

[6] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 117(48):30088–30095, 2020.

[7] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

[8] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[9] Muhammad Asim, Fahad Shamshad, and Ali Ahmed. Blind image deconvolution using deep generative priors. *IEEE Transactions on Computational Imaging*, 6:1493–1506, 2020.

[10] Yuanchao Bai, Gene Cheung, Xianming Liu, and Wen Gao. Graph-based blind image deblurring from a single photograph. *IEEE transactions on image processing*, 28(3):1404–1418, 2018.

[11] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

[12] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017.

[13] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[14] Emrah Bostan, Reinhard Heckel, Michael Chen, Michael Kellman, and Laura Waller. Deep phase decoder: self-calibrating phase microscopy with an untrained deep neural network. *Optica*, 7(6):559–562, Jun 2020.

[15] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. *Advances in neural information processing systems*, 20, 2007.

[16] Claire Boyer, Nicolas Chauffert, Philippe Ciuciu, Jonas Kahn, and Pierre Weiss. On the generation of sampling schemes for magnetic resonance imaging. *SIAM Journal on Imaging Sciences*, 9(4):2039–2072, 2016.

[17] Patrizio Campisi and Karen Egiazarian. *Blind image deconvolution: theory and applications*. CRC press, 2017.

[18] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[19] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European conference on computer vision*, pages 221–235. Springer, 2016.

[20] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998.

[21] Nicolas Chauffert, Philippe Ciuciu, Jonas Kahn, and Pierre Weiss. A projection method on measures sets. *Constructive Approximation*, 45(1):83–111, 2017.

[22] Liang Chen, Faming Fang, Tingting Wang, and Guixu Zhang. Blind image deblurring with local maximum gradient prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1742–1750, 2019.

[23] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

[24] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[25] Valentin Debarnot and Pierre Weiss. Deep-blur: Blind identification and deblurring with convolutional neural networks. 2022.

[26] Steven Diamond, Vincent Sitzmann, Felix Heide, and Gordon Wetzstein. Unrolled optimization with deep priors. *arXiv preprint arXiv:1705.08041*, 2017.

[27] Benjamin E Dietrich, David O Brunner, Bertram J Wilm, Christoph Barmet, Simon Gross, Lars Kasper, Maximilian Haeberlin, Thomas Schmid, S Johanna Vannesjo, and Klaas P Pruessmann. A field camera for mr sequence monitoring and system analysis. *Magnetic resonance in medicine*, 75(4):1831–1840, 2016.

[28] Weisheng Dong, Peiyao Wang, Wotao Yin, Guangming Shi, Fangfang Wu, and Xiaotong Lu. Denoising prior driven deep neural network for image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2305–2318, 2018.

[29] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[30] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *Acm Siggraph 2006 Papers*, pages 787–794. 2006.

[31] Peter I Frazier. Bayesian optimization. In *Recent advances in optimization and modeling of contemporary problems*, pages 255–278. Informs, 2018.

[32] Martin Genzel, Jan Macdonald, and Maximilian Marz. Solving inverse problems with deep neural networks-robustness included. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[33] Davis Gilton, Gregory Ongie, and Rebecca Willett. Model adaptation for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:661–674, 2021.

[34] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision*, pages 622–635. Springer, 2012.

[35] J.W. Goodman. *Introduction to Fourier Optics*. Electrical Engineering Series. McGraw-Hill, 1996.

[36] Alban Gossard, Frédéric de Gournay, and Pierre Weiss. Bayesian optimization of sampling densities in mri. *arXiv preprint arXiv:2209.07170*, 2022.

[37] Alban Gossard, Frédéric de Gournay, and Pierre Weiss. Spurious minimizers in non uniform fourier sampling optimization. *Inverse Problems*, 2022.

[38] Nina M Gottschling, Vegard Antun, Ben Adcock, and Anders C Hansen. The troublesome kernel: why deep learning for inverse problems is typically unstable. *arXiv preprint arXiv:2001.01258*, 2020.

[39] Alexis Goujon, Sebastian Neumayer, Pakshal Bohra, Stanislas Ducotterd, and Michael Unser. A neural-network-based convex regularizer for image reconstruction. *arXiv preprint arXiv:2211.12461*, 2022.

