Spring 5-15-2023

Model-based Deep Learning for Computational Imaging

Xiaojian Xu
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

McKelvey School of Engineering
Department of Computer Science and Engineering

Dissertation Examination Committee:
Ulugbek S. Kamilov, Chair
Tao Ju
Netanel Raviv
Brendt Wohlberg
Dmitriy A. Yablonskiy
William Yeoh

Model-based Deep Learning for Computational Imaging

by

Xiaojian Xu

A dissertation presented to
the McKelvey School of Engineering
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

August 2022
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

If life is a journey, I am very grateful to have the past five years as part of it. I learned, I grew and I changed during this PhD study. Recall this unforgettable journey, there are so many people I want to thank.

First, I want to give my deep and sincere thanks to my PhD advisor, Prof. Ulugbek S. Kamilov, to whom I am much more grateful than I can express. I first got to know Bek at the PhD introduction seminar, when I had no idea about computational imaging, let alone knowing that it will be my passion for my whole PhD or possibly even a life-long career. Bek later took me as his student, he taught me, guided me, helped me and supported me with no reservation ever since. The working experience with Bek totally opened a new door for me to see different things as well as different ways to do things. His humor as a friend, his great passion as a researcher, and his tremendous support as an advisor has slowly but deeply influenced me and set a model for me. The research journey is not always smooth, but Bek being there always gives me confidence. I am more than grateful to him for all the patience, trust and encouragement he gives me. I always remember the conversation happened in the summer of 2019 when Bek told me that we can do the best research because I should not only believe in myself but also believe in him. His words have encouraged from time to time during my PhD study and I will move on with them in my future career.

I also want to thank other members of my committee, Prof. Tao Ju, Prof. Netanel Raviv, Dr. Brendt Wohlberg, Prof. Dmitriy A. Yablonskiy and Prof. William Yeoh, for reviewing the dissertation. I additionally want to thank Dr. Brendt Wohlberg for his guidance as an collaborator and his support on my career pursuit. I want to thank Prof. Tao Ju for his valuable advices on my presentation when I was a junior PhD student, and his efforts on hosting the imaging seminars, where I have greatly broadened my horizon on graphics and vision. I also want to thank Prof. Dimitry A. Yablonskiy for his helpful and valuable inputs to our collaboration.

I would like to thank Prof. Roch Guérin who admitted me to WashU, without which I could not have started such a colorful journey.

I thank the following past and present members of the Computational Imaging Group (CIG): Yu Sun, Jiaming Liu, Weijie Gan, Yuyang Hu, Zhixin Sun, Shirin Shoushtari, Tomas Kerepecky, and Guangxiao Song. I appreciate their valuable inputs to my research and the good moments we shared together. Besides those group members, I also want to thank the following students who have worked with me in CIG: Shiqi Xu, Zihui Wu, Max Troop. I also take this opportunity to express gratitude to all the students whose projects I have supervised: Long Fa, Weijie Gan, Jiarui Xing, Hao Tang, Ryogo Suzuki, Mingyang Xie, YuKun Li, Eddie Chandler, Zhixin Sun, Julia Zeng and Yixuan Luo, EddieChandler, Zhixin Sun, Julia Zeng and Yixuan Luo. I additionally thank Yu Sun, Jiaming Liu, Weijie Gan, Shiqi Xu and Zihui Wu for the fruitful collaboration. I wish them prosperity in their future academic life and careers.

I want to thank my collaborators and their group: Prof. Hongyu An, Dr. Hassan Mansour, Dr. Brendt Wohlberg, Prof. Dmitriy A. Yablonskiy, and Abdullah H. Al-Shabili. I want to

additionally thank Dr. Hassan Mansour, Dr. Petros Boufounos, Dr. Brain Wheelwright and Dr. Melissa Geng for their supervision during my internship in industrial research labs.

I want to thank my friends, Zhihao, Xia, Bei Wu, Wei Tang, Rusi Yan, Jeffery Jung, and Araon Park. I really enjoyed the time we spent together. Seeing them graduating and leaving is surely a sad thing, but I truly wish them great happiness in their new positions. I also want to thank my other friends for their company and help at WashU. They are Ruixuan Dai, Qianyu Li, Yifan Xu, TianTian Zhu, Hao Yan, Dan Zeng, Chenfeng Zhao, Jian Wang, Shenghua He, Chunyu Song and Zhen Zhang. Without them, I would not had such a colorful journey.

I also want to give my thanks to my old friends in China. First, my close friend Chunyang Tong. Chunyang and I have known each other for more than tens years since the very beginning of the college. I feel so lucky to have such a friend who I trust and who also trust me such that I can share both my sorrows and happiness with her. I also want to thank my friend Lei Ren for his caring from time to time.

I want to give my sweet thanks to my boyfriend Kyle Singer, for his constant companionship and support through my whole PhD journey. Kyle is such a sincere and nice person that meeting him is one of the best things happened to me at WashU. When I recall the times we spend together, it is always the good memory that makes me feel cared and loved. Not only Kyle but his whole family, Gail Singer, David Singer, Shannon Oikawa and Kenta Oikawa were all supportive and nice to me. Staying with them always makes me feel relaxed and happy. Those days that we have spend together have all locked in my mind and built up my beautiful memories about St. Louis. And because of them, St. Louis feels like home to me.

Finally, I want to thank my family, my mother Yuxiang Wei, my father Fazhan Xu, my sister Xiaozhen Xu and my brother Chenxi Xu, for their support. My parents have been working

super hard since I was young to support our family in a unfamiliar big city. Their hard working inspired me to also work hard to pursue my dream when I was young. Although they never understand exactly what I work on, they have never ceases to support my decisions. Without their support, I would have not been able to pursue my dream this far. I also want to thank my sister and my brother for their support and care. Additional thanks to my beloved sister, who has been constantly caring for me, making me feel needed and taking care of my parents when I was not around. I wish her all the best in the world.

Xiaojian Xu

Washington University in St. Louis

August 2022

Dedicated to my family, and those who I love.

ABSTRACT OF THE DISSERTATION

Model-based Deep Learning for Computational Imaging

by

Xiaojian Xu

Doctor of Philosophy in Computer Science

Washington University in St. Louis, 2022

Professor Ulugbek S. Kamilov, Chair

This dissertation addresses model-based deep learning for computational imaging. The motivation of our work is driven by the increasing interests in the combination of imaging model, which provides data-consistency guarantees to the observed measurements, and deep learning, which provides advanced prior modeling driven by data. Following this idea, we develop multiple algorithms by integrating the classical model-based optimization and modern deep learning to enable efficient and reliable imaging. We demonstrate the performance of our algorithms by validating their performance on various imaging applications and providing rigorous theoretical analysis.

The dissertation evaluates and extends three general frameworks, plug-and-play priors (PnP), regularized by denoising (RED) and deep unfolding (DU), all of which integrate model-based optimization and deep learning. PnP and RED adopt deep-learned denoisers as image priors inside model-based iterative algorithms, while DU interprets the iterations of a model-based algorithm as layers of a deep neural network and trains it end-to-end in a supervised fashion. We contribute to these research areas by 1) providing the statistical interpretation of the PnP algorithms through the analysis of the priors implicitly represented by denoisers; 2) proposing an incremental variant of the widely-used PnP-ADMM algorithm to handle problems involving large-scale measurements; 3) extending the family of PnP algorithms to the non-Euclidean

setting based on the general Bregman distance; and 4) developing an end-to-end model-based learning framework for the estimation of quantitative maps from under-sampled, noisy and motion-corrupted MRI data.

Part I

Introduction

Chapter 1

Introduction

COMPUTATIONAL imaging is the process of indirectly forming images from measurements using algorithms that rely on a significant amount of computing. Different from traditional imaging, computational imaging systems involve a tight integration of the sensing system and the computation in order to form the images of interest.[1] Benefiting from such hardware and software integration, computational imaging systems are used in broad range of applications including computational microscopy, X-Ray computed tomography(CT) imaging, magnetic resonance imaging(MRI), ultrasound imaging, computational photography. Fig. 1.1 shows some examples of such computational imaging applications.

Image reconstruction is at the core of computational imaging. Such reconstruction is usually formulated as an *inverse problem*, where we use the measurements of the sensing system to compute the unknown desired images. Algorithms that can solve such inverse problems are

[1]Definition from the Wikipedia entry *'Computational Imaging'*.

(a) Imaging with microscopes (b) Imaging with cameras (c) Imaging with CT scanners

Figure 1.1: Some examples of computational imaging applications with different imaging instruments: (a) microscopes, (b) cameras, and (c) CT scanners.

in high demand for modern imaging applications. However, solving imaging inverse problems is very challenging in practice due to the following reasons:

- **Non-unique solution.** Inverse problems are usually ill-posed, which means that many different solutions may be consistent with the measured data.
- **Noisy measurements.** The measurements are usually corrupted with noise during the signal acquisition. Such noise contamination will damage the accuracy of the measured data and consequently mislead the reconstruction method.
- **High computational complexity.** The data can be very high-dimensional in a sense that the measurements and the corresponding solution may contain millions or even billions of data entries, making the reconstruction computationally expensive.

These challenges motivate the development of not only effective but efficient image reconstruction algorithms that can balance the quality of the reconstructed images and the computational cost. In this dissertation, we seek to develop such novel computational imaging algorithms that can take advantage of both the ***physics*** of the imaging systems and advanced ***deep learning priors*** to enable reliable image reconstruction. In particular, we build our work on three different frameworks, namely plug-and-play priors (PnP), regularized by denoising (RED), and deep unfolding (DU), all of which integrate imaging models and

the learning capability of deep neural networks. We contribute to these areas by developing novel imaging algorithms, providing rigorous theoretical analysis, and applying algorithms to various imaging tasks. We summarize our key contributions in this dissertation in the following section.

1.1 Main Contributions

This dissertation contains the following major contributions to computational imaging.

- We present a simple, but effective, *denoiser scaling* technique for improving the performance of PnP algorithms. The proposed technique is shown to be particularly valuable when PnP is used with CNN denoisers that have no explicit tunable parameters. We theoretical justify the denoiser scaling from the perspectives of proximal optimization, statistical estimation, and consensus equilibrium. We show the potential of denoiser scaling to significantly improve the performance of PnP across several inverse problems.

- We establish the first theoretical convergence result for proximal gradient method (PGM) variant of PnP for minimum mean squared error (MMSE) denoisers. We show that the iterates produced by PnP-PGM with an MMSE denoiser converge to a stationary point of some global cost function. We validate our analysis on sparse signal recovery by comparing two types of denoisers, namely the *exact* MMSE denoiser and the *approximate* MMSE denoiser obtained by training a deep neural net. Our results illustrate the potential of denoisers obtained by training deep neural nets, which have been extensively used in practice, to match the performance of the *exact* MMSE denoiser.

- We provides several new insights into the PnP methodology in the context of large-scale imaging problems. First, we propose IPA—a new incremental variant of PnP-ADMM

algorithm—to allow randomized partial processing of measurements in large scale settings. Second, we theoretically analyze IPA under a set of realistic assumptions, showing that in expectation IPA can approximate the convergence behavior of PnP-ADMM to a desired precision by controlling the penalty parameter. Third, we validate the potential of IPA to handle nonsmooth data-fidelity terms, large number of measurements, and DNN priors with multiple imaging tasks, highlighting the effectiveness of IPA for addressing large-scale imaging problems.

- We propose to broaden PnP/RED by considering a non-Euclidean setting based on the more general Bregman distance. This work can be considered as a first step towards extending widely-used PnP/RED to problems where there is a benefit of using non-Euclidean formulations of proximal and projection operators. We present a theoretical convergence result for our method and demonstrate the effectiveness of our algorithms on Poisson linear inverse problems using a deep unfolding architecture.

- We design and apply a model-based learning framework for the quantitative MRI application. Our method is the first method for end-to-end estimation of quantitative MRI maps directly from under-sampled, noisy and motion-corrupted k-space data. Our results show that our method achieves the best performance in different scenarios compared to other widely-used methods, showing its effectiveness and potential in practical applications.

1.2 Organization of the Dissertation

This dissertation is organized as follows: In Part II, we provide the statistical interpretation for PnP through the analysis of the priors implicitly represented by denoisers. We first present an effective denoiser scaling technique that has a potential to broadly improve the performance of PnP algorithms. We provide a theoretical justification linking the technique

to the strength of the regularization within PnP. We then prove that the iterates produced by PnP-PGM algorithm with an MMSE denoiser converge to a stationary point of some global cost function, providing theoretical justification to the performance of the approximate MMSE denoiser obtained by training a deep neural networks. In Part III, we focus on the practical challenge of PnP in large-scale settings by proposing an incremental variant of the widely used PnP-ADMM algorithm, making it scalable to problems involving a large number measurements. We theoretically analyze the convergence of the algorithm under a set of explicit assumptions, extending recent theoretical results in the area. Additionally, we show the effectiveness of our algorithm with nonsmooth data-fidelity terms and deep neural net priors, its fast convergence compared to existing PnP algorithms, and its scalability in terms of speed and memory. In Part IV, we extend the family of PnP algorithms to the non-Euclidean setting. We develop two new Bregman proximal gradient method variants, namely PnP-BPGM and RED-BSD algorithms, by replacing the traditional updates in PnP and RED from the quadratic norms to more general Bregman distance. We present a theoretical convergence result for PnP-BPGM and demonstrate the effectiveness of our algorithms on Poisson linear inverse problems. In Part V, we present an effective end-to-end model-based neural network for solving the accelerated quantitative MRI problem and validate its performance on experimentally collected data.

Chapter 2

Background

This chapter introduces the background material for this dissertation. We start by giving a mathematical formulation for inverse problems, including the forward model that relates to the physics of the imaging systems and the statistical interpretation based on Bayesian estimation. Based on this formulation, we then review the classical model-based optimization methods and the recent popular learning-based methods for solving inverse problems. Through the comparison of these two different types of approaches, we show that model-based optimization and deep learning offer complimentary strategies for handling imaging problems, and algorithms that can take advantage of both have the potential to enable more efficient and more reliable image reconstruction. Finally, we introduce more advanced algorithms, including plug-and-play priors (PnP), regularization by denoising (RED), and deep unfolding (DU), all of which achieved success in imaging by integrating the information of forward models and the learning capability of deep neural networks.

2.1 Imaging as Inverse Problems

The development and analysis of imaging algorithms, which is the core of this dissertation, relies on the formal formulation of the computational imaging problems. In this section, we introduce this formalism that will be used extensively in the sequel, including mathematical definitions of imaging systems and statistical interpretations of imaging problems.

2.1.1 Forward Model

In the context of computational imaging, the measurement process corresponds to the acquisition of measurements $\boldsymbol{y} \in \mathbb{R}^m$ from the unknown target image $\boldsymbol{x} \in \mathbb{R}^n$. An understanding of this acquisition process is a prerequisite for understanding computational imaging algorithms. We refer to this process as a *forward model*, often represented as

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{e}. \tag{2.1}$$

For notational convenience, we assume that the image and measurements are both real-valued; nonetheless, all the algorithms developed in this dissertation can be easily extended to complex-valued data. Note here, the operator $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m$, usually referred to as the *forward operator*, represents the physics of the imaging system. This forward operator varies across imaging modalities but is usually assumed to be known and accurately describes the deterministic physical process of the measurement acquisition. The vector $\boldsymbol{e} \in \mathbb{R}^m$ models the non-deterministic noise corruption in the imaging system. In practice, the causes of this noise are various and often intractable. However, by assuming that the noise sources are independent and adopting the the central limit theorem, we can model $\boldsymbol{e}$ as independent and identically distributed (i.i.d.) Gaussian with zero mean, known as additive white Gaussian

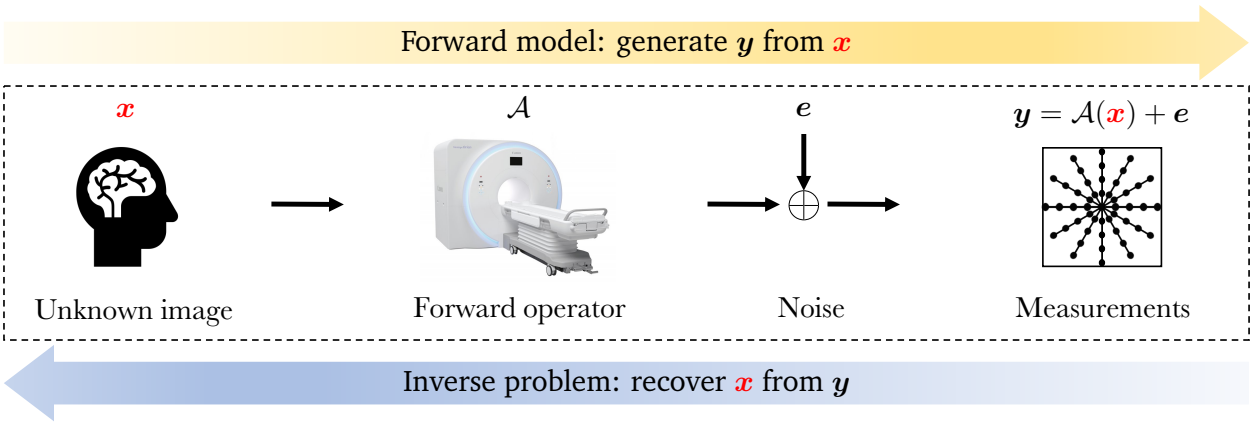

Figure 2.1: The illustration of an imaging pipeline for the forward and inverse problems corresponding to (2.1) in MRI. The forward model refers to the acquisition of measurements $\boldsymbol{y}$ from unknown image $\boldsymbol{x}$, while the inverse problem refers to the recovery of $\boldsymbol{x}$ from $\boldsymbol{y}$.

noise (AWGN). In this dissertation, we denote AWGN as

$$\boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \tag{2.2}$$

where $\mathbf{0} \in \mathbb{R}^n$ is the zero vector and σ the standard deviation, and $\mathbf{I}$ the identity matrix.

The forward model presented in (2.1) models the acquisition of measurements $\boldsymbol{y}$ from an target $\boldsymbol{x}$. The recovery of the unknown image $\boldsymbol{x}$ from the noisy measurements $\boldsymbol{y}$, on the other hand, is referred to as an *inverse problem*. Fig. 2.1 illustrates an imaging pipeline for the forward and inverse problems corresponding to (2.1) in MRI. Inverse problems related to image reconstruction are fundamental in computational imaging. Inverse problems can be divided into continuous problems and discrete problems based on the data type involved. In this dissertation, we only focus on the discrete inverse problems where both image $\boldsymbol{x}$ and measurements $\boldsymbol{y}$ are discrete vectors.

Linear Models

The forward model (2.1) describes the mapping from an image $\boldsymbol{x}$ to the measurements $\boldsymbol{y}$ in a general imaging system. Further simplification of such mapping is the *linear forward model*

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \tag{2.3}$$

where the forward operator $\mathcal{A}$ in (2.1) is re-expressed as a measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{e}$ models the AWGN. In this scenario, the inverse problem of recovering $\boldsymbol{x}$ from noisy measurements $\boldsymbol{y}$ is therefore reduced to a *linear inverse problem*. Linear inverse problems are central to most modern imaging systems, including optical microscopy, digital cameras, MRI, and CT. The major benefits of linear problems is the feasibility of using standard theoretical results from linear algebra in deriving solutions. Most algorithms we develop in this dissertation are particularly well-suited to solve linear inverse problems under AWGN. Nevertheless, it is possible to extend our algorithms to more general settings with other noise types. For example, in Chapter 7, we specifically discuss the development of algorithms for Poisson noise, which usually occurs in the optical devices under a low-light exposure.

Image Denoising

Further simplification of the liner forward model in Eq. (2.1) can be obtained by assuming the forward operator $\mathcal{A}$ is an identity matrix

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}, \tag{2.4}$$

where $\boldsymbol{y}$ is simply a noisy observation of the clean image $\boldsymbol{x}$ corrupted by additive noise $\boldsymbol{e}$. The estimation of clean image $\boldsymbol{x}$ from its noisy version $\boldsymbol{y}$ is known as *image denoising*.

Image denoising is considered as a basic but fundamental problem in computational imaging. As we shall see in detail, the denoising is a key sub-routine influences the quality of the reconstructed image in many methods for solving more general inverse problems.

