



**HAL**  
open science

# Advanced deep neural networks for MRI image reconstruction from highly undersampled data in challenging acquisition settings

Zaccharie Ramzi

► **To cite this version:**

Zaccharie Ramzi. Advanced deep neural networks for MRI image reconstruction from highly undersampled data in challenging acquisition settings. Medical Imaging. Université Paris-Saclay, 2022. English. NNT: 2022UPAST025 . tel-03623570

**HAL Id: tel-03623570**

**<https://theses.hal.science/tel-03623570v1>**

Submitted on 29 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Advanced deep neural networks for MRI  
image reconstruction from highly  
undersampled data in challenging  
acquisition settings

*Réseaux de neurones profonds avancés pour la  
reconstruction d'images IRM à partir de données  
fortement sous-échantillonnées dans des contextes  
d'acquisition complexes*

**Thèse de doctorat de l'université Paris-Saclay**

École doctorale n° 575, Electrical, Optical, Bio : physics and Engineering  
(EOBE)

Spécialité de doctorat : Imagerie et physique médicale

Graduate School : Sciences de l'ingénierie et des systèmes, Référent :  
Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **BAOBAB (Université Paris-Saclay,  
CEA, CNRS)**, sous la direction de **Philippe CIUCIU**, directeur de recherche, la  
co-direction de **Jean-Luc STARCK**, directeur de recherche

**Thèse soutenue à Paris-Saclay, le 18 février 2022, par**

**Zaccharie RAMZI**

**Composition du jury**

**Gabriel PEYRÉ**

Directeur de recherche, CNRS, Paris

**Carola-Bibiane SCHÖNLIEB**

Professeure, University of Cambridge

**Rebecca WILLETT**

Professeure, University of Chicago

**Émilie CHOUZENOUX**

Chargée de recherche, Inria Saclay

**Philippe CIUCIU**

Directeur de recherche, CEA - NeuroSpin

Président

Rapporteuse & Examinatrice

Rapporteuse & Examinatrice

Examinatrice

Directeur de thèse

**Titre:** Réseaux de neurones profonds avancés pour la reconstruction d'images IRM à partir de données fortement sous-échantillonnées dans des contextes d'acquisition complexes

**Mots clés:** réseaux neuronaux, échantillonnage compressif, IRM, problèmes inverses, apprentissage profond, reconstruction d'image

**Résumé:** L'imagerie par résonance magnétique (IRM) est l'une des modalités d'imagerie les plus utilisées au monde. Son objectif principal est de visualiser les tissus mous de manière non invasive et non ionisante. Cependant, son adoption générale est entravée par une durée d'examen globalement élevée. Afin de la raccourcir, plusieurs techniques ont été proposées, parmi lesquelles l'imagerie parallèle (PI) et l'échantillonnage compressif (CS) jouent une place prédominante. Grâce à ces techniques, les données en IRM peuvent être acquises de manière fortement compressée, réduisant ainsi significativement le temps d'acquisition. Cependant, les algorithmes généralement utilisés pour reconstruire les images IRM à partir de ces données sous-échantillonnées sont lents et peu performants dans des scénarios d'acquisition fortement accélérés.

Afin de résoudre ces problèmes, les "réseaux de neurones déroulés" ont été introduits. L'idée centrale de ces modèles est de dérouler ou déplier les itérations des algorithmes de reconstruction classiques en un graphe de calcul fini.

L'objectif principal de cette thèse est de proposer de nouvelles architectures pour des scénarios d'acquisition qui s'écartent de l'acquisition cartésienne 2D typique. À cette fin, nous passons d'abord en revue une poignée de réseaux neuronaux pour la reconstruction IRM. Après avoir sélectionné le plus performant, i.e. le PDNet, nous l'étendons à deux contextes : le challenge fastMRI 2020 et le problème des données 3D non cartésiennes. Nous avons également choisi de répondre aux préoccupations de beaucoup concernant l'applicabilité clinique de l'apprentissage profond pour l'imagerie médicale. Nous le faisons en proposant des moyens de construire des modèles robustes et inspectables, mais aussi en testant simplement les réseaux entraînés dans des contextes qui s'écartent de la distribution d'entraînement. Enfin, après avoir remarqué comment l'outil de l'apprentissage profond implicite peut aider à entraîner des modèles de reconstruction IRM plus profonds, nous introduisons une nouvelle méthode d'accélération (i.e. SHINE) pour l'entraînement de ces modèles.

**Title:** Advanced deep neural networks for MRI image reconstruction from highly undersampled data in challenging acquisition settings

**Keywords:** neural networks, compressed sensing, MRI, inverse problems, deep learning, image reconstruction

**Abstract:** Magnetic Resonance Imaging (MRI) is one of the most prominent imaging techniques in the world. Its main purpose is to probe soft tissues in a non-invasive and non-ionizing way. However, its wider adoption is hindered by an overall high scan time. In order to reduce this duration, several approaches have been proposed, among which Parallel Imaging (PI) and Compressed Sensing (CS) are the most important. Using these techniques, MR data can be acquired in a highly compressed way which allows the reduction of acquisition times. However, the algorithms typically used to reconstruct the MR images from these undersampled data are slow and underperform in highly accelerated scenarios.

In order to address these issues, unrolled neural networks have been introduced. The core idea of these models is to unroll the iterations of classical reconstruction algorithms into a finite computation

graph.

The main objective of this PhD thesis is to propose new architecture designs for acquisition scenarios which deviate from the typical Cartesian 2D sampling. To this end, we first review a handful of neural networks for MRI reconstruction. After selecting the best performer, the PDNet, we extend it to two contexts: the fastMRI 2020 reconstruction challenge and the 3D non-Cartesian data problem. We also chose to address the concerns of many regarding the clinical applicability of deep learning for medical imaging. We do so by proposing ways to build robust and inspectable models, but also by simply testing the trained networks in out-of-distribution settings. Finally, after noticing how the implicit deep learning framework can help implement deeper MRI reconstruction models, we introduce a new acceleration method (called SHINE) for the training of such models.



*Knowledge is in the end based on  
acknowledgement.*  
— Ludwig J. F. Wittgenstein,  
1889 - 1951

---

## Acknowledgements

I have always believed that the intelligence that stems from collective and critical thinking surpasses the one that can be achieved alone. It is therefore necessary for me to value the contributions that many people had in my thesis. Moreover, I felt that these contributions were made in a safe and benevolent environment, which helped me a lot to focus on the science and relax at times.

Firstly, I would like to thank my two directors, **Philippe Ciuciu** and **Jean-Luc Starck**, for their support during my thesis. Both of them handled our relationship with a lot of trust, valued my work and treated me as an equal when it came to research questions. Philippe, in particular, has taken a lot of the administrative burden on his shoulders, like he does for all his students, which made my life and my research so much easier.

Secondly, I would like to thank all the members of my jury especially the reviewers Prof. **Carola-Bibiane Schönlieb** and Prof. **Rebecca Willett** for agreeing to read and expertise my work. I extend my thanks to Prof. **Gabriel Peyré** and Dr. **Emilie Chouzenoux** for their participation in my jury.

Thirdly, I want to say how much I enjoyed being part of the Parietal team. It was an honor to be among so many brilliant people, always fostering a diverse and insightful research environment. I would like to particularly thank **Thomas Moreau** for the fruitful collaborations we had, our elongated debugging sessions and helping me grow as a scientist. *Mens sana in corpore sano*; it has been a pleasure to swim with **Jérôme-Alexis Chevalier** and **Thomas Bazeille**, and of course much more with many others, like **Hamza, Hugo, Pierre, Mathurin, Maëli, Antonia, Patricio, Olivier, David, Jérôme, Kamalakar, Binh** and many more.

Most of my time was spent at NeuroSpin with my colleagues from the CS-MRI team. I have had a wonderful time here, and it started by being welcome by **Loubna El Gueddari** who showed me all the ropes and gave me a few scientific briefing on MRI. I am also deeply appreciative of the interactions I had with **Chaithya Navada** (a.k.a. **Chaithya G R**), and his lively attitude. I am wishing good luck to the newcomers (and not so new) **Zaineb Amor, Guillaume Daval-Frétot**, and **Pierre-Antoine Comby**.

Talking of NeuroSpin, it was also my pleasure to interact with the Metric team. Our discussions helped frame my thesis the real world, and I am grateful to **Alexandre Vignaud** for asking the right and difficult questions (is GRAPPA dead?). I was also helped in my day-to-day tasks, for example when **Franck Mauconduit** helped me understand the inner workings of the ISMRMRD format, or when **Redouane Jamil** became an advocate for Restaurant 1.

Of course, I could not finish on the NeuroSpin faithfuls, without saying how much I enjoyed creating a Deep Learning lecture group with **Louise Guillon, Benoît Dufumier** and **Alexandros Popov**. I am really proud of what we managed to create, and also thankful for the great discussions and presentations we had from all the people who attended.

My thesis's second home, was just next door on the plateau, the Cosmostat team. I was always very happy to come there, not only for the different cafeteria, but also for all the people who were there. In particular, I would like to thank **François Lanusse** for his invaluable knowledge, of which he managed to communicate a part with me. I have a very special thought for **Santiago Casas** who enlightened me with the joy of the navette, my life has not been the same since. I also really valued the discussions and laughs I had with **Samuel Farrens**. Of course, it was my pleasure to meet Albert with **Morgan Schmitz, Fadi Nammour, Tobias Liadat, Virginia Ajani, Benjamin Rémy**, and more who I forget. I also grew up as a professional thanks to my supervision of **Sophie Starck** and **Kevin Michalewicz** who I had the pleasure of working with for 2 and 6 months, respectively.

Finally, I would like to extend my sincere thanks to my friends and my family. I am super lucky to have such a loving support in my life. I will not name all my friends because I have too many, they will know I am talking about them. However, I want to highlight the incredible support I received from my family, especially during the difficult lockdowns we had to suffer. I want to thank **Léa** for being my best friend, family and such an important part of my life.



# Contents

|   |           |
|---|-----------|
| General Introduction                                      | 11        |
| <b>I Context</b>  | <b>17</b> |
| <b>1 Introduction to Magnetic Resonance Imaging</b>       | <b>19</b> |
| 1.1 Motivations for the MRI modality                      | 21        |
| 1.1.1 Context   | 21        |
| 1.1.2 Diagnosis   | 21        |
| 1.1.3 Non-invasiveness and absence of radiation           | 23        |
| 1.1.4 Resolution  | 23        |
| 1.2 Physics of MRI  | 23        |
| 1.2.1 Nuclear Magnetic Resonance                          | 23        |
| 1.2.2 Image Formation                                     | 25        |
| 1.2.3 Parallel Imaging                                    | 27        |
| 1.3 Acceleration in MRI                                   | 27        |
| 1.3.1 Motivations   | 27        |
| 1.3.2 Tools   | 28        |
| 1.3.3 Limitations   | 29        |
| <b>2 Classical Reconstruction in MRI</b>                  | <b>31</b> |
| 2.1 Introduction to Compressed Sensing for MRI            | 32        |
| 2.1.1 Linear Underdetermined Inverse Problems             | 32        |
| 2.1.2 Guarantees of recovery                              | 32        |
| 2.1.3 Application to MRI                                  | 34        |
| 2.2 Sparse reconstruction algorithms                      | 35        |
| 2.2.1 Classical algorithms                                | 35        |
| 2.2.2 Dictionary learning                                 | 37        |
| 2.3 Parallel Imaging Reconstruction                       | 37        |
| 2.3.1 Image-domain techniques                             | 38        |
| 2.3.2 Frequency-domain techniques                         | 39        |
| 2.3.3 Combination with CS techniques                      | 40        |
| 2.4 Quantitative evaluation of the reconstruction         | 41        |
| 2.4.1 Classical quantitative metrics                      | 42        |
| 2.4.2 Advanced metrics                                    | 42        |
| 2.4.3 Specificities of MRI evaluation                     | 43        |
| 2.4.4 Discussion on the relevance of quantitative metrics | 43        |

|          |   |           |
|----------|---|-----------|
| <b>3</b> | <b>Introduction to Deep Learning</b>                      | <b>45</b> |
| 3.1      | Timeline of Deep Learning . . . . .                       | 46        |
| 3.2      | The base ingredients of Deep Learning . . . . .           | 46        |
| 3.2.1    | Formalism . . . . .                                       | 46        |
| 3.2.2    | Backpropagation with the chain rule . . . . .             | 48        |
| 3.2.3    | SGDs . . . . .  | 49        |
| 3.2.4    | Computing power of GPUs . . . . .                         | 50        |
| 3.2.5    | Big Data . . . . .  | 50        |
| 3.3      | Classical Architectural Blocks of Deep Learning . . . . . | 52        |
| 3.3.1    | Perceptron . . . . .                                      | 52        |
| 3.3.2    | Nonlinearities . . . . .                                  | 52        |
| 3.3.3    | Convolutions . . . . .                                    | 54        |
| 3.3.4    | Pooling . . . . .   | 54        |
| 3.3.5    | Normalization . . . . .                                   | 55        |
| 3.3.6    | Residual/skip connections . . . . .                       | 56        |
| 3.3.7    | Attention . . . . .                                       | 56        |
| 3.3.8    | Dropout . . . . .   | 56        |

## **II Methodological Developments 59**

|          |   |           |
|----------|---|-----------|
| <b>4</b> | <b>Review of Deep Learning for MRI reconstruction</b>           | <b>61</b> |
| 4.1      | Paradigms for deep learning use in MRI reconstruction . . . . . | 62        |
| 4.1.1    | Plug-and-Play . . . . .   | 62        |
| 4.1.2    | Agnostic learning . . . . .                                     | 63        |
| 4.1.3    | Single-domain restoration . . . . .                             | 63        |
| 4.1.4    | Adversarial reconstruction . . . . .                            | 64        |
| 4.1.5    | Deep Compressed Sensing . . . . .                               | 64        |
| 4.1.6    | Deep Image Prior . . . . .                                      | 64        |
| 4.1.7    | Self-supervised . . . . .                                       | 65        |
| 4.1.8    | Implicit field learning . . . . .                               | 65        |
| 4.2      | Benchmarking unrolled networks for MRI reconstruction . . . . . | 66        |
| 4.2.1    | Introduction . . . . .  | 66        |
| 4.2.2    | Related works . . . . .   | 67        |
| 4.2.3    | Models . . . . .  | 67        |
| 4.2.4    | Data . . . . .  | 72        |
| 4.2.5    | Results . . . . .   | 73        |
| 4.2.6    | Discussion . . . . .  | 76        |
| 4.3      | Unrolled networks for MRI reconstruction . . . . .              | 77        |
| 4.3.1    | Model-based Deep Learning . . . . .                             | 77        |
| 4.3.2    | Variational Network . . . . .                                   | 77        |
| 4.3.3    | $\Sigma$ -Net . . . . .   | 78        |
| 4.3.4    | End-to-end VarNet . . . . .                                     | 78        |

|          |  |            |
|----------|--|------------|
| 4.3.5    | Neumann Network . . . . .  | 79         |
| <b>5</b> | <b>New unrolled networks for MRI reconstruction</b>  | <b>81</b>  |
| 5.1      | XPDNet . . . . .   | 82         |
| 5.1.1    | Introduction . . . . .   | 82         |
| 5.1.2    | Model . . . . .  | 82         |
| 5.1.3    | Results . . . . .  | 83         |
| 5.1.4    | Conclusion and Discussion . . . . .  | 84         |
| 5.1.5    | Figures . . . . .  | 84         |
| 5.2      | NC-PDNet . . . . .   | 85         |
| 5.2.1    | Introduction . . . . .   | 85         |
| 5.2.2    | Related Works . . . . .  | 86         |
| 5.2.3    | Model . . . . .  | 88         |
| 5.2.4    | Data . . . . .   | 89         |
| 5.2.5    | Results . . . . .  | 90         |
| 5.2.6    | Discussion and conclusion . . . . .  | 113        |
| <b>6</b> | <b>Clinical applicability of deep learning for MRI reconstruction</b>  | <b>115</b> |
| 6.1      | Learnlets . . . . .  | 116        |
| 6.1.1    | Introduction . . . . .   | 116        |
| 6.1.2    | Related Work . . . . .   | 117        |
| 6.1.3    | Learnlets, the model . . . . .   | 117        |
| 6.1.4    | Exact reconstruction . . . . .   | 122        |
| 6.1.5    | Data and Experiments . . . . .   | 122        |
| 6.1.6    | Results . . . . .  | 124        |
| 6.1.7    | Conclusions . . . . .  | 129        |
| 6.2      | Denosing Score-Matching for Uncertainty Quantification in Inverse Problems . . . . .                             | 130        |
| 6.2.1    | Introduction . . . . .   | 130        |
| 6.2.2    | Related Works . . . . .  | 131        |
| 6.2.3    | Deep DSM for Posterior Inference . . . . .   | 132        |
| 6.2.4    | Application to Bayesian Inverse Problems . . . . .   | 133        |
| 6.2.5    | Conclusions and Discussions . . . . .  | 134        |
| 6.3      | Is good old GRAPPA dead? . . . . .   | 135        |
| 6.3.1    | Introduction . . . . .   | 135        |
| 6.3.2    | Methods . . . . .  | 135        |
| 6.3.3    | Results . . . . .  | 136        |
| 6.3.4    | Conclusion and Discussion . . . . .  | 143        |
| <b>7</b> | <b>New learning paradigms for very deep networks</b>   | <b>145</b> |
| 7.1      | SHINE: SHaring the INverse Estimate from the forward pass for bilevel optimization and implicit models . . . . . | 146        |
| 7.1.1    | Introduction . . . . .   | 146        |
| 7.1.2    | Hypergradient Optimization with Approximate Jacobian Inverse . . . . .   | 148        |

|  |  |            |
|--|--|------------|
| 7.1.3                                      | Results  | 154        |
| 7.1.4                                      | Conclusion and Discussion  | 158        |
| 7.2  | Other paradigms for memory reduction when training neural networks | 159        |
| 7.2.1                                      | Gradient checkpointing   | 159        |
| 7.2.2                                      | Invertible Networks  | 159        |
| 7.2.3                                      | IFT-based networks   | 161        |
| <b>General Conclusion and Perspectives</b> |  | <b>163</b> |
| Contributions & Limitations                |  | 163        |
| Perspectives                               |  | 164        |
| <b>A</b>                                   | <b>Additional results</b>  | <b>169</b> |
| A.0.1                                      | Regularized Nonlinear Least Squares                                | 169        |
| <b>B</b>                                   | <b>Training details</b>  | <b>183</b> |
| <b>C</b>                                   | <b>Proofs for SHINE</b>  | <b>187</b> |
| <b>D</b>                                   | <b>Software</b>  | <b>193</b> |
| <b>E</b>                                   | <b>Tutorials, documentation and courses</b>                        | <b>195</b> |
| <b>F</b>                                   | <b>Ideas we tried and did not work</b>                             | <b>197</b> |
| <b>G</b>                                   | <b>Résumé en français (Abstract in French)</b>                     | <b>199</b> |

# General Introduction

## Context & Motivations

**M**AGNETIC Resonance Imaging (MRI) is one of the most important and widely used medical imaging modalities. It can probe soft tissues like the brain, the heart or the knee, but also provide insights on the functional organization of the brain or the layout of its vessels. Moreover, it can do so in a non-invasive and non-radiative way contrarily to for example [Computed Tomography \(CT\)](#) or [Positron Emission Tomography \(PET\)](#). However, MRI at its core is based on an inherently slow physical phenomenon: the magnetic resonance and the corresponding spin relaxation. This translates into long acquisition times and high exam duration, which hinder the wider adoption of MRI as a global imaging technique. Indeed, a high exam duration restricts the use of MRI in certain contexts but also limit the number of patients that can be examined per day. In order to address these limitations, several techniques have been proposed to reduce the number of excitation-relaxation sequences we need. The two main ones are [Parallel Imaging \(PI\)](#) and [Compressed Sensing \(CS\)](#), and they both rely on acquiring an undersampled version of the object to image. In PI, an array of receiver coils is used to gather the [Radio Frequency \(RF\)](#) signals emitted during the relaxation. In CS, which can be combined with PI, some structural prior knowledge is assumed about the object to be reconstructed, and we use it to build a variational formulation whose solution is computed by an optimization algorithm to retrieve an accurate view of the object. These two approaches have allowed the physicians to reduce the scan times, however they have also introduced new challenges. Indeed, the reconstruction of the object is quite slow when using typical CS reconstruction algorithms. In addition, PI and CS are limited to low acceleration factors because they use very basic and hand-crafted knowledge about the object of interest, typically in CS, the sparsity of the decomposed image in a wavelet basis or frame.

Concurrently, [Deep Learning \(DL\)](#) was developed to address challenges related to classification and regression. But more generally, DL has also been demonstrated to be a powerful tool for general function approximation and learning, in particular in imaging. As such, one could use DL to solve the MR image reconstruction problem from undersampled measurements by reformulating it as a *supervised learning problem*. Originally, convolution based networks were proposed to tackle this problem without incorporating physics-based or model-based knowledge into their design. Nonetheless, more recently DL has been used to learn the formerly handcrafted structure prior on the object, in particular in the framework of unrolled neural networks.

The goal of this PhD thesis was to design original DL architectures by unrolling optimization algorithms and learning their parameters in service to challenging



acquisition settings that fit the clinical practice. Additionally, we were interested in demonstrating the performances (accuracy and robustness) of these models in a clinical setting.

## Contributions

The first task in this thesis was to identify ideal candidates for the base of our unrolled network. To this end, we benchmarked several existing unrolled networks on 2 large databases and found consistently that one architecture, the *PDNet* was outperforming its competitors. A side effect of this benchmark was the creation of a GitHub repository gathering the implementations of all these networks, as well as training and evaluation codes.

This benchmark was done on a single-coil 2D Cartesian dataset which is not a clinically relevant setup. A more challenging acquisition scenario, closer to the clinical practice is that of multicoil 2D Cartesian imaging. This is exactly the setup of the 2020 fastMRI challenge we decided to participate in. To do so, we first built an extended version of the *PDNet* which integrated state-of-art building blocks. We termed this network the *XPDNet*, and we secured the second place in the challenge, showing that even in clinically relevant setups, our network was still promising. To further our developments and address the challenge of isotropic high resolution imaging from massively undersampled data, we developed the *NC-PDNet*, an architecture that is capable of handling 3D non-Cartesian k-space data.

This setting is particularly relevant for NeuroSpin, the lab where I effectuated most of my PhD thesis, as the SPARKLING (Spreading Projection Algorithm for Rapid K-space sampLING) technology was developed here for accelerating high resolution anatomical and functional brain imaging at ultra-high magnetic field (7 T and beyond). Great efforts are actually deployed for understanding the best non-Cartesian undersampling schemes.

Importantly, we tested the robustness of these networks in out-of-distribution settings, a potential barrier to the adoption of neural networks in the clinical realm. For the *NC-PDNet*, we made sure to test the network on unseen organs, acquisition trajectories or acceleration factors, and showed that it performed well in these configurations. Our evaluation of the *XPDNet* was focused on the comparison with [Generalized Autocalibrating Partially Parallel Acquisitions \(GRAPPA\)](#), a classical *PI* reconstruction algorithm that is implemented by the vendors in most of [MR](#) systems. This comparison stood in the context of prospective undersampling, a way more complex scenario given the potential additional artifacts related to actual acceleration. Indeed, we mostly trained and evaluated our networks with retrospectively undersampled data, which does not present the same characteristics as prospectively undersampled data. All these evaluations were done in the spirit of testing the clinical applicability of the unrolled networks for [MRI](#) reconstruction. Another aspect of this thesis in that direction is to provide tools to build robust and

inspectable networks. The Learnlets, a type of learned wavelets, was modeled in order to provide practitioners with a tool to build robust and generalizable networks. Moreover, we showed how the [Denoising Score Matching \(DSM\)](#) framework could be used to obtain a sense of the reconstruction error in DL-based reconstructions.

Another barrier to the use of unrolled networks in clinical settings is their weak performance for very high acceleration factors or undersampling factors.<sup>1</sup> As is often the case in DL, we have very good reasons to believe that deeper networks will enable reaching sufficient levels of performance even at very high acceleration factors. When training deep neural networks, the main memory bottleneck usually is the activations (i.e. the stored intermediary outputs of the networks' building blocks) needed to compute the gradients. Therefore, the number of activations and their size are the main obstacle to training even deeper neural networks. To overcome this issue, some DL frameworks introduced schemes that allow the training of neural networks without relying on activations. As such they constitute a promising direction for solving inverse problems, notably MRI reconstruction. However, training such models is more expensive from a computational viewpoint. Hence, we proposed a method to accelerate some of these models, [Deep Equilibrium Network \(DEQs\)](#) part of the implicit DL framework.

## Thesis Outline

Chapter 1: *Introduction to Magnetic Resonance Imaging* introduces the main concepts and motivations for MRI. We focus on how the physical system can be designed to generate the signal of interest.

Chapter 2: *Classical Reconstruction in MRI* presents the main ways to reconstruct the object to image in the undersampled context.

Chapter 3: *Introduction to Deep Learning* reviews briefly the main components of the success of DL and how it is applied in practice.

Chapter 4: *Review of Deep Learning for MRI reconstruction* lists and discusses the principal ways in which DL can be applied to MRI reconstruction. We focus on unrolled neural networks. Contributions:

- Benchmark of unrolled networks for MRI reconstruction.

Chapter 5: *New unrolled networks for MRI reconstruction* presents 2 new unrolled neural networks for different acquisition settings: the *XPDNet* and the *NC-PDNet*. Contributions:

---

<sup>1</sup>The wording acceleration factor is more used for prospective acquisitions while the term undersampling factor often refers to retrospective studies. Both coincide in Cartesian acquisition scenarios whereas they may differ in non-Cartesian ones.

- *XPDNet*: a reconstruction model for 2D multicoil brain data, ranking second in the fastMRI challenge.
- *NC-PDNet*: a reconstruction model for non-Cartesian data.

Chapter 6: *Clinical applicability of deep learning for MRI reconstruction* takes a look at how to build robust and inspectable networks fit for clinical use, and additionally evaluates how the trained neural networks measure their clinical applicability. Contributions:

- *Learnlets*: a wavelet-inspired network.
- Application of *DSM* to *MRI* reconstruction for uncertainty quantification.
- Comparison of the *XPDNet* with *GRAPPA*, and application to prospective data.

Chapter 7: *New learning paradigms for very deep networks* lists different paradigms allowing the training of very deep neural networks. We focus on implicit deep learning, and *DEQs* in particular where we show how the training of these models can be accelerated. Contributions:

- *SHINE*: a method to accelerate the training of *DEQs* and bilevel optimization problems.

Open source contributions:

- *fastmri-reproducible-benchmark*: a repository with different unrolled networks for *MRI* reconstruction.
- *PySAP*: a library to perform sparse signal reconstruction.
- *tfkbnufft*: a library for the Non-Uniform Fast Fourier Transform in TensorFlow.

# Publications

## Accepted articles in Peer-Reviewed Journals

- M. J. Muckley, B. Riemenschneider, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha, S. Arberet, D. Nickel, **Zaccharie Ramzi**, P. Ciuciu, J. L. Starck, J. Teuwen, D. Karkalousos, C. Zhang, A. Sriram, Z. Huang, N. Yakubova, Y. W. Lui and F. Knoll. “Results of the 2020 fastMRI Challenge for Machine Learning MR Image Reconstruction”. In: *IEEE Transactions on Medical Imaging* 40.9 (2021), pp. 2306–2317
- **Zaccharie Ramzi**, P. Ciuciu and J. L. Starck. “Benchmarking MRI reconstruction neural networks on large public datasets”. In: *Applied Sciences (Switzerland)* 10.5 (2020)
- S. Farrens, A. Grigis, L. El Gueddari, **Zaccharie Ramzi**, C. G R, S. Starck, B. Sarthou, H. Cherkaoui, P. Ciuciu and J.-L. Starck. “PySAP: Python Sparse Data Analysis Package for multidisciplinary image processing”. In: *Astronomy and Computing* 32 (2020)

## Submitted articles in Peer-Reviewed Journals

- **Zaccharie Ramzi**, K. Michalewicz, J. L. Starck, T. Moreau and P. Ciuciu. “Wavelets in the deep learning era”. 2021. Under review in *Journal of Mathematical Imaging and Vision*
- **Zaccharie Ramzi**, C. G R, J.-L. Starck and P. Ciuciu. “NC-PDNet: a Density-Compensated Unrolled Network for 2D and 3D non-Cartesian MRI Reconstruction”. In: *IEEE Transactions on Medical Imaging* (2022)

## Accepted papers in Peer-reviewed Conferences

- C. G R, **Zaccharie Ramzi** and P. Ciuciu. “Learning the sampling density in 2D SPARKLING MRI acquisition for optimized image reconstruction”. In: *European Signal Processing Conference*. 2021
- **Zaccharie Ramzi**, J. L. Starck and P. Ciuciu. “Density compensated unrolled networks for non-cartesian MRI reconstruction”. In: *Proceedings - International Symposium on Biomedical Imaging*. Vol. 2021-April. 2021, pp. 1443–1447
- **Zaccharie Ramzi**, J. L. Starck, T. Moreau and P. Ciuciu. “Wavelets in the deep learning era”. In: *European Signal Processing Conference*. Vol. 2021-Janua. 2021, pp. 1417–1421. Oral

- **Zaccharie Ramzi**, P. Ciuciu and J. L. Starck. “Benchmarking Deep Nets MRI Reconstruction Models on the Fastmri Publicly Available Dataset”. In: *Proceedings - International Symposium on Biomedical Imaging*. Vol. 2020-April. 2020, pp. 1441–1445

### Submitted papers in Peer-reviewed Conferences

- **Zaccharie Ramzi**, F. Mannel, S. Bai, J.-L. Starck, P. Ciuciu and T. Moreau. “SHINE: SHaring the INverse Estimate from the forward pass for bi-level optimization and implicit models”. In: *International Conference on Learning Representations*. 2022. Spotlight
- G. Chaithya, **Zaccharie Ramzi** and P. Ciuciu. “Hybrid learning of Non-Cartesian k-space trajectory and MR image reconstruction networks”. 2021
- K. Pooja, **Zaccharie Ramzi**, C. G R and P. Ciuciu. “MC-PDNET: Deep unrolled neural network for multi-contrast mr image reconstruction from undersampled k-space data”. Oct. 2021

### Abstracts in Peer-reviewed Conferences/ Papers in Workshops

- **Zaccharie Ramzi**, P. Ciuciu and J.-L. Starck. “XPDNet for MRI Reconstruction: an application to the 2020 fastMRI challenge”. In: *ISMRM*. 2020, pp. 1–4. Oral
- **Zaccharie Ramzi**, A. Vignaud, J.-L. Starck and P. Ciuciu. “Is good old GRAPPA dead?” In: *ISMRM*. 2021
- B. Riemenschneider, M. Muckley, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha, S. Arberet, D. Nickel, **Zaccharie Ramzi**, P. Ciuciu, J. L. Starck, J. Teuwen, D. Karkalousos, C. Zhang, A. Sriram, Z. Huang, N. Yakubova, Y. W. Lui and F. Knoll. “Results of the 2020 fastMRI Brain Reconstruction Challenge”. In: *ISMRM*. 2021. Oral
- **Zaccharie Ramzi**, B. Remy, F. Lanassee, J.-L. Starck and P. Ciuciu. “De-noising Score-Matching for Uncertainty Quantification in Inverse Problems”. In: *NeurIPS 2020 Deep Learning and Inverse Problems workshop*. 2020
- B. Remy, F. Lanassee, **Zaccharie Ramzi**, J. Liu, N. Jeffrey and J.-L. Starck. “Probabilistic Mapping of Dark Matter by Neural Score Matching”. In: *NeurIPS 2020 Machine Learning for Physical sciences workshop*. 1. 2020, pp. 1–6
- **Zaccharie Ramzi**, P. Ciuciu, J.-L. Starck and J.-L. Starck Benchmarking. “Benchmarking proximal methods acceleration enhancements for CS-acquired MR image analysis reconstruction”. In: *SPARS 2019 - Signal Processing with Adaptive Sparse Structured Representations Workshop*. 2019

# **Part I**

## **Context**



# 1 - Introduction to Magnetic Resonance Imaging

## Chapter Outline

|       |   |    |
|-------|---|----|
| 1.1   | Motivations for the MRI modality . . . . .          | 21 |
| 1.1.1 | Context . . . . .                                   | 21 |
| 1.1.2 | Diagnosis . . . . .                                 | 21 |
| 1.1.3 | Non-invasiveness and absence of radiation . . . . . | 23 |
| 1.1.4 | Resolution . . . . .                                | 23 |
| 1.2   | Physics of MRI . . . . .                            | 23 |
| 1.2.1 | Nuclear Magnetic Resonance . . . . .                | 23 |
| 1.2.2 | Image Formation . . . . .                           | 25 |
| 1.2.3 | Parallel Imaging . . . . .                          | 27 |
| 1.3   | Acceleration in MRI . . . . .                       | 27 |
| 1.3.1 | Motivations . . . . .                               | 27 |
| 1.3.2 | Tools . . . . .                                     | 28 |
| 1.3.3 | Limitations . . . . .                               | 29 |

**M**RI is one of the most widely used techniques to probe the human body in a non-invasive and non-ionizing way. The expected outcome of the exam is to be able to visually diagnose diseases, anomalies or ill-formations. To do so, multiple contrasts of the same organ of interest are generated, however the images are not obtained directly via a photographic procedure. Rather, we use the phenomenon of magnetic resonance to create a signal emanating from the object. The beauty of MRI is then to spatially encode this signal during the acquisition stage in the so-called k-space. The design of the signal generation is such that it can be inverted to obtain the image (potentially in 3D) of the organ. Indeed, in an ideal case without noise or artifacts, the k-space is simply the Fourier transform of the organ. However, the generation of this signal, relying on the relaxation of excited spins, is inherently slow. It is therefore an important research goal to be able to accelerate the MRI acquisition process by limiting the number of required relaxation steps to obtain a clean image.

In this chapter, we will cover first the motivations of MRI and describe its current use. We will move on to explain the underlying physical concepts that enable MRI. Finally, we will describe the acquisition side of acceleration.

An example MR image is presented in [Figure 1.0-1](#).

## 1.1 . Motivations for the MRI modality





Figure 1.0-1: **Example of an MR image:** MR image of the knee taken from the fastMRI dataset [Zbo+18].

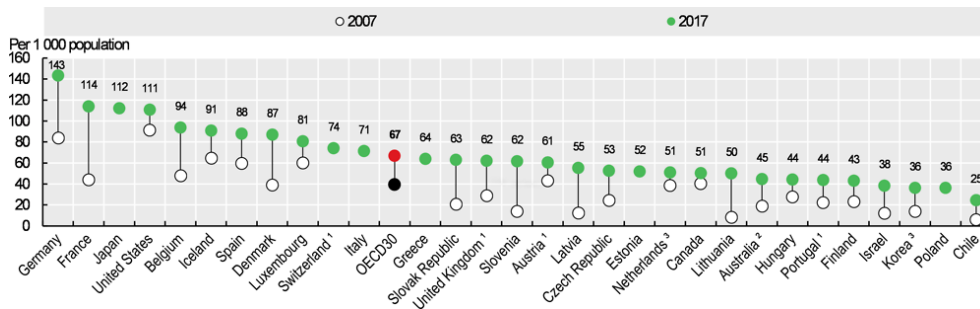


Figure 1.1-2: **Number of MRI scans per year per 1000 population:** figure courtesy of *Health at a Glance 2019: OECD Indicators - Medical technologies* [19d].

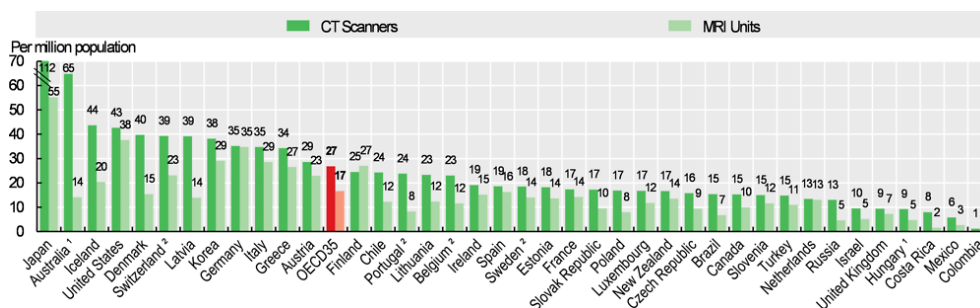


Figure 1.1-3: **Number of MRI and CT machines per year per million population:** figure courtesy of *Health at a Glance 2019: OECD Indicators - Medical technologies* [19d].

### 1.1.1 . Context

The use of MRI is growing rapidly around the world. In order to understand how impactful improving the MRI modality can be, let us give some contextual figures. In France for example, we went from 40 to 114 MRI scans each year per 1000 population between 2007 and 2017, more than doubling [19d]. As a comparison, there are about 2 to 3 times more Computed Tomography (CT) scans per year [19d]. An overview of the worldwide situation can be seen in Figure 1.1-2. The number of MRI scanners is estimated at 36,000 worldwide [Ogb+18]. However, the distribution of MRI scanners is very uneven around the world and developing countries are the most ill-equipped [Ogb+18] as illustrated in Figure 1.1-3.

### 1.1.2 . Diagnosis

MRI is currently used for many diagnosis tasks, in organs and body regions such as the brain, the spine, the neck, the musculoskeletal system, the abdomen,<sup>1</sup>

<sup>1</sup>We also specify that the abdomen encompasses organs such as the spleen, the liver, the biliary system, the pancreas, the kidneys, the GI tract, the bowel and the upper urinary tract.

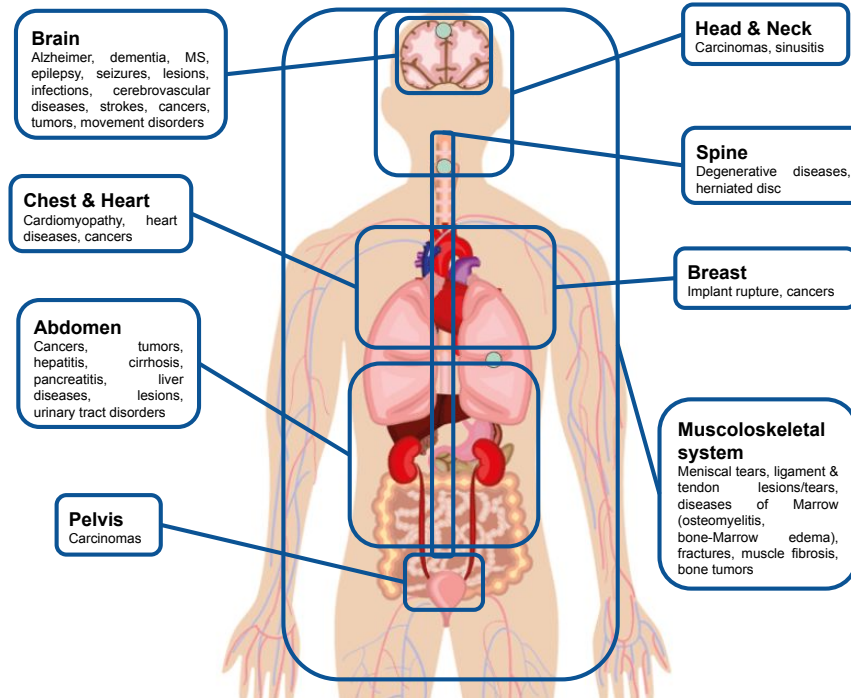


Figure 1.1-4: **What can we diagnose with MRI?** This illustration provides a non-exhaustive list of all the diagnoses that can be carried out with MRI. All the information was compiled from the works of Reimer et al. [Rei+10] and Runge et al. [RTH19].

the pelvis, the chest, the heart and the breast. In this section we provide a non-exhaustive list of all the diagnosis that can currently be performed with MRI in the form of an original image in Figure 1.1-4.

Importantly, not all these diagnoses are primary diagnoses, some of them like cancer or brain tumors can help guide biopsy or invasive surgery more precisely. Some other may rely on modality which are not exactly anatomical MRI: for example the movement disorders diagnosis can be done from Functional MRI (fMRI) of the brain [NH10a]. More information on how these medical investigations are carried out can be found in the works of Reimer et al. [Rei+10] and Runge et al. [RTH19].

For these diagnoses, MRI is not the sole imaging technique available. The typical competitor is CT whose main advantage is to be quicker than MRI. However, in many instances, MRI has been shown to be more sensitive to anomalies and diseases than CT [KM00; Kid+04] and also more accurate in its depiction of said diseases or anomalies [Pat+88].

### 1.1.3 . Non-invasiveness and absence of radiation

**MRI** is typically referred to as a non-invasive exam, which means that the skin or the mucosa is not violated [12a] (except the potential catheter used to inject the contrast agent if any). This is a very suitable property for an imaging modality which makes it more amenable to everyday clinical use, since it can also be used in-vivo. In addition, as opposed to **CT**, it does not make use of radiations to perform the probing of soft tissues which enables its repeated use without dangers [Hol+14]. Although there is a slight increase in temperature potentially harmful to pregnant women [Le +21], the only real danger in **MRI** comes from the magnet which can attract metallic objects implanted in the body.

#### 1.1.4 . Resolution

**MRI** is also favored for its high resolution. It recently reached resolutions of 0.2 mm isotropic [Stu+15], and the promise of scanners delivering a magnetic field of 11.7 **Teslas (T)** opens the door to even higher resolutions.<sup>2</sup> Higher resolution enables several things:

- finer biomarkers and medical diagnosis;
- better structural and functional connectivity mapping in the brain [Now18];
- better pre-surgery mapping [Now18].

## 1.2 . Physics of MRI

The goal of this section is to provide a brief overview on how the **MR** signal gets created. In order to get a deeper understanding, we refer the reader to the following resources which we took inspiration from when writing this section:

- [mriquestions.com](http://mriquestions.com) [EB01]: an online course presented in the form of an FAQ;
- [imaios.com/en/e-Courses/e-MRI](http://imaios.com/en/e-Courses/e-MRI) [08]: an online course with a lot of explanatory videos;
- Bernstein et al. [BKZ04b] and Brown et al. [Bro+14]: classical **MRI** handbooks (the latter one being usually known as Haacke et al. 1999);
- the dissertations of former PhD students who pursued their thesis at NeuroSpin in the same team: Lazarus [Laz18], El Gueddari [El 19].

### 1.2.1 . Nuclear Magnetic Resonance

The phenomenon of **Nuclear Magnetic Resonance (NMR)** is at the core of **MRI**. A hydrogen<sup>3</sup> atom possesses magnetic properties, one of which is its spin,

---

<sup>2</sup>See this press release: [www.cea.fr/english/Pages/News/premieres-images-irm-iseult-2021.aspx](http://www.cea.fr/english/Pages/News/premieres-images-irm-iseult-2021.aspx)

<sup>3</sup>Other atoms can be used in a research setting like Carbon, Sodium, Phosphorus, etc.

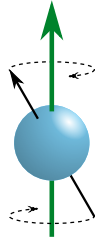


Figure 1.2-5: **Illustration of the precession of a spin in a magnetic field:** the green arrow represents the  $B_0$  magnetic field, while the black arrow represents the magnetic moment of the particle. Illustration courtesy of *Larmor precession Wikipedia page* [12b].

the magnetic moment of the nucleus. A first key property of the spin is that when submitted to an outside magnetic field  $B_0 = B_0 e_z$ , it aligns with it in a parallel or antiparallel way. It has a rotation movement around the magnetic axis characterized by the Larmor frequency  $\omega_0 = \gamma B_0$ , with  $\gamma$  the hydrogen gyrometric ratio. Typical values of  $\omega_0$  are in the **Radio Frequency (RF)** band, i.e. in the MHz range. This rotation movement forms a cone called a precession as can be seen in [Figure 1.2-5](#).

The resonance phenomenon corresponds to the interaction between the rotating spins and an incoming **RF** pulse, denoted  $B_1$ , orthogonal to  $B_0$ . This interaction can only happen if the **RF** pulse has the same frequency as the Larmor frequency  $\omega_0$ . When the **RF** pulse interacts with a spin, it will create an excitation of the spin. In effect, it will increase the precession of the spin by bringing energy to the system and tip it into the transversal plan, orthogonal to  $B_0$ . The amount of increase of the precession will depend on the intensity of the **RF** pulse and its duration.

After the excitation, the spin will enter the relaxation phase. In this phase, the additional energy the spin received during the excitation will be reemitted back in the shape of an **RF** pulse. This **RF** pulse, called the **Free Induction Decay (FID)** can be recorded using an antenna. The characteristics of the **FID** depend on the nature of the tissue where the spin is located. Therefore, we can leverage the **FID** to form an image of the organ because different tissues will emit **RF** pulses with different characteristics.

Formally, denoting  $M$  the total magnetic moment,  $M_0$  its equilibrium state,  $M_{tr}$  its transverse component and  $M_l$  its longitudinal component, the Bloch equations ruling the evolution of the system are:

$$\frac{dM_{tr}}{dt} = -\frac{M_{tr}}{T_2} \quad (1.1)$$

$$\frac{dM_l}{dt} = \frac{M_0 - M_l}{T_1} \quad (1.2)$$

with  $T_1$  and  $T_2$  the 2 characteristic times for the relaxation, whose values depend

on the nature of the tissue.

The solution of this equation, for a position in space  $\mathbf{r}$ , is:

$$\mathbf{M}_{tr}(t, \mathbf{r}) = \mathbf{M}_{tr}(0, \mathbf{r}) e^{-\frac{t}{T_2}} \quad (1.3)$$

$$\mathbf{M}_l(t, \mathbf{r}) = \mathbf{M}_l(0, \mathbf{r}) e^{-\frac{t}{T_1}} + \mathbf{M}_0(1 - e^{-\frac{t}{T_1}}) \quad (1.4)$$

where in particular,  $|\mathbf{M}_{tr}(0, \mathbf{r})| = \frac{1}{4}\rho(\mathbf{r})\frac{\gamma^2\hbar^2}{kT}B_0$  with  $\hbar$  the Planck constant,  $k$  the Boltzmann constant,  $T$  the temperature and  $\rho(\mathbf{r})$  the proton spin density [Bro+14].

### 1.2.2 . Image Formation

However, one can only record a single global FID signal without localization information if no extra adjustments of the experiment are used. In order to encode the spatial information, spatially varying gradients of magnetic field are used. These gradients make the intensity of the magnetic field vary slightly across a certain direction. Three types of gradients, whose total contribution is denoted  $\mathbf{G}(t)$ , exist and allow different kinds of spatial encoding:

- **Slice Selection Gradient:** It allows us to excite only the spins rotating at a certain Larmor frequency band (in practice it can never be a single frequency) by tuning the RF pulse frequency. This frequency band is directly related to the slice thickness.
- **Phase Encoding Gradient:** It allows us to dephase the FID from the different spins even after its application by shortly modifying the frequency of rotation. Temporally speaking, this phase encoding gradient is used after the excitation and the slice selection gradient but prior to data acquisition. Since the dephasing depends on the intensity of the gradient, we can repeat its application with different intensities to obtain different combined dephasing.
- **Frequency Encoding Gradient:** It modifies the frequency of rotation of the spins during the recording and is applied during the signal readout, so after the previous ones and during the opening of the analog-to-digital converter.

For a given slice, the MRI acquisition corresponds to the sequence of sending the same RF pulse sequence with different phase encoding and frequency encoding gradients. In the case of a 3D acquisition, a second phase encoding gradient can be used in the third direction. All of this encoding will naturally lead to a Fourier encoding of the spatial information.

Because  $|\mathbf{M}_{tr}(t, \mathbf{r})|$  is proportional to the spin density  $\rho(\mathbf{r})$ , we would like to get access to this quantity, but the latter cannot be measured directly. Instead, we have access to the electromagnetic force induced by  $\mathbf{M}$  (of compact support  $V_s$ ) in an antenna:

$$S(t) = -\frac{d}{dt} \int_{V_s} \mathbf{B}_1 \cdot \mathbf{M}(t, \mathbf{r}) d\mathbf{r} \quad (1.5)$$

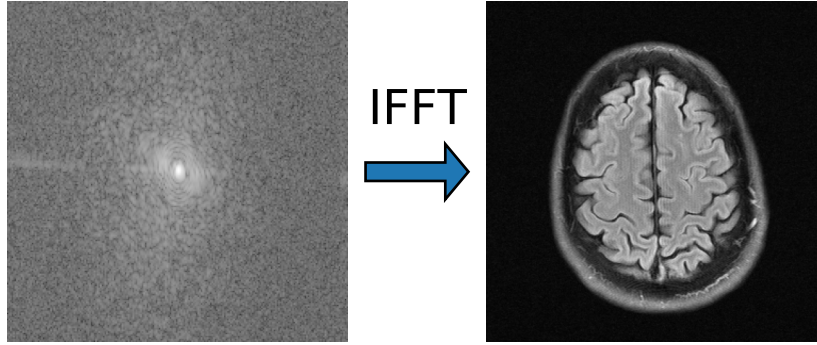


Figure 1.2-6: **Example of a k-space with its corresponding anatomical image:** The raw data is from the fastMRI dataset [Zbo+18]. The k-space is in log-scale and only the magnitude of the 2 images are represented. We selected only a single coil from the 16 coils available for illustrative purposes.

After neglecting low-magnitude derivatives, and focusing on the sinusoidal part by ignoring the relaxation effect, we obtain a simplified MR signal:

$$S(t) \propto \omega_0 \int_{V_s} |\mathbf{B}_{tr}| |M_{tr}(t, \mathbf{r})| \sin\left(\omega_0 t + \gamma \int_0^t \mathbf{G}(\tau) d\tau + \phi_0(\mathbf{r}) - \phi_B(\mathbf{r})\right) d\mathbf{r} \quad (1.6)$$

where  $\mathbf{B}_{tr}$  and  $M_{tr}(t, \mathbf{r})$  are the transverse components of  $\mathbf{B}_1$  and  $\mathbf{M}(t, \mathbf{r})$  respectively,  $\phi_0(\mathbf{r})$  is the transverse magnetization phase at  $t = 0$ , and  $\phi_B(\mathbf{r})$  is the phase of  $\mathbf{B}_{tr}$ .

From there, we can demodulate the signal to get rid of the effect of  $\omega_0 t$ , by multiplying by the appropriate sine and cosine waves at  $\omega_0$  frequency, and using a low-pass filter. The result of these 2 multiplications will form the real and the imaginary parts of the MR signal:

$$S_{tr}(t) \propto \omega_0 \int_{V_s} B_{tr} M_{tr}(t, \mathbf{r}) e^{-i\gamma \mathbf{r} \cdot \int_0^t \mathbf{G}(\tau) d\tau} d\mathbf{r} \quad (1.7)$$

where  $B_{tr} = |\mathbf{B}_{tr}| e^{i\phi_B(\mathbf{r})}$  and  $M_{tr}(t, \mathbf{r}) = |M_{tr}(t, \mathbf{r})| e^{-i\phi_0(\mathbf{r})}$ . Here, the Fourier encoding is apparent as the first part of the integral is proportional to the density of the organ and the term  $\mathbf{k}(t) = \frac{\gamma}{2\pi} \int_0^t \mathbf{G}(\tau) d\tau$  in the complex exponential is controlled by the operator. We refer to  $\mathbf{k}(t)$  as the k-space vector. An example k-space, i.e. the map  $\mathbf{k}(t) \mapsto S_{tr}(t)$ , is given in Figure 1.2-6 along with its corresponding anatomical image. The trajectory defined by  $\mathbf{k}(t)$  over time is called the k-space trajectory. This trajectory, piloted by the operator of the MR exam via the gradient sequence  $\mathbf{G}(t)$ , will define how we traverse the Fourier Transform (FT) space of the organ of interest. A first natural choice is of course to follow

lines on a Cartesian grid and sample also on the grid in order to be able to use the simple [Inverse Fast Fourier Transform \(IFFT\)](#) on the complex MR signal  $S_{tr}(t)$  to obtain the image of the organ. However, this trajectory might have downsides in some situations, for example it implies long scan times and might not be suited to undersampling. Therefore, one might consider other trajectories that do not fall on a Cartesian grid like radial spokes or spiral interleaves. Some trajectories might even be designed or learned in order to satisfy some criterion with respect to the reconstruction of the image in an undersampled setting [[Laz+19](#)]. If one uses a non-Cartesian acquisition scheme, then  $S_{tr}(t)$  is not the FT of the organ anymore, but the [Nonuniform Discrete Fourier Transform \(NDFT\)](#). As a consequence, the [IFFT](#) cannot be used anymore and one has to resort to more complex schemes to invert the forward process as the [NDFT](#) is not invertible. These schemes can be iterative or rely on gridding the data to make the [IFFT](#) applicable.

### 1.2.3 . Parallel Imaging

In order to improve the [Signal-to-Noise Ratio \(SNR\)](#), multiple receiver antennas, also called coils, can be used to receive the MR signal. In effect, they will not record the same signal but one that is dependent on their position with respect to the organ of interest. Because we have extra signal for the same organ, it is possible to sample less lines in the k-space than with a single coil. In the [Parallel Imaging \(PI\)](#) acquisition setup one has to resort to coil combination strategies to obtain the final image. As discussed in the next part, this step may be complexified in presence of undersampling due to inherent aliasing artifacts.

## 1.3 . Acceleration in MRI

### 1.3.1 . Motivations

An [MRI](#) exam might last up to 90 minutes according to *NHS: How it's performed - MRI scan* [[18](#)], and 15 minutes in general. This is unpractical for many reasons:

- **Accessibility:** Some patients, for example young children or people suffering from Parkinson's disease, may not be able to stay still for such an extended period of time. Moreover, patients suffering from claustrophobia could also experience an anxiety crisis. General anesthesia is a solution, but it is a heavy process with additional risks involved.
- **Patient throughput:** A long exam time means that only a reduced number of patients can undergo an [MRI](#) per day. As an example, without acceleration, the [Koyasu Neurosurgical Clinic \(KNC\)](#) was able to perform on average 26.8 patients per day with extra hours on a single [MRI](#) machine [[KS18](#)]. This has two consequences. The first is that [MRI](#) scans are prescribed less often than they should because of a long waiting line and as there needs to be



spots left for emergencies. The second is that the cost of the MRI exam per patient is high: sometimes reaching \$4000 in the United States [19a].

- **Motion:** Patient motion during the MRI acquisition is one of the primary sources of artifacts. However, the probability of motion occurring during the acquisition is not uniformly distributed in time; in particular, it is more likely to happen after a long time in the scanner. Reducing the amount of time spent in the scanner therefore diminishes the probability to suffer from patient-motion-related artifacts.

The MRI exam does not only consist of the actual acquisition. Three phases can be distinguished before the medical diagnosis. The first is the preparation phase where the patient needs to be installed in the MRI scan. This can be long especially for patients with reduced mobility. The second is the acquisition in itself, where the patient lays still in the MRI scanner, and the MR pulse sequences are run. Most often, several sequences are carried out to get the most accurate diagnosis, for instance for brain imaging: T2-weighted imaging, Fluid Attenuated Inversion Recovery (FLAIR), susceptibility weighted imaging and Diffusion-Weighted Imaging (DWI). Taken together these different imaging contrasts provide complementary information on the potential pathology. The third is the image generation or reconstruction, where the actual images used for diagnoses are generated from the MR signal. In the case of an unaccelerated exam, the image generation phase is almost instantaneous, while the preparation takes 5 minutes on average and the acquisition 15 minutes [KS18]. This shows that the acquisition is really the main bottleneck in an unaccelerated MRI exam.

### 1.3.2 . Tools

In order to accelerate the MRI acquisition, there are not many physical leverages. Indeed, the relaxation time is driven by the molecular properties of the tissue (and a little bit by the field strength) which cannot be modified. The T1 relaxation time is for example in the orders of 1 to 4 seconds for aqueous tissues [Bus+11, Chapter 12]. The only physical leverage we can use is the field strength.<sup>4</sup> Indeed, at higher field strengths, the SNR is very high, and we can therefore reduce the scan time for the same resolution [Spr+16].

Lowering the resolution is also a way to accelerate MRI even at a given field strength, although it is not always desirable depending on the task.<sup>5</sup> Some other methods exist that accelerate the MRI acquisition in a straight forward way. The first is partial Fourier sampling. Because the organ of interest has a real-valued image, in theory, its Fourier coefficients have a conjugate Hermitian symmetry [EB01].

---

<sup>4</sup>This is not due to the relaxation time which increases with the field strength for T1 [Bus+11, Chapter 12] and decreases for T2 [EB01].

<sup>5</sup>It is however possible to use a superresolution algorithm/model in order to increase the resolution of the image, but we will not cover this aspect extensively, see [this Siemens press release](#).

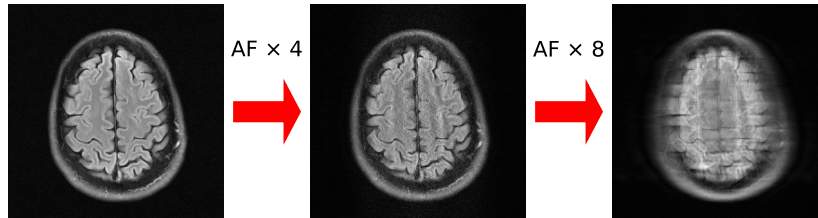


Figure 1.3-7: **Reconstruction in PI at different undersampling factors:** an MR image of a brain reconstructed using GRAPPA [Gri+02], a common reconstruction technique, for different Acceleration Factors (AFs) in 2D multicoil imaging. The raw data is from the fast-MRI dataset [Zbo+18]. The AF of 4 is still readable while the AF of 8 is unpractical.

This symmetry could allow us to sample only half of the k-space and estimate the other half. In practice however, imperfections in the acquisition system prevent us from using this technique as is, and extra acquisition is required, or partial Fourier is applied only to a fraction of the k-space (e.g. 6/8 of data collected).

Finally, in PI, it is possible to use the redundancy of the acquired signal across the multiple coils to undersample the k-space.

### 1.3.3 . Limitations

However, all of these techniques suffer from limitations. For example partial Fourier can only accelerate up to a factor of 1.6, while undersampling with PI can be done up to a factor of 4 in 2D and 10 in 3D. At the core of this problem lies an information problem. Basically one needs to have enough information about the organ under investigation contained in the signal you acquire. Formally in the single-coil case, for a general complex-valued signal, the Nyquist criterion applies. The Nyquist-Shannon theorem is as follows:

**Theorem 1.3.1.** *A band-limited continuous-time signal can be sampled and perfectly reconstructed from its discrete samples if the waveform is sampled at least twice as fast as its highest frequency component.*

If the Nyquist criterion is not met, one might face aliasing artifacts in the generated image that prevent its proper evaluation. An example of these artifacts can be seen in Figure 1.3-7.



## 2 - Classical Reconstruction in MRI

### Chapter Outline

|       |   |    |
|-------|---|----|
| 2.1   | Introduction to Compressed Sensing for MRI . . . . .          | 32 |
| 2.1.1 | Linear Underdetermined Inverse Problems . . . . .             | 32 |
| 2.1.2 | Guarantees of recovery . . . . .                              | 32 |
| 2.1.3 | Application to MRI . . . . .                                  | 34 |
| 2.2   | Sparse reconstruction algorithms . . . . .                    | 35 |
| 2.2.1 | Classical algorithms . . . . .                                | 35 |
| 2.2.2 | Dictionary learning . . . . .                                 | 37 |
| 2.3   | Parallel Imaging Reconstruction . . . . .                     | 37 |
| 2.3.1 | Image-domain techniques . . . . .                             | 38 |
| 2.3.2 | Frequency-domain techniques . . . . .                         | 39 |
| 2.3.3 | Combination with CS techniques . . . . .                      | 40 |
| 2.4   | Quantitative evaluation of the reconstruction . . . . .       | 41 |
| 2.4.1 | Classical quantitative metrics . . . . .                      | 42 |
| 2.4.2 | Advanced metrics . . . . .                                    | 42 |
| 2.4.3 | Specificities of MRI evaluation . . . . .                     | 43 |
| 2.4.4 | Discussion on the relevance of quantitative metrics . . . . . | 43 |

WHEN acceleration is used to collect k-space data faster, via data undersampling, the image formation cannot be performed by simply applying the IFFT. One needs to resort to more complex methods, most of which were developed before the advent of Deep Learning (DL). These methods must leverage the sampled data as well as some knowledge on the underlying problem to generate an image which should ideally be indistinguishable from that computed without undersampling.

In this chapter, we will first introduce Compressed Sensing (CS) [CRT06; Don06], a framework which leverages the compressibility of natural images (and thus medical images like MRI scans) in an appropriate domain, push forward pseudo-random undersampling as a potential means to reach incoherent sampling and make use of nonsmooth convex optimization to perform nonlinear MR image reconstruction from undersampled data. We will then detail how this framework can be used to reconstruct MR images. Further, for comparison and/or possible combination with CS we will introduce more standard methods specific to PI reconstruction. Finally, we will explain how the reconstruction algorithms can be evaluated.

### 2.1 . Introduction to Compressed Sensing for MRI

CS is a mathematical theory allowing the design of algorithms enabling the reconstruction of a signal from noisy undersampled measurements, as well as the design of efficient measurement strategies. The core idea is to leverage the sparsity or the compressibility of the signal in an appropriate domain/basis to decrease the sampling rate far below the Nyquist bound.

For a deeper understanding of CS, we recommend the following resource: Foucart et al. [FR13a]. Most of the material in this section is based on this book.

### 2.1.1 . Linear Underdetermined Inverse Problems

Formally, CS is concerned with tackling the following type of problems: We aim to recover a signal  $\mathbf{x} \in \mathbb{C}^n$  from linear measurements  $\mathbf{y} \in \mathbb{C}^m$  corrupted by some noise  $\epsilon$ .<sup>1</sup> The problem reads:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon \quad (2.1)$$

where  $\mathbf{A} \in \mathbb{C}^{m \times n}$  is called the design matrix, the measurement operator or the forward operator. A simpler noiseless problem can be considered in a simulated setting for instance:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad (2.2)$$

When  $\text{Ker}(\mathbf{A}) = \{0\}$ , the noiseless problem (2.2) is fully determined and can be solved by applying the inverse or the pseudo-inverse of  $\mathbf{A}$  to  $\mathbf{y}$ ,  $\mathbf{x} = \mathbf{A}^\dagger \mathbf{y}$ . On the contrary, when  $\text{Ker}(\mathbf{A}) \neq \{0\}$ , the noiseless problem is underdetermined, and there exists multiple  $\mathbf{x}$  which satisfy Equation 2.2. Indeed, when a solution  $\mathbf{x}_1$  is found,  $\mathbf{x}_1 + \mathbf{x}_0$  where  $\mathbf{x}_0 \in \text{Ker}(\mathbf{A})$  is also a solution to the noiseless problem (2.2). This situation happens when  $m < n$  but can also occur in other situations when  $m \geq n$ , for example in PI. In this case, it is necessary to discriminate between all the possible solutions of the problem by assuming prior knowledge on the solution. This prior knowledge is often encompassed in the form of redundancy or sparsity of the solution in a certain decomposition basis.

### 2.1.2 . Guarantees of recovery

The question we can ask is: "What are the assumptions needed on  $\mathbf{x}$ ,  $\mathbf{A}$  and  $\epsilon$  to guarantee the recovery of the signal  $\mathbf{x}$  or at least a good enough approximation  $\hat{\mathbf{x}}$ ?" Indeed, it is easy to come up with examples in which the signal cannot be recovered: for example, if  $\mathbf{A} = 0$ , then all the information from  $\mathbf{x}$  is destroyed, and none of it can be found in  $\mathbf{y} = \epsilon$ .

Let us first define the notion of sparsity.

**Definition 2.1.1** (*s*-sparse vectors [FR13c]). *A vector  $\mathbf{x} \in \mathbb{C}^n$  is called *s*-sparse if it contains at most *s* non-zero entries.*

---

<sup>1</sup>The distribution of  $\epsilon$  is an entire problem of itself, but it is very often assumed to be i.i.d. zero-mean Gaussian of unknown variance  $\sigma^2$ .

From there, we can lay out the conditions that guarantee the recovery of an  $s$ -sparse vector  $\mathbf{x}$  in Equation 2.2. Let us first start with the following lemma:

**Lemma 2.1.2** (Reformulation of sparse vector recovery [FR13c]). *For a given sparsity  $s$ , and  $s$ -sparse vector  $\mathbf{x}$ :*

(a) *The vector  $\mathbf{x}$  is the unique  $s$ -sparse solution of Equation 2.2, that is  $\{z \in \mathbb{C}^n : \mathbf{A}z = \mathbf{A}\mathbf{x}, \|z\|_0 \leq s\} = \{\mathbf{x}\}$*

(b) *The vector  $\mathbf{x}$  can be reconstructed as the unique solution of:*

$$\min_{z \in \mathbb{C}^n} \|z\|_0 \quad \text{subject to} \quad \mathbf{A}z = \mathbf{y} \quad (2.3)$$

In other words, Lemma 2.1.2 allows us to transform a linear inverse problem for sparse vectors admitting a unique solution into an optimization problem defined by Equation 2.3. We therefore have a tool (we will see later however that it is unpractical) to solve Equation 2.2 if there is a unique solution. Let us now give a theorem which guarantees the uniqueness of the solution for a given sparsity:

**Theorem 2.1.3** ([FR13c, Theorem 2.13]). *The following properties are equivalent:*

(a) *Every  $s$ -sparse vector  $\mathbf{x} \in \mathbb{C}^n$  is the unique  $s$ -sparse solution of  $\mathbf{A}z = \mathbf{A}\mathbf{x}$ , that is, if  $\mathbf{A}\mathbf{x} = \mathbf{A}z$  and both  $\mathbf{x}$  and  $z$  are  $s$ -sparse, then  $\mathbf{x} = z$ .*

(b) *The null space  $\text{Ker}(\mathbf{A})$  does not contain any  $2s$ -sparse vector other than the zero.*

(c) *Every set of  $2s$  columns of  $\mathbf{A}$  is linearly independent.*

Theorem 2.1.3 hands us assumptions needed on  $\mathbf{x}$  (sparsity) and  $\mathbf{A}$  (at least twice more independent measurements than the level of sparsity of  $\mathbf{x}$ ) to guarantee recovery. This is however unsatisfactory in practice, because the resolution of Equation 2.3 is NP-hard [FR13c, Theorem 2.17]. However, this is mostly due to the very general nature of a potential algorithm solving the problem for all  $\mathbf{A}$  and  $\mathbf{x}$ , therefore by refining even more our assumptions, we can hope to achieve tractable recovery guarantees.

In particular, if we consider the following relaxation of Equation 2.3, also called Basis Pursuit:

$$\min_{\mathbf{x} \in \mathbb{C}^n} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y} \quad (2.4)$$

we can find assumptions on  $\mathbf{A}$  that will guarantee the recovery of all sparse vectors  $\mathbf{x}$  when solving it. These assumptions involve the concept of coherence, whose definition we recall next:

**Definition 2.1.4** (Coherence [FR13b, Definition 5.2]). Let  $\mathbf{A} \in \mathbb{C}^{m \times n}$  be a matrix with  $\ell_2$ -normalized columns  $\mathbf{a}_1, \dots, \mathbf{a}_n$ . The  $\ell_1$ -coherence function  $\mu_1$  is defined for  $s \leq n - 1$  by:

$$\mu_1(\mathbf{A}, s) = \max_{i \leq n} \left\{ \sum_{j \in S} |\langle \mathbf{a}_i, \mathbf{a}_j \rangle|, S \subset \{1, \dots, n\}, |S| = s, i \notin S \right\}$$

Armed with this definition, we can give the following recovery guarantee theorem:

**Theorem 2.1.5.** Let  $\mathbf{A} \in \mathbb{C}^{m \times n}$  have  $\ell_2$ -normalized columns. If:

$$\mu_1(\mathbf{A}, s) + \mu_1(\mathbf{A}, s - 1) < 1 \quad (2.5)$$

then every  $s$ -sparse vector  $\mathbf{x} \in \mathbb{C}^n$  is exactly recovered from the measurements  $\mathbf{y} = \mathbf{A}\mathbf{x}$  via basis pursuit.

Because the resolution of basis pursuit is possible in polynomial time, [Theorem 2.1.5](#) provides a practical way to have recovery guarantees when designing a matrix  $\mathbf{A}$ .

### 2.1.3 . Application to MRI

The work of Lustig et al. [LDP07] was seminal in the development of [CS](#) for [MRI](#). In particular, the incoherence needed in [CS](#) was implemented in the design of the k-space trajectories that define the undersampling mask in the Fourier domain for measuring k-space measurements. In addition, because [MR](#) images are not  $s$ -sparse in their original domain, Lustig et al. [LDP07] proposed to use sparsifying transforms to promote their “implicit sparsity”.

Let us denote  $\psi$  a potential sparsifying transform. Ideally, we would have for all [MR](#) images  $\mathbf{x} \in \mathbb{C}^n$ ,  $\psi(\mathbf{x})$  is sparse or at least compressible. Lustig et al. [LDP07] showed that the [Discrete Cosine Transform \(DCT\)](#) and the wavelet transform are both good candidates for  $\psi$ .

Regarding the incoherence, the key point to understand is that the k-space has to be sampled via trajectories respecting some physical constraints. Lustig et al. [LDP07] restrict their analysis to Cartesian sampling, and show that [Variable Density Sampling \(VDS\)](#) is a great candidate sampling strategy. In this case, the measurement operator is simply a masked [FT](#),  $\mathbf{A} = \text{diag}(\mathbb{1}_\Omega)\mathcal{F}$ , the [FT](#) being coherent with the classical sparsity basis in the low frequencies. This sampling allows to break the coherence barrier by sampling fully the center of k-space and sparsely the high frequencies [PVW11; CCW13; Cha+14]. Of course, if we do not restrict ourselves to Cartesian sampling, the space of trajectories we can consider becomes larger. Potential alternatives in that case can be either based on analytical non-Cartesian readouts such as radial spokes [BPM91; GP92] or spiral interleaves [AKC86; Mey+92]. One can then optimize for the best trajectory<sup>2</sup> for

<sup>2</sup>The trajectory requirements might change depending on the acquisition setting.

a given target density like what was done in the SPARKLING approach [Laz+19; Laz+20; Cha+21]. This optimization can also be done based on data using deep learning frameworks [Wei+21; Wan+21a; CZC21].

## 2.2 . Sparse reconstruction algorithms

Let us now focus on concrete examples of MRI reconstruction is implemented in the CS framework. We will try to remain general, so we will consider the 2/3D multicoil non-Cartesian setting. In the single-coil setting, there is only one sensitivity map, and it is equal to the identity. In the Cartesian setting, the NDFT is simply replaced by a masked FT.

### 2.2.1 . Classical algorithms

In an ideal setting (i.e. noiseless and without taking gradient inaccuracies and  $B_0$  inhomogeneities into account), the reconstruction can be done by solving the following *analysis formulation*,<sup>3</sup> as an optimization problem:

$$\arg \min_{\mathbf{x} \in \mathbb{C}^n} \|\psi \mathbf{x}\|_1 \quad \text{subject to } \mathcal{F}_\Omega \mathbf{S}_l \mathbf{x} = \mathbf{y}_l \quad \forall l = 1, \dots, L \quad (2.6)$$

where  $\psi$  is a sparsifying transform (typically a wavelet transform),  $\mathbf{x}$  is the reconstructed image,  $\mathcal{F}_\Omega$  is the NDFT on the  $\Omega$  set embodying all k-space trajectories,  $\mathbf{S}_l$  is the sensitivity map of the  $l$ -th coil,  $\mathbf{y}_l$  is the measurement of the  $l$ -th coil and  $L$  is the number of coils. The sensitivity maps  $(\mathbf{S}_l)_{l=1}^L$  encompass the local sensitivity of each coil, and we assume them to be known. However, in the real life, the noise in the measurements must be taken into account, and we need to relax Equation 2.6 in the following form:

$$\arg \min_{\mathbf{x} \in \mathbb{C}^n} \sum_{l=1}^L \frac{1}{2} \|\mathbf{y}_l - \mathcal{F}_\Omega \mathbf{S}_l \mathbf{x}\|_2^2 + \lambda \|\psi \mathbf{x}\|_1 \quad (2.7)$$

where  $\lambda$  is a hyperparameter controlling the regularization imposed by the sparsity: in essence, the less noise in the data, the smaller  $\lambda$ . The resolution of the optimization problem can be done mainly by two types of algorithms:

- Primal-dual algorithms: Primal-Dual Hybrid Gradient (PDHG) [CP11], Condat-Vu [Con13], Alternating Direction Method of Multipliers (ADMM) [Boy+11].
- Proximal gradient methods: Iterative Soft Thresholding Algorithm (ISTA) [DDD04], Faster ISTA (FISTA) [BT09], Proximal Optimal Gradient Method (POGM') [KF18].