[40] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

[41] Mark A Griswold, Peter M Jakob, Robin M Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase. Generalized autocalibrating partially parallel acquisitions (grappa). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 47(6):1202–1210, 2002.

[42] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1604–1613, 2019.

[43] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

[44] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated mri data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018.

[45] Kerstin Hammernik, Jo Schlemper, Chen Qin, Jinming Duan, Ronald M Summers, and Daniel Rueckert. σ-net: Systematic evaluation of iterative deep neural networks for fast parallel mr image reconstruction. *arXiv preprint arXiv:1912.09278*, 2019.

[46] Johannes Hertrich, Antoine Houdard, and Claudia Redenbach. Wasserstein patch prior for image superresolution. *IEEE Transactions on Computational Imaging*, 8:693–704, 2022.

[47] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[48] Steven M Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[50] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in neural information processing systems*, 22, 2009.

[51] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR 2011*, pages 233–240. IEEE, 2011.

[52] Deepa Kundur and Dimitrios Hatzinakos. Blind image deconvolution. *IEEE signal processing magazine*, 13(3):43–64, 1996.

[53] Vasudevan Lakshminarayanan and Andre Fleck. Zernike polynomials: a guide. *Journal of Modern Optics*, 58(7):545–561, 2011.

[54] Rémi Laumont, Valentin De Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, and Marcelo Pereyra. Bayesian imaging using plug & play priors: when langevin meets tweedie. *SIAM Journal on Imaging Sciences*, 15(2):701–737, 2022.

[55] Carole Lazarus, Pierre Weiss, Nicolas Chauffert, Franck Mauconduit, Loubna El Gueddari, Christophe Destrieux, Ilyess Zemmoura, Alexandre Vignaud, and Philippe Ciuciu. Sparkling: variable-density k-space filling curves for accelerated t2*-weighted mri. *Magnetic resonance in medicine*, 81(6):3643–3661, 2019.

[56] Bruno Lecouat, Thomas Eboli, Jean Ponce, and Julien Mairal. High dynamic range and super-resolution from raw image bursts. *ACM Trans. Graph.*, 41(4), jul 2022.

[57] Bruno Lecouat, Jean Ponce, and Julien Mairal. Lucas-kanade reloaded: End-to-end super-resolution from raw image bursts. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2350–2359. IEEE, 2021.

[58] Johannes Leuschner, Maximilian Schmidt, Daniel Otero Baguer, and Peter Maass. LoDoPaB-CT, a benchmark dataset for low-dose computed tomography reconstruction. *Scientific Data*, 8(1), April 2021.

[59] Yuelong Li, Mohammad Tofighi, Vishal Monga, and Yonina C Eldar. An algorithm unrolling approach to deep image deblurring. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7675–7679. IEEE, 2019.

[60] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[61] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[62] Marina Ljubenović and Mário A. T. Figueiredo. Plug-and-play approach to class-adapted blind image deblurring. *International Journal on Document Analysis and Recognition (IJDAR)*, 22(2):79–97, March 2019.

[63] Michael Lustig, Jin Hyung Lee, David L Donoho, and John M Pauly. Faster imaging with randomly perturbed, under-sampled spirals and ℓ_1 reconstruction. In *Proceedings of the 13th annual meeting of ISMRM*, page 685, Miami Beach, FL, USA, 2005.

[64] Chang Meng and James Nagy. Numerical methods for ct reconstruction with unknown geometry parameters. *Numerical Algorithms*, 92(1):831–847, 2023.

[65] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European conference on computer vision*, pages 783–798. Springer, 2014.

[66] Matthew J Muckley, Bruno Riemenschneider, Alireza Radmanesh, Sunwoo Kim, Geunu Jeong, Jingyu Ko, Yohan Jun, Hyungseob Shin, Dosik Hwang, Mahmoud Mostapha, et al. Results of the 2020 fastmri challenge for machine learning mr image reconstruction. *IEEE transactions on medical imaging*, 40(9):2306–2317, 2021.

[67] Michael K Ng, Pierre Weiss, and Xiaoming Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. *SIAM journal on Scientific Computing*, 32(5):2710–2736, 2010.