2.1.2 Statistical Inference

In practice, inverse problems such as (2.1) are often ill-posed, meaning that it is impossible to recover $\boldsymbol{x}$ by simply inverting $\mathcal{A}$. Therefore, prior knowledge additional to the information of the measurements and forward operators is needed to compute faithful and high-quality images. For example, Bayesian estimation uses a prior probability distribution $p_{\boldsymbol{x}}$, which describes our belief of the distribution of the underlying true image $\boldsymbol{x}$, to impose desired constrains on the solutions. In particular, given the prior $p_{\boldsymbol{x}}$ and following Bayesian theory, the posterior distribution of the true image, denoted as $p_{\boldsymbol{x}|\boldsymbol{y}}$, can be expressed as

$$p_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})p_{\boldsymbol{x}}(\boldsymbol{x})}{p_{\boldsymbol{y}}(\boldsymbol{y})} \propto \underbrace{p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})}_{\text{Likelihood}} \underbrace{p_{\boldsymbol{x}}(\boldsymbol{x})}_{\text{Prior}}. \tag{2.5}$$

Here we use $\propto$ to denote equality after normalization and $p_{\boldsymbol{y}|\boldsymbol{x}}$ to denote the likelihood function that characters the probabilistic relationship between the desired image $\boldsymbol{x}$ and the measurements $\boldsymbol{y}$. The classical *maximum-a-posteriori probability (MAP)* estimator is therefore obtained by maximizing the posterior distribution $p_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y})$

$$\widehat{\boldsymbol{x}}_{\text{MAP}} = \arg\max_{\boldsymbol{x}} \left\{ p_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y}) \right\} \tag{2.6a}$$

$$= \arg\min_{\boldsymbol{x}} \left\{ -\log(p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})) - \log(p_{\boldsymbol{x}}(\boldsymbol{x})) \right\} \tag{2.6b}$$

$$= \arg\min_{\boldsymbol{x}} \{ \underbrace{\mathcal{D}(\boldsymbol{x})}_{\text{Data-fidelity}} + \underbrace{\mathcal{R}(\boldsymbol{x})}_{\text{Regularizer}} \}. \tag{2.6c}$$

Note here by using the Bayes rule, we turn the estimation task into an optimization problem, where the solution is obtained by minimizing an objective function consisting of two terms, $\mathcal{D}(\boldsymbol{x})$ and $\mathcal{R}(\boldsymbol{x})$. In the context of computational imaging, $\mathcal{D}(\boldsymbol{x})$ is usually called the *data-fidelity* term as it controls the data consistency to the measurements, and $\mathcal{R}(\boldsymbol{x})$ is called the *regularizer* or *prior* term as it imposes our preferred properties or prior knowledge to the estimation. By adjusting the regularizer term $\mathcal{R}(\boldsymbol{x})$, in other words, the prior distribution $p_{\boldsymbol{x}}$, we can directly change our preference on the solution. The optimization formulation (2.6) is known as *regularized optimization* and widely adopted in solving inverse problems because 1) it is interpretable, 2) it is usually solvable, and 3) it is friendly to theoretical analysis.

Regularized Least Squares

Regularized optimization (2.6) can accommodate a variety of data-fidelity and regularization terms. For example, under the assumption that the noise corruption in forward model is AWGN with $\boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, the likelihood can be obtained as

$$p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{\sqrt{\det(2\pi\sigma^2\mathbf{I})}} \exp\left(-\frac{1}{2\sigma^2}\|\mathcal{A}(\boldsymbol{x}) - \boldsymbol{y}\|_2^2\right), \tag{2.7}$$

where $\det(\cdot)$ denotes the determinant of a matrix and $\|\cdot\|_2$ denotes the standard ℓ_2-norm in $\mathbb{R}^n$. For the sake of simplicity, we use $\|\cdot\|$ to denote the ℓ_2-norm in the rest of this dissertation. Plugging this result into Eq. (2.6) we have

$$\widehat{\boldsymbol{x}}_{\mathrm{MAP}} = \underset{\boldsymbol{x}}{\mathsf{arg\,max}} \left\{ -\mathsf{log}(p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})) - \mathsf{log}(p_{\boldsymbol{x}}(\boldsymbol{x})) \right\} \tag{2.8a}$$

$$= \underset{\boldsymbol{x}}{\mathsf{arg\,min}} \left\{ \frac{1}{2\sigma^2} \|\mathcal{A}(\boldsymbol{x}) - \boldsymbol{y}\|^2 + \mathcal{R}(\boldsymbol{x}) \right\} \tag{2.8b}$$

$$= \underset{\boldsymbol{x}}{\mathsf{arg\,min}} \left\{ \frac{1}{2} \|\mathcal{A}(\boldsymbol{x}) - \boldsymbol{y}\|^2 + \tau \mathcal{R}(\boldsymbol{x}) \right\}, \tag{2.8c}$$

where in the last equality, σ^2 is absorbed into the regularization parameter $\tau > 0$ to adjusts the relative strength of the regularizer. By substituting forward operator $\mathcal{A}$ with its matrix form $\boldsymbol{A}$ for a linear system, we have the *regularized least-squares optimization* as

$$\widehat{\boldsymbol{x}}^{\mathrm{LS}}_{\mathrm{MAP}} = \underset{\boldsymbol{x}}{\mathsf{arg\,min}} \left\{ \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \tau \mathcal{R}(\boldsymbol{x}) \right\}. \tag{2.9}$$

This regularized least-squares optimization will accompany us throughout most part of this dissertation. However, it worthy mentioning that if distribution changes for noise $\boldsymbol{e}$, we depart form the ℓ_2 data-fidelity form.

2.2 Computational Imaging Algorithms

So far, we have explained the essential recipes for understanding computational imaging algorithms, including how to define the imaging systems with forward models, how to formulate imaging as an inverse problem, and how to solve such inverse problems via Bayesian inference. In fact, since Bayesian inference (see Eq. (2.6)) uses the operator $\mathcal{A}$ of the forward model, it is usually refereed to as *model-based method* in the context of computational imaging. While model-based method is a classical approach, *deep learning (DL)* has recently drawn considerable attention in image reconstruction. Instead of explicitly defining a regularizer, the general idea of DL is to train a *deep neural networks (DNN)* on a large number of data

samples to map low-quality images to their desired high-quality counterparts. In this section, we briefly review these two different types of methods in order to explain the motivation in integrating model-based and learning-based methods.

2.2.1 Model-based Methods

In the classical model-base approach, the reconstruction of image $\boldsymbol{x}$ from measurements $\boldsymbol{y}$ is usually formulated as an regularized optimization problem of the form

$$\widehat{\boldsymbol{x}} = \operatorname*{arg\,min}_{\boldsymbol{x}} f(\boldsymbol{x}), \quad \text{where} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + r(\boldsymbol{x}), \tag{2.10}$$

where we use function g to denote the data-fidelity term, and r to denote the regularizer with the strength parameter τ absorbed. We adopt this widely-used notations in (2.10) through the whole dissertation. In fact, by setting

$$g(\boldsymbol{x}) = -\log(p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})) \quad \text{and} \quad r(\boldsymbol{x}) = -\log(p_{\boldsymbol{x}}(\boldsymbol{x})), \tag{2.11}$$

the minimization problem (2.10) is the same as the MAP estimation shown in (2.6). The major advantage of (2.10) over MAP estimation is that it does not require the exact distribution of the prior, which in practice may not exist or be hard to interpret. The regularized optimization framework (2.10) can accommodate a variety of data-fidelity and regularization terms. For example, under the assumption of a linear forward model $\boldsymbol{A}$ and AWGN $\boldsymbol{e}$, anisotropic *total variation (TV)* regularization [18, 156] is obtained by setting

$$g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 \quad \text{and} \quad r(\boldsymbol{x}) = \tau\|\boldsymbol{D}\boldsymbol{x}\|_1, \tag{2.12}$$

where $\tau > 0$ is the regularization parameter and $\boldsymbol{D}$ is the discrete image gradient operator. Anisotropic TV corresponds to the sparsity-promoting ℓ_1-norm prior on the magnitude of the image gradient. Examples of other popular imaging regularizers include smoothness, nonnegativity, transform-domain sparsity, and self-similarity [45, 54, 62, 90, 156, 198, 202, 208].

Proximal Methods

When the objective function $f(\cdot)$ is smooth, *gradient method (GM)* such as gradient descent can be adopted to efficiently solve the minimization problem (2.10)

$$\boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \nabla f(\boldsymbol{x}^{k-1}), \tag{2.13}$$

where ∇ computes the gradient of a function and $k \geq 1$ denotes the iteration index. Nevertheless, a large number of regularizers used in the context of imaging inverse problems, including ℓ_1-norm and TV, are nonsmooth. *Proximal methods (PMs)* [144] enable efficient minimization of nonsmooth functions, without differentiating them, by using the *proximal operator*, defined as

$$\mathsf{prox}_{\tau r}(\boldsymbol{z}) \coloneqq \underset{\boldsymbol{x} \in \mathbb{R}^n}{\mathsf{arg\,min}} \left\{ \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{z}\|^2 + \tau r(\boldsymbol{x}) \right\}, \tag{2.14}$$

for any proper, closed, and convex function r [144]. Here $\tau > 0$ is a weighting parameter that controls the influence of r. PMs are widely used for solving regularized optimization problem (2.10) due to nonsmoothness of many regularizer. For example, summarized in Algorithm 1 and Algorithm 2 are two known PMs that include the proximal operator in solving minimization problem (2.10) with nonsmooth regularizer terms. We introduces the details these two algorithms below.

Proximal gradient method (PGM). PGM, also known as *iterative shrinkage/thresholding algorithm (ISTA)*, is a standard iterative approach for solving regularized optimization problem formed in (2.10) [18, 19, 46, 61]. The derivation of PGM follows the *majorization-minimization (MM)* method by assuming that the data-fidelity term g in (2.10) is continuously differentiable and has a Lipschitz continuous gradient with constant $L > 0$. A function g has L-Lipschitz gradient if there exists $L > 0$ such that

$$\frac{L}{2}\|\boldsymbol{x}\|^2 - g(\boldsymbol{x}) \quad \text{is convex}, \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n. \tag{2.15}$$

By applying the first-order convexity inequality (see Definition B.5) to (2.15), we obtain the following quadratic upper bound for the data-fidelity term g

$$g(\boldsymbol{x}^+) \leq g(\boldsymbol{x}) + \nabla g(\boldsymbol{x})^\mathsf{T}(\boldsymbol{x}^+ - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x}^+ - \boldsymbol{x}\|^2, \quad \forall\, \boldsymbol{x}^+, \boldsymbol{x} \in \mathbb{R}^n. \tag{2.16}$$

The corresponding proof for the inequality above can be found in Appendix B.5. By replacing L with $1/\gamma$ where $0 < \gamma \leq 1/L$ in (2.16), we obtain the following upper bound function for the data-fidelity term g at $\boldsymbol{x} \in \mathbb{R}^n$

$$q(\boldsymbol{x}^+, \boldsymbol{x}) \coloneqq g(\boldsymbol{x}) + \nabla g(\boldsymbol{x})^\mathsf{T}(\boldsymbol{x}^+ - \boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{x}^+ - \boldsymbol{x}\|^2. \tag{2.17}$$

Function q is called a *quadratic majorizer* of the data-fidelity term g as it is a tangent to g at $\boldsymbol{x}^+ = \boldsymbol{x}$ and lies above g everywhere else, that is

$$q(\boldsymbol{x}^+, \boldsymbol{x}) \geq g(\boldsymbol{x}^+) \quad \forall\, \boldsymbol{x}^+, \boldsymbol{x} \in \mathbb{R}^n \tag{2.18}$$

$$q(\boldsymbol{x}, \boldsymbol{x}) = g(\boldsymbol{x}) \qquad \forall\, \boldsymbol{x} \in \mathbb{R}^n. \tag{2.19}$$

PGM obtains its solution to the optimization problem (2.10) by minimizing the sum of the quadratic majorizer q and the regularizer function r at iteration $k \geq 1$ as

$$\begin{aligned}
\boldsymbol{x}^k &= \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ q(\boldsymbol{x}, \boldsymbol{x}^{k-1}) + r(\boldsymbol{x}) \right\} && (2.20)\\
&= \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ g(\boldsymbol{x}^{k-1}) + \nabla g(\boldsymbol{x}^{k-1})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{x}^{k-1}) + \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{x}^{k-1}\|^2 + r(\boldsymbol{x}) \right\} && (2.21)\\
&= \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{x}^{k-1} - \gamma\nabla g(\boldsymbol{x}^{k-1})\|^2 + \gamma r(\boldsymbol{x}) \right\} && (2.22)\\
&= \mathsf{prox}_{\gamma r}(\boldsymbol{x}^{k-1} - \gamma\nabla g(\boldsymbol{x}^{k-1})). && (2.23)
\end{aligned}$$

By splitting the updates into two steps, we obtain the following widely-adopted form of PGM with the update in iteration $k \geq 1$ being

$$\begin{aligned}
\boldsymbol{z}^k &\leftarrow \boldsymbol{x}^{k-1} - \gamma\nabla g(\boldsymbol{x}^{k-1}) && (2.24a)\\
\boldsymbol{x}^k &\leftarrow \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k), && (2.24b)
\end{aligned}$$

where $\gamma > 0$ is usually referred to as the step-size parameter. When imaging system is linear and g corresponds to the least-square penalty shown in Eq. (2.12), ∇g is given as

$$\nabla g(\boldsymbol{x}) = \boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}), \tag{2.25}$$

where T denotes the transpose operation for a real-valued matrix.[2] It can be shown that when ∇g is Lipschitz continuous with constant $L > 0$, PGM converges for any $\gamma \in (0, 1/L]$ to a minimizer of the objective function f with rate $O(1/t)$, where $t \geq 1$ is the number of PGM

[2]If a matrix is complex-valued, T denotes the conjugate transpose operation.

Algorithm 1 PGM/APGM

input: $\boldsymbol{x}^0 = \boldsymbol{s}^0 \in \mathbb{R}^n$, $\gamma > 0$, $\sigma > 0$, and $\{q_k\}_{k\in\mathbb{N}}$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma\nabla g(\boldsymbol{s}^{k-1})$
  $\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k)$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$
end for

Algorithm 2 ADMM

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\boldsymbol{s}^0 = \boldsymbol{0}$, $\gamma > 0$, and $\sigma > 0$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{z}^k \leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$
  $\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$
end for

iterations [17]. PGM can be further accelerated by adopting a sequence q_k in each iteration

$$\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma\nabla g(\boldsymbol{s}^{k-1}) \tag{2.26a}$$

$$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k) \tag{2.26b}$$

$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + \frac{q_{k-1} - 1}{q_k}(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}). \tag{2.26c}$$

This accelerated version of PGM is known as *accelerated proximal gradient method (APGM)*, also referred to as *fast iterative shrinkage/thresholding algorithm (FISTA)* [16]. Particularly, the values for $\{q_k\} = 1$ and $\{q_k\} = \frac{1}{2}\left(1 + \sqrt{1 + 4q_{k-1}^2}\right)$ for all $k \geq 1$ serve as a switch between the traditional form of PGM and APGM. In this manuscript, we will use the sequence $\{q_k\}$ as a mechanism for switching between the methods. It can be shown that APGM converges to the minimizer of the objective function f with rate $O(1/t^2)$ for any step-size $\gamma \in (0, 1/L]$, which has been proven to be optimal for gradient-based methods [137]. A summary of PGM and APGM is shown in Algorithm 1.

Alternating Directions Method of Multipliers (ADMM). PGM requires the gradient computation of the data-fidelity term g. ADMM [144] is an alternative proximal method to PGM when such gradient is not accessible due to the differentiability of the data-fidelity

term. ADMM algorithm solves problems in the general form

$$\min \quad g(\boldsymbol{z}) + r(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{P}\boldsymbol{x} + \boldsymbol{Q}\boldsymbol{z} = \boldsymbol{c}, \tag{2.27}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{z} \in \mathbb{R}^m$, $\boldsymbol{P} \in \mathbb{R}^{p\times n}$, $\boldsymbol{Q} \in \mathbb{R}^{p\times m}$ and $\boldsymbol{c} \in \mathbb{R}^p$. Specifically, when $\boldsymbol{P} = \boldsymbol{I}$, $\boldsymbol{Q} = -\boldsymbol{I}$ and $\boldsymbol{c} = \boldsymbol{0}$, problem (2.27) can be simplified as

$$\min \quad g(\boldsymbol{z}) + r(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{x} = \boldsymbol{z}, \tag{2.28}$$

which is a constrained version equivalent to our optimization problem (2.10). The only difference from the general unconstrained problem (2.10) is that the variable $\boldsymbol{x}$ has been split into two parts, called $\boldsymbol{x}$ and $\boldsymbol{z}$ here, with the objective function separable across this splitting. To derive the solution of ADMM for (2.28), let's assume g and r are convex and form the *augmented Lagrangian* [139]

$$\begin{aligned} &L_\gamma(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\mu}) \\ &= g(\boldsymbol{z}) + r(\boldsymbol{x}) + \boldsymbol{\mu}^\mathsf{T}(\boldsymbol{z} - \boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{z} - \boldsymbol{x}\|^2 &\text{(2.29a)} \\ &= g(\boldsymbol{z}) + r(\boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{z} - \boldsymbol{x} + \gamma\boldsymbol{\mu}\|_2^2 - \frac{\gamma}{2}\|\boldsymbol{\mu}\|^2 &\text{(2.29b)} \end{aligned}$$

where $\gamma > 0$ is a regularization parameter and $\boldsymbol{\mu} \in \mathbb{R}^n$ is the dual variable. By introducing the *scaled dual variable* $\boldsymbol{s} \coloneqq \gamma\boldsymbol{\mu}$, we obtain the following *scaled augmented Lagrangian*

$$L_\gamma(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{s}) = g(\boldsymbol{z}) + r(\boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{z} - \boldsymbol{x} + \boldsymbol{s}\|^2 - \frac{1}{2\gamma}\|\boldsymbol{s}\|^2. \tag{2.30}$$

Following *method of multipliers* [139], the augmented Lagrangian can be minimized jointly and iteratively with respect to the two primal variables $\boldsymbol{x}$ and $\boldsymbol{z}$, where in each iteration

$k \geq 1$

$$(\boldsymbol{z}^k, \boldsymbol{x}^k) \leftarrow \underset{\boldsymbol{x},\boldsymbol{z}}{\mathsf{arg\,min}}\, L_\gamma(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{s}^{k-1}) \tag{2.31a}$$

$$\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k). \tag{2.31b}$$

The algorithm ADMM obtain its final form by simplifying the joint optimization of $\boldsymbol{x}$ and $\boldsymbol{z}$ to an alternating or sequential fashion, which accounts for the term *alternating direction*

$$\boldsymbol{z}^k \leftarrow \underset{\boldsymbol{z}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, L_\gamma(\boldsymbol{z}, \boldsymbol{x}^{k-1}, \boldsymbol{s}^{k-1}) \tag{2.32a}$$

$$\boldsymbol{x}^k \leftarrow \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, L_\gamma(\boldsymbol{z}^k, \boldsymbol{x}, \boldsymbol{s}^{k-1}) \tag{2.32b}$$

$$\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k). \tag{2.32c}$$

By adopting the definition of proximal operator defined in (2.14), we have

$$\begin{aligned}
\underset{\boldsymbol{z}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, L_\gamma(\boldsymbol{z}, \boldsymbol{x}^{k-1}, \boldsymbol{s}^{k-1}) &= \underset{\boldsymbol{z}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, g(\boldsymbol{z}) + \frac{1}{2\gamma}\|\boldsymbol{z} - \boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1}\|^2 \\
&= \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1}) \qquad (2.33a) \\
\underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, L_\gamma(\boldsymbol{z}^k, \boldsymbol{x}, \boldsymbol{s}^{k-1}) &= \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, r(\boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{z}^k - \boldsymbol{x} + \boldsymbol{s}^{k-1}\|^2 \\
&= \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k + \boldsymbol{s}^{k-1}), \qquad (2.33b)
\end{aligned}$$

which leads to the final form of ADMM algorithm, as summarized in Algorithm 2 (where $\boldsymbol{s}^0$ is initialized with $\mathbf{0}$). Different from PGM, ADMM computes the proximal $\mathsf{prox}_{\gamma g}$ instead of the gradient on the data-fidelity term g. Particularly, when $g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$ (assuming

linearity of the forward mode and AWGN), the close-form solution of $\mathsf{prox}_{\gamma g}$ can be derived as

$$\begin{aligned}\mathsf{prox}_{\gamma g}(\boldsymbol{x}) &= \underset{\boldsymbol{z}\in\mathbb{R}^n}{\arg\min}\left\{\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{x}\|^2+\frac{\gamma}{2}\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_2^2\right\}\\ &= [\boldsymbol{I}+\gamma\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}]^{-1}(\boldsymbol{x}+\gamma\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y}).\end{aligned} \tag{2.34}$$

Therefore, ADMM is known to be fast for forward operators that can be inverted efficiently [7, 126, 193], while PGM is well suited to nonlinear forward models where $\mathsf{prox}_{\gamma g}(\cdot)$ is computationally expensive to evaluate [91, 94]. Theoretically, there are also many convergence results for ADMM discussed in the literature. For example, by assuming the data-fidelity g and regularizer r are convex, closed, and proper, 1) the residue between $\boldsymbol{x}^k$ and $\boldsymbol{z}^k$ converges to 0 as $t\to\infty$, 2) dual variable $\boldsymbol{s}^k$ convergence to a dual optimal point, and 3) and the objective function (2.28) converges to its minimizers at the rate of $O(1/t)$. Although the convergence rate of ADMM is s suboptimal compared with $O(1/t^2)$ convergence rate of APGM, in practice, it often converges to modest accuracy within a few iterations [2]. Therefore, ADMM is practically favored in cases when modest accuracy is sufficient, such as large-scale problems we consider in the Chapter 6.