An extensive review of these algorithms and how they are used in the context of MRI reconstruction was done by Fessler [Fes20]. This review also introduces other

---

<sup>3</sup>When  $\psi$  is invertible an equivalent formulation, called the *synthesis formulation* can be derived where we optimize over the coefficients of the transform rather than the image itself.



regularization schemes. In this section, we will focus on a single algorithm, [FISTA](#), the improved version of [ISTA](#).

In order to introduce [ISTA](#), let us recall the definition of a proximal operator:<sup>4</sup>

**Definition 2.2.1** (Proximal operator [[Mor62](#)]). *For a semi-continuous convex function  $\mathcal{R}$  defined on a real Hilbert space  $\mathcal{H}$  the proximal operator is defined as:*

$$\text{prox}_{\mathcal{R}}(\mathbf{x}) = \arg \min_{\mathbf{z} \in \mathcal{H}} \mathcal{R}(\mathbf{z}) + \frac{1}{2} \|\mathbf{z} - \mathbf{x}\|_2^2 \quad (2.8)$$

The basic idea of [ISTA](#) is to generalize the gradient step to nonsmooth functions which admit a closed-form proximal operator. The way to do so is to rewrite the quadratic approximation leading to gradient descent but this time with the addition of a nonsmooth function. Further details on this can be found in the work of Beck et al. [[BT09](#)]. Rewriting  $\mathcal{A} = (\mathbf{I}_L \otimes \mathcal{F}_\Omega) \mathbb{S}$ ,  $\mathbb{S} = [\mathbf{S}_1^H, \dots, \mathbf{S}_L^H]^\top$ ,  $\mathbf{y} = [\mathbf{y}_1^H, \dots, \mathbf{y}_L^H]^\top$ ,  $\mathcal{R}(\cdot) = \lambda \|\psi \cdot\|_1$ , and with  $^H$  the Hermitian (i.e. transpose conjugate) operator, we have the [ISTA](#) step of step size  $\epsilon_n$  for [Equation 2.7](#):

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n - \epsilon_n \mathcal{A}^H (\mathcal{A} \mathbf{x}_n - \mathbf{y}) \\ \mathbf{x}_{n+1} &= \text{prox}_{\epsilon_n \mathcal{R}}(\mathbf{x}_{n+1}) \end{aligned} \quad (2.9)$$

The first stage is known as the data consistency step, where one follows the gradient of the data consistency term (i.e. the smooth term) in [Equation 2.7](#). The second stage is known as the proximal step, where one is constrained by the proximal operator of the regularization. Informally, this second step makes the solution look more like an [MR](#) image and denoises the output of the gradient step. The idea behind [FISTA](#) is to generalize Nesterov acceleration [[Nes83](#)] to [ISTA](#), giving the following step:

$$\begin{aligned} \mathbf{z}_{n+1} &= \mathbf{x}_n - \epsilon_n \mathcal{A}^H (\mathcal{A} \mathbf{x}_n - \mathbf{y}) \\ \mathbf{z}_{n+1} &= \text{prox}_{\epsilon_n \mathcal{R}}(\mathbf{z}_{n+1}) \\ t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\ \mathbf{x}_{n+1} &= \mathbf{z}_{n+1} + \frac{t_k - 1}{t_{k+1}} (\mathbf{z}_{n+1} - \mathbf{z}_n) \end{aligned} \quad (2.10)$$

Just like many of aforementioned algorithms, [FISTA](#) can also be improved by different techniques such as restart, enhanced momentum sequence and greedy acceleration [[LS18](#)]. These improvements often directly translate to improved [MRI](#) reconstruction speeds [[Zac+19](#)], showcasing the interest of working on abstract versions of these algorithms.

---

<sup>4</sup>In order to familiarize oneself with proximal operators, the website [proximity-operator.net](http://proximity-operator.net) [[Chi+16](#)] is an excellent resource.

**Synthesis vs. Analysis.** The problem presented in Equation 2.7 is known as the analysis formulation for MRI reconstruction. Another way to frame it, is to actually consider that the signal to reconstruct is not the image itself but its coefficients in a transform basis like the wavelets. This framing is known as the synthesis formulation [EMR07], and reads as follows for an invertible wavelet basis  $\psi$ :

$$\arg \min_{z \in \mathbb{C}^n} \sum_{l=1}^L \frac{1}{2} \|\mathbf{y}_l - \mathcal{F}_\Omega \mathbf{S}_l \psi^{-1} z\|_2^2 + \lambda \|z\|_1 \quad (2.11)$$

Cherkaoui et al. [Che+18b] showed that the analysis formulation is superior in the case of MRI reconstruction, since it does not require the wavelet transform to be invertible.

### 2.2.2 . Dictionary learning

The algorithms mentioned above suffer from 2 problems:

- They rely on an iterative scheme which will make a huge number of calls to a computationally heavy  $\mathcal{A}$  operator.
- The prior on the MR images is handcrafted and not specific to MR images but instead selected for a broader class of natural images.

In order to tackle this last problem, Ravishankar et al. [RB11] leveraged *Dictionary Learning* in order to learn a more suited prior. They introduce a dictionary of patches (or atoms)  $\mathbf{D}$  which is learned as part of the overall optimization scheme. This gives the following optimization problem:

$$\min_{\mathbf{x}, \mathbf{D}, \mathbf{\Gamma}} \sum_{l=1}^L \frac{1}{2} \|\mathbf{y}_l - \mathcal{F}_\Omega \mathbf{S}_l \mathbf{x}\|_2^2 + \lambda \sum_{ij} \|\mathbf{R}_{ij} \mathbf{x} - \mathbf{D} \alpha_{ij}\|_2^2 \quad \text{subject to} \quad \|\alpha_{ij}\|_0 \leq T_0 \quad \forall i, j \quad (2.12)$$

with  $T_0$  a hyperparameter controlling the sparsity level,  $\mathbf{R}_{ij}$  the matrix extracting a patch at location  $i, j$  in the image,  $\mathbf{\Gamma}$  the collection of the sparse codes  $\alpha_{ij}$ . In order to solve Equation 2.12, an alternate minimization procedure is used, where in one instance,  $\mathbf{x}$  is fixed, and in the other  $\mathbf{D}$  and  $\mathbf{\Gamma}$  are fixed which leads to a least square problem with an explicit solution. When  $\mathbf{x}$  is fixed (this corresponds to the dictionary learning step) the optimization problem is solved with K-SVD.

Caballero et al. [Cab+14] extended this approach to temporal data.

## 2.3 . Parallel Imaging Reconstruction

In the presentation of the CS-MRI reconstruction problem in the previous part, we assumed the sensitivity maps were known. This assumption is not tenable in practice. Consequently, they must be either acquired in separate scans or internally estimated from k-space data or not used anymore in the formulation of the image

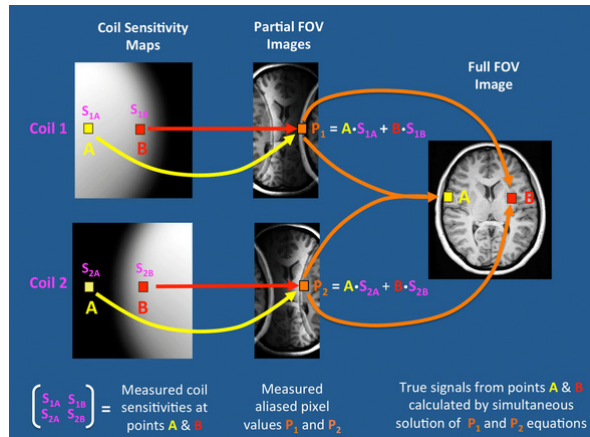


Figure 2.3-1: **SENSE reconstruction**: image courtesy of Elster et al. [EB01].

reconstruction problem. Regarding the latter possibility, we talk about calibration-less MR image reconstruction, and we refer the reader to the work (and references therein) of El Gueddari et al. [El +21] to go further into MR image reconstruction without the explicit use of sensitivity maps. Hereafter, we will see how to estimate these sensitivity maps and in the end practically combine multicoil acquisition with CS reconstruction.

### 2.3.1 . Image-domain techniques

One of the first techniques to deal with PI reconstruction is SENSE [Pru+99], which stands for SENSitivity Encoding. The basic idea of SENSE is to acquire coil sensitivity profiles before the actual acquisition in a low resolution fashion in order to have only a small overhead. At the reconstruction stage, the voxel values of the organ of interest are deduced from the voxel values of the multicoil images by solving a linear system of equations in the image domain involving the voxel values of the multicoil images and the sensitivity maps. In the noiseless setting, the equation for a single voxel in the case of a 2-fold acceleration with 2 coils is illustrated in Figure 2.3-1. Essentially, SENSE operates in the image domain once all coil-specific images have been Fourier-inverted from coil-specific undersampled k-space data. Then SENSE iterates over spatial locations and for a given position poses the problem of PI reconstruction as a fully determined linear inverse problem for which we can simply compute the pseudo-inverse of the measurement matrix and apply it to the multicoil images. As the linear systems are spatially independent their resolution can be carried out in parallel. Another image-domain technique with similar idea is ASSET. We will now focus on a type of reconstruction methods that are used in the context of this thesis (see section 6.3).

### 2.3.2 . Frequency-domain techniques

Another branch of techniques to reconstruct PI data works directly in the k-space. We will focus in this section on GRAPPA [Gri+02], which stands for Generalized Autocalibrating Partially Parallel Acquisitions. The idea behind GRAPPA is to acquire only a subset of lines in the k-space, and fill the rest by interpolation, using kernels calibrated on a fully sampled part of the k-space, the Autocalibration Signal (ACS), as can be seen in Figure 2.3-2. One of the key aspects of GRAPPA, is that the acquisition scheme must be equispaced (except in the ACS) in order for the same set of kernels to be applied similarly to the whole k-space.

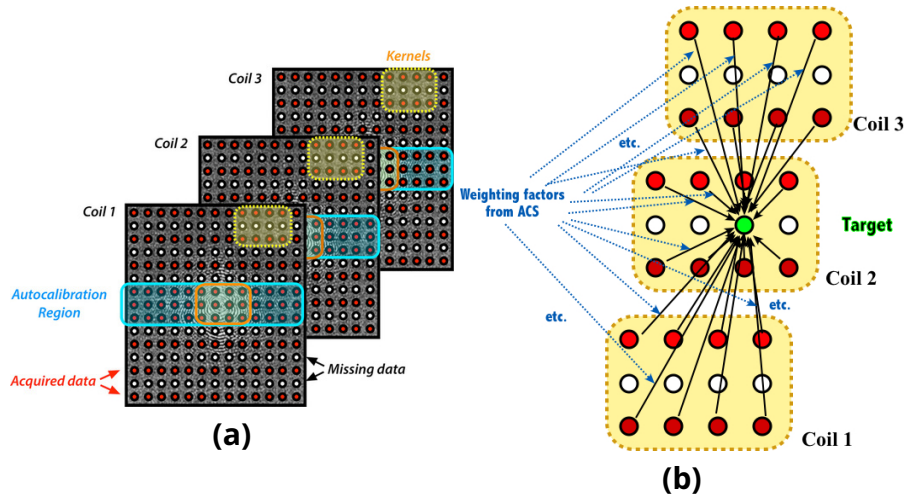


Figure 2.3-2: **(a) Autocalibration Signal - (b) Reconstruction.** The central part of the k-space is fully sampled and can be used as an Autocalibration Signal to calibrate the kernels. The reconstruction is then carried out linearly on the rest of the k-space. Images courtesy of Elster et al. [EB01].

The formal definition of the GRAPPA algorithm steps can be summarized as follows. For an AF of  $r$  (not taking into account the ACS), there are  $r - 1$  different “geometries”, i.e. not sampled lines between sampled lines. This means that we will consider  $r - 1$  different kernels to fill the k-space, each one corresponding to a different geometry, i.e. filling a not sampled k-space line from neighboring sampled k-space lines. Let us denote  $\mathcal{W}_{n_p, n_f}$  the space of all GRAPPA kernels, using  $2n_p$  neighboring sampled lines to fill the unsampled k-space lines ( $n_p$  on each side) and  $n_f$  points on each sampled line. If we denote by  $n_{coils}$  the number of coils, we have  $\mathcal{W}_{n_p, n_f} = \mathbb{R}^{n_{coils} \times 2n_p \times n_f \times n_{coils}}$ . In order to find the kernel for a given geometry  $i < r - 1$ , we need to solve the following optimization problem:

$$\mathbf{w}_i^* = \arg \min_{\mathbf{w} \in \mathcal{W}_{n_p, n_f}} \sum_{j=i+(n_p-1)*r}^{N_{ACS}-n_p*r-1+i} \|\mathbf{y}_{ACS}^{(j)} - \mathbf{w} * \mathbf{y}_{ACS}^{(-j,i)}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \quad (2.13)$$

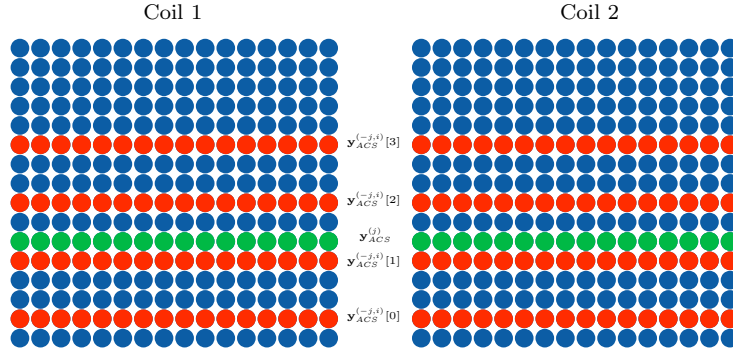


Figure 2.3-3: **ACS lines selection for kernel calibration during GRAPPA**: the target line  $\mathbf{y}_{ACS}^{(j)}$  is in orange, while the source lines  $\mathbf{y}_{ACS}^{(-j,i)}$  are in green. The other sampled lines of the ACS are in blue.

where  $N_{ACS}$  is the number of ACS lines,  $\mathbf{y}_{ACS}^{(j)}$  is the  $j$ -th line of  $\mathbf{y}_{ACS}$ ,  $\mathbf{y}_{ACS}^{(-j,i)}$  are the lines neighboring the  $j$ -th line in the  $i$ -th geometry, and  $\lambda$  is a regularization parameter. An example of how the  $\mathbf{y}_{ACS}^{(-j,i)}$  lines are selected is visible in Figure 2.3-3 for an AF  $r = 3$ ,  $n_{coils} = 2$  coils,  $n_p = 2$  lines taken into account in the grappa kernel on each side, the first geometry (i.e.  $i = 1$ ), and the  $j = 6$  line in this geometry. Once the kernel have been computed, the not sampled lines can be evaluated by the kernel corresponding to their geometry.

$$\mathbf{y}_{interp,i} = (\mathbf{w}_i^* * \mathbf{y}_{ACS}^{(-j,i)})_{j=i,\dots,n}. \quad (2.14)$$

Finally for the reconstructed k-space, we keep the ACS lines and the sampled lines,  $\mathbf{y}_{GRAPPA} = (1 - \Omega) \cdot (\sum_i \mathbf{y}_{interp,i}) + \Omega \cdot \mathbf{y}$ .

### 2.3.3 . Combination with CS techniques

The main takeaway from PI reconstruction techniques is the handling of sensitivity maps. Indeed, CS techniques as presented in section 2.2 do need the exact sensitivity maps in order to have a known forward model. Methods have then been developed in order to take the best of both worlds. For a theoretical analysis of optimal combination of CS with parallel imaging, the reader is invited to look at the following references [CAT15; BBW16].

A first PI reconstruction method to combine elements of CS with PI is SPIRiT [LP10] where ideas from SENSE and GRAPPA are combined to regularization techniques. In this case the reconstruction is carried out fully in the k-space and the output still needs to be combined to form the expected image like for GRAPPA.

Methods like ESPIRiT [Uec+14] which computes sensitivity maps solving an eigenvalue decomposition problem, or self-calibrating MRI reconstruction [EI +18] which relies on a simple thresholding of the central portion of k-space (as sens-

itivity maps are supposed to be smooth in the image domain) followed by a K-means [Llo82] extract estimates of sensitivity maps from the data. They then solve the problem introduced in Equation 2.7 with these estimated sensitivity maps.

## 2.4 . Quantitative evaluation of the reconstruction

With all these techniques in mind, a question arises: Which one should be used to reconstruct MR images? In other words, there is a need for a clear benchmarking strategy between acquisition-reconstruction couples. In order to compare the performances of  $p$  acquisition-reconstruction couples, the ideal setting is to acquire the same organ  $p + 1$  times in a short window of time: once in a fully sampled fashion and  $p$  times with the benchmarked acquisition schemes. The fully sampled acquisition is then reconstructed using the IFFT and the accelerated schemes with the corresponding reconstruction algorithms. The benchmarked couples are then compared to the fully sampled image visually, and the best couple is determined as the one approaching the image quality of fully sampled data (least artifacts and blurring, best contrast and sharpness). However, in practice this comparison procedure suffers from two flaws:

- **Several acquisitions** might take very long, especially for the fully sampled one. This could be difficult to carry for several methods and patients.
- **Experts** are needed to carry out the visual comparison of images, which is time-consuming. Additionally, this procedure be expensive.

In order to overcome these limitations, one can resort to the following solutions:

- **Retrospective studies** where the accelerated acquisitions are *simulated* from the fully sampled one, for example by masking some elements in the fully sampled raw k-space. The problem with this solution is that it does not reproduce the inherent issues that occur during prospective acceleration, in particular for non-Cartesian readouts (e.g. off-resonance effects are not usually modeled in retrospective studies).
- **Quantitative metrics** can be used to replace the experts' evaluation. However, finding a quantitative metric that perfectly reproduces the visual comparison is extremely difficult and a topic of research in itself. See more in subsection 2.4.4.

Although one could use only one of these two methods and for instance consider prospective acquisitions instead of retrospective ones, the assessment of quantitative metrics on MR images which have been prospectively accelerated and collected at two different time points might be very challenging due to potential motion artifacts between the different scans. Retrospective motion correction (i.e. after data acquisition and image reconstruction) would not be helpful in such scenarios

as the corrupted data directly impact the image quality at the reconstruction stage. Prospective motion correction could be considered, however this is still an active topic of research and slows down the final acceleration scheme.

Throughout this thesis, we will mainly rely on retrospective studies and quantitative metrics.<sup>5</sup>

### 2.4.1 . Classical quantitative metrics

Here we introduce the quantitative metrics that will be used in the rest of this dissertation. The first metric is the **Peak Signal-to-Noise Ratio (PSNR)**. It is defined as follows for a magnitude MRI image  $\mathbf{x} \in \mathbb{R}_+^n$  and its estimate  $\hat{\mathbf{x}}$ :

$$\text{PSNR}(\mathbf{x}, \hat{\mathbf{x}}) = 10 \log_{10} \left( \frac{\max_i \mathbf{x}_i}{\frac{1}{n} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2} \right) \quad (2.15)$$

The second metric is the **Structural Similarity Index Measure (SSIM)** [Wan+04]. It was designed to be closer to the visual comparison of images than the PSNR, although some critics have emerged pointing to downfalls of the metric [HZ10; NA20]. It is defined as follows for a magnitude MRI image  $\mathbf{x} \in \mathbb{R}_+^n$  and its estimate  $\hat{\mathbf{x}}$ :

$$\text{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = [l(\mathbf{x}, \hat{\mathbf{x}})]^\alpha \cdot [c(\mathbf{x}, \hat{\mathbf{x}})]^\beta \cdot [s(\mathbf{x}, \hat{\mathbf{x}})]^\gamma \quad (2.16)$$

where  $l(\mathbf{x}, \hat{\mathbf{x}})$  is the luminance measure,  $c(\mathbf{x}, \hat{\mathbf{x}})$  is the contrast measure, and  $s(\mathbf{x}, \hat{\mathbf{x}})$  is the structure measure. More details on these measures can be found in the work of Wang et al. [Wan+04].

### 2.4.2 . Advanced metrics

Some efforts have been put first in enhancing the aforementioned metrics. For example, the **Multiscale-SSIM (MSSIM)** has been introduced by Wang et al. [WSB03]. The idea is to compute the contrast and structure measures at different scales  $j = 1, \dots, M$ , and then compute the MSSIM as follows:

$$\text{MSSIM}(\mathbf{x}, \hat{\mathbf{x}}) = [l(\mathbf{x}, \hat{\mathbf{x}})]^{\alpha M} \cdot \prod_{j=1}^M [c(\mathbf{x}, \hat{\mathbf{x}})]^{\beta_j} \cdot [s(\mathbf{x}, \hat{\mathbf{x}})]^{\gamma_j} \quad (2.17)$$

A different line of research tried to use DL to craft image quality metrics. A notable example is **PieAPP** [Pra+18] which leveraged a newly introduced dataset to train a neural network to compute a metric which resembled human assessment. A more thorough review of advanced metrics both with and without DL can be found in the work of Mikhailiuk [Mik21].

---

<sup>5</sup>With the notable exception of the fastMRI challenge [Muc+21] in which 6 radiologists graded the image quality with respect to artifacts, sharpness and contrast to noise ratio in a double-blind fashion.

### 2.4.3 . Specificities of MRI evaluation

The first problem posed by MRI is that the classical output of the reconstruction is a complex-valued image, for which most visual quality metrics are not suited. This problem is usually avoided by just considering the magnitude of the image, like radiologists would do in most cases. We do not cover here the problem of evaluating the phase of the image, as would be useful for [Susceptibility Weighted Imaging \(SWI\)](#) [BSS15].

The second problem is that the output of the reconstruction is not a 2D image but a 3D volume. The authors of the fastMRI dataset [Zbo+18] have used the generalization of the classical metrics to 3D volumes rather than considering the average over the slices of the volume. Because the fastMRI dataset is becoming a standard in the research, we decided to use this method to compute our metrics, but it is a topic of discussion since the reconstruction is read slice by slice.

The final problem is that the ideal visual comparison that would be done by an expert is not sensitive to the scale differences between the images. However, the SSIM and the PSNR are sensitive to it. One solution to overcome this problem would be to normalize both the fully sampled and the reconstructed accelerated images. However, this approach was not retained in the fastMRI dataset metrics computation, and therefore using it would make the comparison of results between different research works difficult. For this reason we chose throughout this thesis to use *unnormalized* metrics.

### 2.4.4 . Discussion on the relevance of quantitative metrics

Many works have noticed the disagreement between classical quantitative metrics and human or expert ratings [Pra+18; Gu+20; Mie+21]. The solutions proposed have been to gather sizeable datasets of manual image quality evaluations of different types. From these datasets, neural networks can be trained to define a more ad hoc metric. These network-defined metrics have not been tested throughout this thesis, but it might be an interesting research direction to test them. However, one foreseeable problem is that since these networks are not trained on MRI images, they will not be easily transferable to them. On top of that, reproducing the training with MRI images would be challenging since the gathering of equivalent datasets seems out of reach.

\* \* \*  
\* \*  
\*





## 3 - Introduction to Deep Learning

### Chapter Outline

|       |   |    |
|-------|---|----|
| 3.1   | Timeline of Deep Learning . . . . .                       | 46 |
| 3.2   | The base ingredients of Deep Learning . . . . .           | 46 |
| 3.2.1 | Formalism . . . . .                                       | 46 |
| 3.2.2 | Backpropagation with the chain rule . . . . .             | 48 |
| 3.2.3 | SGDs . . . . .  | 49 |
| 3.2.4 | Computing power of GPUs . . . . .                         | 50 |
| 3.2.5 | Big Data . . . . .  | 50 |
| 3.3   | Classical Architectural Blocks of Deep Learning . . . . . | 52 |
| 3.3.1 | Perceptron . . . . .                                      | 52 |
| 3.3.2 | Nonlinearities . . . . .                                  | 52 |
| 3.3.3 | Convolutions . . . . .                                    | 54 |
| 3.3.4 | Pooling . . . . .   | 54 |
| 3.3.5 | Normalization . . . . .                                   | 55 |
| 3.3.6 | Residual/skip connections . . . . .                       | 56 |
| 3.3.7 | Attention . . . . .                                       | 56 |
| 3.3.8 | Dropout . . . . .   | 56 |

**D**EEP learning can be defined as the field of research where the goal is to design highly nonlinear models for a specific task (e.g. regression, classification, segmentation, etc.) and the corresponding algorithms that permit to calibrate these models from data. For this reason, deep learning is of particular interest for Inverse Problems solving where two highly nonlinear dynamics appear obviously:

- the relationship between the signal/image to be recovered  $\boldsymbol{x}$  and its noisy undersampled measurements  $\boldsymbol{y}$ ;
- the prior that exists over the class of signals/images to be recovered.

These two ingredients can be assembled in a probabilistic way by combining the distribution  $p(\boldsymbol{y}|\boldsymbol{x})$  (or the corresponding maximum likelihood estimator  $\arg \max_{\boldsymbol{x}} p(\boldsymbol{y} | \boldsymbol{x})$ ) with the prior density  $p(\boldsymbol{x})$ .

In this chapter, we will introduce the main concepts of deep learning in general while we postpone the presentation of their specific implementation in the MRI reconstruction context to [chapter 4](#).

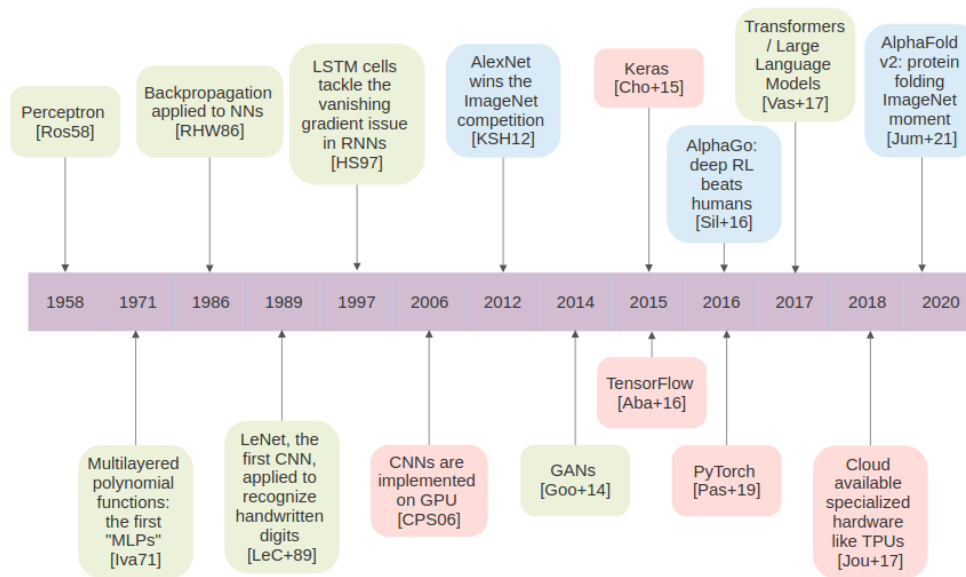


Figure 3.1-1: **Timeline of Deep Learning.**

### 3.1 . Timeline of Deep Learning

The primary message of this section is that deep learning, although a very hot and timely topic, has a longstanding history and was able to achieve tremendous scientific results thanks to a combination of important factors. Those factors can be summarized as follows:

- an extensive, long-lasting research on the founding elements;
- an increased availability of intensive computational resources, notably [Graphical Processing Units \(GPUs\)](#);
- some efforts to collect large scale labelled datasets;
- the emergence of reliable and well-documented open-source software and libraries backed by large companies.

These factors are summarized in the timeline of [Figure 3.1-1](#).

### 3.2 . The base ingredients of Deep Learning

In this section we will review what is typically needed to implement a deep learning algorithm and focus on the modeling aspect in [section 3.3](#).

#### 3.2.1 . Formalism

A deep learning setup usually involves the following elements:

- a dataset  $\mathcal{D}$  composed of elements  $\mathbf{x}_i$  or pairs of elements  $(\mathbf{x}_i, \mathbf{y}_i)$  coming from a stochastic generative process assumed stationary  $\mathcal{P}$ ;
- a function, typically highly nonlinear, parametrized by  $\boldsymbol{\theta} \in \Theta$  whose output depends on the task at hand,  $f_{\boldsymbol{\theta}}(\mathbf{x})$ ;
- an ideal loss function  $\mathcal{L}_{\text{ideal}}$  defining the task to be achieved (and some regularization), which generally depends on the output of the model  $f_{\boldsymbol{\theta}}(\mathbf{x})$ , potentially  $\mathbf{y}$ , the generative process  $\mathcal{P}$  and the model's parameters  $\boldsymbol{\theta}$ ;<sup>1</sup>
- an empirical proxy function  $\mathcal{L}$  which is a proxy of the ideal loss function in the real case setting where we do not have access to the generative process  $\mathcal{P}$ , and depends on the output of the model  $f_{\boldsymbol{\theta}}(\mathbf{x})$ , potentially  $\mathbf{y}$ , some elements of the dataset  $\mathcal{D}$  and the model's parameters  $\boldsymbol{\theta}$ .

The theoretical objective is to solve the following optimization problem:

$$\arg \min_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{P}} \mathcal{L}_{\text{ideal}}(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y}, \mathcal{P}, \boldsymbol{\theta}) \quad (3.1)$$

In practice, the following optimization problem is solved:

$$\arg \min_{\boldsymbol{\theta} \in \Theta} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i, \mathcal{D}, i, \boldsymbol{\theta}) \quad (3.2)$$

This second objective function is generally the empirical mean version of [Equation 3.1](#), but can be more complex in particular regarding the link to the generative process  $\mathcal{P}$ .

An example of this problem is the case of  $\ell_2$ -regularized regression with mean squared error, and a regularization parameter  $\lambda > 0$ :

$$\arg \min_{\boldsymbol{\theta} \in \Theta} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}} \frac{1}{2} \|f_{\boldsymbol{\theta}}(\mathbf{x}_i) - \mathbf{y}_i\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \quad (3.3)$$

In order to solve [Equation 3.2](#), first-order gradient based methods are typically used despite two seemingly contraindicated facts:

- the problem might not be convex (in  $\boldsymbol{\theta}$ );
- the objective function might not be strictly speaking differentiable (w.r.t.  $\boldsymbol{\theta}$ ).

Recently some works have introduced methods that no longer rely on the gradient to train neural networks, like Knyazev et al. [[Kny+21](#)] who use a meta-neural network to predict other neural networks' parameters.

---

<sup>1</sup>We use this very general formalism to also encompass in it [Generative Adversarial Networks \(GANs\)](#) or self-supervised learning among others.

**Theory and depth.** While the theory of deep learning is lacking the full understanding of all the dynamics at play, some results are of particular interest. A particularly interesting one is that neural networks are universal function approximators [HSW89; Bar93; Han19]. This means that any continuous function can be approximated by a neural network if given a sufficient width or depth. This result can be generalized for example to convolutional neural networks which are universal translation-equivariant (or invariant) function approximators [Mar+19; Yar21]. While these results are sometimes obtained by playing on the width of the neural network, in practice, neural networks often reach a better performance when they are deeper, for example in MRI reconstruction [Pez+20]. An intuition about this result has been proposed by Telgarsky [Tel16].

### 3.2.2 . Backpropagation with the chain rule

In general, the function defining the model (also called neural network) is defined as a sequence of simpler operations. Therefore, in order to obtain the gradient of the objective function with respect to the parameters  $\theta$ , one needs to be able to compute the gradient of a composition of functions. This is exactly what the chain rule allows us to achieve.

**Proposition 3.2.1** (Chain rule). *Let  $f$  and  $g$  be two differentiable functions, and let us denote  $h = f \circ g$ . The partial derivative of  $h$  can be computed as follows:*

$$\frac{\partial h}{\partial \cdot} = \frac{\partial f}{\partial g} \frac{\partial g}{\partial \cdot} \quad (3.4)$$

For a neural network defined as  $f_{\theta} = f_{1,\theta_1} \circ f_{2,\theta_2} \circ \dots \circ f_{d,\theta_d}$ , the gradient of the loss with respect to each set of parameters  $\theta_i$  will be given by:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{\partial \mathcal{L}}{\partial f_{1,\theta_1}} \frac{\partial f_{1,\theta_1}}{\partial f_{2,\theta_2}} \dots \frac{\partial f_{i,\theta_i}}{\partial \theta_i} \quad (3.5)$$

We see in Equation 3.5 that the computation of  $e_i = \frac{\partial \mathcal{L}}{\partial f_{1,\theta_1}} \frac{\partial f_{1,\theta_1}}{\partial f_{2,\theta_2}} \dots \frac{\partial f_{i-1,\theta_{i-1}}}{\partial f_{i,\theta_i}}$  is common for all the parameters' gradients of indices  $j \geq i$  and can therefore be reused rather than recomputed for each set of parameters. This method gives its name to the gradient computation in neural networks: backpropagation since we backpropagate the error  $e_i$  to compute  $e_{i+1}$ .

**Activations.** In order to compute efficiently the different gradients involved in the chain rule, we need to store the intermediate results of the computation of  $f_{\theta}$ . These intermediate results are usually referred to as activations, and they typically represent the largest share of the memory requirements of the model training.

A simplified example of why we need these activations for the gradient computation is the following. Suppose we want to compute the gradient of  $f_{\theta}(\mathbf{x}) = \exp(\theta^{\top} \mathbf{x})$ .

$$\frac{\partial f_{\theta}(\mathbf{x})}{\partial \theta} = \exp(\theta^{\top} \mathbf{x}) \cdot \mathbf{x} = f_{\theta}(\mathbf{x}) \cdot \mathbf{x} \quad (3.6)$$

Here, the activations are both  $f_{\theta}(x)$  and  $x$  (usually the output of a previous intermediate operation).

### 3.2.3 . SGDs

In the typical setting formalized in [Equation 3.2](#), the loss is expressed as a sum over elements of the dataset. Therefore, the gradient of the loss can also be expressed as a sum over elements of the dataset, by linearity of the gradient. One can take advantage of this property when the dataset is too large, making the computation of the full gradient too computationally expensive. Indeed rather than computing the full gradient at each gradient descent step, one can compute only a stochastic gradient by summing over a batch of the dataset and not the whole dataset. The stochastic gradient descent algorithm is the *de facto* algorithm

---

#### Algorithm 1 : Stochastic Gradient Descent.

---

**Result :** Optimal parameters  $\theta^*$

```

1 Batch size  $b$ , learning rate  $\eta$ , iteration index  $k = 0$ 
2 while not converged do
3   Sample  $b$  elements indices from the dataset  $\mathcal{D}$ ,  $i_1, \dots, i_b$ 
   uniformly at random without replacement, and remove
   them from  $\mathcal{D}$ 
4    $\theta^{(k)} \leftarrow \theta^{(k-1)} - \eta \sum_{j=1}^b \frac{\partial \mathcal{L}(f_{\theta^{(k-1)}}(\mathbf{x}_{i_j}), \mathbf{y}_{i_j}; \mathcal{D}, i_j, \theta^{(k-1)})}{\partial \theta^{(k-1)}}$ 
5    $k \leftarrow k + 1$ 
6  $\theta^* = \theta^{(k)}$ 

```

---

used to train neural networks [[LeC+89](#)] and as such is summarized in [Algorithm 1](#).

**Enhanced versions.** An entire field of research (called optimization for deep learning) is dedicated to improving the efficiency of stochastic gradient descent in the context of deep learning. Some algorithms can be readily borrowed from the existing first-order optimization literature like the idea of using a momentum [[Sut+13](#)]. More tailored approaches also exist, and they usually try to tackle the fact that different parameters of the models might have different learning rates. Adam [[KB15](#)] and RMSProp [[HSS12](#)] are two examples of such approaches.

**Role on generalization.** Early works [[LeC+12](#)] mentioned [SGD](#) as one of the main factors explaining the generalization capabilities of neural networks. However, this claim has been recently disputed, with some works suggesting that this assertion is right [[SL18](#); [SED20](#)] whereas others argue that full batch gradient descent can perform as well [[Hua+20](#); [Gei+21](#)].

### 3.2.4 . Computing power of GPUs

The vast majority of the blocks used to build deep learning models are linear functions and point-wise nonlinearities. Making sure that these operations can be executed as fast as possible on hardware is key to obtaining a practical tool. GPUs provide the possibility to write these operations in a highly parallel way [CPS06]. This can result in impressive computation gains compared to Central Processing Units (CPUs): typically up to 7 times faster for training [BD18].

In practice, the complexity of using custom operations on GPUs is abstracted away from the end user by the typical deep learning frameworks like TensorFlow [Aba+16] or PyTorch [Pas+19]. This allows a low barrier of entry for researchers in conjunction with the increasing availability of computing resources either via online tools such as Colab [19b] or Kaggle [10], or via public-funded supercomputers like the Jean Zay supercomputer [19c] we used for a large majority of the experiments.

### 3.2.5 . Big Data

As neural networks use a very high number of parameters, they are prone to overfitting in the classical regime.<sup>2</sup> One way to regularize such networks without losing too much of their capacity is to train them with more data. This is why the dataset gathering efforts have also been key to the success of deep learning.

Some examples include:

- ImageNet [Den+09] for image classification;
- WikiText-103 [Mer+16] for language modeling;
- IMDB dataset [Maa+11] for sentiment analysis;
- Cityscapes [Cor+16] for semantic segmentation;
- fastMRI [Zbo+18] for MRI reconstruction;
- YouTube-8M [Abu+16] for video classification;
- howto100m [Mie+19] for text-video representation learning;
- TUH [Har+14] for EEG signals analysis;
- COCO [Lin+14] for object detection;
- CelebA [Liu+15] for facial attributes recognition;
- KITTI [Gei+13] for robotics and vision.

---

<sup>2</sup>although recently the phenomenon of double descent [Bel+19; Nak+20; BHX20] / grokking [Pow+21] has questioned our understanding of overfitting.



Figure 3.2-2: **Samples of the ImageNet dataset** [Den+09], courtesy of Karpathy [Kar19].



**Data augmentation.** Collecting more data to build even bigger datasets is sometimes out of reach due to the potential costs involved. When simulation is not an option to generate data, data augmentation can be used to increase the size of the dataset albeit in a much less rich manner. Data augmentation has however played a huge role in improving the performance of neural networks and is sometimes central to the method used like in contrastive learning.

The concept of data augmentation is to apply a transformation  $\mathcal{T}$  to an element of the dataset  $\mathbf{x}$  or  $(\mathbf{x}, \mathbf{y})$  and add its result to the dataset. The goal is then to find a set of transformations  $\mathcal{S}_{\mathcal{T}}$  that fulfills the invariances or equivariances of the problem.

### 3.3 . Classical Architectural Blocks of Deep Learning

This section presents the typical blocks that go into the design of a neural network. While as presented in [section 3.2](#) the blocks are typically used sequentially, some of the ones presented hereafter do not fit this description. Due to this typically sequential nature, the blocks are usually called layers, but this term is overloaded, and a layer can be defined in the literature as a composition of many of these blocks.

#### 3.3.1 . Perceptron

The Perceptron [[Ros58](#)] is the most widely thought of neural network building block. It serves for example as a basis for the [Multi-Layer Perceptron \(MLP\)](#), which basically chains multiple Perceptron layers. The Perceptron is applied on an input  $\mathbf{x} \in \mathbb{R}^{p_{\text{in}}}$ , and can be defined as:

$$f_{\mathbf{W}, \mathbf{b}}(\mathbf{x}) = \sigma(\mathbf{W}^{\top} \mathbf{x} + \mathbf{b}) \in \mathbb{R}^{p_{\text{out}}} \quad (3.7)$$

where  $\sigma$  is the activation function, typically a [Rectified Linear Unit \(ReLU\)](#) or a sigmoid,  $\mathbf{W} \in \mathbb{R}^{p_{\text{out}} \times p_{\text{in}}}$  is the weight matrix, and  $\mathbf{b} \in \mathbb{R}^{p_{\text{out}}}$  is the bias vector. An [MLP](#) is then defined as a composition of several Perceptron layers, with potentially varying dimensions  $p_{\text{in}}$  and  $p_{\text{out}}$ .

#### 3.3.2 . Nonlinearities

An important aspect of deep learning is the use of nonlinearities in order to create highly non-linear functions. Let us discuss here the most common nonlinearities used in deep learning whose implementation can be found in the common deep learning frameworks, potentially under the name “activation”. Most of the typical nonlinearities in deep learning are defined pointwise, i.e. are applied independently to all the coordinates of the input. For a more extended discussion on nonlinearities we refer the reader to the review of Nwankpa et al. [[Nwa+18](#)].

**ReLU and variants.** The [ReLU](#) [[NH10b](#)] is the most commonly used nonlinearity in deep learning. The [ReLU](#) is applied on an input  $x$  with any shape and

can be defined pointwise as:

$$f(\mathbf{x}) = \max(0, \mathbf{x}) \quad (3.8)$$

2 aspects make the **ReLU** a surprising candidate to build neural networks:

- It is non-differentiable at 0. However, this is often overlooked as the input to it is rarely exactly 0.
- It has a gradient of 0 for negative inputs. This means that in roughly half of the cases, the gradient is annealed by the **ReLU** making the training less efficient.

This last problem is partly what causes the vanishing gradient problem. In order to address this limitation, some variants of the **ReLU** have been proposed. The first is the **Leaky ReLU (LReLU)** [MHN13], which is defined pointwise as:

$$f_{\alpha}(\mathbf{x}) = \max(0, \mathbf{x}) + \alpha \min(0, \mathbf{x}) \quad (3.9)$$

where  $\alpha \in \mathbb{R}^{*,+}$  is a constant scalar usually taken smaller than 1. A natural extension of this is the **Parametric ReLU (PReLU)** [He+15], which is defined pointwise as:

$$f_{\alpha}(\mathbf{x}) = \max(0, \mathbf{x}) + \alpha \min(0, \mathbf{x}) \quad (3.10)$$

where  $\alpha \in (\mathbb{R}^{*,+})^p$  is a learnable parameter of the activation function, typically shared across some dimensions of the input.

**Sigmoid.** The sigmoid is a nonlinearity used when the output needs to be bounded between two values, for example 0 and 1 in the case of an output potentially interpreted as a probability. The sigmoid is defined pointwise as:

$$f(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x})} \quad (3.11)$$

The sigmoid can also suffer from uninformative gradients at the tails and in some situations should be avoided as pointed out by Ramzi et al. [Ram+21].

**SoftMax.** The SoftMax is one of the most widely used output nonlinearities in deep learning when the output is expected to be similar to a probability distribution, for example in classification. The SoftMax is defined on an input  $\mathbf{x} \in \mathbb{R}^p$  as:

$$f(\mathbf{x}) = \frac{e^{\mathbf{x}_i}}{\sum_{j=1}^p e^{\mathbf{x}_j}} \quad (3.12)$$

The SoftMax gets its name from the fact that it is a smooth version of the maximum function. Indeed, the SoftMax exacerbates the differences between the maximum and the other values.

**Tanh.** The hyperbolic tangent is another nonlinearity used in deep learning, although recently fallen out of favor. It is defined pointwise as:

$$f(\mathbf{x}) = \tanh(\mathbf{x}) = \frac{\exp(\mathbf{x}) - \exp(-\mathbf{x})}{\exp(\mathbf{x}) + \exp(-\mathbf{x})} \quad (3.13)$$

**GELU.** The [Gaussian Error Linear Unit \(GELU\)](#) [HG16] is a recently introduced nonlinearity. It is defined pointwise as:

$$f(\mathbf{x}) = \mathbf{x}P(X < \mathbf{x}) \quad (3.14)$$

where  $P(X) \sim \mathcal{N}(0, 1)$ . This function is now the de-facto function used in modern architectures like [MLP-Mixers](#) [Tol+21].

### 3.3.3 . Convolutions

Convolutions are at the core of [Convolutional Neural Networks \(CNNs\)](#) [LeC+89; Kri09], which are the first highly successful types of neural networks. Convolutions are a special case of the Perceptron layer, where the weight matrix is actually a convolution matrix. This is ideal for neural networks because it imposes a strong inductive bias on the learned function which allows to reduce the amount of data needed to train them. Indeed, the convolution is translation-equivariant which is a desired behavior when working on a wide variety of signals, for example natural images or sound. The convolution operation is defined for an input  $\mathbf{x} \in \mathbb{R}^{w \times h \times c}$ , where  $w$  is the width of the image,  $h$  is the height of the image, and  $c_{\text{in}}$  is the number of channels of the image, as:

$$f_{\mathbf{W}, \mathbf{b}}(\mathbf{x})[i, j, k] = \sum_{i'=0}^K \sum_{j'=0}^K \sum_{k'=0}^{C_{\text{in}}} \mathbf{W}_{i', j', k, k'} \mathbf{x}_{i'+i-\frac{K}{2}, j'+j-\frac{K}{2}, k'+k} + \mathbf{b}_k \quad (3.15)$$

where  $i \leq w$ ,  $j \leq h$ , and  $k \leq c_{\text{out}}$ , the number of output channels,  $K \in \mathbb{N}$  is the kernel size,  $\mathbf{W} \in \mathbb{R}^{K, K, C_{\text{in}}, C_{\text{out}}}$  is the convolution kernel and  $\mathbf{b} \in \mathbb{R}^{C_{\text{out}}}$  is the bias. This definition can be easily generalized to 1D or 3D convolutions.

A very nice dynamic explanation of how convolutions are applied to images in CNNs can be found at this page [cs231n.github.io/convolutional-networks/#conv](https://cs231n.github.io/convolutional-networks/#conv) [LK15].

**Padding.** The convolution is in practice not correctly defined at the edges of the image. In order to tackle this, padding, i.e. enlarging the image, can be used. The most common type of padding is zero-padding, where the image is enlarged to a certain width and height by adding successive layers of 0s around it.

### 3.3.4 . Pooling

When working with signals, it is sometimes useful consider multiple scales. This can be done using pooling operations, which reduce the dimension of the signal. This is analogous to the decimation operation used in wavelets.

Two types of pooling operations exist: average pooling and max pooling.

Max pooling is the most common pooling operation used in CNNs, and is defined for an input  $\mathbf{x} \in \mathbb{R}^{w \times h}$  as:

$$f(\mathbf{x})[i, j] = \max_{i'=-\frac{K}{2} \dots \frac{K}{2}; j'=-\frac{K}{2} \dots \frac{K}{2}} \mathbf{x}_{i'+i, j'+j} \quad (3.16)$$

where  $K \in \mathbb{N}$  is the pooling size.

Average pooling is mostly used as a final aggregation layer (called Global Average Pooling in this case) [LCY13]. It is defined for an input  $\mathbf{x} \in \mathbb{R}^{w \times h}$  as:

$$f(\mathbf{x})[i, j] = \frac{1}{K^2} \sum_{i', j'=-\frac{K}{2}}^{\frac{K}{2}} \mathbf{x}_{i'+i, j'+j} \quad (3.17)$$

where  $K \in \mathbb{N}$  is the pooling size.

### 3.3.5 . Normalization

In theory, the normalization of the inputs for neural networks is not necessary since the weights should be learned to take into account the range of the inputs values. However, in practice, because the training happens on computers with finite precision, it is essential to normalize the inputs [Bis+95, Chap. 8]. Moreover, for very deep networks this also becomes true for the intermediary results. For this purpose, batch normalization [IS15] was introduced as a technique to allow the normalization of the intermediary results of neural networks. This technique was then generalized to instance normalization [UVL16], layer normalization [BKH16] and group normalization [WH18] which allow training with smaller batch sizes.<sup>3</sup>

We use the notation of Wu et al. [WH18] to introduce the different types of normalization for images.

$$f_{\mu, \sigma}(\mathbf{x})[b, i, j, k] = \frac{1}{\sigma_{b, i, j, k}} (\mathbf{x}_{b, i, j, k} - \mu_{b, i, j, k}) \quad (3.18)$$

where  $\mu_{b, i, j, k} = \frac{1}{m} \sum_{(b', i', j', k') \in \mathcal{S}_{b, i, j, k}} \mathbf{x}_{b', i', j', k'}$  and  $\sigma_{b, i, j, k} = \sqrt{\frac{1}{m} \sum_{(b', i', j', k') \in \mathcal{S}_{b, i, j, k}} (\mathbf{x}_{b', i', j', k'} - \mu_{b, i, j, k})^2}$  + define the mean and the standard deviation of the normalization, and  $m$  is the size of the set  $\mathcal{S}_{b, i, j, k}$  (usually constant). What defines each normalization is the choice of the set  $\mathcal{S}_{b, i, j, k}$ , as can be seen in Figure 3.3-3.

- For **Batch normalization**,  $\mathcal{S}_{b, i, j, k} = \{(b', i', j', k') | k = k'\}$ .
- For **Instance normalization**,  $\mathcal{S}_{b, i, j, k} = \{(b', i', j', k') | b = b', k = k'\}$ .
- For **Layer normalization**,  $\mathcal{S}_{b, i, j, k} = \{(b', i', j', k') | b = b'\}$ .
- For **Group normalization**,  $\mathcal{S}_{b, i, j, k} = \{(b', i', j', k') | b = b', \lfloor \frac{k}{c/g} \rfloor = \lfloor \frac{k'}{c/g} \rfloor\}$ .

Notably some works tried to show that normalization might not be needed within the network to obtain good performances [Bro+21].

<sup>3</sup>Weight normalization [SK16] is not a normalization technique but a reparameterization technique

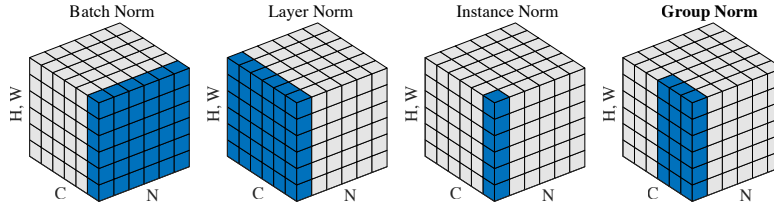


Figure 3.3-3: **Illustration of the different choices of  $\mathcal{S}_{b,i,j,k}$ .** Courtesy of Wu et al. [WH18].

### 3.3.6 . Residual/skip connections

The residual and skip connections are the simplest blocks presented here. They were introduced in order to solve the vanishing gradient problem [He+16]. These 2 blocks are not layers as such, but actually wrappers around other layers.

The residual connection is defined for a layer  $g$  and an input  $\mathbf{x}$  as:

$$f(\mathbf{x}) = g(\mathbf{x}) + \mathbf{x} \quad (3.19)$$

On the other hand, the skip connection is defined for a layer  $g$  and an input  $\mathbf{x}$  as:

$$f(\mathbf{x}) = \mathbf{x} \parallel g(\mathbf{x}) \quad (3.20)$$

where  $\parallel$  is the concatenation operator.

### 3.3.7 . Attention

The conception of attention stems from the need to query memories/dictionaries based on the data. The indexing or query mechanisms are not differentiable, making them unfit for neural networks design. With this view, attention can be seen as a “soft-indexing” or “soft-querying”. In its original form [BCB15], for keys  $\mathbf{K}$  and corresponding values  $\mathbf{v}$  (coming from a potentially anterior computation), attention can be defined as:

$$f(\mathbf{x}; \mathbf{K}, \mathbf{v}) = \text{SoftMax}(\mathbf{x}^\top \mathbf{K}) \mathbf{v} \quad (3.21)$$

This concept was generalized by Vaswani et al. [Vas+17] in the self-attention mechanism core to the Transformer architectures.

$$f_{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V}(\mathbf{x}) = \text{SoftMax}((\mathbf{x}^\top \mathbf{W}_Q)^\top \mathbf{x}^\top \mathbf{W}_K) \mathbf{x}^\top \mathbf{W}_V \quad (3.22)$$

We refer the reader to this widely acclaimed blog post for a more thorough and intuitive explanation of the self-attention mechanism in Transformers: [jalammr.github.io/illustrated-transformer](http://jalammr.github.io/illustrated-transformer).

### 3.3.8 . Dropout

Dropout [Sri+14] used in the training of neural networks to regularize the model. Intuitively, it tries to drop randomly parts of the connections between

layers to make them less dependent on each other. In practice dropout is defined by at training time for an input  $\mathbf{x} \in \mathbb{R}^d$  as:

$$f_p(\mathbf{x}) = \frac{1}{p} \mathbf{M} \odot \mathbf{x} \quad \text{where} \quad \mathbf{M} \sim \mathcal{B}^d(p) \quad (3.23)$$

where  $p \in [0, 1]$  is the dropout rate, and  $\mathbf{M}$  is a binary mask.

Notice that the dropout introduces a scaling factor  $\frac{1}{p}$  in order to compensate the fact that some entries of the vector are set to 0, therefore reducing the total “energy” of the vector. At test time, the dropout is just the identity, without masking or scaling.

**Other regularization techniques.** Although dropout is one of the most common regularization techniques in deep learning, it is not the only one.

- **Data augmentation** is sometimes cited as a regularization technique simply because it reduces the overfitting by in practice increasing the size of the training data.
- **Weight decay**, another name for L2-regularization,<sup>4</sup> penalizes the weights of the network based on their norm in the loss function.

\* \* \*  
\* \*  
\*

---

<sup>4</sup>the difference between the use of the 2 terms being just the implementation, see [this medium article for example](#)



# **Part II**

## **Methodological Developments**





## 4 - Review of Deep Learning for MRI reconstruction

### Chapter Outline

|       |   |    |
|-------|---|----|
| 4.1   | Paradigms for deep learning use in MRI reconstruction . . . . . | 62 |
| 4.1.1 | Plug-and-Play . . . . .   | 62 |
| 4.1.2 | Agnostic learning . . . . .                                     | 63 |
| 4.1.3 | Single-domain restoration . . . . .                             | 63 |
| 4.1.4 | Adversarial reconstruction . . . . .                            | 64 |
| 4.1.5 | Deep Compressed Sensing . . . . .                               | 64 |
| 4.1.6 | Deep Image Prior . . . . .                                      | 64 |
| 4.1.7 | Self-supervised . . . . .                                       | 65 |
| 4.1.8 | Implicit field learning . . . . .                               | 65 |
| 4.2   | Benchmarking unrolled networks for MRI reconstruction . . . . . | 66 |
| 4.2.1 | Introduction . . . . .  | 66 |
| 4.2.2 | Related works . . . . .   | 67 |
| 4.2.3 | Models . . . . .  | 67 |
| 4.2.4 | Data . . . . .  | 72 |
| 4.2.5 | Results . . . . .   | 73 |
| 4.2.6 | Discussion . . . . .  | 76 |
| 4.3   | Unrolled networks for MRI reconstruction . . . . .              | 77 |
| 4.3.1 | Model-based Deep Learning . . . . .                             | 77 |
| 4.3.2 | Variational Network . . . . .                                   | 77 |
| 4.3.3 | $\Sigma$ -Net . . . . .   | 78 |
| 4.3.4 | End-to-end VarNet . . . . .                                     | 78 |
| 4.3.5 | Neumann Network . . . . .                                       | 79 |

The second section of this chapter was published in a peer-reviewed journal:

**Zaccharie Ramzi**, P. Ciuciu and J. L. Starck. “Benchmarking MRI reconstruction neural networks on large public datasets”. In: *Applied Sciences (Switzerland)* 10.5 (2020)

This work was also presented in an international peer-reviewed conference with proceedings:

**Zaccharie Ramzi**, P. Ciuciu and J. L. Starck. “Benchmarking Deep Nets MRI Reconstruction Models on the Fastmri Publicly Available Dataset”. In:

THE main reason we want to use deep learning for MRI reconstruction is that it should allow us to learn a prior on MR images from k-space data which is more faithful and thus more powerful than the usual fixed sparsifying transforms. However, there are many ways to tackle the learning of this prior and use it afterwards in the reconstruction process. In this chapter we will present a selection of works that have used deep learning in different ways to tackle the problem of MRI reconstruction and compare some of them together.

## 4.1 . Paradigms for deep learning use in MRI reconstruction

In this section, we intentionally omit the unrolled framework which we will discuss in depth in the following sections. In order to get an in-depth view of some of the points discussed here, we recommend the review written by Ongie et al. [Ong+20]. Additionally, it is worth mentioning that in this section we do not tackle some specific issues related to MRI reconstruction such as multi-contrast or dynamic image reconstruction, unpaired datasets, motion correction, B0 field inhomogeneities or gradient inaccuracies.

### 4.1.1 . Plug-and-Play

The idea behind Plug-and-Play (P&P) did not appear with deep learning. It was originally introduced by Venkatakrishnan et al. [VBW13] where the focus was on existing denoising algorithms like BM3D [Dab+06], which were not deep learning based. The key observation is that the proximal operator involved in Equation 2.9 plays the role of a denoiser. Therefore, one could try to replace this proximal operator with a performant denoiser, which does not necessarily match the proximal operator of a regularization term.

This idea can then be used with *trained* denoisers which in practice are outperforming *classical* denoisers. This was implemented in many recent works [Zha+17b; ZZZ19; MMC17; Ryu+19; Xu+20], some even using Reinforcement Learning to tune the parameters of the optimization algorithm [Wei+20].

**Bayesian view and score.** The P&P approach can be well understood by considering the Bayesian formulation of inverse problems and score matching. The Bayesian view of inverse problems reads the following:

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) \quad (4.1)$$

where  $\mathbf{x}$  is the object (i.e. signal or image) to recover and  $\mathbf{y}$  are the noisy measurements. In Equation 4.1  $p(\mathbf{y}|\mathbf{x})$  stands for the likelihood function, i.e. the

probability of observing the actual measurements  $\mathbf{y}$  given the object  $\mathbf{x}$  and  $p(\mathbf{x})$  the prior distribution on  $\mathbf{x}$ .

We can use [Equation 4.1](#) to estimate the most likely  $\mathbf{x}$ , by maximizing the posterior distribution:

$$\hat{\mathbf{x}}^{\text{MAP}} = \arg \max_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) \quad (4.2)$$

$$= \arg \min_{\mathbf{x}} -\log p(\mathbf{x}|\mathbf{y}) \quad (4.3)$$

$$= \arg \min_{\mathbf{x}} [-\log p(\mathbf{x}) - \log p(\mathbf{y}|\mathbf{x})] \quad (4.4)$$

which is equivalent to [Equation 2.7](#) with  $-\log p(\mathbf{x}) = \lambda \|\psi \mathbf{x}\|_1$  and  $-\log p(\mathbf{y}|\mathbf{x}) = \sum_{l=1}^L \frac{1}{2} \|\mathbf{y}_l - \mathcal{F}_\Omega \mathbf{S}_l \mathbf{x}\|_2^2$ . In the Bayesian framework, assuming that  $\log p$  is differentiable, the [ISTA](#) step (i.e. [Equation 2.9](#)) is then rewritten as:

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + \epsilon_n \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})(\mathbf{x}_n) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \epsilon_n \nabla_{\mathbf{x}} \log p(\mathbf{x}_{n+1}) \end{aligned} \quad (4.5)$$

In [Equation 4.5](#) we clearly see that the proximal step is equivalent to the gradient step of the log prior over the object to recover. This term is called the score of the prior distribution. Interestingly, it has been shown that this score can be well approximated (up to a Gaussian kernel convolution) by a denoiser of the object of interest [[Hyv05](#); [AB13](#)].

#### 4.1.2 . Agnostic learning

From a machine learning perspective, one could consider the problem of [MRI](#) reconstruction a supervised regression problem (if given a sufficiently large database of fully-sampled [MR](#) images). The problem then reduces to finding the best function  $f$  that maps k-space data  $\mathbf{y}$  to [MR](#) images  $\mathbf{x}$ , by minimizing an  $\ell_2$  loss:  $\arg \min_f \frac{1}{2} \sum_i \|\mathbf{x}_i - f(\mathbf{y}_i)\|_2^2$ . This approach was taken by [Zhu et al.](#) [[Zhu+18](#)] in their seminal paper introducing [AUTOMAP](#).

Their model is very simple in that it consists in applying [Multi-Layer Perceptron \(MLP\)](#) followed by convolutions to the k-space data. Because it uses [MLP](#) on the full data, it does not scale well to images of high resolution, multicoil data or 3D. However, this work was instrumental in showing the promises that deep learning holds for [MRI](#) reconstruction.

#### 4.1.3 . Single-domain restoration

In [MRI](#) reconstruction, one has access to a naive reconstructor in the form of  $\mathcal{A}^H$  (taking the notations from [subsection 2.2.1](#)). Therefore, one can work on either restoring the k-space before the application of  $\mathcal{A}^H$  [[HSY19](#)], or one can work on restoring the aliased image  $\mathcal{A}^H \mathbf{y}$  [[Lee+18](#); [Han+18](#)]. The first approach will solve the problem  $\arg \min_f \frac{1}{2} \sum_i \|\mathbf{x}_i - \mathcal{A}^H f(\mathbf{y}_i)\|_2^2$  while the second will solve  $\arg \min_f \frac{1}{2} \sum_i \|\mathbf{x}_i - f(\mathcal{A}^H \mathbf{y}_i)\|_2^2$ . In effect, we can observe that those amount to rewriting agnostic learning where we introduce knowledge about the underlying physics to solve the reconstruction problem.

#### 4.1.4 . Adversarial reconstruction

In order to improve the results one might get by training a neural network in a supervised way, like in the single-domain restoration setting, it might be necessary to work on the loss function. Indeed, the  $\ell_2$  loss is just a proxy measure of the quality of the reconstruction, which in the end comes down to the appreciation by experts. A better proxy might be the SSIM [Wan+04] or the MSSIM [WSB03], as these metrics will be used to judge the reconstructions quantitatively. An even better solution is to use a neural network to determine if the image really looks like an MR image, and is typically free from common artifacts one might find with supervised reconstruction using the  $\ell_2$  loss. This can be achieved by adding an adversarial loss to the  $\ell_2$  loss, inspired by GANs [Goo+14] and recent results on conditional GANs [Zhu+17]. The problem to solve then becomes:

$$\arg \max_g \arg \min_f \mathbb{E} \|\mathbf{x} - f(\mathbf{y})\|_2^2 + \mathbb{E}_{\mathbf{x}} \log g(\mathbf{x}) - \mathbb{E}_{\mathbf{y}} \log g(f(\mathbf{y})) \quad (4.6)$$

This was successfully implemented for MRI reconstruction [MNJ18; Dra+17], although it is difficult to judge the results in practice as demonstrated by Hammernik et al. [Ham+19] given the lower performance on image quality quantitative metrics.

#### 4.1.5 . Deep Compressed Sensing

Another GAN-inspired technique is that of Deep CS [Bor+17; WRL19]. The idea is to train a GAN denoted  $g$  to reconstruct MR images from random vectors  $z$  drawn from a Gaussian distribution. We can then use this GAN as a prior to reconstruct the MR images, by solving the following equation:

$$\mathbf{z}^* = \arg \min_z \|\mathbf{y} - \mathcal{A}g(\mathbf{z})\|_2^2 \quad (4.7)$$

The resulting image is then  $\mathbf{x} = g(\mathbf{z}^*)$ .

This process has been further refined to train the generator to tackle specifically the inverse problem [WRL19] as well as the out-of-distribution problem [Dar+21].

#### 4.1.6 . Deep Image Prior

The Deep Image Prior (DIP) [UVL18] technique is based on the following assumption: the architecture of neural networks is already a sufficiently strong prior to represent natural images. The way this prior is used in inverse problems, is by solving the following problem:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathcal{A}f_{\boldsymbol{\theta}}(\mathbf{z})\|_2^2 \quad (4.8)$$

where  $z$  is just a random vector. The resulting image is then  $\mathbf{x} = f_{\boldsymbol{\theta}^*}(\mathbf{z})$ . In other words, DIP overfits a neural network with an appropriate architecture (typically a CNN) to fit the measurements.

This technique was applied to MRI reconstruction by Darestani et al. [DH20]. A great advantage of this technique is that it does not require any training data

except to validate the architecture of the neural network. In this sense, DIP stands out a bit in the MRI applications of deep learning, as it does not learn a prior from the data, but rather uses the architectural prior of CNNs.

**Inference time training.** The idea of inference time training has been used outside the context of DIP. The goal in this case is to slightly adapt the weights of the neural network to a potentially new distribution [Sun+20]. This was done in the context of MRI reconstruction by Hammernik et al. [Ham+19], and referred to as semi-supervised fine-tuning.

#### 4.1.7 . Self-supervised

Aside from DIP, most of the techniques introduced above require a substantial amount of fully-sampled (i.e. not accelerated or not undersampled) training data. Most of the time, only undersampled data are available because it is closer to the clinical routine. Therefore, techniques that can leverage such data are desired.

In MRI, such techniques will generally rely on partitioning the undersampled k-space data into 2 subsets. Yaman et al. [Yam+20] used one subset as the input to a reconstruction network whose goal is to infer the other subset, by first reconstructing the image and then using the measurement operator associated with this second subset. The problem to solve then reads:

$$\arg \min_f \|\mathbf{y}^{(2)} - \mathcal{A}_2 f(\mathbf{y}^{(1)})\|_2^2 \quad (4.9)$$

where  $\mathbf{y}^{(i)}$  denotes the  $i$ -th subset of the undersampled k-space data, and  $\mathcal{A}_2$  denotes the measurement operator associated with the second k-space subset. Hu et al. [Hu+21] reconstruct an image with independent networks for each of the two subsets and then make sure that the two reconstructions are consistent between them and with the original full measurements. The problem to solve is then the following:

$$\arg \min_{f,g} \|f(\mathbf{y}^{(1)}) - g(\mathbf{y}^{(2)})\|_2^2 + \alpha \|\mathbf{y} - \mathcal{A}f(\mathbf{y}^{(1)})\|_2^2 + \beta \|\mathbf{y} - \mathcal{A}g(\mathbf{y}^{(2)})\|_2^2 \quad (4.10)$$

where  $\alpha$  and  $\beta$  are hyperparameters. At test time one can use  $f$  or  $g$  to reconstruct the undersampled k-space data.

#### 4.1.8 . Implicit field learning

The implicit field learning technique has been popularized by recent works on scene rendering [Mil+20; Sit+20]. The concept of implicit field learning is to learn a point-by-point representation of an object. In the case of scene rendering, it can be a mapping from a 3D position in space and a viewing direction (as well as other parameters such as the light direction) to a color in RGB space and a density. The implicit field network  $f_\theta$  will then be learned by minimizing for a given field  $\mathbf{x}$ :

$$\|f_\theta(x, y, z) - \mathbf{x}[x, y, z]\|_2^2 \quad (4.11)$$

In MRI reconstruction, we have the option to learn one of two possible fields:

- the field of the measurements, i.e. the k-space;
- the field of the reconstructed image.

The measurements field is the one computed by Sun et al. [Sun+21], where the measurements field is not reconstructed in its entirety, but rather completed to be used in a subsequent reconstruction algorithm. In this case the implicit field is used to extrapolate unsampled k-space data. Shen et al. [SPX21] instead learn an implicit field on a reference image and fine-tune it at test time on a new image from the same patient. This idea can be applied to contexts where the same patient is scanned several times in a restricted amount of time, for example to follow up on the evolution of a condition or before and after contrast agent (e.g. Gadolinium) injection. A more complete survey on implicit fields for computer vision was recently carried out by Xie et al. [Xie+21].

## 4.2 . Benchmarking unrolled networks for MRI reconstruction

### 4.2.1 . Introduction

Some works [AÖ18b; Eo+18; Sch+18] have tried to inspire themselves from existing classical methods in order to leverage problem specific properties, but also from expertise gained in the field. However, they have not been compared against each other on a large dataset containing *complex-valued raw* data.

A recently published dataset, i.e. fastMRI [Zbo+18], allows this comparison, although it is still to be done and requires an implementation of the different networks in the same framework to allow for a fairer comparison in terms for example of runtime.

Our contribution is exactly this, that is:

- Benchmark different neural networks for MRI reconstruction on 2 datasets: the fastMRI dataset, containing raw complex-valued knee data, and the OASIS dataset [LaM+18] containing DICOM real-valued brain data.
- Provide reproducible code and the networks' weights<sup>1</sup>, using Keras [Cho+15] with a TensorFlow backend [Aba+16].

While our benchmark focuses on classical MRI modalities reconstruction, it is worth noting that other works have applied deep learning to other modalities like MR fingerprinting [VYL18] or diffusion-weighted MRI [AMJ20]. The networks studied here could be applied, but would not benefit from some invariants of the problem, especially in the fourth (contrast-related) dimension introduced. We also specify that this benchmark was carried out in the 2D single-coil Cartesian (i.e. simplest) setting.

---

<sup>1</sup>[github.com/zaccharieramzi/fastmri-reproducible-benchmark](https://github.com/zaccharieramzi/fastmri-reproducible-benchmark)

### 4.2.2 . Related works

In this section we briefly discuss other works presenting benchmarks on reconstruction neural networks.

Minh Quan et al. [MNJ18] benchmark their (adversarial training based) algorithms against classical methods and against Cascade-net (which they call Deep Cascade) [Sch+18] and ADMM-net (which they call DeepADMM) [Pui+16]. They train and evaluate quantitatively the networks on 2 datasets, selecting each time 100 images for train, 100 images for test:

- the IXI database<sup>2</sup> (brains);
- the Data Science Bowl challenge<sup>3</sup> (chests).

While these two datasets provide a sufficient number of samples to have a trustworthy estimate of the performance of the networks, they are not composed of raw complex-valued data, but of DICOM magnitude-only (i.e. positive real) data. Minh Quan et al. [MNJ18] evaluate their algorithms on a raw complex-valued dataset,<sup>4</sup> but it only features 20 acquisitions, and therefore the comparison is only done qualitatively.

Eo et al. [Eo+18] benchmark their algorithm against classical methods. They train and evaluate their network on 3 different datasets:

- the brain magnitude-only data set provided by the Alzheimer's Disease Neuroimaging Initiative (ADNI) [Pet+10];
- 2 proprietary datasets with raw complex-valued brain data.

Again, the only public dataset they used features magnitude-only data. Additionally, it is worth mentioning that their code is not open, i.e. it cannot be found online.

### 4.2.3 . Models

#### Baseline U-net

We use a U-net-like [RFB15] architecture as a baseline single-domain restoration network. This network was originally built for image segmentation, but has since been used for a wide variety of image-to-image tasks, mainly as a strong baseline. Han et al. [HSY19] used a U-net to apply on the undersampled k-space measurements before performing the inverse FT. Hyun et al. [Hyu+18] used a U-net to apply on the zero-filled reconstruction and correct the output of the U-net with a DC step (where they replace sampled values in the k-space). The network we implemented was however vanilla, without this extra DC step. Our implementation features the following cascade of number of filters: 16, 32, 64, 128. The original U-net is illustrated in Figure 4.2-1 where the number of filters used in each layer is 4 times what we used.

