Training Adaptive Reconstruction Networks for Blind Inverse Problems

Alban Gossard, Pierre Weiss

Training Adaptive Reconstruction Networks for 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

February 25, 2022

Abstract

Neural networks are full of promises for the resolution of ill-posed inverse problems. In particular, physics informed approaches already progressively replace carefully hand-crafted reconstruction algorithms, for their superior quality. The aim of this paper is twofold. First we show a limitations of these networks: when trained on a given forward operator, they do not generalize well to a different one. Second, we show that training the network with a family of forward operators allows to solve the adaptivity problem without compromising the reconstruction quality significantly. All our experiments are carefully devised on partial Fourier sampling problems arising in magnetic resonance imaging (MRI).

Index terms— Inverse problems, adaptivity, model based reconstruction, convolutional neural network, unrolled networks, MRI reconstruction

1 Introduction

The primary concern of this paper is the design of model-based neural networks to solve families of linear inverse problems. Many sensing devices like cameras, magnetic resonance imaging (MRI) or computerized tomography (CT) measure a signal $\mathbf{x} \in \mathbb{C}^N$ through a linear operator $\mathbf{A}(\boldsymbol{\xi}) \in \mathbb{C}^{M\times N}$. The parameter $\boldsymbol{\xi} \in \mathbb{R}^P$ reflects the fact that the sensing operator may change depending on the imaging conditions. For instance, it could encode the point spread function in image deblurring, the projection angles in CT or the Fourier sampling locations in MRI. This leads to measurements of the form:

$$\mathbf{y} = \mathcal{P}(\mathbf{A}(\boldsymbol{\xi})\mathbf{x}), \tag{1}$$

where $\mathcal{P} : \mathbb{C}^M \to \mathbb{C}^M$ is a perturbation (e.g. additive Gaussian noise, quantization). A model based inverse problem consists in recovering an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ from $\mathbf{y}$ and $\mathbf{A}(\boldsymbol{\xi})$.

Regularization theory From a historical perspective, the first solvers were based on simple inverses or approximate inverses of $\mathbf{A}(\boldsymbol{\xi})$. This approach provides low quality results when the matrix $\mathbf{A}(\boldsymbol{\xi})$ has a non trivial kernel or when the conditioning number of $\mathbf{A}^*(\boldsymbol{\xi})$ 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 (e.g. total variation regularization). This area culminated

with the development of the compressed sensing theory. Starting from 2015, impressive performance gains have occurred with the advent of neural networks. They are likely to replace the initial methods in a growing number of technologies [18].

Learned reconstructions There are two main approaches to attack reconstruction problems using machine learning [4].

A first solution is *end-to-end networks* where the neural network is agnostic to the operator $\mathbf{A}(\boldsymbol{\xi})$. It gets trained through pairs $(\mathbf{y}_i, \mathbf{x}_i)$ generated with the model (1). A popular example is AUTOMAP [23]. In this technology, the network needs to infer the 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. They are often praised for their interpretability. They can be considered as mappings of the form:

$$\begin{aligned} \mathcal{N}: \quad \mathbb{R}^D \times \mathbb{R}^P \times \mathbb{C}^M &\to \mathbb{R}^N \\ (\mathbf{w}, \boldsymbol{\xi}, \mathbf{y}) &\mapsto \mathcal{N}[\mathbf{w}, \boldsymbol{\xi}, \mathbf{y}]. \end{aligned}$$

Given a weight $\mathbf{w} \in \mathbb{R}^D$, a forward operator parametrization $\boldsymbol{\xi}$ and a measurement vector $\mathbf{y}$, the network $\mathcal{N}$ outputs an estimate $\hat{\mathbf{x}}$. The weights $\mathbf{w}$ are trained so as to minimize a quality metric between $\hat{\mathbf{x}}$ and $\mathbf{x}$. A typical choice is the empirical risk $E(\mathbf{w}) \stackrel{\text{def}}{=} \frac{1}{2I} \sum_{i=1}^{I} \|\mathcal{N}[\mathbf{w}, \boldsymbol{\xi}, \mathbf{y}_i] - \mathbf{x}_i\|_2^2$, where $(\mathbf{x}_i)_{1 \leq i \leq I}$ is a collection of training images and $\mathbf{y}_i$ are the corresponding measurements generated using the model 1. Two popular families among this class are:

- *Denoising nets*: There, the reconstruction network performs a crude inversion followed by a denoising network such as a U-Net, to remove the remaining artifacts, see e.g. [11].
- *Unrolled nets*: Many efficient iterative methods have been developed to solve convex optimization problems (forward-backward, Douglas-Rachford, ADMM, ...) [7]. They have the general form:

  $$\mathbf{x}_{k+1} = \text{prox}_R \left(M(\mathbf{A}(\boldsymbol{\xi}), \mathbf{y}, \mathbf{x}_k) \right), \tag{2}$$

  for $k = 1$ to $K \in \mathbb{N}$. The mapping M is linear and can be interpreted as a crude way to invert the operator, in the sense that $\mathbf{A}(\boldsymbol{\xi}) M(\mathbf{A}(\boldsymbol{\xi}), \mathbf{y}, \mathbf{x}_k) \simeq \mathbf{y}$. The term prox_R can be interpreted as a way to regularize (denoise) the remaining artifacts.

  The unrolled networks draw their inspiration from (2). They consist in replacing the human-made proximal operator prox_R by a sequence of neural networks $\mathcal{N}_k[\mathbf{w}_k]$ promoting an output $\mathbf{x}_K$ similar to the images observed during the training phase. Examples of approaches in this category include [1, 2]. These algorithms are currently amongst the best performers for MRI reconstruction [13].

What this work is about We focus on model-based reconstruction networks. While they provide state-of-the art results in a large panel of applications, we showcase a limitation: their *lack of adaptivity*. Let us give a precise meaning to this assertion. Let $\boldsymbol{\xi}_0 \neq \boldsymbol{\xi}_1$ denote two different operator parameterizations. Let $\mathbf{y}_0 = \mathcal{P}(\mathbf{A}(\boldsymbol{\xi}_0)\mathbf{x})$ and $\mathbf{y}_1 = \mathcal{P}(\mathbf{A}(\boldsymbol{\xi}_1)\mathbf{x})$. Assume that $\mathbf{w}_0^\star$ and $\mathbf{w}_1^\star$ are the weights of a reconstruction network optimized for $\mathbf{A}(\boldsymbol{\xi}_0)$ and $\mathbf{A}(\boldsymbol{\xi}_1)$ respectively. The adaptivity issue means that the quality of $\mathcal{N}(\mathbf{w}_1^\star, \boldsymbol{\xi}_0, \mathbf{y}_0)$ is significantly worst than that of $\mathcal{N}(\mathbf{w}_0^\star, \boldsymbol{\xi}_0, \mathbf{y}_0)$. In other words: a network trained for a specific operator $\mathbf{A}(\boldsymbol{\xi}_0)$ may have a significant performance drop if used for another operator $\mathbf{A}(\boldsymbol{\xi}_1)$.

The main idea explored in this paper is simple. We propose to train the network with a whole family of forward operators $\mathcal{A} = \{\mathbf{A}(\boldsymbol{\xi}), \boldsymbol{\xi} \in \Xi\}$, where Ξ is a set of parameters characterizing the device. We evaluate the performance of the resulting network for a specific problem of MRI image

reconstruction from under-sampled data. In that application, the variety of operators stems from various sampling patterns that may differ from one application to the next. We carefully analyze the results, showing that this learning approach yields a reconstruction network which is much more stable to variations of the operator. In particular, the performance of an unrolled network trained on a 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.