2.2.2 Learning-based Methods

In the past few years, DL has gained great popularity in solving imaging inverse problems due to its excellent performance (see reviews in [100, 119, 127, 140, 187]). Different from the model-based optimization approach where an explicit prior is needed, deep neural networks provide a state-of-the-art tool for representing and enforcing implicit but sophisticated structural information of images through end-to-end learning. Generally, such methods are based on training the weights of a DNN over a dataset in order for the network to produce an accurate estimate of the desired images. The traditional supervised learning for a DNN $\mathcal{F}_{\boldsymbol{\phi}}$

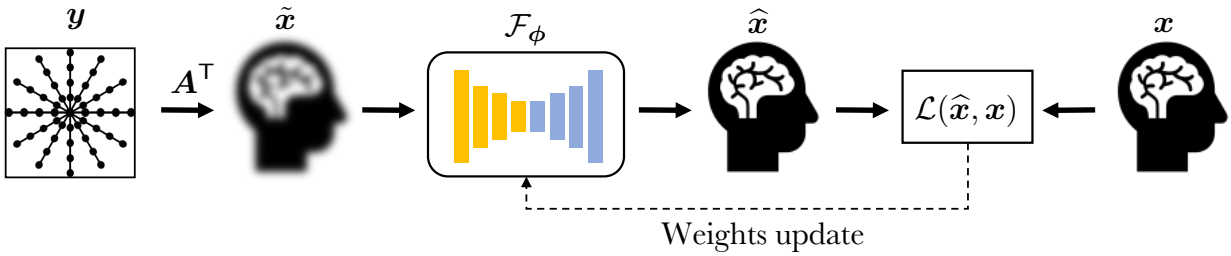

Figure 2.2: An illustration of the widely-used DL framework. The input to the network is initialized with the simple backprojection of the measurements $\boldsymbol{y}$. The weights of the network are trained by minimizing the loss between its output $\widehat{\boldsymbol{x}}$ and the corresponding ground truth image $\boldsymbol{x}$ on a large number of training samples.

characterized by its parameter $\boldsymbol{\phi}$ is formulated as an optimization problem over a training set consisting of data pairs $\{\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i\}$ as follows

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{F}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{x}}_i), \boldsymbol{x}_i), \tag{2.35}$$

where i indexes the samples in the training set of N samples. The loss function $\mathcal{L}$ measures the discrepancy between the reconstruction $\widehat{\boldsymbol{x}}_i = \mathcal{F}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{x}}_i)$, which is generated by the neural network from low-quality images $\tilde{\boldsymbol{x}}_i$, and the ground-truth $\boldsymbol{x}_i$. Typical choices for $\mathcal{L}$ include the ℓ_1 and the ℓ_2 distances. For a linear forward model, the low-quality image $\tilde{\boldsymbol{x}}_i$ can be $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y}_i$, which is a simple backprojection from measurements. This minimization problem can be solved by using stochastic gradient-based optimization algorithms such as Adam [25, 99]. Once the optimal set of parameters $\boldsymbol{\phi}^*$ are learned by minimizing the optimization problem on the training dataset, which consists of many samples, the well-trained network $\mathcal{F}_{\boldsymbol{\phi}^*}$ can be applied to unseen data for image reconstruction tasks. Figure 2.2 shows a widely-used end-to-end DL scheme for solving inverse problems. Many DNN architectures were designed targeting at different imaging applications. Here we introduce two that are close to our work presented in the following chapters of this dissertation, known as U-Net and DnCNN.

U-Net. U-Net is a well-known CNN named after its U-shaped architecture. U-net was first proposed in [155] for image segmentation. Its design was later adopted in [87] for image reconstruction, where it achieved the state-of-the-art performance in the context of X-Ray CT imaging. The U-shaped architecture of U-Net results from a combination of a contracting path and an expanding path, where the contracting path relies on a repeated usage of convolutions (conv), each followed by a rectified linear unit (ReLU) and a max pooling operation to encode the spatial information, and the expanding path uses sequence of up-convolutions and concatenations with high-resolution features from the contracting path to increase the resolution of the output. The spatial information is reduced while feature information is increased during the contracting path, making the effective size of its filters in the middle layers larger than that of the early and late layers [87]. Such multi-scale structure leads to a large receptive field of the CNN that has been shown to be effective for removing globally spread imaging artifacts typical in medical imaging [73, 102].

DnCNN. DnCNN is a popular CNN architecture for image denoising, for which the original design can be found in [216]. DnCNN consists of a sequence of layers, where the first is a conv layer followed by ReLU, the middle ones are the combination of conv, batch normalization (BN) and ReLU, and the last is a simple conv. Different from U-Net, DnCNN does not change the spatial resolution of inputs across layers. DnCNN is trained using the strategy of residue learning , where its outputs are the artifacts in the inputs, and the clean predictions are obtained by subtracting those artifacts from the inputs. After it was proposed, DnCNN has gained significant popularity for its simple implementation and start-of-the-art performance in various denoising tasks.

Discussion

So far, we have reviewed some key ideas of the two approaches in solving inverse problems—classical model-based methods and recent learning-based methods. In particular, model-based methods look into the problem by integrating forward models and handcrafted designed explicit prior knowledge. While learning-based methods attempt to use the representation power of deep neural networks to learn the sophisticated structural information and statistical priors through end-to-end training. The two approaches have all established great success in various imaging tasks, showing distinct advantages from different perspectives. I will now briefly discuss the pros and cons provided by those two approaches and from there, highlight the benefits of their integration in advanced algorithm design.

On the one hand, the model-based optimization methods can explicitly guarantee the data-consistency and desired properties in the reconstruction by taking advantages of priors and forward models regarding the unknown image. However, the design of a good prior term is not easy as it is highly dependent on the estimation of distribution of the unknown image. Meanwhile, the iterative mechanism in solving the objective function (2.10) limits their application in scenarios with large-scale data and high speed computing requirements. Those drawbacks therefore reduce the popularity of such model-based optimization approaches in solving modern complex imaging problems.

On the other hand, DL shows its advantages over the traditional model-based approaches as it shifts the prior design from explicit human estimation to an implicit manner driven by data. Powered by a large number of data, this end-to-end learning approach provides a more flexible, sophisticated, and data-adaptive tool for characterizing imaging priors. This representation power has been widely demonstrated in various image reconstruction applications, showing the state-of-the-art performance [119, 127, 140, 187]. However, as the information of the

forward model is generally not utilized in the pipeline, one typical loss of such learning scheme is the data-consistency guarantee, which degrades the reliability and robustness of the reconstruction. Meanwhile, the common need for a huge amount of training data also limits its application in some scenarios such as medical imaging where paired training data is usually inaccessible.

Therefore, it can be seen that model-based methods benefit most from their utilization of forward models but are limited to the prior design, while learning-based methods benefit most from the advanced prior representation but suffer from the missing of imaging models. Those two approaches naturally compliment to each other, resulting in an appeal for their integration. In the next section, we introduce some advances that realize such integration, completing the background of this dissertation.

2.3 Integrating Models and Learning for Imaging

As discussed above, model-based optimization methods and end-to-end learning-based methods offer complimentary strategies for handling image reconstruction problems. Modern complex and large-scale image reconstruction problems require fast and reliable methods that can combine the benefits of both. In this section, we will introduce some advances in such combination. Note that our goal here is not to include all the existing methods but to highlight some key ones related to the dissertation. We refer the interested readers to [23, 64, 82, 83, 95, 106, 130, 131, 145, 162, 171, 174] for a boarder range of methods and discussion.

2.3.1 Using Learning Priors inside Model-based Methods

One way to integrate models and learning is to replace the handcrafted priors in the model-based approaches by deep-learned priors. By leveraging the power of deep learning as priors,

such model-based methods can be greatly boosted. In this section, we introduce the following two known approaches originated from this idea.

Plug and Play Priors (PnP). Plug-and-play priors (PnP), first proposed in [186], is a methodology for regularized image reconstruction that specifies the prior through an image denoiser. It is motivated by the observation that the proximal operator (2.14) can be mathematically interpretative an MAP image denoiser for AWGN. To see this interpretation, let's consider the following AWGN denoising problem with variance of τ

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e} \quad \text{where} \quad \boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \tau\mathbf{I}). \tag{2.36}$$

Following the similar derivation in Eq. (2.7) and Eq. (2.8), we can establish the the MAP estimation for $\boldsymbol{x}$ as

$$\widehat{\boldsymbol{x}}_{\text{MAP}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \tau r(\boldsymbol{x}) \right\} = \mathsf{prox}_{\tau r}(\boldsymbol{y}), \tag{2.37}$$

which is the proximal operator involved in proximal algorithms. The PnP methodology, proposed to replace the the proximal operator $\mathsf{prox}_{\tau r}(\cdot)$, within a proximal algorithm, with a more general image denoiser $\mathsf{D}(\cdot)$, such as BM3D [44] or DnCNN [216]. Two popular PnP algorithm, PnP-PGM (also referred to as PnP-ISTA) [91] and PnP-ADMM [186], which are originated from proximal-based algorithm PGM and ADMM, are summarized in Algorithm 3 and Algorithm 4, where in analogy to $\mathsf{prox}_{\tau r}(\cdot)$ we also introduce the parameter $\sigma > 0$ to characterize the denoising strength of the denoiser $\mathsf{D}_\sigma(\cdot)$. Unlike traditional regularized optimization, PnP does not require the prior to be expressible in the form of a regularization function. This flexibility enables PnP algorithms to exploit the most effective image denoisers, especially powerful denoising CNNs, leading to their state-of-the-art performance in various imaging tasks [31, 35, 91, 141, 166, 177, 217]. It is worth briefly mentioning that learned

Algorithm 3 PnP-PGM/PnP-APGM

input: $\boldsymbol{x}^0 = \boldsymbol{s}^0 \in \mathbb{R}^n$, $\gamma > 0$, $\sigma > 0$, and $\{q_k\}_{k\in\mathbb{N}}$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$
  $\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k)$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$
end for

Algorithm 4 PnP-ADMM

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\boldsymbol{s}^0 = \boldsymbol{0}$, $\gamma > 0$, and $\sigma > 0$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{z}^k \leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$
  $\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$
end for

denoisers have also been adopted for a class of algorithms in compressive sensing known as *approximate message passing (AMP)* [64, 130, 131, 174]. The key difference of PnP from AMP is that it does not assume random measurement operators. A recent line of work has also investigated the recovery and convergence guarantees for priors specified by *generative adversarial networks (GANs)* [23, 82, 83, 106, 145, 162]. PnP does not seek to project its iterates to the range of a GAN, instead it directly uses the output of a simple AWGN denoiser to improve the estimation quality. This simplifies the training and application of learned priors within the PnP methodology.

Regularized by Denoising (RED). Note the use of an arbitrary denoiser inside PnP immediately results in the missing of the objective function as it is usually hard to relate a denoiser as a regularizer function. The RED framework, first proposed in [153], is an alternative scheme where the denoiser $\mathsf{D}_\sigma(\cdot)$ can sometimes lead to an explicit regularization function. The widely-used gradient-method-based RED algorithm *RED-GM* (also called *steepest descent variant of RED (RED-SD)*) updates its iteration as

$$\boldsymbol{x}^k = \boldsymbol{x}^{k-1} - \gamma \mathsf{G}(\boldsymbol{x}^{k-1}) \quad \text{with} \quad \mathsf{G}(\boldsymbol{x}) \coloneqq \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x})), \tag{2.38}$$

where $\gamma > 0$ is the step size and $\tau > 0$ is a regularization parameter that balances the strength between the gradient of the data-fidelity term and the noise residual. RED-GM algorithms seek a fixed point $\boldsymbol{x}^*$ that satisfies

$$\mathsf{G}(\boldsymbol{x}^*) = \nabla g(\boldsymbol{x}^*) + \tau(\boldsymbol{x}^* - \mathsf{D}_\sigma(\boldsymbol{x}^*)) = \mathbf{0}. \tag{2.39}$$

Equivalently, $\boldsymbol{x}^*$ satisfies

$$\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \mathsf{G}(\boldsymbol{x}) = \mathbf{0}\}. \tag{2.40}$$

When the denoiser is locally homogeneous and has a symmetric Jacobian [149, 153], the *noise residue* term

$$\mathsf{R}(\boldsymbol{x}) = \tau(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x})) \tag{2.41}$$

corresponds to the gradient of the RED regularizer

$$r(\boldsymbol{x}) = (\tau/2)\boldsymbol{x}^\mathsf{T}(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x})), \tag{2.42}$$

which enables a simple interpretation of RED as an instance of the regularized optimization (2.10). For the reference, we summarize these two conditions on the denoisers as follows:

1) **(Local) Homogeneity [153].** A denoiser applied to a positively scaled image should result in a scaled version of the original image, that is, for any (small) $c \geq 0$

$$\mathsf{D}_\sigma(c \cdot \boldsymbol{x}) = c \cdot \mathsf{D}_\sigma(\boldsymbol{x}).$$

This condition actually implies

$$\mathsf{D}_\sigma(\boldsymbol{x}) = \nabla \mathsf{D}_\sigma(\boldsymbol{x})\boldsymbol{x}.$$

2) **Symmetric Jacobian [149].** Denoiser D_σ has a symmetric Jacobian, that is

$$\nabla \mathsf{D}_\sigma(\boldsymbol{x}) = [\nabla \mathsf{D}_\sigma(\boldsymbol{x})]^\mathsf{T}$$

It was shown that many popular denoisers (e.g. BM3D, TNRD, and DnCNN) cannot meet these conditions [149], which consequently breaks the connection between the noise residue term in Eq. (2.41) and the explicit regularizer function in Eq. (2.42). Nevertheless, the gradient-based updates of RED-SD in (2.38) still lead to an interpretable fixed-point solution illustrated in Eq. (2.39), making RED a popular framework for image reconstruction. The excellent performance of RED together with learned CNN denoisers has been reported in a broad range of imaging applications such as super-resolution, phase retrieval, and compressed sensing [129, 170].

2.3.2 Including Imaging Models inside Learning-based Methods

As an alternative to PnP/RED, the combination of deep learning and model-based optimization can also be realized by merging the model information into the learning approach. This idea has been discussed in various work [3, 115, 182, 214]. We use the following two approaches as examples to elaborate this idea.

Deep image prior (DIP). DIP [185] is a recent regularization framework that uses the architecture of the CNN itself as an prior for image reconstruction without data-driven training. Given a CNN $\mathcal{F}_{\boldsymbol{\phi}}$ characterized by its parameter $\boldsymbol{\phi}$, DIP optimizes $\boldsymbol{\phi}$ by minimizing

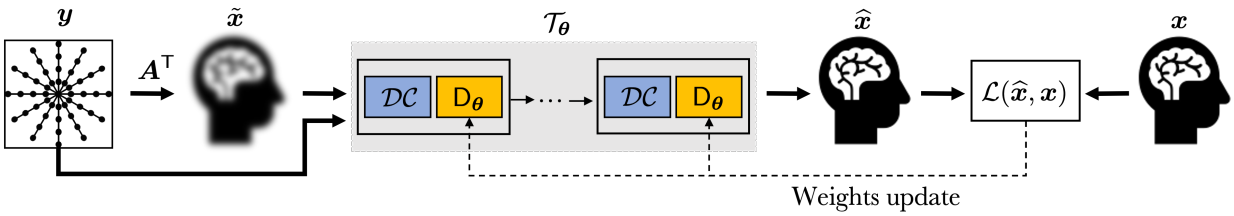

Figure 2.3: An illustration of the widely-used DU framework $\mathcal{T}_{\boldsymbol{\theta}}$, which contains multiple recursive layers and each layer contains a data-consistency module $\mathcal{DC}$ and a CNN module $\mathsf{D}_{\boldsymbol{\theta}}$. Note here we assume all $\mathsf{D}_{\boldsymbol{\theta}}$ across layers share the same architecture and weights but they can also adopt different architectures. The input to the network are the measurements $\boldsymbol{y}$ and their simple backprojection $\boldsymbol{A}^\mathsf{T}\boldsymbol{y}$ to the image domain. The weights of the CNNs $\mathsf{D}_{\boldsymbol{\theta}}$ are trained by minimizing the loss between the output $\widehat{\boldsymbol{x}}$ and the corresponding ground truth image $\boldsymbol{x}$ on many training samples.

the loss function $\mathcal{L}(\mathcal{A}(\mathcal{F}_{\boldsymbol{\phi}}(\boldsymbol{z}), \boldsymbol{y})$, where $\boldsymbol{z}$ is a randomly vector sampled from Gaussian distribution and $\boldsymbol{y}$ is the noisy measurements illustrated in the forward model (2.1). The intuition behind DIP is that natural images can be well represented by CNNs, which is not the case for the random noise and certain other image degradations. DIP was shown to achieve remarkable performance on a number of image reconstruction tasks [65, 112, 185]. Note the success of DIP is benefited from not only the prior presented by the CNN, but also forward operator $\mathcal{A}$ inside the loss function that enforces the data-consistency. A similarly idea has also been adopted in the recent work of *self-supervised learning*, where a model is trained using a *pretext* (or *auxiliary*) task, but tested on the actual *desired* task [38, 72, 98, 116, 161, 182, 206]. Such learning can also enabled by utilizing the information of forward operator inside the loss function. We will illustrate more details of this idea on a specific application in Chapter 8.

Deep Unfolding/Unrolling (DU). DU is a combination of DL and model-based iterative algorithm that interprets the iterations of an image recovery algorithm as layers of a neural network and trains it end-to-end in a supervised fashion [1, 3, 75, 79, 134, 205, 214]. A typical DU architecture $\mathcal{T}_{\boldsymbol{\theta}}$ is shown in Fig. 2.3, where it contains multiple recursive layers and

each layer contains a data-consistency module $\mathcal{DC}$ and a CNN module $\mathsf{D}_{\boldsymbol{\theta}}$ with $\boldsymbol{\theta}$ being the trainable parameters. Such unfolding methods have been shown to be effective in a number of problems [21, 79]. Many PnP/RED algorithms that combine the forward operator and the DL prior (discussed in Section 2.3) have also been turned into DU architectures by truncating the algorithm to a fixed number of iterations, producing high-quality results within fewer iterations. But unlike in PnP/RED, the CNN component $\mathsf{D}_{\boldsymbol{\theta}}$ in DU is trained jointly with the imaging model, leading to an image prior optimized for a given inverse problem. In this dissertation, with an emphasize on model-based deep learning, we will develop several DU algorithms for different imaging tasks.

2.4 Summary

In this chapter, we introduced the background of computational imaging, including how to model the imaging problems as inverse problems and how to statistically interpret such problems. We reviewed the classical model-based and popular learning-based methods for solving inverse problems and highlighted the benefits of their integration. We then introduced some advanced algorithms that successfully realize such integration. In the following chapters, with an emphasis on model-based deep learning methods, we will introduce our work in integrating models and learnings for computational imaging. Our contribution includes both the development of novel algorithms and rigorous theoretical analysis. As a reference, we summarize and provide useful mathematical materials including the definitions and propositions in monotone operator theory that are related to our analysis in the rest of the dissertation in Appendix B.

Part II

Statistical Interpretation for Plug-and-Play Priors

Chapter 3

Overview

Plug-and-play priors (PnP) is a simple yet flexible methodology for imposing statistical priors without explicitly forming an objective function [166, 186]. As what discussed in Section 2.3, PnP algorithms alternate between imposing data consistency by minimizing a data-fidelity term and imposing a statistical prior by applying an AWGN denoiser. By adopting the advanced denoisers such as the ones trained with DNNs, PnP integrates physical and learned models thus achieving its state-of-the-art performance in a variety of applications [4, 49, 190, 215, 217]. However, the use of general denoisers blurs the connection between PnP and classical regularized optimization, and brings new challenges for statistical interpretation of PnP. In this chapter, we review PnP and discuss its theoretical challenges.