---

<sup>2</sup>[brain-development.org/ixi-dataset/](http://brain-development.org/ixi-dataset/)

<sup>3</sup>[kaggle.com/c/second-annual-data-science-bowl/data](https://kaggle.com/c/second-annual-data-science-bowl/data)

<sup>4</sup>[mridata.org/list?project=Stanford%20Fullysampled%203D%20FSE%20Knees](http://mridata.org/list?project=Stanford%20Fullysampled%203D%20FSE%20Knees)



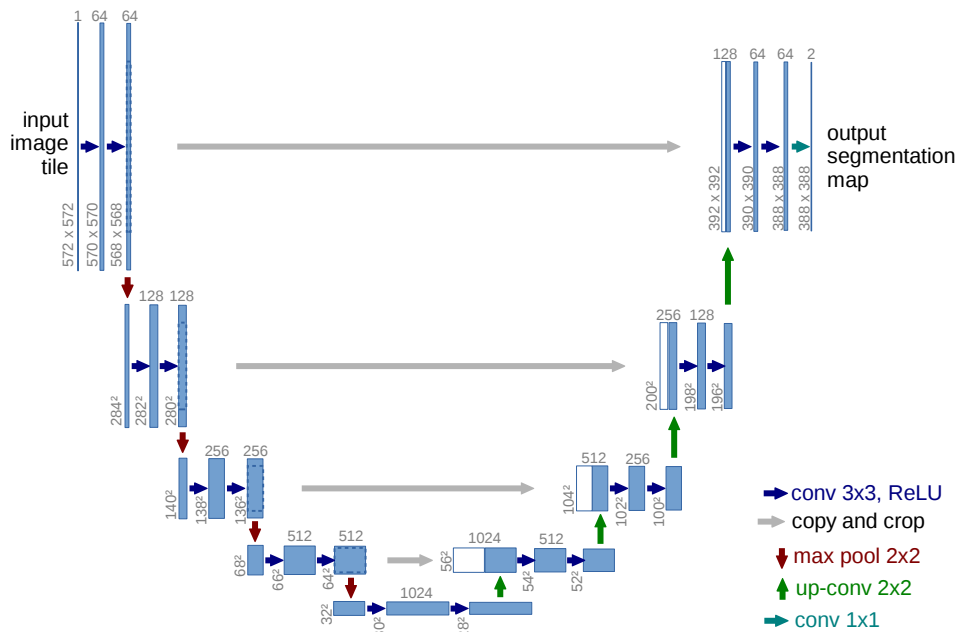


Figure 4.2-1: **Illustration of the U-net**, courtesy of Ronneberger et al. [RFB15]. In our case the output is not a segmentation map but a reconstructed image of the same size (we perform zero-padding to prevent decreasing sizes in convolutions).

## Unrolled networks

The second class of networks we introduce, are unrolled networks (or cross-domain networks). The key intuitive idea is that they correct the data in both the  $k$ -space and the image space alternatively, using the FT to go from one space to the other. They are derived from the optimization algorithms used to solve the optimization problems introduced before, using the idea of “unrolling” introduced by Gregor et al. [GL10]. An illustration of this class of networks is presented in Figure 4.2-2.

As these networks work directly on the input data (and not on a primarily reconstructed version of it), they need to handle complex-valued data. In particular, the classical deep learning frameworks (TensorFlow and Pytorch) do not feature the ability to perform complex convolutions off-the-shelf. The way convolution is performed in the original papers is therefore to concatenate the real and imaginary parts of the image (respectively the  $k$ -space), making it a 2-channel image, perform the series of convolutions, and have the output be a 2-channel image then transformed back in a complex image (respectively  $k$ -space).

The **Cascade-net** [Sch+18] is based on the dictionary learning optimization problem (2.12). The idea is to replace the dictionary learning step by convolutional neural networks and still keep the data consistency step in the  $k$ -space. The optimization algorithm is then unrolled to allow the back-propagation to be performed.

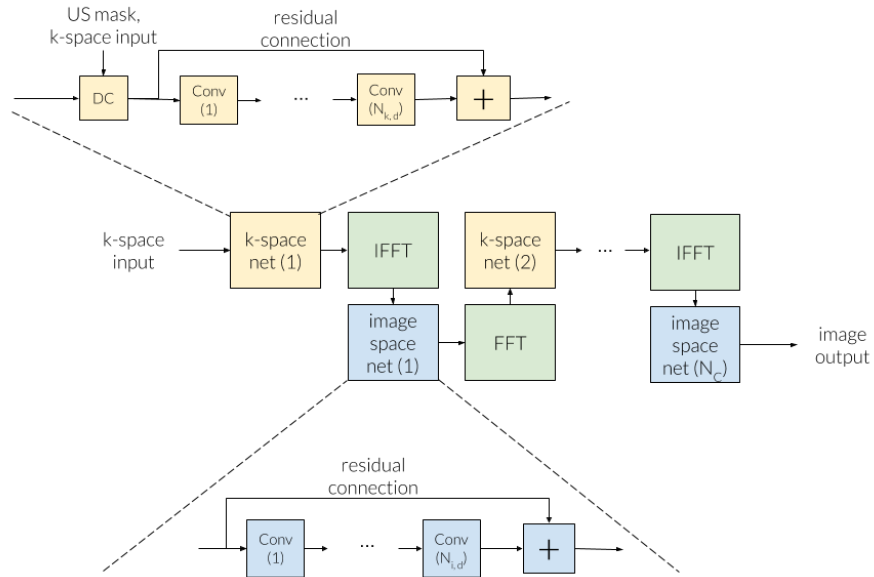


Figure 4.2-2: **Unrolled networks.** The common backbone between the Cascade net, the KIKI-net and the PD-net. US mask stands for undersampling mask. DC stands for data consistency. (I)FFT stands for (Inverse) Fast Fourier Transform.  $N_{k,d}$  is the number of convolution layers applied in the k-space.  $N_{i,d}$  is the number of convolution layers applied in the image space.  $N_C$  is the total number of alternations between the k-space and the image-space. It is worth mentioning that in the case of PD-net, the data consistency step is not performed with a replacement operator but with a residual, the Fourier operators are carried out with the original undersampling mask, and a buffer is concatenated along with the current iteration to allow for some memory between iterations and learn the acceleration (in the k-space net -dual net- it is also concatenated with original k-space input). In the case of the Cascade net,  $N_{k,d} = 0$ , only the data consistency is performed in the k-space. In the case of the KIKI-net, there is no residual connection in the k-space. However, the k-space and image space nets could potentially be any kind of image-to-image neural network.

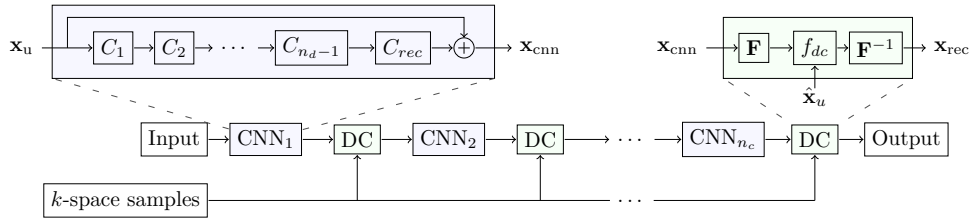


Figure 4.2-3: **Illustration of the Cascade-net**, courtesy of Schlemper et al. [Sch+18]. Here each  $C_i$  is a convolutional block of 64 filters (48 in our implementation) followed by a **ReLU** nonlinearity,  $n_d$  is the number of such convolutional blocks forming a convolutional subnetwork between each data consistency layer  $DC$ , and  $n_c$  is the number of convolutional subnetworks.

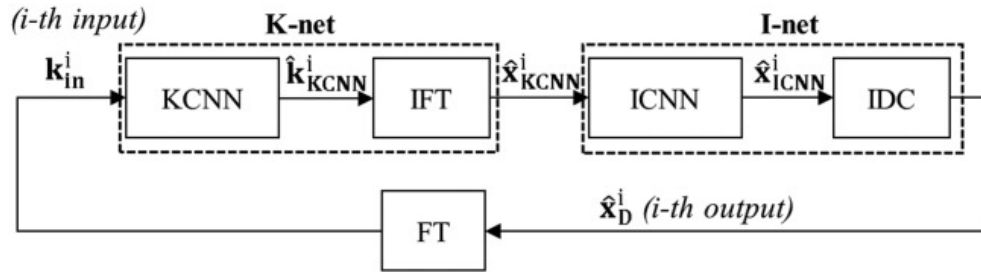


Figure 4.2-4: **Illustration of the KIKI-net**, courtesy of Eo et al. [Eo+18]. The KCNN and ICNN are convolutional neural networks composed of a number of convolutional blocks varying between 5 and 25 (we implemented 25 blocks for both KCNN and ICNN), each followed by a **ReLU** nonlinearity and featuring between 8 and 64 filters (we implemented 32 filters). For both the varying numbers, Eo et al. [Eo+18] show that the higher, the better. The ICNN also features a residual connection.

Schlemper et al. [Sch+18] show that we can perform back-propagation through the data consistency step (which is linear), and derive the corresponding Jacobian. The parameters used here for the implementation are the same as those in the original paper, except the number of filters which was decreased from 64 to 48 to fit on a single GPU. This network is illustrated in Figure 4.2-3.

The **KIKI-net** [Eo+18] is an extension of the Cascade-net where they additionally perform convolutions after the data consistency step in the  $k$ -space. The parameters used here for the implementation are the same as those in the original paper. This network is illustrated in Figure 4.2-4.

The **Primal-Dual-net (PD-net)** was introduced by Adler et al. [AÖ18b] and applied to MRI by Cheng et al. [Che+19]. It is based on the resolution of Equation 2.7 with the PDHG [CP11] algorithm. Here this algorithm is unrolled and the proximity operators (present in the general case of PDHG) are replaced by convo-

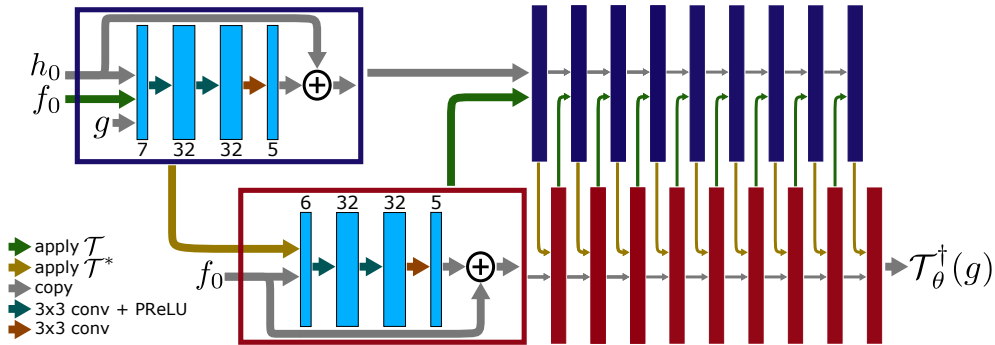


Figure 4.2-5: **Illustration of the PD-net**, courtesy of Adler et al. [AÖ18b]. Here  $\mathcal{T}$  denotes the measurement operator, which in our case is the undersampled FT,  $\mathcal{T}^*$  its adjoint,  $g$  is the measurements, which in our case are the undersampled k-space measurements, and  $f_0$  and  $h_0$  are the initial guesses for the direct and measurement spaces (the image and k-space in our case). The initial guesses are zero tensors. Because we transform complex-valued data into 2-channel real-valued data, the number of channels at the input and the output of the convolutional subnetworks are multiplied by 2 in our implementation.

lutional neural networks. In our implementation, for a fairer comparison with the Cascade-net and U-net, we used a ReLU nonlinearity instead of a PReLU [He+15]. This network is illustrated in Figure 4.2-5.

## Training

The training was done with the same parameters for all the networks. The optimiser used was Adam [KB15], with a learning rate of  $10^{-3}$  and default parameters of Keras ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , the exponential decay rates for the moment estimates). The gradient norm was clipped to 1 to avoid the exploding gradient problems [PMB13]. The batch size was 1 (i.e. one slice) for every network except the U-Net where the whole volume was used for each step. For all networks, to maximise the efficiency of the training, the slices were selected in the 8 innermost slices of the volumes, because the outer slices do not have much signal. No early stopping or learning rate schedule was used (except for KIKI-net to allow for a stable training where we used the learning rate schedule proposed by Eo et al. [Eo+18]). The number of epochs used was 300 for all networks trained end-to-end. For the iterative training of the KIKI-net, the total number of epochs was 200 (50 per subtraining). Batch normalization was not used, however, in order to have the network learn more efficiently, a scaling of the input data was done. Both the k-space and the image were multiplied by  $10^6$  for fastMRI and by  $10^2$  for OASIS, because the k-space measurements had values of mean  $10^{-7}$  (looking separately at the real and imaginary parts) for fastMRI and of mean  $10^{-3}$  for OASIS.

Without this scaling operation, the training proved to be impossible with bias in the convolutions and very inefficient without bias in the convolutions.

#### 4.2.4 . Data

##### Undersampling

The undersampling was done retrospectively using a Cartesian mask described in the data set paper [Zbo+18], and an AF of 4 (i.e. only 25% of the k-space was kept). It contains a fully-sampled region in the lower frequencies, and randomly selects phase encoding lines in the higher frequencies.

It is to be noted that different undersampling strategies exist in CS-MRI. Some of them are listed by Chauffert et al. [Cha+14], like for example spiral or radial. These strategies allow for a higher image quality while having the same AF or the same image quality with a higher AF. Typically, the spiral undersampling scheme was designed to allow fast coronary imaging [IN95; Mey+92]. These undersampling strategies must take into account kinematic constraints (both physically and safety based), but also should also be with variable density [Cha+14]. Recent works even try to optimize the undersampling strategy under these kinematic constraints [Laz+19]. Others have tried to learn the undersampling strategy in a supervised way. Sanchez et al. [San+20a] learned the undersampling strategy with a greedy optimization. Sherry et al. [She+20] used a gradient descent optimization. Some approaches [AMJ20; WRL19; Wei+21] even try to learn jointly the optimal undersampling strategy along with the reconstruction.

##### fastMRI

The data used for this benchmark is the emulated single-coil k-space data of the fastMRI knee dataset [Zbo+18], along with the corresponding ground truth images. The acquisition was done with a 15-channel phased array coil, in Cartesian 2D Turbin Spin Echo (TSE). The pulse sequences were proton-density weighting, half with fat suppression, half without, some at 3.0 T others at 1.5 T. The sequence parameters were as follows: Echo train length 4, matrix size  $320 \times 320$ , in-plane resolution  $0.5\text{mm} \times 0.5\text{ mm}$ , slice thickness 3 mm, no gap between slices. In total, there are 973 volumes (34, 742 slices) for the training subset and 199 volumes (7135 slices) for the validation subset.

Since the k-spaces are of different sizes, therefore resulting in images of different sizes, the outputs of the unrolled networks were cropped to a central  $320 \times 320$  region. For the U-net, the input of the network was cropped.

##### OASIS

The Open Access Series of Imaging Studies (OASIS) brain database [LaM+18] is a database including MRI scans of 1068 participants, yielding 2168 MR sessions. Of these 2168, we select only 2164 sessions which feature T1-weighted sequences.

1878 of these were acquired on a 3.0 T 236 at 1.5 T and the remaining are undisclosed (50). The slice size is majorly  $256 \times 256$ , and sometimes  $240 \times 256$  (rarely it can be some other sizes). The number of slices per scan is majorly 176, and sometimes 160 (rarely it can be smaller).

The data was then separated in a training and a validation set. The split was participant-based, that is a participant cannot have a scan in both sets. The split was of 90% for the training set and 10% for the validation set. We further reduced the training data to make it comparable to fastMRI, to 1000 scans randomly selected for the training subset and 200 scans randomly selected for the validation subset.

Contrarily to fastMRI, the OASIS data is available only in magnitude and therefore is only real-valued. The k-space is computed as the inverse FT of the magnitude image.

#### 4.2.5 . Results

##### Metrics

In order to evaluate the reconstruction quality of the networks, we used the PSNR and SSIM, both of which are discussed in subsection 2.4.1. As mentioned in subsection 2.4.3, we compute these metrics volume-wise even though our reconstruction setup is 2D.

While the 2 aforementioned metrics control the reconstruction quality, it is important to note that this is not the only factor to take into account when designing reconstruction techniques. Because the reconstruction has to happen fast enough for the MR physician to decide whether to re-conduct the exam or not, it is important for the proposed technique to have a reasonable reconstruction speed. For real-time MRI applications or dynamic MRI (e.g. cardiac imaging), it is even more important (for example in the context of monitoring surgical operations [Hor+07]). The runtimes were measured on a computer equipped with a single GPU Quadro P5000 with 16 GB of RAM.

Concurrently, the number of parameters has to stay relatively low to allow the implementation on the different machines with potentially limited memory, which will probably need to have multiple models (for various imaging contrasts, organs or undersampling schemes including several AFs).

In summary, we use 4 metrics to compare the different approaches:

- PSNR;
- SSIM;
- reconstruction time;
- number of parameters.

## Quantitative results

Table 4.1: **Quantitative results for the fastMRI dataset.** PSNR and SSIM mean and standard deviations are computed over the 200 validation volumes. Runtimes are given for the reconstruction of a volume with 35 slices.

| Network            | PSNR-mean (std) (dB) | SSIM-mean (std)     | #params | Runtime (s) |
|--------------------|----------------------|---------------------|---------|-------------|
| <b>Zero-filled</b> | 29.61 (5.28)         | 0.657 (0.23)        | 0       | 0.68        |
| <b>KIKI-net</b>    | 31.38 (3.02)         | 0.712 (0.13)        | 1.25M   | 8.22        |
| <b>U-net</b>       | 31.78 (6.53)         | 0.720 (0.25)        | 482k    | <b>0.61</b> |
| <b>Cascade net</b> | 31.97 (6.95)         | 0.719 (0.27)        | 425k    | 3.58        |
| <b>PD-net</b>      | <b>32.15 (6.90)</b>  | <b>0.729 (0.26)</b> | 318k    | 5.55        |

The quantitative results in Table 4.1–Table 4.4 show that the PD-net [AÖ18b] outperforms its competitors in terms of image quality metrics but also has the least amount of trainable parameters. It is slightly slower than the Cascade-net [Sch+18] though which can be explained by its higher number of iterations, involving therefore more costly FT (inverse or direct) operations. These results hold true on the 2 data sets, fastMRI [Zbo+18] and OASIS [LaM+18]. The only exception is that KIKI-net [Eo+18] is slightly better than the U-net [RFB15] on the OASIS data set, but still far from the best performers. We can also note that the standard deviation of the image quality metrics is way higher in the fastMRI data set than in the OASIS data set. This higher standard deviation is explained by the fact that the 2 contrasts present in the fastMRI dataset, Proton Density with and without Fat Suppression (PD/PDFS), have widely different image metrics values. The standard deviations when we compute the metrics for each contrast separately are more in-line with the OASIS ones. The range of the image quality metrics is also much higher in the OASIS results.

Table 4.2: **Quantitative results for the fastMRI dataset with the Proton-Density with Fat Suppression (PDFS) contrast.** PSNR and SSIM mean and standard deviations are computed over the 99 validation volumes. Runtimes are given for the reconstruction of a volume with 35 slices.

| Network            | PSNR-mean (std) (dB) | SSIM-mean (std)      | # params | Runtime (s) |
|--------------------|----------------------|----------------------|----------|-------------|
| <b>Zero-filled</b> | 28.44 (2.62)         | 0.578 (0.095)        | 0        | 0.41        |
| <b>KIKI-net</b>    | 29.57 (2.64)         | 0.6271 (0.10)        | 1.25M    | 8.88        |
| <b>Cascade-net</b> | 29.88 (2.82)         | 0.6251 (0.11)        | 425K     | 3.57        |
| <b>U-net</b>       | 29.89 (2.74)         | 0.6334 (0.10)        | 482K     | <b>1.34</b> |
| <b>PD-net</b>      | <b>30.06 (2.82)</b>  | <b>0.6394 (0.10)</b> | 318K     | 5.38        |

Table 4.3: **Quantitative results for the fastMRI dataset with the Proton-Density (PD) contrast.** PSNR and SSIM mean and standard deviations are computed over the 100 validation volumes. Runtimes are given for the reconstruction of a volume with 40 slices.

| Network     | PSNR-mean (std) (dB) | SSIM-mean (std)      | # params | Runtime (s) |
|-------------|----------------------|----------------------|----------|-------------|
| Zero-filled | 30.63 (2.1)          | 0.727 (0.087)        | 0        | 0.52        |
| KIKI-net    | 32.86 (2.4)          | 0.797 (0.082)        | 1.25M    | 11.83       |
| U-net       | 33.64 (2.6)          | 0.807 (0.084)        | 482K     | <b>1.07</b> |
| Cascade-net | 33.98 (2.7)          | 0.811 (0.086)        | 425K     | 4.22        |
| PD-net      | <b>34.2 (2.7)</b>    | <b>0.818 (0.084)</b> | 318280   | 6.08        |

Table 4.4: **Quantitative results for the OASIS dataset.** PSNR and SSIM mean and standard deviations are computed over the 200 validation volumes. Runtimes are given for the reconstruction of a volume with 32 slices.

| Network     | PSNR-mean (std) (dB) | SSIM-mean (std)       | # params | Runtime (s)  |
|-------------|----------------------|-----------------------|----------|--------------|
| Zero-filled | 26.11 (1.45)         | 0.672 (0.0307)        | 0        | 0.165        |
| U-net       | 29.8 (1.39)          | 0.847 (0.0398)        | 482k     | <b>1.202</b> |
| KIKI-net    | 30.08 (1.43)         | 0.853 (0.0336)        | 1.25M    | 3.567        |
| Cascade-net | 32.0 (1.731)         | 0.887 (0.0327)        | 425k     | 2.234        |
| PD-net      | <b>33.22 (1.912)</b> | <b>0.910 (0.0358)</b> | 318k     | 2.758        |

Reference Zero-filled KIKI-net U-net Cascade-net PD-net

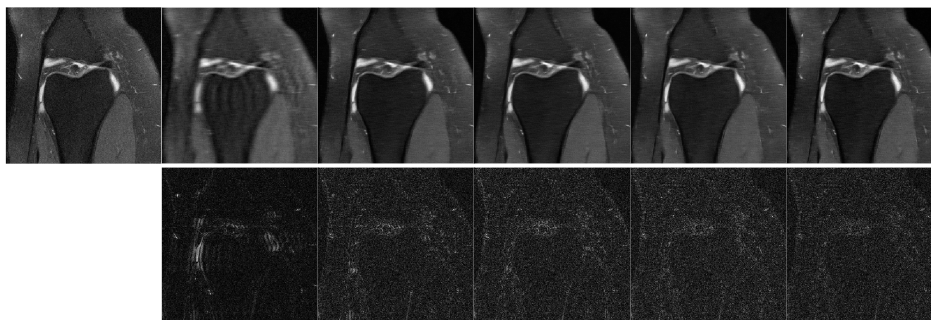


Figure 4.2-6: **Reconstruction results for a specific slice (16th slice of file1000196, part of the validation set).** The first row represents the reconstruction using the different methods, while the second represents the absolute error when compared to the reference.



**Reference Zero-filled KIKI-net U-net Cascade-net PD-net**

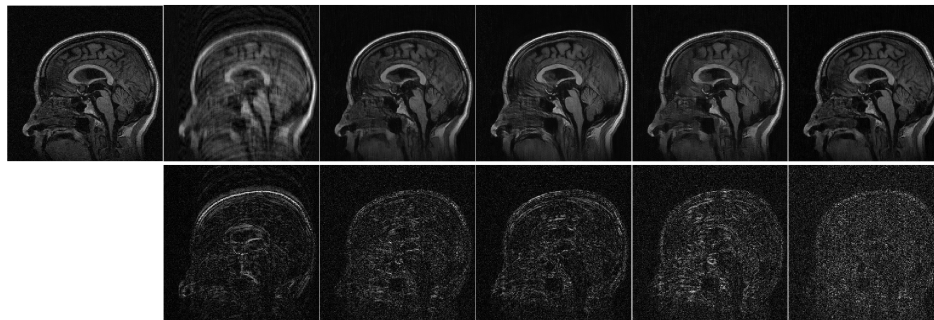


Figure 4.2-7: **Reconstruction results for a specific slice (15th slice of sub-OAS30367\_ses-d3396\_T1w.nii.gz, part of the validation set).** The top row represents the reconstruction using the different methods, while the bottom row represents the absolute error when compared to the reference.

### Qualitative results

The qualitative results shown in Figures 4.2-6 – 4.2-7 confirm the quantitative ones on the image quality aspect. The PD-net [AÖ18b] is much better at conserving the high-frequency parts of the original image, as can be seen when looking at the reconstruction error, which is quite flat over the whole image.

#### 4.2.6 . Discussion

This section tackled the important task of comparing recent deep learning approaches for MRI reconstruction on a sizeable dataset. The results suggest that for unrolled networks, the tradeoff between a high number of iterations and a richer correction in a certain domain (by having deeper networks) is in favor of having more iterations (i.e. alternating more between domains). It is however unclear how to best address the reconstruction in the k-space, since the convolutional networks make a shift invariance hypothesis which is not tenable in the Fourier space where the coefficients corresponding to the high frequencies should probably not be treated in the same way as with the low frequencies. This leaves room for improvement in the near future.

Due to a lack of computing resources, not all unrolled networks and their variants could be tested, and therefore this benchmark can be judiciously complemented with the work of Hammernik et al. [Ham+19]. The latter has not looked at the tradeoff between image correction subnetwork size and number of unrolled steps. It would be however interesting to extend this benchmark to include the remaining unrolled networks.

Although this benchmark was carried out in a simple acquisition setting, it gives us grounds on how to most efficiently build networks that will perform well in more challenging settings, like multicoil, non-Cartesian sampling and 3D imaging.

### 4.3 . Unrolled networks for MRI reconstruction

In this section we enumerate the different unrolled networks for MRI reconstruction that were not covered by our benchmark.

#### 4.3.1 . Model-based Deep Learning

Model-based Deep Learning (MoDL) [AMJ19] is the network obtained when unrolling the Conjugate Gradient (CG) algorithm. Its iterations read as follows:

$$\mathbf{x}_{n+1} = \arg \min_{\mathbf{x}} \|\mathcal{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x} - \mathbf{z}_n\|_2^2 \quad (4.12)$$

$$\mathbf{z}_{n+1} = f_{\theta}(\mathbf{x}_n) \quad (4.13)$$

Two important observations can be made regarding this model:

- Weight sharing is used between the different unrolled blocks.
- The data consistency layer is written as an optimization problem. In practice this is solved via the CG algorithm, and the backpropagation is feasible as the optimization problem has a closed form solution involving a large matrix inversion. The authors claim that this way of imposing data consistency is more accurate therefore reducing the number of necessary unrolled steps.

Aggarwal et al. [AMJ19] carried out ablation studies to evaluate the importance of both aspects, showing that they were indeed providing a better performance in their setting. Additionally, it can be hypothesized from their study that the reason why weight sharing works well in this context is because they used a relatively small dataset.

#### 4.3.2 . Variational Network

The VarNet [Ham+18] is obtained when unrolling the Gradient Descent (GD) algorithm derived from the classical MRI reconstruction optimization problem where the regularization is a Field of Experts (FoE) model. Its iterations read as follows:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \sum_l f_{\theta_{l,n}}(\mathbf{x}_n) - \lambda_n \mathcal{A}^{\top}(\mathcal{A}\mathbf{x}_n - \mathbf{y}) \quad (4.14)$$

where  $f_{\theta_{l,n}}$  is a 2-layer convolutional neural network with a Gaussian Radial Basis Function (RBF) with learned parameters as activation function. Hammernik et al. [Ham+18] also add constraints on the convolution kernels to make sure that they respect the condition of being derived from an FoE:

- The kernels have to be zero-mean.
- The kernels have to be of unit norm.

### 4.3.3 . $\Sigma$ -Net

The  $\Sigma$ -Net [Ham+19] is actually an ensemble of unrolled networks. The iterations are obtained by unrolling ISTA steps:

$$\begin{aligned}\mathbf{x}_{n+\frac{1}{2}} &= \mathbf{x}_n - f_{\theta_n}(\mathbf{x}_n) \\ \mathbf{x}_{n+1} &= \mathbf{x}_{n+\frac{1}{2}} - \frac{\eta_n}{\lambda_n} \mathcal{A}^\top (\mathcal{A} \mathbf{x}_{n+\frac{1}{2}} - \mathbf{y})\end{aligned}\tag{4.15}$$

Hammernik et al. [Ham+19] then ensemble 3 types of network:

- one trained classically;
- one trained with an adversarial loss;
- one trained classically and using inference time training.

This allows them to keep the performance of the vanilla network while also improving the texture of the reconstructed images. They also perform an extensive ablation study to understand the impact of the following blocks on the performance:

- They looked at the impact of the Data Consistency layer. They show that the impact of the MoDL [AMJ19] data consistency step is minimal, and that sticking with the descent step is a decent option. Additionally, they demonstrated that using data consistency is essential compared to just cascading image enhancement networks.
- They looked at the impact of considering that  $\mathbf{x}$  is a single aggregated image (what they term Sensitivity Networks) or multicoil images (what they term Parallel Coil Networks or **Parallel Coil Network (PCN)**). They showed that this choice is not significant, although they noted that **PCN** do not require sensitivity maps to be extracted. The authors did not however test the impact of coil configuration on **PCN**.
- They finally benchmarked the different learning paradigms, and showed that adversarial training and inference time training can help the reconstruction texture and qualitative performance but will degrade the quantitative performances (as expected).

### 4.3.4 . End-to-end VarNet

The essential addition of the End-to-End VarNet [Sri+20] is the sensitivity maps estimation. Indeed, most works prior to it [Ham+18; AMJ19; Ham+19] relied on ESPIRiT [Uec+14] to extract the sensitivity maps  $\mathcal{S}$ . Sriram et al. [Sri+20] proposed to use a **CNN** to refine sensitivity maps coarsely estimated by the inverse **FT** of the masked k-space data in the low frequencies. This sensitivity maps refinement module is embedded in the unrolled network in an end-to-end manner and shared across coils.

This work highlighted how deep learning can help in more than just learning a prior over the object (here an image) to recover in inverse problems: it can also help refine our knowledge of the measurements operator.

#### 4.3.5 . Neumann Network

The Neumann Network [GOW19] is obtained when unrolling the Neumann series used to solve the optimization problem  $\arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathcal{A}\mathbf{x}\|_2^2 + R(\mathbf{x})$  when  $R$  is a quadratic term. Its iterations can be classically written as follows:

$$\mathbf{x}_{n+1} = (\mathbf{I} - \eta \mathcal{A}^\top \mathcal{A}) \mathbf{x}_n - \eta f_{\theta}(\mathbf{x}_n) \quad (4.16)$$

The final output of the network is actually the sum of all the unrolled blocks outputs:  $\sum_n \mathbf{x}_n$ .

Gilton et al. [GOW19] also proposed a preconditioning mechanism using  $(\mathcal{A}^H \mathcal{A} + \lambda \mathbf{I})^{-1}$  and a CG step to compute it as introduced in Aggarwal et al. [AMJ19].

Interestingly, this unrolled network is the only one to not rely on a data consistency step, the measurements being used only at the network initialization stage:  $\mathbf{x}_0 = \eta \mathcal{A}^\top \mathbf{y}$ .

\* \* \*  
\* \*  
\*



## 5 - New unrolled networks for MRI reconstruction

### Chapter Outline

|       |                                     |     |
|-------|-------------------------------------|-----|
| 5.1   | XPDNet . . . . .                    | 82  |
| 5.1.1 | Introduction . . . . .              | 82  |
| 5.1.2 | Model . . . . .                     | 82  |
| 5.1.3 | Results . . . . .                   | 83  |
| 5.1.4 | Conclusion and Discussion . . . . . | 84  |
| 5.1.5 | Figures . . . . .                   | 84  |
| 5.2   | NC-PDNet . . . . .                  | 85  |
| 5.2.1 | Introduction . . . . .              | 85  |
| 5.2.2 | Related Works . . . . .             | 86  |
| 5.2.3 | Model . . . . .                     | 88  |
| 5.2.4 | Data . . . . .                      | 89  |
| 5.2.5 | Results . . . . .                   | 90  |
| 5.2.6 | Discussion and conclusion . . . . . | 113 |

The first section of this chapter was presented as an oral in a peer-reviewed conference:

**Zaccharie Ramzi**, P. Ciuciu and J.-L. Starck. “XPDNet for MRI Reconstruction: an application to the 2020 fastMRI challenge”. In: *ISMRM*. 2020, pp. 1–4. Oral

The second section of this chapter was accepted for publication in a peer-reviewed journal:

**Zaccharie Ramzi**, C. G R, J.-L. Starck and P. Ciuciu. “NC-PDNet: a Density-Compensated Unrolled Network for 2D and 3D non-Cartesian MRI Reconstruction”. In: *IEEE Transactions on Medical Imaging* (2022)

Parts of this second section were also presented in an international peer-reviewed conference with proceedings:

**Zaccharie Ramzi**, J. L. Starck and P. Ciuciu. “Density compensated unrolled networks for non-cartesian MRI reconstruction”. In: *Proceedings - International Symposium on Biomedical Imaging*. Vol. 2021-April. 2021, pp. 1443–1447

IN this chapter, we introduce two new unrolled networks for MRI reconstruction. Based on our benchmark, we based both these architectures on the PDNet [AÖ18b]. These architectures essentially extend the original PDNet to more challenging settings, first the 2D multicoil setting, and then the non-Cartesian acquisition setting all the way to 3D imaging.

## 5.1 . XPDNet

### 5.1.1 . Introduction

The fastMRI 2020 brain MRI reconstruction challenge was organized by Facebook and New York University (NYU), Langone Medical Center [Muc+21] in order to evaluate the performance of MRI reconstruction algorithms. To participate, we extended the promising PDNet architecture to the 2D multicoil setting, incorporating recent advances in denoising and MRI reconstruction along the way. The resulting architecture is called the XPDNet, where the X stands for the fact that it is implemented in a modular fashion where the image correction subnetwork can be replaced by any performing denoising network. In this section we will present this architecture and its results first on the validation set and then on the official challenge results.

### 5.1.2 . Model

#### **Unrolled networks.**

The general intuition behind unrolled networks is that we are going to alternate the correction between the image space and the measurements (i.e.  $k$ -space) space. The key tool for that is the unrolling of optimization algorithms introduced by Gregor et al. [GL10]. An illustration of what unrolled networks generally look like is provided in Figure 5.1-1.

#### **Unrolling the PDHG.**

The *XPDNet* is a particular instance of cross-domain networks. It is inspired by the PDNet introduced by Adler et al. [AÖ18b] by unrolling the PDHG algorithm [CP11]. In particular, a main feature of the PDNet is its ability to learn the optimization parameters using a buffer of iterates, here of size 5.

#### **Image correction network.**

The plain CNN is replaced by a Multiscale Wavelet CNN (MWCNN) [Liu+18], but the code<sup>1</sup> allows for it to be any denoiser, hence the presence of X in its name. We chose to use a smaller image correction network than that presented in the original paper [Liu+18], in order to afford more unrolled iterations in memory [ZCS20b].

---

<sup>1</sup>[github.com/zaccharieramzi/fastmri-reproducible-benchmark](https://github.com/zaccharieramzi/fastmri-reproducible-benchmark)

Additionally, because we use a small batch size, we removed batch normalization layers from the network.

### **k-space.**

In this challenge, since the data is multicoil, we did not use any k-space correction network which would be very demanding in terms of memory footprint. However, following the idea of Sriram et al. [Sri+20], we introduced a refinement network for  $S$ , initially estimated from the lower frequencies of the retrospectively undersampled coil measurements. This sensitivity maps refiner [Sri+20] is chosen to be a simple U-net [RFB15].

We therefore have 25 unrolled iterations, an **MWCNN** that has twice as fewer filters in each scale, a sensitivity maps refiner smaller than that of Sriram et al. [Sri+20] and no k-space correction network.

**Training details.** The loss used for the network training was a compound loss introduced by Pezzotti et al. [Pez+20], consisting of a weighted sum of an  $L_1$  loss and the **MSSIM** [Wan+04]. The optimizer was the **Rectified ADAM** (RAdam) [Liu+20] with default parameters.<sup>2</sup> The training was carried for 100 epochs (batch size of 1) and separately for **AFs** 4 and 8. The networks were then fine-tuned for each contrast for 10 epochs. Masks offset for the equidistant masks<sup>3</sup> are sampled on-the-fly. On a single V100 **GPU**, the training lasted 1 week for each acceleration.

**Data.** The network was trained on the brain part of the fastMRI dataset [Zbo+18]. The training set consists of 4,469 volumes from 4 different contrasts: T1, T2, **FLAIR** and T1 with admissions of contrast agent (labelled T1POST). The validation was carried over 30 contrast-specific volumes from the validation set.

#### **5.1.3 . Results**

**Quantitative.** We used the **PSNR** and **SSIM** metrics to quantitatively compare the reconstructed magnitude image and the ground truth. They are given for each contrast and for the 2 **AFs** in the Figs. 5.1-2- 5.1-3. Similar results are available on the public fastMRI leaderboard,<sup>4</sup> although generally slightly better. It is a bit difficult to consider these results when compared to only the zero-filled metrics, but these quantitative metrics do not accurately capture the performance of the **GRAPPA** algorithm [Gri+02]. However, at the time of submission, this approach ranks 2nd in the fastMRI leaderboards for the **PSNR** metric, and finished 2nd in the 4 $\times$  and 8 $\times$  tracks of the fastMRI 2020 brain reconstruction challenge [Muc+21].

---

<sup>2</sup>[tensorflow.org/addons/api\\_docs/python/tfa/optimizers/RectifiedAdam](https://tensorflow.org/addons/api_docs/python/tfa/optimizers/RectifiedAdam)

<sup>3</sup>To see more about the exact nature of the masks: [github.com/facebookresearch/fastMRI/issues/54](https://github.com/facebookresearch/fastMRI/issues/54)

<sup>4</sup>[fastmri.org/leaderboards](https://fastmri.org/leaderboards)



**Qualitative.** The visual inspection of the images reconstructed (available in Figure 5.1-2) at AF 4 shows little to no visible difference with the ground truth original image. However, when increasing the AF to 8, we can see that smoothing starts to appear which leads to a loss of structure as can be seen in Figure 5.1-3.

#### 5.1.4 . Conclusion and Discussion

We managed to gather insights from many works on computer vision and MRI reconstruction to build a modular approach. Currently, our solution XPDNet is among the best in PSNR and Normalized Mean Squared Error (NMSE) for both the multicoil knee and brain tracks at the AFs 4 and 8. Furthermore, the modularity of the current architecture allows us to use the newest denoising architectures when they become available. However, the fact that this approach fails to outperform the others on the SSIM metric is to be investigated in further work.

#### 5.1.5 . Figures

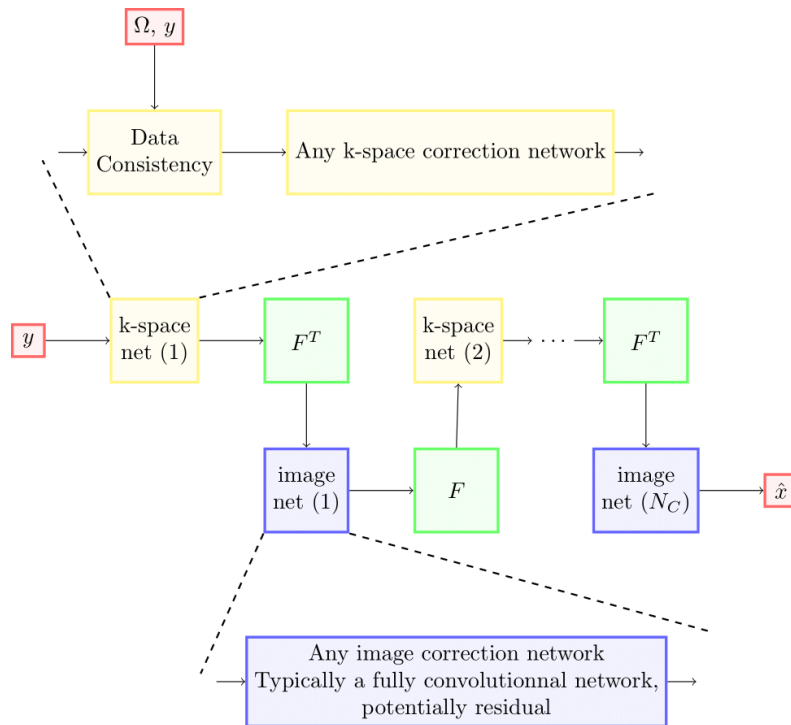


Figure 5.1-1: **General unrolled networks architecture.** Skip and residual connection are omitted for the sake of clarity.  $y$  are the under-sampled measurements, in our case the k-space measurements,  $\Omega$  is the undersampling scheme,  $F$  is the measurement operator, in our case the  $FT$ , and  $\hat{x}$  is the recovered solution.

## 5.2 . NC-PDNet

### 5.2.1 . Introduction

The most well-known MRI reconstruction methods are employed to reconstruct images from k-space data collected on a Cartesian grid, and therefore cannot be tested in the more general non-Cartesian acquisition setting, when the measurements are collected off the grid.

Non-Cartesian MRI acquisitions are of interest for multiple purposes. For example, the radial acquisition that happens with projection acquisition can be used for shorter echo times, reduced sensitivity to motion, and improved temporal resolution in some applications (e.g. angiography, dynamic MRI) [Pet+00; BKZ04a; Fen+14]. The same statement holds for spiral imaging that remains insensitive to in-plane flow-related artifacts [BKZ04a; NIM95]. As these setups are used in clinical routine, it is crucial to push the development of DL reconstruction methods for their application to these types of data.

Moreover, the interest in going from 2D to 3D imaging can be justified by a higher SNR requested for targeting higher resolution in clinical use [Blo+14]. Although multiple approaches exist to efficiently sample the 3D k-space [The+99; Fen+14; Laz+20; Cha+21], we will focus on full 3D radial undersampling as the stacking strategies (e.g. stack of stars) can be tackled using a 2D (i.e. slicewise) reconstruction method. However, due to hardware (i.e. GPU memory) constraints, we focus on single-coil 3D imaging rather than on multicoil 3D imaging, and leave this extension for future work.

We contributed to the domain of non-Cartesian MRI reconstruction by:

- introducing the *NC-PDNet* (i.e. Non-Cartesian PDNet), the first density-compensated unrolled neural network for non-Cartesian k-space data;
- implementing the first unrolled networks for non-Cartesian multicoil MRI 2D and single-coil MRI 3D data;
- open sourcing our implementation for all the networks, data processing, training, and evaluation<sup>5</sup> in TensorFlow [Aba+16], in particular a version of the *Nonuniform Fast Fourier Transform* (NUFFT) in the TensorFlow framework, usable for multicoil and 3D data (with the corresponding density compensation code);<sup>6</sup>
- performing some out-of-distribution performance tests for the trained networks in multiple acquisition scenarios and comparing them against the state of the art.

---

<sup>5</sup>[github.com/zaccharieramzi/fastmri-reproducible-benchmark](https://github.com/zaccharieramzi/fastmri-reproducible-benchmark)

<sup>6</sup>[github.com/zaccharieramzi/tfkbnuFFT](https://github.com/zaccharieramzi/tfkbnuFFT)

### 5.2.2 . Related Works

**Correcting k-space for non-Cartesian reconstruction.** This network strongly builds on the seminal paper by Pipe et al. [PM99] introducing [Density Compensation \(DCp\)](#)<sup>7</sup> for non-Cartesian k-space data. In this work, the authors provide a solution for computing the DCp which avoids some requirements that may be difficult to meet in practice, such as the knowledge of the k-space readouts or the constraint associated with the Nyquist criterion. However, this work does not use a learning based reconstruction strategy to complement the application of the DCp.

Some other works have explored other strategies to correct the k-space for non-Cartesian reconstruction, using a pre-conditioning of the k-space in iterative CS procedures in order to accelerate the convergence (not necessarily to obtain better image quality for an unlimited computational budget) [Trz+14]. To improve the latter approach, Ong et al. [OUL20] introduced a diagonal pre-conditioner optimized for the [Mean Squared Error \(MSE\)](#) that did not need an inner loop. However, these works use classical iterative algorithms which have been shown to be outperformed by end-to-end learned unrolled approaches in various settings [AÖ18b; AMJ19; Sch+18; Eo+18].

**Classical non-Cartesian reconstruction.** Classical reconstruction algorithms like SPIRiT and [GRAPPA](#) [Gri+02] have been adapted to the non-Cartesian setting and a summary of these adaptations was written by Wright et al. [Wri+14].

Another typical setting where non-Cartesian reconstruction is needed is dynamic MRI reconstruction. In this context, many works have used model-based optimization to reconstruct MR images from non-Cartesian acquisitions. The [Smoothness Regularization on Manifolds \(SToRM\)](#) method [PJ16] assumes that the different frames of the dynamic MRI acquisition evolve in the same low-dimensional manifold. The core idea is the following: As the underlying organ does not evolve too much in time, navigators (i.e. a calibration signal) can be used to recognize similar frames and build a Laplacian matrix from the latter in order to obtain a manifold regularizer. This idea is not applicable to the acquisition scenarios we cover in this paper as it strongly relies on the temporal nature of the signal to build a fitting regularization. In contrast, we consider here non-Cartesian reconstruction as a standalone task. Works derived from [SToRM](#) or similar to it [Pod+19; Nak+17; Ahm+19] also make use of the temporal nature of dynamic MRI to introduce a manually crafted regularization.

**Neural networks for non-Cartesian reconstruction.** The Nonuniform Variational Network [Sch+18] was the first unrolled network to be designed for non-Cartesian MRI reconstruction. This network unrolls a proximal gradient descent

---

<sup>7</sup>The acronym DC is usually used to denote Data Consistency in the MRI reconstruction field.

algorithm and operates on  $192 \times 192$  single-coil images with a variable density acquisition scheme. This network, however, does not include a DCp scheme, which is most likely due to the fact that the trajectories studied in this work do not present a variable density as recommended in the CS literature [Cha+14; Adc+17; Boy+16]. Additionally, this network was only applied to single-coil 2D data whose phase was simulated. Moreover, there is no open-source implementation for this work to date. In contrast, our network is applied to both single and multicoil 2D raw k-space data as well as to magnitude-only 3D data (i.e. no phase information available).

A perceptual complex neural network was designed by Shen et al. [She+21] to tackle 2D + time real-time cine MRI in cardiac imaging. However, this architecture is not an unrolled network and was applied to the gridding-reconstructed MR image. Shen et al. [She+21] specify that they did not consider data consistency-based methods due to the increased computation cost of the NUFFT. They actually pointed out that efficient tools like `torchknufft` [Muc+20] would be helpful in this regard. Here, we show that using a DCp step, a modeling choice allowed by an efficient implementation of the NUFFT, is actually critical to obtain improved results compared to baseline neural networks.

Similarly to our work, Malavé et al. [Mal+20] used an unrolled network to reconstruct non-Cartesian MR Angiography data. While they did use a density compensation scheme, they did not perform an ablation study, and did not use modern techniques for MRI unrolled networks such as buffers or sensitivity maps refinement. Furthermore, their code is not available online neither for the network, nor for the NUFFT.

Another direction of research for non-Cartesian MRI reconstruction is the use of untrained networks with the DIP [UVL18] framework. Yoo et al. [Yoo+21] designed a DIP approach tailored for dynamic MRI using a manifold to sample from for the random input. While this specificity is not applicable to our setting, it is a strong contender in its time independent (i.e. static) version, especially for the out-of-distribution performance evaluations. For this reason, we will perform a comparative analysis to the DIP framework.

**Neural networks for 3D reconstruction.** Küstner et al. [Küs+20] introduced a complex unrolled network for multicoil 3D + time reconstruction with two unrolled iterations for Cartesian acquisitions. However, due to the memory requirements of the NUFFT, it is not clear how to adapt this solution to the non-Cartesian case. Multicoil 3D MRI reconstruction is also tackled by Kellman et al. [Kel+20], who designed a memory-efficient algorithm to train unrolled networks, however again this approach was restricted to Cartesian acquisitions.

### 5.2.3 . Model

We use a classical unrolled network based on the PDNet to design our *NC-PDNet*. The major changes come from the fact that we use the non-uniform Fourier Transform as a forward operator.

## Non-Uniform Fourier Transform

The **NDFT** is the generalization of the discrete Fourier Transform to positions in the Fourier space that are off the Cartesian grid and not necessarily equispaced. An approximate algorithm to have an efficient computation of the **NDFT** was introduced by Fessler et al. [FS03] and Beatty et al. [BNP05]. This algorithm uses an oversampled grid and an optimal interpolation to perform the **NDFT** efficiently at the cost of an approximation. We refer to this algorithm as the **NUFFT** and highlight that unlike the **FFT**, it is not an exact algorithm.

While the **NUFFT** is a more efficient algorithm than the direct application of the definition of the **NDFT**, it is still a very computationally demanding algorithm compared to the **FFT** for the same image dimensions. An alternative to the **NDFT** could be to grid the k-space measurements and simply use the **FFT** to compute the data consistency. In practice, this would mean using  $\mathbf{y}_{grid} = grid(\mathbf{y}, \Omega)$  (where *grid* is a linear gridding operation) instead of  $\mathbf{y}$  and  $\mathcal{F}_\Omega$  would simply be the **FFT**.

**NDFT vs. NUFFT.** We draw the reader attention to the potential confusion between the two acronyms. The **NDFT** is the original transform, while the **NUFFT** is the approximate fast algorithm to compute the transform.

## Data Consistency

The data consistency is the step in the unrolled network allowing us to inject the initial measurements  $\mathbf{y}$  by comparing them to the current estimate's  $\mathbf{x}_b[0]$  measurements. The formula we have chosen for data consistency stems from the **Additive White Gaussian Noise (AWGN)** model, by taking the gradient of the  $\ell_2$ -norm in Equation 2.7:

$$\mathbf{x}_{dc} = \mathcal{A}^H \mathbf{d}(\mathcal{A}\mathbf{x}_b[0] - \mathbf{y}). \quad (5.1)$$

## Density Compensation

In a fully sampled setting, unlike the Cartesian case, the adjoint operator of the **NDFT** is not always its inverse operator. Worse, in most cases, the **NDFT** does not admit an inverse operator, even when the Nyquist criterion is met. The application of the adjoint Fourier operator  $\mathcal{F}_\Omega^H$  to the single coil k-space data (or its multicoil extension  $\mathcal{A}^H$  to the multicoil data) can therefore be very far from the solution to Equation 2.2.

To circumvent this, **DCp** has been introduced [PM99]. Indeed, the main problem with the classical non-Cartesian trajectories like radial, spiral or any variable density sampler [Cha+14], is that they densely sample the center of k-space. Therefore, when computing the adjoint operator ( $\mathcal{F}_\Omega^H$  or  $\mathcal{A}^H$ , respectively) a lot of weight is assigned to the densely sampled region at the center of the k-space, resulting in an image with abnormally large values. **DCp** is just the action of using factors that weigh the different sample locations so that they all play an even role during the application of the adjoint. **DCp** is particularly needed for deep learning approaches because the values entering convolutional layers need to have normalized (or close to normalized) values in order to avoid numerical issues. An interpretation of **DCp** is that it applies a preconditioning to the forward operator.

In practice, for both the radial and spiral trajectories, we obtain the **DCp** factors  $\mathbf{d}$  by applying the adjoint and forward operators<sup>8</sup> iteratively for  $N$  iterations, starting from  $\mathbf{d}_0 = \mathbf{1}$ :

$$\mathbf{d}_{n+1} = \frac{\mathbf{d}_n}{\mathcal{F}_\Omega \mathcal{F}_\Omega^H \mathbf{d}_n} \quad (5.2)$$

where the division is here pointwise. The final weights are  $\mathbf{d}_N$ . In practice, we took  $N = 10$ .

#### 5.2.4 . Data

We used the *NC-PDNet* on different data acquisition scenarios. In all of them, the data was retrospectively undersampled using fixed non-Cartesian trajectories. Specifically in this work we used multi-shot spiral and radial trajectories for 2D imaging, while we restricted our numerical studies to the use of full 3D radial spokes for 3D imaging. A schematic view (i.e. with a much higher undersampling of the trajectories) is available in Figs. 5.2-4 and 5.2-5. In 2D, most of the numerical experiments were conducted with an *AF* of 4, up to an exception where  $AF = 8$ , compared to the full Cartesian acquisition. This factor is defined as follows for 2D imaging:  $AF = \frac{N}{N_s}$  where  $n = N \times N$  is the image dimension ( $N$  is the base resolution) and  $N_s$  is the number of shots involved in the undersampling pattern. This basically means that  $m = n/4$  as we performed only retrospective studies.<sup>9</sup> For 3D imaging,  $AF = \frac{N \times N_z}{N_s}$  where  $n = N^2 \times N_z$  is the volume dimension and  $N_z$  the number of slices. In this setup, we also chose the value of  $AF = 4$ .

### fastMRI

A description of the fastMRI dataset is provided in [subsubsection 4.2.4](#). To obtain the non-Cartesian measurements  $\mathbf{y}$ , we simply used the inverse Fourier trans-

<sup>8</sup>In practice, we use the interpolation operator of the **NUFFT** rather than the whole operator.

<sup>9</sup>In **CS** prospective acquisitions, oversampling can be applied along each shot to fulfill Nyquist criteria without increasing the scan time. This would make the undersampling factor  $n/m$  lower compared to *AF*.

form (denoted as  $\mathbf{F}^\top$ ) of the full Cartesian k-space data  $\mathbf{y}_{or,\ell}$  for each coil  $\ell$  separately before applying the [NUFFT](#) according to the non-Cartesian undersampling pattern  $\Omega$  as follows:

$$\forall \ell = 1, \dots, L \quad \mathbf{y}_\ell = \mathcal{F}_\Omega \mathbf{F}^\top \mathbf{y}_{or,\ell} \quad (5.3)$$

We did not use the fastMRI data to perform the 3D experiments for the following reason. The fastMRI dataset was collected using a 2D multislice acquisition that uses anisotropic resolution to maintain a good [SNR](#) (in-plane resolution  $0.5\text{mm} \times 0.5\text{mm}$ , slice thickness  $3\text{mm}$ ). Consequently, the number of slices is very limited (roughly 30 to 40 for each acquisition) and the 3D networks will therefore not be able to fully take advantage of the third dimension.

## OASIS

A description of the [OASIS](#) dataset is provided in [subsection 4.2.4](#). We decided to use the [OASIS](#) dataset with data recast to matrix size of  $176 \times 256 \times 256$ , using zero-padding. Because [OASIS](#) data is magnitude only, the [NDFT](#) was applied to the magnitude volume, and we did not simulate the phase, a very challenging task according to Sandino et al. [[San+20b](#)]. The 3D fully-sampled dataset provided by Epperson et al. [[Epp+13](#)] was considered as a potential alternative as it contains raw data collected at isotropic resolution. Unfortunately it is not sizeable enough for deep learning tasks as specified by Zbontar et al. [[Zbo+18](#)]. Consequently, this basically means that the learned model from [OASIS](#) database cannot likely be used as is in a clinical setting, and would need to be further validated, even fine-tuned.

### 5.2.5 . Results

#### Experimental setup

**Metrics.** In order to evaluate the reconstruction quality of the networks, we used the [PSNR](#) and [SSIM](#), both of which are discussed in [subsection 2.4.1](#). As mentioned in [subsection 2.4.3](#), we compute these metrics volume-wise even when the reconstruction setup is 2D.

In order for the reader to get a better sense of the distribution of the quantitative image quality, we also provide detailed box-plots for all the quantitative results. For the sake of clarity, we chose to separate the different contrasts which present different quantitative metrics ranges in the box plots.

**Comparison.** To illustrate the need for all the key components of the *NC-PDNet*, we carried out an ablation study against other neural networks where we removed some aspects of the *NC-PDNet*.

- An unrolled network without [DCp](#) but only a normalization mechanism (without any normalization the network does not train due to very high values), to show the need for the [DCp](#).
- An unrolled network without [NUFFT](#) and instead a gridded version of the k-space (the operator is only the much faster [FFT](#)), to show the need for a better approximation of the [NDFT](#).
- A U-net [[RFB15](#)] applied directly to the density-compensated adjoint of the k-space measurements, to show the need for the unrolling framework (3D convolutions and pooling operations are used in the 3D setting).
- The density-compensated adjoint of the k-space measurements as very naive baseline.

For the 2D settings, the unrolled networks have  $N_C = 10$  unrolled iterations, use a buffer size  $N_P = 5$ , and a number of convolution filters  $N_f = 32$ . For the 3D setting, we needed to reduce those numbers for the network to fit on a single [GPU](#), and used  $N_C = 6$  unrolled iterations, a buffer size  $N_P = 2$ , and  $N_f = 16$ . The U-net was trained residually and had a base number of filters of 16.

We carried out the training and evaluation of the variants of the unrolled networks only in the 2D single-coil setting in order to save some computation time since we observed that the unrolled variants were not obtaining good performances.

We also compared to [DIP](#), an approach that was shown to reach state-of-the-art results in dynamic [MRI](#) reconstruction [[Yoo+21](#)], and that will help grasp better the generalization capacities of the networks as it is an untrained method. Importantly, in order to keep the reconstruction time reasonable, we used the method advocated by Darestani et al. [[DH20](#)]. This means that for a given contrast, [AF](#), trajectory and coil number, we reuse previously trained weights in order to initialize the weights of the [DIP](#). Similarly to prior observations [[DH20](#)], we found that this strategy allows us to reduce the number of epochs by a factor of 10. For the multicoil data, we also used the method advocated by Darestani et al. [[DH20](#)] to generate the coil images with the same network, and aggregate them using [Root-sum-of-squares \(RSS\)](#).

**Generalizability.** We also studied the generalizability of the trained networks to other settings, first to different sampling trajectories. In practice, we evaluated the networks trained on the spiral trajectory in the 2D multicoil setting on the radial undersampling pattern and vice-versa. We term this the *reverse setting*.

Second, we studied the generalizability of the networks to different organs and contrasts simultaneously. We took advantage of the 2D multicoil brain dataset available in fastMRI to carry out this analysis. This dataset features brain data acquired with four different weighting contrasts (T1, T2, [FLAIR](#) and T1 after Gadolinium injection), collected on the same scanners, which allows us to test for



Table 5.1: Mean PSNR / SSIM on the validation volumes of the different approaches averaged over both contrasts (knee fastMRI) in the single-coil setting. The best results are in bold font.

| Model                  | Radial                | Spiral                | # Parameters |
|------------------------|-----------------------|-----------------------|--------------|
| PDNet no DCp           | 27.02 / 0.6747        | 28.02 / 0.6946        | 156k         |
| Adjoint + DCp          | 27.11 / 0.6471        | 31.70 / 0.7213        | 0            |
| DIP                    | 29.57 / 0.5148        | 29.79 / 0.5249        | 0            |
| PDNet w gridding       | 31.12 / 0.6887        | 31.45 / 0.7126        | 156k         |
| U-net on Adjoint + DCp | 32.26 / 0.7224        | 32.82 / 0.7460        | 481k         |
| <b>NC-PDNet</b>        | <b>32.66 / 0.7327</b> | <b>33.08 / 0.7534</b> | 156k         |

generalizability solely on organ/contrast. Due to this large number of contrasts and the diversity in the number of coils for the brain data, we chose to not run the DIP experiments on brain data in order to keep the amount of computation reasonable.

Third, we tested the trained networks on an  $AF = 8$ , i.e. two times larger than the  $AF$  they were confronted in training.

**Computational efficiency.** The time spent for image reconstruction by a given method is also an important factor to take into account when choosing which algorithm to be used in the clinical realm. Indeed, in clinical applications we need to visualize the reconstructed MR images prior to the end of the exam to permit any potential re-start of the pulse MR sequence in case of the presence of significant artifacts impeding accurate image-based diagnosis. Additionally, this allows the physician to report on the scans directly to the patient once the exam is completed. For that reason, we decided to report the reconstruction times associated with the different methods and networks.

## Quantitative results

**2D single-coil fastMRI dataset.** The quantitative results in Table 5.1 show that *NC-PDNet* outperforms the concurrent approaches. Moreover, these results outline that combining unrolling and DCp with an accurate NDFT approximation (i.e. the NUFFT) within the same deep learning architecture is instrumental in non-Cartesian MR image reconstruction as using only one of these three ingredients will lead to degraded performance. It is worth mentioning that DIP performs poorly compared to U-net. Additionally, the quantitative scores are slightly improved for spiral readout compared to radial whatever the reconstruction strategy, likely because this undersampling scheme has a better coverage of high frequencies in the k-space.

Table 5.2: Mean PSNR / SSIM on the validation volumes of the different approaches averaged over both contrasts (knee fastMRI) in the multicoil setting. The best results are in bold font.

| Model                  | Radial                | Spiral                | # Parameters |
|------------------------|-----------------------|-----------------------|--------------|
| Adjoint + DCp          | 25.91 / 0.6486        | 31.36 / 0.7197        | 0            |
| DIP                    | 29.21 / 0.5834        | 29.19 / 0.5832        | 0            |
| U-net on Adjoint + DCp | 38.78 / 0.9106        | 40.02 / 0.9215        | 481k         |
| <b>NC-PDNet</b>        | <b>40.00 / 0.9191</b> | <b>40.68 / 0.9255</b> | 163k         |

**2D multicoil fastMRI dataset.** Regarding the 2D multicoil numerical experiments, we selected only the two best performers of the single-coil studies for comparison purposes, along with the adjoint baseline for reference and DIP being a strong contender for the generalization studies. As reported in Table 5.2 the Adjoint with DCp has a very low performance for the radial trajectory, hence the U-net working on top of this raw strategy is not able to correct sufficiently well for aliasing artifacts in the image domain in order to be competitive with the NC-PDNet. This illustrates that NC-PDNet has the ability to overcome situations where the trajectory might be causing issues to naive methods. We also observed for the NC-PDNet that the PSNR improvement in spiral imaging compared to radial imaging is larger in the multicoil setting. Finally, it can be noted that the sensitivity maps refinement has a relatively low impact on the number of parameters in the NC-PDNet, with only 7k additional parameters in  $\theta_r$  (163k vs 156k parameters in bottom rows of Table 5.2 vs Table 5.1).

Table 5.3: Mean PSNR / SSIM on the validation volumes of the different approaches for the OASIS brain dataset (3D setting). The best results are in bold font.

| Model                  | Radial                | #Parameters |
|------------------------|-----------------------|-------------|
| Adjoint + DCp          | 27.51 / 0.5183        | 0           |
| U-net on Adjoint + DCp | 31.42 / 0.8432        | 1.6M        |
| <b>NC-PDNet</b>        | <b>33.76 / 0.9160</b> | 67k         |

**3D single-coil brain OASIS dataset.** We can see in Table 5.3 that the NC-PDNet outperforms the baseline models on the OASIS dataset by a significant margin, even though its size was reduced due to memory constraints. We specify that the SSIM was computed on slices along the first dimension of the volumes in the 3D case. This choice was arbitrary.

**Reverse trajectories in 2D multicoil knee imaging.** The quantitative results for the reverse setting shown in Table 5.4 reveal contrasting results. The

Table 5.4: Mean PSNR/SSIM on the validation volumes of the different approaches averaged over both contrasts (knee fastMRI) in the multicoil reverse setting. Best results are in bold font.

| Model   | Radial                | Spiral                | # Parameters |
|---|-----------------------|-----------------------|--------------|
| <b>Adjoint + DCp</b>  | 25.91 / 0.6486        | 31.36 / 0.7197        | 0            |
| <b>DIP</b>  | 29.21 / 0.5834        | 29.19 / 0.5832        | 0            |
| <b>U-net on Adjoint + DCp (trained on different trajectory)</b> | 27.94 / 0.87          | 26.35 / 0.8850        | 481k         |
| <b>NC-PDNet (trained on different trajectory)</b>               | <b>37.86 / 0.9079</b> | <b>36.28 / 0.9052</b> | 163k         |

drop in PSNR is significant for both networks (compare Table 5.4 line by line with the corresponding reference in Table 5.2): The weaker performance is achieved for the U-net trained on radial and evaluated on spiral (by more than 13 dB), while the drop for NC-PDNet is limited to approximately 4.5 dB in the same use case and even less in the radial validation case (37.86 dB vs 40 dB). In contrast, the situation in SSIM is very much controlled in the NC-PDNet's case (a bit less so for the U-net).

**2D multicoil brain fastMRI dataset.** The quantitative results for the brain dataset are not to be compared head-to-head to the results obtained for the knee data. Indeed, there is a significant mismatch in reconstruction difficulty, the brain data being easier to reconstruct than the knee data metrics-wise, based on the fastMRI public leaderboard.<sup>10</sup> However, in Table 5.5 we can see that the neural networks still largely outperform the baseline (Adjoint + DCp) for both trajectories. We also observed that the NC-PDNet outperforms the U-net on both trajectories, this time with a bigger gap, suggesting that it generalizes in a better way.

Table 5.5: Mean PSNR/SSIM on the validation volumes of the different approaches averaged over all brain fastMRI imaging contrasts in the multicoil setting.

| Model  | Radial                | Spiral                | # Parameters |
|--|-----------------------|-----------------------|--------------|
| <b>Adjoint + DCp</b>                                 | 27.31 / 0.6028        | 32.23 / 0.6603        | 0            |
| <b>U-net on Adjoint + DCp (trained on knee data)</b> | 37.88 / 0.9234        | 38.76 / 0.9302        | 481k         |
| <b>NC-PDNet (trained on knee data)</b>               | <b>39.48 / 0.9368</b> | <b>39.81 / 0.9390</b> | 163k         |

<sup>10</sup>[fastmri.org/leaderboards/](https://fastmri.org/leaderboards/)

**Higher AF.** The quantitative results in Table 5.6 for a higher AF in validation, namely  $AF = 8$  compared to  $AF = 4$  in training, allow us to draw the same conclusions as the previous generalization experiments: the design of *NC-PDNet* is instrumental in getting more robust results for different trajectories.

Table 5.6: Mean PSNR / SSIM on the validation volumes of the different approaches for both contrasts (knee fastMRI) in the multicoil setting for  $AF = 8$ .

| Model                         | Radial                | Spiral                |
|-------------------------------|-----------------------|-----------------------|
| <b>Adjoint + DCp</b>          | 25.62 / 0.6097        | 30.00 / 0.6686        |
| <b>DIP</b>                    | 29.08 / 0.5782        | 29.21 / 0.5841        |
| <b>U-net on Adjoint + DCp</b> | 34.11 / 0.8592        | 34.60 / 0.8665        |
| <b>NC-PDNet</b>               | <b>36.71 / 0.8887</b> | <b>37.47 / 0.8967</b> |

**Reconstruction times.** The reconstruction times given in Table 5.7 show that the use of the NUFFT in the unrolled setting induces a 4.5-fold to 9-fold increase in execution time for 2D and 3D image reconstruction.<sup>11</sup> However, the use of the DCp merely increases the reconstruction time by only 33 ms.

A number that stands out is of course the reconstruction time for DIP. This was already noted by Darestani et al. [DH20], who proposed an acceleration strategy that we implemented. However, the computation time remains unrealistic for clinical applications.

Table 5.7: Reconstruction times of a single slice in milliseconds for the different networks in the different acquisition scenarios based on 2D/3D radial undersampling.

| Model                         | Single-coil 2D | Multicoil 2D | 3D |
|-------------------------------|----------------|--------------|----|
| <b>PDNet no DCp</b>           | 446            | NA           | NA |
| <b>Adjoint + DCp</b>          | 101            | 135          | 7  |
| <b>DIP</b>                    | 110k           | 133k         | NA |
| <b>PDNet w gridding</b>       | 112            | NA           | NA |
| <b>U-net on Adjoint + DCp</b> | 110            | 145          | 9  |
| <b>NC-PDNet</b>               | 479            | 661          | 80 |

<sup>11</sup>The reconstruction times observed for 3D are much less than those in 2D because of the different image sizes as well as smaller sizes of the networks.

## Qualitative results

For all qualitative results, the images were selected at random and do not necessarily have the same contrast. For each setting, the missing acquisition trajectory's reconstruction figures (if any) can be found in the Appendix in Figs. A-1-A-3. We chose to highlight some key parts of the reconstruction to pay attention to for the reader to better understand the difference between the U-net reconstruction and ours.

**2D single-coil knee fastMRI dataset.** The visual inspection of the reconstructed MR images confirm the quantitative measurements. In particular, one can visualize in Figure 5.2-12 that the image's inner contrast is better recovered by the *NC-PDNet* and U-net, and the structures are sharper for both architectures compared to the other competitors. The zooms allow us to identify the blurriness in DIP, the benefit of using DCp in the *NC-PDNet* architecture compared to the gridded and vanilla versions of PDNet.

**2D multicoil knee fastMRI dataset.** The qualitative results for the multicoil setting, available in Figure 5.2-13 confirm the quantitative results. We see on the right part of the knee (cf. red frame and zooms) that the reconstruction by the U-net after the application of the adjoint with DCp is not completely faithful. The reconstruction by the *NC-PDNet* is much more representative of the ground truth anatomy. Regarding the DIP solution, it appears very blurry compared to U-Net and *NC-PDNet* and hence this confirms the results we obtained in the single-coil setting.

**3D single-coil brain imaging (OASIS dataset).** Similarly to previous numerical experiments, the qualitative results shown in Figure 5.2-14 confirm the quantitative scores for the 3D brain imaging setting. Although the *NC-PDNet*'s reconstruction is blurred compared to the ground truth, we notice a much better resolution than the baselines as can be seen in the magnified views.

**Reverse trajectories in 2D multicoil knee imaging.** The qualitative results for the reverse setting, shown in Figs. 5.2-15-5.2-16, allow us to look at the quantitative metrics from a different perspective. We actually observed that the reconstructions are excellent in terms of image quality and that the degradation compared to the regular setting is minimally visible (except the radial artifacts for the *NC-PDNet* trained on spiral acquisition and evaluated on radial spokes in the right part of the knee). This confirms that the SSIM is the best metric to monitor in order for measuring success in generalization to other undersampling patterns.

**2D multicoil brain fastMRI dataset.** The qualitative results shown in [Figure 5.2-17](#) for the 2D multicoil brain FLAIR image confirm that the *NC-PDNet* was able to better generalize than the U-net.

**Higher AF.** We see in [Figure 5.2-18](#) that the higher AF has a very severe effect on all the reconstructions in terms of blur. Although the image based on the *NC-PDNet* architecture also suffers from a loss in resolution, it remains the best reconstruction with the least amount of blur and the best recovery of finer details.