Related works Neural network reconstruction can suffer from severe instabilities. This issue was notably discussed in [3], where it was shown that well chosen additive noise (an adversarial attack) or modifications of the forward operator could lead to disastrous hallucinations for some specific architectures. This problem was studied with care in [9]. There, the authors have shown that careful training procedures could fix this issue and yield robust and state-of-the-art reconstruction results.

A paper closely related to our work is [10]. The authors study the same robustness issue to model mismatches. The authors propose two distinct algorithmic approaches to attack it. The first one is called Parametrize & Perturb. 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 will 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 $\mathbf{A}(\boldsymbol{\xi}_0)$, and then use this network in an iterative procedure, accounting for the data consistency term $\|\mathbf{A}(\boldsymbol{\xi}_1)\mathbf{x}-\mathbf{y}\|_2^2$. The approach we propose in this paper is somewhat simpler, since we just train the network with a family of operators.

An older and popular alternative consists in replacing the proximal operator in (2) by a denoiser. This approach is often called a Plug & Play ($P\&P$) prior [17]. It has the huge asset of adapting painlessly to arbitrary inverse problems. We will propose some comparisons and discuss the pros and cons of each approach, see Section 3.

2 The experimental setting

We consider the problem of image reconstruction from under-sampled Fourier samples. This corresponds to imaging modalities such as MRI or radio-interferometry.

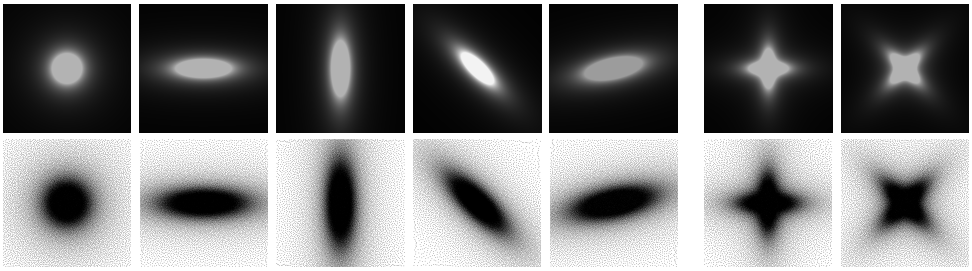
The forward models Let $\mathbf{A}(\boldsymbol{\xi})$ denote the non-uniform Fourier transform (NUFT) [14] defined for all $\mathbf{x}$ by

$$[\mathbf{A}(\boldsymbol{\xi})\mathbf{x}]_m = \sum_{n=1}^{N} x_n e^{-i\langle \mathbf{p}_n, \boldsymbol{\xi}_m\rangle},$$

where $(\mathbf{p}_n)$ is a set of 2D positions on a grid. We used the efficient cuFINUFFT transform [16], which is the fastest available library in our experiments (see `https://github.com/albangossard/Bindings-NUFFT-pytorch` for comparisons). We assume that $\mathbf{y} = \mathbf{A}(\boldsymbol{\xi})\mathbf{x} + \mathbf{b}$, where $\mathbf{b} \sim \mathcal{N}(0, \sigma^2 \mathrm{Id})$ is a simulated additive white complex Gaussian noise.

A restricted sampling family We propose to generate the undersampled sampling schemes following the ideas from [5, 6, 12]. In all experiments, we set $M = 0.25N$, i.e. a 4x downsampling rate. The sampling locations $\boldsymbol{\xi}$ are drawn w.r.t. a target probability measure ρ, by solving

$$\boldsymbol{\xi}(\rho) \overset{\text{def}}{=} \underset{\boldsymbol{\xi}\in\mathbb{R}^{2M}}{\arg\min}\, \mathrm{dist}\left(\frac{1}{M}\sum_{m=1}^{M} \delta_{\boldsymbol{\xi}_m}, \rho\right), \tag{3}$$



(a) ◯ (b) − (c) | (d) (e) (f) + (g) ×

Figure 1: Densities (top) and corresponding sampling schemes (bottom). Fig. 1a, 1b, 1c, 1d and 1e belong to the family Ξ. Fig. 1f and 1g (crosses) do not. Notice that the sampling patterns are diverse with significant differences from one to the other.