3.1 Recap of PnP

Let's recall the key idea and derivation of PnP introduced in Section 2.3. As discussed in Chapter 2, image formation is naturally posed as an inverse problem, which is often ill-posed.

Regularized optimization is a widely adopted framework for dealing with such ill-posed inverse problems by taking advantage of prior information regarding the unknown image. Since imaging priors are often nondifferentiable, proximal algorithms [144], such as PGM (also referred to as ISTA) and ADMM (see Algorithm 1 and Algorithm 2), are extensively used in image reconstruction. These algorithms avoid differentiating the regularizer by using the proximal operator $\mathsf{prox}_{\tau r}(\cdot)$ (see definition in Eq. (2.14)) on the prior/regularizer function r [144]. The observation that the proximal operator is an image denoiser for AWGN prompted the development of PnP [186], where the operator $\mathsf{prox}_{\tau r}(\cdot)$, within a proximal algorithm, is replaced with a more general image denoiser $\mathsf{D}_{\sigma}(\cdot)$

$$\underbrace{\text{PGM / ADMM}}_{\text{Proximal algorithms}} \xrightarrow{\text{replace } \mathsf{prox}_{\tau r}(\cdot) \text{ with } \mathsf{D}_{\sigma}(\cdot)} \underbrace{\text{PnP-PGM / PnP-ADMM}}_{\text{PnP algorithms}}. \tag{3.1}$$

The two known PnP variants derived from PGM and ADMM are summarized in Algorithm 3 and Algorithm 4, known as PnP-PGM (also refereed to as PnP-ISTA) ands PnP-ADMM. Recent results have shown that by using advanced image denoisers in iterative image reconstruction, PnP algorithms achieve state-of-the-art performance in many imaging problems [35, 91, 141, 166, 177, 217].

3.2 Theoretical Challenges

PnP algorithms have been successfully combined with many powerful denoisers, such as DnCNN [216], for exploiting learned imaging priors while enforcing fidelity to the measured data [128, 158, 168, 181, 200]. However, the flexibility of using powerful denoisers also brought some new challenges compared to proximal methods. For example, we cannot interpret the iterates of PnP as the minimization of an objective function as not every denoiser is expressible in the form of a regularizer. Without an objective function, the convergence analysis of PnP

is unclear. Also for many denoisers, we do no have tunable parameters that can control their influence within PnP. In the table below, we summarize these benefits and challenges.

Benefits	**Challenges**
1. No need for the handcrafted image priors and no need for these priors to be expressible in the form of a regularization function.	1. Missing connection to an objective function and the corresponding convergence analysis.
2. Excellent or even state-of-the art performance in many imaging tasks.	2. Missing control on the relative strength between the prior imposed by the denoisers and the data fidelity.

To address these challenges, in the following two chapters, we propose methods and conduct analysis for better understanding PnP with different type of denoisers. We present: 1) a denoiser scaling technique that boosts the performance of PnP, and 2) a theoretical analysis that establishes the convergence and statistical interpretation for PnP.

Chapter 4

Boosting the Performance of Plug-and-Play Priors via Denoiser Scaling

The use of image denoisers as imaging priors brings a lot of flexibility to PnP. For example, unlike traditional regularized optimization, PnP does not require the prior to be expressible in the form of a regularization function. This flexibility enables PnP algorithms to exploit the most effective image denoisers, leading to their state-of-the-art performance in various imaging tasks. However, many powerful denoisers, such as the ones based on CNNs, do not have tunable parameters that would allow controlling their influence within PnP. To address this issue, in this chapter, we introduce a *scaling parameter* that adjusts the magnitude of the denoiser input and output. We theoretical justify the denoiser scaling from the perspectives of proximal optimization, statistical estimation, and consensus equilibrium. Finally, we provide numerical experiments demonstrating the ability of denoiser

scaling to systematically improve the performance of PnP for denoising CNN priors that do not have explicitly tunable parameters.[3]

4.1 Introduction

The *scaling parameter* we introduce for PnP is independent from the intrinsic parameters of the denoiser or iterative algorithms. For denoisers based on CNNs, this parameter additionally avoids training several network instances at multiple noise levels and therefore potentially leads to the optimal performance. We summarize our key contributions on this topic are as follows:

- We introduce a new *denoiser scaling* technique that simply scales the denoiser input by a positive constant and its output by the inverse of the same constant. The technique is broadly applicable to all PnP algorithms, and provides a mechanism to adjust the denoiser strength in a way that is independent of traditional approaches.
- We present a detailed theoretical justification of denoiser scaling for several classes of denoisers. We show that, unlike the intrinsic parameters of the denoiser, the new scaling parameter can be explicitly related to the trade-off between the data-fidelity and the prior.
- We extensively validate denoiser scaling by showing its potential to address the suboptimal performance of denoising CNNs within PnP algorithms. Our results show that denoiser scaling is a simple yet effective approach for boosting the performance of CNN priors within PnP.

[3]This chapter is based on our paper [200].

Algorithm 5 Scaled PnP-PGM/PnP-APGM

input: $\boldsymbol{x}^0 = \boldsymbol{s}^0 \in \mathbb{R}^n$, $\gamma > 0$, $\sigma > 0$, $\mu > 0$ and $\{q_k\}_{k\in\mathbb{N}}$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$
  $\boldsymbol{x}^k \leftarrow \mathsf{D}_\mu(\boldsymbol{z}^k)$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$
end for

Algorithm 6 Scaled PnP-ADMM

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\boldsymbol{s}^0 = \boldsymbol{0}$, $\gamma > 0$, $\sigma > 0$ and $\mu > 0$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{z}^k \leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$
  $\boldsymbol{x}^k \leftarrow \mathsf{D}_\mu(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$
end for

4.2 Background

To explain the idea of our scaling technique, let's consider the recovery of an unknown image $\boldsymbol{x} \in \mathbb{R}^n$ from noisy measurements $\boldsymbol{y} \in \mathbb{R}^m$ for a linear system defined in (2.3). Following the discussion in Section 2.1.2, a common approach is to formulate the problem as regularized optimization, expressed as an optimization problem of the form

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, f(\boldsymbol{x}) \quad \text{with} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + \lambda r(\boldsymbol{x}), \tag{4.1}$$

where g is the data-fidelity term, r is the regularizer, and $\lambda > 0$ is a regularization parameter that adjusts their relative strengths. And by setting

$$g(\boldsymbol{x}) = -\mathsf{log}(p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})) \quad \text{and} \quad r(\boldsymbol{x}) = -(1/\lambda)\mathsf{log}(p_{\boldsymbol{x}}(\boldsymbol{x})),$$

where $p_{\boldsymbol{y}|\boldsymbol{x}}$ denotes the likelihood function characterizing the imaging system and $p_{\boldsymbol{x}}$ denotes a probability distribution over $\boldsymbol{x}$, one obtains the classical MAP estimator discussed in 2.1.2. Proximal algorithms [144] enable efficient minimization of nonsmooth functions, without differentiating them, by using the proximal operator $\mathsf{prox}_{\tau r}(\boldsymbol{z})$ defined in Eq. (2.14), where

$\tau > 0$ is a scaling parameter that controls the influence of r. Note that the proximal operator can be interpreted as a MAP image denoiser for AWGN with variance of τ.

The observation that the proximal operator is an image denoiser for AWGN prompted the development of PnP [186], where the operator $\mathsf{prox}_{\tau r}(\cdot)$, within a proximal algorithm, is replaced with a more general image denoiser $\mathsf{D}(\cdot)$, such as BM3D [44] or DnCNN [216]. In traditional proximal optimization, the scaling parameter τ of the proximal operator (2.14) is directly related to the regularization parameter λ. For example, by setting $\tau = \gamma\lambda$ within traditional ADMM or PGM, one minimizes the objective function in (4.1). However, this explicit relationship between the scaling parameter and the regularization parameter is lost in the context of more general denoisers. Since some popular image denoisers, such as BM3D, accept a parameter corresponding to the noise variance, current PnP algorithms generally treat it as a proxy for the regularization parameter. For example, if σ in the notation for $\mathsf{D}_\sigma(\cdot)$ in Algorithm 4 and Algorithm 3 denotes the standard deviation parameter accepted by the denoiser, the common strategy is to set it as $\sigma = \sqrt{\gamma\lambda}$ [35]. However, this strategy does not work with all denoisers, since some do not have a dedicated parameter for noise variance. In particular, many denoising CNNs do not have a parameter for the noise standard deviation, which is often addressed by training multiple neural nets at different noise levels and using σ to select the most suitable one for a given problem. The denoiser scaling technique introduced in the next section enables the control of the regularization strength for denoisers that have no intrinsic parameters analogous to σ.

4.3 Proposed Method

We introduce denoiser scaling for explicitly controlling the regularization strength in PnP. Remarkably, the technique can be theoretically justified from multiple perspectives, including

from that of proximal optimization, statistical estimation, and consensus equilibrium [31]. Our experimental results in Section 6.5 corroborate the ability of denoiser scaling to control the relative influence of the denoiser.

4.3.1 Denoiser Scaling

Consider an image denoiser $\mathsf{D} : \mathbb{R}^n \to \mathbb{R}^n$, where we omit the parameter σ from the notation as it is not available for all denoisers. We define the scaled denoiser as

$$\mathsf{D}_\mu(\boldsymbol{z}) \coloneqq (1/\mu)\mathsf{D}(\mu\boldsymbol{z}), \quad \boldsymbol{z} \in \mathbb{R}^n, \tag{4.2}$$

where we will refer to the parameter $\mu > 0$ as the *denoiser scaling parameter*. The scaled PnP algorithms equipped with such denoiser scaling technique is summarized in Algorithm 5 Algorithm 6, where in in line 4 the scaling parameter μ is adopted. Note that the scaling in (4.2) is *complimentary* to any intrinsic parameter of $\mathsf{D}(\cdot)$. For example, if the underlying denoiser $\mathsf{D}(\cdot)$ additionally accepts σ as a parameter, $\mathsf{D}_\mu(\cdot)$ will also accept the same parameter. However, as discussed below, the parameter μ will enable control of the strength of regularization when σ is not available.

4.3.2 Proximal Operator Denoisers

We first consider the case in which $\mathsf{D}(\cdot)$ is an *implicit* proximal operator of some *unknown* r, which is a common interpretation for PnP algorithms [166]. For convenience, we assume r to be closed, convex, and proper [144]; however, this assumption can be dropped as long as $\mathsf{prox}_r(\cdot)$ is well defined for the given r. We state the following result for the scaled denoisers.

Proposition 4.1. *Suppose* $\mathsf{D}(\boldsymbol{z}) = \mathsf{prox}_r(\boldsymbol{z})$*, where* r *is a closed, convex, and proper function. Then, we have*

$$\mathsf{D}_\mu(\boldsymbol{z}) = \mathsf{prox}_{\mu^{-2}r(\mu\cdot)}(\boldsymbol{z}), \quad \boldsymbol{z} \in \mathbb{R}^n. \tag{4.3}$$

Proof. See Appendix C.1.

Proposition 4.1 indicates that by scaling an implicit proximal operator, one directly adjusts the strength of regularization via the scaling of the regularizer by $1/\mu^2$ and of the input to r by μ. While the relationship between μ and the regularization parameter in front of r is not linear, μ still provides an explicit mechanism to tune the denoiser. If the denoiser corresponds to a 1-*homogeneous* regularizer r, we have $r(\mu\cdot) = \mu \cdot r(\cdot)$, which directly implies

$$\mathsf{D}_\mu(\boldsymbol{z}) = \mathsf{prox}_{\mu^{-1}r}(\boldsymbol{z}), \quad \boldsymbol{z} \in \mathbb{R}^n. \tag{4.4}$$

This means that the denoiser scaling becomes equivalent to tuning the traditional regularization parameter in regularized optimization. Since any norm and semi-norm is 1-homogeneous, the strength of many implicit and explicit regularizers, such as the ℓ_1-norm or TV penalty, can be directly adjusted through denoiser scaling. This equivalence is confirmed numerically for the TV denoiser in Section 6.5.

4.3.3 Mean Squared Error Optimal Denoisers

We now consider the case of a denoiser that performs the minimum mean-squared error (MMSE) estimation of a vector from its AWGN corrupted version [71, 93, 96]. MMSE denoisers are optimal with respect to the ubiquitous image-quality metrics, such as signal-to-noise ratio (SNR). Additionally, many popular denoisers (such as BM3D and certain denoising

CNNs) are often interpreted as *empirical* MMSE denoisers. We state the following result for the scaled MMSE denoisers.

Proposition 4.2. *Suppose* $\mathsf{D}(\cdot)$ *computes the MMSE solution of the following denoising problem*

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{n} \quad \textit{with} \quad \boldsymbol{x} \sim p_{\boldsymbol{x}} \quad \textit{and} \quad \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}),$$

where $\boldsymbol{I}$ *is an identity matrix. Then, the denoiser* (4.2) *computes the MMSE solution of*

$$\boldsymbol{z} = \boldsymbol{u} + \boldsymbol{e} \quad \textit{with} \quad \boldsymbol{u} \sim p_{\boldsymbol{x}}(\mu \cdot) \quad \textit{and} \quad \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \mu^{-2}\boldsymbol{I}).$$

Proof. See Appendix C.2.

Similarly to Proposition 4.1, Proposition 4.2 indicates that the scaling of the MMSE denoiser enables the direct control of the strength of regularization via the scaling of the noise variance by $1/\mu^2$ and of the input to the prior $p_{\boldsymbol{x}}$ by μ. If $p_{\boldsymbol{x}}$ assigns equal probabilities to all images that have undergone rescaling, we have $p_{\boldsymbol{x}}(\mu \cdot) = p_{\boldsymbol{x}}(\cdot)$, which implies that the scaled denoiser directly adjusts the variance of AWGN in the MMSE estimation.

4.3.4 Consensus Equilibrium Interpretation

Consensus equilibrium (CE) [31] is a recent framework for interpreting the solution of regularized optimization methods in terms of a set of balancing equations for the forward and prior models, without an explicit cost function The solutions obtained by both PnP-ADMM and PnP-PGM can be expressed in terms of the same set of CE equations

$$\boldsymbol{x} = \mathsf{F}(\boldsymbol{x} + \boldsymbol{s}) \tag{4.5a}$$

$$\boldsymbol{x} = \mathsf{D}(\boldsymbol{x} - \boldsymbol{s}), \tag{4.5b}$$

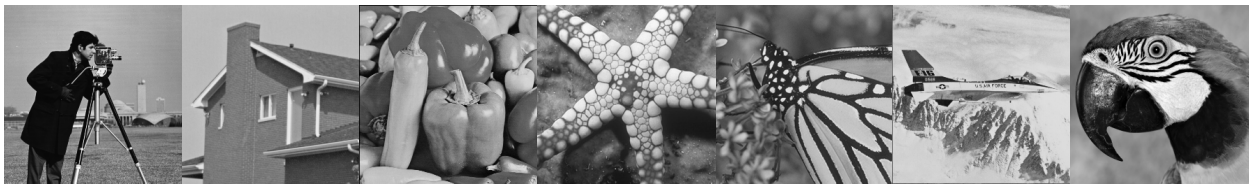

Figure 4.1: Test images used for the quantitative performance evaluation. From left to right: *Cameraman*, *House*, *Pepper*, *Starfish*, *Butterfly*, *Plane*, *Parrot*.

where $\mathsf{F}(\cdot) \coloneqq \mathsf{prox}_{\gamma g}(\cdot)$ and $\gamma > 0$ is an algorithm tuning parameter. We use the CE framework to state the following result for the scaled denoisers.

Proposition 4.3. *Let g be a smooth, convex function and $\mathsf{D}(\cdot)$ be a continuous denoiser. The fixed point $(\boldsymbol{x}, \boldsymbol{s})$ of PnP-ADMM and PnP-PGM for the scaled denoiser satisfies*

$$\begin{cases} \mu\boldsymbol{x} = \mathsf{prox}_{(\gamma\mu^2)g(\cdot/\mu)}\left(\mu\boldsymbol{x} + \boldsymbol{s}\right) \\ \mu\boldsymbol{x} = \mathsf{D}\left(\mu\boldsymbol{x} - \boldsymbol{s}\right). \end{cases}$$

Proof. See Appendix C.3.

This result establishes a direct relationship between the scaling parameter $\mu > 0$ and the rescaling of g with respect to the denoiser. Note that while Proposition 4.1 and 4.2 discuss the impact of denoisier scaling on the implicit prior, Proposition 4.3 highlights its impact on the relative influence between the denoiser and the data-fidelity term via the weighting in front of g. Since our only assumption is the continuity of the denoiser, Proposition 4.3 also relaxes the assumptions on the denoiser. While the relationship between μ and the set of equilibrium points is nontrivial, the proposition implies that one can still adjust the amount of regularization by tuning μ.

Table 4.1: SNR performances of several image denoisers at different noise levels.

Input SNR	TV		BM3D			DnCNN*		
	Scaled	Optimized	Unscaled	Scaled	Optimized	Unscaled	Scaled	Optimized
15 dB	22.77	22.77	16.58	24.55	24.51	16.49	24.30	24.21
20 dB	25.80	25.80	25.58	27.34	27.34	24.83	27.42	27.23
25 dB	29.09	29.09	29.61	30.43	30.43	29.83	30.51	30.35
30 dB	32.66	32.66	33.49	33.63	33.73	33.54	33.82	33.72

4.4 Numerical Validation

In this section, we demonstrate the ability of denoiser scaling to boost the performance of PnP. Our experiments consider imaging inverse problems of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{y} \in \mathbb{R}^m$ denotes the measurements, $\boldsymbol{e} \in \mathbb{R}^m$ is a vector of AWGN with zero mean and standard deviation σ, and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ denotes the forward operator. We focus on three inverse problems: image denoising, subsampled Fourier optimization, which commonly used in magnetic resonance imaging (MRI), and single image super-resolution (SR), where the forward models correspond to the identity matrix, radially subsampled two-dimensional Fourier transform, and blurring-downsampling operator, respectively. For each simulation, the measurements are corrupted with AWGN quantified through the input signal-to-noise ratio (SNR). We will consider three denoisers for PnP: TV, BM3D, and our own residual DnCNN*. DnCNN* is a simplified variant of the standard DnCNN [216], where our simplifications correspond to the removal of batch-normalization layers and reduction in the total number of layers (see Appendix C.4 for details). This simplification reduces the computational cost of applying the denoiser across multiple PnP iterations. We use σ to parameterize BM3D and DnCNN*. For BM3D σ represents the parameter of the denoiser representing the standard deviation of noise and for DnCNN* it represents the standard deviation of the noise used for training the CNN. We follow [40] and use 400 images of size 180×180 to train three

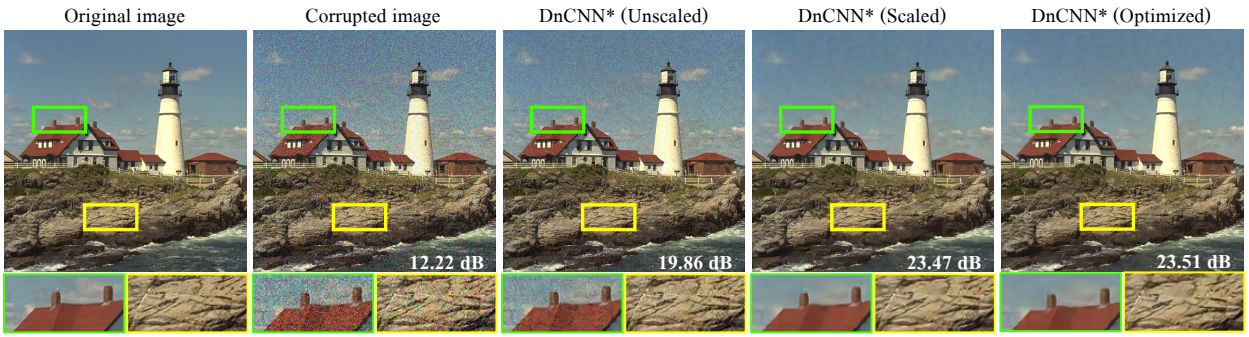

Figure 4.2: Illustration of denoiser scaling on the color image *Lighthouse*. The noise levels is $\sigma = 30$. DnCNN* (Optimized) corresponds to the CNN denoiser trained using the correct noise levels. On the other hand, DnCNN* (Unscaled) and DnCNN* (Scaled) use the same CNN trained at a mismatched noise level of $\sigma = 20$. By adjusting μ, DnCNN* trained at a suboptimal σ can be made to match the performance of DnCNN* trained using the correct noise level.