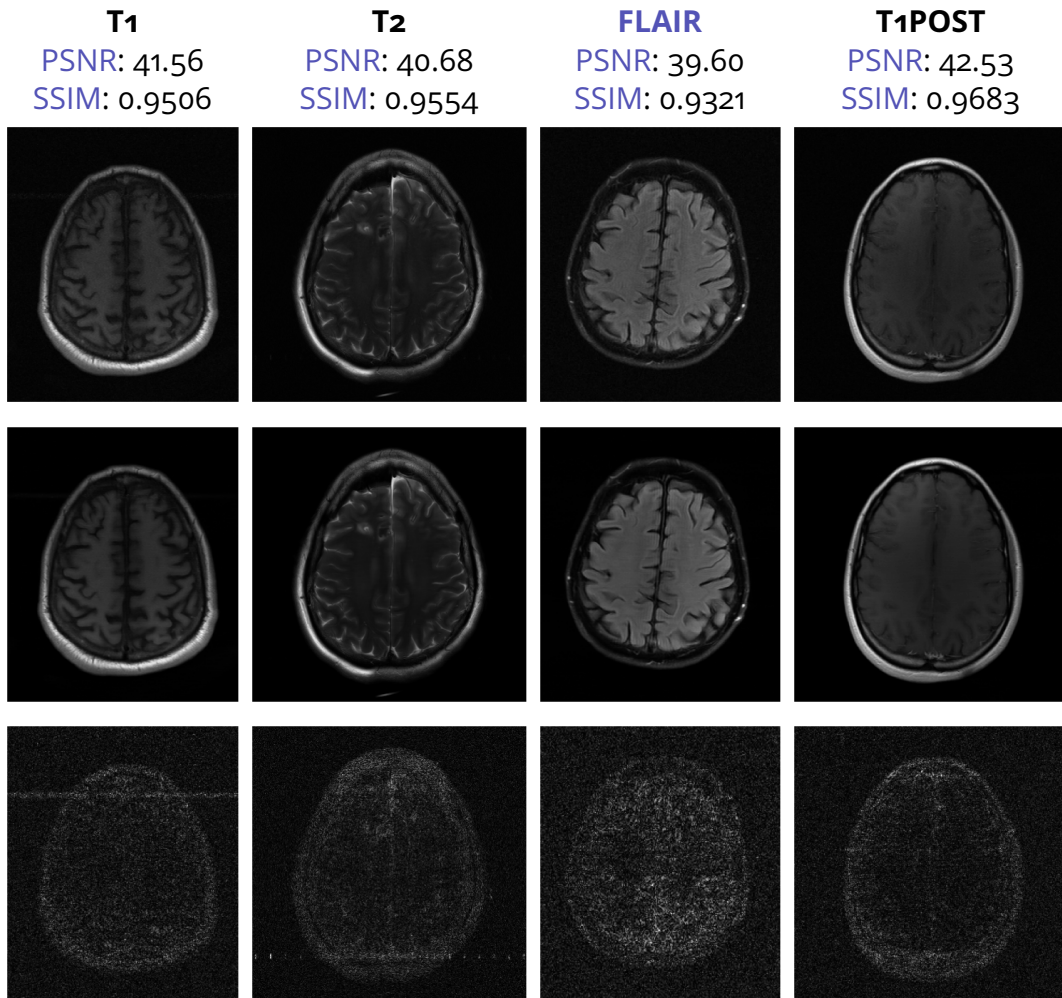


Figure 5.1-2: **Magnitude reconstruction results for the different fastMRI contrasts at AF 4.** The top row represents the ground truth, the middle on represents the reconstruction from a retrospectively undersampled k-space, and the bottom row represents the absolute error when comparing the two. The average image quantitative metrics are given for 30 validation volumes.



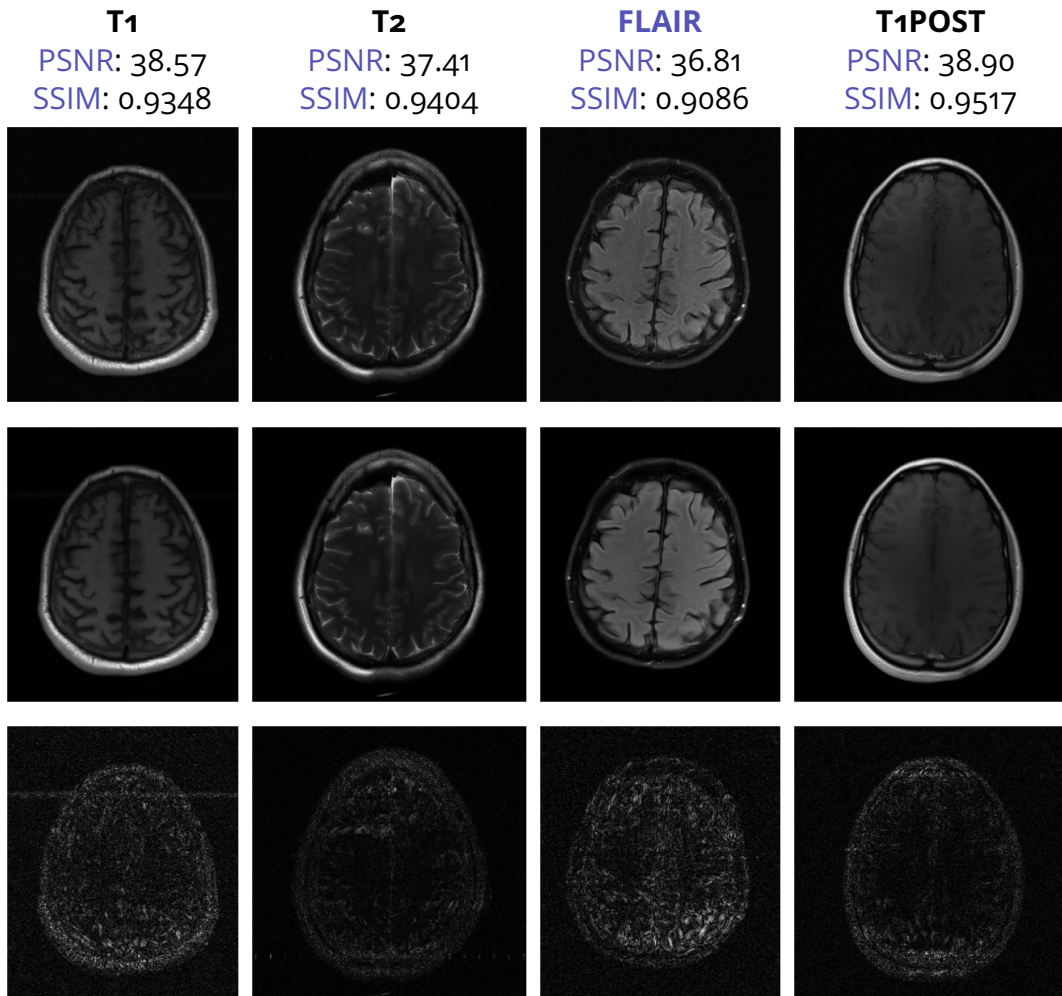


Figure 5.1-3: **Magnitude reconstruction results for the different fastMRI contrasts at AF 8.** The top row represents the ground truth, the middle one represents the reconstruction from a retrospectively undersampled k-space, and the bottom row represents the absolute error when comparing the two. The average image quantitative metrics are given for 30 validation volumes.





Figure 5.2-4: **Schematic representation of the two multi-shot non-Cartesian (radial and spiral) readouts considered here for 2D imaging.** In our setting, we used  $N_s = 100$  shots, each of them consisting of 640 samples giving a total of  $m = 64,000$  k-space measurements. Here only 10 shots are presented.

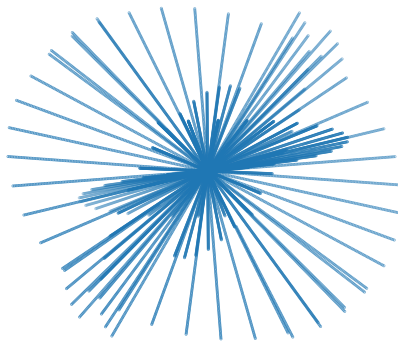


Figure 5.2-5: **Schematical illustration of the k-space trajectory considered in this work for 3D imaging.** Each of them uses 100 spokes and has a total of 64k measurements stacked 176 times across the additional dimension.

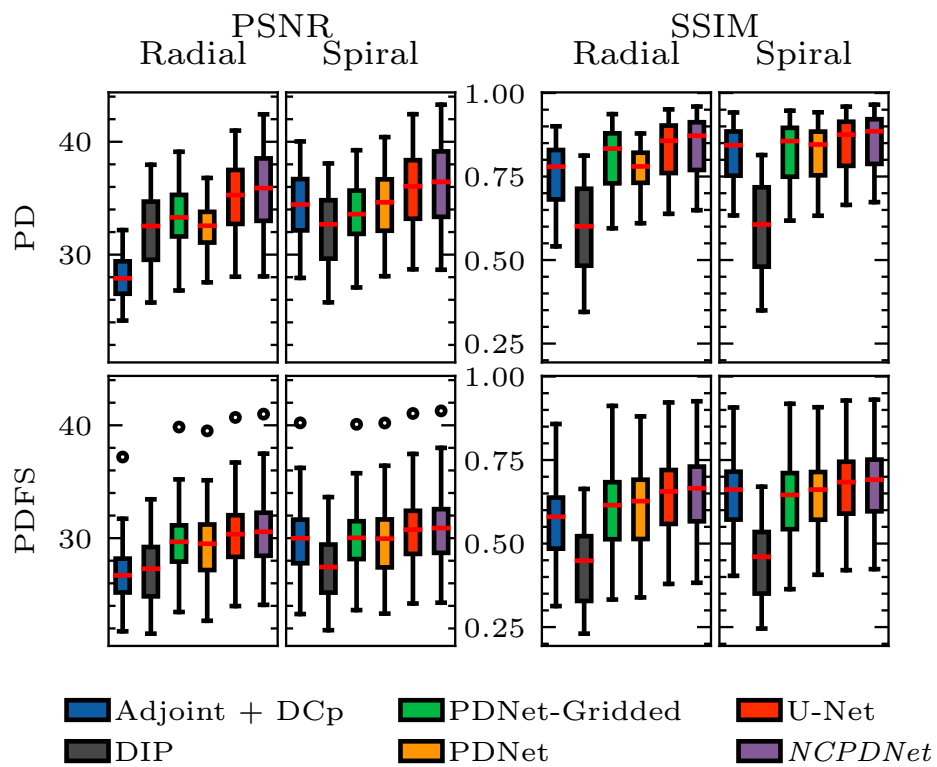


Figure 5.2-6: **Single-coil knee dataset:** Quantitative results of the different networks in the single-coil setting for both fastMRI contrasts.

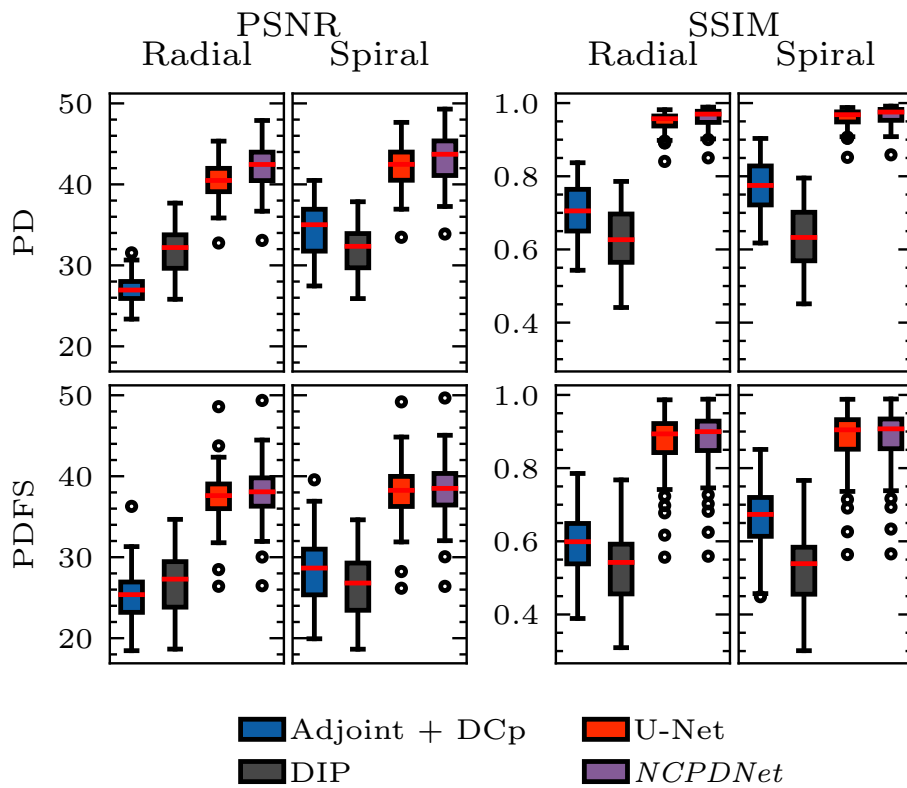


Figure 5.2-7: **Multicoil knee dataset**: Quantitative results of the different networks in the multicoil setting for both fastMRI contrasts.

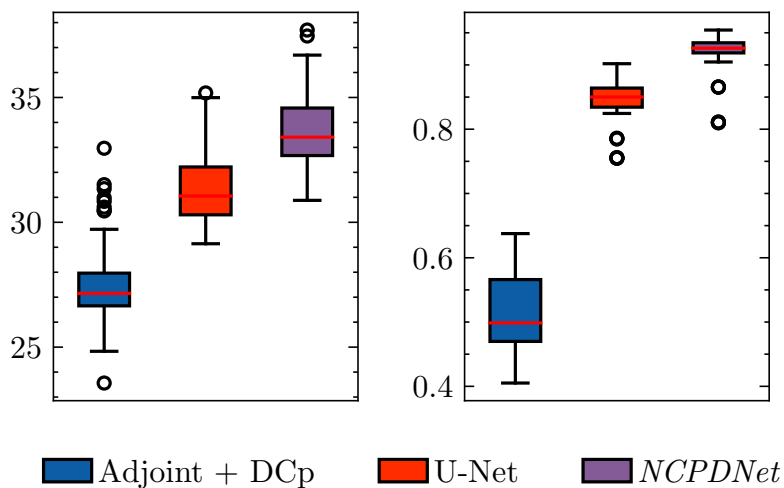


Figure 5.2-8: **3D OASIS brain dataset**: PSNR distribution of the different networks in the 3D radial undersampling scenario.

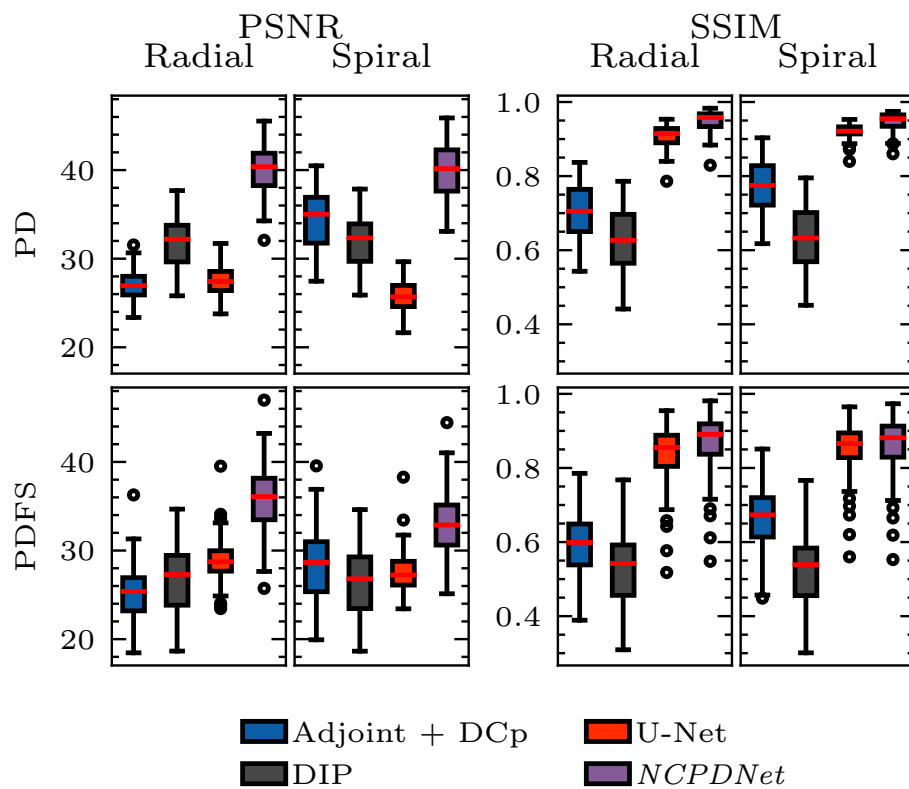


Figure 5.2-9: **Multicoil knee dataset (reverse setting)**: Quantitative results of the different networks in the multicoil reverse setting for both fastMRI contrasts.

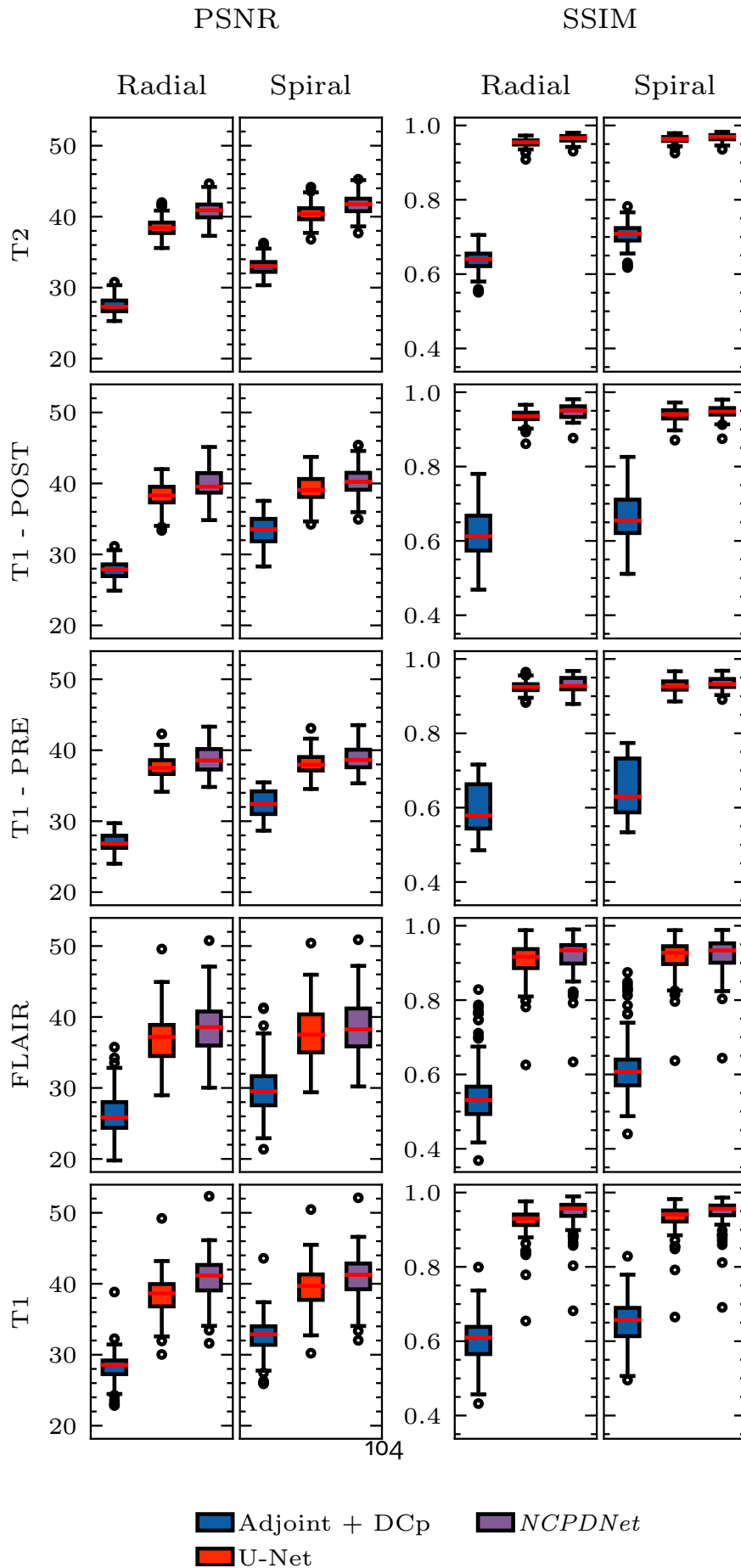


Figure 5.2-10: **2D multicoil fastMRI brain dataset**: Quantitative results of the different networks in the brain multicoil setting for the mul-

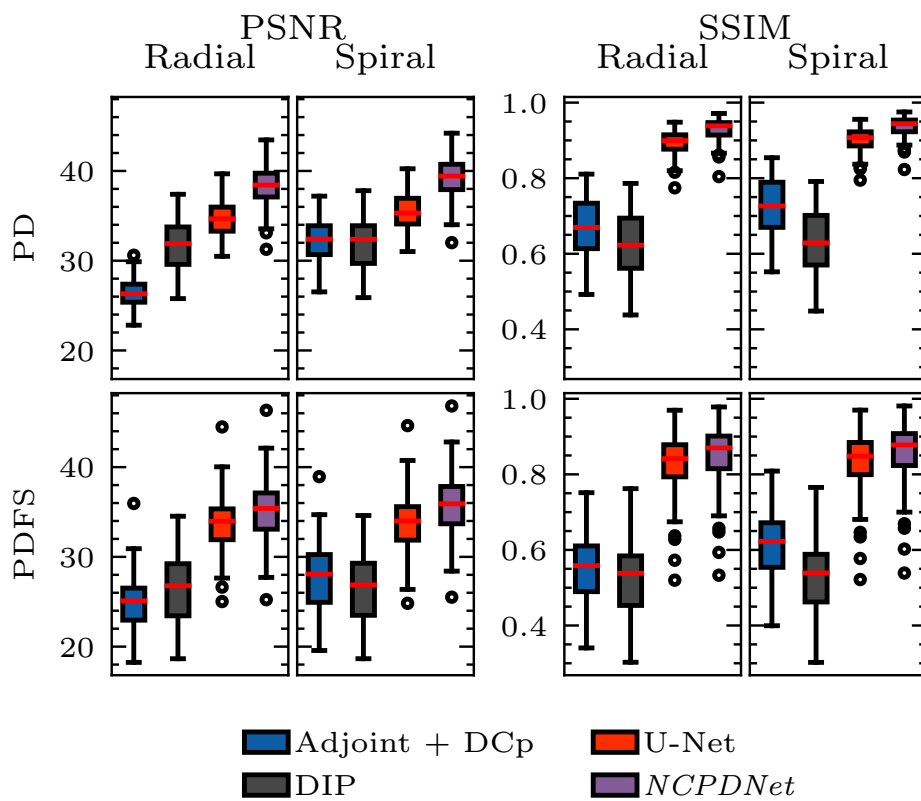


Figure 5.2-11: **2D multicoil** ( $AF = 8$ ): Quantitative results of the different networks in the multicoil setting for both fastMRI contrasts (knee dataset) with a higher  $AF$  during validation ( $AF = 8$ ) compared to training ( $AF = 4$ ).

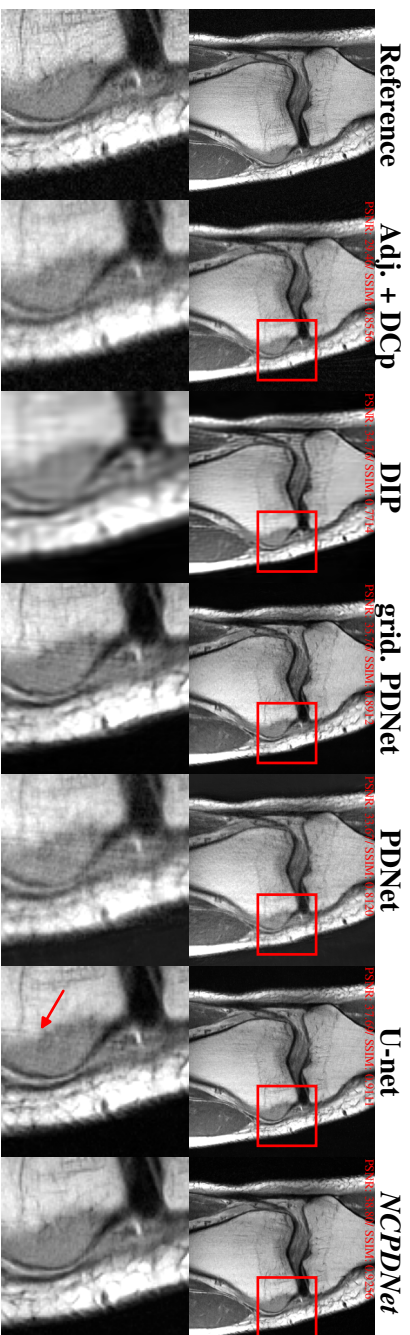


Figure 5.2-12: **2D single-coil radial acquisition (knee fastMRI dataset, PD contrast)**: Reconstruction results for a specific slice (16th slice of  $\text{flr1001184}$ , part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise **PSNR** and **SSIM** scores are shown on the top of each full FOV image.

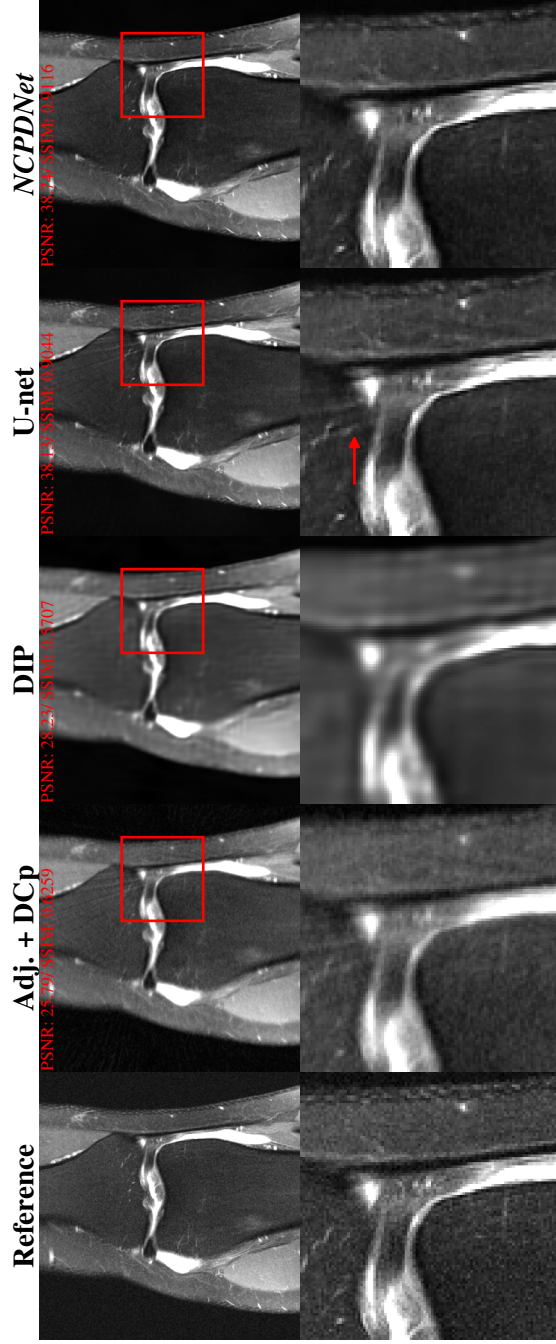


Figure 5.2-13: **2D multicoil radial acquisition (knee fastMRI dataset, PDFS contrast)**: Reconstruction results for a specific slice (16th slice of file100000, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise **PSNR** and **SSIM** scores are shown on the top of each full FOV image.



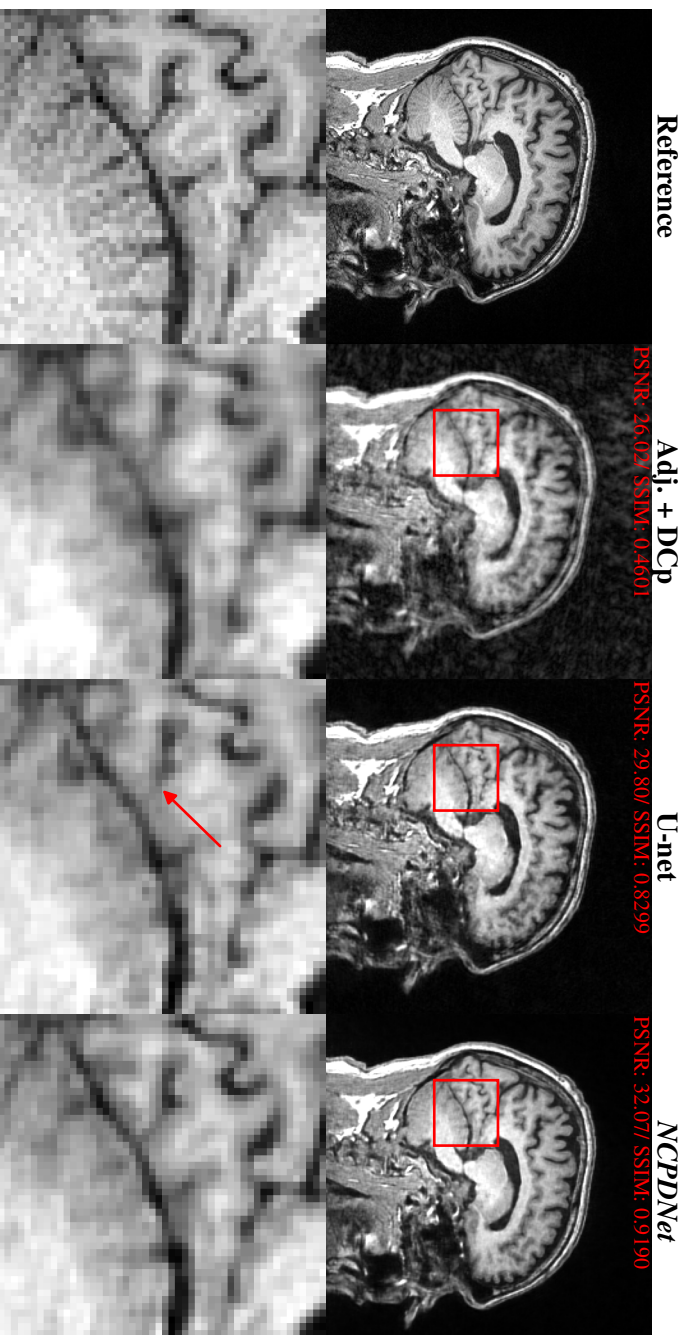


Figure 5.2-14: **3D radial acquisition (OASIS dataset, T1 contrast)**: Reconstruction results for a specific slice (101st slice of sub-OAS30001\_ses-d0129\_run-01\_T1w, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise **PSNR** and **SSIM** scores are shown on the top of each full FOV image.

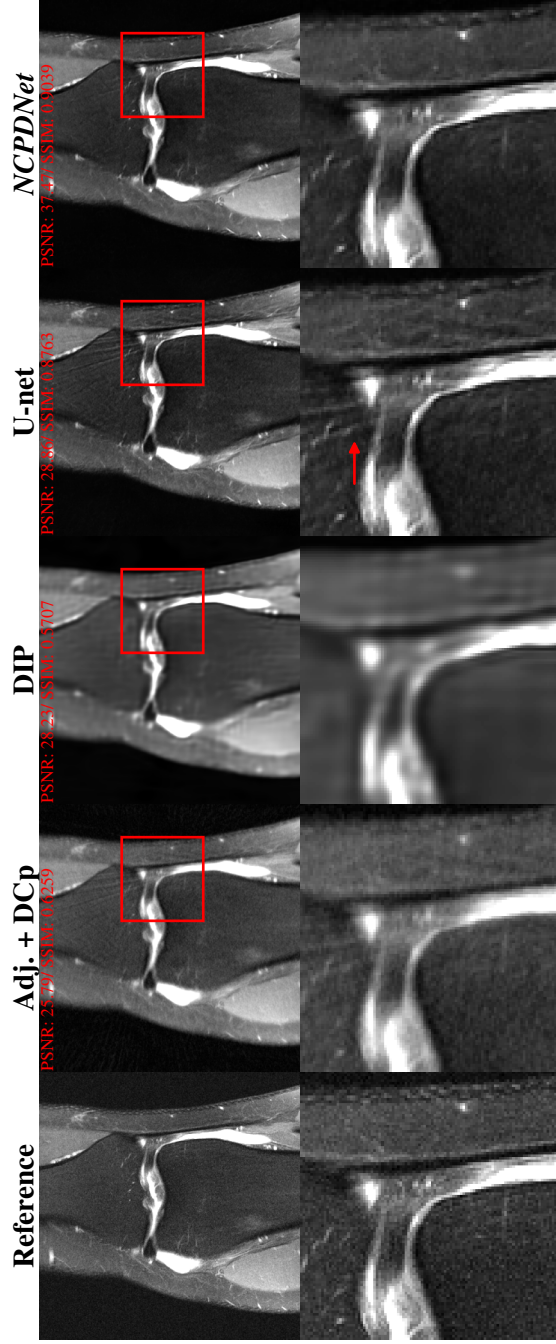


Figure 5.2-15: **2D multicoil radial acquisition (reverse setting, knee dataset)**: Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set) with networks trained with spiral trajectories. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red square in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image.

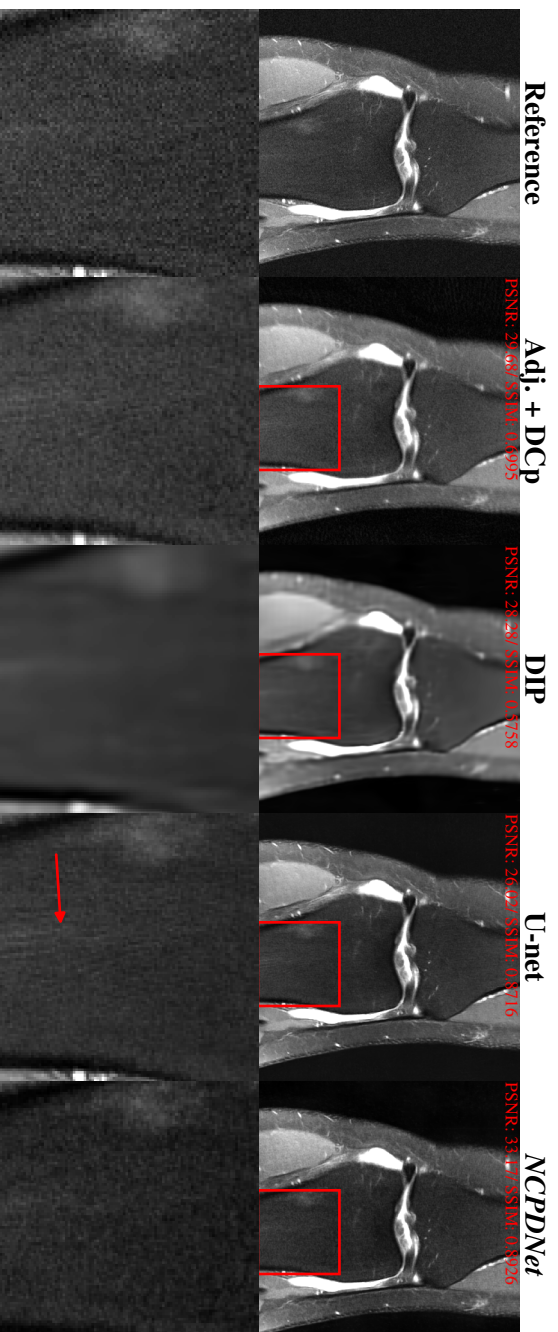


Figure 5.2-16: **2D multicoil spiral acquisition (reverse setting, knee dataset)**: Reconstruction results for a specific slice (16th slice of `file1000000`, part of the validation set) with networks trained on radial trajectories. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image.

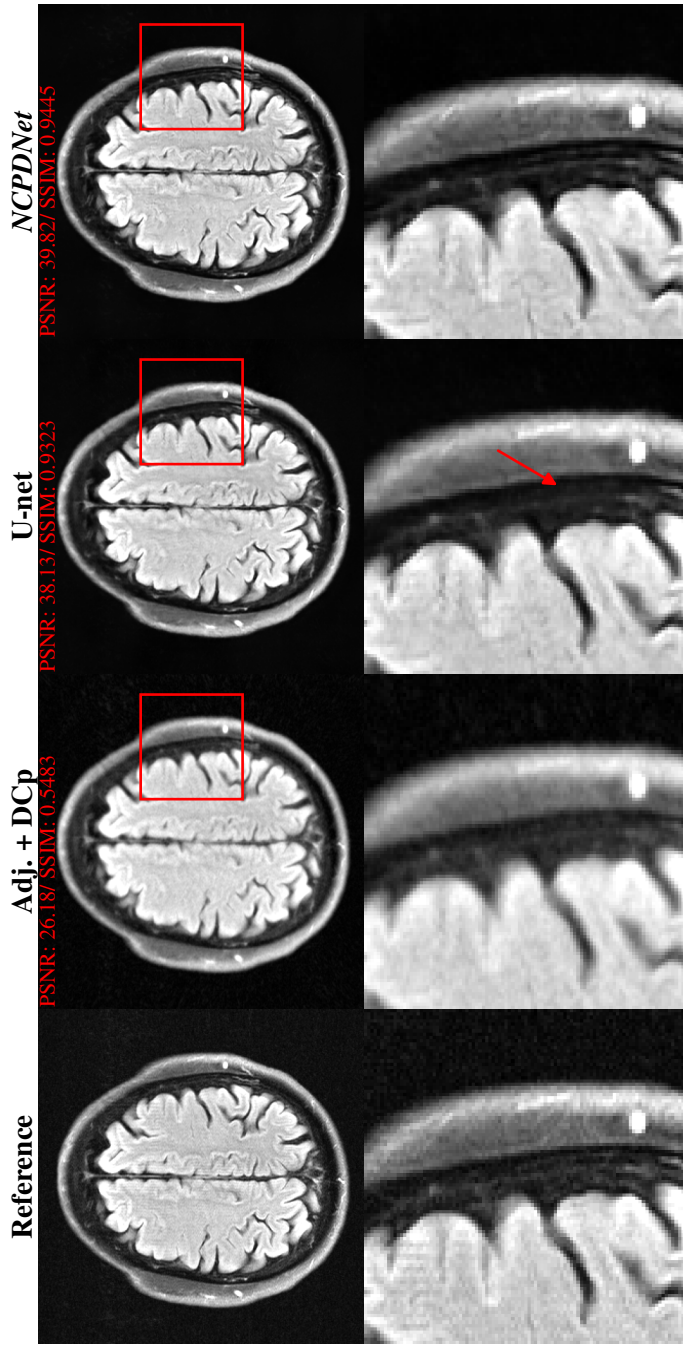


Figure 5-2-17: **2D multicoil radial acquisition (brain fastMRI dataset, FLAIR contrast)**: Reconstruction results for a specific slice (6th slice of file\_brain\_AXFLAIR\_200\_600247) from the brain fastMRI dataset with networks trained on knee data. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image.

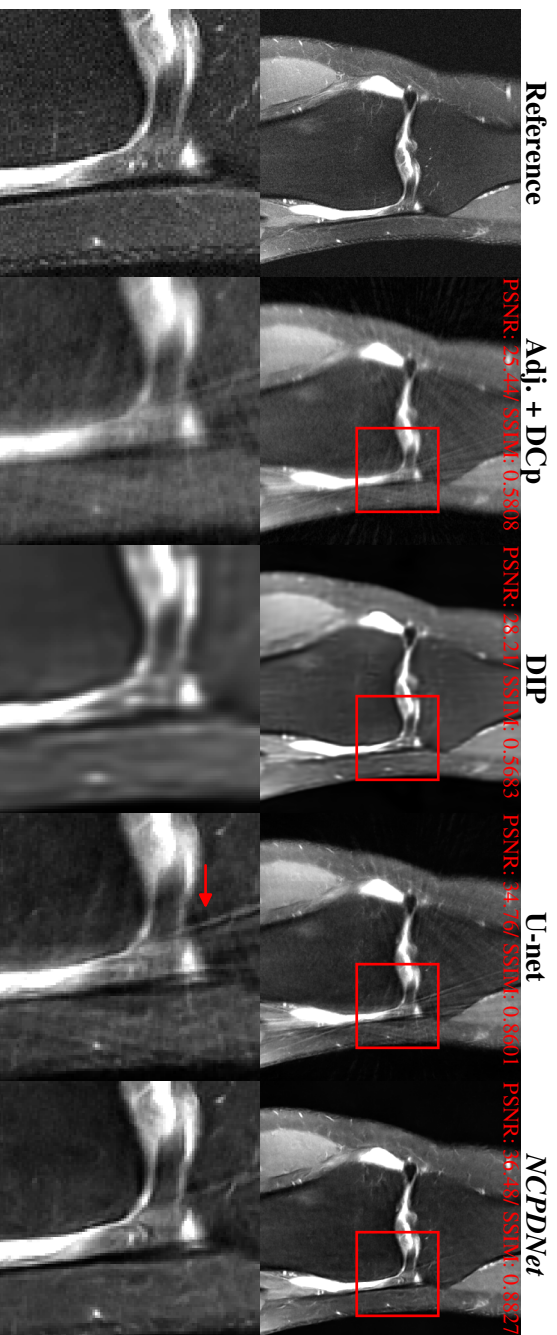


Figure 5.2-18: **2D multicoil radial acquisition (knee fastMRI dataset, PDFS contrast)  $AF = 8$** : Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set) for an  $AF = 8$ . The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row.



### 5.2.6 . Discussion and conclusion

In this section, we demonstrated how to apply the framework of unrolled neural networks to the problem of non-Cartesian MRI reconstruction. In particular, we showed via an ablation study, the importance of using the mechanism of density compensation [PM99] for this setting and how it is instrumental in obtaining the best possible results in a deep learning framework. We also managed to show how this mechanism can be extended to challenging acquisition scenarios like multicoil and 3D reconstruction with very minor adjustments. We conducted additional generalizability experiments to measure the robustness of the trained networks to out-of-distribution settings, and concluded that the visual degradation is minimal for all networks, and that additionally the *NC-PDNet* is quantitatively consistent. In particular, we found out that the generalization to other organs is possible even when using a single organ in training. Finally, the modular design of the code allows implementation and improvement of the results.

One disadvantage of the current implementation of unrolled networks in the non-Cartesian setting is their slowness compared to other baseline networks. This has been diagnosed in this work to be due to the NUFFT. Recent work has been carried out to implement an efficient NUFFT on the GPU, *cufiNUFFT* [SBB21], although not directly in a framework allowing auto-differentiation, making its use at training time difficult. However, fast implementations could very well be embedded in the network at reconstruction allowing us to alleviate the slowness of NUFFT in auto-differentiation frameworks. Complementary to that, a recent work [Lu+21] also tackles the problem of the slowness of the NUFFT by introducing a TPU implementation.

Some issues regarding the transferability/generalizability of this work still remain open. Indeed, these networks have been trained on a specific dataset with given experimental conditions, a given MR system vendor (Siemens-Healthineers, Erlangen, Germany), and a specific acquisition setup and organ. The question of generalizability from one vendor to another has been partially answered by Muckley et al. [Muc+21], where it has been observed that the results degrade when the networks trained with one vendor are applied to another. Although we partially addressed the concerns regarding the acquisition setup and the organs with our out-of-distribution experiments, a more systematic study is still required.

A more challenging question that can be raised is whether a lack of generalizability is intrinsic to the training process or not: The need for training data generally translates to using retrospective undersampling to be able to learn how to correct for aliasing artifacts from a “correct” ground truth. How do the networks generalize to prospective undersampling, especially in non-Cartesian acquisition scenarios where off-resonance effects come into play? This question is very difficult to answer systematically. In the case of prospective acquisition, as no ground truth is available, the methods cannot be compared one another.

One could try to simulate the non-idealities of the clinical setting (such as B0

field inhomogeneities, gradient inaccuracies/delays, eddy currents, etc...) as was done by Schlemper et al. [SSS20], but the gain and correct implementation of such simulations remains an open question. A final issue on transferability arises in this section for the 3D data, which consisted of magnitude volumes (and not of raw k-space data as in the 2D case). How can the networks generalize to complex-valued data? There exists ways to generate an artificial phase information, and they could be of help in order to be closer to the real use-case [SSS20]. Relevant data augmentation mechanisms [FHS21] can be one of these solutions.



## 6 - Clinical applicability of deep learning for MRI reconstruction

### Chapter Outline

|       |   |     |
|-------|---|-----|
| 6.1   | Learnlets . . . . .   | 116 |
| 6.1.1 | Introduction . . . . .  | 116 |
| 6.1.2 | Related Work . . . . .  | 117 |
| 6.1.3 | Learnlets, the model . . . . .  | 117 |
| 6.1.4 | Exact reconstruction . . . . .  | 122 |
| 6.1.5 | Data and Experiments . . . . .  | 122 |
| 6.1.6 | Results . . . . .   | 124 |
| 6.1.7 | Conclusions . . . . .   | 129 |
| 6.2   | Denoising Score-Matching for Uncertainty Quantification in Inverse Problems . . . . . | 130 |
| 6.2.1 | Introduction . . . . .  | 130 |
| 6.2.2 | Related Works . . . . .   | 131 |
| 6.2.3 | Deep DSM for Posterior Inference . . . . .  | 132 |
| 6.2.4 | Application to Bayesian Inverse Problems . . . . .                                    | 133 |
| 6.2.5 | Conclusions and Discussions . . . . .   | 134 |
| 6.3   | Is good old GRAPPA dead? . . . . .  | 135 |
| 6.3.1 | Introduction . . . . .  | 135 |
| 6.3.2 | Methods . . . . .   | 135 |
| 6.3.3 | Results . . . . .   | 136 |
| 6.3.4 | Conclusion and Discussion . . . . .   | 143 |

The first section of this chapter was submitted to a peer-reviewed journal and is currently under review:

**Zaccharie Ramzi**, K. Michalewicz, J. L. Starck, T. Moreau and P. Ciuciu. “Wavelets in the deep learning era”. 2021. Under review in *Journal of Mathematical Imaging and Vision*

It was previously presented as an oral in a peer-reviewed conference:

**Zaccharie Ramzi**, J. L. Starck, T. Moreau and P. Ciuciu. “Wavelets in the deep learning era”. In: *European Signal Processing Conference*. Vol. 2021-Janua. 2021, pp. 1417–1421. Oral



The remaining sections of this chapter were presented as abstracts in peer-reviewed workshops/conferences:

**Zaccharie Ramzi**, B. Remy, F. Lanassee, J.-L. Starck and P. Ciuciu. “Denoising Score-Matching for Uncertainty Quantification in Inverse Problems”. In: *NeurIPS 2020 Deep Learning and Inverse Problems workshop*. 2020

**Zaccharie Ramzi**, A. Vignaud, J.-L. Starck and P. Ciuciu. “Is good old GRAPPA dead?” In: *ISMRM*. 2021

**W**HILE the promise of faster MRI thanks to deep learning is highly appealing, these methods have to be properly vetted before they can be deployed in clinical settings. In this chapter, we will review 3 contributions that we made in order to provide guidelines for designing clinically applicable deep learning methods. These contributions deal with robustness to out-of-distribution settings, error quantification and prospective clinical applications.

## 6.1 . Learnlets

### 6.1.1 . Introduction

In this section, we cover the topic of robustness of neural networks, and propose a novel architecture, the *Learnlets*, inspired by wavelets denoising.

This architecture was designed to understand the gap between modern deep learning architectures used for denoising, like the U-net [RFB15] and the wavelets. Indeed, a large part of the success of deep learning is not well understood.

On the other hand, wavelets-based approaches are not state-of-the-art anymore for denoising but are theoretically grounded [Don95]. For applications where guarantees are needed – such as medical applications – this makes them ideal candidates.

Similarly to wavelets, U-nets present a multiscale approach, which allows us to analyze the signal at different resolutions. Their main difference lies in the number of nonlinearity applications. Indeed, while wavelets apply only one nonlinearity when performing denoising – a method called wavelet shrinkage –, the U-net architecture relies on several **Rectified Linear Unit (ReLU)** and max-poolings. These chained nonlinearities make the analysis of the denoising in U-nets very complicated. In particular, it is difficult to see how a network trained on one type of noise can be applied to other types of noises. Some works [Got+20] even show that classical neural networks can fail to recover elements that classical methods do, suggesting a trade-off between quality and stability.

In this section, we investigate whether using massive learning and large available datasets in a sparsity framework could allow us to achieve U-nets performance or if

the chained nonlinearities are equally important. We propose a new network, called *Learnlets*, which makes use of one of the strongest advantages of neural networks, learning via gradient descent to enhance the expressive power of wavelets, while keeping some interesting wavelet properties such as exact reconstruction.

We choose to test this network on a denoising problem, a task where wavelets have historically well-performed but are now overtaken by deep learning approaches. In parallel, we also propose a new U-net denoising scheme that guarantees an exact reconstruction when the noise tends to zero.

The full implementation of our method is open source in Python.<sup>1</sup>

### 6.1.2 . Related Work

Different studies have attempted to work at the intersection of wavelets and neural networks. Recoskie et al. [RM18] cast the wavelet transform as an auto-encoder where the latent representation has to be sparse and learn the filters. In this architecture only a simple high-pass and low-pass filter pair is learned. Similarly, but pushing further the idea, Jawali et al. [JKS19] developed a learning strategy to design new wavelet filters with certain properties imposed such as what they call perfect reconstruction (which we termed exact reconstruction) or vanishing moments. Their work was inspired by that of Pfister et al. [PB19], where the authors chose to use data patches to learn their transform rather than noise as was done by Jawali et al. [JKS19].

Observing U-nets, two parts are very similar to synthesis and analysis concepts in wavelet decompositions, Ye et al. [YHC18] proposed to use the wavelet transform to perform a better pooling/unpooling strategy than simply max-pooling/bilinear upsampling. Fan et al. [Fan+20] inspired themselves from the cascading wavelet shrinkage systems to enhance denoising autoencoders. In brief, they proved that using a soft-thresholding nonlinearity provided more power to the denoising autoencoders than other non-linearities.

In these related papers, non-linearities (namely ReLU) are in majority applied to the low frequencies rather than the high frequencies, contrarily to what is common in the wavelet framework. In this section, we don't try to modify U-nets by importing wavelet ingredients, but rather try to push the limits of sparsity based approach by using learning while keeping sparsity concept unchanged. This allows us to recover the classical properties of wavelets i.e. decomposition with exact reconstruction, thresholding and reconstruction, while using a learning based approach.

### 6.1.3 . Learnlets, the model

Let  $\mathbf{x} \in \mathbb{R}^{n \times n}$  be an image. Let  $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$  be the version of this image corrupted by an additive white Gaussian noise  $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{n \times n})$  whose variance  $\sigma^2$  is assumed known. Let  $\Sigma$  be a compact set of possible values for  $\sigma$ , we chose

---

<sup>1</sup>[github.com/zaccharieramzi/understanding-unets](https://github.com/zaccharieramzi/understanding-unets)

to have  $\sigma \sim \mathcal{U}(\Sigma)$ . For a given number of scales  $m$  and a given set of parameters  $\theta = (\theta_S, \theta_T, \theta_A) \in \Theta_m$ , we defined the learnlets as function  $\mathbf{f}_\theta$  from  $(\mathbb{R}^{n \times n} \times \Sigma)$  to  $\mathbb{R}^{n \times n}$ :

$$\mathbf{f}_\theta(\tilde{\mathbf{x}}, \sigma) = \mathbf{S}_{\theta_S}(\mathbf{T}_{\theta_T}(\mathbf{A}_{\theta_A}(\tilde{\mathbf{x}}), \sigma)) \quad (6.1)$$

where we have:

1.  $\mathbf{A}_{\theta_A}$ , the analysis function defined in [subsection 6.1.3](#).
2.  $\mathbf{T}_{\theta_T}$ , the thresholding function defined in [subsection 6.1.3](#).
3.  $\mathbf{S}_{\theta_S}$ , the synthesis function defined in [subsection 6.1.3](#).

An illustration of the learnlets is given in [Figure 6.1-1](#).

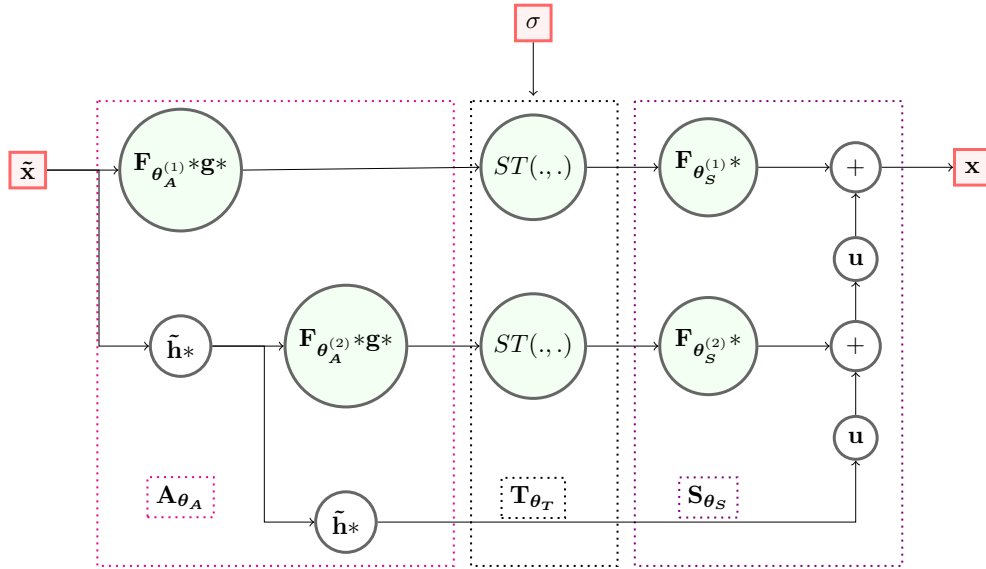


Figure 6.1-1: **Schematic representation of the learnlets model, with  $m = 2$  scales.** The red nodes are inputs/outputs. The lightly green nodes correspond to functions whose parameters can be learned. Note that the standard deviation of the noise before thresholding is not learned but rather estimated, and is omitted in this diagram for clarity.

## Analysis

Intuitively, one can see the analysis function as the equivalent of the wavelet transform with some learned filters. This linear function is defined as:

$$\mathbf{A}_{\theta_A}(\tilde{\mathbf{x}}) = \left( \left( \mathbf{F}_{\theta_A^{(i)}} * \mathbf{g} \left( \tilde{\mathbf{h}}^{i-1}(\tilde{\mathbf{x}}) \right) \right)_{i=1}^m, \tilde{\mathbf{h}}^m(\tilde{\mathbf{x}}) \right) \quad (6.2)$$

where we have:

- $\mathbf{F}_{\theta_A^{(i)}}$ , the filter bank at scale  $i$ . The convolutions are done without bias.  $\theta_A^{(i)}$  are the  $J_i$  convolution kernels all of the same square size  $(k_A, k_A)$  (for now  $J_i = J_m$ ).
- $\tilde{\mathbf{h}} = \bar{\mathbf{u}} \circ \mathbf{h}$ , the low-pass filtering ( $\mathbf{h}$ ) followed by a decimation ( $\bar{\mathbf{u}}$ ). The decimation is performed by taking one line out of 2 and one row out of 2, in line with the way it is done in wavelet transforms.
- $\mathbf{g}$  the high-pass filtering defined as:  $\mathbf{g}(\mathbf{y}) = \mathbf{y} - \mathbf{u}(\tilde{\mathbf{h}}(\mathbf{y}))$ , with  $\mathbf{u}$  the upsampling operation performed with a bicubic interpolator.

For ease of manipulation we rewrite  $\mathbf{A}_{\theta_A}(\tilde{\mathbf{x}}) = ((\mathbf{d}_i)_{i=1}^m, \mathbf{c})$ , with  $\mathbf{d}_i \in \mathbb{R}^{\frac{n}{2^{i-1}} \times \frac{n}{2^{i-1}} \times J_i}$  the detail coefficients and  $\mathbf{c}$  the coarse coefficients.

Note that low and high pass filters ( $\mathbf{h}, \mathbf{g}$ ) are fixed, and only  $\mathbf{F}_{\theta_A^{(i)}}$  filters are learned. As  $\mathbf{g}$  has a zero mean, all coefficients  $\mathbf{d}_i$  have by construction a zero mean. This wavelet property is fundamental to model the noise on the coefficients. Indeed, in the absence of signal, the coefficients follow a Gaussian distribution with a zero mean, and a standard  $k\sigma$  thresholding can be applied,  $\sigma$  being the noise standard deviation. With wavelets,  $k$  would be chosen between 3 and 5, and would be a user parameter. In this setting, this  $k$  value can be learned, and can be different at each scale.

## Thresholding

The nonlinearity function used for wavelet shrinkage is typically either a hard-thresholding or a soft-thresholding [Don95]. The soft-thresholding offers more stability, therefore we made this choice for our architecture. The thresholding function, in the case of a white Gaussian noise of variance  $\sigma^2$ , is defined as:

$$\mathbf{T}_{\theta_T}(((\mathbf{d}_i)_{i=1}^m, \mathbf{c}), \sigma) = \left( \left( (t_{ij}(d_{ij}, \sigma))_{i=1}^{J_i} \right)_{i=1}^m, \mathbf{c} \right) \quad (6.3)$$

where  $t_{ij}(\mathbf{d}, \sigma) = \hat{\sigma}_{ij} ST\left(\frac{1}{\hat{\sigma}_{ij}} d_{ij}, \theta_T^{(ij)} \sigma\right)$ , with:

- $d_{ij} \in \mathbb{R}^{\frac{n}{2^{i-1}} \times \frac{n}{2^{i-1}}}$  the output of the  $j$ -th filter of  $i$ -th scale.
- $\hat{\sigma}_{ij}$  the estimated standard deviation of  $d_{ij}$  when the input of the transform is set to be a white Gaussian noise of variance 1. This ensures the noise coming just before the thresholding is of variance approximately  $\sigma$ . The threshold is therefore truly  $\theta_T^{(ij)} \sigma$ .
- $\theta_T^{(ij)}$  is the thresholding level applied at scale  $i$  on the  $j$ -th analysis filter.

- $ST(\mathbf{d}, \mathbf{s})$  is the soft-thresholding function applied point-wise on  $\mathbf{d}$  with threshold  $\mathbf{s}$ :  $ST(\mathbf{d}, \mathbf{s}) = \text{sign}(\mathbf{d}) \max(|\mathbf{d}| - \mathbf{s}, 0)$ .

It is important to notice that, thanks to the linearity of the analysis operator, the thresholding strategy can be very easily adapted to non-stationary Gaussian noise or to any other kind of noise, such as Poisson noise or a mixture of Gaussian and Poisson noise.

## Synthesis

Intuitively, one can see the synthesis function as the equivalent of the wavelet reconstruction operator, with learned filters. It is important to note that the synthesis function is linear. The synthesis function is defined recurrently as:

$$\mathbf{S}_{\theta_S}((\mathbf{d}_i)_{i=1}^m, \mathbf{c}) = \mathbf{S}_{\theta_S}^{(m-1)}\left((\mathbf{d}_i)_{i=1}^{m-1}, \mathbf{u}(\mathbf{c}) + \mathbf{F}_{\theta_S^{(m)}} * \mathbf{d}_m\right) \quad (6.4)$$

where  $\mathbf{S}_{\theta}(\emptyset, \mathbf{c}) = \mathbf{c}$  and:

- $\mathbf{F}_{\theta_S^{(i)}}$ , the filter bank at scale  $i$ , used for regrouping. The convolutions are done without bias and added all together.  $\theta_S^{(i)}$  are the  $J_i$  convolution kernels all of the same square size  $(k_S, k_S)$ .
- $\mathbf{u}$ , the upsampling operation performed with a bicubic interpolator.

## Constraints

Some constraints are used on the parameters of the learnlets to make them as close as possible to the wavelets and therefore make them understandable:

- The analysis filters are forced to have a unit norm.
- The thresholding levels are in  $[0, 5]$ .

## Learning

The optimization problem is given as:

$$\underset{\theta \in \Theta}{\text{argmin}} \mathbb{E}_{\mathbf{x}, \sigma} [L_f(\theta)] \quad (6.5)$$

where  $L_f(\theta) = \|\mathbf{x} - \mathbf{f}_{\theta}(\tilde{\mathbf{x}}, \sigma)\|_2^2$  and the expected value is computed empirically, via the empirical mean over a batch.

## Learnlets as the bridge between Sparsity and U-nets

The learnlet transform is very similar in its spirit to the curvelet transform. Indeed, in both transforms the image is first decomposed into a set of wavelet scales, and filters are applied on each scale. In a curvelet decomposition, it would be directional and fixed filters, while filters are learned in our proposed scheme. Obtained coefficients can be manipulated exactly the same way as wavelets or curvelets coefficients.

It is interesting to notice that after training, the pixels of the learnlets' (with exact reconstruction) analysis filters constitute meaningful designs. This can be observed in [Figure 6.1-2](#): lines with different slopes are displayed in scale-zero filters (details about the data and the training are given in [subsection 6.1.5](#)).

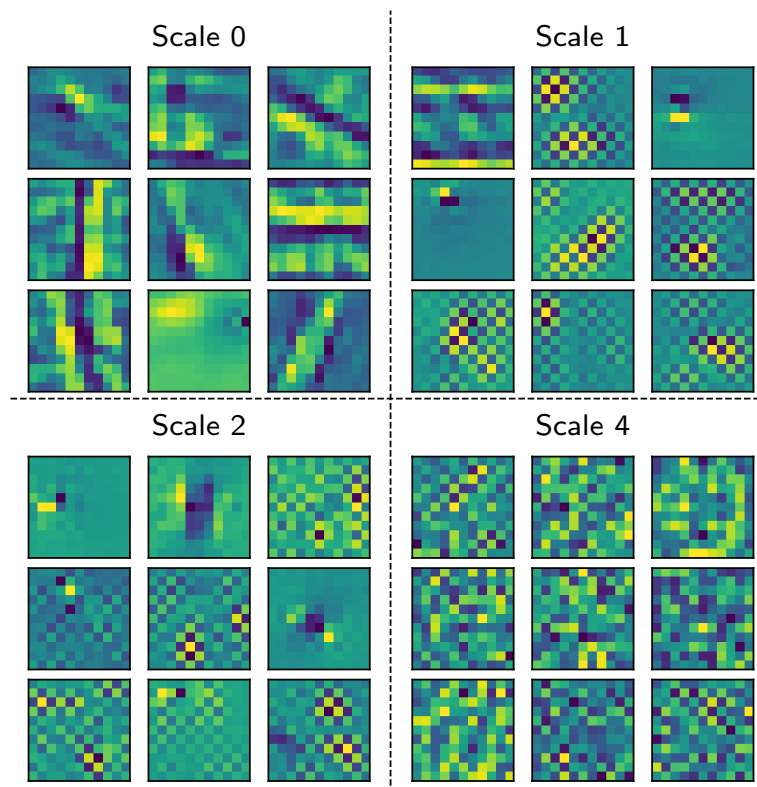


Figure 6.1-2: **Filters.** Visualization of learnlets analysis filters for four different scales.

On the other hand, learnlets share very similar properties with U-nets. For example, they make use of gradient-based learning, but they also feature a multiscale analysis along with the use of non-linearities.

### 6.1.4 . Exact reconstruction

## Learnlets

Exact reconstruction guarantees that if no noise is present, the signal will be perfectly reconstructed, without any error. This can be achieved using the analysis filter previously fixed as identity. In particular, let's consider a single scale  $i$ , after the application of the  $g$  filter. The operation carried out by the network, without thresholding can be written as:

$$\mathbf{x}_{\text{out}}^{(i)} = \sum_{j=1}^N \mathbf{F}_{\theta_S^{(i,j)}} * \mathbf{F}_{\theta_A^{(i,j)}} * \mathbf{x}_{\text{in}} \quad (6.6)$$

where  $N$  is the number of filters at that scale. Since we have  $\mathbf{F}_{\theta_A^{(i,1)}} = \mathbf{Id}$ , we can also fix the corresponding synthesis filter  $\mathbf{F}_{\theta_S^{(i,1)}} = \mathbf{Id} - \sum_{j=2}^N \mathbf{F}_{\theta_S^{(i,j)}} * \mathbf{F}_{\theta_A^{(i,j)}}$ . This trivially gives without thresholding,  $\mathbf{x}_{\text{out}} = \mathbf{x}_{\text{in}}$ . We implemented this constraint in the network, allowing to learn a different thresholding level for this filter.

## The general case

In order to better understand the properties of exact reconstruction in the learnlets, we can study whether it is possible to enforce it as well for black-box residual neural networks. A simple solution, given a known noise level  $\sigma$  is to use the following general expression:

$$\mathbf{g}_{\theta}(\tilde{\mathbf{x}}, \sigma) = \tilde{\mathbf{x}} - \sigma \mathbf{f}_{\theta}(\tilde{\mathbf{x}}) \quad (6.7)$$

where  $\mathbf{f}_{\theta}$  is the output of the network without exact reconstruction. It can be noted that when  $\sigma$  tends to zero, then  $\mathbf{g}_{\theta}(\tilde{\mathbf{x}}, \sigma) \rightarrow \tilde{\mathbf{x}}$  and we can assure that the output will retrieve the input signal. It should be noted that this formulation might be unstable as it can amplify errors at high noise levels. This aspect will be analyzed in the next section.

### 6.1.5 . Data and Experiments

The implementation was done in Python 3.6, using the TensorFlow 2.1 framework [Aba+16] for model design. The training was done on the Jean Zay public supercomputer, using for each job a single GPU Nvidia Tesla V100 SXM2 with 32 GB of RAM.

## Data

The data used was the BSD500 dataset [Arb+11]. This data consists of natural images of sizes  $481 \times 321$  and  $321 \times 481$ . The train and tests subsets of BSD500 were used as the training dataset. The validation subset of BSD500, containing the BSD68 [Mar+01] images was left out. We used BSD68 as the test dataset. This choice is motivated by the fact that many other denoising studies [Zha+17a; Lef18] use this dataset for comparison.

## Pre-processing

For training, patches of size  $256 \times 256$  were randomly extracted on-the-fly. The images were then linearly mapped from  $[0, 255]$  to the  $[-0.5, 0.5]$  interval and converted from RGB to grayscale using the function provided by TensorFlow.<sup>2</sup> In addition, data augmentation techniques such as random flipping and random  $\theta$ -degree ( $\theta = 90^\circ, 180^\circ, 270^\circ$ ) rotations were applied. Noise was then added by first drawing uniformly at random in the specified interval  $\Sigma$  a noise level  $\sigma$ , then generating a  $256 \times 256$  white Gaussian noise patch  $\epsilon$  with this standard deviation. It is to note that during training, a single batch can feature different noise standard deviations.

At test time, the images were mirror-padded to a  $352 \times 512$  size (or  $512 \times 352$ ), in order to avoid shape mismatches when downsampling and upsampling, and the image quality metric was computed only on the original image shape. The test images were also corrupted by an additive white Gaussian noise for various standard deviations  $\sigma$ :  $\{0.0001, 5, 15, 20, 25, 30, 50, 55, 60, 75, 85, 95, 100\}$ . This allowed us to test the performance of our method in different noise level settings.

## Model and training

**Models design.** We compare the learnlets with the U-net for the task of denoising. For the U-net, we used the architecture described in [YHC18, Fig.10.(a)] which contains 124 million parameters for the case of a network with 128 base filters.

Unless specified otherwise, the learnlets parameters were chosen as:

- $m = 5$  scales.
- 256 learnable analysis filters + 1 fixed analysis filters being just the identity,  $\mathbf{F}_{\theta_A^{(i)}}$ , of size  $11 \times 11$ .
- 257 learnable synthesis filters,  $\mathbf{F}_{\theta_S^{(i)}}$ , of size  $13 \times 13$ .
- the thresholding levels only depend on the scale,  $\theta_T^{(ij)} = \theta_T^{(i)}$ .

This amounts to 372k trainable parameters, only three hundredths of the size of the U-net.

**Training parameters.** The networks were both trained on the mean squared error in line with (6.5). Each epoch consisted of 200 batches of 8 extracted patches, and their respective noise level in the case of the learnlets. The training noise standard deviation range was chosen as  $\Sigma = [0; 55]$ . The networks were trained with an Adam optimizer [KB15]. The learning rate was set at  $10^{-3}$ , then decreased

---

<sup>2</sup>TensorFlow Documentation for RGB to grayscale



by half every 25 epochs, until it reached a minimum of  $10^{-5}$ . The trainings took about 8 hours for 500 epochs each.

## Evaluation

**Evaluation metric.** For the evaluation of the performance of the different models we used the PSNR metric defined in subsection 2.4.1. For each test noise standard deviation  $\sigma$ , we compute the mean of the PSNR of the denoised images, for all BSD68 images.

**Testing.** In addition, the networks were compared to wavelets denoising [Mal99], which was implemented by using the code of PySAP [Far+20]. The wavelets family was the Biorthogonal 7.9, 5 scales were used, a hard-thresholding was used with a thresholding level of 3 (except for the first scale where it was 4).

### 6.1.6 . Results

#### Quantitative results

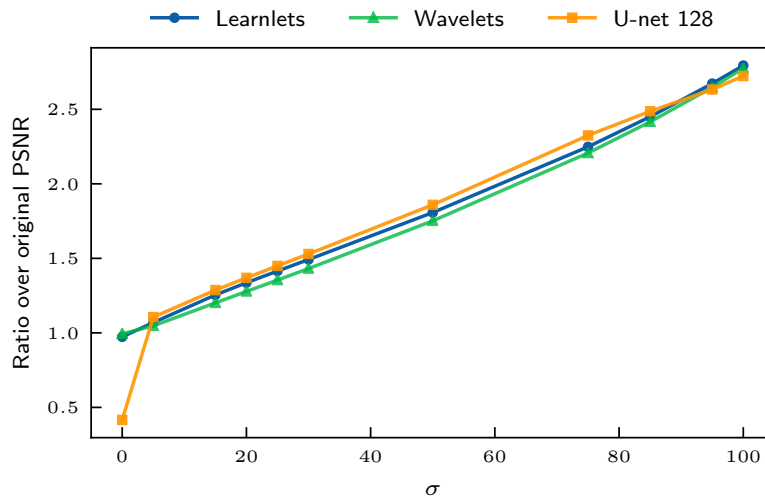


Figure 6.1-3: **Denoising performance.** Ratio of the denoised image PSNR compared to the original noisy image PSNR for different standard deviations of the noise added to the test images for all considered models. The train noise standard deviation range was  $[0; 55]$ .

**Comparison with other methods.** We compared the U-net and learnlets with exact reconstruction against algorithms not involving learning, namely wavelets shrinkage. Figure 6.1-3 shows that for a large part of the band  $[5; 55]$ , where they have been trained, the wavelets have a performance that is degraded compared to the learnlets. Using learning, the learnlets enhance their decomposition

| Model name                    | Wavelets | U-net 128 | Learnlets | U-net 64 |
|-------------------------------|----------|-----------|-----------|----------|
| Denoising runtime in ms (std) | 274 (21) | 272 (18)  | 106 (12)  | 64 (1)   |

Table 6.1: Runtimes of the different models for the denoising of one image. Parameters used are the same as [Figure 6.1-3](#).

power compare to the original wavelet model with no learning. For small noise level, the U-net gets degraded performances compared with learnlets with exact reconstruction and wavelets. In this setting, the denoiser must act as the identity. Finally, we can see that for unseen test noise levels (i.e. 95), the performance of U-net drops slightly while the learnlets keep relatively good performances. This suggests that the learnlets generalize better than U-nets on unseen noise levels.

In addition, we can see in [Table 6.1](#) that the learnlets benefit from their GPU implementation and run faster than both the wavelets and U-net 128.

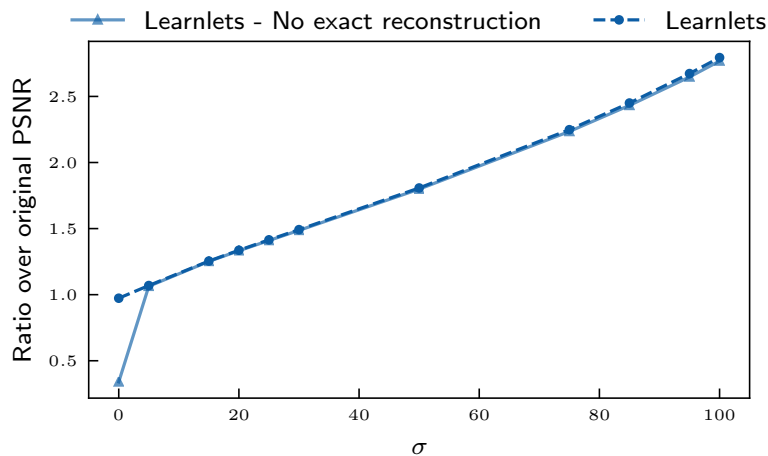


Figure 6.1-4: **Exact reconstruction.** Ratio of the denoised image PSNR compared to the original noisy image PSNR for different standard deviations of the noise added to the test images for learnlets with and without forcing exact reconstruction. The train noise standard deviation range was  $[0; 55]$ . The number of filters used was 64.