where dist is a discrepancy between probability measures. The target densities ρ are anisotropic power decaying distributions, which have been truncated at the center to avoid too dense sampling. They are parameterized by a vector $\boldsymbol{\lambda} \in \Lambda \stackrel{\text{def}}{=} [1,3] \times [0,\pi] \times [1.2,2]$. Its coordinates encode the anisotropy, a rotation angle and an oversampling ratio at the center. The value $\lambda_3 = 1$ means that we sample the center exactly at the Shannon-Nyquist limit. The feasible set of sampling schemes is $\Xi = \{\boldsymbol{\xi}(\rho(\boldsymbol{\lambda})), \boldsymbol{\lambda} \in \Lambda\}$. Examples of densities and sampling patterns $\boldsymbol{\xi}(\rho(\boldsymbol{\lambda}))$ are displayed in Fig. 1a-1e.

The model-based reconstruction networks We consider two networks. The first one denoted $\mathcal{N}^d$ is a *denoising network* of the form

$$\mathcal{N}^d[\mathbf{w}, \boldsymbol{\xi}, \mathbf{y}] = \mathcal{D}[\mathbf{w}, \mathbf{A}(\boldsymbol{\xi})^+(\mathbf{y})],$$

where $\mathcal{D}$ is a deep iterative down-up network (DIDN) [21], with a lower memory footprint than the popular U-Net [15].

The second network is an *unrolled proximal gradient network* with $K = 10$ iterations. Letting $F(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}(\boldsymbol{\xi})\mathbf{x} - \mathbf{y}\|_2^2$, it takes the sequential form:

$$\begin{aligned} \mathbf{x}_0 &= \mathbf{A}(\boldsymbol{\xi})^+\mathbf{y} \\ \mathbf{x}_{k+1} &= \mathcal{D}[\mathbf{w}_k, \mathbf{x}_k - \gamma \nabla F(\mathbf{x}_k)]. \end{aligned}$$

where $\mathcal{D}$ is a DIDN and $\gamma = \frac{1}{\|\mathbf{A}(\boldsymbol{\xi})\|_{2\to 2}^2}$ is a step-size. Hence the weights $\mathbf{w}$ to be trained are $\mathbf{w} = [\mathbf{w}_0, \ldots, \mathbf{w}_{K-1}]$ and they are not shared across iterations. The network is denoted $\mathcal{N}^u : (\mathbf{w}, \boldsymbol{\xi}, \mathbf{y}) \mapsto \mathbf{x}_K$.

The training All the networks are trained by minimizing the average risk:

$$\inf_{\mathbf{w} \in \mathbb{R}^D} \frac{1}{2} \mathbb{E}\|\mathcal{N}[\mathbf{w}, \boldsymbol{\xi}, \mathbf{y}] - \mathbf{x}\|_2^2, \tag{4}$$

where the expectation is taken with respect to the noise $\mathbf{b}$, to the forward models $\boldsymbol{\xi}$ and to the images $\mathbf{x}$.

As for $\boldsymbol{\xi}$, we have pre-computed 1000 sampling patterns by solving (3), where the densities ρ have been generated by picking $\boldsymbol{\lambda}$ uniformly at random in Λ. Fig. 1a-1e show some densities the networks were trained on.

The training database is the fastMRI knee training dataset [22]. It contains 34742 images of size 320×320. All the models were trained using the Adam optimizer in PyTorch. For the experiments, 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 $\mathcal{N}^u$ was trained on 14 epochs with a learning rate of 10^{-4} and an exponential step decay of 0.95 after each epoch. Both trainings took about 24h on an Nvidia V100.

3 Results

All evaluations were performed on the validation set of the fastMRI knee database containing 7135 2D slices.