[68] Robert J Noll. Zernike polynomials and atmospheric turbulence. *JOsA*, 66(3):207–211, 1976.

[69] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908, 2014.

[70] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1628–1636, 2016.

[71] Daniel Potts, Gabriele Steidl, and Manfred Tasche. Fast fourier transforms for nonequispaced data: A tutorial. *Modern sampling theory*, pages 247–270, 2001.

[72] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Société Française de Statistique*, 158(1):7–36, 2017.

[73] Klaas P Pruessmann, Markus Weiger, Markus B Scheidegger, and Peter Boesiger. Sense: sensitivity encoding for fast mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 42(5):952–962, 1999.

[74] Wenqi Ren, Xiaochun Cao, Jinshan Pan, Xiaojie Guo, Wangmeng Zuo, and Ming-Hsuan Yang. Image deblurring via enhanced low-rank prior. *IEEE Transactions on Image Processing*, 25(7):3426–3437, 2016.

[75] Nicolai André Brogaard Riis, Yiqiu Dong, and Per Christian Hansen. Computed tomography reconstruction with uncertain view angles by iteratively updated model discrepancy. *Journal of Mathematical Imaging and Vision*, 63(2):133–143, 2021.

[76] Matteo Ronchetti. Torchradon: Fast differentiable routines for computed tomography. *arXiv preprint arXiv:2009.14788*, 2020.

[77] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[78] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557. PMLR, 2019.

[79] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2015.

[80] Yu-hsuan Shih, Garrett Wright, Joakim Andén, Johannes Blaschke, and Alex H Barnett. cufinufft: a load-balanced gpu library for general-purpose nonuniform ffts. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 688–697. IEEE, 2021.

[81] Daniel K Sodickson and Warren J Manning. Simultaneous acquisition of spatial harmonics (smash): fast imaging with radiofrequency coil arrays. *Magnetic resonance in medicine*, 38(4):591–603, 1997.

[82] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

[83] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 769–777, 2015.

[84] Jian Sun, Huibin Li, Zongben Xu, et al. Deep admm-net for compressive sensing mri. *Advances in neural information processing systems*, 29, 2016.

[85] Tijmen Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera neural networks for machine learning. *Mach. Learn*, 6:26–31, 2012.

[86] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[87] S Johanna Vannesjo, Nadine N Graedel, Lars Kasper, Simon Gross, Julia Busch, Maximilian Haeberlin, Christoph Barmet, and Klaas P Pruessmann. Image reconstruction using a gradient impulse response model for trajectory prediction. *Magnetic resonance in medicine*, 76(1):45–58, 2016.

[88] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[89] Ge Wang, Jong Chu Ye, Klaus Mueller, and Jeffrey A Fessler. Image reconstruction is a new frontier of machine learning. *IEEE transactions on medical imaging*, 37(6):1289–1296, 2018.

[90] Guanhua Wang, Tianrui Luo, Jon-Fredrik Nielsen, Douglas C Noll, and Jeffrey A Fessler. B-spline parameterized joint optimization of reconstruction and k-space trajectories (bjork) for accelerated 2d mri. *IEEE Transactions on Medical Imaging*, 2022.

[91] Renke Wang, Roxana Alexandru, and Pier Luigi Dragotti. Perfect reconstruction of classes of non-bandlimited signals from projections with unknown angles. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5877–5881. IEEE, 2022.

[92] Tomer Weiss, Ortal Senouf, Sanketh Vedula, Oleg Michailovich, Michael Zibulevsky, and Alex Bronstein. Pilot: Physics-informed learned optimal trajectories for accelerated mri. *Journal of Machine Learning for Biomedical Imaging*, 2021.

[93] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013.

[94] Ruomei Yan and Ling Shao. Blind image blur estimation via deep learning. *IEEE Transactions on Image Processing*, 25(4):1910–1921, 2016.

[95] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastMRI: An open dataset and benchmarks for accelerated MRI. *arXiv preprint arXiv:1811.08839*, 2018.

[96] Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837, 2018.

[97] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[98] Meina Zhang, Yingying Fang, Guoxi Ni, and Tieyong Zeng. Pixel screening based intermediate correction for blind deblurring. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 01–09, 2022.

[99] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.