DnCNN* instances, on natural grayscale images, for the removal of AWGN at three noise levels, $\sigma \in \{1, 5, 10\}$. For some of the experiments, we also use a constrained variant of BM3D, where σ is restricted to the same set. For the medical knee images, we follow the work in [169] to train our 7-layer DnCNN* on NYU fastMRI dataset [56] for $\sigma \in \{1, 5, 10\}$. For color images, we use DnCNN* trained on 4744 images from the Waterloo Exploration Database [121] and use the CBM3D denoiser [43]. For all experiments in SR and Fourier, we use PnP-PGM in Algorithm 3 as the reconstruction algorithm with total 500 iterations, where as a hyperparameter, the scaling parameter μ is fixed for each iteration. All the quantitative results in the tables are averaged over seven test images shown in Figure 4.1 with hyperparameters optimized individually for each image for the best SNR performance using grid search. All visual results are shown with SNR value displayed directly on the images. None of the test images were used in training.

Table 4.1 shows the ability of denoiser scaling to adjust the denoising strength for three different image denoisers at four input SNR levels: 15 dB, 20 dB, 25 dB and 30 dB. In the table, TV (Optimized) is obtained by tuning the regularization parameter λ for each test

Table 4.2: Average SNRs obtained for different inverse problems and image denoisers.

A	Input SNR	TV		BM3D			DnCNN*	
		Scaled	Optimized	Unscaled	Scaled	Optimized	Unscaled	Scaled
Fourier	**30 dB**	27.15	27.15	27.29	28.49	28.51	27.92	28.44
	40 dB	27.79	27.79	28.97	29.17	29.21	29.72	29.89
SR	**30 dB**	19.86	19.87	14.32	20.65	20.64	13.59	20.05
	40 dB	22.42	22.41	22.64	22.78	22.71	23.03	23.07

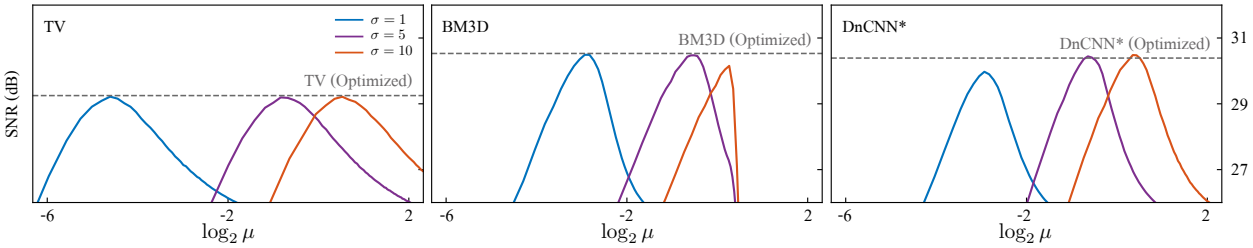

Figure 4.3: The influence of the scaling parameter μ on the denoising performance for *Pepper* for AWGN with input SNR of 25 dB ($\sigma = 7.23$). We show the SNR evolution against μ for the variants of TV, BM3D, and DnCNN* designed for the mismatched noise levels. The horizontal line shows the performance of the corresponding denoiser optimized for input SNR of 25 dB. Note how by adjusting μ, one can achieve nearly optimal performance for all three denoisers.

image. For TV (Scaled), we fix $\lambda = 1$ and tune the scaling parameter μ for the best result. The performances of BM3D (Unscaled and Scaled) and DnCNN* (Unscaled and Scaled) correspond to the best instance selected from the limited set of $\sigma \in \{1, 5, 10\}$. For reference, we also show BM3D (Optimized), which uses fully optimized σ, and DnCNN (Optimized), which is trained on noisy images with the true input SNR. Table 4.1 highlights the equivalence between TV with an optimized λ and TV with $\lambda = 1$, but optimized μ, which validates eq. (4.4). Table 4.1 also shows that denoiser scaling significantly improves the performance of sub-optimally tuned BM3D and DnCNN* to achieve the performance of the corresponding denoiser with optimized σ.

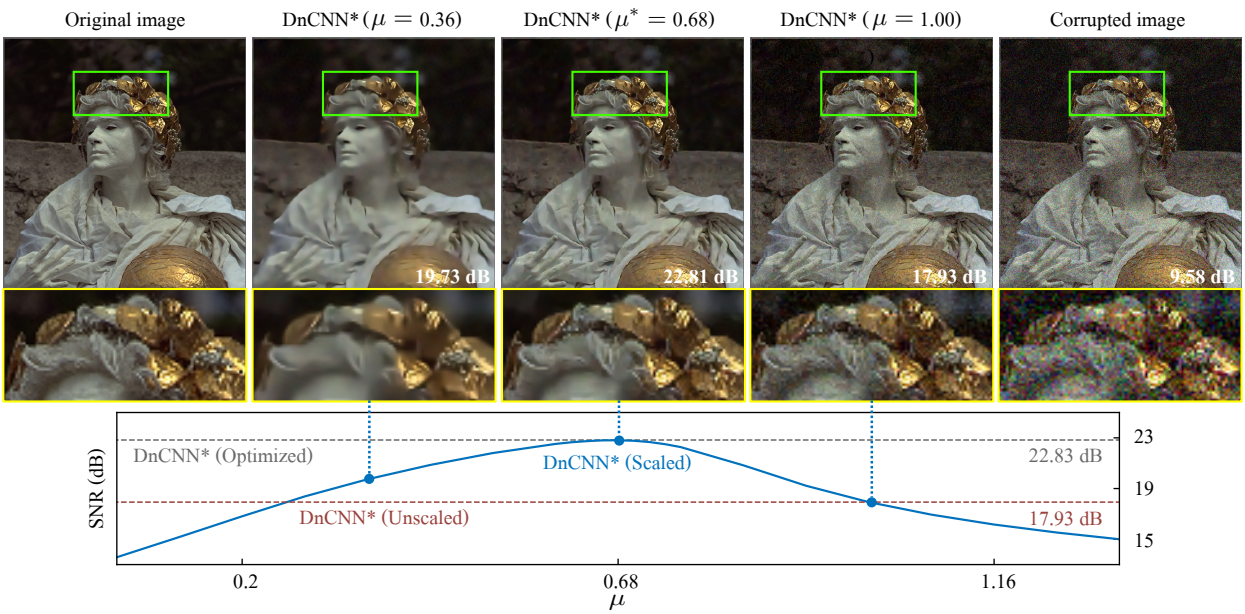

Figure 4.4: The influence of the denoiser scaling parameter μ on the denoising performance of DnCNN* on the color image *Statue*. The noise in the image corresponds to $\sigma = 30$, while DnCNN* was trained for the removal of noise corresponding to $\sigma = 20$. The top row images illustrate the visual performance at μ values of 0.36, 0.68, and 1.00. The bottom plot shows the SNR evolution against the parameter μ for a wider range of values. The scaled DnCNN* achieves its best performance at $\mu^* = 0.68$. Note how unscaled DnCNN* ($\mu = 1.00$) leads to an insufficient amount of regularization, while a smaller scaling parameter $\mu = 0.36$ leads to oversmoothing. This figure highlights the ability of μ to control the strength of regularization with a CNN denoiser.

Figure 4.2 visually illustrates the performance of denoiser scaling on the problem of color image denoising for AWGN of $\sigma = 30$. In the figure, DnCNN* (Optimized) denotes to the denoiser trained using the dataset with the correct noise level. On the other hand, DnCNN* (Unscaled) and DnCNN* (Scaled) correspond to the same CNN instance, trained for noise level $\sigma = 20$. DnCNN* (Unscaled) uses $\mu = 1$, while DnCNN* (Scaled) optimizes μ for the best SNR performance. By simply adjusting μ, the suboptimaly trained DnCNN* achieves the performance of DnCNN* trained using the correct noise level.

Figure 4.3 considers the problem of denoising an image with the input SNR of 25 dB, which corresponds to the noise level $\sigma = 7.23$. The figure shows the influence of the parameter

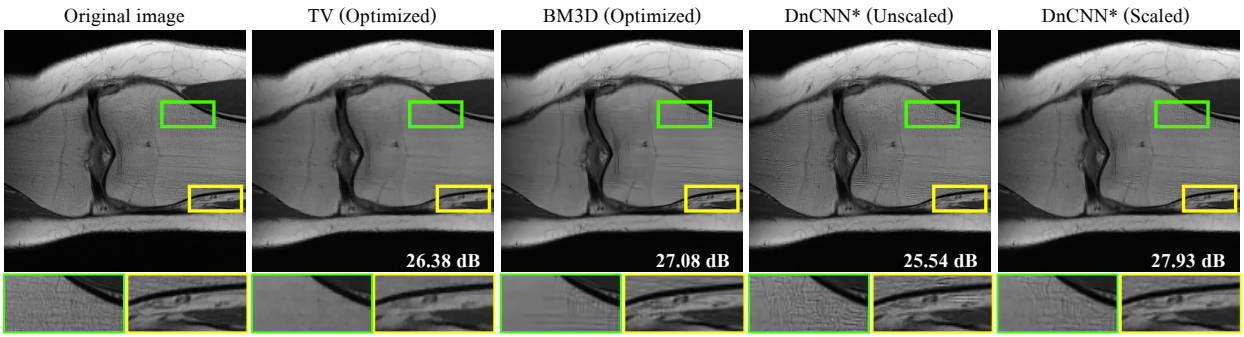

Figure 4.5: Visual illustration of denoiser scaling on the subsampled Fourier operator and one medical image *Knee* from the fastMRI dataset. The sampling ration is $m/n = 1/3$ and input SNR is 30 dB. DnCNN* is selected from $\sigma \in \{1, 5, 10\}$ that produces the best SNR performance. DnCNN* (Scaled) relies on the same CNN selected by DnCNN* (Unscaled). Note how DnCNN* (Scaled) improves the visual quality of results compared to DnCNN* (Unscaled).

μ for improving the performance of denoisers at mismatched values of $\sigma = 1$, 5, and 10. The SNR value after denoising is plotted against the logarithm of μ. For TV, we fixed the regularization parameter λ to its optimal value for the noise levels $\sigma \in \{1, 5, 10\}$ and then adjusted the parameter μ. For each plot, we also provide the performance of the denoisers with optimized σ. The comparison between unscaled denoisers at $\mu = 1$ and their scaled counterparts demonstrate the potential of denoiser scaling to influence the final performance, validating the theoretical conclusion that the denoiser scaling directly controls the strength of regularization. This is particularly appealing for DnCNN*, which does not have a tunable parameter σ.

Figure 4.4 visually and quantitatively illustrates the influence of the parameter μ for color image denoising. The DnCNN* trained on noise level $\sigma = 20$ is applied for denoising an image with a noise level $\sigma = 30$. The performance of scaled DnCNN achieves its peak SNR at $\mu^* = 0.68$ and leads to either under- or over-regularization at either side of this value.

Table 4.2 shows results of image reconstruction for both Fourier and SR under the noise levels 30 dB and 40 dB. Here, we fix $\sigma = 1$ for both BM3D and DnCNN*. We also show

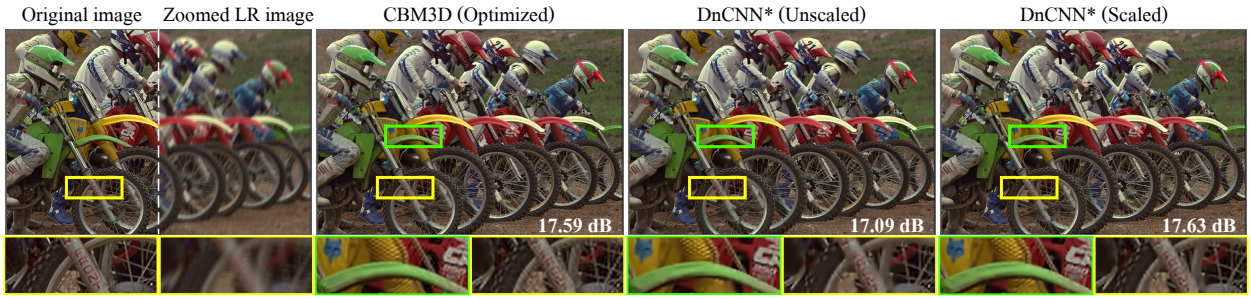

Figure 4.6: Results of SR simulation on color image *Bikes*. The input SNR is 40 dB. DnCNN* is selected from $\sigma \in \{1, 5, 10\}$ that produces the best SNR performance. DnCNN* (Scaled) relies on the same CNN selected by DnCNN* (Unscaled). One can see while DnCNN* blur out the details in some regions, DnCNN* (Scaled) can generate images with more details and sharper edges.

BM3D (Optimized), which uses fully optimized σ. For Fourier, the measurement ratio is set to be approximately $m/n = 1/3$. For SR, the low resolution (LR) image is simulated by convolving the high resolution image (HR) with a Gaussian motion-blur kernel of size 19×19 from [112], followed by down-sampling with scale factor 2. Note that TV (Scaled) has a fixed $\lambda = 1$ and a scaling parameter μ optimized for each test image. Some visual results are shown in Figures 4.5, and 4.6. These results clearly highlight the potential of denoiser scaling to significantly boost the performance of PnP.

4.5 Summary

In this chapter, we have presented a simple, but effective, technique for improving the performance of PnP algorithms. The approach is justified theoretically by connecting the denoiser scaling with the strength of the effective regularization introduced by PnP. The proposed technique is shown to be particularly valuable when PnP is used with CNN denoisers that have no explicit tunable parameters. Our experimental results show the potential of denoiser scaling to significantly improve the performance of PnP algorithms across several inverse problems. While we have focused on PnP, the denoiser scaling approach can be

applied more broadly to improve the performance of related methods, such as AMP [64, 131, 174] and RED [125, 149, 153, 169].

Chapter 5

Provable Convergence of Plug-and-Play Priors with MMSE Denoisers

While PnP algorithms are well understood for denoisers performing MAP estimation, they have not been analyzed for the MMSE denoisers. This chapter addresses this gap by establishing the first theoretical convergence result for proximal gradient method (PGM) variant of PnP for MMSE denoisers. We show that the iterates produced by PnP-PGM with an MMSE denoiser converge to a stationary point of some global cost function. We validate our analysis on sparse signal recovery in compressive sensing by comparing two types of denoisers, namely the *exact* MMSE denoiser and the *approximate* MMSE denoiser obtained by training a DNN.[4]

[4]This chapter is based on our paper [199].

5.1 Introduction

Consider the following regularized optimization problem for solving the linear inverse problem (2.3)

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, f(\boldsymbol{x}) \quad \text{with} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + r(\boldsymbol{x}), \tag{5.1}$$

where g is the data-fidelity term, r is the regularizer. Recall the known iterative proximal algorithm PGM (see Algorithm 1) developed for solving this problem when regularizer is nonsmooth

$$\boldsymbol{z}^k = \boldsymbol{x}^{k-1} - \gamma\nabla g(\boldsymbol{x}^{k-1}) \tag{5.2a}$$

$$\boldsymbol{x}^k = \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k), \tag{5.2b}$$

and its PnP variant PnP-PGM (see Algorithm 3) equipped with a denoiser D_σ

$$\boldsymbol{z}^k = \boldsymbol{x}^{k-1} - \gamma\nabla g(\boldsymbol{x}^{k-1}) \tag{5.3a}$$

$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k), \tag{5.3b}$$

where $\gamma > 0$ is the step-size parameter and $\sigma > 0$ in analogy to γ for controlling the relative strength of the denoiser D_σ. As we have discussed in Section 2.3, the transition from PGM to PnP-PGM is inspired by the fact that $\mathsf{prox}_{\gamma r}$ can be interpreted as a MAP estimator for the AWGN denoising problem

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{n} \quad \text{where} \quad \boldsymbol{x} \sim p_{\boldsymbol{x}}, \quad \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \gamma\mathbf{I}) \tag{5.4}$$

by setting $r(\boldsymbol{x}) = -\mathsf{log}(p_{\boldsymbol{x}}(\boldsymbol{x}))$.

Recent work has provided theoretical convergence guarantees for PnP algorithms under various assumptions on the data-fidelity term and the denoiser [31, 36, 67, 158, 166, 168, 176]. However, PnP has *not* been investigated for denoisers performing *minimum mean squared error (MMSE)* estimation on (5.4)

$$\mathsf{D}_\sigma(\boldsymbol{z}) = \mathbb{E}[\boldsymbol{x}|\boldsymbol{z}] = \int_{\mathbb{R}^n} \boldsymbol{x} p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x}|\boldsymbol{z}) \mathsf{d}\boldsymbol{x}. \tag{5.5}$$

MMSE denoisers are "optimal" with respect to broadly used image-quality metrics, such as signal-to-noise ratio (SNR). However, they are generally *not* nonexpansive [177] and their direct computation is often intractable in high-dimensions [96]. Insights into the performance of PnP for MMSE denoisers are valuable as many denoisers (pre-trained CNNs, NLM, BM3D) can be interpreted as *approximate* or *empirical* MMSE denoisers. In this chapter, we show that PnP-PGM with an MMSE denoiser converges to a stationary point of a certain (possibly nonconvex) cost function. To the best of our knowledge, this explicit link between PnP-PGM and MMSE estimation is missing in the current literature on PnP. Our analysis builds on an elegant formulation by Gribonval [70], establishing a direct link between MMSE estimation and regularized optimization. We validate our analysis on sparse signal recovery in compressive sensing by comparing PnP-PGM with two types of denoisers—the *exact* MMSE denoiser and the *approximate* MMSE denoiser obtained by training DnCNN [216] to minimize MSE. Our simulations show convergence of PnP-PGM for both denoisers, highlight their close agreement in terms of performance, and illustrate the limitation of using an AWGN denoiser as a prior within PGM.

5.2 Theoretical Analysis

Our analysis requires three assumptions that serve as sufficient conditions for establishing theoretical convergence.

Assumption 5.1. The prior $p_{\boldsymbol{x}}$ is non-degenerate over $\mathbb{R}^n$.

As a reminder, a probability distribution $p_{\boldsymbol{x}}$ is *degenerate* over $\mathbb{R}^n$, if it is supported on a space of lower dimensions than n. Consider the image set of the MMSE denoiser $\mathcal{X} \coloneqq \mathsf{Im}(\mathsf{D}_\sigma)$. Assumption 5.1 is required for establishing an explicit link between (5.5) and the following function [70]

$$r(\boldsymbol{x}) \coloneqq \begin{cases} -\frac{1}{2\gamma}\|\boldsymbol{x} - \mathsf{D}_\sigma^{-1}(\boldsymbol{x})\|^2 + \frac{\sigma^2}{\gamma} r_\sigma(\mathsf{D}_\sigma^{-1}(\boldsymbol{x})) & \text{for } \boldsymbol{x} \in \mathcal{X} \\ +\infty & \text{for } \boldsymbol{x} \notin \mathcal{X}, \end{cases} \tag{5.6}$$

where $\gamma > 0$ is the step-size and $\mathsf{D}_\sigma^{-1} : \mathcal{X} \to \mathbb{R}^n$ is the inverse mapping, which is well defined and smooth over $\mathcal{X}$ (see Appendix D.1). The definition of r includes the function $r_\sigma(\cdot) \coloneqq -\mathsf{log}(p_{\boldsymbol{z}}(\cdot))$, where $p_{\boldsymbol{z}}$ is the probability distribution of the AWGN corrupted observation (5.4). As discussed in Appendix D.1, the function r is smooth for any $\boldsymbol{x} \in \mathcal{X}$, which is the consequence of the smoothness of both D_σ^{-1} and r_σ.

Assumption 5.2. The function g is continuously differentiable and has a Lipschitz continuous gradient with constant $L > 0$.

This is a standard assumption used extensively in the analysis of gradient-based algorithms (see for example [136]).

Assumption 5.3. The function f has a finite infimum $f^* > -\infty$.

This mild assumption ensures that the function f is bounded from below. We can now establish the following result.

Theorem 5.1. *Run PnP-PGM with a denoiser* (5.5) *under Assumptions 5.1-5.3 using a fixed step-size* $0 < \gamma \leq 1/L$. *Then, the sequence* $\{f(\boldsymbol{x}^k)\}_{k\geq 0}$ *with* r *defined in* (5.6) *monotonically decreases and* $\|\nabla f(\boldsymbol{x}^k)\| \to 0$ *as* $k \to \infty$.