3.1 Training on fixed sampling schemes

In this section, we trained the two reconstruction networks on 5 different schemes: a radial one ($\bigcirc$, Fig. 1a), an horizontal one ($-$, Fig. 1b) and a vertical one ($|$, Fig. 1c). In addition, we used two crosses, which do not belong to the training family Ξ. The first one is aligned with the axes ($+$, Fig. 1f) and the other one with the diagonals ($\times$, Fig. 1g). In Table 1, we report the average peak signal to noise ratio, on the validation set for the two architectures. Table 1 illustrates various facts listed below.

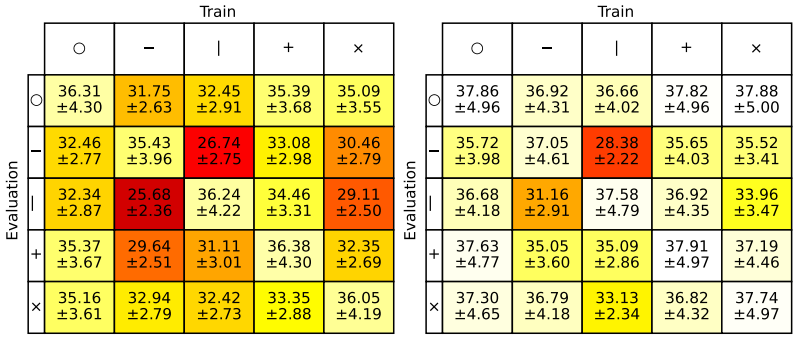
Lack of adaptivity Without surprise, the values on the diagonal are higher than the off-diagonal terms. This means that the best way to reconstruct images for a given scheme is to train the network for this specific scheme. The drop of PSNR when using a network trained with the wrong operator can be as high as 10dB for the denoising net and 9dB for the unrolled net (see the pairs $-$ and $|$). This is a striking illustration of the strong dependency of a reconstruction net to the operator used at the training stage. The corresponding images are shown in Fig. 2.

The superiority of unrolled nets The unrolled network provides better reconstruction results than the denoising net. The overall gain on the diagonal varies between 1dB and 1.7dB for this particular application, which is significant. This is in accordance with recent comparisons of both strategies [13].

Partial adaptivity While the training on the vertical scheme $|$ provides catastrophic results when used on the horizontal scheme $-$, the networks trained on the radial $\bigcirc$ scheme provide rather good results uniformly on the 4 other sampling schemes. Indeed, we see that the performance drop when used on a distinct scheme raises up to 4dB for the denoising net and 'only' 1.3dB for the unrolled network. Overall, the unrolled network provides a significantly better alternative when it comes to adaptivity.

Optimal sampling scheme Some sampling schemes make the reconstruction easier than others, which is in accordance with the compressed sensing theory. The knee images have large vertical and horizontal edges. This seems to favor the $+$ sampling scheme, which was already observed in other works [19, 20].

A simple explanation We focus on the denoising network. The pseudo-inverse $\mathbf{A}(\boldsymbol{\xi})^+$ applied to $\mathbf{y} = \mathbf{A}(\boldsymbol{\xi}) + \mathbf{b}$ yields a vector $\tilde{\mathbf{x}}$ of the form $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{k} + \mathbf{n}$, where $\mathbf{k} \in \ker(\mathbf{A}(\boldsymbol{\xi}))$ and $\mathbf{n}$ is correlated Gaussian noise living in $\mathrm{ran}(\mathbf{A}(\boldsymbol{\xi}))$. Hence the denoising network $\mathcal{D}$ serves two



(a) Denoising net (b) Unrolled net

Table 1: Average PSNR and its standard deviation evaluated on the fastMRI validation dataset for the two architectures.

purposes: 1) recover the missing data $\mathbf{k}$ in the kernel of $\mathbf{A}(\boldsymbol{\xi})$ and 2) remove the correlated noise $\mathbf{n}$. Each of these two tasks is clearly highly dependent on $\boldsymbol{\xi}$. If trained with a single scheme, the network may get specialized very well for these specific statistical patterns and not extrapolate to other ones. It is therefore tempting to see if a training with a larger variety of operators mitigates the performance drop.