**Learnlets with exact reconstruction.** We saw in [Figure 6.1-3](#) that learnlets with exact reconstruction compete with classical methods for a wide range of noise standard deviations. [Figure 6.1-4](#) shows that the performance of the network with forced exact reconstruction is almost the same as the one without forced exact reconstruction (we only lose 0.1 dB at  $\sigma = 30$  for example) on the majority of the test noise standard deviations. However, for low noise standard deviations, the network with forced exact reconstruction completely overpowers the other one. This is due to the fact that, at low noise standard deviations, for the  $i$ -th scale, the

term  $x_{out}^{(i)}$  is practically the same as its thresholded version, because the thresholds  $\theta_T^{(ij)}\sigma$  are going to be low. Therefore, it is compensated in the corresponding synthesis filter used for exact reconstruction at that scale,  $\mathbf{F}_{\theta_A^{(i,1)}}$ . This allows to guarantee, in this case, no loss of information in the signal if it is clearly present.

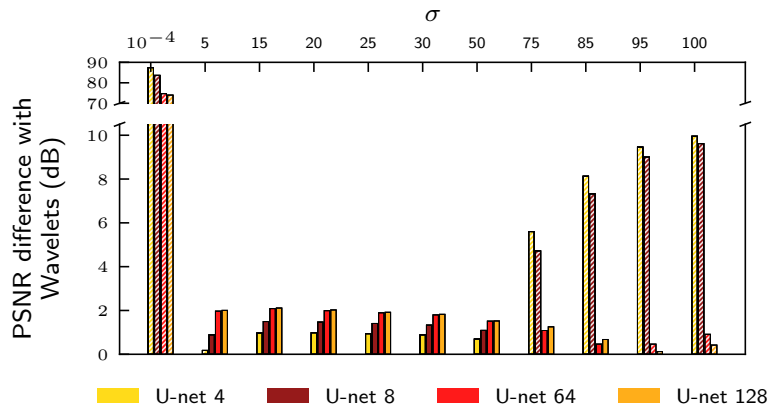


Figure 6.1-5: **Influence of model size.** PSNR difference (in dB) with respect to wavelets denoising for different standard deviations of the noise added to the test images for U-nets of various sizes. The striped bars correspond to negative differences. The train noise standard deviation range was  $[0; 55]$ .

**U-nets of different sizes.** Due to their reduced number of parameters, *overfitting* is *a priori* less likely to occur in small neural networks than in larger ones. Therefore, we studied whether the generalization to high noise levels would be better with small-sized U-nets. To do this, the number of base filters was modified with respect to the original case, obtaining Figure 6.1-5. We can see that deeper networks perform better for seen and unseen noise levels.

In terms of generalization, the PSNR difference between U-nets with a low quantity of filters (4 or 8) and those with a larger amount (64 or 128) is amplified for the range  $[55; 100]$ .

**U-net with exact reconstruction.** It was of interest to know if the exact reconstruction could be implemented in U-nets to avoid a large drop in performance for low noise levels. Figure 6.1-6 shows that the application of the general case equation (6.7) yields a PSNR ratio of approximately 1 when  $\sigma \rightarrow 0$ . Apart from that, the PSNR remains similar for higher, but seen, noise standard deviation values (for instance, there is a loss of only 0.03 dB at  $\sigma = 30$ ). However, the exact reconstruction is incompatible with generalization at high noise levels in the case of U-nets, as can be observed from the low performance in the interval  $[55; 100]$ .

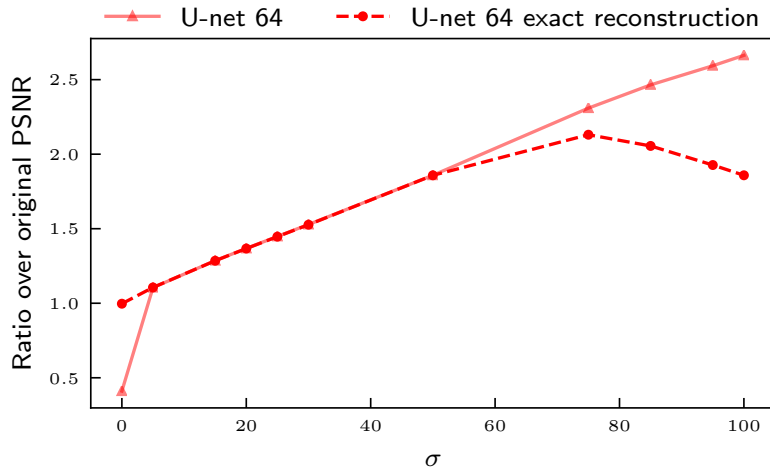


Figure 6.1-6: **Exact reconstruction for U-nets.** Ratio of the denoised image PSNR compared to the original noisy image PSNR for different standard deviations of the noise added to the test images for U-net with and without exact reconstruction. The train noise standard deviation range was  $[0; 55]$ .

**Generalization test: Denoising astrophysical images.** Another interesting test to evaluate how a network generalizes consists in denoising images that are different from the training dataset. This is for instance similar to what was done by Gottschling et al. [Got+20], where letters were added in the test image, while no image contains letters in the training data set. Here we applied our trained neural networks on a simulated astronomical image, contaminated with noise of a standard deviation  $\sigma = 50$ . This experience complements the previous generalization ones where the test images belong to the same class of natural images, but some noise levels were higher than those in the training data.

The astronomical image was firstly normalized such that every pixel had a value in the  $[-0.5; 0.5]$  interval. It was then fed to pre-trained U-net and learnlet models. The noisy, original and denoised versions, as well as the subtraction of the latter to the original image are presented in Figure 6.1-7. Using the MSE metric, learnlets (MSE of 20.83) perform almost twice as well as U-nets (MSE of 41.25).

Hence, similarly to the previous experiment, the generalization is much better for learnlets than for U-nets.

**Influence of the number of samples.** In a lot of Computer Vision problems, training data is scarce. It is reasonable to think that a small network (i.e. low quantity of parameters) would start performing better than a deeper one as fewer samples become available. To test this, three models were examined: two U-nets of different sizes (8 and 64) and learnlets without exact reconstruction. The first

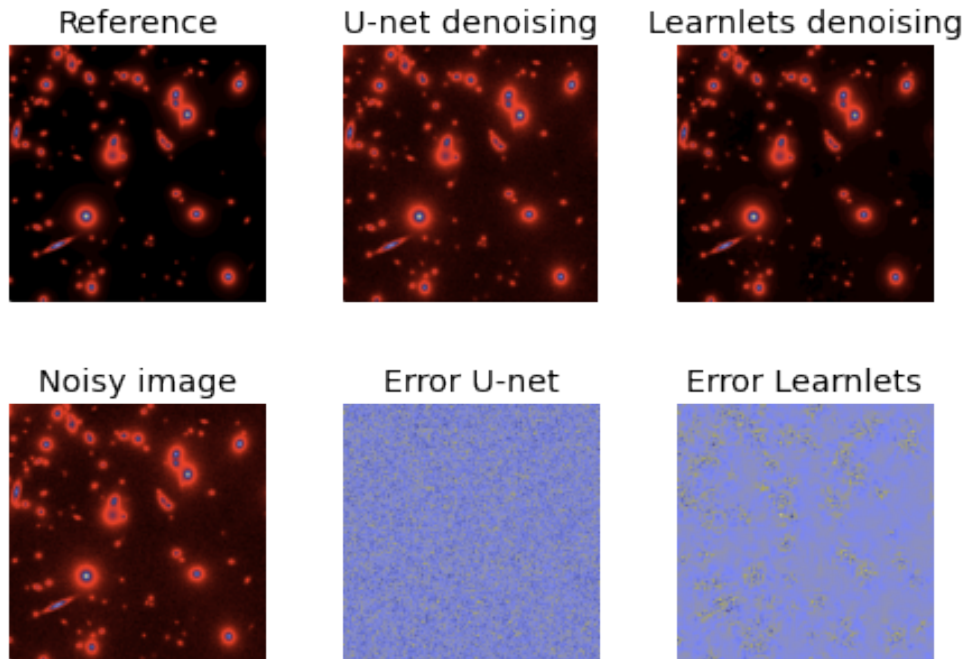


Figure 6.1-7: **Generalization to astrophysical images.** Denoising results for an astrophysical image contaminated with a noise of  $\sigma = 50$ . The last two images correspond to the subtraction of the original image to its denoised version.

aspect that can be mentioned about [Figure 6.1-8](#) is that for the three networks the PSNR does not vary significantly when reducing the original number of samples all the way down to 50. Despite learnlets overcoming U-nets with 64 base filters for the lowest number of samples considered, they fail to outperform a U-net model with 8 base filters. It can be inferred that a reduced number of parameters tends to improve the robustness of a given neural network to the number of samples. In other words, relatively few samples are required to obtain top performance.

### Qualitative results

**Comparison with other methods.** The [Figure 6.1-9](#) shows that the learnlets suffer from some drawbacks of the wavelets like the creation of artifacts in the high frequency parts of the image. However, the results are less blurred in comparison. Compared to the U-net, the learnlets are clearly suffering visually from a loss of contrast. This is a known effect of the soft thresholding which inherently biases the results. This could be improved by the use of reweighting [[CWB08](#)] to further approach the hard-thresholding, which does not bias the results.

#### 6.1.7 . Conclusions

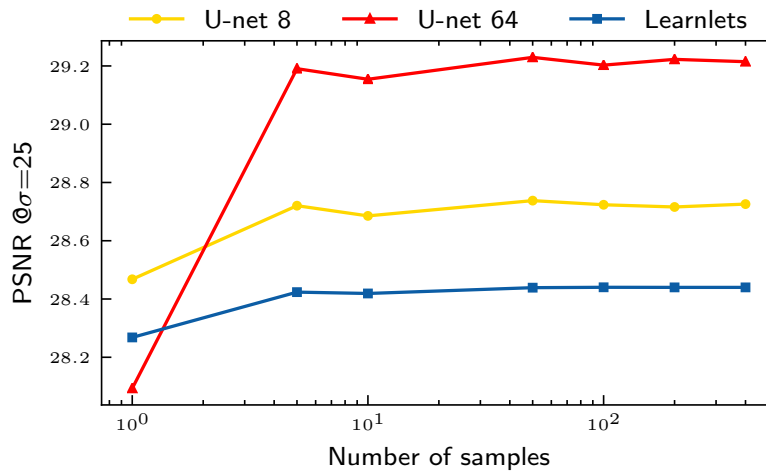


Figure 6.1-8: **Low-data regimes.** The PSNR of the denoised image at  $\sigma = 25$  added to the test images as a function of the number of samples used during training. The train noise standard deviation range was  $[0; 55]$ .

Pushing the limits of sparsity using massive learning and training data, we have proposed a novel neural network architecture – named *Learnlets* – with the following properties:

- Although their performances are inferior to U-nets, learnlets generalize better on noise levels that were not present in the training data and in the exact reconstruction domain. Additionally, they perform better on the astronomical image which is different from the images present in the training dataset. In this case, the U-net’s result is very poor.
- Learnlets can be forced to guarantee exact reconstruction when no thresholding is applied. This allows an embedding of the learnlets in applications where there is a need for guarantees of retrieval like in medical imaging. In contrast, U-nets suffer from a loss of performance at high noise levels.

Learnlets therefore bridge the gap between parsimony and neural networks, by combining massive learning and the computing power of GPUs as in neural networks, but keeping a perfect understanding of how results are obtained, with all the theoretical guarantees existing in the area of parsimony. Learnlets do clearly not outperform U-nets in denoising images compatible with the training dataset, which would indicate that massive learning and large datasets are not enough to explain the difference between sparse techniques and neural networks. The highly nonlinear processing in U-nets brings certainly a critical aspect in achieving high quality results.

Our main message is that we have clearly identified in this study a trade-off to be made in any application between performance and generalization. For performance, standard U-nets should clearly be chosen, while if the generalization is important, learnlets give the security of sparse techniques, using massive learning and GPU tools.

The future directions of this work are to try to adapt what has been successful in the sparse domain to this network. For example, reweighting [CWB08] could help us to get rid of the loss of contrast. Curvelet filters [SCD02] could also be used as a good initialization or as complementary filters for the analysis. Apart from that, just like with the wavelets, different types of noise – such as Poisson or spatially non-uniform white Gaussian noise – could be taken into account with a single model when implemented in an undecimated way, by adapting the thresholding function to the noise. Finally, it would be interesting to study the impact of incorporating learnlets as building blocks of the DIDN architecture [YPJ19].

## 6.2 . Denoising Score-Matching for Uncertainty Quantification in Inverse Problems

### 6.2.1 . Introduction

In this section, we discuss how to build uncertainty quantification models for MRI reconstruction. One promising lead for this is to be able to generate multiple possible reconstructions from a single set of measurements. This enables users to recognize what parts of the reconstructed image are constrained by the data, and what parts are derived from the prior model. A promising technique for this goal is constrained generative modeling.

Generative modeling has enjoyed remarkable success in recent years with models such as GANs [Goo+14] reaching extremely high quality results on complex high resolution images [Kar+20]. Yet, GANs are still prone to issues including unstable training and mode collapse, i.e. a lack of diversity in generated images. In addition, GANs only provide a convenient way to sample from the learned distribution, they do not give access to the density function itself. Another class of commonly used models, Variational Auto-Encoders (VAEs) [KW14] have also been able to reach high quality sample resolutions [VK20], but can only provide a bound on the density function of the model.

In some practical applications of these generative models however, the most desirable aspect is not necessarily to be able to sample from the model, or directly have access to its density functions, but have access to gradients of the log density function, i.e. its score. In particular, in this section we focus on the generic problem of performing Bayesian inference to solve an inverse problem (i.e. deconvolution, denoising, inpainting, etc.), using a learned generative model as a Bayesian prior. Such problems can typically be solved by gradient-based inference methods such as Variational Inference (VI), and Langevin or Hamiltonian Monte-Carlo (HMC),

but all these methods have in common that they *only rely on the score of the generative model*.

Recent work [Lim+20; SE19] has investigated various approaches to directly train a deep neural network (coined, a *score network*) to estimate the score of a density function instead of the density itself. One of the most promising and scalable approach relies on denoising score-matching, where the score network is trained using a denoising loss on an augmented dataset where various amounts of Gaussian noise is added to the data. Not only has this approach already been demonstrated to reach convincing results in score estimation, but Song et al. [SE20] for instance demonstrated that these score networks can be used to sample high resolution natural images through Langevin dynamics.

In this section, we develop the first application of deep DSM to MRI reconstruction and its use for Uncertainty Quantification (UQ) in the context of imaging inverse problems.

### 6.2.2 . Related Works

**Score Matching.** Score Matching was introduced by Hyvärinen [Hyv05]. It was later revisited by Vincent [Vin11] and Alain et al. [AB13] in the context of Deep Learning and Denoising Auto-Encoder (DAE) in particular. Recently, Song et al. [SE19] applied these findings to images showing the potential of this approach combined with sampling techniques. They refined their work [SE20] to take it to high dimension and provide some techniques for hyperparameter setting.

**P&P Priors.** The idea of using a denoising model as a prior for inverse problems solving was introduced by Venkatakrisnan et al. [VBW13] who used non-learned denoisers to replace the proximity operator in the ADMM algorithm [Boy+11]. More recently, Meinhardt et al. [MMC17] have made use of learned denoiser networks in P&P inverse problems solving. Annealed HMC sampling can be seen as a fuzzy-version of these approaches with theoretical grounds.

**Deep UQ in Inverse Problems.** The idea of using generative models for inverse problems such as data imputation and denoising was for instance suggested in an early VAE work [RMW14]. In this approach, a generative network is first trained on high quality data, and inference is then performed in the latent space of the model typically using VI. The main advantage comes from the reduced dimensionality of the parameter space. Recent examples of this approach include the works of Wu et al. [WDS18] and Böhm et al. [BLS19], and an application to MRI was carried out by Edupuganti et al. [Edu+20]. An important aspect of this approach is that it relies on the known likelihood of observations.

A different approach was proposed by Adler et al. [AÖ18a] based on a Conditional Wasserstein Generative Adversarial Network (cWGAN). In this formulation, a generative model is trained to sample high quality images conditioned on degraded



observations. At test time, independent samples from the posterior distribution induced by the GAN are obtained by simply sampling different latent space variables. Similar frameworks based on other conditional generative models have also been proposed, using conditional Normalizing Flows [Den+20], or using conditional VAEs [Ton+19]. While this conditional generative model approach allows for fast inference, thanks to ancestral sampling, it is worth noting that none of these models include an explicit *data consistency* step, nor do they make use of a test-time likelihood. As a direct consequence, for instance, these models cannot be used on data with different noise levels as seen during training.

### 6.2.3 . Deep DSM for Posterior Inference

Our main objective in this section is to perform probabilistic inference over a Bayesian model typically represented as

$$p(\mathbf{x} | \mathbf{y}) \propto p(\mathbf{y} | \mathbf{x}) p(\mathbf{x}) \quad (6.8)$$

where  $\mathbf{y}$  are some measurements and the posterior  $p(\mathbf{x} | \mathbf{y})$  is the distribution of possible solutions  $\mathbf{x}$  compatible with observations and  $p(\mathbf{x})$  the prior knowledge. The problem-specific likelihood  $p(\mathbf{y} | \mathbf{x})$  encodes the forward process of the model and accounts for observational noise, while the prior  $p(\mathbf{x})$  encapsulates any a priori information we have on the solution of the problem. Our goal for UQ is to sample solutions  $\mathbf{x}$  belonging to that posterior. Multiple inference techniques can be leveraged for sampling from this posterior  $p(\mathbf{x} | \mathbf{y})$ , but for high-dimensional problems modern techniques rely on gradient-based methods, including Variational Inference [Hof+13] and Langevin Diffusion, or HMC [Nea11]. All of these techniques have in common that they only require having access to the score  $\nabla_x \log p(x)$  of the target distribution (which in our case is the posterior distribution  $p(\mathbf{x} | \mathbf{y})$ ). Two terms will contribute to the score of the posterior distribution in Equation 6.8, the score of the likelihood, and the score of the prior. In many problems, such as in the MRI problem presented later in this work, the likelihood score can be derived analytically, only the score of the prior remains unknown but can be learned from data by score matching.

**Deep DSM.** As originally identified by Vincent [Vin11] and Alain et al. [AB13], the score of a given target distribution  $P$  can be modeled using a DAE, i.e. by introducing an auto-encoding function  $r : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$  trained to reconstruct under an  $\ell_2$  loss a true  $\mathbf{x} \sim P$  given a noisy version  $\mathbf{x}' = \mathbf{x} + \mathbf{n}$  with  $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ . An optimal denoiser  $r^*$  would then be achieved for:

$$r^*(\mathbf{x}', \sigma) = \mathbf{x}' + \sigma^2 \nabla_x \log p_{\sigma^2}(\mathbf{x}') \quad (6.9)$$

where  $p_{\sigma^2} = p * \mathcal{N}(0, \sigma^2)$ . In other words, the optimal denoiser is closely related to the score we wish to learn and when the noise variance  $\sigma^2$  tends to zero, should

exactly match the score of the target density. In practice to learn this score efficiently, we adopt the residual noise-conditional **DSM** technique proposed by Lim et al. [Lim+20], and train a model to minimize the following **DSM** loss:

$$\mathcal{L}_{DSM} = \mathbb{E}_{\mathbf{x} \sim P} \mathbb{E}_{\substack{\mathbf{u} \sim \mathcal{N}(0, I) \\ \sigma_s \sim \mathcal{N}(0, s^2)}} \|\mathbf{u} + \sigma_s \mathbf{r}_\theta(\mathbf{x} + \sigma_s \mathbf{u}, \sigma_s)\|_2^2 \quad (6.10)$$

In this formulation, the network  $\mathbf{r}_\theta(\mathbf{x}, \sigma)$  is now directly modeling the score  $\nabla_{\mathbf{x}} \log p_{\sigma^2}(\mathbf{x})$  of the Gaussian-convolved target distribution. Note that the noise level  $\sigma_s$  can be negative intentionally. This was recommended by Lim et al. [Lim+20] in order for the network to learn interpolation rather than extrapolation when  $\sigma_s$  tends to zero. This does not affect the noise distribution.

**Annealed HMC Sampling.** Given the noise-conditional neural scores learned with the procedure described above, it is now possible to use a variety of inference methods to access the Bayesian posterior. In this section, we adopt an annealed **HMC** procedure which provides an efficient way to obtain parallel batches of independent samples from the target posterior despite the high dimensionality of the problem. This is a sampling procedure closely related to the Annealed Langevin Diffusion proposed by Song et al. [SE19; SE20], but benefits from the faster Hamiltonian dynamics and Metropolis-Hastings calibration.

To build our procedure, we consider a Gaussian-convolved version of our target density:

$$\log p_{\sigma^2}(\mathbf{x} | \mathbf{y}) = \log p_{\sigma^2}(\mathbf{y} | \mathbf{x}) + \log p_{\sigma^2}(\mathbf{x}) + cst \quad (6.11)$$

where  $\sigma^2$  plays the role of the inverse temperature found in classical annealing. The likelihood can be obtained analytically, and in the case of a Gaussian likelihood takes the following form:  $\log p_{\sigma^2}(\mathbf{y} | \mathbf{x}) = -\frac{\|\mathbf{x} - f(\mathbf{x})\|_2^2}{2(\sigma_n^2 + \sigma^2)} + cst$ , where  $\sigma_n^2$  is the noise variance in the measurements. As for the prior term, the noise-conditional score network introduced above already models the score of  $\log p_{\sigma^2}$ . This distribution is gradually annealed to low temperatures and the chain progressively moves towards a point in the target distribution.

#### 6.2.4 . Application to Bayesian Inverse Problems

**The MRI Problem.** **MRI** is a non-invasive modality used to probe soft tissues. Compressed sensing, introduced for **MRI** by Lustig et al. [LDP07] is used to reduce its significant acquisition time, and recently, deep learning approaches [Sch+18; Ham+18; Pez+20] have been shown to perform extremely well on the reconstruction problem. The idealized reconstruction problem is usually formalized in the following way, for the single coil setting, with uniform acquisition:

$$\mathbf{y} = M_\Omega \mathbf{F} \mathbf{x} + \mathbf{n} \quad (6.12)$$

where  $\mathbf{y}$  is the acquired Fourier coefficients, also called the k-space data,  $M_\Omega$  is a mask,  $\mathbf{F}$  is the classical 2D **FT**,  $\mathbf{x}$  is the anatomical image, and  $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2)$  is

measurement noise (we set  $\sigma_n = 0.1$  in our experiments). The data we consider is the single coil data with the PD contrast from the fastMRI dataset introduced by Zbontar et al. [Zbo+18]. For comparison with deep learning approaches, we use the state-of-the-art Primal-Dual net enhanced with a U-net for image correction (UPDNet) [ZCS20c], which is basically an *XPDNet* with a U-net instead of a *MWCNN*. The undersampling was done retrospectively using an AF of 4 and a random mask as described for knee images by Zbontar et al. [Zbo+18].

**Results.** We use a simple residual U-net inspired by Ronneberger et al. [RFB15] with ResNet building blocks from the work of He et al. [He+16] as our score network. In order to promote the regularity of the learned scores, we regularize the spectral norm of each convolutional layer, and we find that setting the spectral norm to  $\simeq 2$  yields the best results. More details about the network architecture and training can be found in the Appendix, Table B. We then sample from the posterior following our annealed HMC procedure down to relatively low temperatures, and apply one final denoising step on the last sample from the chain, following the Expected Denoised Sample (EDS) scheme [Jol+21]. Figure 6.2-10 compares samples from the MRI posterior to the ground truth (top left), zero-filled image (top second left), and UPDNet reconstruction (top center). We see that although individual samples carry slightly fewer details than the neural network reconstruction, confidence in any particular part of the image can be gauged by looking for variability or stability across multiple independent posterior samples. We highlight in particular the red region, where posterior samples show significant variability, indicating that this part of the image is poorly constrained by data. In contrast, a direct neural network reconstruction (top center) does not match the ground truth in that region, and does not provide an estimate of uncertainty which may lead a physician to misinterpret the image.

### 6.2.5 . Conclusions and Discussions

We have presented in this section the first (to the best of our knowledge) instance of a framework for Bayesian inverse problems based on Deep DSM and applied to MRI reconstruction and UQ. We illustrated the merits of this approach on an MRI example where the ability to sample from the full posterior highlights what features present in a reconstruction are not actually constrained by data. We also showed that the Deep DSM approach can effortlessly be used for dimensions as high as  $320 \times 320 \times 2$ . It is therefore much more scalable than GAN- or VAE-based models. We note however that finding an optimal HMC annealing schedule and temperature-adaptive step-size proved difficult. In general, we did not manage to lower the temperature below a certain level, prompting us to resort to a denoising step on the last samples from our HMC chains. This was a direction for further research. The new developments that occurred in score-based generative models, namely Stochastic Differential Equations (SDE) based models, have

allowed to lift these problems. Some works have leveraged this new technique to solve inverse problems in imaging and applied it to MRI [Son+21; CY21].

## 6.3 . Is good old GRAPPA dead?

### 6.3.1 . Introduction

One of the most challenging aspects of building better MRI reconstruction is evaluation. We discussed this point in detail in section 2.4, and illustrate partially this pain point in this section.

In this section, we compare a state-of-the-art approach, the XPDNet [ZCS20c], to GRAPPA [Gri+02] on the task of reconstructing periodically undersampled MR images in different qualitative settings. This type of comparison to GRAPPA had not been performed before, even though GRAPPA is used in all the Siemens scanners, the most distributed in the world, as the default method for MR image reconstruction in the case of periodic undersampling (and similar approaches for other manufacturers).

We show that for this algorithm a visual evaluation is necessary. This point is critical, because replacing existing algorithms with neural networks will need comparative evaluation. We also show how the XPDNet can without adjustment be used to reconstruct prospectively accelerated data.

### 6.3.2 . Methods

**Network.** The XPDNet is a type of unrolled network that secured the second place in the 2020 fastMRI brain reconstruction challenge [Muc+21]. Very basically, it unrolls the PDHG [CP11] algorithm using a MWCNN [Liu+18] as the learned proximity operator. It has 25 unrolled iterations, and also features a sensitivity maps refinement module. Two networks were trained for AFs 4 and 8, using retrospectively undersampled data from the fastMRI dataset [Zbo+18] with equidistant Cartesian masks.<sup>3</sup> We chose to use non fine-tuned versions of the networks (i.e. trained on the four available imaging contrasts).

**GRAPPA.** We used the vanilla version of GRAPPA without noise handling. We use kernels that span 5 points in the readout direction and 2 in the phase direction. We manually set the regularization parameter  $\lambda > 0$  to obtain the best compromise between quantitative and qualitative evaluation, therefore biasing the analysis towards GRAPPA.<sup>4</sup> We leave the analysis of a smart setting of  $\lambda$  for future works.

**Data.** We used 3 data sets to perform our comparison on:

---

<sup>3</sup>[facebookresearch/fastMRI/issues/54](https://facebookresearch/fastMRI/issues/54)

<sup>4</sup>[youtube.com/watch?v=PngT6chFy6c](https://youtube.com/watch?v=PngT6chFy6c)

- a brain slice from the fastMRI validation data set [Zbo+18] (the state-of-the-art network was trained on the training data set), with T2 contrast, **retrospectively** undersampled at AFs 4 and 8;
- a brain slice acquired at a different resolution ( $0.25mm \times 0.25mm$ ) using a different magnetic field strength (7 T), orientation and AF than the fastMRI brain data set and featuring the cerebellum (not present in the fastMRI brain data set), with T2 contrast, **prospectively** undersampled at AF 2 – this allows us to test the robustness of the network to somewhat unseen settings [Mar+16];
- a NIST phantom, **prospectively** undersampled at AF 8, acquired at 3 T with 64 coils and a matrix size of  $256 \times 256$ .

All the data is periodically undersampled with an ACS.

**Code.** We use the code of Zaccharie Ramzi et al. [ZCS20c] for the network.<sup>5</sup> We used our own implementation of GRAPPA with a TensorFlow backend.<sup>6</sup>

### 6.3.3 . Results

**On the fastMRI brain slice.** At AF 4, the quantitative results seem to show that the XPDNet has an overwhelmingly better image quality than GRAPPA. However, upon visual inspection of the images available in Figure 6.3-11, we see that the image reconstructed by GRAPPA is only degraded by some noise not deteriorating its interpretability. At AF 8, the quantitative metrics once again show a clear advantage of the XPDNet over GRAPPA. This time, it is clearly confirmed by the visual inspection of the images presented in Figure 6.3-12.

**On the out-of-distribution brain slice.** The image reconstructed with the XPDNet shows some faint smoothing in the cerebellum as shown in the bottom row of Figure 6.3-13. However, the overall image is artifacts-free and very difficult to distinguish from the GRAPPA -reconstructed one.

**On the NIST phantom.** The phantom reconstructed using GRAPPA and XPDNet at AF 8 are very poor. For GRAPPA, the noise completely obfuscates the signal while for XPDNet the artifacts are present everywhere as can be seen in Figure 6.3-14.

---

<sup>5</sup>[github.com/zaccharieramzi/fastmri-reproducible-benchmark](https://github.com/zaccharieramzi/fastmri-reproducible-benchmark)

<sup>6</sup>[github.com/zaccharieramzi/grappa](https://github.com/zaccharieramzi/grappa)

**Original image**



**Noisy image**



**Wavelets: Image denoised**



**Learnlets: Image denoised**



**U-net: Image denoised**



Figure 6.1-9: **Denoising results for a specific image in the BSD68 dataset.** The noise standard deviation used was of 30. Parameters used for the methods are the same as for [Figure 6.1-3](#).

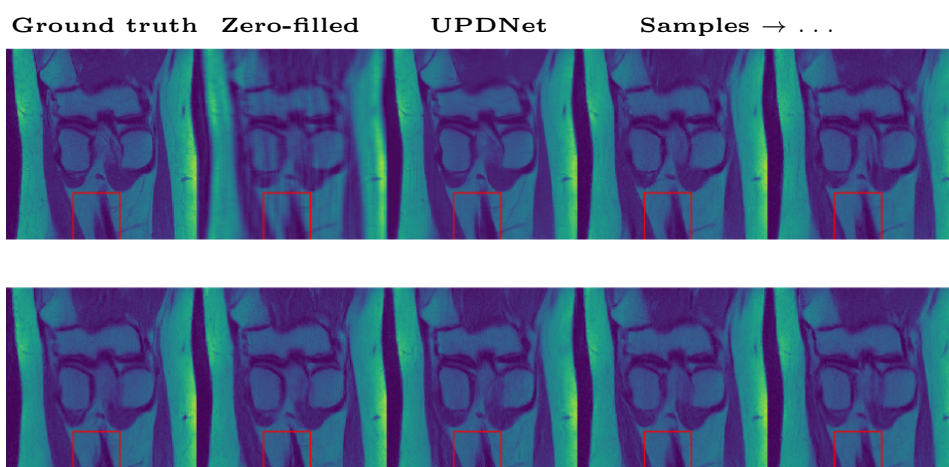


Figure 6.2-10: **Bayesian posterior sampling for MRI reconstruction.** The top leftmost image is the ground truth image. The top second to the left image is the zero-filled retrospectively undersampled image  $F^T y$ . The top third to the left image is the reconstruction of the undersampled image by the UPDNet. All the other images are denoised samples from the estimated posterior distribution obtained by a tempered HMC. The zero-filling achieves a PSNR of 25.55 dB, each sample 27.63 dB on average, the mean of the samples 30.04 dB and the neural network 32.15 dB. A zoom of the region in the red square is provided in the Appendix (Table A). An animation of posterior samples is available online [▶](#).



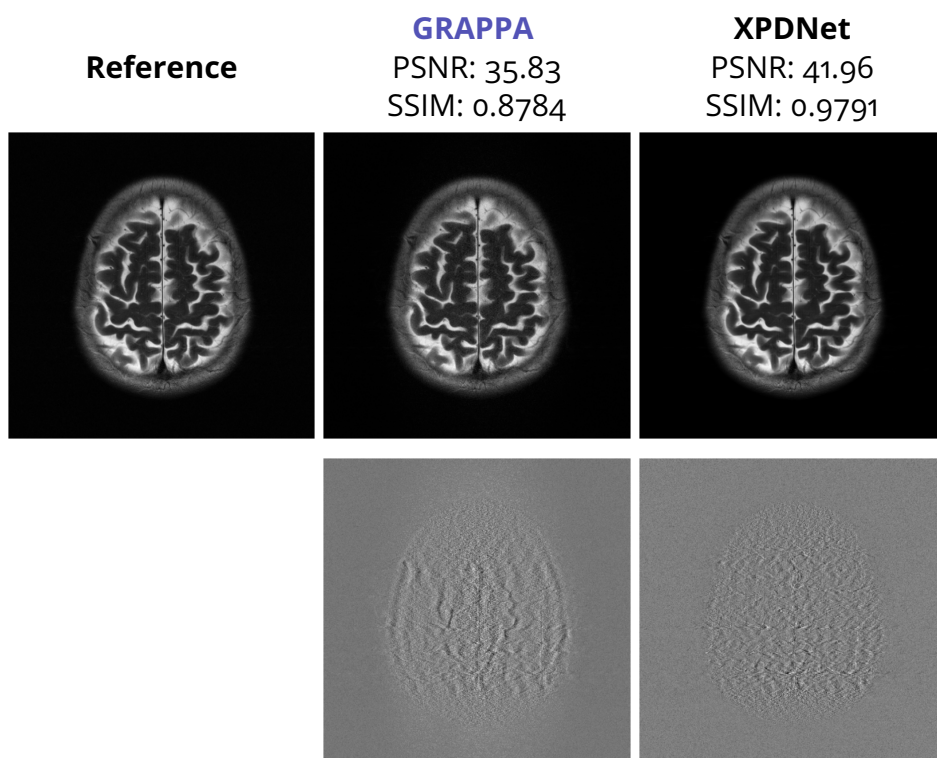


Figure 6.3-11: **Magnitude reconstruction results for a specific fast-MRI slice at AF 4.** The top row represents the reconstruction using the different methods, while the bottom one represents the error when compared to the reference.



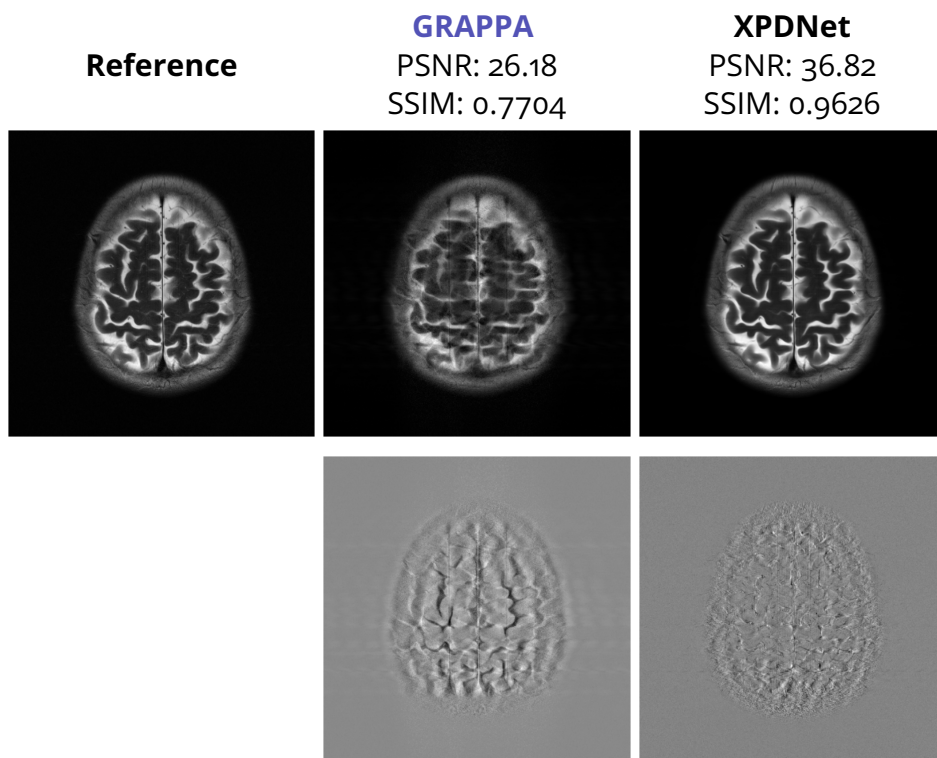


Figure 6.3-12: **Magnitude reconstruction results for a specific fast-MRI slice at AF 8.** The top row represents the reconstruction using the different methods, while the bottom row represents the error when compared to the reference.

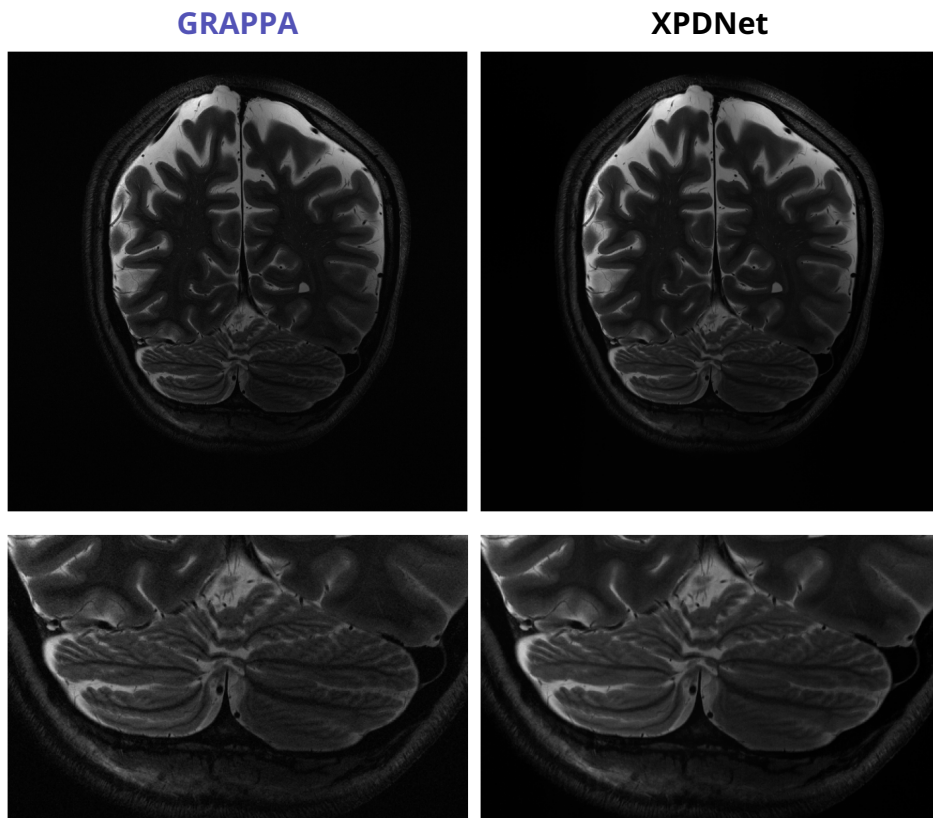


Figure 6.3-13: **Magnitude reconstruction results for a brain acquired at AF 2, contrast T2, and field strength of 7T.** The top row represents the reconstruction using the different methods, while the bottom one represents a zoom in the cerebellum region, an anatomical feature that was not present in the XPDNet training set.

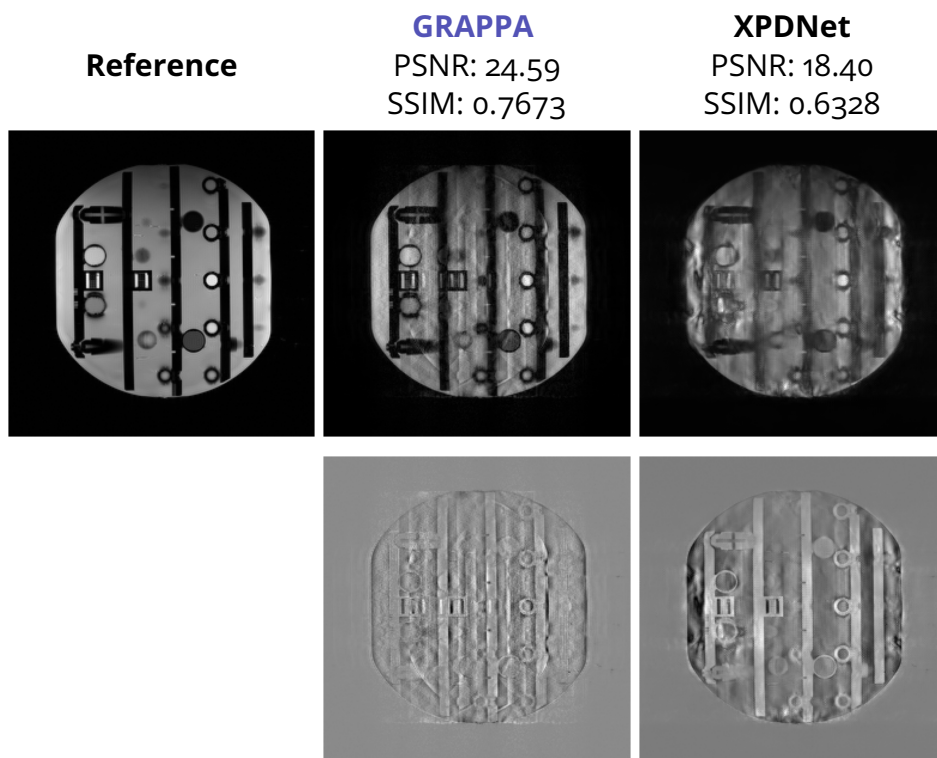


Figure 6.3-14: **Magnitude reconstruction results for a phantom acquired at AF 8.** The top row represents the reconstruction using the different methods, while the bottom one represents the error when compared to the reference.

#### 6.3.4 . Conclusion and Discussion

The Deep Learning techniques seem ready for a substitution test at AF 4, however, they do not seem to provide an overwhelming advantage over GRAPPA visually. The AF of 8 looks like an attainable target, and it would drastically improve the image quality when compared to GRAPPA, even when using the latest noise handling techniques. We also showed that the XPDNet is robust enough to be adapted to settings relatively different from the training distribution. However, if trained on brains a network cannot reconstruct objects that are too dissimilar, like phantoms. We conclude that it is therefore important to test visually the results of a reconstruction network at low AFs to measure the difference compared to GRAPPA, and that the high AFs are currently the real target for Deep Learning.

This comparison demands other types of robustness and sanity tests such as (but not limited to) receiver array coil design, SNR level, contrasts, organs, and orientation.





## 7 - New learning paradigms for very deep networks

### Chapter Outline

|       |  |     |
|-------|--|-----|
| 7.1   | SHINE: SHaring the INverse Estimate from the forward pass for bilevel optimization and implicit models . . . . . | 146 |
| 7.1.1 | Introduction . . . . .   | 146 |
| 7.1.2 | Hypergradient Optimization with Approximate Jacobian Inverse . . . . .   | 148 |
| 7.1.3 | Results . . . . .  | 154 |
| 7.1.4 | Conclusion and Discussion . . . . .  | 158 |
| 7.2   | Other paradigms for memory reduction when training neural networks . . . . .                                     | 159 |
| 7.2.1 | Gradient checkpointing . . . . .   | 159 |
| 7.2.2 | Invertible Networks . . . . .  | 159 |
| 7.2.3 | IFT-based networks . . . . .   | 161 |

The first section of this chapter was accepted as a spotlight to peer-reviewed conference:

**Zaccharie Ramzi**, F. Mannel, S. Bai, J.-L. Starck, P. Ciuciu and T. Moreau. “SHINE: SHaring the INverse Estimate from the forward pass for bi-level optimization and implicit models”. In: *International Conference on Learning Representations*. 2022. Spotlight

**I**N order to maintain excellent performances in terms of image quality while drastically accelerating (i.e. undersampling) data acquisition, deep learning models for MRI reconstruction will most likely need to be very deep. The most significant limitation to this evolution is currently the fact that the memory required for training grows with the depth of the network. This growth is driven not by the size of the model weights but by the activations as we explained in [paragraph 3.2.2](#).

As this problem is not specific to MRI reconstruction or even to inverse problems solving, a lot of effort has been put in trying to reduce the memory needed for training, by relying on fewer activations. However, most of the current techniques rely on trading off memory for computation. In this chapter, we will first introduce an original approach that enables faster training for DEQs [BZK19; BKK20], a

type of models enabling memory-less training. These models are an instance of the implicit DL framework. In this framework, the output of the model is not defined by an explicit sequence of operations, but rather as the solution to a parametrized equation. Thanks to this formulation, it is then possible to use the [Implicit Function Theorem \(IFT\)](#) to compute the derivatives of the parameters without having to rely on activations. In turn, this allows memory-less training. We will then describe other frameworks alleviating the memory requirements for training deep neural networks.

## 7.1 . SHINE: SHaring the INverse Estimate from the forward pass for bilevel optimization and implicit models

### 7.1.1 . Introduction

In general, the formulation of DEQs can be cast as a bilevel problem of the following form:

$$\arg \min_{\theta} \mathcal{L}(z^*) \quad \text{subject to} \quad g_{\theta}(z^*) = 0 \quad (7.1)$$

We will refer to the root finding problem  $g_{\theta}(z^*) = 0$  as the *inner problem*, and call its resolution the *forward pass*. On the other hand, we will refer to  $\arg \min_{\theta} \mathcal{L}(z^*)$  as the *outer problem*, and call the computation of the gradient of  $\mathcal{L}(z^*)$  w.r.t.  $\theta$  the *backward pass*. The core idea for DEQs is that their output  $z^*$  is expressed as a fixed point of a parametric function  $f_{\theta}$  from  $\mathbb{R}^d$  to  $\mathbb{R}^d$ , i.e.,  $g_{\theta}(z^*) = z^* - f_{\theta}(z^*) = 0$ .<sup>1</sup> This model is said to have infinitely many weight-tied layers as  $z^*$  can be obtained by successively applying the layer  $f_{\theta}$  infinitely many times, provided  $f_{\theta}$  is contractive. In practice, DEQs' forward pass is not computed by applying successively the function but usually relies on [Quasi-Newton \(qN\)](#) algorithms, such as Broyden's method [[Bro65](#)], which approximates efficiently the Jacobian matrix  $\frac{\partial g_{\theta}}{\partial z}$  and its inverse for root-finding.

To compute DEQs' gradient efficiently and avoid high memory cost, one does not rely on back-propagation but uses the [IFT](#) [[KP13](#)] which gives an analytical expression of the Jacobian of  $z^*$  with respect to  $\theta$ ,  $\frac{\partial z^*}{\partial \theta}$ . While this method is memory efficient, it requires the computation of matrix-vector products involving the inverse of a large Jacobian matrix, which is computationally demanding. To make this computation tractable, one needs to rely on an iterative algorithm based on vector-Jacobian products, which renders the training particularly slow, as highlighted by the original authors [[BKK20](#)] (see also the breakdown of the computational effort in [Table A.0.1](#)).

Moreover, the formulation [Equation 7.1](#) allows us to also consider general bilevel problems such as hyperparameter optimization under the same framework. For instance, hyperparameter optimization for [Logistic Regression \(LR\)](#) can be

---

<sup>1</sup>Here, we do not explicitly write the dependence of  $f_{\theta}$  on the input  $x$  of the DEQ, usually referred to as the injection.

written as

$$\min_{\theta} \mathcal{L}_{\text{val}}(z^*) \quad \text{subject to} \quad z^* = \min_z r_{\theta}(z) \triangleq \mathcal{L}_{\text{train}}(z) + \theta \|z\|_2^2 \quad (7.2)$$

where  $\mathcal{L}_{\text{train}}$  and  $\mathcal{L}_{\text{val}}$  correspond to the training and validation losses from the LR problem [Ped16]. Here,  $z$  corresponds to the weights of the LR model while  $\theta$  is the regularisation parameter. As the training loss is smooth and convex, the inner problem can be written as in Equation 7.1 with  $g_{\theta} = \nabla_z r_{\theta}$  to fit Equation 7.1. Similarly to DEQ, the inner problem is often solved using qN methods, which approximate the inverse of the Hessian in the direction of the steps, such as the LBFGS algorithm [LN89], and the gradient computation suffers from the same drawback as it is also obtained using the IFT. Lorraine et al. [LVD20] review the different hypergradient approximations for bilevel optimization and evaluate them on multiple tasks.

With the increasing popularity of DEQs and the ubiquity of bilevel problems in machine learning, a core question is how to reduce the computational cost of the resolution of Equation 7.1. This would make these methods more accessible for practitioners and reduce the associated energy cost. In this section, we propose to exploit the estimates of the (inverse of the) Jacobian/Hessian produced by qN methods in the hypergradient computation. Moreover, we also propose extra updates of the qN matrices which maintain the approximation property in the direction of the steps, and ensure that the inverse Jacobian is approximated in an additional direction. In effect, we can compute the gradient using the inverse of the final qN matrix instead of an iterative algorithm to invert the Jacobian in the gradient’s direction, while stressing that the inverse of a qN matrix, and thus the multiplication with it, can be computed very efficiently.

We emphasize that the goal of this section is neither to improve the algorithms used to compute  $z^*$ , nor is it to demonstrate how to perform the inversion of a matrix in a certain direction as a standalone task. Rather, we are describing an approach that combines the resolution of the inner problem with the computation of the hypergradient to accelerate the overall process. Our work is the first to consider modifying the inner problem resolution in order to account for the bilevel structure of the optimization. The idea to use additional updates of the qN matrices to ensure additional approximation properties is not new, and it is also known that a full matrix inversion can be accomplished in this way. For instance, Gower et al. [GR17] used sketching to design appropriate extra secant conditions in order to obtain guarantees of uniform convergence towards the inverse of the Jacobian. The novelty in our work is that we integrate additional update to yield the inverse in a specific direction, which is substantially cheaper than computing the inverse. A concurrent work by Fung et al. [Fun+21] is also concerned with the acceleration of DEQs’ training, where the inverse Jacobian is approximated with the identity. Under strong contractivity and conditioning assumptions, it is proven that the resulting approximation is a descent direction and the authors show good



empirical performances for small scale problems.

The contributions of this section are the following:

- We introduce a new method to greatly accelerate the backward pass of DEQs (and generally, the differentiation of bilevel problems) using qN matrices that are available as a by-product of the forward computations. We call this method **SHINE** (**SH**aring the **IN**verse **E**stimate).
- We enhance this method by incorporating knowledge from the outer problem into the inner problem resolution. This allows us to provide strong theoretical guarantees for this approach in various settings.
- We additionally showcase its use in hyperparameter optimization. Here, we demonstrate that it provides a gain in computation time compared to state-of-the-art methods.
- We test it for DEQs for the classification task on two datasets, CIFAR and ImageNet. Here, we show that it decreases the training time while remaining competitive in terms of performance.
- We extend the empirical evaluation of the Jacobian-Free method to large scale **Multiscale Deep Equilibrium Network (DEQs)** and show that it performs well in this setting. We also show that it is not suitable for more general bilevel problems.
- We propose and evaluate a natural refinement strategy for approximate Jacobian inversion methods (both **SHINE** and Jacobian-Free) that allows a trade-off between computational cost and performances.

### 7.1.2 . Hypergradient Optimization with Approximate Jacobian Inverse

#### **SHINE: Hypergradient Descent with Approximate Jacobian Inverse**

**Hypergradient Optimization.** Hypergradient optimization is a first-order method used to solve Equation 7.1. We recall that in the case of smooth convex optimization,  $\frac{\partial g_\theta}{\partial z}$  is the Hessian of the inner optimization problem, while for deep equilibrium models, it is the Jacobian of the root equation. In the rest of this section, with a slight abuse of notation, we will refer to both these matrices with  $J_{g_\theta}$  whenever the results can be applied to both contexts. To enable Hypergradient Optimization, i.e. gradient descent on  $\mathcal{L}$  with respect to  $\theta$ , Bai et al. [BZK19, Theorem 1] show the following theorem, which is based on implicit differentiation [KP13]:

**Theorem 1** (Hypergradient [KP13; BZK19]) *Let  $\theta \in \mathbb{R}^p$  be a set of parameters, let  $\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}$  be a loss function and  $g_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  be a root-defining function. Let*

$z^* \in \mathbb{R}^d$  such that  $g_\theta(z^*) = 0$  and  $J_{g_\theta}(z^*)$  is invertible, then the gradient of the loss  $\mathcal{L}$  wrt.  $\theta$ , called Hypergradient, is given by

$$\frac{\partial \mathcal{L}}{\partial \theta} \Big|_{z^*} = \nabla_z \mathcal{L}(z^*)^\top J_{g_\theta}(z^*)^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z^*}. \quad (7.3)$$

---

**Algorithm 2 :** qN method to solve  $g_\theta(z^*) = 0$ .

---

**Result :** Root  $z^*$ , qN matrix  $B$

- 1  $b = \text{true}$  if using Broyden's method,  $b = \text{false}$  if using BFGS
- 2  $n = 0$ ,  $z_0 = 0$ ,  $B_0 = I$
- 3 **while** not converged **do**
- 4      $p_n = -B_n^{-1} g_\theta(z_n)$ ,  $z_{n+1} = z_n + \alpha_n p_n$  //  $\alpha_n$  can be 1 or determined by line-search
- 5      $y_n = g_\theta(z_{n+1}) - g_\theta(z_n)$
- 6      $s_n = z_{n+1} - z_n$
- 7     **if**  $b$  **then**
- 8          $B_{n+1} = \arg \min_{X: X s_n = y_n} \|X - B_n\|_F$
- 9     **else**
- 10          $B_{n+1} = \arg \min_{X: X = X^\top \wedge X s_n = y_n} \|X^{-1} - B_n^{-1}\|$  // The norm used in BFGS is a weighted Frobenius norm
- 11      $n \leftarrow n + 1$
- 12  $z^* = z_n$ ,  $B = B_n$

---

In practice, we use an algorithm to approximate  $z^*$ , and [Theorem 1](#) gives a plug-in formula for the backward pass. Note that this formula is independent of the algorithm chosen to compute  $z^*$ . Moreover, as opposed to explicit networks, we do not need to store intermediate activations, resulting in the aforementioned training time memory gain for DEQs. Once  $z^*$  has been obtained, one of the major bottlenecks in the computation of the Hypergradient is the inversion of  $J_{g_\theta}(z^*)$  in the directions  $\frac{\partial g_\theta}{\partial \theta} \Big|_{z^*}$  or  $\nabla_z \mathcal{L}(z^*)$ .

**qN methods.** In practice, the forward pass is often carried out with qN methods. For instance, in the case of bilevel optimization for LR, Pedregosa [[Ped16](#)] used L-BFGS [[LN89](#)], while for Deep Equilibrium Models, Bai et al. [[BZK19](#)] used Broyden's method [[Bro65](#)], later adapted to the multiscale case in a limited-memory version [[BKK20](#)].

These qN methods were first inspired by Newton's method, which finds the root of  $g_\theta$  via the recurrent Jacobian-based updates  $z_{n+1} = z_n - J_{g_\theta}(z_n)^{-1} g_\theta(z_n)$ . Specifically, they replace the Jacobian  $J_{g_\theta}(z_n)$  by an approximation  $B_n$  that is based on available values of the iterates  $z_n$  and  $g_\theta$  rather than its derivative. These  $B_n$ , called qN matrices, are defined recursively via an optimization problem with constraints called secant conditions. Solving this problem leads to expressing

$B_n$  as a rank-one or rank-two update of  $B_{n-1}$ , so that  $B_n$  is the sum of the initial guess  $B_0$  (in our settings, the identity) and  $n$  low-rank matrices (less than  $n$  in limited memory settings). This low rank structure allows efficient multiplication by  $B_n$  and  $B_n^{-1}$ . We now explain how the use of **qN** methods as inner solver can be exploited to resolve this computational bottleneck.

**SHINE.** Roughly speaking, our proposition is to use  $B^{-1} = \lim_{n \rightarrow \infty} B_n^{-1}$  as a replacement for  $\mathbf{J}_{g_\theta}(z^*)^{-1}$  in Equation 7.3, i.e. to share the inverse estimate between the forward and the backward passes. This gives the approximate Hypergradient

$$\mathbf{p}_\theta = \nabla_z \mathcal{L}(z^*) B^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z^*} \quad (7.4)$$

In practice, we will consider the nonasymptotical direction  $\mathbf{p}_\theta^{(n)} = \nabla_z \mathcal{L}(z_n) B_n^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z_n}$ .

Thanks to the Sherman-Morrison formula [SM50], the inversion of  $B_n$  can be done very efficiently (using scalar products) compared to the iterative methods needed to invert the true Jacobian  $\mathbf{J}_{g_\theta}(z^*)$ . In turn, this significantly reduces the computational cost of the Hypergradient computation.

**Relationship to the Jacobian-Free method.** Because  $B_0 = I$  in our setting, we may regard  $B$  as an identity matrix perturbed by a few rank-one updates. In the directions that are used for updates,  $B$  is going to be different from the identity, and hopefully closer to the true Jacobian in those directions. However, in all orthogonal directions we fall exactly into the setting of the Jacobian-Free method introduced by Fung et al. [Fun+21]. In that work,  $\mathbf{J}_{g_\theta}(z^*)^{-1}$  is approximated by  $I$ , and the authors highlight that this is equivalent to using a preconditioner on the gradient. Under strong assumptions on  $g_\theta$  they show that this preconditioned gradient is still a descent direction.

**Transition to the exact Jacobian Inverse.** The approximate gradient  $\mathbf{p}_\theta^{(n)}$  can also be used as the initialization of an iterative algorithm for inverting  $\mathbf{J}_{g_\theta}(z^*)$  in the direction  $\nabla_z \mathcal{L}(z^*)$ . With a good initialization, faster convergence can be expected. Moreover, if the iterative algorithm is also a **qN** method, which is the case in practice in the **MDEQ** implementation, we can use the **qN** matrix  $B$  from the forward pass to initialize the **qN** matrix of this algorithm. We refer to this strategy as the *refine strategy*. Because the refine strategy is essentially a smart initialization scheme, it recovers all the theoretical guarantees of the original method [Ped16; BZK19; BKK20].

## Convergence to the true gradient

To further justify and formalize the idea of **SHINE**, we show that the direction  $\mathbf{p}_\theta^{(n)}$  converges to the Hypergradient  $\frac{\partial \mathcal{L}}{\partial \theta} \Big|_{z^*}$ . We now collect the assumptions that will

be used for this purpose.

**Assumption 7.1.1** (Uniform Linear Independence (ULI) [LZZ98]). *There exist a positive constant  $\rho > 0$  and natural numbers  $n_0 \geq 0$  and  $m \geq d$  with the following property: For any  $n \geq n_0$  we can find indices  $n \leq n_1 \leq \dots \leq n_d \leq n + m$  such that, for  $\mathbf{p}_n$  defined in Algorithm 2, the smallest singular value of the  $d \times d$  matrix*

$$\left( \frac{\mathbf{p}_{n_1}}{\|\mathbf{p}_{n_1}\|}, \frac{\mathbf{p}_{n_2}}{\|\mathbf{p}_{n_2}\|}, \dots, \frac{\mathbf{p}_{n_d}}{\|\mathbf{p}_{n_d}\|} \right)$$

is no smaller than  $\rho$ .

**Assumption 7.1.2** (Smoothness and convergence to the fixed point). (i)  $\sum_{n=0}^{\infty} \|z_n - z^*\| < \infty$  for some  $z^*$  with  $g_{\theta}(z^*) = 0$ ; (ii)  $g_{\theta}$  is  $C^1$ ,  $\mathbf{J}_{g_{\theta}}$  is Lipschitz continuous near  $z^*$ , and  $\mathbf{J}_{g_{\theta}}(z^*)$  is invertible; (iii)  $\nabla_z \mathcal{L}$  is continuous, and  $\forall \theta$ ,  $\frac{\partial g_{\theta}}{\partial \theta}$  is continuous.

**Remark** The Assumption 7.1.2 (i) implies  $\lim_{n \rightarrow \infty} z_n = z^*$ . The existence of the Jacobian and its inverse are assumptions that are already made in the regular DEQ setting just to train the model.

**Theorem 2** (Convergence of SHINE to the Hypergradient using ULI) *Let us denote  $\mathbf{p}_{\theta}^{(n)}$ , the SHINE direction for iterate  $n$  in Algorithm 2 with  $b = \text{true}$ . Under Assumptions 7.1.1 and 7.1.2, for a given parameter  $\theta$ ,  $(z_n)$  converges  $q$ -superlinearly to  $z^*$  and*

$$\lim_{n \rightarrow \infty} \mathbf{p}_{\theta}^{(n)} = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{z^*}$$

*Proof.* From More et al. [MT76, Theorem 5.7] we obtain that  $\lim_{n \rightarrow \infty} \mathbf{B}_n = \mathbf{J}_{g_{\theta}}(z^*)$ . We can then conclude using the continuity of the inversion operator on the space of invertible matrices and of the right and left matrix vector multiplications. A complete proof is given in Appendix C.  $\square$

Theorem 2 establishes convergence of the SHINE direction to the true Hypergradient, but relies on Assumption 7.1.1 (ULI). While ULI is often used to prove convergence results for qN matrices [LZZ98; NW06; CGT91], it is a strong assumption whose satisfaction in practice is debatable [FBS93]. For Broyden’s method, Mannel [Man21a; Man21b; Man20] showed that ULI is violated in all numerical experiments, and they also proved that ULI is necessarily violated in certain settings (but the setting of this work is not covered). In the following we therefore derive results that do not involve ULI.

## Outer Problem Awareness

The ULI assumption guarantees convergence of  $\mathbf{B}_n^{-1}$  to  $\mathbf{J}_{g_{\theta}}(z^*)^{-1}$ . However, Equation 7.3 only requires the multiplication of  $\mathbf{J}_{g_{\theta}}(z^*)^{-1}$  with  $\frac{\partial g_{\theta}}{\partial \theta} \Big|_{z^*}$  from the right and  $\nabla_z \mathcal{L}(z^*)$  from the left.

**BFGS with OPA.** In order to strengthen [Theorem 2](#), let us consider the setting of bilevel optimization with a single regularizing hyperparameter  $\theta$ . There, the partial derivative  $\frac{\partial g_\theta}{\partial \theta}|_{z^*}$  is a  $d$ -dimensional vector, and it is possible to compute its approximation  $\frac{\partial g_\theta}{\partial \theta}|_{z_n}$  at a reasonable cost. We propose to incorporate additional updates of the [qN](#) matrix  $\mathbf{B}_n$  into [Algorithm 2](#) that improve the approximation quality of  $\mathbf{B}_n^{-1}$  in the direction  $\frac{\partial g_\theta}{\partial \theta}|_{z_n}$  (thus asymptotically in the direction  $\frac{\partial g_\theta}{\partial \theta}|_{z^*}$ ). Given a current iterate pair  $(z_n, \mathbf{B}_n)$ , these additional updates only change  $\mathbf{B}_n$ , but not  $z_n$ . We will demonstrate that a suitable update direction  $e_n \in \mathbb{R}^d$  is given by

$$e_n = t_n \mathbf{B}_n^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z_n}, \quad (7.5)$$

where  $(t_n) \subset [0, \infty)$  satisfies  $\sum_n t_n < \infty$ . This update direction will be used to create an extra secant condition  $\mathbf{X}^{-1}(g_\theta(z_n + e_n) - g_\theta(z_n)) = e_n$  for the additional update of  $\mathbf{B}_n$ . Since this extra update is based on the outer problem, we refer to this technique as [Outer-Problem Awareness \(OPA\)](#). The complete pseudocode of the [OPA](#) method in the LBFSG algorithm [[LN89](#)] is given in [Figure A](#).

We now prove that if extra updates are applied at a fixed frequency, then fast ( $q$ -superlinear) convergence of  $(z_n)$  to  $z^*$  is retained, while convergence of the [SHINE](#) direction to the true Hypergradient is also ensured. To show this, we use the following assumption.

**Assumption 7.1.3** (Assumptions for BFGS). *Let  $g_\theta(z) = \nabla_z r_\theta(z)$  for some  $C^2$  function  $r_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$ . Consider [Algorithm 2](#) with  $b = \text{false}$ . We assume some regularity on  $r$  and that an appropriate line search is used. An extended version of this assumption is given in [Theorem C](#) ([Assumption C.0.1](#)).*

**Theorem 3** (Convergence of [SHINE](#) to the Hypergradient for BFGS with [OPA](#)) *Let us consider  $\mathbf{p}_\theta^{(n)}$ , the [SHINE](#) direction for iterate  $n$  in [Algorithm 2](#) that is enriched by extra updates in the direction  $e_n$  defined in [Equation 7.5](#). Under [Assumptions 7.1.2](#) (ii-iii) and [7.1.3](#), for a given parameter  $\theta$ , we have the following: [Algorithm 2](#), for any symmetric and positive definite matrix  $\mathbf{B}_0$ , generates a sequence  $(z_n)$  that converges  $q$ -superlinearly to  $z^*$ , and there holds*

$$\lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{z^*} \quad (7.6)$$

*Proof.* It follows from known results that the extra updates do not destroy the  $q$ -superlinear convergence of  $(z_n)$ . The proof of [Equation 7.6](#) relies firstly on the fact that by continuity of the derivative of  $g_\theta$ , we have  $\lim_{n \rightarrow \infty} \frac{\partial g_\theta}{\partial \theta}|_{z_n} = \frac{\partial g_\theta}{\partial \theta}|_{z^*}$ . Due to the extra updates we can show convergence of the [qN](#) matrices to the true Hessian in the direction of the extra steps  $e_n$ , from which [Equation 7.6](#) follows. A full proof is provided in [Theorem C](#).  $\square$

**Remark** [Theorem 3](#) also holds without line searches (i.e.,  $\alpha_n = 1$  for all  $n$ ) and any  $C^2$  function  $r_\theta$  (such that  $g_\theta(z) = \nabla_z r_\theta(z)$ ) with locally Lipschitz continuous Hessian if  $z_0$  is close enough to some  $z^*$  with  $\nabla_z r_\theta(z^*) = 0$  and  $\nabla_{zz}^2 r_\theta(z^*)$  positive definite.

We note that [Theorem 3](#) guarantees fast convergence of the iterates  $(z_n)$  and that  $z_0$  does not have to be close to  $z^*$  for that guarantee. Also, there is no restriction on  $B_0$  other than being symmetric and positive definite (which is satisfied for our choice  $B_0 = I$ ). Finally, [Theorem 3](#) does not rely on ULI. From a practical standpoint we thus regard [Theorem 3](#) as a much stronger result than [Theorem 2](#).

**Adjoint Broyden with OPA.** It is not practical to use the partial derivative  $\frac{\partial g_\theta}{\partial \theta}$  in the DEQ setting because it is a huge Jacobian that we do not have access to in practice. In order to still leverage the core idea of OPA, we propose to use extra updates that ensure that  $B_n^{-1}$  approximates  $J_{g_\theta}(z^*)^{-1}$  in the direction  $\nabla_z \mathcal{L}(z^*)$  applied by left-multiplication, as required by [Equation 7.3](#). An appropriate secant condition is given by

$$\mathbf{v}_n^\top \mathbf{B}_{n+1} = \mathbf{v}_n^\top \mathbf{J}_{g_\theta}(z_{n+1}), \quad (7.7)$$

where

$$\mathbf{v}_n^\top = \nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1}. \quad (7.8)$$

To incorporate the secant condition [Equation 7.7](#), we use the Adjoint Broyden's method [[SGW10](#)], a qN method relying on the efficient vector-Jacobian multiplication by  $\mathbf{J}_{g_\theta}$  using auto-differentiation tools. To prove convergence of the SHINE direction for this method, we need the following assumption.

**Assumption 7.1.4** (Uniform boundedness of the inverse qN matrices). *The sequence  $(\mathbf{B}_n)$  generated by [Algorithm 2](#) satisfies*

$$\sup_{n \in \mathbb{N}} \|\mathbf{B}_n^{-1}\| < \infty.$$

**Remark** *Convergence results for qN methods usually include showing that [Assumption 7.1.4](#) holds, cf. Broyden et al. [[BDM73](#), [Theorem 3.2](#)] for Broyden's method and the BFGS method, respectively, Schlenkrich et al. [[SGW10](#), [Theorem 1](#)] for the Adjoint Broyden's method. It can also be proved that [Assumption 7.1.4](#) holds for globalized variants of these methods, e.g., for the line-search globalizations of Broyden's method proposed by Li et al. [[LFoo](#)]. We point out that [Assumption 7.1.1](#) entails  $\lim \mathbf{B}_n = \mathbf{J}_{g_\theta}(z^*)$  and thus  $\lim \mathbf{B}_n^{-1} = \mathbf{J}_{g_\theta}(z^*)^{-1}$ , so it is clearly stronger than [Assumption 7.1.4](#).*

**Theorem 4** (Convergence of SHINE to the Hypergradient for Adjoint Broyden with OPA) *Let us consider  $\mathbf{p}_\theta^{(n)}$ , the SHINE direction for iterate  $n$  in [Algorithm 2](#)*

with the Adjoint Broyden secant condition [Equation 7.7](#) and extra update in the direction  $v_n$  defined in [Equation 7.8](#). Under [Assumptions 7.1.2](#) and [7.1.4](#), for a given parameter  $\theta$ , we have  $q$ -superlinear convergence of  $(z_n)$  to  $z^*$  and

$$\lim_{n \rightarrow \infty} p_\theta^{(n)} = \left. \frac{\partial \mathcal{L}}{\partial \theta} \right|_{z^*}$$

*Proof.* The  $q$ -superlinear convergence of  $(z_n)$  follows from Schlenkrich et al. [[SGW10](#), Theorem 2]. To establish convergence of the SHINE direction, we proceed in three steps. First, it is shown that for  $\nabla_z \mathcal{L}(z^*) = 0$  the claim holds due to continuity and [Assumption 7.1.4](#). Then  $\nabla_z \mathcal{L}(z^*) \neq 0$  is considered, and it is proved that the desired convergence holds on the subsequence that corresponds to the additional updates. Lastly, this result is transferred to the entire sequence by involving the fixed frequency of the additional updates. The complete proof is provided in [Equation C](#).  $\square$

Using the Adjoint Broyden’s method comes at a computational cost. Indeed, because we now rely on  $J_{g_\theta}$ , we have to store the activations of  $g_\theta(z)$  (which has a computational cost in addition to a memory cost), but also perform the vector-Jacobian product in addition to the function evaluation.

### 7.1.3 . Results

We test our method in 3 different setups and compare it to the original iterative inversion and its closest competitor, the Jacobian-Free method [[Fun+21](#)]. We draw the reader’s attention to the fact that although the Jacobian-Free method [[Fun+21](#)] is used outside the assumptions needed to have theoretical guarantees<sup>2</sup> of descent, it still performs relatively well in the Deep Equilibrium setting. The same is true for SHINE: While the ULI assumption is not met (and we are in practice far from the fixed point convergence), it performs well in practice.

**Implementations.** All the bilevel optimization experiments were done using the HOAG code [[Ped16](#)],<sup>3</sup> which is based on the Python scientific ecosystem [[Har+20](#); [Vir+20](#); [Ped+11](#)]. Deep Equilibrium experiments were done using the PyTorch [[Pas+19](#)] code for MDEQ [[BKK20](#)],<sup>4</sup> which was distributed under the MIT license. Plots were done using Matplotlib [[Hun07](#)], with Science Plots style [[GP21](#)]. DEQ trainings were done in a publicly funded HPC, using nodes with 4 V100 GPUs.

In practice, we never reach convergence of  $(z_n)$ , hence the approximate gradient might be far from the true gradient. To improve the approximation quality, we now propose a variant of our method.

<sup>2</sup>See the results on contractivity in [Figure A.o.1](#).