The proof is provided in Appendix D.2. Theorem 5.1 establishes convergence of PnP-PGM with MMSE denoisers to a stationary point of the problem (5.1) where r is specified in (5.6). The proof relies on the *majorization-minimization (MM) strategy* widely used in the context of both convex and nonconvex optimization [16, 47, 105, 122, 148]. It is important to note that the theorem does *not* assume that g or r are convex and that the denoiser is nonexpansive. The convexity of r is equivalent to the log-concavity of $p_{\boldsymbol{y}}$ [71], which is not true for a wide variety of priors, such as mixtures of Gaussians [177]. In fact, D_σ is a proximal operator of a proper, closed, and convex function r if and only if D_σ is monotone and nonexpansive [42]. Finally, note that the definition of r in (5.6) depends on both γ and σ, both of which influence the relative weighting between g and r. This is the consequence of r being specified by *reverse engineering* the MMSE denoiser D_σ, which leads to the explicit dependence of r on the problem parameters.

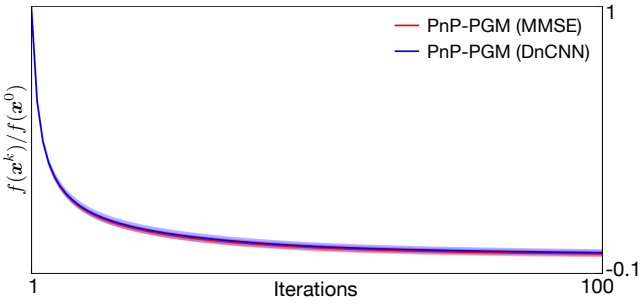

Figure 5.1: Convergence of PnP-PGM for exact and approximate MMSE denoisers. The latter corresponds to DnCNN trained to minimize MSE. Average normalized cost $f(\boldsymbol{x}^k)/f(\boldsymbol{x}^0)$ is plotted against the iteration number with the shaded areas representing the range of values attained over 100 experiments. Note the monotonic decrease of the cost function f as predicted by our analysis as well as the excellent agreement of both denoisers.

5.3 Numerical Evaluation

We illustrate PnP-PGM with both *exact* and *approximate* MMSE denoisers on the problem of sparse signal recovery in compressive sensing [33, 50]. It is important to point out that our aim here is *not* to justify PGM as a superior sparse recovery algorithm or the MMSE denoiser as a superior signal prior. Instead, we seek to gain new insights into the behavior of PnP-PGM with MMSE priors in highly controlled setting.

We generate $\boldsymbol{x} \in \mathbb{R}^n$ with $n = 4096$ as a sparse independent and identically distributed (i.i.d.) Bernoulli-Gaussian vector. Each component of $\boldsymbol{x}$ is thus generated from the distribution $p_x(x) = \alpha\phi_{\sigma_x}(x) + (1-\alpha)\delta(x)$, where δ is the Dirac delta function and ϕ_{σ_x} is the Gaussian probability density function with zero mean and $\sigma_x > 0$ standard deviation. The parameter $0 \leq \alpha \leq 1$ in p_x controls the sparsity of the signal and we fix $\sigma_x^2 = 1/\alpha$. Since the distribution $p_z = (\phi_\sigma * p_x)$ is not log-concave [70], the Bernoulli-Gaussian prior leads to a nonconvex regularizer and an expansive denoiser. The entries of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ are generated as i.i.d. Gaussian random variables $\mathcal{N}(0, 1/m)$. For each experiment, we additionally corrupt measurements with AWGN of variance σ_e^2 corresponding to input SNR of 20 dB. Accordingly,

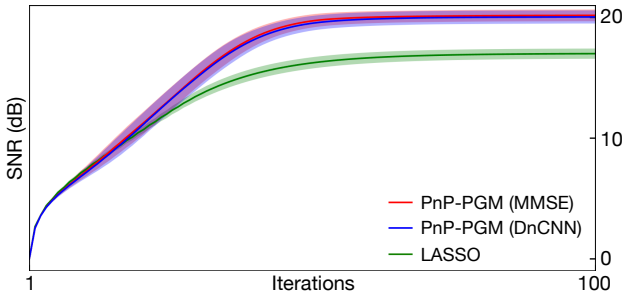

Figure 5.2: Convergence of PnP-PGM for exact and approximate MMSE denoisers. The latter corresponds to DnCNN trained to minimize MSE. Average SNR (dB) is plotted against the iteration number with the shaded areas representing the range of values attained over 100 experiments. The SNR behavior of LASSO, implemented using PGM with the ℓ_1-norm prior, is also provided for reference. We highlight excellent agreement of both denoisers and their superior SNR performance compared to the ℓ_1 regularization.

the data fidelity term is set as least-squares $g(\boldsymbol{x}) = (1/2)\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$. All plots are obtained by averaging results over 100 random trials.

We consider two baseline signal recovery algorithms extensively used in compressive sensing. The first is the standard *least absolute shrinkage and selection operator (LASSO)* [180], which computes (5.1) with an ℓ_1-norm regularizer $r(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$. The regularization parameter $\lambda > 0$ of LASSO is optimized for each experiment to maximize SNR. The second baseline method is the MMSE variant of the *generalized approximate message passing (GAMP)* [147], which is known to be nearly optimal for sparse signal recovery in compressive sensing [103]. The parameters of GAMP are set to the actual statistical parameters $(\alpha, \sigma_x, \sigma_e)$ of the problem. While the suboptimality of PGM to GAMP for random measurement matrices is well known, our aim is to illustrate the relative performance of "optimal" PGM with the MMSE denoiser D_σ.

Since $\boldsymbol{x}$ is a vector with i.i.d. elements, the exact MMSE denoiser D_σ can be evaluated as a sequence of scalar integrals. As an approximate MMSE denoiser, we use DnCNN with depth 4 (see [216] for more details). To that end, we train 9 different networks for the removal of

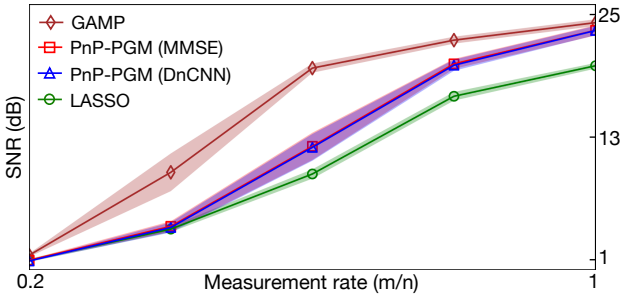

Figure 5.3: Illustration of the recovery performance of PnP-PGM for exact and approximate MMSE denoisers. Average SNR (dB) is plotted against the measurement rate (m/n) with the shaded areas representing the range of values attained over 100 experiments. We also provide the performance of LASSO and GAMP, two widely used algorithms for sparse recovery in compressive sensing. The figure highlights the suboptimality of both variants of PnP-PGM compared to GAMP, which stems from their assumption that errors in every PGM iteration are AWGN. One can also observe the remarkable agreement between two variants of PnP-PGM in all experiments.

AWGN at noise levels in the range from 0.01 to 0.37. The training was conducted over 2000 random realizations of the signal $\boldsymbol{x} \sim p_{\boldsymbol{x}}$ using the ℓ_2-loss. For each experiment, we select the network achieving the highest SNR value under the scaling technique from [199].

Theorem 5.1 establishes monotonic convergence of PnP-PGM in terms of the cost function f. This is illustrated in Fig. 5.1 for the measurement rate $m/n = 0.8$. The average normalized cost $f(\boldsymbol{x}^k)/f(\boldsymbol{x}^0)$ is plotted against the iteration number for both exact and approximate MMSE denoisers. The shaded areas indicate the range of values taken over 100 random trials. Fig. 5.2 illustrates the convergence behaviour of PnP-PGM in terms of SNR (dB) for identical experimental setting by additionally including the SNR performance of LASSO as a reference. First, note the monotonic convergence of $\{f(\boldsymbol{x}^k)\}_{k\geq 0}$ as predicted by our analysis. Second, note the excellent agreement between two variants of PnP-PGM. This close agreement is encouraging as deep neural nets have been extensively used as practical strategies for regularizing large-scale imaging problems.

The underlying assumption in PnP-PGM is that errors within every PGM iteration can be modeled as AWGN, which is known to be false [51]. This makes both exact and approximate MMSE denoisers "suboptimal" when used within PnP-PGM. Unlike PGM, GAMP explicitly ensures AWGN errors in every iteration *for random measurement matrices*, making it a valid upper bound in our experimental setting. Fig. 5.2 illustrates the suboptimality of "optimal" PGM for different measurement rates, highlighting the necessity of developing more accurate error models for PnP iterations [55]. Note again the remarkable agreement between DnCNN and the exact MMSE estimator, which highlights practical relevance of our analysis.

5.4 Summary

This chapter provides several new insights into the widely used PnP methodology by considering "optimal" MMSE denoisers. First, we have theoretically analyzed the convergence of PnP-PGM for MMSE denoisers. Our analysis reveals that the algorithm converges even when the data-fidelity term is *nonconvex* and denoiser is *not* nonexpansive. This has not been shown in the prior work on PnP. Second, our simulations on sparse signal recovery illustrate the potential of *approximate* MMSE denoisers—obtained by training deep neural nets—to match the performance of the *exact* MMSE denoiser. The latter is intractable for high-dimensional imaging problems, while the former has been extensively used in practice. Third, our simulations highlight the *suboptimality* of "optimal" PGM with an MMSE denoiser, due to the assumption that error within PGM iterations are Gaussian. We hypothesize that a similar phenomenon is present in the context of imaging inverse problems, which indicates to possible performance improvements by using more refined statistical models for characterizing errors within PnP algorithms.

Part III

Adapting Plug-and-Play Priors for Large-scale Problems

Chapter 6

Incremental Plug-and-Play Alternating Direction Method of Multipliers

Although PnP is broadly applicable for solving inverse problems, current PnP algorithms are impractical in large-scale settings due to their heavy computational and memory requirements. In this chapter, we address this issue by proposing an *incremental* variant of the widely used PnP-ADMM algorithm, making it scalable to problems involving a large number measurements. We theoretically analyze the convergence of the algorithm under a set of explicit assumptions, extending recent theoretical results in the area. Additionally, we show the effectiveness of our algorithm with nonsmooth data-fidelity terms and deep neural net priors, its fast convergence compared to existing PnP algorithms, and its scalability in terms of speed and memory.[5]

[5]This chapter is based on our paper [172].

6.1 Introduction

The empirical success of PnP has spurred a follow-up work that provided theoretical justifications to PnP in various settings [31, 35, 128, 158, 168, 177, 181, 199, 200], including the ones we presented in Chapter 4 and Chapter 5. Despite this progress, the computation and memory requirements of current PnP algorithms makes them impractical in problems with a large number of measurements. One prior work on developing PnP algorithms for processing large-scale measurements is the *stochastic gradient descent variant of PnP (PnP-SGD)*, whose fixed-point convergence was recently analyzed for smooth data-fidelity terms [168].

In this chapter, we present a new *incremental PnP-ADMM (IPA)* algorithm for dealing with large-scale measurements. As an extensions of the widely used PnP-ADMM [166, 186], IPA can integrate statistical information from a data-fidelity term and a pre-trained deep neural net. However, unlike PnP-ADMM, IPA can effectively scale to datasets that are too large for traditional batch processing by using a single element or a small subset of the dataset at a time. The memory and per-iteration complexity of IPA is independent of the number of measurements, thus allowing it to deal with very large datasets. Additionally, unlike PnP-SGD [168], IPA can effectively address problems with *nonsmooth* data-fidelity terms, and generally has faster convergence. We present a detailed convergence analysis of IPA under a set of explicit assumptions on the data-fidelity term and the denoiser. Our analysis extends the recent fixed-point analysis of PnP-ADMM in [158] to partial randomized processing of data. To the best of our knowledge, the proposed scalable PnP algorithm and corresponding convergence analysis are absent from the current literature in this area. Our numerical validation demonstrates the practical effectiveness of IPA for integrating nonsmooth data-fidelity terms and deep neural net priors, its fast convergence compared to PnP-SGD, and its scalability in terms of both speed and memory. In summary, we establish

IPA as a flexible, scalable, and theoretically sound PnP algorithm applicable to a wide variety of large-scale problems.

6.2 Background

To explain the details of our IPA algorithm, let's recall the original PnP-ADMM algorithm summarized in Algorithm 4

$$\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1}) \tag{6.1a}$$

$$\boldsymbol{x}^k = \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k - \boldsymbol{s}^{k-1}) \tag{6.1b}$$

$$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k \ . \tag{6.1c}$$

A elegant fixed-point convergence analysis of PnP-ADMM was presented in [158]. By substituting $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ into PnP-ADMM, the algorithm is expressed in terms of an operator

$$\mathsf{P} \coloneqq \frac{1}{2}\mathsf{I} + \frac{1}{2}(2\mathsf{G} - \mathsf{I})(2\mathsf{D}_\sigma - \mathsf{I}) \quad \text{with} \quad \mathsf{G} \coloneqq \mathsf{prox}_{\gamma g} \ , \tag{6.2}$$

where I denotes the identity operator. The convergence of PnP-ADMM is then established through its equivalence to the fixed-point convergence of the sequence $\boldsymbol{v}^k = \mathsf{P}(\boldsymbol{v}^{k-1})$. The equivalence of PnP-ADMM to the iterations of the operator (6.2) originates from the well-known relationship between ADMM and the Douglas-Rachford splitting [31, 53, 144, 158].

Since PnP-ADMM can integrate powerful deep neural net denoisers, there is a need to understand its theoretical properties and ability to process a large number of measurements. In this chapter, we address this issue by providing new conceptual, theoretical, and empirical insights into incremental ADMM optimization under statistical priors specified as deep neural net denoisers. Scalable optimization algorithms have become increasingly important in the

context of large-scale problems arising in machine learning and data science [26]. Stochastic and online optimization techniques have been investigated for traditional ADMM [81, 142, 173, 188, 220], where $\mathsf{prox}_{\gamma g}$ is approximated using a subset of observations (with or without subsequent linearization). Our work contributes to this area by investigating the scalability of PnP-ADMM that is *not* minimizing any explicit objective function.

6.3 Incremental PnP-ADMM

Batch PnP algorithms operate on the whole observation vector $\boldsymbol{y} \in \mathbb{R}^m$. We are interested in partial randomized processing of observations by considering the decomposition of $\mathbb{R}^m$ into $b \geq 1$ blocks

$$\mathbb{R}^m = \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_b} \quad \text{with} \quad m = m_1 + m_2 + \cdots + m_b \ .$$

We thus consider data-fidelity terms of the form

$$g(\boldsymbol{x}) = \frac{1}{b} \sum_{i=1}^{b} g_i(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n \ , \tag{6.3}$$

where each g_i is evaluated only on the subset $\boldsymbol{y}_i \in \mathbb{R}^{m_i}$ of the full data $\boldsymbol{y}$.

The proposed IPA algorithm seeks to avoid the direct computation of $\mathsf{prox}_{\gamma g}$ in PnP-ADMM. As shown in Algorithm 7, it extends stochastic variants of traditional ADMM [81, 142, 173, 188, 220] by integrating denoisers D_σ that are *not* associated with any h. Its per-iteration complexity is independent of the number of data blocks b, since it processes only a single component function g_i at every iteration.

It is important to note that in some applications [7, 146, 193], the $\mathsf{prox}_{\gamma g}$ step of PnP-ADMM can be efficiently evaluated by leveraging the structure of the measurement operator (such as

Algorithm 7 Incremental Plug-and-Play ADMM (IPA)

input: initial values $\boldsymbol{x}^0, \boldsymbol{s}^0 \in \mathbb{R}^n$, parameters $\gamma, \sigma > 0$.
for $k = 1, 2, 3, \ldots$ **do**
  Choose an index $i_k \in \{1, \ldots, b\}$
  $\boldsymbol{z}^k \leftarrow \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1})$ where $\mathsf{G}_{i_k} \coloneqq \mathsf{prox}_{\gamma g_{i_k}}$
  $\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k - \boldsymbol{s}^{k-1})$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k$
end for

diagonalization by Fourier transform). Nonetheless, IPA provides flexibility for controlling the number of measurements $1 \leq m_i \leq m$ used in every iteration, which makes it a useful alternative to PnP-ADMM, when the memory/computational benefits for evaluating $\mathsf{prox}_{\gamma g_i}$ (which uses only $\boldsymbol{y}_i \in \mathbb{R}^{m_i}$ and $\boldsymbol{A}_i \in \mathbb{R}^{m_i \times n}$) outweigh those of $\mathsf{prox}_{\gamma g}$ (which uses $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$).

In principle, IPA can be implemented using different block selection rules. The strategy adopted for our theoretical analysis focuses on the usual strategy of selecting indices i_k as independent and identically distributed (i.i.d.) random variables distributed uniformly over $\{1, \ldots, b\}$. An alternative would be to proceed in epochs of b consecutive iterations, where at the start of each epoch the set $\{1, \ldots, b\}$ is reshuffled, and i_k is selected from this ordered set [20]. In some applications, it might also be beneficial to select indices i_k in an online data-adaptive fashion by taking into account the statistical relationships among observations [97, 179].

Unlike PnP-SGD, IPA does not require smoothness of the functions g_i. Instead of computing the partial gradient ∇g_i, as is done in PnP-SGD, IPA evaluates the partial proximal operator G_i. Nonsmooth data-fidelity terms have been extensively used in many applications, including wavelet inpainting, tensor factorization, feature selection, dictionary learning, and phase unwrapping [8, 37, 76, 86, 92, 117, 138]. The maximal benefit of IPA over PnP-SGD is expected for problems in which G_i is efficient to evaluate. This is a case for a number of

functions commonly used in many applications (see the extensive discussion on proximal operators in [15]). For example, the proximal operator of the ℓ_2-norm data-fidelity term $g_i(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{A}_i\boldsymbol{x}\|_2^2$ has a closed-form solution

$$\mathsf{G}_i(\boldsymbol{z}) = \mathsf{prox}_{\gamma g_i}(\boldsymbol{z}) = \left(\mathbf{I} + \gamma \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i\right)^{-1}\left(\boldsymbol{z} + \gamma \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{y}_i\right) \tag{6.4}$$

for $\gamma > 0$ and $\boldsymbol{z} \in \mathbb{R}^n$. Prior work has extensively discussed efficient strategies for evaluating (6.4) for a variety of linear operators, including convolutions, partial Fourier transforms, and subsampling masks [2, 7, 146, 193]. As a second example, consider the ℓ_1-data fidelity term $g_i(\boldsymbol{x}) = \|\boldsymbol{y}_i - \boldsymbol{A}_i\boldsymbol{x}\|_1$, which is nonsmooth. The corresponding proximal operator has a closed form solution for any orthogonal operator $\boldsymbol{A}_i$ and can also be efficiently computed in many other settings [15].

IPA can also be implemented as a *minibatch* algorithm, processing several blocks in parallel at every iteration, thus improving its efficiency on multi-processor hardware architectures. Algorithm 8 presents the minibatch version of IPA that averages several proximal operators evaluated over different data blocks. When the minibatch size $p = 1$, Algorithm 8 reverts to Algorithm 7. The main benefit of minibatch IPA is its suitability for parallel computation of $\widehat{\mathsf{G}}$, which can take advantage of multi-processor architectures.

Minibatch IPA is related to the *proximal average* approximation of $\mathsf{G} = \mathsf{prox}_{\gamma g}$ [13, 211]

$$\overline{\mathsf{G}}(\boldsymbol{x}) = \frac{1}{b}\sum_{i=1}^{b} \mathsf{prox}_{\gamma g_i}(\boldsymbol{x}) \quad \boldsymbol{x} \in \mathbb{R}^n \, .$$

When Assumption 6.1, introduced in Section 6.4, is satisfied, then the approximation error is bounded for any $\boldsymbol{x} \in \mathbb{R}^n$ as

$$\|\mathsf{G}(\boldsymbol{x}) - \overline{\mathsf{G}}(\boldsymbol{x})\| \leq 2\gamma L \, .$$

Algorithm 8 Minibatch IPA

input: initial values $\boldsymbol{x}^0, \boldsymbol{s}^0 \in \mathbb{R}^n$, parameters $\gamma, \sigma > 0$, minibatch size $p \geq 1$.
for $k = 1, 2, 3, \ldots$ **do**
    Choose indices $i_1, \ldots, i_p$ from the set $\{1, \ldots, b\}$.
    $\boldsymbol{z}^k \leftarrow \widehat{\mathsf{G}}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1})$ where $\widehat{\mathsf{G}} \coloneqq \frac{1}{p}\sum_{j=1}^{p} \mathsf{prox}_{\gamma g_{i_j}}$
    $\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k - \boldsymbol{s}^{k-1})$
    $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k$
end for

Minibatch IPA thus simply uses a minibatch approximation $\widehat{\mathsf{G}}$ of the proximal average $\overline{\mathsf{G}}$. One implication of this is that even when the minibatch is *exactly* equal to the full measurement vector, minibatch IPA is not exact due to the approximation error introduced by the proximal average. However, the resulting approximation error can be made as small as desired by controlling the penalty parameter $\gamma > 0$.