3.2 Training on an operator family

In this section, we trained the reconstruction networks by varying the forward operators, as in (4). In what follows, we let ID and IU denote the "ideal" denoising network and the "ideal" unrolled network respectively. 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. We let FD and FU denote the "family" denoising and unrolled networks, which have been trained over a complete family. We also tested the P&P approach. We used an unrolled proximal gradient for $K = 10$ iterations with a DIDN network as an embedded denoiser. It was trained specifically to denoise the images with various levels of white Gaussian noise. Finally, we implemented the R&R network composed of $K = 10$ iterations. The embedded inversion network consists of a pseudo-inverse, followed by a DIDN network trained for the ◯ sampling scheme. The hyperparameter in the method (see [10]) was tuned to produce the best results.

Table 2 shows the performance of the different architectures. The following remarks can be formulated:

Denoising network The denoising network trained on the whole family drops by less than 0.3dB compared to ID, when tested with schemes that belong to the training family.

However, the performance is altered (0.9dB) for the two schemes outside the training family. This suggests that a proper training of a denoising net should cover a sufficiently vast family of operators.

Unrolled networks The unrolled networks trained with a family shows a performance drop of at most 0.2dB compared to IU for schemes inside *and outside* the family. This is a remarkable feature: the scheme extrapolates to some extent outside the training set. This illustrates the take home message of our paper: training unrolled networks on a family does not degrade much the performance while providing adaptivity of the networks.

Evaluation \ Model	ID	FD	IU	FU	P&P	R&R
○	36.31 ±4.30	36.13 ±4.17	37.86 ±4.96	37.84 ±4.95	35.76 ±3.81	35.22 ±3.56
−	35.43 ±3.96	35.08 ±3.74	37.05 ±4.61	36.82 ±4.46	34.36 ±3.43	32.97 ±3.34
|	36.24 ±4.22	35.91 ±4.01	37.58 ±4.79	37.37 ±4.65	35.75 ±3.81	34.23 ±3.28
+	36.38 ±4.30	35.42 ±3.72	37.91 ±4.97	37.72 ±4.82	35.91 ±3.85	34.15 ±3.35
×	36.05 ±4.19	35.23 ±3.61	37.74 ±4.97	37.53 ±4.80	35.22 ±3.63	33.67 ±3.16

Table 2: Average PSNR and standard deviation (in dB) of various reconstruction approaches applied to various operators.

In addition, the price of adaptivity seems affordable for most applications since a 0.2dB loss is marginal.

Plug & Play (P&P) When comparing Table 1 and 2, we see that the P&P approach performs better uniformly than models trained on a single operator. However, its performance drops by $0.5 - 1$dB compared to ID and $1.8 - 2.7$dB compared to IU.