<sup>3</sup>[github.com/fabianp/hoag](https://github.com/fabianp/hoag)

<sup>4</sup>[github.com/locuslab/mdeq](https://github.com/locuslab/mdeq)

**Fallback in the case of wrong inversion.** Empirically, we noticed that using  $B$  can sometimes produce bad approximations, although with very low probability. We propose to detect this with by monitoring a telltale sign based on the norm of the approximation, as we verified on several examples that cases with a huge norm compared to the correct inversion also had a very bad correlation with the correct inversion. In these cases, we can simply fall back onto another inversion method. For the Deep Equilibrium experiments, when the norm of the inversion using SHINE is 1.3 times above the norm of the inversion using the Jacobian-Free method (which is available at no extra computational cost), we use the Jacobian-Free inversion. We refer to this strategy as the *fallback strategy*.

## Bilevel optimization – Hyperparameter optimization in LR

We first test SHINE in the simple setting of bilevel optimization for  $\ell_2$ -regularized LR, using the code from Pedregosa [Ped16] and the same datasets. Convergence on unseen data is illustrated in Figure 7.1-1.<sup>5</sup> An acceptable level of performance is reached twice faster for the SHINE method compared to any other competitor. Another finding is that the refine strategy does not provide a definitive improvement over the vanilla version of SHINE. In order to verify that the performance gain of SHINE is not simply driven by truncated inversion, we also run HOAG with limited number of inversion iteration and showed that this degrades its performances (see HOAG limited backward in 14).

We also tested our implementation of OPA on the 20news dataset and present the results in Figure 7.1-2. In order to get a fair comparison, we implemented both SHINE, SHINE-OPA and HOAG using the same full Python code instead of relying on the original code which relied on the Fortran implementation of L-BFGS from Virtanen et al. [Vir+20]. While SHINE with OPA does not outperform the vanilla SHINE, it reaches similar performances, outperforming HOAG, and comes with strong theoretical grounding. Additional results on hyperparameter optimization for the regularized nonlinear least squares problem are available in subsection A.0.1.

We also showed on a smaller dataset, the breast cancer dataset [DG17], that OPA indeed ensures a better approximation of the inverse in the prescribed direction. For a given split of the data, we compared the quality of the approximation of the inversion in three different directions: a prescribed direction chosen randomly but used for the OPA update, the Krylov direction  $\left. \frac{\partial g_\theta}{\partial z} \right|_{z^*} (z_n - z_{n-1})$  and a random direction not used in the qN algorithm. The results for 100 runs with different random seeds are depicted in Figure 7.1-2, where we can observe that OPA indeed ensures a better inversion in the prescribed direction compared to a

---

<sup>5</sup>To facilitate the reader’s understanding of the figures, we plot the empirical sub-optimality, but we do remind them that there is no guarantee of convergence on held-out test data ; the kink present in the case of the real-sim dataset is an example of that.



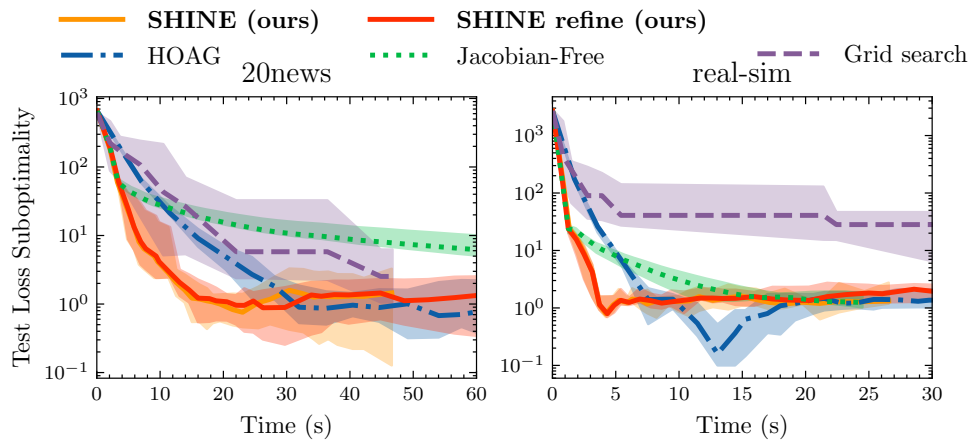


Figure 7.1-1: **Bilevel optimization:** Convergence of held-out test loss for different hyperparameter optimization methods on the  $\ell_2$ -regularized LR problem for the 2 datasets (20news [Lan95] and real-sim [Fan11]) SHINE achieves the best performances for both problems while the Jacobian-Free method is much slower, in particular on 20news. Note that the kink for HOAG on real-sim does not mean it is better as the optimization stops once the validation loss has converged and not the test one. The typical loss order of magnitude is  $10^2$ . An extended figure with more methods is provided in 14.

random direction. We also notice that a poor direction for the inversion seems correlated with a small magnitude.

## Deep Equilibrium Models

Next, we tested SHINE on the more challenging DEQ setup. Two experiments illustrate the performance of SHINE on the image classification task on two datasets. For both datasets, we used the same model configuration as in the original MDEQ paper [BKK20] and did not fine tune any hyperparameter. For the different DEQ training methods, models for a given seed share the same unrolled-pretraining steps. We do not include OPA in the DEQ results because while the gradients are well correlated with the true ones (see Figure A-8), we observe a sharp initial performance drop that reduces its performance on Imagenet. We provide partial results in Table A.0.1.

**CIFAR-10.** The first dataset is CIFAR-10 [Kri09] which features 60,000  $32 \times 32$  images representing 10 classes. For this dataset, the size of the multiscale fixed point is  $d = 50k$ . We train the models for five different random seeds.

The results in Figure 7.1-3 show that for the vanilla version, SHINE slightly outperforms the Jacobian-Free method [Fun+21]. Additionally, our results sug-

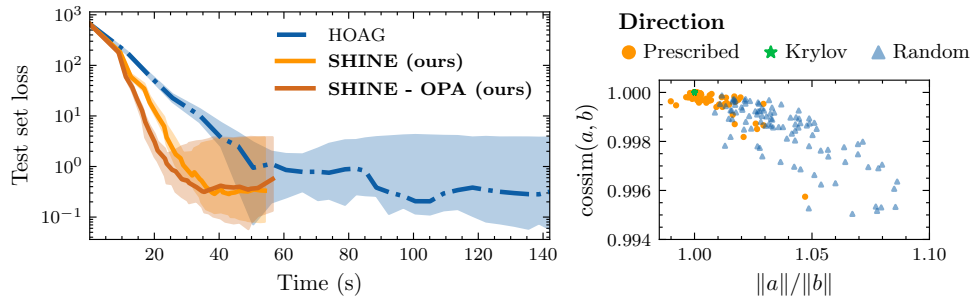


Figure 7.1-2: **Bilevel optimization with OPA:** (left) Convergence of different hyperparameter optimization methods on the  $\ell_2$ -regularized LR problem for the 20news dataset [Lan95] on held-out test data. SHINE with OPA achieves similar performance as SHINE without OPA but with better convergence guarantees. (right) Evaluation of the inversion quality in direction  $v$  using OPA  $b = B_n^{-1}v$  compared to the exact inverse  $a = J_{g_\theta}(z^*)^{-1}v$  for 3 different directions: the prescribed direction, the Krylov direction and a random direction. The points represent the cosine similarity between  $a$  and  $b$  as a function of the ratio of their norm and the closer to  $(1, 1)$  the better. The inverse in the prescribed direction is better than in random direction.

gest that SHINE (in its vanilla version) is able to reduce the time taken for the backward pass almost 10-fold compared to the original method while retaining a competitive performance (on par with Res-Net-18 [He+16] at 92.9%). Finally, we do highlight that the Jacobian-Free method [Fun+21] is able to perform well outside the scope of its theoretical assumptions, albeit with slightly worse performance than SHINE. We conjecture that the batched stochastic gradient descent helps accelerated methods by averaging out the errors made in the approximation.

**ImageNet.** The second dataset is the ImageNet dataset [Den+09] which features 1.2 million images cropped to  $224 \times 224$ , representing 1000 classes. This dataset is recognized as a large-scale computer vision problem and the dimension of the fixed point to find is  $d = 190k$ .

For this challenging task, we noticed that the vanilla version of SHINE was suffering a big drop just after the transition from unrolled pre-training to actual equilibrium training. To remedy partly this problem, we introduced the fallback to Jacobian-Free inversion. The results for a single random seed presented in Figure 7.1-3 for the ImageNet dataset are given for SHINE with fallback. We verified that the fallback is barely used: in 1000 batches of size 32, only 2 samples used fallback, a proportion of  $6.25 \times 10^{-5}$ .

Despite the drop suffered at the beginning of the equilibrium training, SHINE in its refined version is able to perform on par with the Jacobian-Free method [Fun+21]. We also confirm the importance of choosing the right initialization to perform ac-

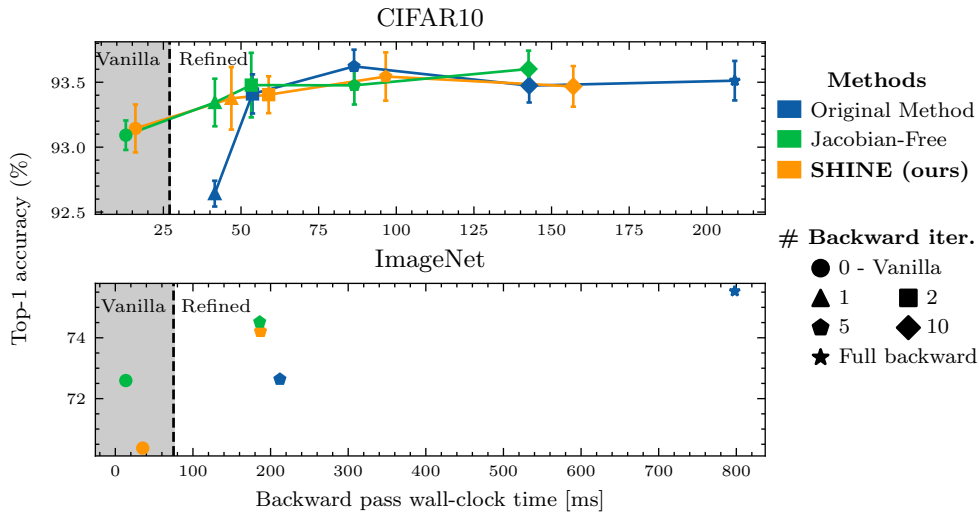


Figure 7.1-3: **DEQ**: Top-1 accuracy function of backward pass runtime for the different methods considered to train **DEQs**, on CIFAR [Kri09] and ImageNet [Den+09]. The original **DEQ** training method corresponds to the Full backward pass points and the vanilla **SHINE** and Jacobian-Free methods correspond to direct use of the inverse approximation without further refinement. The other points correspond to further refinements of the different methods with different number of iterations used to invert  $\mathbf{J}_{g_\theta}(z^*)$  in the direction of  $\nabla_z \mathcal{L}(z^*)$ . This highlights the trade-off between computations and performances driving the refinement choice.

celerated backpropagation, by showing that with a limited iterative inversion, the performance of the original method deteriorates. Finally, while the drop in performance for the accelerated methods is significant when applied in their vanilla version, we remind the reader that no fine-tuning was performed on the training hyperparameters, making those results encouraging (on par with architectures like ResNet-18 [He+16]).

The key take-away from Figure 7.1-3 is that both **SHINE** and Jacobian-Free approximation methods allow us to accelerate the **DEQ**'s backward pass at a relatively low accuracy cost.<sup>6</sup> Moreover, using the proposed refined versions of these methods, the performance drop can be reduced by reducing the acceleration.

#### 7.1.4 . Conclusion and Discussion

We introduced **SHINE**, a method that leverages the **qN** matrices from the forward pass to obtain an approximation of the gradient of the loss function, thereby reducing the time needed to compute this gradient. We showed that this method can be used on a wide range of applications going from bilevel optimization to

<sup>6</sup>More on the overall computational effort can be found in Table A.7

small and large scale computer vision tasks. We found that both [SHINE](#) and the Jacobian-Free method reduce the required amount of time for the backward pass of implicit models, potentially lowering the barriers for training implicit models.

As those methods still suffer from a small performance drop, there is room for further improvement. In particular, a potential experimentation avenue would be to understand how to balance the efforts of the Adjoint Broyden method in order to come closer to guaranteeing the asymptotical correctness of the approximate inversion. On the theoretical side, this may involve the rate of convergence of the approximated gradient. It also seems desirable to develop a version of [Theorem 4](#) in which convergence of  $(z_n)$  to  $z^*$  is not an assumption but rather follows from the assumptions, as achieved in [Theorem 3](#). We have no doubt that the contraction assumption used for the Jacobian-Free method would allow us to prove such a result, but expect that a significantly weaker assumption will suffice.

A first attempt to adapt [DEQs](#) to [MRI](#) reconstruction was made by Gilton et al. [[GOW21](#)]. Although they used a constrained network, applied to a 2D single-coil setting, they showed promising results. Using such networks to tackle the problems encountered when scaling the NC-PDNet to 3D is a definite goal for future research.

## 7.2 . Other paradigms for memory reduction when training neural networks

Although they have been successfully applied to [Natural Language Processing \(NLP\)](#) tasks [[BZK19](#)] and Computer Vision tasks [[BKK20](#)], [DEQs](#) are not the only reduced-training-memory framework. In this section, we describe other frameworks that could be used to achieve the elusive goal of  $\mathcal{O}(1)$ -memory during training.

### 7.2.1 . Gradient checkpointing

The first work to tackle the problem of memory in neural networks training from a modeling perspective introduces the gradient checkpointing scheme [[Che+16](#)]. The idea behind this method is to omit the saving of some activations and recompute them during the backpropagation. The tradeoff is extremely clear, as instead of saving activations during the forward pass, you accept to recompute them during the backward pass. An excellent dynamic visualization of this tradeoff is provided by Salimans et al. [[SB17](#)] on [github.com/cybertronai/gradient-checkpointing](https://github.com/cybertronai/gradient-checkpointing).

### 7.2.2 . Invertible Networks

Gradient checkpointing still relies on storing some activations during the forward pass. There exists of course a possible extreme scheme of gradient checkpointing where you only store the last activation, and always re-perform the forward pass to get the previous activation, called a memory-poor scheme. However, the computational cost is then in  $\mathcal{O}(n)^2$ , which is not acceptable in most deep learning applications.

In order to still be able to store only the last activation, you need to be able to compute the previous one from it. This is exactly what invertible neural networks propose to do, using only layers that are invertible.

We describe 3 types of designs for invertible layers.

**RevNets.** The idea of invertible networks was originally introduced by Gomez et al. [Gom+17], under the name reversible networks (RevNets). The design of their layer is the following: given an input  $(\mathbf{x}_1, \mathbf{x}_2)$ , we have the outputs  $(\mathbf{y}_1, \mathbf{y}_2)$  defined as:

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{x}_1 + f_{\theta_1}(\mathbf{x}_2) \\ \mathbf{y}_2 &= \mathbf{x}_2 + f_{\theta_2}(\mathbf{y}_1).\end{aligned}\tag{7.9}$$

This operation is by design invertible, since we can recover the inputs from the outputs using the following operations:

$$\begin{aligned}\mathbf{x}_2 &= \mathbf{y}_2 - f_{\theta_2}(\mathbf{y}_1) \\ \mathbf{x}_1 &= \mathbf{y}_1 - f_{\theta_1}(\mathbf{x}_2).\end{aligned}\tag{7.10}$$

This way, during the backpropagation you can recompute the activations  $(\mathbf{x}_1, \mathbf{x}_2)$  from  $(\mathbf{y}_1, \mathbf{y}_2)$ , and therefore save the memory cost of storing  $(\mathbf{x}_1, \mathbf{x}_2)$ .

**i-RIM.** **Recurrent Inference Machines (RIM)** [PW17] are a type of unrolled network which, like Adler et al. [AÖ18b], propose to learn a nonlinear combination of previous iterates and the data consistency gradient. The equivalent of the buffer [AÖ18b] is the memory state, which in the case of RIM is decoupled from the current iterates. Putzky et al. [PW19] made each RIM unrolled iteration invertible in order to have an overall invertible network. This allowed them to push the limits of the total number of iterations achievable. However, to this end, they had to compromise some of the nonlinear dynamics of their network. As i-RIM is specifically adapted to inverse problems, we will describe its equations in the context of MRI reconstruction.

$$\begin{aligned}\mathbf{s}'_t &= \mathbf{s}_t + g_{\theta_g}(\mathcal{A}^H(\mathcal{A}f_{\theta_f}(\boldsymbol{\eta}_t) - \mathbf{y})) \\ \boldsymbol{\eta}_{t+1}, \mathbf{s}_{t+1} &= h_{\theta_h}(\boldsymbol{\eta}_t, \mathbf{s}'_t).\end{aligned}\tag{7.11}$$

Assuming  $h_{\theta_h}$  is invertible, we can invert this layer using the following operations:

$$\begin{aligned}\boldsymbol{\eta}_t, \mathbf{s}'_t &= h_{\theta_h}^{-1}(\boldsymbol{\eta}_{t+1}, \mathbf{s}_{t+1}) \\ \mathbf{s}_t &= \mathbf{s}'_t - g_{\theta_g}(\mathcal{A}^H(\mathcal{A}f_{\theta_f}(\boldsymbol{\eta}_t) - \mathbf{y})).\end{aligned}\tag{7.12}$$

**MomentumNets.** Momentum Residual Networks (MomentumNets) [San+21] are designed thanks to a slight modification in the implementation of classical Residual Networks. Its equations are the following:

$$\begin{aligned}\mathbf{v}_{n+1} &= \gamma \mathbf{v}_n + (1 - \gamma) f_{\theta}(\mathbf{x}_n) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{v}_{n+1}.\end{aligned}\tag{7.13}$$

We can then invert the layer in the following way:

$$\begin{aligned} \mathbf{x}_n &= \mathbf{v}_{n+1} - \mathbf{x}_{n+1} \\ \mathbf{v}_n &= \frac{1}{\gamma}(\mathbf{v}_{n+1} - (1 - \gamma)f_{\theta}(\mathbf{x}_n)). \end{aligned} \tag{7.14}$$

One important point is that while this scheme enables in theory to achieve an  $\mathcal{O}(1)$  memory during training, its memory consumption is actually dependent on the depth of the network. This is due to the fact that the multiplication by  $\gamma$  will occur with finite precision, and will therefore be a non-invertible operation. It is therefore important to store the lost bits when doing the forward pass. This information buffer is then a source of memory consumption which can be adjusted with the parameter  $\gamma$ . Sander et al. [San+21] show that the memory is arbitrarily reduced with this parameter.

### Memory-efficient Learning for Large-Scale Computational Imaging.

Kellman et al. [Kel+20] and Wang et al. [Wan+21b] proposed a reversible scheme for the learned proximal gradient descent described in Equation 4.15 (and the ModL network described in Equation 4.12). In order to do so, they used a revertible  $f_{\theta_n}$ , and designed the gradient step inversion with step size  $\alpha_n$  to be the solution of the following fixed-point equation for the variable  $\mathbf{x}$ :

$$\mathbf{x}_{n+1} = \mathbf{x} - \alpha_n \mathcal{A}^{\top}(\mathcal{A}\mathbf{x} - \mathbf{y}) \tag{7.15}$$

#### 7.2.3 . IFT-based networks

While DEQs [BZK19; BKK20] are a very successful application of the IFT to DL, previous works paved the way for the use of implicit layers. The main difference with DEQs is that the implicit layer features an equation that is not a fixed point equation. Amos et al. [AZ17] used an optimization problem implicit solution as a layer in their network, but they did not mention memory gains (probably because this layer was used in an otherwise classical architecture). Chen et al. [Che+18a] chose to use the solution to an Ordinary Differential Equation as the output of their neural network. They do mention memory-efficiency as being one of the strong points of their method.

\* \* \*  
\* \*  
\*



# General Conclusions and Perspectives

## Chapter Outline

|   |     |
|---|-----|
| <a href="#">Contributions &amp; Limitations</a> . . . . . | 163 |
| <a href="#">Perspectives</a> . . . . .                    | 164 |

At the end of these 3 years, we managed to give a significant boost to the MRI reconstruction community focusing on deep learning. Not only did we try to compare different approaches, and provide the tools to do so to practitioners, we have also designed new architectures, questioned their limitations and proposed methodological developments reaching even beyond the scope of MRI reconstruction.

## Contributions & Limitations

We started this thesis by taking a first look at the state of the art in the field of deep learning for MRI reconstruction. After noticing a lack of direct comparisons between approaches, the release of the fastMRI dataset [Zbo+18] was an opportunity for us to build a reproducible benchmark between some of them. The repository where this benchmark was implemented, [github.com/zaccharieramzi/fastmri-reproducible-benchmark](https://github.com/zaccharieramzi/fastmri-reproducible-benchmark), served as a basis for the rest of the thesis, but it gained a lot of attention as can be assessed by the number of GitHub stars, 85 at the time of writing, the second most for fastmri-tagged repositories on GitHub,<sup>7</sup> but also by the public discussions with external people in Issues. The current major limitation of this repository, and of the corresponding benchmark, is that it does not feature a lot of original approaches developed in the literature prior to and after it, whether they concern unrolled networks or other DL approaches, as discussed in chapter 4. The limited computation resources we had at the time of writing (one local GPU with 16 GB of memory) also prevented a more extended review. The easier access to supercomputers such as Jean Zay will help us realize exhaustive benchmarks.

We added to this repository two new architectures: the *XPDNet* and the *NC-PDNet*. The *XPDNet* is the model that allowed us to secure the second spot in the fastMRI 2020 brain reconstruction challenge. It is implemented in a modular way in order to facilitate the embedding of any newly developed denoising architecture.

---

<sup>7</sup>[github.com/topics/fastmri](https://github.com/topics/fastmri)



The *NC-PDNet* was designed to handle non-Cartesian data. Its key ingredient is the density compensation [PM99], which we show to be crucial for the design of networks that aim to reconstruct non-Cartesian data. These networks have yet to be tested extensively before they can be used in practice. Moreover, their implementation on MRI scanners, for example via the Gadgetron framework [HS13], is not an easy task.

We then looked at the questions of the clinical applicability of deep learning for MRI reconstruction. We started by deriving a new wavelet-inspired architecture capable of retaining the robustness properties of wavelets, while enjoying the performance boost of deep learning, the *Learnlets*. Additionally, we proposed to use the newly introduced Denoising for Score Matching approach to create an uncertainty quantification method (or metric) for MRI reconstruction. Finally, we evaluated our proposed *XPNet* against *GRAPPA*, in particular in the prospective setting. One common weakness of the methods proposed to improve the clinical applicability of deep learning for MRI reconstruction is that they obtain worse performance than other unconstrained methods. Therefore, it is unclear whether the most promising avenue is to increase the performance of these methods, or work on more robust prospective tests for existing models.

Finally, when we realized that *DEQs* were a very promising approach to build bigger and more performant networks, we decided to investigate their internal functioning. We identified a severe bottleneck in the backpropagation of these models, and proposed a solution, *SHINE*, which turned out to be even more successful for Hyperparameter Optimization. However, there are still a lot of questions to be answered, especially regarding the success of the Outer-Problem-Awareness patch, but also the expected speed improvement of *SHINE* theoretically.

## Perspectives

Several perspectives emerge from the work done in this PhD thesis.

The first one concerns the design of even deeper networks, using memory-reduction techniques. As we know that the image quality will only get better with deeper networks (since we have not reached the overfitting limit on the fastMRI dataset), the current barrier is the memory size on GPUs. While model parallelism techniques are part of the solution, they are not widely applicable, and in practice lead to long software iteration cycles. Of course, augmenting the memory size on GPUs is a definite option, but this is beyond the scope of our developments. The benefits of the modeling approaches are then 2-fold: they are unbounded and very affordable. Although it is clear how these techniques can be applied, the missing bit is the dataset that will enable their use in a truly meaningful setting: a large 3D multicoil complex-value raw k-space data. At this point, finding ways to synthesize raw data from magnitude volumes might become crucial. Another possibility is to be able to leverage accelerated data, for example in a self-supervised

or semi-supervised way.

The second perspective emerges from the availability of the *NC-PDNet* and the need to push further the efficiency of the overall *MRI* pipeline. Indeed, in this thesis we have not questioned the acquisition that we have considered fixed, but we can indeed optimize it. This requires carefully respecting the *MRI* hardware constraints involved on the gradient system and designing an adapted learning protocol, but promises to deliver new insights into how to best sample the *k*-space. This line of work has already been undertaken by Chaithya et al. [CZC21].

Finally, while the sensitivity maps refinement module introduced by Sriram et al. [Sri+20] is a way to refine our estimation of the forward operator, many more adjustments can be made. These adjustments can be easily integrated in the networks implemented in this thesis. To name one, the *B0* correction can be integrated and further refined end-to-end in the networks. This is at the core of current developments conducted by my peers in the *CS-MRI* team at *NeuroSpin*, notably Guillaume Daval-Fr erot.

\* \* \*  
\* \*  
\*



# Appendices



## A - Additional results

### Qualitative results for NC-PDNet

In [Figure A-1](#), for the spiral acquisition, as was hinted by the quantitative results in [Table 5.1](#) in the main text, we can see that the difference between the *NC-PDNet* and U-net with *DCp* is harder to grasp.

In [Figure A-2](#), we can see that for the spiral acquisition, there is virtually no difference between the U-net and *NC-PDNet*'s results.

### Values of figures for Learnlets

In [Tables A.1](#), [A.2](#), [A.3](#) and [A.4](#) there can be found all the *PSNR* values as a function of  $\sigma$  for each model in [Figures 6.1-3](#), [6.1-4](#), [6.1-5](#) and [6.1-6](#) respectively. Conversely, [Table A.5](#) corresponds to the information presented in [Figure 6.1-8](#).

### DSM zooms

The zoom on this part of the reconstruction shown in [Figure A-5](#), illustrates a failure of the Neural network reconstruction. It indeed produces a sharp artifact that might hinder the examiner's conclusions. The bayesian sampling allows us to see that this region is not necessarily very sharp.

### OPA algorithm for SHINE

**Remark** *A possible choice for  $(t_n)$  is to use an arbitrary  $t_0 > 0$  and  $t_n := \|s_{n-1}\|$  for  $n \geq 1$ .*

### SHINE Bilevel optimization extended

In order to make sure that *SHINE* was indeed improving over *HOAG* [[Ped16](#)], we also looked at the results obtained when performing an inversion with a precision lower than that prescribed by *Pedregosa* [[Ped16](#)] originally (i.e. truncating the iterative inversion). These results, also complemented with *Random Search* [[BB12](#)], can be seen in [Figure A-6](#). They confirm that the advantage provided by *SHINE* cannot be retrieved with a looser tolerance on the inversion.

#### A.0.1 . Regularized Nonlinear Least Squares

In order to further validate the efficiency of *SHINE* compared to competing methods, we also benchmarked it on the regularized nonlinear least squares task.

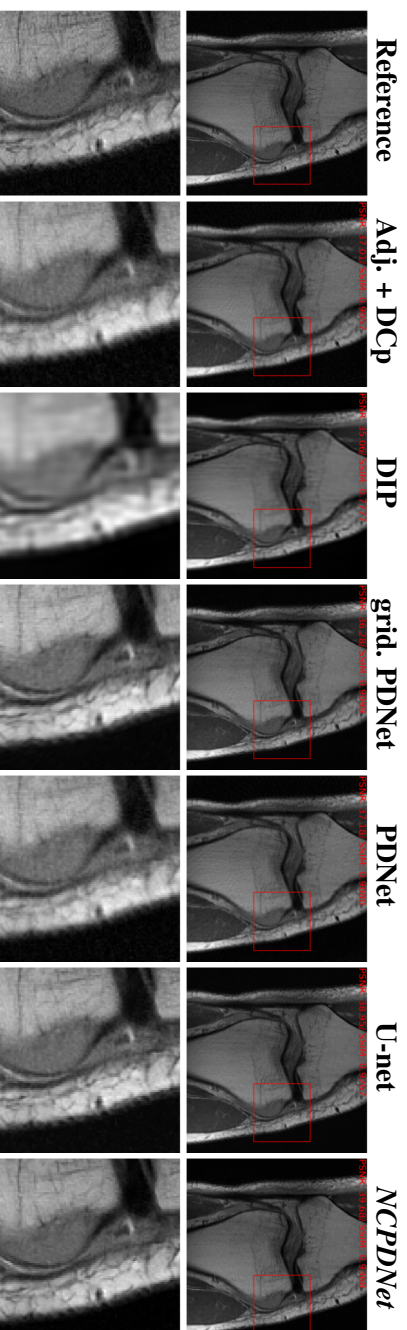


Figure A-1: **2D single-coil spiral acquisition (knee fastMRI dataset, PD contrast)**: Reconstruction results for a specific slice (16th slice of `file001184`, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel.

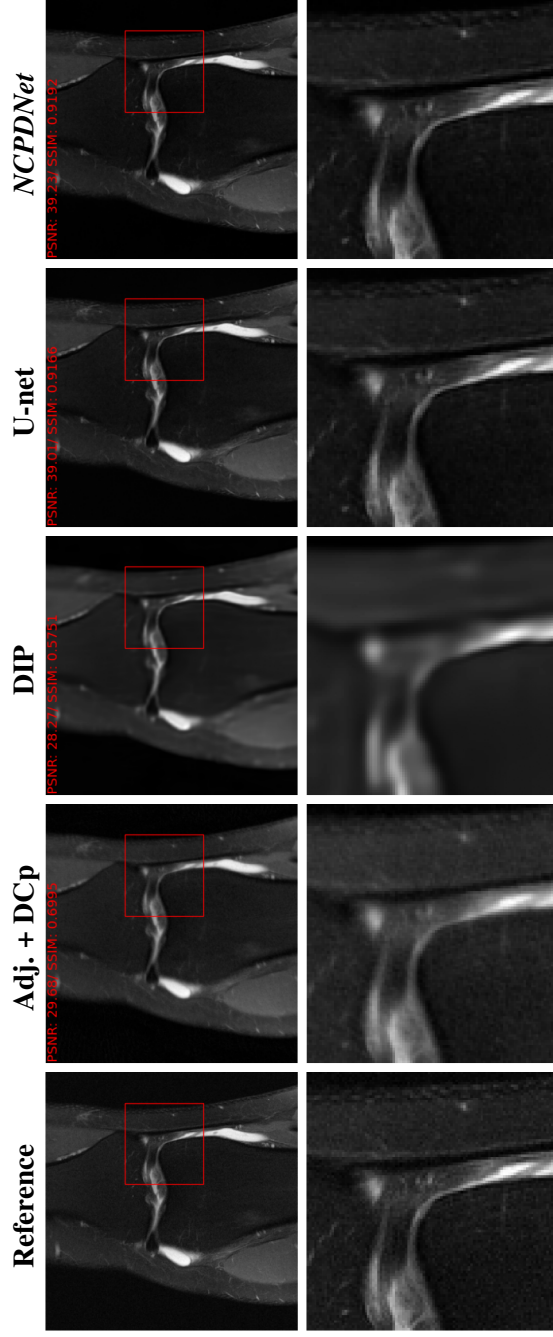


Figure A-2: **2D multicoil spiral acquisition (knee fastMRI dataset, PDFS contrast)**: Reconstruction results for a specific slice (16th slice of file100000, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel.



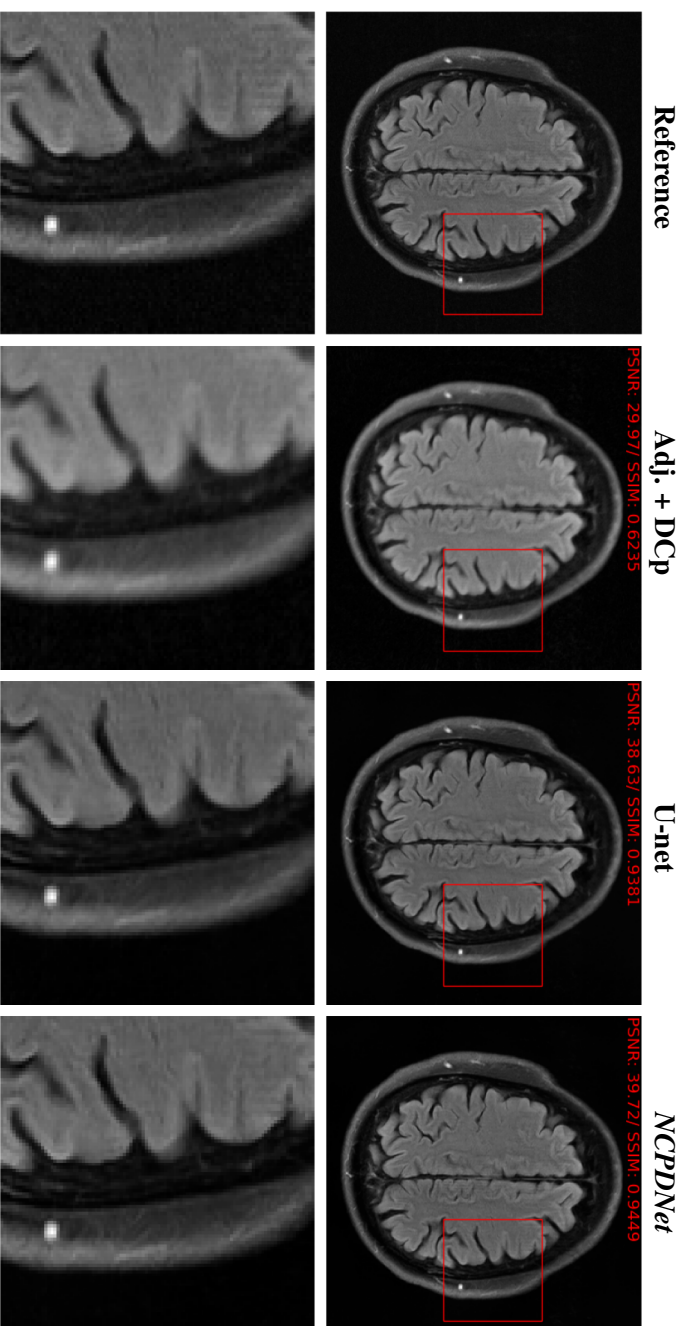


Figure A-3: **2D multicoil spiral acquisition (brain fastMRI dataset, FLAIR contrast)**: Reconstruction results for a specific slice (6th slice of `file_brain_AXFLAIR_200_6002447`) from the brain fastMRI dataset with networks trained on knee data. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel.

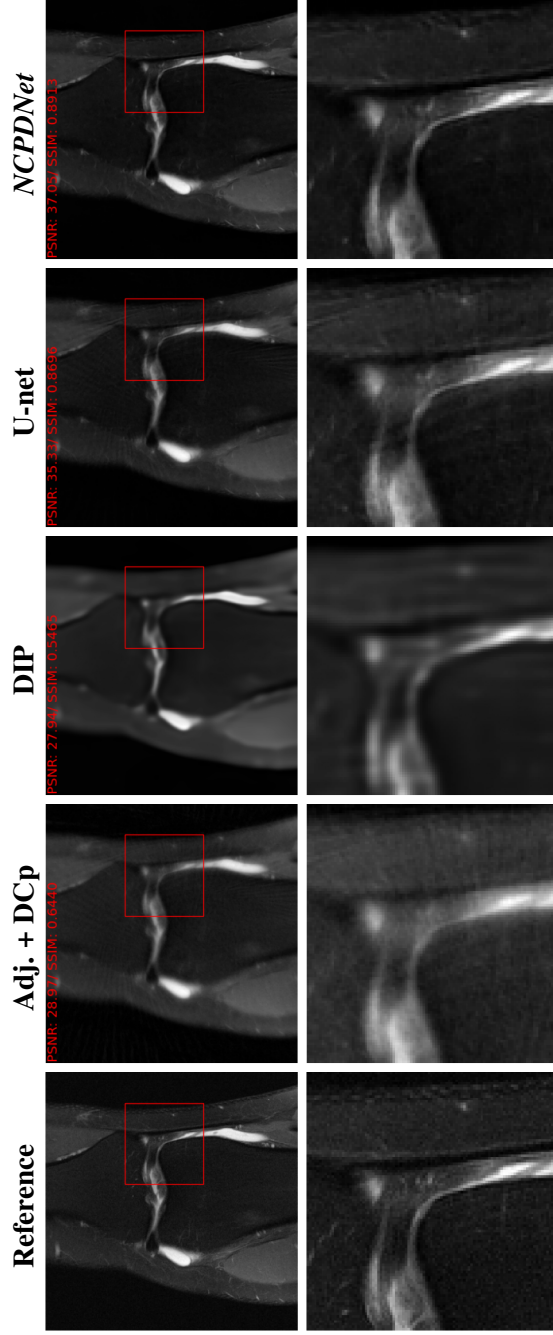


Figure A-4: **2D multicoil spiral acquisition (knee fastMRI dataset, PDFS contrast) AF 8**: Reconstruction results for a specific slice (16th slice of  $\text{fl1e100000}$ , part of the validation set) for an AF 8. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel.

| $\sigma$      | <b>Original</b> | <b>Learnlets</b> | <b>U-net 128</b> | <b>Wavelets</b> |
|---------------|-----------------|------------------|------------------|-----------------|
| <b>0.0001</b> | 128.13          | 124.69           | 53.29            | 127.30          |
| <b>5.0</b>    | 34.15           | 36.51            | 37.76            | 35.76           |
| <b>15.0</b>   | 24.61           | 30.87            | 31.67            | 29.56           |
| <b>20.0</b>   | 22.11           | 29.55            | 30.27            | 28.25           |
| <b>25.0</b>   | 20.17           | 28.54            | 29.24            | 27.32           |
| <b>30.0</b>   | 18.59           | 27.74            | 28.43            | 26.61           |
| <b>50.0</b>   | 14.15           | 25.58            | 26.31            | 24.79           |
| <b>75.0</b>   | 10.63           | 23.90            | 24.71            | 23.46           |
| <b>85.0</b>   | 9.54            | 23.38            | 23.74            | 23.06           |
| <b>95.0</b>   | 8.58            | 22.93            | 22.59            | 22.71           |
| <b>100.0</b>  | 8.13            | 22.71            | 22.14            | 22.56           |

Table A.1: PSNR for different standard deviations of the noise added to the test images for every model in Figure 6.1-3 and the original noisy images.

| $\sigma$      | <b>Original</b> | <b>Learnlets</b> | <b>Learnlets<br/>No exact recon.</b> |
|---------------|-----------------|------------------|--------------------------------------|
| <b>0.0001</b> | 128.13          | 124.69           | 43.56                                |
| <b>5.0</b>    | 34.15           | 36.51            | 36.32                                |
| <b>15.0</b>   | 24.61           | 30.87            | 30.81                                |
| <b>20.0</b>   | 22.11           | 29.55            | 29.46                                |
| <b>25.0</b>   | 20.17           | 28.54            | 28.44                                |
| <b>30.0</b>   | 18.59           | 27.74            | 27.63                                |
| <b>50.0</b>   | 14.15           | 25.58            | 25.45                                |
| <b>75.0</b>   | 10.63           | 23.90            | 23.76                                |
| <b>85.0</b>   | 9.54            | 23.38            | 23.21                                |
| <b>95.0</b>   | 8.58            | 22.93            | 22.71                                |
| <b>100.0</b>  | 8.13            | 22.71            | 22.50                                |

Table A.2: PSNR for different standard deviations of the noise added to the test images for both models in Figure 6.1-4 and the original noisy images.

| $\sigma$      | Original | U-net 4 | U-net 8 | U-net 64 | U-net 128 | Wavelets |
|---------------|----------|---------|---------|----------|-----------|----------|
| <b>0.0001</b> | 128.13   | 39.89   | 43.67   | 52.60    | 53.29     | 127.30   |
| <b>5.0</b>    | 34.15    | 35.58   | 36.65   | 37.73    | 37.76     | 35.76    |
| <b>15.0</b>   | 24.61    | 30.53   | 31.06   | 31.65    | 31.67     | 29.56    |
| <b>20.0</b>   | 22.11    | 29.23   | 29.72   | 30.24    | 30.27     | 28.25    |
| <b>25.0</b>   | 20.17    | 28.26   | 28.73   | 29.21    | 29.24     | 27.32    |
| <b>30.0</b>   | 18.59    | 27.50   | 27.94   | 28.41    | 28.43     | 26.61    |
| <b>50.0</b>   | 14.15    | 25.49   | 25.88   | 26.30    | 26.31     | 24.79    |
| <b>75.0</b>   | 10.63    | 17.86   | 18.74   | 24.55    | 24.71     | 23.46    |
| <b>85.0</b>   | 9.54     | 14.93   | 15.74   | 23.52    | 23.74     | 23.06    |
| <b>95.0</b>   | 8.58     | 13.25   | 13.70   | 22.25    | 22.59     | 22.71    |
| <b>100.0</b>  | 8.13     | 12.60   | 12.94   | 21.65    | 22.14     | 22.56    |

Table A.3: PSNR for different standard deviations of the noise added to the test images for all the models in Figure 6.1-5 and the original noisy images.

| $\sigma$      | Original | U-net 64 | U-net 64<br>Exact recon. |
|---------------|----------|----------|--------------------------|
| <b>0.0001</b> | 128.13   | 52.60    | 127.73                   |
| <b>5.0</b>    | 34.15    | 37.73    | 37.77                    |
| <b>15.0</b>   | 24.61    | 31.65    | 31.63                    |
| <b>20.0</b>   | 22.11    | 30.24    | 30.22                    |
| <b>25.0</b>   | 20.17    | 29.21    | 29.18                    |
| <b>30.0</b>   | 18.59    | 28.41    | 28.38                    |
| <b>50.0</b>   | 14.15    | 26.30    | 26.30                    |
| <b>75.0</b>   | 10.63    | 24.55    | 22.64                    |
| <b>85.0</b>   | 9.54     | 23.52    | 19.61                    |
| <b>95.0</b>   | 8.58     | 22.25    | 16.53                    |
| <b>100.0</b>  | 8.13     | 21.65    | 15.11                    |

Table A.4: PSNR for different standard deviations of the noise added to the test images for both models in Figure 6.1-6 and the original noisy images.

| <b>Samples</b> | <b>U-net 8</b> | <b>U-net 64</b> | <b>Learnlets<br/>No exact recon.</b> |
|----------------|----------------|-----------------|--------------------------------------|
| <b>1</b>       | 28.47          | 28.09           | 28.27                                |
| <b>5</b>       | 28.72          | 29.19           | 28.42                                |
| <b>10</b>      | 28.69          | 29.15           | 28.42                                |
| <b>50</b>      | 28.74          | 29.23           | 28.44                                |
| <b>100</b>     | 28.72          | 29.20           | 28.44                                |
| <b>200</b>     | 28.71          | 29.22           | 28.44                                |
| <b>400</b>     | 28.73          | 29.21           | 28.44                                |

Table A.5: PSNR at  $\sigma = 25$  added to the test images as a function of the number of samples used during training for the three models in Figure 6.1-8.

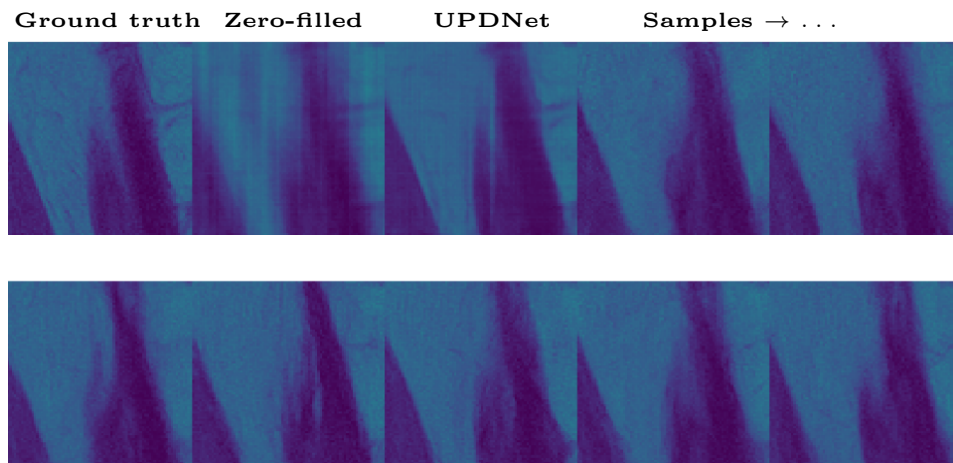


Figure A-5: Zoom of Bayesian posterior sampling for MRI reconstruction. The ordering is the same as in Figure 6.2-10.

---

**Algorithme LBFGS:** (Limited memory) BFGS method with OPA.
 

---

**Input :** initial guess  $(z_0, B_0^{-1})$ , where  $B_0^{-1}$  is symmetric and positive definite, tolerance  $\epsilon > 0$ , frequency of additional updates  $M \in \mathbb{N}$ , memory limit  $L \in \mathbb{N} \cup \{\infty\}$ ,  $(t_n)$  a null sequence of positive numbers with  $\sum_n t_n < \infty$

1 Let  $F := \nabla_z g_\theta$   
 2 **for**  $n = 0, 1, 2, \dots$  **do**  
 3     **if**  $\|F(z_n)\| \leq \epsilon$  **then** let  $z^* := z_n$  and let  $B := B_n$ ; **STOP**

4     Let  $\hat{B}_n^{-1} := B_n^{-1}$   
 5     **if**  $(n \bmod M) = 0$  **let**  $e_n := t_n B_n^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z_n}$ ,  
        $\hat{y}_n := F(z_n + e_n) - F(z_n)$  and  $\hat{r}_n := (e_n)^\top \hat{y}_n$   
 6     **if**  $\hat{r}_n > 0$  **let**  $\hat{a}_n := e_n - B_n^{-1} \hat{y}_n$  and **let**

$$\hat{B}_n^{-1} := B_n^{-1} + \frac{\hat{a}_n (e_n)^\top + e_n (\hat{a}_n)^\top}{\hat{r}_n} - \frac{(\hat{a}_n)^\top \hat{y}_n}{(\hat{r}_n)^2} e_n (e_n)^\top$$

      Let  $B_n^{-1} := \hat{B}_n^{-1}$   
 7     **if**  $n \geq L$  **then** remove update  $n - L$  from  $B_n^{-1}$   
 8     Let  $p_n := -B_n^{-1} F(z_n)$   
 9     Obtain  $\alpha_n$  via line-search and let  $s_n := \alpha_n p_n$   
 10     Let  $z_{n+1} := z_n + s_n$ ,  $y_n := F(z_{n+1}) - F(z_n)$  and  
        $r_n := (s_n)^\top y_n$

11     **if**  $r_n > 0$  **then**  
 12         let  $a_n := s_n - B_n^{-1} y_n$  and **let**

$$B_{n+1}^{-1} := B_n^{-1} + \frac{a_n (s_n)^\top + s_n (a_n)^\top}{r_n} - \frac{(a_n)^\top y_n}{(r_n)^2} s_n (s_n)^\top$$

13     **else** let  $B_{n+1}^{-1} := B_n^{-1}$   
 14     **if**  $n \geq L$  **then** remove update  $n - L$  from  $B_{n+1}^{-1}$

**Output :**  $z^*, B$

---

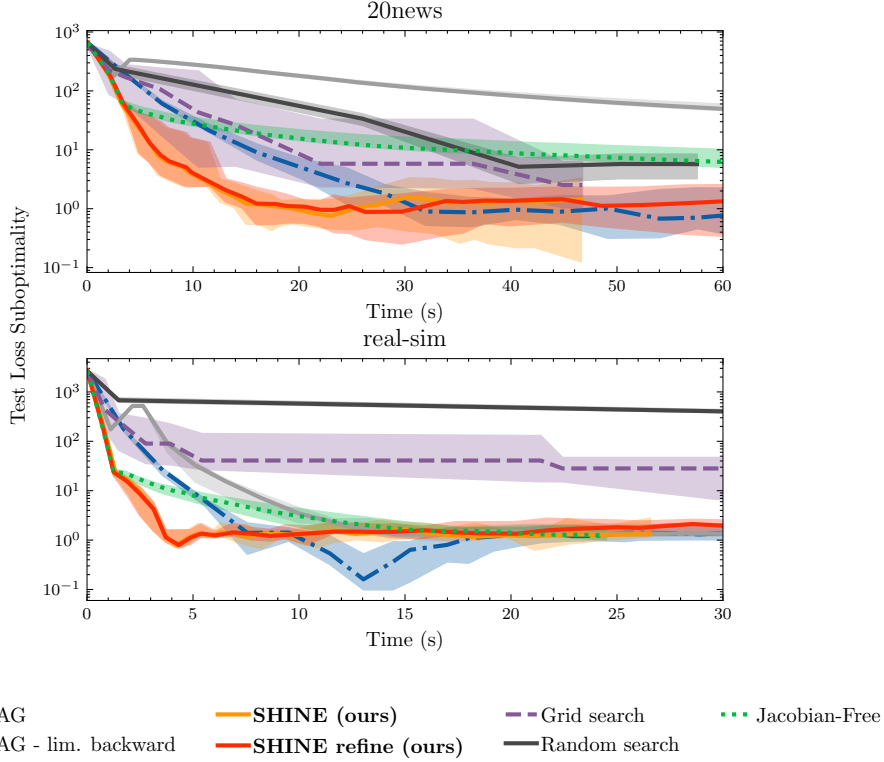


Figure A-6: **Bilevel optimization:** Convergence of different hyperparameter optimization methods on the  $\ell_2$ -regularized LR problem for two datasets (20news [Lan95] and real-sim [Fan11]) on held-out test data.

For a training set  $(\mathbf{x}_{train,i}, y_{train,i})_{i=1}^N$  and a test set  $(\mathbf{x}_{test,i}, y_{test,i})_{i=1}^M$ , this problem reads

$$\begin{aligned} \min_{\theta} \frac{1}{2} \sum_{i=1}^M \|y_{test,i} - \sigma((\mathbf{z}^*)^\top \mathbf{x}_{test,i})\|_2^2 \\ \mathbf{z}^* = \arg \min_{\mathbf{z}} \frac{1}{2} \sum_{j=1}^N \|y_{train,j} - \sigma(\mathbf{z}^\top \mathbf{x}_{train,j})\|_2^2 + \frac{\theta}{2} \|\mathbf{z}\|_2^2 \end{aligned} \quad (\text{A.1})$$

where  $\sigma$  denotes the sigmoid function  $\sigma(x) = \frac{1}{1+e^{-x}}$ . For a fixed hyperparameter  $\theta$ , this task is typically solved using L-BFGS [XRM20; Ber+21].

We can see in Figure A-7 that SHINE clearly outperforms the Jacobian-Free method, and it is also quicker to converge compared to HOAG. We can also notice the benefit of OPA compared to the vanilla SHINE method is more pronounced. We hypothesize that this is due to the nonconvex nature of the inner problem making the Hessian inverse approximation more difficult, as was noted by Berahas et al. [Ber+21].

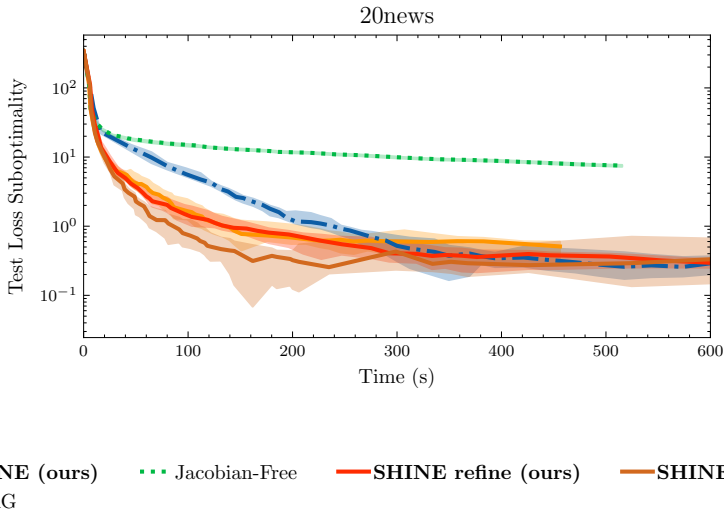


Figure A-7: **Bilevel optimization on regularized nonlinear least squares:** Convergence of different hyperparameter optimization methods on the  $\ell_2$ -regularized nonlinear least squares for the 20news [Lan95] dataset on held-out test data.

Table A.6: Nonlinear spectral radius obtained by the power method for the fixed-point defining subnetwork for the 3 different methods.

| Method        | Nonlinear spectral radius |
|---------------|---------------------------|
| Original      | 230.5                     |
| Jacobian-Free | 193.7                     |
| SHINE         | 234.2                     |

### SHINE contractivity assumption

One of the main limiting assumptions in the original Jacobian-Free method work [Fun+21], is the contractivity assumption. We showed here that it was not important to enforce this in order to achieve excellent results, but one can wonder whether this assumption is not met in practice thanks to the unrolled pretraining of DEQs. We looked at the contractivity of the fixed-point defining subnetwork empirically by using the power-method applied to a nonlinear function, in the CIFAR setting. The results, summarized in Table A.6, show that the fixed-point defining subnetwork is not contractive at all.



Table A.7: The time required for each method on the different datasets during the equilibrium training. For the forward and backward passes, the time is measured offline, for a single batch of 32 samples, with a single GPU, using the median to avoid outliers. This time is given in milliseconds. For the epochs, the time is measured by taking an average of the 6 first epochs, and given in hours-minutes for Imagenet and minutes-seconds for CIFAR. The epoch time for SHINE without improvement on Imagenet is not given because it never reaches the 26 forward steps: the implicit depth is too short. Fallback is not used for CIFAR. Numbers in parentheses indicate the number of inversion steps for the refined versions.

| Dataset Name                    | CIFAR [Kriog] |          |          | ImageNet [Den+09] |          |        |
|---------------------------------|---------------|----------|----------|-------------------|----------|--------|
|                                 | Forward       | Backward | Epoch    | Forward           | Backward | Epoch  |
| Original [BKK20]                | 256           | 210      | 4 min 40 | 644               | 798      | 3 h 38 |
| Jacobian-Free [Fun+21]          | 249           | 12.9     | 3 min 10 | 621               | 13.5     | 2 h 02 |
| SHINE Fallback (ours)           | 218           | 16.0     | 3 min 20 | 622               | 35.3     | 2 h 13 |
| SHINE Fallback refine (5, ours) | 272           | 96.6     | 3 min 50 | 622               | 212      | 2 h 44 |
| Jacobian-Free refine (5)        | 260           | 86.5     | 3 min 40 | 620               | 186      | 2 h 43 |
| Original limited backprop       | 281           | 86.4     | 3 min 50 | 653               | 187      | 2 h 40 |

### SHINE Time gains

Because the total training time is not only driven by backward pass but also by the forward pass and the evaluation, we show for completeness in Table A.7 the time gains for the different acceleration methods for the overall epoch. We do not report in this table the time taken for pretraining which is equivalent across all methods, and is not something on which SHINE has an impact. It is clear in Table A.7 that accelerated methods can have a significant impact on the training of DEQs because we see that half the time of the total pass is spent on the backward pass (more on ImageNet [Den+09]). We also notice that while SHINE has a slightly slower backward pass than the Jacobian-Free method [Fun+21], the difference is negligible when compared to the total pass computational cost.

### DEQ OPA results

We can clearly see in Figure A-8 that in the case of DEQs, OPA also significantly improves the inversion over the other accelerated methods. We also see that the improvements of SHINE over the Jacobian-Free method without OPA are marginal.

Because the inversion is so good, we would expect that the performance of SHINE with OPA would be on par with the original method's. However, this is not what we see in the results presented in Table A.8. Indeed, OPA does improve on SHINE with only Adjoint Broyden, but it does not outperform SHINE done with Broyden.

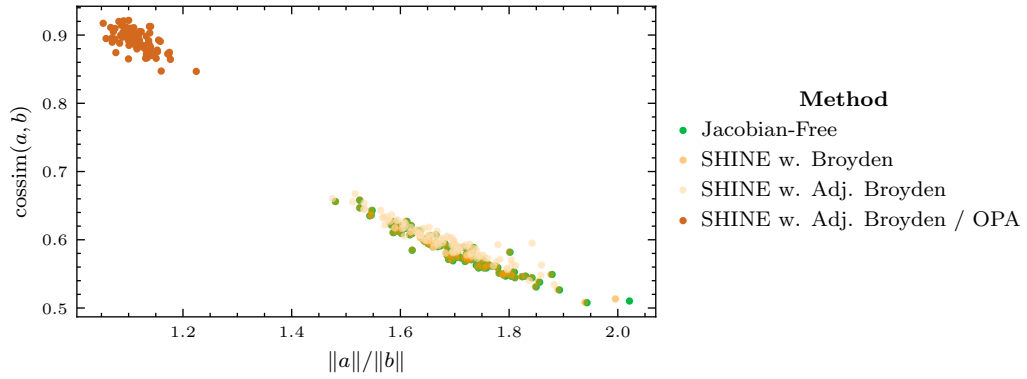


Figure A-8: **Quality of the inversion using OPA in DEQs**: Ratio of the inverse approximation over the exact inverse function of the cosine similarity between the inverse approximation  $\mathbf{b} = \nabla_z \mathcal{L}(z^*) \mathbf{B}_n^{-1}$  and the exact inverse  $\mathbf{a} = \nabla_z \mathcal{L}(z^*) \mathbf{J}_{g_\theta}(z^*)^{-1}$  for different methods. For OPA, the extra update frequency is 5. 100 runs were performed with different batches.

Table A.8: **CIFAR DEQ OPA results**: Top-1 accuracy of different methods on the CIFAR dataset, and epoch mean time.

| Method name                     | Top-1 Accuracy (%) | Epoch mean time |
|---------------------------------|--------------------|-----------------|
| <b>Original</b>                 | 93.51              | 4 min 40        |
| <b>Jacobian-Free</b>            | 93.09              | 3 min 10        |
| <b>SHINE (Broyden)</b>          | 93.14              | 3 min 20        |
| <b>SHINE (Adj. Broyden)</b>     | 92.89              | 4 min           |
| <b>SHINE (Adj. Broyden/OPA)</b> | 93.04              | 4 min 40        |



## B - Training details

### NC-PDNet training details

The training is done with a compound  $\ell_1$  - MSSIM [WSB03] loss as advised by the winners of the 2019 fastMRI challenge [Pez+20]. We used the Rectified Adam optimizer [Liu+20] with default parameters from the TensorFlow implementation<sup>1</sup> and a learning rate of  $10^{-4}$ . We used a batch size of 1, in order for the models to fit on a single V100 GPU with 32 GB of memory. For the 2D settings, we used 100 epochs for training (97.3k gradient steps), where an epoch is defined as seeing one slice of each volume in the dataset. For the 3D setting we used 8 epochs for training ( $\sim 26$ k gradient steps). For the fastMRI data, the k-space is scaled up by a factor of  $10^6$  as per [ZCS20b]. We scaled the OASIS data by a factor of  $10^{-2}$ . The training times for the different networks in the different settings can be found in Table B.1.

### DSM training

#### Network architecture

The network is a 3-scale U-net with residual blocks composed of 3 convolutions followed by a batch normalisation. Each batch normalisation is followed by a ReLU non-linearity except the last one. A projection is used for the input of residual blocks whose number of input channels is not the same as that of output channels. Each scale uses 2 residual blocks for the downsampling path and 2 residual blocks for the upsampling path. Downsampling is performed via average pooling and upsampling is performed via up-convolution (as designed by Ronneberger et al. [RFB15]), in order to avoid checkerboard artefacts. We use the following sequence of number of channels: [32, 64, 128]. In order to deal with MRI images, that are complex in nature, we concatenate the real and imaginary part of the image at the input of the network, forming effectively a 2-channel  $320 \times 320$  image. For input noise level conditioning, we concatenate a noise standard deviation map to the input and in the lower scale of the network. Following the recommendation of Song et al.

<sup>1</sup>[tensorflow.org/addons/api\\_docs/python/tfa/optimizers/RectifiedAdam](https://tensorflow.org/addons/api_docs/python/tfa/optimizers/RectifiedAdam)

Table B.1: Training times in hours (h) for the different networks in the different settings.

| Model                  | Single-coil 2D | Multicoil 2D | Single-coil 3D |
|------------------------|----------------|--------------|----------------|
| U-net on Adjoint + DCp | 8 h            | 20 h         | 22 h           |
| NC-PDNet               | 24 h           | 34 h         | 196 h          |

[SE20], we also divide the output of the network by the absolute value of the noise power. Finally, we use Spectral Norm regularisation [YM17] in order to make sure the score does not take inconsistent values in low-density regions. The Spectral Norm regularisation indeed forces the spectral norm (maximum eigenvalue) of each layer to a certain value which in turn lowers the Lipschitz constant of the network, preventing it to blow in unseen regions. The influence of the Spectral Norm has been studied in the case of P&P approaches by Ryu et al. [Ryu+19].

### Network training

We use the Adam optimizer for network training, with a learning rate of  $10^{-4}$ . We add a white Gaussian noise of variance  $\sigma_s^2$  to the images scaled by a factor of  $10^6$ .  $\sigma_s$  is drawn on-the-fly from a Gaussian distribution of variance  $s = 50$ . This means that at training time, the standard deviation of the noise can be negative, following the recommendation of Lim et al. [Lim+20] to go from extrapolation to interpolation.

### HMC procedure

The HMC procedure starts from a zero-filled reconstruction of the image with added white Gaussian noise of variance  $\sigma_{init} = 100$ . For the reduction of noise standard deviation at reduction step, we use a factor  $\gamma = 0.995$ , following the recommendation of Song et al. [SE20]. We take a step size  $\alpha$  dependent of the sampling temperature at step  $i$ ,  $\alpha = \epsilon \left(\frac{\sigma_i}{\sigma_0}\right)^{1.5}$  with  $\epsilon = 10$ .

## SHINE experiments details

### Logistic Regression Hyperparameters

For both datasets we split the data randomly (with a different seed for each run) between training-validation-test, with the following proportions: 90%-5%-5%. The hyperparameters are the same as in the original HOAG work [Ped16], except:

- We use a memory limitation of 30 updates (not grid-searched) for accelerated methods (Jacobian-Free and SHINE), compared to 10 for the original method. This is because the approximation should be better using more updates. We verified that using 30 updates for the original method does not improve the convergence speed. That number is 60 for OPA.
- We use a smaller exponential decrease of 0.78 (not grid-searched) for the accelerated methods, compared to 0.99 for the original method. This is because in the very long run, the approximation can cause oscillations.

We also use the same setting as Pedregosa [Ped16] for the Grid and Random Search. Finally, we highlight that warm restart is used for both the inner problem and the Hessian inversion in the direction of the gradient.

**OPA inversion experiments.** For the [OPA](#) experiments, we used a memory limitation of 60, and a tolerance of  $10^{-6}$ . The [OPA](#) update is done every 5 regular updates.

#### DEQ training details

The training details are the same as the original [MDEQ](#) paper [[BKK20](#)]: all the hyperparameters are kept the same and not fine-tuned, and the data split is the same. We recall here some important aspects. For both datasets, the network is first trained in an unrolled weight-tied fashion for a few epochs in order to stabilize the training.

We also underline that the [DEQ](#) models, in addition to having a fixed-point-defining subnetwork, also have a classification and a projection head.

Finally, for [Figure 7.1-3](#), the median backward pass is computed with 100 samples on a single V100 [GPU](#) for a batch size of 32.

**CIFAR.** Adam optimizer [[KB15](#)] is used with a  $10^{-3}$  start learning rate, and a cosine annealing schedule.

**ImageNet.** The Stochastic Gradient Descent optimizer is used with a  $5 \times 10^{-2}$  start learning rate, and a cosine annealing schedule.

The images are downsampled 2 times before being fed to the fixed-point defining subnetwork.



## C - Proofs for SHINE

To facilitate reading, we restate the results before proving them.

### Convergence using ULI

**Theorem 2** (Convergence of SHINE to the Hypergradient using ULI) *Let us denote  $\mathbf{p}_\theta^{(n)}$ , the SHINE direction for iterate  $n$  in Algorithm 2 with  $b = \text{true}$ . Under Assumptions 7.1.1 and 7.1.2, for a given parameter  $\theta$ ,  $(z_n)$  converges  $q$ -superlinearly to  $z^*$  and*

$$\lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \left. \frac{\partial \mathcal{L}}{\partial \theta} \right|_{z^*}$$

*Proof.* Under Assumptions 7.1.1 and 7.1.2, More et al. [MT76, Theorem 5.7] shows that  $\mathbf{B}_n$  satisfies

$$\lim_{n \rightarrow \infty} \mathbf{B}_n = \mathbf{J}_{g_\theta}(z^*)$$

The inversion operator is continuous in the space of invertible matrices, so we have:

$$\lim_{n \rightarrow \infty} \mathbf{B}_n^{-1} = \mathbf{J}_{g_\theta}(z^*)^{-1}$$

Because  $\nabla_z \mathcal{L}$  and  $\frac{\partial g_\theta}{\partial \theta}$  are continuous at  $z^*$  by Assumption 7.1.2 (iii), we also have thanks to Assumption 7.1.2 (i):

$$\lim_{n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) = \nabla_z \mathcal{L}(z^*) \quad \text{and} \quad \lim_{n \rightarrow \infty} \left. \frac{\partial g_\theta}{\partial \theta} \right|_{z_n} = \left. \frac{\partial g_\theta}{\partial \theta} \right|_{z^*}$$

By continuity, we then deduce that, as claimed,

$$\lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \lim_{n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1} \frac{\partial g_\theta}{\partial \theta}(z_n) = \nabla_z \mathcal{L}(z^*) \mathbf{J}_{g_\theta}(z^*)^{-1} \left. \frac{\partial g_\theta}{\partial \theta} \right|_{z^*} = \left. \frac{\partial \mathcal{L}}{\partial \theta} \right|_{z^*} \quad \square$$

### Convergence for BFGS with OPA

**Assumption C.o.1** (Extended Assumptions for BFGS). *Let  $g_\theta(z) = \nabla_z r_\theta(z)$  for some  $C^2$  function  $r_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$ . Consider Algorithm 2 with  $b = \text{false}$  and suppose that*

1. *the set  $\Omega := \{z \in \mathbb{R}^d : r_\theta(z) \leq r_\theta(z_0)\}$  is convex;*
2.  *$r_\theta$  is strongly convex in an open superset of  $\Omega$  (this implies that  $r_\theta$  has a unique global minimizer  $z^*$ ) and has a Lipschitz continuous Hessian near  $z^*$ ;*



3. there are positive constants  $\eta_1, \eta_2$  such that the line search used in the algorithm ensures that for each  $n \geq 0$  either

$$r_\theta(\mathbf{z}_{n+1}) \leq r_\theta(\mathbf{z}_n) - \eta_1 \left[ \frac{\nabla r_\theta(\mathbf{z}_n)^\top \mathbf{p}_n}{\|\mathbf{p}_n\|} \right]^2 \quad \text{or} \quad r_\theta(\mathbf{z}_{n+1}) \leq r_\theta(\mathbf{z}_n) + \eta_2 \nabla r_\theta(\mathbf{z}_n)^\top \mathbf{p}_n$$

is satisfied;

4. the line search has the property that  $\alpha_n = 1$  will be used if both

$$\frac{\|(\mathbf{B}_n - \mathbf{J}_{g_\theta}(\mathbf{z}_n))\mathbf{s}_n\|}{\|\mathbf{s}_n\|} \quad \text{and} \quad \|\mathbf{z}_n - \mathbf{z}^*\|$$

are sufficiently small.

**Remark** The requirements 3. and 4. on the line search are, for instance, satisfied under the well-known Wolfe conditions, see Byrd et al. [BSS88, section 3] for further comments.

**Theorem 3** (Convergence of SHINE to the Hypergradient for BFGS with OPA) Let us consider  $\mathbf{p}_\theta^{(n)}$ , the SHINE direction for iterate  $n$  in Algorithm 2 that is enriched by extra updates in the direction  $\mathbf{e}_n$  defined in Equation 7.5. Under Assumptions 7.1.2 (ii-iii) and 7.1.3, for a given parameter  $\theta$ , we have the following: Algorithm 2, for any symmetric and positive definite matrix  $\mathbf{B}_0$ , generates a sequence  $(\mathbf{z}_n)$  that converges  $q$ -superlinearly to  $\mathbf{z}^*$ , and there holds

$$\lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{\mathbf{z}^*} \quad (7.6)$$

*Proof.* The proof is divided into four steps. The first step is to establish the  $q$ -superlinear convergence of  $(\mathbf{z}_n)$  to  $\mathbf{z}^*$ . Denoting by  $N_e \subset \{0, M, 2M, \dots\}$  the set of indices of extra updates that are actually applied, the second step consists of showing

$$\lim_{N_e \ni n \rightarrow \infty} (\mathbf{B}_n - \mathbf{J}_{g_\theta}(\mathbf{z}^*)) \frac{\mathbf{e}_n}{\|\mathbf{e}_n\|} = 0, \quad (C.1)$$

where, in this proof,  $\mathbf{B}_n$  always represents the matrix from Algorithm LBFBS before the update in the direction  $\mathbf{e}_n$  is applied, i.e., the matrix whose inverse appears in the definition of  $\mathbf{e}_n$ , while  $\hat{\mathbf{B}}_n$  always represents the matrix from Algorithm LBFBS after the update in the direction  $\mathbf{e}_n$  has been applied; if the update in the direction  $\mathbf{e}_n$  is not applied, then  $\mathbf{B}_n = \hat{\mathbf{B}}_n$ . The third step is to prove that Equation C.1 implies the desired convergence Equation 7.6 of the SHINE direction if the limit  $n \rightarrow \infty$  is replaced by  $N_e \ni n \rightarrow \infty$ , i.e., the limit is taken on the subsequence corresponding to  $N_e$ . The fourth step is then to transfer the convergence to the entire sequence.