6.4 Theoretical Analysis

We now present a theoretical analysis of IPA. We first present an intuitive interpretation of its solutions, and then present our convergence analysis under a set of explicit assumptions.

6.4.1 Fixed Point Interpretation

PnP cannot be interpreted using the standard tools from convex optimization, since its solution is generally not a minimizer of an objective function. Nonetheless, we develop an intuitive operator based interpretation.

Consider the following set-valued operator

$$\mathsf{T} \coloneqq \gamma \partial g + (\mathsf{D}_\sigma^{-1} - \mathsf{I}) \quad \gamma > 0 \,, \tag{6.5}$$

where ∂g is the subdifferential of the data-fidelity term and $\mathsf{D}_\sigma^{-1}(\boldsymbol{x}) \coloneqq \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}_\sigma(\boldsymbol{z})\}$ is the inverse operator of the denoiser D_σ. The details for obtaining (6.5) from (6.1) are provided in Appendix E.3.1. Note that this inverse operator exists even when D_σ is not one-to-one [53, 157]. By characterizing the fixed points of PnP algorithms, it can be shown that their solutions can be interpreted as vectors in the zero set of T

$$\begin{aligned} &\mathbf{0} \in \mathsf{T}(\boldsymbol{x}^*) = \gamma \partial g(\boldsymbol{x}^*) + (\mathsf{D}_\sigma^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^*) \\ \Leftrightarrow \quad &\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{T}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \mathsf{T}(\boldsymbol{x})\} \, . \end{aligned}$$

Consider the following two sets

$$\begin{aligned} \mathsf{zer}(\partial g) &\coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \partial g(\boldsymbol{x})\} \quad \text{and} \\ \mathsf{fix}(\mathsf{D}_\sigma) &\coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}_\sigma(\boldsymbol{x})\} \, , \end{aligned}$$

where $\mathsf{zer}(\partial g)$ is the set of all critical points of the data-fidelity term and $\mathsf{fix}(\mathsf{D}_\sigma)$ is the set of all fixed points of the denoiser. Intuitively, the fixed points of D_σ correspond to all vectors that are *not* denoised, and therefore can be interpreted as vectors that are *noise-free* according to the denoiser.

If $\boldsymbol{x}^* \in \mathsf{zer}(\partial g) \cap \mathsf{fix}(\mathsf{D}_\sigma)$, then $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{T})$, which implies that $\boldsymbol{x}^*$ is one of the solutions. Hence, any vector that minimizes a convex data-fidelity term g and noiseless according to D_σ is in the solution set. On the other hand, when $\mathsf{zer}(\partial g) \cap \mathsf{fix}(\mathsf{D}_\sigma) = \varnothing$, then $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{T})$ corresponds to an equilibrium point between two sets.

This interpretation of PnP highlights one important aspect that is often overlooked in the literature, namely that, unlike in the traditional formulation (2.10), the regularization in PnP depends on both the denoiser parameter $\sigma > 0$ and the penalty parameter $\gamma > 0$, with

both influencing the solution. Hence, the best performance is obtained by jointly tuning both parameters for a given experimental setting. In the special case of $\mathsf{D}_\sigma = \mathsf{prox}_{\gamma r}$ with $\gamma = \sigma^2$, we have

$$\mathsf{fix}(\mathsf{D}_\sigma) = \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \partial r(\boldsymbol{x})\} \quad \text{and}$$
$$\mathsf{zer}(\mathsf{T}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \partial g(\boldsymbol{x}) + \partial r(\boldsymbol{x})\} \,,$$

which corresponds to the optimization formulation (2.10) whose solutions are independent of γ.

6.4.2 Convergence Analysis

Our analysis requires three assumptions that jointly serve as sufficient conditions.

Assumption 6.1. Each g_i is proper, closed, convex, and Lipschitz continuous with constant $L_i > 0$. We define the largest Lipschitz constant as $L = \mathsf{max}\{L_1, \ldots, L_b\}$.

This assumption is commonly adopted in nonsmooth optimization and is equivalent to existence of a global upper bound on subgradients [28, 142, 211]. It is satisfied by a large number of functions, such as the ℓ_1-norm. The ℓ_2-norm also satisfies Assumption 6.1 when it is evaluated over a bounded subset of $\mathbb{R}^n$. We next state our assumption on D_σ.

Assumption 6.2. The residual $\mathsf{R}_\sigma \coloneqq \mathsf{I} - \mathsf{D}_\sigma$ of the denoiser D_σ is firmly nonexpansive.

We review firm nonexpansiveness and other related concepts in the Appendix E.3. Firmly nonexpansive operators are a subset of *nonexpansive* operators (those that are Lipschitz continuous with constant one). A simple strategy to obtain a firmly nonexpansive operator is to create a (1/2)-averaged operator from a nonexpansive operator [144]. The residual R_σ is

firmly nonexpansive *if and only if* D_σ is firmly nonexpansive. It is worth noting that (a) any explicit or implicit proximal operator is firmly nonexpansive, and (b) any symmetric matrix with eigenvalues in $[0, 1]$ is firmly nonexpansive. This implies that many recently designed denoisers for PnP, such as those discussed in [67, 135, 166, 176, 177] automatically satisfy Assumption 6.2.

The rationale for stating Assumption 6.2 for R_σ is based on our interest in *residual* deep neural nets. The success of residual learning in the context of image restoration is well known [216]. Prior work has also shown that Lipschitz constrained residual networks yield excellent performance without sacrificing stable convergence [158, 169]. Additionally, there has recently been an explosion of techniques for training Lipschitz constrained and firmly nonexpansive deep neural nets [58, 133, 158, 178].

Assumption 6.3. The operator T in (6.5) is such that $\mathsf{zer}(\mathsf{T}) \neq \varnothing$. There also exists $R < \infty$ such that

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 \leq R \quad \text{for all} \quad \boldsymbol{x}^* \in \mathsf{zer}(\mathsf{T}) \; .$$

The first part of the assumptions simply ensures the existence of a solution. The existence of the bound R often holds in practice, as many denoisers have bounded range spaces. In particular, this is true for a number of image denoisers whose outputs live within the bounded subset $[0, 255]^n \subset \mathbb{R}^n$.

We will state our convergence results in terms of the operator $\mathsf{S} : \mathbb{R}^n \to \mathbb{R}^n$ defined as

$$\mathsf{S} \coloneqq \mathsf{D}_\sigma - \mathsf{G}(2\mathsf{D}_\sigma - \mathsf{I}) \; . \tag{6.6}$$

Both IPA and PnP-ADMM can be interpreted as algorithms for computing an element in $\mathsf{zer}(\mathsf{S})$, which is equivalent to finding an element of $\mathsf{zer}(\mathsf{T})$ (see details in Appendix E.3).

We are now ready to state our main result on IPA.

Theorem 6.1. *Run IPA for $t \geq 1$ iterations with random i.i.d. block selection under Assumptions 6.1-6.3 using a penalty parameter $\gamma > 0$. Then, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ satisfies*

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t}\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2\right] \leq \frac{(R+2\gamma L)^2}{t} + \mathsf{max}\{\gamma, \gamma^2\}C\ , \tag{6.7}$$

where $C \coloneqq 4LR + 12L^2$ is a positive constant.

In order to contextualize this result, we also review the convergence of the traditional PnP-ADMM.

Theorem 6.2. *Run PnP-ADMM for $t \geq 1$ iterations under Assumptions 6.1-6.3 using a penalty parameter $\gamma > 0$. Then, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ satisfies*

$$\frac{1}{t}\sum_{k=1}^{t}\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2 \leq \frac{(R+2\gamma L)^2}{t}\ . \tag{6.8}$$

Both proofs are provided in the Appendix E.1. The proof of Theorem 6.2 is a modification of the analysis in [158], obtained by relaxing the *strong convexity* assumption in [158] by Assumption 6.1 and replacing the assumption that R_σ is a *contraction* in [158] by Assumption 6.2. Theorem 6.2 establishes that the iterates of PnP-ADMM satisfy $\|\mathsf{S}(\boldsymbol{v}^t)\| \to 0$ as $t \to \infty$. Since S is firmly nonexpansive (see Appendix E.3.3) and D_σ is nonexpansive, the Krasnosel'skii-Mann theorem (see Section 5.2 in [12]) directly implies that $\boldsymbol{v}^t \to \mathsf{zer}(\mathsf{S})$ and $\boldsymbol{x}^t = \mathsf{D}_\sigma(\boldsymbol{v}^t) \to \mathsf{zer}(\mathsf{T})$.

Theorem 6.1 establishes that *in expectation*, IPA has a similar convergence behavior to PnP-ADMM up to an error term that depends on the penalty parameter γ. One can precisely control the accuracy of IPA by setting γ to a desired level. In practice, γ can be treated as

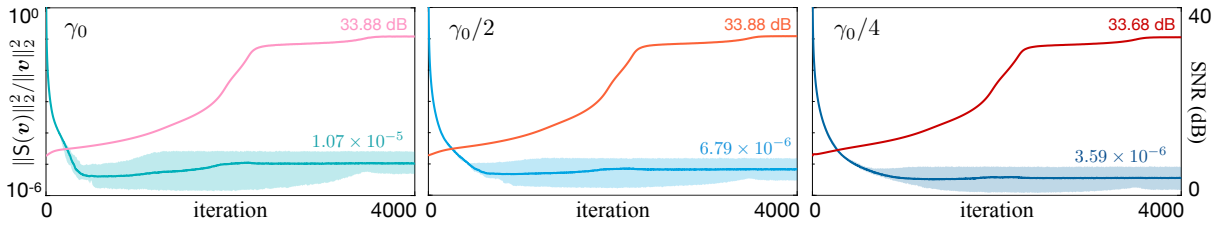

Figure 6.1: Illustration of the influence of the penalty parameter $\gamma > 0$ on the convergence of IPA for a DnCNN prior. The average normalized distance to $\mathsf{zer}(\mathsf{S})$ and SNR (dB) are plotted against the iteration number with the shaded areas representing the range of values attained over 12 test images. The accuracy of IPA improves for smaller values of γ. However, the SNR performance is nearly identical, indicating that in practice IPA can achieve excellent results for a range of fixed γ values.

a hyperparameter and tuned to maximize performance for a suitable image quality metric, such as SNR or SSIM. Our numerical results in Section 6.5 corroborate that excellent SNR performance of IPA can be achieved without taking $\|\mathsf{S}(\boldsymbol{v}^t)\|_2$ to zero, which simplifies practical applicability of IPA. (Note that the convergence analysis for IPA in Theorem 6.1 can be easily extended to minibatch IPA with a straightforward extension of Lemma E.1 in Appendix E.1.2 to several indices, and by following the steps of the main proof in Appendix E.1.1.)

Finally, note that the convergence of the IPA iterates can also be analyzed under assumptions adopted in [158], namely that g_i are strongly convex and R_σ is a contraction. Such an analysis leads to the statement

$$\mathbb{E}\left[\|\boldsymbol{x}^t - \boldsymbol{x}^*\|_2\right] \leq \eta^t(2R + 4\gamma L) + (4\gamma L)/(1-\eta)\,, \tag{6.9}$$

where $0 < \eta < 1$. Equation (6.9) establishes a linear convergence to $\mathsf{zer}(\mathsf{T})$ up to an error term. A proof of (6.9) is provided in the Appendix E.2. As corroborated by our simulations in Section 6.5, the actual convergence of IPA holds even more broadly than suggested by both sets of sufficient conditions. This suggests a possibility of future analysis of IPA under more relaxed assumptions.

6.5 Numerical Validation

Recent work has shown the excellent performance of PnP for smooth data-fidelity terms using advanced denoising priors. Our goal in this section is to extend these studies with simulations validating the effectiveness of IPA for nonsmooth data-fidelity terms and deep neural net priors, as well as demonstrating its scalability to large-scale inverse problems. We consider two applications of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{e} \in \mathbb{R}^m$ denotes the noise and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ denotes either a random Gaussian matrix in *compressive sensing (CS)* or the transfer function in *intensity diffraction tomography (IDT)* [109].

Our deep neural net prior is based on the DnCNN architecture [216], with its batch normalization layers removed for controlling the Lipschitz constant of the network via spectral normalization [160] (see details in Appendix E.6.1). We train a nonexpansive residual network R_σ by predicting the noise residual from its noisy input. While this means that R_σ is not trained to be firmly nonexpansive, we observed that nonexpansiveness was sufficient for empirical convergence. Note also that a nonexpansive R_σ satisfies the necessary (but not sufficient) condition for firm nonexpansiveness of D_σ. It is also worth mentioning that denoiser design, which is not our main focus, is an active area of research in the context of PnP. The training data is generated by adding AWGN to the BSD400 images [124]. The reconstruction quality is quantified using the signal-to-noise ratio (SNR) in dB. We pre-train several deep neural net models as denoisers for $\sigma \in [1, 10]$, using σ intervals of 0.5, and use the denoiser achieving the best SNR.

6.5.1 Integration of Nonsmooth Data-Fidelity Terms and Pretrained Deep Priors

We first test IPA on non-smooth data-fidelity terms. The matrix $\boldsymbol{A}$ is generated with i.i.d. zero-mean Gaussian random elements of variance $1/m$, and $\boldsymbol{e}$ as a sparse Bernoulli-Gaussian vector with the sparsity ratio of 0.1. This means that, in expectation, ten percent of the elements of $\boldsymbol{y}$ are contaminated by AWGN. The sparse nature of the noise motivates the usage of the ℓ_1-norm $g(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1$, since it is less sensitive to extreme values. The nonsmoothness of ℓ_1-norm prevents the usage of gradient-based algorithms such as PnP-SGD. On the other hand, the application IPA is facilitated by efficient strategies for computing the proximal operator [18, 34].

Note that the focus of this section is on using CS as a convenient application for demonstrating some of the key properties of IPA, and is not on achieving the state-of-the-art subsampling in CS [104, 131, 163, 210, 214]. For any subsampling rate, the reconstruction quality of IPA is expected to match that of PnP-ADMM, which has been extensively studied in prior work. In particular, a recent work [114] has extensively compared the recovery performance of PnP relative to several widely-used algorithms in CS.

We set the measurement ratio to be approximately $m/n = 0.7$ with AWGN of standard deviation 5. Twelve standard images from *Set12* [216] are used in testing, each resized to 64×64 pixels for rapid parameter tuning and testing. We quantify the convergence accuracy using the normalized distance $\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2/\|\boldsymbol{v}^k\|_2^2$, which is expected to approach zero as IPA converges to a fixed point.

Theorem 6.1 characterizes the convergence of IPA in terms of $\|\mathsf{S}(\boldsymbol{v}^k)\|_2$ up to a constant error term that depends on γ. This is illustrated in Fig. 6.1 for three values of the penalty parameter

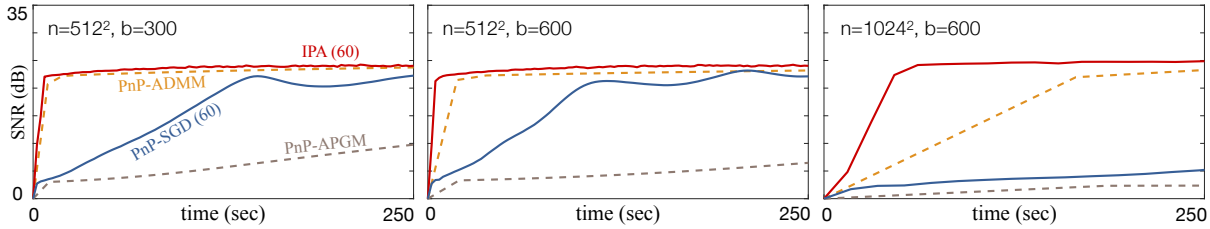

Figure 6.2: Illustration of scalability of IPA and several widely used PnP algorithms on problems of different sizes. The parameters n and b denote the image size and the number of acquired intensity images, respectively. The average SNR is plotted against time in seconds. Both IPA and PnP-SGD use random minibatches of 60 measurements at every iteration, while PnP-ADMM and PnP-APGM use all the measurements. The figure highlights the fast empirical convergence of IPA compared to PnP-SGD as well as its ability to address larger problems compared to PnP-ADMM and PnP-APGM.

$\gamma \in \{\gamma_0, \gamma_0/2, \gamma_0/4\}$ with $\gamma_0 = 0.02$. The average normalized distance $\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2/\|\boldsymbol{v}^k\|_2^2$ and SNR are plotted against the iteration number and labeled with their respective final values. The shaded areas represent the range of values attained across all test images. IPA is implemented to use a random half of the elements in $\boldsymbol{y}$ in every iteration to impose the data-consistency. Fig. 6.1 shows the improved convergence of IPA to $\mathsf{zer}(\mathsf{S})$ for smaller values of γ, which is consistent with our theoretical analysis. Specifically, the final accuracy improves approximately $3\times$ (from 1.07×10^{-5} to 3.59×10^{-6}) when γ is reduced from γ_0 to $\gamma_0/4$. On the other hand, the SNR values are nearly identical for all three experiments, indicating that in practice different γ values lead to fixed points of similar quality. This indicates that IPA can achieve high-quality result without taking $\|\mathsf{S}(\boldsymbol{v}^k)\|_2$ to zero.

6.5.2 Scalability in Large-scale Optical Tomography

We now discuss the scalability of IPA on intensity diffraction tomography (IDT), which is a data intensive computational imaging modality [109]. The goal is to recover the spatial distribution of the *complex* permittivity contrast of an object given a set of its intensity-only measurements. In this problem, $\boldsymbol{A}$ consists of a set of b complex matrices $[\boldsymbol{A}_1, \ldots, \boldsymbol{A}_b]^\mathsf{T}$, where

Table 6.1: Final average SNR (dB) and Runtime obtained by several PnP algorithms on all test images.

Simulations	**Parameters**		$n = 512^2$ ($b = 300$)	$n = 512^2$ ($b = 600$)	$n = 1024^2$ ($b = 600$)
Algorithms	σ	γ	SNR in dB (Runtime)		
PnP-APGM	1	5×10^{-4}	22.60 (19.4 min)	22.79 (42.6 min)	23.56 (8.1 hr)
PnP-SGD (60)	1	5×10^{-4}	22.31 (7.1 min)	22.74 (5.2 min)	23.42 (44.3 min)
PnP-ADMM	2.5	1	**24.23** (7.4 min)	**24.40** (14.7 min)	**25.50** (1.4 hr)
IPA (60)	2.5	1	23.65 (**1.7 min**)	23.88 (**2 min**)	24.95 (**11 min**)

each $\boldsymbol{A}_i$ is a convolution corresponding to the ith measurement $\boldsymbol{y}_i$. We adopt the ℓ_2-norm loss $g(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$ as the data-fidelity term to empirically compare the performance of IPA and PnP-SGD on the same problem. PnP-SGD has been implemented with Nesterov acceleration, as in [168].

In the simulation, we follow the experimental setup in [109] under AWGN corresponding to an input SNR of 20 dB. We select six images from the CAT2000 dataset [24] as our test examples, each cropped to n pixels. We assume real permittivity functions, but still consider complex valued measurement operator $\boldsymbol{A}$ that accounts for both absorption and phase [109]. Due to the large size of data, we process the measurements in epochs using minbatches of size 60.