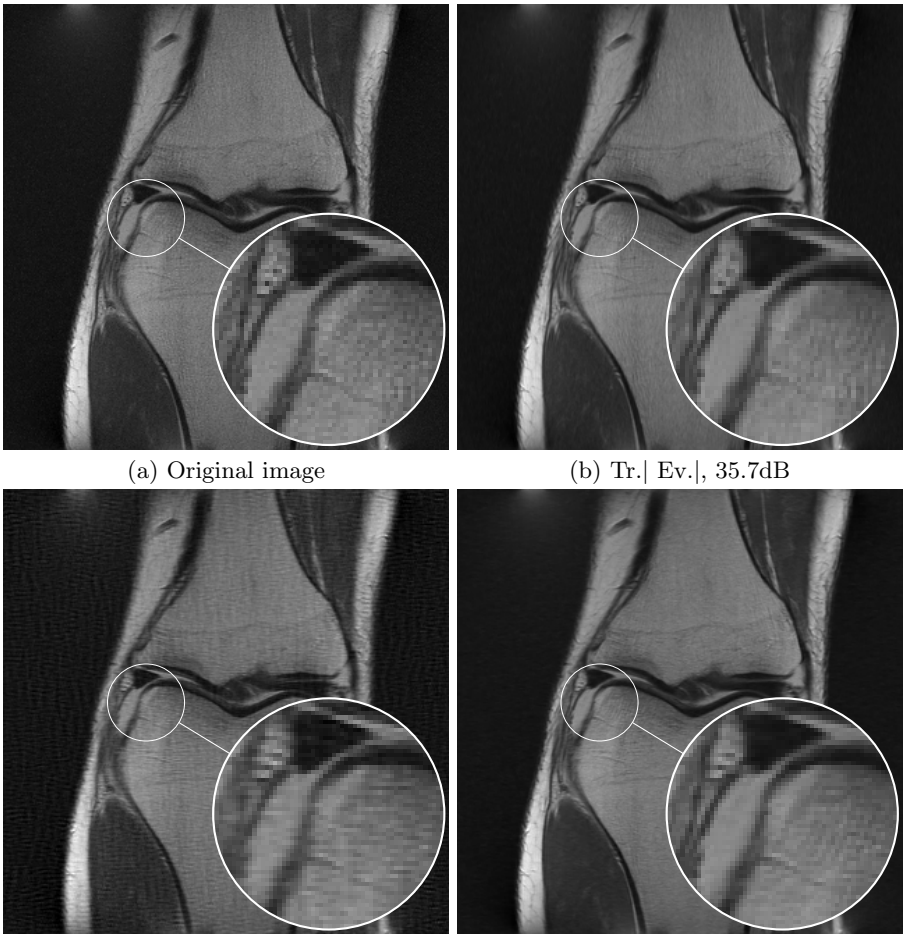
It is also significantly less accurate than FU for an identical computational cost. This suggests that for a given reconstruction architecture, it is beneficial to train the proximal networks for a specific task rather than using a *universal* denoiser, as is the case in P&P.

Notice however, that FU does not 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. There, the P&P approach was considerably more consistent. In a sense, we can see the proposed training as an intermediate step between the plug&play 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 does improve the results by up to 1.9dB compared to a model trained on a single operator. However, it seems that our simpler training approach provides significantly better results. Notice that R&R has a wider scope since it also deals with unknown perturbations of the forward operator (i.e. blind inverse problems), while we only consider the non blind case.

4 Conclusion

In this work we studied the stability of model-based reconstruction networks to variations of the acquisition operator. We first illustrated significant performance drops when training the models on a single forward model. We then demonstrated that a simple solution to mitigate this effect and ensure a good adaptivity is to train the model on a family of operators. This study is preliminary since it was conducted solely on a realistic MRI reconstruction problem. It opens new interesting perspectives for computational imaging. A recent trend consists in optimizing the forward model and the reconstruction algorithm jointly (see e.g. [20, 8, 19] 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.



(a) Original image (b) Tr.| Ev.|, 35.7dB

(c) Tr.| Ev.−, 30.0dB (d) FU Ev.−, 35.1dB

Figure 2: Examples of reconstructions using the unrolled network. We trained it on | (2b, 2c) and on a family (2d). We tested it on | (2b) and − (2c, 2d).

Acknowledgement

This work was performed using HPC resources from GENCI-IDRIS (Grant 2021-AD011012210R1). P. Weiss acknowledges a support from ANR-3IA Artificial and Natural Intelligence Toulouse Institute and ANR Micro-Blind.

References

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

[2] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

[8] Frédéric de Gournay, Alban Gossard, and Pierre Weiss. Spurious minimizers in non uniform fourier sampling optimization. 2021.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[16] 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.

[17] 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.

[18] 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.

[19] 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. *arXiv preprint arXiv:2101.11369*, 2021.

[20] 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.

[21] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up CNN for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[22] 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.

[23] 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.