It is easy to check that instead of updating  $\mathbf{B}_n^{-1}$ , respectively,  $\hat{\mathbf{B}}_n^{-1}$ , we can also obtain the sequences  $(\mathbf{B}_n)$  and  $(\hat{\mathbf{B}}_n)$  by updating according to

$$\mathbf{B}_{n+1} = \mathbf{B}_n + \frac{\mathbf{y}_n \mathbf{y}_n^\top}{\mathbf{y}_n^\top \mathbf{s}_n} - \frac{\mathbf{B}_n \mathbf{s}_n (\mathbf{B}_n \mathbf{s}_n)^\top}{\mathbf{s}_n^\top \mathbf{B}_n \mathbf{s}_n}$$

for the usual update (skipping the update if  $\mathbf{y}_n^\top \mathbf{s}_n \leq 0$ ), respectively,

$$\hat{\mathbf{B}}_n = \mathbf{B}_n + \frac{\hat{\mathbf{y}}_n \hat{\mathbf{y}}_n^\top}{\hat{\mathbf{y}}_n^\top \mathbf{e}_n} - \frac{\mathbf{B}_n \mathbf{e}_n (\mathbf{B}_n \mathbf{e}_n)^\top}{\mathbf{e}_n^\top \mathbf{B}_n \mathbf{e}_n}$$

for the extra update (skipping the update if  $\hat{\mathbf{y}}_n^\top \mathbf{e}_n \leq 0$ ). Here, the quantities  $\mathbf{y}_n$ ,  $\hat{\mathbf{y}}_n$  and  $\mathbf{e}_n$  are defined as in [Algorithm LBFGS](#). We can now argue essentially as in the proof of Byrd et al. [[BSS88](#), Theorem 3.1] to show that  $(z_n)$  converges q-superlinearly to  $z^*$ . As part of that proof we obtain that  $\hat{\mathbf{B}}_n \neq \mathbf{B}_n$  for at least  $\lceil 0.5Q \rceil$  of the indices  $n = 0, M, 2M, \dots, QM$  for any  $Q \in \mathbb{N}$  (namely for all  $n \in N_e$  satisfying  $n \leq QM$ ) and that we can apply Byrd et al. [[BN89](#), Theorem 3.2], which yields

$$\lim_{n \rightarrow \infty} (\hat{\mathbf{B}}_n - \mathbf{J}_{g_\theta}(z^*)) \frac{\mathbf{s}_n}{\|\mathbf{s}_n\|} = 0 \quad \text{and} \quad \lim_{N_e \ni n \rightarrow \infty} (\mathbf{B}_n - \mathbf{J}_{g_\theta}(z^*)) \frac{\mathbf{e}_n}{\|\mathbf{e}_n\|} = 0. \quad (\text{C.2})$$

For the third step, we abbreviate  $\mathbf{v}_n := \frac{\partial g_\theta}{\partial \theta} \Big|_{z_n}$ . From the definition of  $\mathbf{e}_n$  and [Equation C.2](#) we infer that

$$0 = \lim_{N_e \ni n \rightarrow \infty} (\mathbf{B}_n - \mathbf{J}_{g_\theta}(z^*)) \frac{\mathbf{e}_n}{\|\mathbf{e}_n\|} = \lim_{N_e \ni n \rightarrow \infty} (\mathbf{I} - \mathbf{J}_{g_\theta}(z^*) \mathbf{B}_n^{-1}) \frac{\mathbf{v}_n}{\|\mathbf{B}_n^{-1} \mathbf{v}_n\|}.$$

After multiplication with  $\mathbf{J}_{g_\theta}(z^*)^{-1}$  this entails

$$\lim_{N_e \ni n \rightarrow \infty} (\mathbf{J}_{g_\theta}(z^*)^{-1} - \mathbf{B}_n^{-1}) \frac{\mathbf{v}_n}{\|\mathbf{B}_n^{-1} \mathbf{v}_n\|} = 0,$$

which shows that

$$\lim_{N_e \ni n \rightarrow \infty} \mathbf{B}_n^{-1} \mathbf{v}_n = \lim_{N_e \ni n \rightarrow \infty} \mathbf{J}_{g_\theta}(z^*)^{-1} \mathbf{v}_n = \mathbf{J}_{g_\theta}(z^*)^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z^*}$$

by [Assumption 7.1.2](#) (iii). Using [Assumption 7.1.2](#) (iii) again it follows that

$$\lim_{N_e \ni n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \lim_{N_e \ni n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z_n} = \nabla_z \mathcal{L}(z^*) \mathbf{J}_{g_\theta}(z^*)^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z^*} = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{z^*},$$

concluding the third step. To infer that [Equation 7.6](#) holds, it suffices to show that  $\lim_{N_e \ni n \rightarrow \infty} \|\mathbf{B}_n - \mathbf{B}_{j_n}\| = 0$  for any sequence  $(j_n)_{n \in N_e} \subset \mathbb{N}$  such that  $\{j_n, j_n + 1, \dots, n - 1\} \cap N_e = \emptyset$  for all  $n \in N_e$  sufficiently large. Indeed, since for  $C := \max\{\sup_n \|\mathbf{B}_n\|, \sup_n \|\mathbf{B}_n^{-1}\|\}$ , which is finite by Byrd et al. [[BN89](#), Theorem 3.2], there holds

$$(\mathbf{B}_n) \subset \left\{ \mathbf{A} \in \mathbb{R}^{d \times d} : \mathbf{A}^{-1} \text{ exists, } \|\mathbf{A}\| \leq C, \|\mathbf{A}^{-1}\| \leq C \right\}$$

and the set on the right-hand side of the inclusion is compact by the Banach lemma, inversion is a *uniformly* continuous operation on this set, hence  $\lim_{N_e \ni n \rightarrow \infty} \|\mathbf{B}_n^{-1} - \mathbf{B}_{j_n}^{-1}\| = 0$ , so

$$\lim_{N_e \ni n \rightarrow \infty} \|\mathbf{p}_\theta^{(n)} - \mathbf{p}_\theta^{(j_n)}\| = 0$$

by continuity, and therefore

$$\lim_{N_e \ni n \rightarrow \infty} \mathbf{p}_\theta^{(j_n)} = \lim_{N_e \ni n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{\mathbf{z}^*}$$

by the third step, establishing the claim.

It remains to show the validity of  $\lim_{N_e \ni n \rightarrow \infty} \|\mathbf{B}_n - \mathbf{B}_{j_n}\| = 0$  for any sequence  $(j_n)_{n \in N_e}$  such that  $\{j_n, j_n + 1, \dots, n - 1\} \cap N_e = \emptyset$  for all  $n \in N_e$  sufficiently large. Since at least every second extra update is actually carried out, the condition on the intersection implies  $n - j_n \leq 2M - 1$  for all these  $n$ . Now let  $(j_n)_{n \in N_e}$  be any such sequence. Then  $\mathbf{B}_n - \mathbf{B}_{j_n} = \sum_{m=j_n}^{n-1} \mathbf{B}_{m+1} - \mathbf{B}_m$  is a sum of at most  $2M - 1$  BFGS updates in search directions, but contains no extra updates. Hence, the secant conditions  $\mathbf{B}_{n-l} \mathbf{s}_{n-1-l} = \mathbf{y}_{n-1-l}$ ,  $l \in \{0, 1, \dots, n - j_n\}$ , are satisfied, allowing us to deduce

$$\begin{aligned} \|\mathbf{B}_{n-l} - \mathbf{B}_{n-l-1}\| &= \frac{\|(\mathbf{B}_{n-l} - \mathbf{B}_{n-l-1}) \mathbf{s}_{n-l-1}\|}{\|\mathbf{s}_{n-l-1}\|} \\ &\leq \frac{\|\mathbf{y}_{n-l-1} - \mathbf{J}_{g_\theta}(\mathbf{z}^*) \mathbf{s}_{n-l-1}\|}{\|\mathbf{s}_{n-l-1}\|} + \frac{\|(\mathbf{B}_{n-l-1} - \mathbf{J}_{g_\theta}(\mathbf{z}^*)) \mathbf{s}_{n-l-1}\|}{\|\mathbf{s}_{n-l-1}\|} \end{aligned}$$

for all  $l \in \{0, 1, \dots, n - j_n - 1\}$ . For each of these  $l$ , both terms on the right-hand side tend to zero for  $N_e \ni n \rightarrow \infty$  (for the second term this follows from the first identity in Equation C.2 due to  $\mathbf{B}_{n-l-1} = \hat{\mathbf{B}}_{n-l-1}$ ). Recalling that  $\mathbf{B}_n - \mathbf{B}_{j_n} = \sum_{m=j_n}^{n-1} \mathbf{B}_{m+1} - \mathbf{B}_m$  we find  $\lim_{N_e \ni n \rightarrow \infty} \|\mathbf{B}_n - \mathbf{B}_{j_n}\| = 0$ , which finishes the fourth step and thus concludes the proof.  $\square$

## Convergence for Adjoint Broyden with OPA

**Theorem 4** (Convergence of SHINE to the Hypergradient for Adjoint Broyden with OPA) *Let us consider  $\mathbf{p}_\theta^{(n)}$ , the SHINE direction for iterate  $n$  in Algorithm 2 with the Adjoint Broyden secant condition Equation 7.7 and extra update in the direction  $\mathbf{v}_n$  defined in Equation 7.8. Under Assumptions 7.1.2 and 7.1.4, for a given parameter  $\theta$ , we have  $q$ -superlinear convergence of  $(z_n)$  to  $\mathbf{z}^*$  and*

$$\lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{\mathbf{z}^*}$$

*Proof.* Due to Assumption 7.1.2, the superlinear convergence of  $(z_n)$  follows from Schlenkrich et al. [SGW10, Theorem 2]. The proof of the remaining claim is divided into two cases.

Case 1: Suppose that  $\nabla_z \mathcal{L}(z^*) = 0$ . By continuity this implies  $\lim_{n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) = 0$ . Since the sequence  $(\mathbf{B}_n^{-1} \frac{\partial g_\theta}{\partial \theta} |_{z_n})$  is bounded by [Assumption 7.1.4](#), it follows that

$$\lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \lim_{n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z_n} = 0 = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{z^*},$$

as claimed.

Case 2: Suppose that  $\nabla_z \mathcal{L}(z^*) \neq 0$ . By continuity this implies  $\nabla_z \mathcal{L}(z_n) \neq 0$  for all sufficiently large  $n \in \mathbb{N}$ . Let us denote by  $N_e \subset \mathbb{N}$  the set of indices of extra updates. We stress that this set is infinite since, by construction, every  $M$ -th update is an extra update. We have  $\mathbf{v}_n \neq 0$  for all sufficiently large  $n \in N_e$ , hence Schlenkrich et al. [[SGW10](#), Lemma 3] yields

$$\lim_{N_e \ni n \rightarrow \infty} \frac{\|\nabla_z \mathcal{L}(z_n)(\mathbf{I} - \mathbf{B}_n^{-1} \mathbf{J}_{g_\theta}(z^*))\|}{\|(\nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1})^\top\|} = \lim_{N_e \ni n \rightarrow \infty} \frac{\|(\mathbf{v}_n)^\top (\mathbf{B}_n - \mathbf{J}_{g_\theta}(z^*))\|}{\|\mathbf{v}_n\|} = 0.$$

This implies

$$\lim_{N_e \ni n \rightarrow \infty} \frac{\|\nabla_z \mathcal{L}(z_n)(\mathbf{J}_{g_\theta}(z^*)^{-1} - \mathbf{B}_n^{-1})\|}{\|\nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1}\|} = 0,$$

thus necessarily

$$\lim_{N_e \ni n \rightarrow \infty} \|\nabla_z \mathcal{L}(z_n)(\mathbf{J}_{g_\theta}(z^*)^{-1} - \mathbf{B}_n^{-1})\| = 0.$$

Since  $\lim_{N_e \ni n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) \mathbf{J}_{g_\theta}(z^*)^{-1} = \nabla_z \mathcal{L}(z^*) \mathbf{J}_{g_\theta}(z^*)^{-1}$  by continuity, we find

$$\lim_{N_e \ni n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1} = \nabla_z \mathcal{L}(z^*) \mathbf{J}_{g_\theta}(z^*)^{-1},$$

whence

$$\lim_{N_e \ni n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \lim_{N_e \ni n \rightarrow \infty} \nabla_z \mathcal{L}(z_n) \mathbf{B}_n^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z_n} = \nabla_z \mathcal{L}(z^*) \mathbf{J}_{g_\theta}(z^*)^{-1} \frac{\partial g_\theta}{\partial \theta} \Big|_{z^*} = \frac{\partial \mathcal{L}}{\partial \theta} \Big|_{z^*}, \quad (\text{C.3})$$

where we have used continuity again. To prove that these limits hold not only for  $N_e \ni n \rightarrow \infty$  but in fact for all  $\mathbb{N} \ni n \rightarrow \infty$ , we establish, as intermediate claim, that for any fixed  $m \in \mathbb{N}$  we have  $\lim_{n \rightarrow \infty} \|\mathbf{B}_{n+m} - \mathbf{B}_n\| = 0$ . Note that this claim is equivalent to  $\lim_{n \rightarrow \infty} \|\mathbf{B}_{n+1} - \mathbf{B}_n\| = 0$ . Denoting by  $L \geq 0$  the Lipschitz constant of  $\mathbf{J}_{g_\theta}$  near  $z^*$ , we find

$$\begin{aligned} \|\mathbf{B}_{n+1} - \mathbf{B}_n\| &= \frac{\|\mathbf{v}_n \mathbf{v}_n^\top [\mathbf{J}_{g_\theta}(z_{n+1}) - \mathbf{B}_n]\|}{\|\mathbf{v}_n\|^2} \leq \|\mathbf{J}_{g_\theta}(z_{n+1}) - \mathbf{J}_{g_\theta}(z^*)\| + \frac{\|[\mathbf{J}_{g_\theta}(z^*) - \mathbf{B}_n]^\top \mathbf{v}_n\|}{\|\mathbf{v}_n\|} \\ &\leq L \|z_{n+1} - z^*\| + \frac{\|\mathbf{E}_n^\top \mathbf{v}_n\|}{\|\mathbf{v}_n\|}. \end{aligned}$$

Both terms on the right-hand side go to zero as  $n$  goes to infinity: the first one due to  $\lim_{n \rightarrow \infty} z_n = z^*$  and the second one since  $\lim_{n \rightarrow \infty} \frac{\|\mathbf{E}_n^\top \mathbf{v}_n\|}{\|\mathbf{v}_n\|} = 0$  by

Schlenkrich et al. [SGW10, Lemma 3]. This shows that  $\lim_{n \rightarrow \infty} \|\mathbf{B}_{n+1} - \mathbf{B}_n\| = 0$ , which concludes the proof of the intermediate claim.

From  $\lim_{n \rightarrow \infty} \|\mathbf{B}_{n+m} - \mathbf{B}_n\| = 0$  for any fixed  $m \in \mathbb{N}$  it follows that for any sequence  $(j_n) \subset \mathbb{N}$  with  $\sup_n |j_n - n| < \infty$  there holds  $\lim_{n \rightarrow \infty} \|\mathbf{B}_{j_n} - \mathbf{B}_n\| = 0$ . This implies for any such sequence  $(j_n)$  the limit  $\lim_{n \rightarrow \infty} \|\mathbf{B}_{j_n}^{-1} - \mathbf{B}_n^{-1}\| = 0$ . To establish this, note that for  $C := \max\{\sup_n \|\mathbf{B}_n\|, \sup_n \|\mathbf{B}_n^{-1}\|\}$ , which is finite by Assumption 7.1.4 and the combination of the bounded deterioration principle [SGW10, Lemma 2] with Assumption 7.1.2 (i), the set

$$\{\mathbf{A} \in \mathbb{R}^{d \times d} : \mathbf{A}^{-1} \text{ exists, } \|\mathbf{A}\| \leq C, \|\mathbf{A}^{-1}\| \leq C\}$$

includes the sequence  $(\mathbf{B}_n)$  and is compact by the Banach lemma, so inversion is a *uniformly* continuous operation on this set.

Now let us construct a sequence  $(j_n) \subset N_e$  by defining, for every  $n \in \mathbb{N}$ ,  $j_n := \arg \min_{m \in N_e} |n - m|$ . That is, for every  $n$ ,  $j_n$  denotes the member of  $N_e$  with the smallest distance to  $n$ . It is clear that  $|n - j_n| \leq M - 1$  for all  $n$ , hence  $\lim_{n \rightarrow \infty} \|\mathbf{B}_{j_n}^{-1} - \mathbf{B}_n^{-1}\| = 0$ . Using this and, again, continuity it is easy to see that

$$\lim_{n \rightarrow \infty} \|\mathbf{p}_\theta^{(n)} - \mathbf{p}_\theta^{(j_n)}\| = 0,$$

which implies by Equation C.3 that

$$\lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \lim_{n \rightarrow \infty} \mathbf{p}_\theta^{(j_n)} = \lim_{N_e \ni n \rightarrow \infty} \mathbf{p}_\theta^{(n)} = \left. \frac{\partial \mathcal{L}}{\partial \theta} \right|_{z^*},$$

thereby establishing the claim.  $\square$

**Remark** An inspection of the proof reveals that if  $\mathbf{B}_n$  is never updated in the direction  $z_n$ , but only updated in the direction  $v_n$  defined in Equation 7.8, then Assumption 7.1.4 can be replaced by the significantly weaker assumption that the sequence  $(\mathbf{B}_n^{-1} \frac{\partial g_\theta}{\partial \theta} |_{z_n})$  is bounded. The price to pay is that the convergence rate of  $(z_n)$  to  $z^*$  will be slower ( $q$ -linear instead of  $q$ -superlinear) since the updates in the direction  $z_n$  are critical for ensuring fast convergence of  $(z_n)$  to  $z^*$ .

## D - Software

During this thesis, writing code brought me a lot of joy. I have tried as much as possible to contribute to open source packages both in and out of the team. In this section I will present 3 repositories I contributed to in the scope of MRI reconstruction from a software perspective. All of these repositories are Python based.

### PySAP (Python Sparse data Analysis Package)

**PySAP** is a software package that is the outcome of the **COSMIC** interdisciplinary research project (2016-2020) between the CS-MRI team at NeuroSpin and the CosmoStat laboratory, the two CEA entities where I pursued my PhD thesis. At its core, PySAP is a sparse reconstruction package that is intended to be used in multiple science contexts: astrophysics, medical imaging, non-destructive evaluation using tomographic and ultrasound imaging. It features multiple parts of which we can identify:

- the core, which features efficient wavelet implementations;
- **ModOpt**, a module that contains the optimization algorithms used in PySAP;
- the plugins, like **PySAP-MRI**, which combine the first two components with problem specific functions and classes.

The current version is tagged 0.0.5 and the next release is planned for the end of 2021.

### fastmri-reproducible-benchmark

**fastmri-reproducible-benchmark** is the repository that contains a large part of the work I did during my PhD thesis. It is very much reflecting the evolution of my code writing style during this period, and I am both ashamed of some practices I have used at the beginning of my work but also proud that it proved reproducible as intended.

The earlier models are for example implemented using the functional **API** of Keras, which is not really suited for research because it is not very flexible and does not allow appropriate debugging. I later switched for the more comfortable subclassing **API**, which is basically an object-oriented way of writing models much like PyTorch.

The trainings and evaluations carried for the benchmark were originally done using Jupyter notebooks, which are convenient when prototyping code, but do

not fit at all a proper reproducible checklist. I later used scripts instead, which can be better tracked by git, but also fit much more the supercomputer preferred workflow.

There is unfortunately too much redundancy in the code, which stems from the tradeoffs I had to make when developing research ideas. Fortunately, a lot of the code is unit tested with continuous integration, which should hopefully make all the refactoring safer.

The current big piece missing from this repository are example notebooks detailing how one can use the different networks. Models should also be easily downloadable, but GitHub is not a viable option, while the HuggingFace Hub seems much more suited.

### **tfkbnufft**

**tfkbnufft** was originally developed as a translation of the Torchkbnufft package of Muckley et al. [Muc+20]. It provides functions related to the **NUFFT**, which is crucial in non-Cartesian **MRI**. We then incorporated a better density compensation computation, suited for more general trajectories, but also gradients with respect to the k-space trajectories in order to be able to learn sampling patterns.

This package still suffers from speed limitations, but it was used in different works where it proved enough in its current state.

This package was also the first one I published on PyPI with continuous integration. Additionally, it has an example notebook which makes it the neatest package in my eyes.

**On the use of TensorFlow vs. PyTorch** I do not have a strong opinion on which framework is best in general. In my case, I carried over a legacy of previously using TensorFlow (and then Keras), when PyTorch was not available. Moreover, at the beginning of my PhD this legacy was reinforced by the support of complex tensors in TensorFlow which was not available in PyTorch. It therefore led to a lot of complex arithmetic rewriting. Also, because I was used to the way TensorFlow worked, I did not suffer from it being less user-friendly than PyTorch. Moving forward, given the efforts done by the fastMRI team who uses PyTorch, I would probably advise anyone starting fresh to use PyTorch. To nuance this advice, one might also consider that a lot of our team's efforts have been in TensorFlow, in particular regarding reconstruction networks, non-Cartesian handling, and trajectory optimization.

## E - Tutorials, documentation and courses

The desire to teach and disseminate knowledge is a core aspect of my personality. For this reason, during my thesis I tried as much as possible to help others, be they other students in the lab, strangers on the internet or hypothetical future students. While I am proud of my [StackOverflow account's content](#), I will focus in this section on the artifacts of teaching that were created in the context of my thesis.

### Git Tutorial

Because code is such an important aspect of a computational research scientist's life, sharing it and versioning it is instrumental. For these purposes, Git is the go-to tool of most people, which is why it is important to learn how to use it.

I have created a tutorial which tackles the most important aspects of using Git with GitHub in an open-source setting: it features [a presentation](#) and [a short practical exercise](#).

### Jean Zay

In order to help foster the development of the Jean Zay (JZ) ecosystem, I co-created with Loïc Estève a user documentation for the supercomputer: [jean-zay-doc.readthedocs.io](#). This documentation is accompanied by [a GitHub repo](#) where users can improve it, and [a Gitter forum](#), that is now very active and where users can exchange questions and issues they face when using JZ. The goals of this ecosystem were the following:

- Provide a more intuitive and user-friendly documentation than the [official documentation](#), which is more technical.
- Allow knowledge to be shared between different users, rather than having many users email the same questions to the assist team.

In addition to this, I added NeuroSpin-specific instructions to the [NeuroSpin wiki](#).

Finally, I tried to advocate the use of JZ by always referring to it at the end of my presentations. I also shared my personal experience with it during the [Artificial Intelligence 2021](#) event.

**More general considerations on the use of supercomputers for research.** The reason I am trying hard to have people use supercomputers, and JZ in particular in France, is because I think that it allows better research.



Firstly, they enable extended ablation studies and runs with multiple random seeds. This makes it very easy to identify truly essential components of a method, and the measure of their significance. Secondly, they provide researchers with means rivaling that of big companies, which means that they can then reproduce large scale results and experiments. Thirdly, they force you to implement your code in a way that favors reproducible research. Finally, I think that from a bigger perspective, the mutualization of resources is indispensable for the computational scientific community. Having GPUs sitting idle in laboratories, with installations and maintenance being handled individually is a loss of time and money. Mutualization helps to reduce the costs, lowers the barrier to entry and allows knowledge to be shared.

However, I do understand that some aspects of supercomputers might be detrimental to research. They can sometimes be too rigid in their use. For example, the first supercomputer I used was the TGCC of CEA, which had a very restricted internet access, unpractical for a research project. The fact that you have to ask the assist team for the installation of specific libraries is also problematic. Another big hurdle in the case of JZ, is that it relies on SLURM, a queuing system which is sometimes overwhelmed at the worst moments, i.e. AI/CV conferences deadlines.

### **NeuroSpin Deep Learning Lecture Group**

With fellow PhD students Louise Guillon, Benoît Dufumier and Alexandros Popov, we founded a lecture group on the topic of deep learning for students at NeuroSpin. The idea was that a lot of students were working on this topic in silos and not sharing their knowledge or questions, both on the theoretical and practical sides.

Therefore, in addition to functioning as a classical lecture group, where one person presents a research paper weekly, we also encouraged discussions regarding the implementation aspects of deep learning. Many papers were presented, and we managed to keep the dynamic during the pandemic with the meetings gathering generally around 10 people.

### **Python TA**

I had the opportunity to lead Python practical sessions at the IUT of Orsay. I was teaching the very basics of Python to students that just got out of high school. This experience was very interesting, but unfortunately I do not think it was fulfilling, in the sense that I was not able to explain clearly core concepts such as the difference between a variable and its value, or between the variable `a` and the string `"a"`.

## F - Ideas we tried and did not work

This section is intended to provide ideas that were promising but that we did not manage to make work correctly for various reasons. With a little bit of more effort or thought they could be great.

### AIRS-Net

One question that we often got when presenting the *XPDNet* was: “What did the winners do better than you?”. It is unfortunately hard to answer this question because the winners did not provide an open source implementation or description of their solution. The only brief idea we can have is a 10-minute presentation done at the NeurIPS Medical Imaging Workshop 2021. We tried implementing some ideas presented in this talk, but to no avail. The implementations are present in the [fastmri-reproducible-benchmark repository](#).

### Does denoising performance translate to inverse problem performance?

When we understood that the image subnetwork of unrolled networks could be thought of as a denoiser, one question immediately arose: “Can we know in advance the performance of an unrolled network for an inverse problem based on the performance of its subnetwork on image denoising?”.

We tried to answer this question empirically by comparing different subnetwork architectures on image denoising and single-coil 2D MRI reconstruction when embedded in an unrolled network. We did not find any correlation between the two.

### MomentumVarNet

The MomentumNet [San+21] is a promising framework for memory-efficient training of deep networks. Moreover, it enjoys a very practical implementation which can transform any PyTorch-implemented residual network in a momentum network. We therefore tried to transform the VarNet [Sri+20] into a momentum network and train it. Unfortunately, when increasing the number of unrolled iterations too much in order to improve the reconstruction performance, the training was not successful.

We think that it amounts to correctly tuning the different hyperparameters of training in order to achieve a correct performance. One might also need to have two or more chained momentum networks rather than one, mimicking the restart strategy of accelerated optimization algorithms.

## MDEQs for MRI reconstruction

We tried to use [MDEQs](#) [BKK20] with few architectural changes to perform MRI reconstruction. We did not manage to achieve decent performances, in particular because we had convergence issues in the backward pass. It's actually these issues that led us to investigate the internals of the backpropagation in [DEQs](#), eventually giving birth to the idea of [SHINE](#).

## Total Deep Variation reproduction

We tried to reproduce the Total Deep Variation [Kob+20] results. This architecture is very interesting and exciting because its forward pass already uses differentiation which makes it challenging to implement.

The way it was originally implemented by Kobler et al. [Kob+20] in PyTorch was by writing manually the differentiations for each component. We wanted to try to use auto-differentiation in TensorFlow to have a more concise and error-proof code. This did not work out.

## Dynamic denoising using soft-thresholding

One of the interesting aspects of the *Learnlets*, inspired by wavelets, is that the denoising is dynamic thanks to the adaptive thresholding carried out in the soft-thresholding layer. Of course this led us to try out this layer in classical architectures, for example DnCNN [Zha+17a]. We also tried out an anti-soft-thresholding layer, that would hopefully let through noise and stop signal via a residual connection  $f(\mathbf{x}, \sigma) = \mathbf{x} - \text{ST}(\mathbf{x}, \sigma)$ , because the intuition behind DnCNN is that the network is tasked with identifying the noise. While successful, this approach did not perform better than a naive dynamic denoising where you just concatenate a map with the noise to the input of each convolution.

## Model Parallelism for MRI reconstruction

In order to build even bigger networks, model parallelism can be used. It is however quite difficult to put in place with the current set of frameworks available. The tool we had identified to turn the *NC-PDNet* in a model parallel version was Mesh TensorFlow [Sha+18]. While we were having issues with its use, we found out that it was a discontinued project that was no longer maintained (or just very sparsely). We did not go any further down that road, after several tries at fixing our problem.

## G - [

Abstract in French]Résumé en français

### *Abstract in French*

*Sujet* : Réseaux neuronaux avancés pour la reconstruction d'images IRM à partir de données fortement sous-échantillonnées dans des contextes d'acquisition complexes.

Nous résumons ici les différents aspects abordés au travers de cette thèse. Après avoir décrit les enjeux et motivations qui nous ont poussé au développement des méthodes abordées dans ce travail, nous résumerons chacune des contributions.

#### **Motivations et contexte**

L'Imagerie par résonance magnétique (IRM) est une technique d'imagerie permettant de visualiser les tissus mous du corps humain de manière non invasive. C'est donc une modalité de visualisation très utilisée pour le diagnostic de nombreuses maladies et traumatismes. Cependant, l'un de ses principaux problèmes est un temps d'acquisition très élevé, de 15 à 90 minutes, qui peut la rendre peu applicable ou accroître les délais d'attente pour obtenir une date d'examen. Pour permettre d'accélérer cette acquisition, le sous-échantillonnage des données constitue l'approche la plus pertinente, mais cela crée une étape complexe de reconstruction de l'image à partir de données incomplètes. L'enjeu de la reconstruction est dès lors de tirer parti de connaissances *a priori* sur les images obtenues par IRM en général pour reconstruire une image donnée. Il existe cependant deux problèmes avec les techniques classiques basées sur la théorie de l'échantillonnage compressif :

- Le temps de reconstruction de ces méthodes itératives, qui utilisent des opérateurs parfois complexes, est long.
- Les connaissances *a priori* sur les images IRM sont souvent encodées de manière manuelle et restent de ce fait relativement simplistes.

Dans le même temps, l'apprentissage profond (*Deep Learning* en anglais) a émergé comme une technique permettant d'apprendre des connaissances *a priori* sur des objets à partir de grandes quantités de données. C'est pourquoi, de nombreux travaux de recherches essaient d'utiliser cette technique pour reconstruire des images IRM. Dans cette thèse, nous avons essayé de développer cette idée en proposant de nouvelles architectures de réseaux neuronaux pour la reconstruction d'images IRM, mais aussi en interrogeant leur applicabilité clinique et en améliorant des méthodes permettant la création de réseaux très profonds.

## Introduction à l'IRM

L'IRM par sa capacité à diagnostiquer de nombreuses pathologies et traumatismes commence à s'imposer comme une modalité d'imagerie indispensable du parcours clinique. Le grand avantage de l'IRM par rapport au scanner, son concurrent direct en imagerie, est qu'elle n'utilise pas de radiation ionisante. Il est donc possible d'utiliser l'IRM de manière répétée sans dangers pour les patients.

Essayons donc de comprendre comment il est physiquement possible de générer une image des tissus mous du corps humain sans utiliser de radiations. L'IRM se base sur le phénomène de la Résonance Magnétique Nucléaire (RMN). La RMN a lieu lorsque qu'un atome associé à un spin non-nul (p. ex. hydrogène, sodium, phosphore, etc.), est plongé dans un champ magnétique statique  $\mathbf{B}_0$ . Le spin de l'atome va alors s'aligner sur le champ magnétique et initier un mouvement de rotation en cône, appelé précession, autour de l'axe du champ magnétique statique. La résonance correspond à l'interaction entre les spins en rotation et une impulsion Radio-Fréquence (RF), aussi appelée champ  $B_1^+$ , délivrée à la fréquence de Larmor des spins précessant autour de  $\mathbf{B}_0$ . Cette interaction va exciter le spin en apportant de l'énergie au système, et en pratique augmenter l'angle de la précession. Une fois cette phase d'excitation finie, l'énergie apportée par l'impulsion est relâchée par le spin sous la forme d'une autre impulsion RF (champ RF  $B_1^-$ ) qui peut être enregistrée par une antenne sous la forme d'un signal d'induction libre (Free Induction Decay). Les caractéristiques de ce signal FID dépendent du tissu (p. ex. matière blanche ou grise dans le cerveau) dans lequel se situent les spins. Pour former une image à partir de ce signal, on procède à un encodage spatial grâce à l'application de gradients dans les trois directions de l'espace (sélection de coupe, encodage de phase et en fréquence), ce qui permet de former une image, car les différents tissus émettront des impulsions avec des caractéristiques différentes.

L'impulsion RF qui peut être enregistrée contient la contribution de tous les tissus plongés dans le champ magnétique. Les caractéristiques de celle-ci sont donc globales, et afin d'obtenir une information locale, il faut donc utiliser des variations locales du champ magnétique pour enregistrer plusieurs informations globales dont la variation dépend des variations locales du champ magnétique. Ces variations locales du champ magnétique sont appelées gradients et l'acquisition IRM correspond à l'envoi de la même impulsion RF accompagnée de différents gradients. En l'effet, les gradients font varier la fréquence de résonance, mais aussi la phase des différents spins de manière spatialement dépendante.

En négligeant certains phénomènes et en démodulant et filtrant le signal analogique, on peut obtenir le signal suivant :

$$S_{tr}(t) \propto \omega_0 \int_{V_s} B_{tr} M_{tr}(t, \mathbf{r}) e^{-\nu \gamma \mathbf{r} \cdot \int_0^t \mathbf{G}(\tau) d\tau} d\mathbf{r} \quad (\text{G.1})$$

avec  $\omega_0$  la fréquence de Larmor à laquelle tournent les spins,  $B_{tr}$  la partie transverse de l'impulsion RF,  $M_{tr}(t, \mathbf{r})$  le champ magnétique transverse,  $\gamma$  le rapport

gyrométrie de l'hydrogène et  $\mathbf{G}$  la résultante des gradients.

En notant  $\mathbf{k}(t) = \frac{\gamma}{2\pi} \int_0^t \mathbf{G}(\tau) d\tau$ , on s'aperçoit que l'opérateur qui contrôle les gradients peut faire en sorte que le signal  $S_{tr}(t)$  soit la transformée de Fourier de  $M_{tr}(t, \mathbf{r})$  qui est proportionnel à la densité de l'organe. On appelle  $\mathbf{k}$  le vecteur de l'espace  $k$ . Le choix de la trajectoire  $\mathbf{k}(t)$  définit la manière dont on traverse l'espace de Fourier de l'organe que l'on veut imager, et cela peut être fait de manière cartésienne sur une grille ou non-cartésienne pour donner une transformée de Fourier non-uniforme.

Afin d'améliorer le flux de patient, l'accessibilité de l'IRM à tous (notamment les populations sujettes aux mouvements comme les jeunes enfants et les personnes atteintes de la maladie de Parkinson), l'accélération de l'acquisition en IRM est un enjeu majeur. Pour cela, il n'y a pas beaucoup de possibilités physiques, hormis l'augmentation de la force du champ magnétique ce qui est technologiquement complexe. Il faut donc échantillonner moins de points dans l'espace  $k$ , et donc reconstruire une image à partir d'une information de Fourier incomplète.

Une des premières approches consiste à utiliser un réseau d'antennes pour obtenir une certaine redondance dans le signal enregistré. On peut aussi s'appuyer sur la symétrie hermitienne des coefficients de Fourier d'une image réelle afin de réaliser une acquisition partielle.

## Reconstruction classique en IRM

Les techniques de reconstruction en IRM sont basées sur la théorie de l'échantillonnage compressif [LDP07]. Le cœur de cette théorie est que l'on peut reconstruire parfaitement (c.-à-d. sans erreur) un signal inconnu à partir d'un nombre de mesures plus faible que celui prescrit par le théorème de Nyquist-Shannon si on le mesure de la « bonne manière » en supposant une certaine parcimonie dans le signal. Cette bonne manière de mesurer est caractérisée par la notion de cohérence de l'opérateur de mesure. La cohérence d'un opérateur est la mesure de décorrélation des vecteurs de la base d'acquisition par rapport aux vecteurs canoniques de la base de parcimonie. Si l'on considère le problème inverse :

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{G.2}$$

où  $\mathbf{y} \in \mathbb{R}^m$  décrit les observations, l'opérateur  $\mathbf{A} \in \mathbb{R}^m \times \mathbb{R}^n$  doit être incohérent dans la base où le signal à reconstruire  $\mathbf{x} \in \mathbb{R}^n$  est parcimonieux, du moins si l'on suit les théories initiales développées sur l'échantillonnage compressif [CRT06]. La notion de parcimonie signifie qu'il existe une base  $\Psi$  où la représentation de  $\mathbf{x}$  comporte peu de coefficients non nuls ( $\mathbf{x} = \Psi\mathbf{z}$  avec  $s = \text{Card}\{s_i \neq 0, \forall i = 1, \dots, n\} \ll n$ ). Dans ces cas-là, il existe des algorithmes permettant de reconstruire le signal  $\mathbf{x}$  de façon exacte, dans une situation sans bruit, à partir des mesures  $\mathbf{y}$ . En présence de bruit, la reconstruction est à erreur bornée, qui dépend du niveau de bruit [CRT06].

Cette théorie a été appliquée avec succès à l'IRM dès 2007 [LDP07] même si en réalité l'opérateur de mesure, c.-à-d. transformée de Fourier  $\mathcal{F}_\Omega$ , est cohérente dans les basses fréquences avec les bases de parcimonie classiques comme les décompositions en ondelettes  $\Psi$ . Pour pallier cette difficulté, une solution consiste à échantillonner complètement (c.-à-d. de façon déterministe) le centre de l'espace  $k$  et à ensuite introduire un échantillonnage plus parcimonieux sur les hautes fréquences. Ce principe a donné lieu à la notion d'échantillonnage à densité variable (cf. [PVW11; CCW13; Cha+14] pour les détails) et sa justification théorique tient au fait qu'il permet de casser la barrière de cohérence, donc in fine de réduire le nombre de mesures nécessaires  $m$  pour garantir une reconstruction exacte du signal  $\mathbf{x}$  à partir des données  $\mathbf{y}$  de l'espace  $k$ .

Ainsi, on peut reconstruire une image IRM même en mesurant seulement une partie de l'espace  $k$ , avec des algorithmes itératifs. Les plus typiques sont les algorithmes de gradient proximal qui résolvent les problèmes d'optimisation du type :<sup>1</sup>

$$\arg \min_{\mathbf{x} \in \mathbb{C}^n} \frac{1}{2} \|\mathbf{y} - \mathcal{F}_\Omega \mathbf{x}\|_2^2 + \lambda \|\Psi \mathbf{x}\|_1 \quad (\text{G.3})$$

où  $\Psi$  est la base dans laquelle  $\mathbf{x}$  est parcimonieux, typiquement une transformée en ondelettes. En notant  $\mathcal{R}(\cdot) = \lambda \|\Psi \cdot\|_1$ , l'algorithme proximal s'écrit :

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n - \epsilon_n \mathcal{F}_\Omega^H (\mathcal{F}_\Omega \mathbf{x}_n - \mathbf{y}) \\ \mathbf{x}_{n+1} &= \text{prox}_{\epsilon_n \mathcal{R}} (\mathbf{x}_{n+1}) \end{aligned} \quad (\text{G.4})$$

Il existe également des techniques propres à la reconstruction en IRM multicanaux. Ces techniques s'appliquent soit dans le domaine image comme SENSE [Pru+99] et se basent sur une acquisition préalable de cartes de sensibilités des canaux, soit dans le domaine de l'espace  $k$  et essaient de le compléter en calibrant des noyaux sur les parties acquises du signal. Des combinaisons entre ces techniques existent, et également avec les algorithmes de l'échantillonnage compressif comme ESPI-RiT [Uec+14].

## Introduction à l'apprentissage profond

Deux problèmes de reconstruction avec les techniques d'échantillonnage compressif empêchent cependant d'atteindre de hauts facteurs d'accélération :

- Le temps de reconstruction élevé des algorithmes itératifs.
- La base de parcimonie manuellement définie.

L'apprentissage profond pourrait permettre de casser ces deux barrières.

---

<sup>1</sup>On présente ici seulement la version mono-canal du problème IRM par souci de simplicité.

En effet, l'apprentissage profond se base sur des blocs fonctionnels relativement simples pour construire des fonctions hautement non linéaires qui ont la capacité d'approximer n'importe quelle fonction. Ces fonctions sont paramétrées par un très grand nombre de variables et leur calibration se fait à l'aide de l'algorithme de descente de gradient stochastique sur un très grand nombre de données.

## Revue de l'apprentissage profond pour la reconstruction en IRM

Partant de cet ensemble riche, il existe de nombreuses manières d'utiliser l'apprentissage profond pour la reconstruction en IRM. Il existe cependant un principe général sous-tendant toutes ces approches, la recherche d'un *a priori* pour contraindre la reconstruction d'une image IRM.

- **Plug-and-Play** [VBW13; Zha+17b; ZZZ19; MMC17; Ryu+19; Xu+20; Wei+20] : cette méthode consiste à remplacer l'opérateur proximal des algorithmes itératifs par un réseau de neurones de débruitage pré-entraîné.
- **Apprentissage agnostique** [Zhu+18] : dans ce cadre, on considère que le problème de reconstruction en IRM est un problème d'apprentissage supervisé sans connaissances particulières sur la physique du problème et le réseau de neurones passe directement de l'espace  $k$  à l'image.
- **Restauration monodomaine** [HSY19; Lee+18; Han+18] : cette méthode est relativement similaire à l'apprentissage agnostique sauf que le réseau doit cette fois apprendre la correspondance entre l'image artefactée (où l'on a appliqué la transformée de Fourier sur des coefficients incomplets) et l'image reconstruite, ou entre l'espace  $k$  incomplet et l'espace  $k$  complet.
- **Reconstruction antagoniste (adversariale en anglais)** [MNJ18; Dra+17; Ham+19] : cette méthode vient se greffer sur une autre technique d'apprentissage, en proposant de rajouter une composante antagoniste à la fonction de perte évaluée pendant la phase d'entraînement.
- **Échantillonnage Compressif Profond** [Bor+17; WRL19; Dar+21] : ce paradigme utilise un réseau génératif antagoniste (GAN en anglais) pré-entraîné pour contraindre la recherche de solutions satisfaisant les données.
- **A priori Profond sur les Images** [DH20] : cette approche est assez similaire à la précédente, mais ici, on sur-apprend un réseau convolutif sur les données, et on considère que l'*a priori* contenu dans l'architecture est suffisant pour reconstruire l'image.
- **Auto-supervisé** [Yam+20; Hu+21] : lorsque des données acquises parfaitement pour servir de vérité terrain ne sont pas disponibles, le paradigme de l'auto-supervision suggère de créer un problème à résoudre pour le réseau de neurones afin de l'entraîner sur des données incomplètes seulement.



Dans le cas de la reconstruction en IRM on peut par exemple rétrospectivement sous-échantillonner encore plus les données et demander au réseau de reconstruire celles manquantes pour lesquelles on a une vérité terrain.

- **Apprentissage de Champ Implicite** [Sun+21 ; SPX21] : l'utilisation de cette méthode est encore récente pour la reconstruction en IRM, mais prometteuse, car versatile : le concept est d'apprendre un *a priori* sous la forme d'une fonction (un champ) mettant en correspondance des coordonnées spatiales (dans l'espace image ou l'espace k) avec leur valeur dans l'espace modélisé.
- **Réseaux déroulés** [AMJ19 ; Ham+18 ; Ham+19 ; Sri+20 ; GOW19 ; Eo+18 ; AÖ18b ; Sch+18] : cette méthode est la plus utilisée aujourd'hui pour la reconstruction en IRM. Elle consiste à dérouler un algorithme de reconstruction itératif et à remplacer l'opérateur proximal par un réseau de neurones. À la différence de la méthode Plug-and-Play, le tout est appris de bout en bout sans avoir à pré-entraîner le réseau de neurones remplaçant l'opérateur proximal.

Dans cet ensemble de méthodes, il existe peu de comparaisons sur un pied d'égalité. Nous avons souhaité en réaliser une à petite échelle afin de commencer à y voir plus clair. Cette comparaison a impliqué les réseaux U-net [RFB15], KIKI-net [Eo+18], Cascade-net [Sch+18] et PDNet [AÖ18b]. Les résultats quantitatifs (cf. [tableau G.1](#)) et qualitatifs (cf. [figure G-1](#)) ont permis de constater que le PDNet était le réseau le plus prometteur des quatre, et ainsi une base pertinente pour développer de nouveaux réseaux.

Cette comparaison a été faite avec le jeu de données fastMRI [Zbo+18] qui est le premier à être suffisamment conséquent en taille pour permettre des applications d'apprentissage profond. De plus, il contient des données brutes complexes d'imagerie multicanaux, il est donc tout à fait approprié pour effectuer cette analyse dans un cadre d'acquisition réaliste.

Table G.1 : Résultats quantitatifs pour le jeu de données **fastMRI**. La moyenne et l'écart type (std) de PSNR et de SSIM sont calculés sur les 200 volumes de validation. Le temps d'exécution est donné pour la reconstruction d'un volume avec 35 coupes.

| Réseau              | PSNR-moyen (std) (dB) | SSIM-moyenne (std)   | #params | Temps d'exec. (s) |
|---------------------|-----------------------|----------------------|---------|-------------------|
| <b>Fourier adj.</b> | 29,61 ( 5,28)         | 0,657 ( 0,23)        | 0       | 0,68              |
| <b>KIKI-net</b>     | 31,38 (3,02)          | 0,712 (0,13)         | 1,25 M  | 8,22              |
| <b>U-net</b>        | 31,78 ( 6,53)         | 0,720 ( 0,25)        | 482k    | <b>0,61</b>       |
| <b>Cascade net</b>  | 31,97 ( 6,95)         | 0,719 ( 0,27)        | 425k    | 3,58              |
| <b>PD-net</b>       | <b>32,15</b> ( 6,90)  | <b>0,729</b> ( 0,26) | 318k    | 5,55              |

Référence   Fourier adj.   KIKI-net   U-net   Cascade-net   PD-net

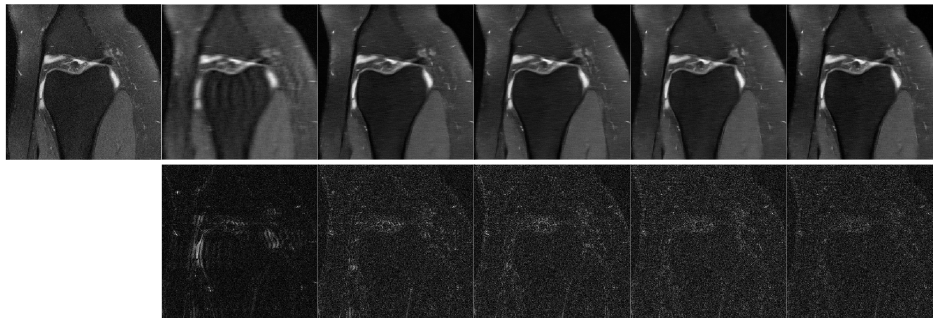


Figure G-1 : Résultats de reconstruction pour une coupe spécifique (16ème coupe de `file1000196`, de l'ensemble de validation). La ligne du haut représente la reconstruction avec les différentes méthodes, et celle du bas l'erreur absolue lors de la comparaison avec la référence.

### Nouveaux réseaux déroulés pour la reconstruction en IRM

Partant de cette conclusion, nous avons développé deux nouveaux réseaux déroulés pour la reconstruction en IRM basés sur le PDNet. Ces nouveaux réseaux devaient répondre au besoin de reconstruire des images dans des contextes plus complexes que le cadre simpliste d'imagerie 2D monocanal. Le premier réseau est le *XPDNet*, qui permet de reconstruire en imagerie 2D multicanaux. Ce réseau a été développé dans le but de participer au défi fastMRI 2020 de reconstruction de cerveaux en IRM. Il met en place les avancées les plus récentes de l'état de l'art en débruitage d'images, en réseaux déroulés et en IRM. Grâce à *XPDNet*, nous avons pu nous classer 2èmes de ce défi, dont l'évaluation finale était déterminée par un ensemble de radiologues en double aveugle. Son implémentation est disponible sur [GitHub](#). Le deuxième réseau est le *NCPDNet* dont le but est de reconstruire des données acquises de manière non-cartésienne, en imagerie 2D et 3D. Une représentation de ce réseau est disponible en [figure G-2](#). La partie originale de ce réseau concerne l'intégration d'une compensation de la densité d'échantillonnage, un mécanisme crucial pour la reconstruction d'images non-cartésiennes en IRM, mais peu utilisé dans les réseaux déroulés jusqu'à présent. Nous avons conduit une étude d'ablation ainsi qu'une étude de robustesse pour mieux comprendre comment se comportait ce réseau et conclu à l'importance de chacune des composantes, mais aussi à une certaine robustesse de celui-ci.

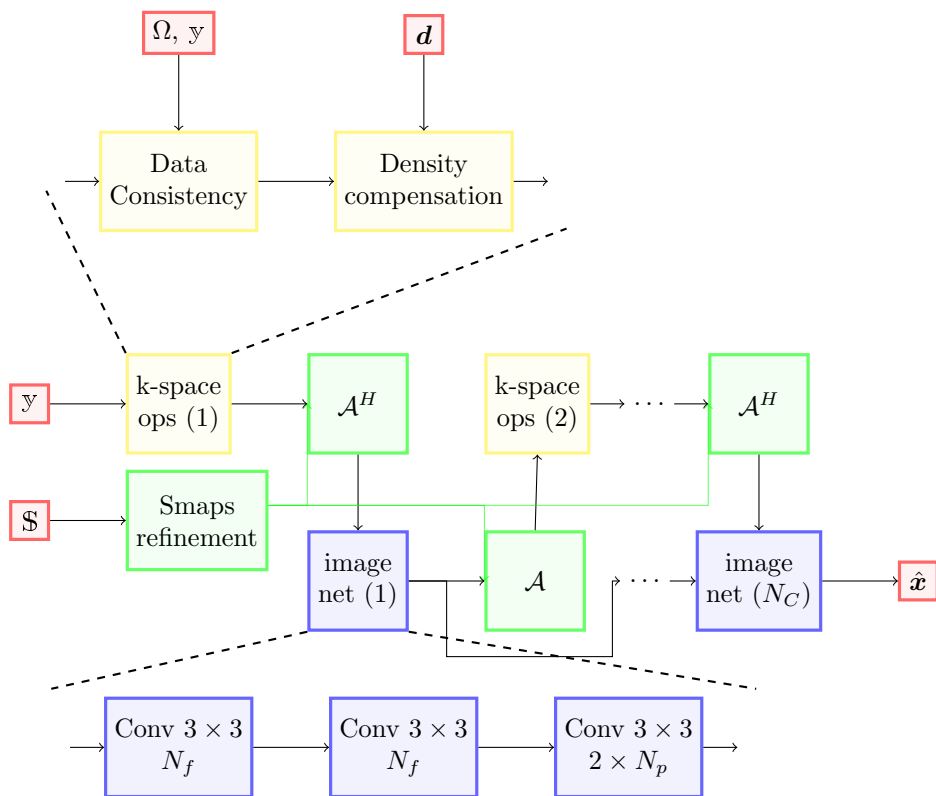


Figure G-2 : Représentation du NCPDNet.

## Applicabilité clinique de l'apprentissage profond pour la reconstruction en IRM

Nous nous sommes justement ensuite intéressés à la question de l'applicabilité clinique des réseaux de neurones. En effet, de nombreuses questions sont soulevées quant à leur utilisation dans un contexte médical et notamment leurs potentielles erreurs [Got+20]. Nous avons donc étudié cette question sous 3 angles différents.

Dans un premier temps, nous nous sommes interrogés sur les liens qui existent entre les réseaux de neurones typiques de la vision par ordinateur comme le U-net [RFB15] et les ondelettes classiquement utilisées en échantillonnage compressif. À partir de cette étude, nous avons développé un nouveau type de réseau de neurones, les *Learnlets*, qui sont en fait des ondelettes avec une forte composante apprise. Nous avons montré que les *Learnlets* permettent d'apprendre une fonction plus robuste que les réseaux de neurones classiques tout en ayant de meilleures performances que les ondelettes en débruitage.

Dans un deuxième temps, nous avons essayé de créer une méthode de quantification de l'erreur de reconstruction d'image en IRM en s'aidant de l'approche récente de *correspondance de score par débruitage* [SE19 ; SE20]. Cette méthode est basée sur l'entraînement d'un réseau de neurones pour du débruitage sur un signal et l'interprète comme le gradient de la log-distribution, appelé le score [Hyv05]. Ce score est ensuite utilisé pour échantillonner des images reconstruites, permettant ainsi de distinguer les éléments propres aux données acquises et ceux propres à l'*a priori*.

Enfin, dans un troisième temps, nous avons réalisé une comparaison du *XPDNet* avec GRAPPA [Gri+02], un algorithme d'imagerie parallèle de référence utilisé dans les scanners IRM. Cette comparaison a permis d'identifier les avantages de *XPDNet*, en particulier dans un contexte prospectif où il n'avait jamais été utilisé. Ce dernier résultat est visible en [figure G-3](#).

## Nouveaux paradigmes d'apprentissage pour les réseaux très profonds

Afin de pousser également les performances pures des modèles d'apprentissage profond, il est important de pouvoir construire de plus grands réseaux de neurones. Cependant, l'entraînement de ceux-ci est rapidement impossible du fait des contraintes de mémoire imposées par la rétro-propagation du gradient dans l'algorithme de descente de gradient stochastique. Nous nous sommes donc intéressés aux méthodes permettant la réduction de ces contraintes de mémoire, et notamment aux Modèles d'Équilibre Profonds (DEQs en anglais) [BZK19 ; BKK20]. Ces modèles proposent de définir la sortie d'un réseau de neurones de manière implicite comme la solution d'une équation de point fixe. Ainsi, il est possible d'optimiser les poids de ce réseau en ne se basant plus sur la rétro-propagation classique, mais en utilisant le théorème des fonctions implicites. En s'intéressant à ce type de modèles,

prometteurs pour la reconstruction en IRM [GOW21], nous nous sommes aperçus qu'ils étaient extrêmement lents lors de la phase d'entraînement. Nous avons donc proposé une méthode, SHINE (pour SHaring the INverse Estimate en anglais), qui se base sur un produit de l'algorithme de résolution de l'équation de point fixe pour accélérer le calcul de la dérivée. En plus de cela, nous avons pu montrer la généralité de SHINE en l'appliquant aussi à des problèmes d'optimisation biniveaux pour le réglage des hyper-paramètres [Ped16].

## Conclusion

En conclusion, dans cette thèse, nous avons pu produire des contributions de différents ordres. Certaines sont applicatives et d'autres méthodologiques, certaines sont des idées nouvelles ou d'autres correspondent à un parangonnage de méthodes issues de la littérature sur des grands jeux de données ouvertes. La force de ces études réside dans la mise à disposition d'un code documenté en libre accès. Nous espérons ainsi avoir aidé la communauté de la reconstruction en IRM à travers ces efforts.

Les perspectives de développement se concentrent autour de 3 aspects :

- L'application à la reconstruction d'images en IRM dans des contextes complexes d'idées d'entraînement de réseaux de neurones frugales, c.-à-d. à faible consommation de mémoire.
- Les entraînements hybrides ou conjoints du réseau de reconstruction et du schéma d'acquisition.
- L'intégration de corrections physiques au-delà du modèle d'acquisition idéalisé de Fourier au sein des réseaux déroulés.

\* \* \*  
\* \*  
\*

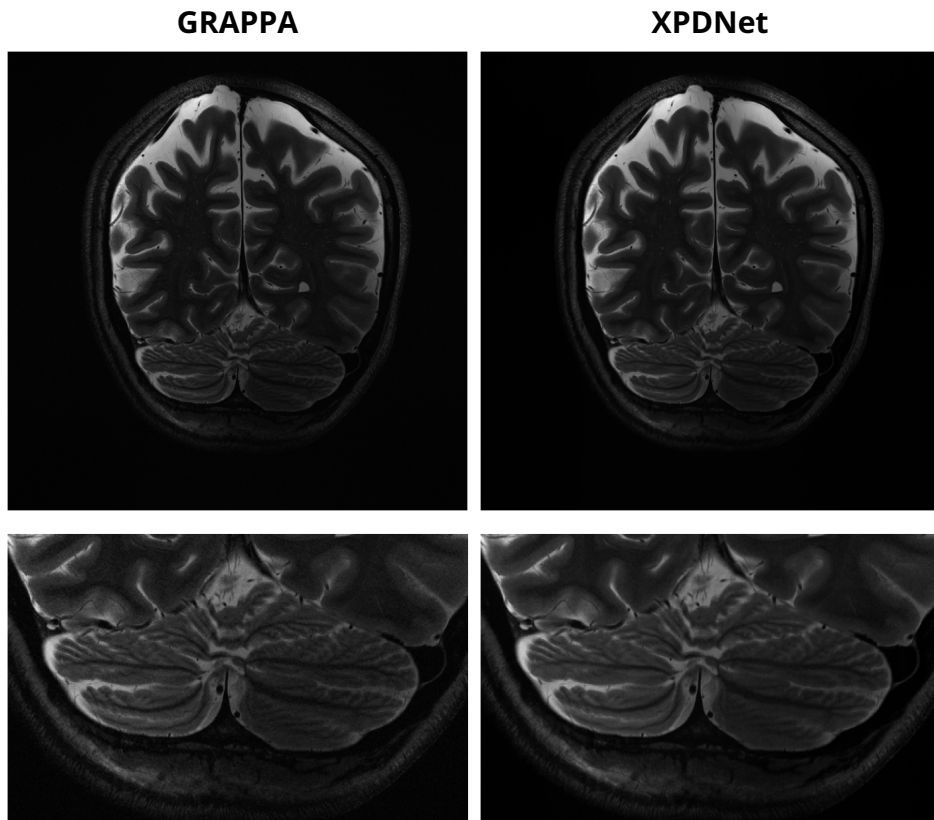


Figure G-3 : Images des modules des résultats de reconstruction pour un cerveau [Mar+16] acquis avec un facteur d'accélération de 2, un contraste  $T_2$  et une intensité de champ magnétique de 7T. La ligne supérieure représente la reconstruction avec les différentes méthodes et la ligne inférieure représente un zoom dans le cervelet, un élément anatomique qui n'était pas présent dans l'ensemble d'entraînement de *XPDNet*.



## List of Figures

|  |    |
|--|----|
| 1.0-1 <b>Example of an MR image:</b> MR image of the knee taken from the fastMRI dataset [Zbo+18]. . . . .   | 20 |
| 1.1-2 <b>Number of MRI scans per year per 1000 population:</b> figure courtesy of <i>Health at a Glance 2019: OECD Indicators - Medical technologies</i> [19d]. . . . .  | 21 |
| 1.1-3 <b>Number of MRI and CT machines per year per million population:</b> figure courtesy of <i>Health at a Glance 2019: OECD Indicators - Medical technologies</i> [19d]. . . . .   | 21 |
| 1.1-4 <b>What can we diagnose with MRI?</b> This illustration provides a non-exhaustive list of all the diagnoses that can be carried out with MRI. All the information was compiled from the works of Reimer et al. [Rei+10] and Runge et al. [RTH19]. . . . .  | 22 |
| 1.2-5 <b>Illustration of the precession of a spin in a magnetic field:</b> the green arrow represents the $B_0$ magnetic field, while the black arrow represents the magnetic moment of the particle. Illustration courtesy of <i>Larmor precession Wikipedia page</i> [12b]. . . . .  | 24 |
| 1.2-6 <b>Example of a k-space with its corresponding anatomical image:</b> The raw data is from the fastMRI dataset [Zbo+18]. The k-space is in log-scale and only the magnitude of the 2 images are represented. We selected only a single coil from the 16 coils available for illustrative purposes. . . . .                            | 26 |
| 1.3-7 <b>Reconstruction in PI at different undersampling factors:</b> an MR image of a brain reconstructed using GRAPPA [Gri+02], a common reconstruction technique, for different AFs in 2D multicoil imaging. The raw data is from the fastMRI dataset [Zbo+18]. The AF of 4 is still readable while the AF of 8 is unpractical. . . . . | 29 |
| 2.3-1 <b>SENSE reconstruction:</b> image courtesy of Elster et al. [EB01]. . . . .   | 38 |
| 2.3-2 <b>(a) Autocalibration Signal - (b) Reconstruction.</b> The central part of the k-space is fully sampled and can be used as an Autocalibration Signal to calibrate the kernels. The reconstruction is then carried out linearly on the rest of the k-space. Images courtesy of Elster et al. [EB01]. . . . .                         | 39 |
| 2.3-3 <b>ACS lines selection for kernel calibration during GRAPPA :</b> the target line $\mathbf{y}_{ACS}^{(j)}$ is in orange, while the source lines $\mathbf{y}_{ACS}^{(-j,i)}$ are in green. The other sampled lines of the ACS are in blue. . . . .  | 40 |
| 3.1-1 <b>Timeline of Deep Learning.</b> . . . . .  | 46 |




|  |    |
|--|----|
| 3.2-2 <b>Samples of the ImageNet dataset</b> [Den+09], courtesy of Karpathy [Kar19]. . . . .   | 51 |
| 3.3-3 <b>Illustration of the different choices of <math>\mathcal{S}_{b,i,j,k}</math></b> . Courtesy of Wu et al. [WH18]. . . . .   | 56 |
| 4.2-1 <b>Illustration of the U-net</b> , courtesy of Ronneberger et al. [RFB15]. In our case the output is not a segmentation map but a reconstructed image of the same size (we perform zero-padding to prevent decreasing sizes in convolutions). . . . .  | 68 |
| 4.2-2 <b>Unrolled networks</b> . The common backbone between the Cascade net, the KIKI-net and the PD-net. US mask stands for under-sampling mask. DC stands for data consistency. (I)FFT stands for (Inverse) Fast Fourier Transform. $N_{k,d}$ is the number of convolution layers applied in the k-space. $N_{i,d}$ is the number of convolution layers applied in the image space. $N_C$ is the total number of alternations between the k-space and the image-space. It is worth mentioning that in the case of PD-net, the data consistency step is not performed with a replacement operator but with a residual, the Fourier operators are carried out with the original undersampling mask, and a buffer is concatenated along with the current iteration to allow for some memory between iterations and learn the acceleration (in the k-space net -dual net- it is also concatenated with original k-space input). In the case of the Cascade net, $N_{k,d} = 0$ , only the data consistency is performed in the k-space. In the case of the KIKI-net, there is no residual connection in the k-space. However, the k-space and image space nets could potentially be any kind of image-to-image neural network. . . . . | 69 |
| 4.2-3 <b>Illustration of the Cascade-net</b> , courtesy of Schlemper et al. [Sch+18]. Here each $C_i$ is a convolutional block of 64 filters (48 in our implementation) followed by a ReLU nonlinearity, $n_d$ is the number of such convolutional blocks forming a convolutional sub-network between each data consistency layer $DC$ , and $n_c$ is the number of convolutional subnetworks. . . . .   | 70 |
| 4.2-4 <b>Illustration of the KIKI-net</b> , courtesy of Eo et al. [Eo+18]. The KCNN and ICNN are convolutional neural networks composed of a number of convolutional blocks varying between 5 and 25 (we implemented 25 blocks for both KCNN and ICNN), each followed by a ReLU nonlinearity and featuring between 8 and 64 filters (we implemented 32 filters). For both the varying numbers, Eo et al. [Eo+18] show that the higher, the better. The ICNN also features a residual connection. . . . .   | 70 |

|   |     |
|---|-----|
| 4.2-5 <b>Illustration of the PD-net</b> , courtesy of Adler et al. [AÖ18b]. Here $\mathcal{T}$ denotes the measurement operator, which in our case is the undersampled FT, $\mathcal{T}^*$ its adjoint, $g$ is the measurements, which in our case are the undersampled k-space measurements, and $f_0$ and $h_0$ are the initial guesses for the direct and measurement spaces (the image and k-space in our case). The initial guesses are zero tensors. Because we transform complex-valued data into 2-channel real-valued data, the number of channels at the input and the output of the convolutional subnetworks are multiplied by 2 in our implementation. . . . . | 71  |
| 4.2-6 <b>Reconstruction results for a specific slice (16th slice of file1000196, part of the validation set)</b> . The first row represents the reconstruction using the different methods, while the second represents the absolute error when compared to the reference. . . . .  | 75  |
| 4.2-7 <b>Reconstruction results for a specific slice (15th slice of sub-OAS30367_ses-d3396_T1w.n part of the validation set)</b> . The top row represents the reconstruction using the different methods, while the bottom row represents the absolute error when compared to the reference. . . . .  | 76  |
| 5.1-1 <b>General unrolled networks architecture</b> . Skip and residual connection are omitted for the sake of clarity. $y$ are the undersampled measurements, in our case the k-space measurements, $\Omega$ is the undersampling scheme, $F$ is the measurement operator, in our case the FT, and $\hat{x}$ is the recovered solution. . . . .  | 84  |
| 5.1-2 <b>Magnitude reconstruction results for the different fastMRI contrasts at AF 4</b> . The top row represents the ground truth, the middle one represents the reconstruction from a retrospectively undersampled k-space, and the bottom row represents the absolute error when comparing the two. The average image quantitative metrics are given for 30 validation volumes. . . . .   | 98  |
| 5.1-3 <b>Magnitude reconstruction results for the different fastMRI contrasts at AF 8</b> . The top row represents the ground truth, the middle one represents the reconstruction from a retrospectively undersampled k-space, and the bottom row represents the absolute error when comparing the two. The average image quantitative metrics are given for 30 validation volumes. . . . .   | 99  |
| 5.2-4 <b>Schematic representation of the two multi-shot non-Cartesian (radial and spiral) readouts considered here for 2D imaging</b> . In our setting, we used $N_s = 100$ shots, each of them consisting of 640 samples giving a total of $m = 64,000$ k-space measurements. Here only 10 shots are presented. . . . .  | 100 |

|        |   |     |
|--------|---|-----|
| 5.2-5  | <b>Schematical illustration of the k-space trajectory considered in this work for 3D imaging.</b> Each of them uses 100 spokes and has a total of 64k measurements stacked 176 times across the additional dimension. . . . .   | 100 |
| 5.2-6  | <b>Single-coil knee dataset:</b> Quantitative results of the different networks in the single-coil setting for both fastMRI contrasts. . . .  | 101 |
| 5.2-7  | <b>Multicoil knee dataset:</b> Quantitative results of the different networks in the multicoil setting for both fastMRI contrasts. . . . .  | 102 |
| 5.2-8  | <b>3D OASIS brain dataset:</b> PSNR distribution of the different networks in the 3D radial undersampling scenario. . . . .   | 102 |
| 5.2-9  | <b>Multicoil knee dataset (reverse setting):</b> Quantitative results of the different networks in the multicoil reverse setting for both fastMRI contrasts. . . . .  | 103 |
| 5.2-10 | <b>2D multicoil fastMRI brain dataset:</b> Quantitative results of the different networks in the brain multicoil setting for the multiple contrasts (T1, T2, FLAIR) with some distinction on T1-w imaging related to Gadolinium injection (T1-PRE, T1-POST). . . . .  | 104 |
| 5.2-11 | <b>2D multicoil (<math>AF = 8</math>):</b> Quantitative results of the different networks in the multicoil setting for both fastMRI contrasts (knee dataset) with a higher AF during validation ( $AF = 8$ ) compared to training ( $AF = 4$ ). . . . .   | 105 |
| 5.2-12 | <b>2D single-coil radial acquisition (knee fastMRI dataset, PD contrast):</b> Reconstruction results for a specific slice (16th slice of file1001184, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image. . . . .     | 106 |
| 5.2-13 | <b>2D multicoil radial acquisition (knee fastMRI dataset, PDFS contrast):</b> Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image. . . . .     | 107 |
| 5.2-14 | <b>3D radial acquisition (OASIS dataset, T1 contrast):</b> Reconstruction results for a specific slice (101st slice of sub-OAS30001_ses-d0129_run-01_T1w, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image. . . . . | 108 |

|       |   |     |
|-------|---|-----|
| 5.2-1 | <b>2D multicoil radial acquisition (reverse setting, knee dataset):</b><br>Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set) with networks trained with spiral trajectories. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red square in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image. . . . .              | 109 |
| 5.2-1 | <b>2D multicoil spiral acquisition (reverse setting, knee dataset):</b><br>Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set) with networks trained on radial trajectories. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image. . . . .                 | 110 |
| 5.2-1 | <b>2D multicoil radial acquisition (brain fastMRI dataset, FLAIR contrast):</b> Reconstruction results for a specific slice (6th slice of file_brain_AXFLAIR_200_6002447) from the brain fastMRI dataset with networks trained on knee data. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The volume-wise PSNR and SSIM scores are shown on the top of each full FOV image. . . . . | 111 |
| 5.2-1 | <b>2D multicoil radial acquisition (knee fastMRI dataset, PDFS contrast) <math>AF = 8</math>:</b> Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set) for an AF 8. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. . . . .   | 112 |
| 6.1-1 | <b>Schematic representation of the learnlets model, with <math>m = 2</math> scales.</b> The red nodes are inputs/outputs. The lightly green nodes correspond to functions whose parameters can be learned. Note that the standard deviation of the noise before thresholding is not learned but rather estimated, and is omitted in this diagram for clarity. . . . .   | 118 |
| 6.1-2 | <b>Filters.</b> Visualization of learnlets analysis filters for four different scales. . . . .  | 121 |
| 6.1-3 | <b>Denoising performance.</b> Ratio of the denoised image PSNR compared to the original noisy image PSNR for different standard deviations of the noise added to the test images for all considered models. The train noise standard deviation range was [0; 55]. . . .   | 124 |

|   |     |
|---|-----|
| 6.1-4 <b>Exact reconstruction.</b> Ratio of the denoised image PSNR compared to the original noisy image PSNR for different standard deviations of the noise added to the test images for learnlets with and without forcing exact reconstruction. The train noise standard deviation range was $[0; 55]$ . The number of filters used was 64. . . . .  | 125 |
| 6.1-5 <b>Influence of model size.</b> PSNR difference (in dB) with respect to wavelets denoising for different standard deviations of the noise added to the test images for U-nets of various sizes. The striped bars correspond to negative differences. The train noise standard deviation range was $[0; 55]$ . . . . .   | 126 |
| 6.1-6 <b>Exact reconstruction for U-nets.</b> Ratio of the denoised image PSNR compared to the original noisy image PSNR for different standard deviations of the noise added to the test images for U-net with and without exact reconstruction. The train noise standard deviation range was $[0; 55]$ . . . . .  | 127 |
| 6.1-7 <b>Generalization to astrophysical images.</b> Denoising results for an astrophysical image contaminated with a noise of $\sigma = 50$ . The last two images correspond to the subtraction of the original image to its denoised version. . . . .   | 128 |
| 6.1-8 <b>Low-data regimes.</b> The PSNR of the denoised image at $\sigma = 25$ added to the test images as a function of the number of samples used during training. The train noise standard deviation range was $[0; 55]$ . . . . .   | 129 |
| 6.1-9 <b>Denoising results for a specific image in the BSD68 dataset.</b> The noise standard deviation used was of 30. Parameters used for the methods are the same as for Figure 6.1-3. . . . .  | 137 |
| 6.2-1 <b>Bayesian posterior sampling for MRI reconstruction.</b> The top leftmost image is the ground truth image. The top second to the left image is the zero-filled retrospectively undersampled image $F^T y$ . The top third to the left image is the reconstruction of the undersampled image by the UPDNet. All the other images are denoised samples from the estimated posterior distribution obtained by a tempered HMC. The zero-filling achieves a PSNR of 25.55 dB, each sample 27.63 dB on average, the mean of the samples 30.04 dB and the neural network 32.15 dB. A zoom of the region in the red square is provided in the Appendix (Table A). An animation of posterior samples is available online  . . . . . | 138 |
| 6.3-1 <b>Magnitude reconstruction results for a specific fastMRI slice at AF 4.</b> The top row represents the reconstruction using the different methods, while the bottom one represents the error when compared to the reference. . . . .  | 139 |

|        |   |     |
|--------|---|-----|
| 6.3-1  | <b>Magnitude reconstruction results for a specific fastMRI slice at AF 8.</b> The top row represents the reconstruction using the different methods, while the bottom row represents the error when compared to the reference. . . . .  | 140 |
| 6.3-13 | <b>Magnitude reconstruction results for a brain acquired at AF 2, contrast T2, and field strength of 7T.</b> The top row represents the reconstruction using the different methods, while the bottom one represents a zoom in the cerebellum region, an anatomical feature that was not present in the XPDNet training set. . . . .   | 141 |
| 6.3-14 | <b>Magnitude reconstruction results for a phantom acquired at AF 8.</b> The top row represents the reconstruction using the different methods, while the bottom one represents the error when compared to the reference. . . . .  | 142 |
| 7.1-1  | <b>Bilevel optimization:</b> Convergence of held-out test loss for different hyperparameter optimization methods on the $\ell_2$ -regularized LR problem for the 2 datasets (20news [Lan95] and real-sim [Fan11]) SHINE achieves the best performances for both problems while the Jacobian-Free method is much slower, in particular on 20news. Note that the kink for HOAG on real-sim does not mean it is better as the optimization stops once the validation loss has converged and not the test one. The typical loss order of magnitude is $10^2$ . An extended figure with more methods is provided in 14. . . . .  | 156 |
| 7.1-2  | <b>Bilevel optimization with OPA:</b> ( <i>left</i> ) Convergence of different hyperparameter optimization methods on the $\ell_2$ -regularized LR problem for the 20news dataset [Lan95] on held-out test data. SHINE with OPA achieves similar performance as SHINE without OPA but with better convergence guarantees. ( <i>right</i> ) Evaluation of the inversion quality in direction $\mathbf{v}$ using OPA $\mathbf{b} = \mathbf{B}_n^{-1}\mathbf{v}$ compared to the exact inverse $\mathbf{a} = \mathbf{J}_{g_\theta}(\mathbf{z}^*)^{-1}\mathbf{v}$ for 3 different directions: the prescribed direction, the Krylov direction and a random direction. The points represent the cosine similarity between $\mathbf{a}$ and $\mathbf{b}$ as a function of the ratio of their norm and the closer to (1, 1) the better. The inverse in the prescribed direction is better than in random direction. . . . . | 157 |

|       |   |     |
|-------|---|-----|
| 7.1-3 | <b>DEQ</b> : Top-1 accuracy function of backward pass runtime for the different methods considered to train DEQs, on CIFAR [Kri09] and ImageNet [Den+09]. The original DEQ training method corresponds to the Full backward pass points and the vanilla SHINE and Jacobian-Free methods correspond to direct use of the inverse approximation without further refinement. The other points correspond to further refinements of the different methods with different number of iterations used to invert $\mathbf{J}_{g\theta}(z^*)$ in the direction of $\nabla_z \mathcal{L}(z^*)$ . This highlights the trade-off between computations and performances driving the refinement choice. . . . . | 158 |
| A-1   | <b>2D single-coil spiral acquisition (knee fastMRI dataset, PD contrast)</b> : Reconstruction results for a specific slice (16th slice of file1001184, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel. . . . .   | 170 |
| A-2   | <b>2D multicoil spiral acquisition (knee fastMRI dataset, PDFS contrast)</b> : Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set). The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel. . . . .   | 171 |
| A-3   | <b>2D multicoil spiral acquisition (brain fastMRI dataset, FLAIR contrast)</b> : Reconstruction results for a specific slice (6th slice of file_brain_AXFLAIR_200_6002447) from the brain fastMRI dataset with networks trained on knee data. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel. . . . .   | 172 |
| A-4   | <b>2D multicoil spiral acquisition (knee fastMRI dataset, PDFS contrast) AF 8</b> : Reconstruction results for a specific slice (16th slice of file1000000, part of the validation set) for an AF 8. The top row represents the reconstruction using the different methods, while the bottom one represents the zoom highlighted by a red frame in the top row. The PSNR/SSIM scores are inserted in red in the top left corner in each panel. . . . .  | 173 |
| A-5   | Zoom of Bayesian posterior sampling for MRI reconstruction. The ordering is the same as in Figure 6.2-10. . . . .   | 176 |

|     |   |     |
|-----|---|-----|
| A-6 | <b>Bilevel optimization:</b> Convergence of different hyperparameter optimization methods on the $\ell_2$ -regularized LR problem for two datasets (20news [Lan95] and real-sim [Fan11]) on held-out test data. . . . .   | 178 |
| A-7 | <b>Bilevel optimization on regularized nonlinear least squares:</b> Convergence of different hyperparameter optimization methods on the $\ell_2$ -regularized nonlinear least squares for the 20news [Lan95] dataset on held-out test data. . . . .   | 179 |
| A-8 | <b>Quality of the inversion using OPA in DEQs :</b> Ratio of the inverse approximation over the exact inverse function of the cosine similarity between the inverse approximation $\mathbf{b} = \nabla_z \mathcal{L}(\mathbf{z}^*) \mathbf{B}_n^{-1}$ and the exact inverse $\mathbf{a} = \nabla_z \mathcal{L}(\mathbf{z}^*) \mathbf{J}_{g_\theta}(\mathbf{z}^*)^{-1}$ for different methods. For OPA, the extra update frequency is 5. 100 runs were performed with different batches. . . . . | 181 |
| G-1 | Résultats de reconstruction pour une coupe spécifique (16ème coupe de file1000196, de l'ensemble de validation). La ligne du haut représente la reconstruction avec les différentes méthodes, et celle du bas l'erreur absolue lors de la comparaison avec la référence. . . .  | 205 |
| G-2 | Représentation du <i>NCPDNet</i> . . . . .  | 206 |
| G-3 | Images des modules des résultats de reconstruction pour un cerveau [Mar+16] acquis avec un facteur d'accélération de 2, un contraste $T_2$ et une intensité de champ magnétique de 7T. La ligne supérieure représente la reconstruction avec les différentes méthodes et la ligne inférieure représente un zoom dans le cervelet, un élément anatomique qui n'était pas présent dans l'ensemble d'entraînement de <i>XPDNet</i> . . .   | 209 |





## List of Tables

|     |   |    |
|-----|---|----|
| 4.1 | <b>Quantitative results for the fastMRI dataset.</b> PSNR and SSIM mean and standard deviations are computed over the 200 validation volumes. Runtimes are given for the reconstruction of a volume with 35 slices. . . . .   | 74 |
| 4.2 | <b>Quantitative results for the fastMRI dataset with the Proton-Density with Fat Suppression (PDFS) contrast.</b> PSNR and SSIM mean and standard deviations are computed over the 99 validation volumes. Runtimes are given for the reconstruction of a volume with 35 slices. . . . . | 74 |
| 4.3 | <b>Quantitative results for the fastMRI dataset with the Proton-Density (PD) contrast.</b> PSNR and SSIM mean and standard deviations are computed over the 100 validation volumes. Runtimes are given for the reconstruction of a volume with 40 slices. . . . .                       | 75 |
| 4.4 | <b>Quantitative results for the OASIS dataset.</b> PSNR and SSIM mean and standard deviations are computed over the 200 validation volumes. Runtimes are given for the reconstruction of a volume with 32 slices. . . . .   | 75 |
| 5.1 | Mean PSNR / SSIM on the validation volumes of the different approaches averaged over both contrasts (knee fastMRI) in the single-coil setting. The best results are in bold font. . . . .   | 92 |
| 5.2 | Mean PSNR / SSIM on the validation volumes of the different approaches averaged over both contrasts (knee fastMRI) in the multicoil setting. The best results are in bold font. . . . .   | 93 |
| 5.3 | Mean PSNR / SSIM on the validation volumes of the different approaches for the OASIS brain dataset (3D setting). The best results are in bold font. . . . .   | 93 |
| 5.4 | Mean PSNR/SSIM on the validation volumes of the different approaches averaged over both contrasts (knee fastMRI) in the multicoil reverse setting. Best results are in bold font. . . . .   | 94 |
| 5.5 | Mean PSNR/SSIM on the validation volumes of the different approaches averaged over all brain fastMRI imaging contrasts in the multicoil setting. . . . .  | 94 |
| 5.6 | Mean PSNR / SSIM on the validation volumes of the different approaches for both contrasts (knee fastMRI) in the multicoil setting for $AF = 8$ . . . . .  | 95 |

|     |  |     |
|-----|--|-----|
| 5.7 | Reconstruction times of a single slice in milliseconds for the different networks in the different acquisition scenarios based on 2D/3D radial undersampling. . . . .  | 95  |
| 6.1 | Runtimes of the different models for the denoising of one image. Parameters used are the same as Figure 6.1-3. . . . .   | 125 |
| A.1 | PSNR for different standard deviations of the noise added to the test images for every model in Figure 6.1-3 and the original noisy images. . . . .  | 174 |
| A.2 | PSNR for different standard deviations of the noise added to the test images for both models in Figure 6.1-4 and the original noisy images. . . . .  | 174 |
| A.3 | PSNR for different standard deviations of the noise added to the test images for all the models in Figure 6.1-5 and the original noisy images. . . . .   | 175 |
| A.4 | PSNR for different standard deviations of the noise added to the test images for both models in Figure 6.1-6 and the original noisy images. . . . .  | 175 |
| A.5 | PSNR at $\sigma = 25$ added to the test images as a function of the number of samples used during training for the three models in Figure 6.1-8. . . . .   | 176 |
| A.6 | Nonlinear spectral radius obtained by the power method for the fixed-point defining subnetwork for the 3 different methods. . . . .  | 179 |
| A.7 | The time required for each method on the different datasets during the equilibrium training. For the forward and backward passes, the time is measured offline, for a single batch of 32 samples, with a single GPU, using the median to avoid outliers. This time is given in milliseconds. For the epochs, the time is measured by taking an average of the 6 first epochs, and given in hours-minutes for Imagenet and minutes-seconds for CIFAR. The epoch time for SHINE without improvement on Imagenet is not given because it never reaches the 26 forward steps: the implicit depth is too short. Fallback is not used for CIFAR. Numbers in parentheses indicate the number of inversion steps for the refined versions. . . . . | 180 |
| A.8 | <b>CIFAR DEQ OPA results</b> : Top-1 accuracy of different methods on the CIFAR dataset, and epoch mean time. . . . .  | 181 |
| B.1 | Training times in hours (h) for the different networks in the different settings. . . . .  | 183 |

G.1 Résultats quantitatifs pour le jeu de données **fastMRI**. La moyenne et l'écart type (std) de PSNR et de SSIM sont calculés sur les 200 volumes de validation. Le temps d'exécution est donné pour la reconstruction d'un volume avec 35 coupes. . . . . 204



## Bibliography

- [08] *e-MRI courses by IMAIOS*. <https://www.imaios.com/en/e-Courses/e-MRI>. Accessed: 2021-10-08. 2008.
- [10] *Kaggle*. <https://www.kaggle.com/>. Accessed: 2021-11-15. 2010.
- [12a] *Definition of non-invasive in the Free Medical Dictionary*. <https://medical-dictionary.thefreedictionary.com/non-invasive>. Accessed: 2021-10-07. 2012.
- [12b] *Larmor precession Wikipedia page*. [https://en.wikipedia.org/wiki/Larmor\\_precession](https://en.wikipedia.org/wiki/Larmor_precession). Accessed: 2021-10-09. 2012.
- [18] *NHS: How it's performed - MRI scan*. <https://www.nhs.uk/conditions/mri-scan/what-happens/>. Accessed: 2021-10-11. 2018.
- [19a] *GE Healthcare: How much does an MRI cost?* <https://www.gehealthcare.com/article/much-does-an-mri-cost>. Accessed: 2021-10-11. 2019.
- [19b] *Google Colaboratory*. <https://colab.research.google.com>. Accessed: 2021-11-15. 2019.
- [19c] *Jean Zay supercomputer*. <http://www.idris.fr/>. Accessed: 2021-11-15. 2019.
- [19d] *Health at a Glance 2019: OECD Indicators - Medical technologies*. <https://www.oecd-ilibrary.org/sites/eadc09d-en/index.html?itemId=/content/en>. Accessed: 2021-10-06. 2019.
- [AB13] G. Alain and Y. Bengio. "What regularized auto-encoders learn from the data generating distribution". In: *1st International Conference on Learning Representations, ICLR 2013 - Conference Track Proceedings*. Vol. 15. 2013, pp. 3743-3773.
- [Aba+16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Tech. rep. 2016.
- [Abu+16] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan and S. Vijayanarasimhan. *Youtube-8m: A large-scale video classification benchmark*. Tech. rep. 2016.

- [Adc+17] B. Adcock, A. C. Hansen, C. Poon and B. Roman. “Breaking the coherence barrier: A new theory for compressed sensing”. In: *Forum of Mathematics, Sigma*. Vol. 5. Cambridge University Press. 2017.
- [Ahm+19] A. H. Ahmed, R. Zhou, Y. Yang, P. Nagpal, M. Salerno and M. Jacob. “Free-breathing and ungated dynamic MRI using navigator-less spiral STORM”. In: *IEEE Transactions on Medical Imaging* (2019).
- [AKC86] C. Ahn, J. Kim and Z. Cho. “High-speed spiral-scan echo planar NMR imaging-I”. In: *IEEE transactions on medical imaging* 5.1 (1986), pp. 2–7.
- [AMJ19] H. K. Aggarwal, M. P. Mani and M. Jacob. “MoDL: Model-Based Deep Learning Architecture for Inverse Problems”. In: *IEEE Transactions on Medical Imaging* 38.2 (2019), pp. 394–405.
- [AMJ20] H. K. Aggarwal, M. P. Mani and M. Jacob. “Modl-mussels: Model-based deep learning for multishot sensitivity-encoded diffusion mri”. In: *IEEE Transactions on Medical Imaging* 39.4 (2020), pp. 1268–1277.
- [AÖ18a] J. Adler and O. Öktem. *Deep Bayesian Inversion*. Tech. rep. Nov. 2018.
- [AÖ18b] J. Adler and O. Öktem. “Learned Primal-Dual Reconstruction”. In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1322–1332.
- [Arb+11] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik. “Contour Detection and Hierarchical Image Segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.5 (May 2011), pp. 898–916.
- [AZ17] B. Amos and J. Zico Kolter. “OptNet: Differentiable Optimization as a Layer in Neural Networks”. In: *ICML*. 2017.
- [Bar93] A. R. Barron. “Universal approximation bounds for superpositions of a sigmoidal function”. In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945.
- [BB12] J. Bergstra and Y. Bengio. “Random Search for Hyper-Parameter Optimization Yoshua Bengio”. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.
- [BBW16] J. Bigot, C. Boyer and P. Weiss. “An analysis of block sampling strategies in compressed sensing”. In: *IEEE transactions on information theory* 62.4 (2016), pp. 2125–2139.

- [BCB15] D. Bahdanau, K. Cho and Y. Bengio. "Neural machine translation by jointly learning to align and translate". In: *ICLR*. 2015.
- [BD18] E. Buber and B. Diri. "Performance analysis and CPU vs GPU comparison for deep learning". In: *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*. IEEE. 2018, pp. 1–6.
- [BDM73] C. G. Broyden, J. E. jun. Dennis and J. J. More. "On the local and superlinear convergence of quasi-Newton methods". English. In: *Journal of the Institute of Mathematics and its Applications* 12 (1973), pp. 223–245.
- [Bel+19] M. Belkin, D. Hsu, S. Ma and S. Mandal. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.
- [Ber+21] A. S. Berahas, M. Jahani, P. Richtárik and M. Takáč. "Quasi-Newton Methods for Machine Learning: Forget the Past, Just Sample". In: *Optimization Methods and Software* (2021).
- [BHX20] M. Belkin, D. Hsu and J. Xu. "Two models of double descent for weak features". In: *SIAM Journal on Mathematics of Data Science* 2.4 (2020), pp. 1167–1180.
- [Bis+95] C. M. Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [BKH16] J. L. Ba, J. R. Kiros and G. E. Hinton. *Layer normalization*. Tech. rep. 2016.
- [BKK20] S. Bai, V. Koltun and J. Z. Kolter. "Multiscale deep equilibrium models". In: *Advances in Neural Information Processing Systems*. Vol. 2020-Decem. 2020.
- [BKZ04a] M. A. Bernstein, K. F. King and X. J. Zhou. "Advanced pulse sequence techniques". In: *Handbook of MRI Pulse Sequences*. Elsevier, 2004, pp. 802–954.
- [BKZ04b] M. A. Bernstein, K. F. King and X. J. Zhou. "Signal acquisition and k-space sampling". In: *Handbook of MRI Pulse Sequences*. Elsevier, 2004, pp. 367–442.



- [Blo+14] K. T. Block, H. Chandarana, S. Milla, M. Bruno, T. Mulholland, G. Fatterpekar, M. Hagiwara, R. Grimm, C. Geppert, B. Kiefer and D. K. Sodickson. "Towards Routine Clinical Use of Radial Stack-of-Stars 3D Gradient-Echo Sequences for Reducing Motion Sensitivity". In: *Journal of the Korean Society of Magnetic Resonance in Medicine* 18.2 (2014), p. 87.
- [BLS19] V. Böhm, F. Lanusse and U. Seljak. "Uncertainty Quantification with Generative Models". In: *NeurIPS workshop on Bayesian Deep Learning*. Oct. 2019.
- [BN89] R. H. Byrd and J. Nocedal. "A tool for the analysis of quasi-Newton methods with application to unconstrained minimization". English. In: *SIAM Journal on Numerical Analysis* 26.3 (1989), pp. 727–739.
- [BNP05] P. J. Beatty, D. G. Nishimura and J. M. Pauly. "Rapid gridding reconstruction with a minimal oversampling ratio". In: *IEEE Transactions on Medical Imaging* 24.6 (2005), pp. 799–808.
- [Bor+17] A. Bora, A. Jalal, E. Price and A. G. Dimakis. "Compressed sensing using generative models". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 537–546.
- [Boy+11] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (Jan. 2011), pp. 1–122.
- [Boy+16] C. Boyer, N. Chauffert, P. Ciuciu, J. Kahn and P. Weiss. "On the generation of sampling schemes for magnetic resonance imaging". In: *SIAM Journal on Imaging Sciences* 9.4 (2016), pp. 2039–2072.
- [BPM91] C. J. Bergin, J. M. Pauly and A. Macovski. "Lung parenchyma: Projection reconstruction MR imaging". In: *Radiology* 179.3 (1991), pp. 777–781.
- [Bro+14] R. W. Brown, Y.-C. N. Cheng, E. M. Haacke, M. R. Thompson and R. Venkatesan. *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons, 2014.
- [Bro+21] A. Brock, S. De, S. L. Smith and K. Simonyan. *High-performance large-scale image recognition without normalization*. Tech. rep. 2021.
- [Bro65] C. G. Broyden. "A Class of Methods for Solving Nonlinear Simultaneous Equations". In: *Mathematics of Computation* 19.92 (1965), pp. 577–593.

- [BSS15] J. H. O. Barbosa, A. C. Santos and C. E. G. Salmon. "Susceptibility weighted imaging: differentiating between calcification and hemosiderin". In: *Radiol Bras* 48.2 (2015), pp. 93–100.
- [BSS88] R. H. Byrd, R. B. Schnabel and G. A. Shultz. "Parallel quasi-Newton methods for unconstrained optimization". English. In: *Mathematical Programming. Series A. Series B* 42.2 (B) (1988), pp. 273–306.
- [BT09] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm". In: *Society for Industrial and Applied Mathematics Journal on Imaging Sciences* 2.1 (2009), pp. 183–202.
- [Bus+11] J. T. Bushberg, J. A. Seibert, E. M. J. Leidholt and J. M. Boone. *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2011.
- [BZK19] S. Bai, J. Zico Kolter and V. Koltun. "Deep equilibrium models". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.
- [Cab+14] J. Caballero, A. N. Price, D. Rueckert and J. V. Hajnal. "Dictionary learning and time sparsity for dynamic MR data reconstruction". In: *IEEE Transactions on Medical Imaging* 33.4 (2014), pp. 979–994.
- [CAT15] I. Y. Chun, B. Adcock and T. M. Talavage. "Efficient compressed sensing SENSE pMRI reconstruction with joint sparsity promotion". In: *IEEE transactions on Medical Imaging* 35.1 (2015), pp. 354–368.
- [CCW13] N. Chauffert, P. Ciuciu and P. Weiss. "Variable density compressed sensing in MRI. Theoretical vs heuristic sampling strategies". In: *2013 IEEE 10th International Symposium on Biomedical Imaging*. IEEE. 2013, pp. 298–301.
- [CGT91] A. R. Conn, N. I. Gould and P. L. Toint. "Convergence of quasi-Newton matrices generated by the symmetric rank one update". In: *Mathematical Programming* 50.1-3 (1991), pp. 177–195.
- [Cha+14] N. Chauffert, P. Ciuciu, J. Kahn and P. Weiss. "Variable density sampling with continuous trajectories". In: *SIAM Journal on Imaging Sciences* 7.4 (2014), pp. 1962–1992.

- [Cha+21] G. Chaithya, P. Weiss, G. Daval-Fr erot, A. Massire, A. Vignaud and P. Ciuciu. "Optimizing full 3D SPARKLING trajectories for high-resolution T2\*-weighted Magnetic Resonance Imaging". Nov. 2021. Under review in IEEE Transactions on Medical Imaging.
- [Che+16] T. Chen, B. Xu, C. Zhang and C. Guestrin. *Training Deep Nets with Sublinear Memory Cost*. Tech. rep. 2016, pp. 1–12.
- [Che+18a] R. T. Chen, Y. Rubanova, J. Bettencourt and D. Duvenaud. "Neural Ordinary differential equations". In: *NeurIPS*. 2018.
- [Che+18b] H. Cherkaoui, L. El Gueddari, C. Lazarus, A. Grigis, F. Poupon, A. Vignaud, S. Farrens, J.-L. Starck and P. Ciuciu. "Analysis vs synthesis-based regularization for combined compressed sensing and parallel MRI reconstruction at 7 tesla". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE. 2018, pp. 36–40.
- [Che+19] J. Cheng, H. Wang, L. Ying and D. Liang. "Model learning: Primal dual networks for fast MR imaging". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 21–29.
- [Chi+16] G. Chierchia, E. Chouzenoux, P. L. Combettes and J.-C. Pesquet. *The Proximity Operator Repository. User's guide*. <http://proximity-operator.net/>. Accessed: 2021-10-21. 2016.
- [Cho+15] F. Chollet et al. *Keras*. [keras.io](https://keras.io). 2015.
- [Con13] L. Condat. "A Primal-Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms". In: *Journal of Optimization Theory and Applications* 158.2 (2013), pp. 460–479.
- [Cor+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele. "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.
- [CP11] A. Chambolle and T. Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". In: *Journal of Mathematical Imaging and Vision* 40 (2011), pp. 120–145.

- [CPS06] K. Chellapilla, S. Puri and P. Simard. "High performance convolutional neural networks for document processing". In: *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft. 2006.
- [CRT06] E. J. Candès, J. Romberg and T. Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transactions on Information Theory* 52.2 (2006), pp. 489–509.
- [CWBo8] E. J. Candès, M. B. Wakin and S. P. Boyd. "Enhancing sparsity by reweighted l1 minimization". In: *Journal of Fourier analysis and applications* 14.5-6 (2008), pp. 877–905.
- [CY21] H. Chung and J. C. Ye. *Score-based diffusion models for accelerated MRI*. Tech. rep. 2021.
- [CZC21] G. Chaithya, **Zaccharie Ramzi** and P. Ciuciu. "Hybrid learning of Non-Cartesian k-space trajectory and MR image reconstruction networks". 2021.
- [Dab+06] K. Dabov, A. Foi, V. Katkounnik and K. Egiazarian. "Image denoising with block-matching and 3D filtering". In: *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*. Vol. 6064. International Society for Optics and Photonics. 2006, p. 606414.
- [Dar+21] G. Daras, J. Dean, A. Jalal and A. G. Dimakis. *Intermediate layer optimization for inverse problems using deep generative models*. Tech. rep. 2021.
- [DDD04] I. Daubechies, M. Defrise and C. De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Communications on Pure and Applied Mathematics* 57.11 (2004), pp. 1413–1457.
- [Den+09] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *CVPR*. Institute of Electrical and Electronics Engineers (IEEE), 2009.
- [Den+20] A. Denker, M. Schmidt, J. Leuschner, P. Maass and J. Behrmann. "Conditional Normalizing Flows for Low-Dose Computed Tomography Image Reconstruction". In: *ICML*. June 2020.
- [DG17] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017.
- [DH20] M. Z. Darestani and R. Heckel. "Accelerated MRI with Untrained Neural Networks". 2020.

- [Dono6] D. L. Donoho. "Compressed sensing". In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306.
- [Dong95] D. L. Donoho. "De-noising by soft-thresholding". In: *IEEE transactions on information theory* 41.3 (1995), pp. 613–627.
- [Dra+17] P. L. Dragotti, H. Dong, G. Yang, Y. Guo, D. Firmin, G. Slabaugh, S. Yu, J. Keegan, X. Ye, F. Liu and S. Arridge. "DAGAN: Deep De-Aliasing Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction". In: *IEEE Transactions on Medical Imaging* 37.6 (2017), pp. 1310–1321.
- [EB01] A. D. Elster and J. H. Burdette. *Questions and answers in magnetic resonance imaging*. Mozby, 2001.
- [Edu+20] V. Edupuganti, M. Mardani, S. Vasanawala and J. Pauly. "Uncertainty Quantification in Deep MRI Reconstruction". In: *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*. 2020.
- [El +18] L. El Gueddari, C. Lazarus, H. Carrie, A. Vignaud and P. Ciuciu. "Self-Calibrating Nonlinear Reconstruction Algorithms for Variable Density Sampling and Parallel Reception MRI". In: *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop*. 2018, pp. 415–419.
- [El +21] L. El Gueddari, C. Giliyar Radhakrishna, E. Chouzenoux and P. Ciuciu. "Calibration-Less Multi-Coil Compressed Sensing Magnetic Resonance Image Reconstruction Based on OSCAR Regularization". In: *Journal of Imaging* 7.3 (2021), p. 58.
- [El 19] L. El Gueddari. "Proximal structured sparsity regularization for online reconstruction in high-resolution accelerated Magnetic Resonance". PhD thesis. 2019.
- [EMR07] M. Elad, P. Milanfar and R. Rubinstein. "Analysis versus synthesis in signal priors". In: *Inverse problems* 23.3 (2007), p. 947.
- [Eo+18] T. Eo, Y. Jun, T. Kim, J. Jang, H. J. Lee and D. Hwang. "KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance images". In: *Magnetic Resonance in Medicine* 80.5 (2018), pp. 2188–2201.
- [Epp+13] K. Epperson, A. M. Sawyer, M. Lustig, M. Alley, M. Uecker, P. Virtue, P. Lai and S. Vasanawala. "Creation of Fully Sampled MR Data Repository for Compressed Sensing of the Knee Purpose". In: *Proceedings of the 22nd Annual Meeting for Section for Magnetic Resonance Technologists*. 2013.

- [Fan+20] F. Fan, M. Li, Y. Teng and G. Wang. "Soft Autoencoder and Its Wavelet Adaptation Interpretation". In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 1245–1257.
- [Fan11] R.-E. Fan. *LIBSVM datasets*. [csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/](http://csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/). Accessed: 2021-05-06. 2011.
- [Far+20] S. Farrens, A. Grigis, L. El Gueddari, **Zaccharie Ramzi**, C. G R, S. Starck, B. Sarthou, H. Cherkaoui, P. Ciuciu and J.-L. Starck. "PySAP: Python Sparse Data Analysis Package for multidisciplinary image processing". In: *Astronomy and Computing* 32 (2020).
- [FBS93] H. Fayez Khalfan, R. H. Byrd and R. B. Schnabel. "A theoretical and experimental study of the symmetric rank-one update". English. In: *SIAM Journal on Optimization* 3.1 (1993), pp. 1–24.
- [Fen+14] L. Feng, R. Grimm, K. T. Block, H. Chandarana, S. Kim, J. Xu, L. Axel, D. K. Sodickson and R. Otazo. "Golden-angle radial sparse parallel MRI: combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric MRI". In: *Magnetic resonance in medicine* 72.3 (2014), pp. 707–717.
- [Fes20] J. A. Fessler. "Optimization methods for image reconstruction problems". In: *IEEE Signal Processing Magazine* 37.1 (2020), pp. 33–40.
- [FHS21] Z. Fabian, R. Heckel and M. Soltanolkotabi. "Data augmentation for deep learning based accelerated MRI reconstruction with limited data". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3057–3067.
- [FR13a] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. New York, NY: Birkhäuser, 2013.
- [FR13b] S. Foucart and H. Rauhut. "Coherence". In: *A Mathematical Introduction to Compressive Sensing*. 9780817649470. Birkhäuser, 2013, pp. 111–131.
- [FR13c] S. Foucart and H. Rauhut. "Sparse Solutions of Underdetermined Systems". In: *A Mathematical Introduction to Compressive Sensing*. Birkhauser, 2013. Chap. 2, pp. 41–59.
- [FS03] J. A. Fessler and B. P. Sutton. "Nonuniform Fast Fourier Transforms Using Min-Max Interpolation". In: *IEEE Transactions on Signal Processing* 51.2 (2003), pp. 560–74.

- [Fun+21] S. W. Fung, H. Heaton, Q. Li, D. Mckenzie, S. Osher and W. Yin. *Fixed Point Networks: Implicit Depth Models with Jacobian-Free Backprop*. Tech. rep. 2021.
- [Gei+13] A. Geiger, P. Lenz, C. Stiller and R. Urtasun. "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.
- [Gei+21] J. Geiping, M. Goldblum, P. E. Pope, M. Moeller and T. Goldstein. *Stochastic Training is Not Necessary for Generalization*. 2021.
- [GL10] K. Gregor and Y. LeCun. "Learning fast approximations of sparse coding". In: *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*. 2010, pp. 399–406.
- [Gom+17] A. N. Gomez, M. Ren, R. Urtasun and R. B. Grosse. "The Reversible Residual Network: Backpropagation Without Storing Activations". In: *NIPS*. 2017.
- [Goo+14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Mirza, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio. "Generative Adversarial Nets". In: *Advances in neural information processing systems*. 2014.
- [Got+20] N. M. Gottschling, V. Antun, B. Adcock and A. C. Hansen. "The troublesome kernel: why deep learning for inverse problems is typically unstable". 2020.
- [GOW19] D. Gilton, G. Ongie and R. Willett. "Neumann Networks for Linear Inverse Problems in Imaging". In: *IEEE Transactions on Computational Imaging* 6 (2019), pp. 328–343.
- [GOW21] D. Gilton, G. Ongie and R. Willett. "Deep Equilibrium Architectures for Inverse Problems in Imaging". 2021.
- [GP21] J. D. Garrett and H.-H. Peng. *garrettj403SciencePlots*. Version 1.0.7. Feb. 2021.
- [GP92] G. H. Glover and J. M. Pauly. "Projection reconstruction techniques for reduction of motion effects in MRI". In: *Magnetic resonance in medicine* 28.2 (1992), pp. 275–289.
- [GR17] R. M. Gower and P. Richtárik. "Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms". In: *SIAM Journal on Matrix Analysis and Applications* 38.4 (2017), pp. 1380–1409.

- [Gri+02] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer and A. Haase. "Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA)". In: *Magnetic Resonance in Medicine* 47.6 (June 2002), pp. 1202–1210.
- [Gu+20] J. Gu, H. Cai, H. Chen, X. Ye, J. S. Ren, C. Dong and X. Ye. "Image Quality Assessment for Perceptual Image Restoration: A New Dataset, Benchmark and Metric". 2020.
- [GZC21] C. G R, **Zaccharie Ramzi** and P. Ciuciu. "Learning the sampling density in 2D SPARKLING MRI acquisition for optimized image reconstruction". In: *European Signal Processing Conference*. 2021.
- [Ham+18] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock and F. Knoll. "Learning a variational network for reconstruction of accelerated MRI data". In: *Magnetic Resonance in Medicine* 79.6 (2018), pp. 3055–3071.
- [Ham+19] K. Hammernik, J. Schlemper, C. Qin, J. Duan, R. M. Summers and D. Rueckert.  $\Sigma$ -net: Systematic Evaluation of Iterative Deep Neural Networks for Fast Parallel MR Image Reconstruction. Tech. rep. 2019.
- [Han+18] Y. Han, J. Yoo, H. H. Kim, H. J. Shin, K. Sung and J. C. Ye. "Deep learning with domain adaptation for accelerated projection-reconstruction MR". In: *Magnetic resonance in medicine* 80.3 (2018), pp. 1189–1205.
- [Han19] B. Hanin. "Universal function approximation by deep neural nets with bounded width and relu activations". In: *Mathematics* 7.10 (2019), p. 992.
- [Har+14] A. Harati, S. Lopez, I. Obeid, J. Picone, M. Jacobson and S. Tobochnik. "The TUH EEG CORPUS: A big data resource for automated EEG interpretation". In: *2014 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. IEEE. 2014, pp. 1–5.
- [Har+20] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362.



- [He+15] K. He, X. Zhang, S. Ren and J. Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *ICCV*. 2015.
- [He+16] K. He, X. Zhang, S. Ren and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-Decem. 2016, pp. 770–778.
- [HG16] D. Hendrycks and K. Gimpel. *Gaussian error linear units (gelus)*. Tech. rep. 2016.
- [Hof+13] M. D. Hoffman, D. M. Blei, C. Wang and J. Paisley. "Stochastic variational inference". In: *Journal of Machine Learning Research* 14 (2013), pp. 1303–1347.
- [Hol+14] S. K. Holland, M. Altaye, S. Robertson, A. W. Byars, E. Plante and J. P. Szaflarski. "Data on the safety of repeated MRI in healthy children". In: *NeuroImage: Clinical* 4 (2014), pp. 526–530.
- [Hor+07] K. A. Horvath, M. Li, D. Mazilu, M. A. Guttman and E. R. McVeigh. "Real-time Magnetic Resonance Imaging Guidance for Cardiovascular Procedures". In: *Seminars in thoracic and cardiovascular surgery*. Vol. 19. 2007, pp. 330–335.
- [HS13] M. S. Hansen and T. S. Sørensen. "Gadgetron: an open source framework for medical image reconstruction". In: *Magnetic resonance in medicine* 69.6 (2013), pp. 1768–1776.
- [HS97] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [HSS12] G. Hinton, N. Srivastava and K. Swersky. *Neural networks for machine learning lecture 6a overview of mini-batch gradient descent*. Tech. rep. 2012.
- [HSW89] K. Hornik, M. Stinchcombe and H. White. "Multilayer feed-forward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.
- [HSY19] Y. Han, L. Sunwoo and J. C. Ye. "k-space deep learning for accelerated MRI". In: *IEEE transactions on medical imaging* (2019).
- [Hu+21] C. Hu, C. Li, H. Wang, Q. Liu, H. Zheng and S. Wang. "Self-supervised Learning for MRI Reconstruction with a Parallel Network Training Framework Self-supervised MRI Reconstruction 383". In: *MICCAI*. Vol. 12906. 2021, pp. 382–391.

- [Hua+20] W. R. Huang, Z. Emam, M. Goldblum, L. Fowl, J. K. Terry, F. Huang and T. Goldstein. "Understanding Generalization Through Visualizations". In: *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*. Ed. by J. Zosa Forde, F. Ruiz, M. F. Pradier and A. Schein. Vol. 137. Proceedings of Machine Learning Research. PMLR, Dec. 2020, pp. 87–97.
- [Hun07] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science and Engineering* 9.3 (2007), pp. 90–95.
- [Hyu+18] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee and J. K. Seo. "Deep learning for undersampled MRI reconstruction". In: *Physics in Medicine & Biology* 63.13 (June 2018), p. 135007.
- [Hyv05] A. Hyvärinen. "Estimation of Non-Normalized Statistical Models by Score Matching". In: *Journal of Machine Learning Research* 6 (2005), pp. 695–708.
- [HZ10] A. Horé and D. Ziou. "Image quality metrics: PSNR vs. SSIM". In: *Proceedings - International Conference on Pattern Recognition*. 2010, pp. 2366–2369.
- [IN95] P. Irarrazabal and D. G. Nishimura. "Fast Three Dimensional Magnetic Resonance Imaging". In: *Magnetic Resonance in Medicine* 33.5 (1995), pp. 656–662.
- [IS15] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.
- [Iva71] A. G. Ivakhnenko. "Polynomial theory of complex systems". In: *IEEE transactions on Systems, Man, and Cybernetics* 4 (1971), pp. 364–378.
- [JKS19] D. Jawali, A. Kumar and See. "A Learning Approach for Wavelet Design". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2019, pp. 5018–5022.
- [Jol+21] A. Jolicoeur-Martineau, R. Piché-Taillefer, R. T. des Combes and I. Mitliagkas. "Adversarial score matching and improved sampling for image generation". In: *ICLR*. 2021.
- [Jou+17] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers et al. "In-datacenter performance analysis of a tensor processing unit". In: *Proceedings of the 44th annual international symposium on computer architecture*. 2017, pp. 1–12.

- [Jum+21] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589.
- [Kar+20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila. “Analyzing and Improving the Image Quality of StyleGAN”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8110–8119.
- [Kar19] A. Karpathy. *t-SNE visualization of CNN codes*. <https://cs.stanford.edu/people/karpathy/> Accessed: 2021-11-15. 2019.
- [KB15] D. P. Kingma and J. L. Ba. “Adam: A method for stochastic optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, Dec. 2015.
- [Kel+20] M. Kellman, K. Zhang, J. Tamir, E. Bostan, M. Lustig and L. Waller. “Memory-efficient learning for large-scale computational imaging”. In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 1403–1414.
- [KF18] D. Kim and J. A. Fessler. “Adaptive Restart of the Optimized Gradient Method for Convex Optimization”. In: *Journal of Optimization Theory and Applications* 178.1 (2018), pp. 240–263.
- [Kid+04] C. S. Kidwell, J. A. Chalela, J. L. Saver, S. Starkman, M. D. Hill, A. M. Demchuk, J. A. Butman, N. Patronas, J. R. Alger, L. L. Latour, M. L. Luby, A. E. Baird, M. C. Leary, M. Tremwel, B. Ovbiagele, A. Fredieu, S. Suzuki, J. P. Villablanca, S. Davis, B. Dunn, J. W. Todd, M. A. Ezzeddine, J. Haymore, J. K. Lynch, L. Davis and S. Warach. “Comparison of MRI and CT for Detection of Acute Intracerebral Hemorrhage”. In: *Journal of the American Medical Association* 292.15 (Oct. 2004), pp. 1823–1830.
- [KM00] J. Kalita and U. K. Misra. “Comparison of CT scan and MRI findings in the diagnosis of Japanese encephalitis”. In: *Journal of the Neurological Sciences* 174.1 (Mar. 2000), pp. 3–8.
- [Kny+21] B. Knyazev, M. Drozdal, G. W. Taylor and A. Romero-Soriano. “Parameter Prediction for Unseen Deep Architectures”. In: (2021).

- [Kob+20] E. Kobler, A. Effland, K. Kunisch and T. Pock. "Total deep variation for linear inverse problems". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7549–7558.
- [KP13] S. G. Krantz and H. R. Parks. *The Implicit Function Theorem: History, Theory, and Applications*. Springer New York, Jan. 2013, pp. 1–163.
- [Kri09] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Tech. rep. 2009.
- [KS18] H. Koyasu and K. Shiba. "KNC can scan three more patients per day with Compressed SENSE". In: *FieldStrength* (Aug. 2018).
- [KSH12] A. Krizhevsky, I. Sutskever and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.
- [Küs+20] T. Küstner, N. Fuin, K. Hammernik, A. Bustin, H. Qi, R. Hajhosseiny, P. G. Masci, R. Neji, D. Rueckert, R. M. Botnar and C. Prieto. "CINENet: deep learning-based 3D cardiac CINE MRI reconstruction with multi-coil complex-valued 4D spatio-temporal convolutions". In: *Scientific Reports* 10.1 (2020), pp. 1–13.
- [KW14] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. ML. 2014, pp. 1–14.
- [LaM+18] P. J. LaMontagne, S. Keefe, W. Lauren, C. Xiong, E. A. Grant, K. L. Moulder, J. C. Morris, T. L. Benzinger and D. S. Marcus. "OASIS-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer's disease". In: *Alzheimer's and Dementia: The Journal of the Alzheimer's Association* 14.7 (2018), P1097.
- [Lang95] K. Lang. "NewsWeeder: Learning to Filter Netnews". In: *ICML*. Elsevier, 1995, pp. 331–339.
- [Laz+19] C. Lazarus, P. Weiss, N. Chauffert, F. Mauconduit, L. El Gueddari, C. Destrieux, I. Zemmoura, A. Vignaud and P. Ciuciu. "SPARKLING: variable-density k-space filling curves for accelerated T2\*-weighted MRI". In: *Magnetic resonance in medicine* 81.6 (2019), pp. 3643–3661.

- [Laz+20] C. Lazarus, P. Weiss, L. El Gueddari, F. Mauconduit, A. Massire, M. Ripart, A. Vignaud and P. Ciuciu. "3D variable-density SPARKLING trajectories for high-resolution T2\*-weighted magnetic resonance imaging". In: *NMR in Biomedicine* 33.9 (2020), e4349.
- [Laz18] C. Lazarus. "Compressed Sensing in MRI : optimization-based design of k-space filling curves for accelerated MRI". PhD thesis. 2018.
- [LCY13] M. Lin, Q. Chen and S. Yan. *Network in network*. Tech. rep. 2013.
- [LDP07] M. Lustig, D. Donoho and J. M. Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging". In: *Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195.
- [Le +21] C. Le Ster, F. Mauconduit, C. Mirkes, M. Bottlaender, F. Boumezbeur, B. Djemai, A. Vignaud and N. Boulant. "Radiofrequency heating measurement in vivo using MR thermometry and field monitoring: methodological considerations". In: *Magnetic Resonance in Medecine* 85.3 (Mar. 2021).
- [LeC+12] Y. A. LeCun, L. Bottou, G. B. Orr and K.-R. Müller. "Efficient backprop". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.
- [LeC+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.
- [Lee+18] D. Lee, J. Yoo, S. Tak and J. C. Ye. "Deep residual learning for accelerated MRI using magnitude and phase networks". In: *IEEE Transactions on Biomedical Engineering* 65.9 (2018), pp. 1985–1995.
- [Lef18] S. Lefkimmiatis. "Universal denoising networks: a novel CNN architecture for image denoising". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3204–3213.
- [LFoo] D. Li and M. Fukushima. "A derivative-free line search and global convergence of Broyden-like method for nonlinear equations". English. In: *Optimization Methods & Software* 13.3 (2000), pp. 181–201.

- [Lim+20] J. H. Lim, A. Courville, C. Pal and C.-W. Huang. "AR-DAE: Towards Unbiased Neural Entropy Gradient Estimation". In: *ICML*. 2020.
- [Lin+14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.
- [Liu+15] Z. Liu, P. Luo, X. Wang and X. Tang. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.
- [Liu+18] P. Liu, H. Zhang, K. Zhang, L. Lin and W. Zuo. "Multi-level Wavelet-CNN for Image Restoration". In: *CVPR NTIRE Workshop*. 2018.
- [Liu+20] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao and J. Han. "On the Variance of the Adaptive Learning Rate and Beyond". In: *Proceedings of International Conference for Learning Representations*. 2020, pp. 1–3.
- [LK15] F.-F. Li and A. Karpathy. *CS231n Convolutional Neural Networks for Visual Recognition*. <https://cs231n.github.io/convolutional-networks>. Accessed: 2021-11-16. 2015.
- [Llo82] S. P. Lloyd. "Least Squares Quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137.
- [LN89] D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming, Series B* 45.3 (1989), pp. 503–528.
- [LP10] M. Lustig and J. M. Pauly. "SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space". In: *Magnetic Resonance in Medicine* 64.2 (2010), pp. 457–471.
- [LS18] J. Liang and C.-B. Schönlieb. *Improving FISTA: Faster, Smarter and Greedier*. Tech. rep. 2018.
- [Lu+21] T. Lu, T. Marin, Y. Zhuo, Y.-f. Chen and C. Ma. "NONUNIFORM FAST FOURIER TRANSFORM ON TPUS". In: *ISBI*. 2021, pp. 783–787.
- [LVD20] J. Lorraine, P. Vicol and D. Duvenaud. "Optimizing Millions of Hyperparameters by Implicit Differentiation". In: *AISTATS*. 2020.
- [LZZ98] D. Li, J. Zeng and S. Zhou. "Convergence of Broyden-Like Matrix". In: *Applied Mathematics Letter* 11.5 (1998), pp. 35–37.

- [Maa+11] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts. "Learning word vectors for sentiment analysis". In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 2011, pp. 142–150.
- [Mal+20] M. O. Malavé, C. A. Baron, S. P. Koundinyan, C. M. Sandino, F. Ong, J. Y. Cheng and D. G. Nishimura. "Reconstruction of undersampled 3D non-Cartesian image-based navigators for coronary MRA using an unrolled deep learning model". In: *Magnetic Resonance in Medicine* 84.2 (Aug. 2020), pp. 800–812.
- [Mal99] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.
- [Man20] F. Mannel. "On the convergence of the Broyden-like matrices". 2020.
- [Man21a] F. Mannel. "Convergence properties of the Broyden-like method for mixed linear–nonlinear systems of equations". In: *Numerical Algorithms* (2021), pp. 1–29.
- [Man21b] F. Mannel. "On the convergence of Broyden's method and some accelerated schemes for singular problems". 2021.
- [Mar+01] D. Martin, C. Fowlkes, D. Tal and J. Malik. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics". In: *Proc. 8th Int'l Conf. Computer Vision*. Vol. 2. 2001, pp. 416–423.
- [Mar+16] L. Marrakchi-Kacem, A. Vignaud, J. Sein, J. Germain, T. R. Henry, C. Poupon, L. Hertz-Pannier, S. Lehéricy, O. Colliot, P. F. Van de Moortele and M. Chupin. "Robust imaging of hippocampal inner structure at 7T: in vivo acquisition protocol and methodological choices". In: *Magnetic Resonance Materials in Physics, Biology and Medicine* 29.3 (2016), pp. 475–489.
- [Mar+19] H. Maron, E. Fetaya, N. Segol and Y. Lipman. "On the universality of invariant networks". In: *International conference on machine learning*. PMLR. 2019, pp. 4363–4371.
- [Mer+16] S. Merity, C. Xiong, J. Bradbury and R. Socher. *Pointer sentinel mixture models*. Tech. rep. 2016.
- [Mey+92] C. H. Meyer, B. S. Hu, D. G. Nishimura and A. Macovski. "Fast spiral coronary artery imaging". In: *Magnetic resonance in medicine* 28.2 (1992), pp. 202–213.

- [MHN13] A. L. Maas, A. Y. Hannun and A. Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. 2013, p. 3.
- [Mie+19] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev and J. Sivic. "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2630–2640.
- [Mie+21] J. C. Mier, E. Huang, H. Talebi, F. Yang and P. Milanfar. "Deep perceptual image quality assessment for compression". In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 1484–1488.
- [Mik21] A. Mikhailiuk. *Deep Image Quality Assessment*. <https://towardsdatascience.com/deep-image-quality-assessment-30ad71641fac>. Accessed: 2021-10-18. 2021.
- [Mil+20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi and R. Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *European conference on computer vision*. Springer. 2020, pp. 405–421.
- [MMC17] T. Meinhardt, M. Moeller and D. Cremers. "Learning Proximal Operators : Using Denoising Networks for Regularizing Inverse Imaging Problems". In: *ICCV*. 2017, pp. 1781–1790.
- [MNJ18] T. Minh Quan, T. Nguyen-Duc and W.-K. Jeong. "Compressed Sensing MRI Reconstruction using a Generative Adversarial Network with a Cyclic Loss". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1488–1497.
- [Mor62] J. J. Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences*. 1962, pp. 2897–2899.
- [MT76] J. More and J. Trangenstein. "On the global convergence of Broyden's method". English. In: *Mathematics of Computation* 30 (1976), pp. 523–540.
- [Muc+20] M. J. Muckley, R. Stern, T. Murrell and F. Knoll. "TorchKbNufft: A High-Level, Hardware-Agnostic Non-Uniform Fast Fourier Transform". In: *ISMRM Workshop on Data Sampling & Image Reconstruction*. 2020.



- [Muc+21] M. J. Muckley, B. Riemenschneider, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha, S. Arberet, D. Nickel, **Zaccharie Ramzi**, P. Ciuciu, J. L. Starck, J. Teuwen, D. Karkalousos, C. Zhang, A. Sriram, Z. Huang, N. Yakubova, Y. W. Lui and F. Knoll. "Results of the 2020 fastMRI Challenge for Machine Learning MR Image Reconstruction". In: *IEEE Transactions on Medical Imaging* 40.9 (2021), pp. 2306–2317.
- [NA20] J. Nilsson and T. Akenine-möller. *Understanding SSIM*. Tech. rep. 2020.
- [Nak+17] U. Nakarmi, K. Slavakis, J. Lyu and L. Ying. "M-MRI: A manifold-based framework to highly accelerated dynamic magnetic resonance imaging". In: *Proceedings - International Symposium on Biomedical Imaging* (2017), pp. 19–22.
- [Nak+20] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak and I. Sutskever. "Deep double descent: Where bigger models and more data hurt". In: *ICLR*. 2020.
- [Nea11] R. M. Neal. "MCMC using hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* (2011), pp. 113–162.
- [Nes83] Y. E. Nesterov. "A method for solving the convex programming problem with convergence rate  $O(1/k^2)$ ". In: *Dokl. Akad. Nauk SSSR* 269 (1983), pp. 543–547.
- [NH10a] F. B. Nahab and M. Hallett. "Current Role of fMRI in diagnosis of movement disorders". In: *Neuroimaging Clinics of North America* 20.1 (2010), p. 103.
- [NH10b] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *icml*. 2010.
- [NIM95] D. G. Nishimura, P. Irarrazabal and C. H. Meyer. "A velocity k-space analysis of flow effects in echo-planar and spiral imaging". In: *Magnetic resonance in medicine* 33.4 (1995), pp. 549–556.
- [Now18] A. Nowogrodzki. "The world's strongest MRI machines are pushing human imaging to new limits". In: *Nature* 563 (Nov. 2018), pp. 24–26.
- [NW06] J. Nocedal and S. Wright. "Quasi-Newton Methods". In: *Numerical Optimization*. 2006, pp. 135–163.
- [Nwa+18] C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall. "Activation functions: Comparison of trends in practice and research for deep learning". In: (2018).

- [Ogb+18] G. I. Ogbole, A. O. Adeyomoye, A. Badu-Pepurah, Y. Mensah and D. A. Nzeh. "Survey of magnetic resonance imaging availability in West Africa". In: *PanAfrican Medical Journal* 30.1 (2018).
- [Ong+20] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis and R. Willett. "Deep learning techniques for inverse problems in imaging". In: *IEEE Journal on Selected Areas in Information Theory* 1.1 (2020), pp. 39–56.
- [OUL20] F. Ong, M. Uecker and M. Lustig. "Accelerating Non-Cartesian MRI Reconstruction Convergence Using k-Space Preconditioning". In: *IEEE Transactions on Medical Imaging* 39.5 (2020), pp. 1646–1654.
- [Pas+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimsheine, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *NeurIPS*. Dec. 2019.
- [Pat+88] D. W. Paty, J. J. F. Oger, L. F. Kastrukoff, S. A. Hashimoto, J. P. Hooge, A. A. Eisen, K. A. Eisen, S. J. Purves, M. D. Low, V. Brandejs, W. D. Robertson and D. K. B. Li. "MRI in the diagnosis of MS". In: *Neurology* 38.2 (Feb. 1988).
- [PB19] L. Pfister and Y. Bresler. "Learning Filter Bank Sparsifying Transforms". In: *IEEE Transactions on Signal Processing* 67.2 (2019), pp. 504–519.
- [Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [Ped16] F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: *33rd International Conference on Machine Learning, ICML 2016* 2 (2016), pp. 1150–1159.
- [Pet+00] D. C. Peters, F. R. Korosec, T. M. Grist, W. F. Block, J. E. Holden, K. K. Vigen and C. A. Mistretta. "Undersampled projection reconstruction applied to MR angiography". In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 43.1 (2000), pp. 91–101.

- [Pet+10] R. C. Petersen, P. Aisen, L. A. Beckett, M. Donohue, A. Gamst, D. J. Harvey, C. Jack, W. Jagust, L. Shaw, A. Toga et al. "Alzheimer's disease neuroimaging initiative (ADNI): clinical characterization". In: *Neurology* 74.3 (2010), pp. 201–209.
- [Pez+20] N. Pezzotti, S. Yousefi, M. S. Elmahdy, J. H. F. van Gemert, C. Schuelke, M. Doneva, T. Nielsen, S. Kastrulin, B. P. Lelieveldt, M. J. van Osch, E. D. Weerdt and M. Staring. "An adaptive intelligence algorithm for undersampled knee MRI reconstruction". In: *IEEE Access* 8 (2020), pp. 204825–204838.
- [PJ16] S. Poddar and M. Jacob. "Dynamic MRI Using Smoothness Regularization on Manifolds (SToRM)". In: *IEEE Transactions on Medical Imaging* 35.4 (2016), pp. 1106–1115.
- [PM99] J. G. Pipe and P. Menon. "Sampling density compensation in MRI: Rationale and an iterative numerical solution". In: *Magnetic Resonance in Medicine* 41.1 (1999), pp. 179–186.
- [PMB13] R. Pascanu, T. Mikolov and Y. Bengio. "On the difficulty of training recurrent neural networks". In: *ICML*. 2013.
- [Pod+19] S. Poddar, Y. Q. Mohsin, D. Ansah, B. Thattaliyath, R. Ashwath and M. Jacob. "Manifold Recovery Using Kernel Low-Rank Regularization: Application to Dynamic Imaging". In: *IEEE Transactions on Computational Imaging* 5.3 (2019), pp. 478–491.
- [Poo+21] K. Pooja, **Zaccharie Ramzi**, C. G R and P. Ciuciu. "MC-PDNET: Deep unrolled neural network for multi-contrast mr image reconstruction from undersampled k-space data". Oct. 2021.
- [Pow+21] A. Power, Y. Burda, H. Edwards, I. Babuschkin and V. Misra. "Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets". In: *ICLR MATH-AI Workshop*. 2021.
- [Pra+18] E. Prashnani, H. Cai, Y. Mostofi and P. Sen. "PieAPP: Perceptual Image-Error Assessment Through Pairwise Preference". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1808–1817.
- [Pru+99] K. P. Pruessmann, M. Weiger, M. B. Scheidegger and P. Boesiger. "SENSE: Sensitivity encoding for fast MRI". In: *Magnetic Resonance in Medicine* 42.5 (Nov. 1999), pp. 952–962.

- [Pui+16] C. W. Pui, G. Chen, W. K. Chow, K. C. Lam, J. Kuang, P. Tu, H. Zhang, E. F. Young and B. Yu. "RippleFPGA: A routability-driven placement for large-scale heterogeneous FPGAs". In: *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*. Vol. 07-10-Nove. 2016, pp. 10–18.
- [PVW11] G. Puy, P. Vandergheynst and Y. Wiaux. "On variable density compressive sampling". In: *IEEE signal processing letters* 18.10 (2011), pp. 595–598.
- [PW17] P. Putzky and M. Welling. *Recurrent Inference Machines for Solving Inverse Problems*. Tech. rep. 2017.
- [PW19] P. Putzky and M. Welling. "Invert to Learn to Invert". In: *Advances in neural information processing systems*. 2019.
- [Ram+21] E. Ramzi, N. THOME, C. Rambour, N. Audebert and X. Bitot. "Robust and Decomposable Average Precision for Image Retrieval". In: *Thirty-Fifth Conference on Neural Information Processing Systems*. 2021.
- [RB11] S. Ravishankar and Y. Bresler. "Magnetic Resonance Image Reconstruction from Highly Undersampled K-Space Data Using Dictionary Learning". In: *IEEE Transactions on Medical Imaging* 30.5 (2011), pp. 1028–1041.
- [Rei+10] P. Reimer, P. M. Parizel, J. F. Meaney and F. A. Stichnoth. *Clinical MR imaging*. Springer, 2010.
- [Rem+20] B. Remy, F. Lanusse, **Zaccharie Ramzi**, J. Liu, N. Jeffrey and J.-L. Starck. "Probabilistic Mapping of Dark Matter by Neural Score Matching". In: *NeurIPS 2020 Machine Learning for Physical sciences workshop*. 1. 2020, pp. 1–6.
- [RFB15] O. Ronneberger, P. Fischer and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [RHW86] D. E. Rumelhart, G. E. Hinton and R. J. Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

- [Rie+21] B. Riemenschneider, M. Muckley, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha, S. Arberet, D. Nickel, **Zaccharie Ramzi**, P. Ciuciu, J. L. Starck, J. Teuwen, D. Karkalousos, C. Zhang, A. Sriram, Z. Huang, N. Yakubova, Y. W. Lui and F. Knoll. "Results of the 2020 fastMRI Brain Reconstruction Challenge". In: *ISMRM*. 2021. Oral.
- [RM18] D. Recoskie and R. Mann. "Learning filters for the 2D wavelet transform". In: *Proceedings - 2018 15th Conference on Computer and Robot Vision, CRV 2018*. IEEE, 2018, pp. 198–205.
- [RMW14] D. J. Rezende, S. Mohamed and D. Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *31st International Conference on Machine Learning, ICML 2014*. Vol. 4. Jan. 2014, pp. 3057–3070.
- [Ros58] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.
- [RTH19] V. M. Runge, H. von Tengg-Kobligk and J. Heverhagen. *Essentials of Clinical MR*. Thieme, 2019.
- [Ryu+19] E. K. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang and W. Yin. "Plug-and-play methods provably converge with properly trained denoisers". In: *36th International Conference on Machine Learning, ICML 2019*. Vol. 2019-June. 2019, pp. 9764–9775.
- [San+20a] T. Sanchez, B. Gozcu, R. B. Van Heeswijk, A. Eftekhari, E. Ilıcak, T. Cukur and V. Cevher. "Scalable Learning-Based Sampling Optimization for Compressive Dynamic MRI". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings 2020-May*. 16066 (2020), pp. 8584–8588.
- [San+20b] C. M. Sandino, J. Y. Cheng, F. Chen, M. Mardani, J. M. Pauly and S. S. Vasanawala. "Compressed sensing: From research to clinical practice with deep neural networks". In: *IEEE Signal Processing Magazine* 37.1 (2020), pp. 117–127.
- [San+21] M. E. Sander, P. Ablin, M. Blondel and G. Peyré. "Momentum Residual Neural Networks". In: *ICML*. 2021.
- [SB17] T. Salimans and Y. Bulatov. *Saving memory using gradient-checkpointing*. <https://github.com/cybertronai/gradient-checkpointing>. Accessed: 2021-11-24. 2017.

- [SBB21] Y.-h. Shih, J. Blaschke and A. H. Barnett. *cuFINUFFT: a load-balanced GPU library for general-purpose nonuniform FFTs*. Tech. rep. 2021.
- [SCD02] J.-L. Starck, E. J. Candes and D. L. Donoho. "The curvelet transform for image denoising". In: *IEEE Transactions on image processing* 11.6 (2002), pp. 670–684.
- [Sch+18] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price and D. Rueckert. "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction". In: *IEEE Transactions on Medical Imaging* 37.2 (2018), pp. 491–503.
- [SE19] Y. Song and S. Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.
- [SE20] Y. Song and S. Ermon. "Improved Techniques for Training Score-Based Generative Models". In: *NeurIPS*. 2020.
- [SED20] S. Smith, E. Elsen and S. De. "On the Generalization Benefit of Noise in Stochastic Gradient Descent". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9058–9067.
- [SGW10] S. Schlenkrich, A. Griewank and A. Walther. "On the local convergence of adjoint Broyden methods". In: *Mathematical Programming* 121.2 (2010), pp. 221–247.
- [Sha+18] N. Shazeer, Y. Cheng, N. Parmar, D. Tran, A. Vaswani, P. Koanantakool, P. Hawkins, H. J. Lee, M. Hong, C. Young, R. Sepassi and B. Hechtman. "Mesh-tensorflow: Deep learning for supercomputers". In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.
- [She+20] F. Sherry, M. Benning, J. C. De los Reyes, M. J. Graves, G. Maierhofer, G. Williams, C. B. Schönlieb and M. J. Ehrhardt. "Learning the sampling pattern for MRI". In: *IEEE Transactions on Medical Imaging* (2020).
- [She+21] D. Shen, S. Ghosh, H. Haji-Valizadeh, A. Pathrose, F. Schiffrers, D. C. Lee, B. H. Freed, M. Markl, O. S. Cossairt, A. K. Katsaggelos and D. Kim. "Rapid reconstruction of highly undersampled, non-Cartesian real-time cine k-space data using a perceptual complex neural network (PCNN)". In: *NMR in Biomedicine* 34.1 (2021), pp. 1–12.

- [Sil+16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.
- [Sit+20] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell and G. Wetzstein. "Implicit Neural Representations with Periodic Activation Functions". In: *NeurIPS*. 2020.
- [SK16] T. Salimans and D. P. Kingma. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks". In: *Advances in neural information processing systems* 29 (2016), pp. 901–909.
- [SL18] S. L. Smith and Q. V. Le. "A Bayesian Perspective on Generalization and Stochastic Gradient Descent". In: *International Conference on Learning Representations*. 2018.
- [SM50] J. Sherman and W. J. Morrison. "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix". In: *The Annals of Mathematical Statistics* 21.1 (1950), pp. 124–127.
- [Son+21] Y. Song, L. Shen, L. Xing and S. Ermon. *Solving Inverse Problems in Medical Imaging with Score-Based Generative Models*. Tech. rep. 2021.
- [Spr+16] E. Springer, B. Dymerska, P. Lima Cardoso, S. D. Robinson, C. Weisstanner, R. Wiest, B. Schmitt and S. Trattnig. "Comparison of Routine Brain Imaging at 3 T and 7 T". In: *Investigative Radiology* 51.8 (Aug. 2016), pp. 469–482.
- [SPX21] L. Shen, J. Pauly and L. Xing. "NeRP : Implicit Neural Representation Learning with Prior Embedding for Sparsely Sampled Image Reconstruction". 2021.
- [Sri+14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.
- [Sri+20] A. Sriram, J. Zbontar, T. Murrell, A. Defazio, C. L. Zitnick, N. Yakubova, F. Knoll and P. Johnson. "End-to-End Variational Networks for Accelerated MRI Reconstruction". In: *MICCAI*. 2020.
- [SSS20] J. Schlemper, M. Salehi and M. Sofka. *Mutli-coil Magnetic Resonance Imaging Using Deep Learning*. 2020.

- [Stu+15] D. Stucht, K. A. Danishad, P. Schulze, F. Godenschweger, M. Zaitsev and O. Speck. "Highest Resolution In Vivo Human Brain MRI Using Prospective Motion Correction". In: *PLOS One* 10.7 (July 2015).
- [Sun+20] Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros and M. Hardt. "Test-Time Training for Out-of-Distribution Generalization". In: *ICML*. 2020.
- [Sun+21] Y. Sun, J. Liu, M. Xie, B. Wohlberg and U. S. Kamilov. "CoLL: Coordinate-based Internal Learning for Imaging Inverse Problems". In: *IEEE Transactions on Computational Imaging* (2021).
- [Sut+13] I. Sutskever, J. Martens, G. Dahl and G. Hinton. "On the importance of initialization and momentum in deep learning". In: *International conference on machine learning*. PMLR. 2013, pp. 1139–1147.
- [Tel16] M. Telgarsky. "Benefits of depth in neural networks". In: *Conference on learning theory*. PMLR. 2016, pp. 1517–1539.
- [The+99] D. R. Thedens, P. Irarrazaval, T. S. Sachs, C. H. Meyer and D. G. Nishimura. "Fast magnetic resonance coronary angiography with a three-dimensional stack of spirals trajectory". In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 41.6 (1999), pp. 1170–1179.
- [Tol+21] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit et al. *Mlp-mixer: An all-mlp architecture for vision*. Tech. rep. 2021.
- [Ton+19] F. Tonolini, J. Radford, A. Turpin, D. Faccio and R. Murray-Smith. "Variational Inference for Computational Imaging Inverse Problems". In: *Journal of Machine Learning Research* 21 (Apr. 2019), pp. 1–46.
- [Trz+14] J. D. Trzasko, A. Manduca, Y. Shu, J. Huston and M. A. Bernstein. "A Preconditioned ADMM Strategy for Field-Corrected Non-Cartesian MRI Reconstruction". In: *ISMRM*. Vol. 22. 2014, p. 1535.
- [Uec+14] M. Uecker, P. Lai, M. J. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S. Vasanawala and M. Lustig. "ESPIRiT-An Eigenvalue Approach to Autocalibrating Parallel MRI: Where SENSE meets GRAPPA". In: *Magnetic Resonance in Medicine* 71.3 (2014), pp. 990–1001.



- [UVL16] D. Ulyanov, A. Vedaldi and V. Lempitsky. *Instance normalization: The missing ingredient for fast stylization*. Tech. rep. 2016.
- [UVL18] D. Ulyanov, A. Vedaldi and V. Lempitsky. "Deep Image Prior". In: *CVPR*. 2018, pp. 9446–9454.
- [Vas+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.
- [VBW13] S. V. Venkatakrisnan, C. A. Bouman and B. Wohlberg. "Plug-and-Play priors for model based reconstruction". In: *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*. 2013, pp. 945–948.
- [Vin11] P. Vincent. "A connection between scorematching and denoising autoencoders". In: *Neural Computation* 23.7 (2011), pp. 1661–1674.
- [Vir+20] P. Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272.
- [VK20] A. Vahdat and J. Kautz. "NVAE: A Deep Hierarchical Variational Autoencoder". In: *NeurIPS*. July 2020.
- [VYL18] P. Virtue, S. X. Yu and M. Lustig. "Better than real: Complex-valued neural nets for MRI fingerprinting". In: *Proceedings - International Conference on Image Processing, ICIP*. Vol. 2017-Septe. 2018, pp. 3953–3957.
- [Wan+04] Z. Wang, A. C. Bovik, H. Rahim Sheikh and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE TRANSACTIONS ON IMAGE PROCESSING* 13.4 (2004).
- [Wan+21a] G. Wang, T. Luo, J.-F. Nielsen, D. C. Noll and J. A. Fessler. *B-spline Parameterized Joint Optimization of Reconstruction and K-space Trajectories (BJORK) for Accelerated 2D MRI*. Tech. rep. 2021.
- [Wan+21b] K. Wang, M. Kellman, C. M. Sandino, K. Zhang, S. S. Vasanawala, J. I. Tamir, S. X. Yu and M. Lustig. "Memory-efficient Learning for High-Dimensional MRI Reconstruction". 2021.
- [WDS18] G. Wu, J. Domke and S. Sanner. *Conditional Inference in Pre-trained Variational Autoencoders via Cross-coding*. Tech. rep. May 2018.

- [Wei+20] K. Wei, A. Aviles-rivero, J. Liang, Y. Fu, C.-b. Schönlieb and H. Huang. "Tuning-free Plug-and-Play Proximal Algorithm for Inverse Imaging Problems". In: *ICML*. 2020.
- [Wei+21] T. Weiss, O. Senouf, S. Vedula, O. Michailovich, M. Zibulevsky and A. Bronstein. "PILOT : Physics-Informed Learned Optimized Trajectories for Accelerated MRI". In: *Journal of Machine Learning for Biomedical Imaging* 1.6 (2021), pp. 1–23.
- [WH18] Y. Wu and K. He. "Group normalization". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.
- [Wri+14] K. L. Wright, J. I. Hamilton, M. A. Griswold, V. Gulani and N. Seiberlich. "Non-Cartesian parallel imaging reconstruction". In: *Journal of Magnetic Resonance Imaging* 40.5 (2014), pp. 1022–1040.
- [WRL19] Y. Wu, M. Rosca and T. Lillicrap. "Deep Compressed Sensing". In: *International Conference on Machine Learning (ICML)*. 2019.
- [WSB03] Z. Wang, E. Simoncelli and A. C. Bovik. "Multi-Scale Structural Similarity for Image Quality Assessment". In: *Proceedings of the 37th IEEE Asilomar Conference on Signals, Systems and Computers*. Vol. 2. 2003, pp. 9–13.
- [Xie+21] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann and S. Sridhar. *Neural Fields in Visual Computing and Beyond*. Tech. rep. 2021.
- [XRM20] P. Xu, F. Roosta-Khorasani and M. W. Mahoney. "Second-Order Optimization for Non-Convex Machine Learning: An Empirical Study". In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. 2020.
- [Xu+20] X. Xu, J. Liu, Y. Sun, B. Wohlberg and U. S. Kamilov. "Boosting the Performance of Plug-and-Play Priors via Denoiser Scaling". In: (2020), pp. 1–16.
- [Yam+20] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Ugurbil and M. Akcakaya. "Self-Supervised Physics-Based Deep Learning MRI Reconstruction Without Fully-Sampled Data". In: *Proceedings - International Symposium on Biomedical Imaging*. Vol. 2020-April. 2020, pp. 921–925.
- [Yar21] D. Yarotsky. "Universal approximations of invariant maps by neural networks". In: *Constructive Approximation* (2021), pp. 1–68.

- [YHC18] J. C. Ye, Y. Han and E. Cha. "Deep convolutional framelets: A general deep learning framework for inverse problems". In: *SIAM Journal on Imaging Sciences* 11.2 (2018), pp. 991–1048.
- [YM17] Y. Yoshida and T. Miyato. *Spectral Norm Regularization for Improving the Generalizability of Deep Learning*. Tech. rep. 2017.
- [Yoo+21] J. Yoo, K. Hwan Jin, H. Gupta, J. Yerly, M. Stuber and M. Unser. "Time-Dependent Deep Image Prior for Dynamic MRI". In: *IEEE Transactions on Medical Imaging* (2021).
- [YPJ19] S. Yu, B. Park and J. Jeong. "Deep iterative down-up CNN for image denoising". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019.
- [Zac+19] **Zaccharie Ramzi**, P. Ciuciu, J.-L. Starck and J.-L. Starck Benchmarking. "Benchmarking proximal methods acceleration enhancements for CS-acquired MR image analysis reconstruction". In: *SPARS 2019 - Signal Processing with Adaptive Sparse Structured Representations Workshop*. 2019.
- [Zac+20] **Zaccharie Ramzi**, B. Remy, F. Lanusse, J.-L. Starck and P. Ciuciu. "Denoising Score-Matching for Uncertainty Quantification in Inverse Problems". In: *NeurIPS 2020 Deep Learning and Inverse Problems workshop*. 2020.
- [Zac+21a] **Zaccharie Ramzi**, K. Michalewicz, J. L. Starck, T. Moreau and P. Ciuciu. "Wavelets in the deep learning era". 2021. Under review in *Journal of Mathematical Imaging and Vision*.
- [Zac+21b] **Zaccharie Ramzi**, J. L. Starck, T. Moreau and P. Ciuciu. "Wavelets in the deep learning era". In: *European Signal Processing Conference*. Vol. 2021-Janua. 2021, pp. 1417–1421. Oral.
- [Zac+21c] **Zaccharie Ramzi**, A. Vignaud, J.-L. Starck and P. Ciuciu. "Is good old GRAPPA dead?" In: *ISMRM*. 2021.
- [Zac+22a] **Zaccharie Ramzi**, C. G R, J.-L. Starck and P. Ciuciu. "NC-PDNet: a Density-Compensated Unrolled Network for 2D and 3D non-Cartesian MRI Reconstruction". In: *IEEE Transactions on Medical Imaging* (2022).
- [Zac+22b] **Zaccharie Ramzi**, F. Mannel, S. Bai, J.-L. Starck, P. Ciuciu and T. Moreau. "SHINE: SHaring the INverse Estimate from the forward pass for bi-level optimization and implicit models". In: *International Conference on Learning Representations*. 2022. Spotlight.

- [Zbo+18] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdal, A. Romero, M. Rabbat, P. Vincent, N. Yakubova, J. Pinkerton, D. Wang, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson and Y. W. Lui. *fastMRI: An Open Dataset and Benchmarks for Accelerated MRI*. Tech. rep. 2018, pp. 1–35.
- [ZCS20a] **Zaccharie Ramzi**, P. Ciuciu and J. L. Starck. “Benchmarking Deep Nets MRI Reconstruction Models on the Fastmri Publicly Available Dataset”. In: *Proceedings - International Symposium on Biomedical Imaging*. Vol. 2020-April. 2020, pp. 1441–1445.
- [ZCS20b] **Zaccharie Ramzi**, P. Ciuciu and J. L. Starck. “Benchmarking MRI reconstruction neural networks on large public datasets”. In: *Applied Sciences (Switzerland)* 10.5 (2020).
- [ZCS20c] **Zaccharie Ramzi**, P. Ciuciu and J.-L. Starck. “XPDNet for MRI Reconstruction: an application to the 2020 fastMRI challenge”. In: *ISMRM*. 2020, pp. 1–4. Oral.
- [Zha+17a] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang. “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising”. In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155.
- [Zha+17b] K. Zhang, W. Zuo, S. Gu and L. Zhang. “Learning deep CNN denoiser prior for image restoration”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. 2017, pp. 2808–2817.
- [Zhu+17] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2242–2251.
- [Zhu+18] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen and M. S. Rosen. “Image reconstruction by domain-transform manifold learning”. In: *Nature* 555.7697 (Mar. 2018), pp. 487–492.
- [ZSC21] **Zaccharie Ramzi**, J. L. Starck and P. Ciuciu. “Density compensated unrolled networks for non-cartesian MRI reconstruction”. In: *Proceedings - International Symposium on Biomedical Imaging*. Vol. 2021-April. 2021, pp. 1443–1447.

- [ZZZ19] K. Zhang, W. Zuo and L. Zhang. "Deep plug-and-play super-resolution for arbitrary blur kernels". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1671–1681.