Fig. 6.2 illustrates the evolution of average SNR against runtime for several PnP algorithms, namely PnP-ADMM, PnP-APGM, PnP-SGD, and IPA, for images of size $n \in \{512 \times 512, 1024 \times 1024\}$ and the total number of intensity measurements $b \in \{300, 600\}$. The final values of SNR as well as the total runtimes are summarized in Table 6.1. The table highlights the overall best SNR performance in bold and the shortest runtime in light-green. In every iteration, PnP-ADMM and PnP-APGM use all the measurements, while IPA and PnP-SGD

Table 6.2: Per-iteration memory usage specification for reconstructing 1024×1024 images

Algorithms		PnP-ADMM		IPA (**Ours**)	
Variables		size	memory	size	memory
$\{\boldsymbol{A}_i\}$	real	$1024 \times 1024 \times 600$	9.38 GB	$1024 \times 1024 \times 60$	0.94 GB
	imaginary	$1024 \times 1024 \times 600$	9.38 GB	$1024 \times 1024 \times 60$	0.94 GB
$\{\boldsymbol{y}_i\}$		$1024 \times 1024 \times 600$	18.75 GB	$1024 \times 1024 \times 60$	1.88 GB
others combined		—	0.13 GB	—	0.13 GB
Total			**37.63 GB**		**3.88 GB**

use only a small subset of 60 measurements. IPA thus retains its effectiveness for large values of b, while batch algorithms become significantly slower. Moreover, the scalability of IPA over PnP-ADMM becomes more notable when the image size increases. For example, Table 6.1 highlights the convergence of IPA to 24.95 dB within 11 minutes, while PnP-ADMM takes 1.4 hours to reach a similar SNR value. Note the rapid progress of PnP-ADMM in the first few iterations, followed by a slow but steady progress until its convergence to the values reported in Table 6.1. This behavior of ADMM is well known and has been widely reported in the literature (see *Section 3.2.2 "Convergence in Practice"* in [29]). We also observe faster convergence of IPA compared to both PnP-SGD and PnP-APGM, further highlighting the potential of IPA to address large-scale problems where partial proximal operators are easy to evaluate.

Another key feature of IPA is its memory efficiency due to incremental processing of data. The memory considerations in optical tomography include the size of all the variables related to the desired image $\boldsymbol{x}$, the measured data $\{\boldsymbol{y}_i\}$, and the variables related to the forward model $\{\boldsymbol{A}_i\}$. Table 6.2 records the total memory (GB) used by IPA and PnP-ADMM for reconstructing a 1024×1024 pixel permittivity image, with the smallest value highlighted in light-green. PnP-ADMM requires 37.63 GB of memory due to its batch processing of the

whole dataset, while IPA uses only 3.88 GB—nearly *one-tenth* of the former—by adopting incremental processing of data. In short, our numerical evaluations highlight both fast and stable convergence and flexible memory usage of IPA in the context of large-scale optical tomographic imaging.

6.6 Summary

This chapter provides several new insights into the widely used PnP methodology in the context of large-scale imaging problems. First, we have proposed IPA as a new incremental PnP algorithm. IPA extends PnP-ADMM to randomized partial processing of measurements and extends traditional optimization-based ADMM by integrating pre-trained deep neural nets. Second, we have theoretically analyzed IPA under a set of realistic assumptions, showing that in expectation IPA can approximate the convergence behavior of PnP-ADMM to a desired precision by controlling the penalty parameter. Third, our simulations highlight the potential of IPA to handle nonsmooth data-fidelity terms, large number of measurements, and deep neural net priors. We observed faster convergence of IPA compared to several baseline PnP methods, including PnP-ADMM and PnP-SGD, when partial proximal operators can be efficiently evaluated. IPA can thus be an effective alternative to existing algorithms for addressing large-scale imaging problems. For future work, we would like to explore strategies to further relax our assumptions and explore distributed variants of IPA to enhance its performance in parallel settings.

Part IV

Extending Plug-and-Play Priors to the Non-Euclidean Setting

Chapter 7

Bregman Plug-and-Play Priors

The state-of-the-art performance of PnP, RED, and DU has been validated in a variety o applications. However, the current paradigm for designing such algorithms is inherently Euclidean, due to the usage of the quadratic norm within the projection and proximal operators. We propose to broaden this perspective by considering a non-Euclidean setting based on the more general Bregman distance. Our new Bregman Proximal Gradient Method variant of PnP (PnP-BPGM) and Bregman Steepest Descent variant of RED (RED-BSD) replace the traditional updates in PnP and RED from the quadratic norms to more general Bregman distance. We present a theoretical convergence result for PnP-BPGM and demonstrate the effectiveness of our algorithms on Poisson linear inverse problems. [6]

[6]This chapter is based on our paper [6].

Algorithm 9 PnP-BPGM

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\gamma > 0$
for $k = 1, 2, \ldots$ **do**
    $\boldsymbol{x}^k \leftarrow \mathsf{D}_{\boldsymbol{\theta}} \left(\nabla h^* \left(\nabla h - \gamma \nabla g \right) \right) \left(\boldsymbol{x}^{k-1} \right)$
end for

7.1 Introduction

In previous chapters, we have focused on the linear forward model (2.3) that models the relationship between the measurements $\boldsymbol{y} \in \mathbb{R}^m$ and an unknown signal $\boldsymbol{x} \in \mathbb{R}^n$ as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \tag{7.1}$$

where where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a linear forward operator and we assume that the noise $\boldsymbol{e}$ is AWGN. Now let's consider a more general model with noise corruption denoted by the operator $\mathcal{P}$

$$\boldsymbol{y} = \mathcal{P}(\boldsymbol{A}\,\boldsymbol{x}). \tag{7.2}$$

We note that $\mathcal{P}$ models a more general noise corruption than the additive one in formulation (7.1), in a sense that it could be either additive (e.g., Gaussian noise) or non-additive (e.g., Poisson noise). Similar to the problem (7.1), the solution of ill-posed inverse problems (7.2) can also be formulated as an regularized optimization problem

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\mathsf{arg\,min}}\, f(\boldsymbol{x}), \quad \text{where} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + r(\boldsymbol{x}), \tag{7.3}$$

where g is the data-fidelity term and r is the regularizer.

Most of the current work in PnP is fundamentally based on the traditional definition of the proximal operator that relies on the squared Euclidean norm. Under this definition the

Algorithm 10 RED-BSD

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\gamma > 0$
for $k = 1, 2, \ldots$ **do**
    $\boldsymbol{x}^k \leftarrow \nabla h^* \left(\nabla h - \gamma \left(\nabla g + \tau \left(I - \mathsf{D}_{\boldsymbol{\theta}} \right) \right) \right) (\boldsymbol{x}^{k-1})$
end for

proximal operator can be naturally interpreted as the Gaussian denoiser. In this chapter, we seek to broaden the family of PnP algorithms to the non-Euclidean setting by building on the recent work on Bregman proximal algorithms [11, 118, 175]. Specifically, we propose to generalize the well-known PnP-PGM [91] and RED-SD [153] algorithms to their Bregman counterparts, PnP-BPGM and RED-BSD algorithms, by using the Bregman distance. We learn the corresponding artifact-removal operators by unfolding the iterations of our algorithms. Finally, we present the theoretical convergence analysis of PnP-BPGM and test our algorithms on Poisson linear inverse problems.

7.2 Background

7.2.1 Recap of PGM

As what we have introduce in Section 2.2.1, PGM can be interpreted as the majorization-minimization (MM) method for solving the composite optimization problem in (7.3). Each iteration of PGM can be expressed as a minimization of a quadratic majorizer

$$\boldsymbol{x}^k = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\operatorname{arg\,min}} \left\{ \boldsymbol{x}^{\mathsf{T}} \nabla g(\boldsymbol{x}^{k-1}) + \frac{L}{2} \left\| \boldsymbol{x} - \boldsymbol{x}^{k-1} \right\|^2 + r(\boldsymbol{x}) \right\}, \tag{7.4}$$

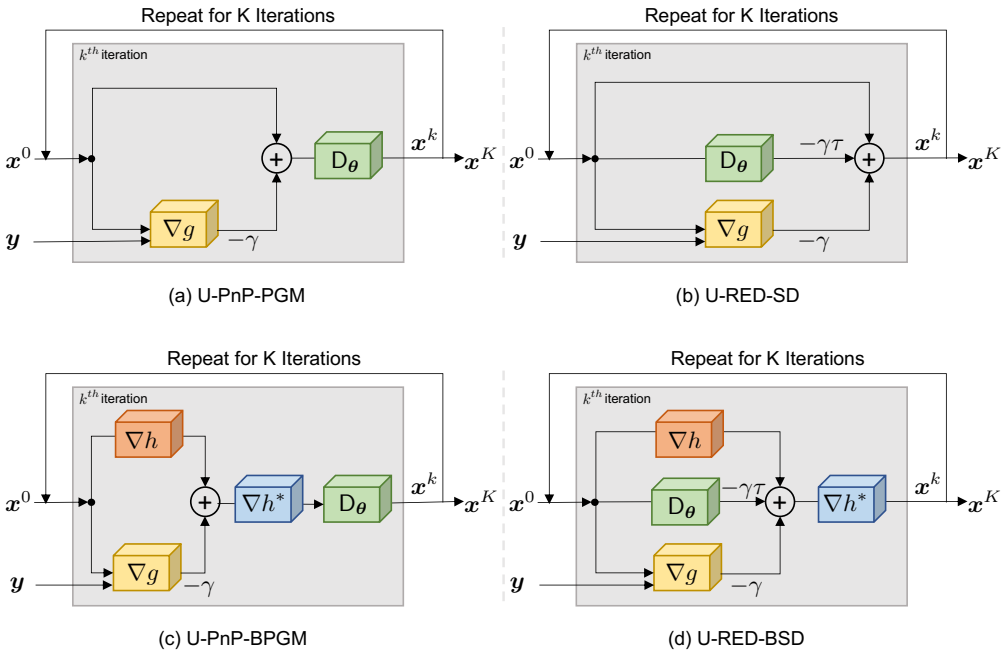

Figure 7.1: The proposed PnP-BPGM and RED-BSD methods replace the quadratic penalty in PnP-PGM and RED-SD by a more general Bregman distance. Both algorithms rely on data-driven regularizers obtained by training an artifact-removal operator $\mathsf{D}_{\boldsymbol{\theta}}$ via deep unfolding.

where g is assumed to have a L-Lipschitz continuous gradient. Eq. (7.4) can also be expressed in the following standard form

$$\boldsymbol{z}^k = \boldsymbol{x}^{k-1} - \gamma \nabla g(\boldsymbol{x}^{k-1}) \tag{7.5a}$$

$$\boldsymbol{x}^k = \mathsf{prox}_{\gamma r}(\boldsymbol{z}^k), \tag{7.5b}$$

where $0 < \gamma \leq 1/L$ is the step size.

7.2.2 The Bregman Distance

Given a differentiable convex reference function h defined on a closed convex set $C \subset \mathbb{R}^n$, the Bregman distance $B_h : \operatorname{dom} h \times \operatorname{int} \operatorname{dom} h \to [0, \infty)$ [30] is defined by

$$B_h(\boldsymbol{x}; \boldsymbol{y}) \coloneqq h(\boldsymbol{x}) - h(\boldsymbol{y}) - \nabla h(\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}). \tag{7.6}$$

The Bregman distance[7] is an extension of the classical squared Euclidean distance which is recovered when $h(\boldsymbol{x}) = 1/2\|\boldsymbol{x}\|_2^2$. Typical examples of the Bregman distance include the following:

- Squared Euclidean distance with $h(\boldsymbol{x}) = 1/2\|\boldsymbol{x}\|^2$,
- Squared Mahalanobis distance with $h(\boldsymbol{x}) = 1/2\|\boldsymbol{x}\|_{\boldsymbol{Q}}^2 = 1/2\, \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{x}, \boldsymbol{Q} \succcurlyeq 0$,
- The generalized Kullback–Leibler divergence with Shannon Entropy $h(\boldsymbol{x}) = \boldsymbol{x}\mathsf{log}(\boldsymbol{x})$,
- Itakura–Saito (IS) distance with Brug's entropy $h(\boldsymbol{x}) = -\mathsf{log}(\boldsymbol{x})$.

7.2.3 Using Learning Priors in Deep Unfolding

Compared with the methods that integrate pre-trained DL denoisers into iterative algorithms such as PnP and RED, *deep unfolding (DU)* is a related strategy based on unfolding an iterative algorithm and including trainable blocks within it [69]. Compared to the black-box DL, model-based DL methods integrate the physics-based knowledge of the measurement model. Their empirical success [115, 166], has spurred a number of algorithmic extensions [5, 91, 141], as well as theoretical convergence analyses [39, 158, 168].

[7]Note that the Bregman distance is a pesudodistance, because it does not satisfy the triangle inequality, and is generally asymmetric.

7.3 Proposed Method

The path to Bregman-based proximal algorithm starts from observing that the quadratic majorization step in the classical PGM in eq. (7.4) is equivalent to the following condition

$$\frac{L}{2}\|\boldsymbol{x}\|^2 - g(\boldsymbol{x}) \quad \text{is convex,} \tag{7.7}$$

where the equivalence follows from the first-order convexity inequality (see Section 2.2.1). To bypass the Lipschitz gradient assumption, the work in [11, 118] has proposed to generalize the condition in eq. (7.7) by using a possibly non-quadratic reference function h

$$L\,h(\boldsymbol{x}) - g(\boldsymbol{x}) \quad \text{is convex.} \tag{7.8}$$

Such functions g can be referred to as L-smooth relative to h. Then, by using the first-order convexity inequality, one can obtain a Bregman majorizer of the data-fidelity term

$$g(\boldsymbol{x}) \leq g(\boldsymbol{x}^k) + \nabla g(\boldsymbol{x}^k)^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{x}^k) + L\,B_h(\boldsymbol{x}, \boldsymbol{x}^k). \tag{7.9}$$

This inequality directly leads to the Bregman PGM (BPGM) method, which generalizes the classical PGM using a Bregman majorizer as

$$\boldsymbol{x}^k = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}} \left\{ \boldsymbol{x}^{\mathsf{T}} \nabla g(\boldsymbol{x}^{k-1}) + L\,B_h(\boldsymbol{x}, \boldsymbol{x}^{k-1}) + r(\boldsymbol{x}) \right\}. \tag{7.10}$$

The BPGM method shares the same structural splitting mechanism as the classical PGM [175], which allows one to express (7.10) as

$$\boldsymbol{z}^k = \nabla h^* \left(\nabla h - \gamma \nabla g \right) (\boldsymbol{x}^{k-1}) \tag{7.11a}$$

$$\boldsymbol{x}^k = \left(\nabla h + \gamma \, \partial r \right)^{-1} \nabla h(\boldsymbol{z}^k) \tag{7.11b}$$

where $0 < \gamma \leq 1/L$ is the step size and h^* denotes the Fenchel conjugate of h. Note that the PGM is a special case of the BPGM obtained by setting $h(\boldsymbol{x}) = 1/2\|\boldsymbol{x}\|_2^2$. The first step of the BPGM in (7.11a) is known as the *Mirror Descent (MD)* algorithm, which generalizes the classical gradient method. The second step is known as the *left* Bregman proximal operator (BPO) defined as

$$\mathsf{prox}_{\gamma r}^{h}(\boldsymbol{z}) \coloneqq \underset{\boldsymbol{x} \in \mathbb{R}^n}{\mathsf{arg\,min}} \left\{ B_h(\boldsymbol{x}, \boldsymbol{z}) + \gamma r(\boldsymbol{x}) \right\}. \tag{7.12}$$

Traditionally, the BPO is motivated from a computational perspective, e.g., Bregman projection onto the simplex with $h(\boldsymbol{x}) = \boldsymbol{x}^\mathsf{T} \mathsf{log}(\boldsymbol{x})$ is simpler than the corresponding classical proximal operator (2.14). Moreover, selecting the reference function h provides more flexibility depending on the problem settings [11, 118, 175]. Meanwhile, similar to the case where the classical proximal operator can be interpreted as a Gaussian denoiser, under some conditions on the reference function h, the BPO can be interpreted as an exponential family mean estimator. As a reference, we include such statistical interpretation of the backward step in the left BPO in Appendix F.1.

7.3.1 Bregman PnP and RED Algorithms

In this section, we propose two algorithms, *PnP-BPGM* and *RED-BSD* that extend existing two algorithms PnP-PGM (summarized in Algorithm 5.3) and RED-SD (summarized in

Algorithm 2.38), respectively. PnP-BPGM is obtained by replacing the BPO in (7.11b) with a DL network $\mathsf{D}_{\boldsymbol{\theta}}$

$$\boldsymbol{z}^k = \nabla h^* \left(\nabla h - \gamma \nabla g\right)(\boldsymbol{x}^{k-1}) \tag{7.13a}$$

$$\boldsymbol{x}^k = \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{z}^k), \tag{7.13b}$$

where $\boldsymbol{\theta}$ are the learnable parameters that characterize the network $\mathsf{D}_{\boldsymbol{\theta}}$. Similarly, the Bregman variant of RED-SD is obtained as

$$\boldsymbol{x}^k = \nabla h^* \left(\nabla h - \gamma \left(\nabla g + \tau \left(\mathsf{I} - \mathsf{D}_{\boldsymbol{\theta}}\right)\right)\right)(\boldsymbol{x}^{k-1}), \tag{7.14}$$

where I is an identity operator. When the assumptions for the existence of the explicit RED regularizer in [153] hold, then RED-BSD can be interpreted as the mirror descent algorithm. Note that the PnP-PGM [186] and RED-SD [153] are recovered when $h(\boldsymbol{x}) = 1/2\|\boldsymbol{x}\|^2$ and $\mathsf{D}_{\boldsymbol{\theta}}$ being a Gaussian denoiser.

Algorithm 9 and Algorithm 10 summarize the proposed PnP-BPGM and RED-BSD algorithms. In this work, the regularizer $\mathsf{D}_{\boldsymbol{\theta}}$ is implemented using the deep unfolded strategy, so we refer to the proposed algorithms as unfolded PnP-BPGM (U-PnP-BPGM) and unfolded RED-BSD (U-RED-BSD). Similarly, the unfolded version of PnP-PGM, and RED-SD are referred as U-PnP-PGM and U-RED-SD. All four different unfolding architectures are shown in Fig. 7.1 and will be compared in the next section.

Recent work has explored the convergence properties of various PnP/RED algorithms [39, 158, 168]. Similar results can be also established for both PnP-BPGM and RED-BSD. The following theorem presents the analysis of PnP-BPGM for a strongly convex function g and a Lipschitz continuous operator $\mathsf{D}_{\boldsymbol{\theta}}$. While these assumptions are too strong for some

applications, they provide the first steps for the broader analysis of Bregman PnP/RED methods.

Theorem 7.1. *Assume h be μ_h-strongly convex with L_h-Lipschitz continuous gradient, and g be μ_g-strongly convex function with L_g-Lipschitz continuous gradient. Assume $\mathsf{D}_{\boldsymbol{\theta}}$ be an M-Lipschitz operator. Then, the iteration in* (7.13) *converges to a fixed point if*

$$M < \frac{\mu_h \left(\mu_g + L_g\right)}{L_h \, L_g - \mu_h \, \mu_g} \tag{7.15}$$

and the step size $\frac{\mu_h}{\mu_g}\left(\frac{L_h}{\mu_h} - \frac{1}{M}\right) < \gamma < \frac{\mu_h}{L_g}\left(1 + \frac{1}{M}\right)$.

Proof. See Appendix F.2.

7.3.2 Poisson Linear Inverse Problem

We empirically evaluated the proposed methods on Poisson linear inverse problems. Poisson noise is a signal dependent noise whose negative log-likelihood function results in the following data-fidelity term and its gradient

$$g(\boldsymbol{x}) = \mathbf{1}^{\mathsf{T}} \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}^{\mathsf{T}} \mathsf{log}\left(\boldsymbol{A}\boldsymbol{x}\right) + \mathbf{1}^{\mathsf{T}} \mathsf{log}(\boldsymbol{y}!) \tag{7.16a}$$

$$\nabla g(\boldsymbol{x}) = \boldsymbol{A}^{\mathsf{T}} \left(\mathbf{1} - \boldsymbol{y} \oslash \left(\boldsymbol{A}\boldsymbol{x}\right)\right) \tag{7.16b}$$

where $\mathbf{1}$ is a vector of ones, and $\oslash$ denotes element-wise division.

Classical algorithms for solving Poisson linear inverse problems include the Richardson–Lucy (RL) algorithm and transform-based methods [48, 52, 74, 123, 167]. Several ADMM-based algorithms were proposed that handle the data fidelity via its proximal operator [63, 154]. In [11] it was showed that by using the Burg's entropy as a reference function $h(\boldsymbol{x}) = -\mathbf{1}^{\mathsf{T}} \mathsf{log}(\boldsymbol{x})$,

one can satisfy Eq. (7.8) with $L \geq \|\boldsymbol{y}\|_1$. Therefore, using (7.13) we can obtain the following simple iteration for PnP-BPGM

$$\boldsymbol{z}^k = \frac{\boldsymbol{x}^{k-1}}{\mathbf{1} + \gamma\, \boldsymbol{x}^{k-1} \odot \nabla g(\boldsymbol{x}^{k-1})} \tag{7.17a}$$

$$\boldsymbol{x}^k = \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{z}^k), \tag{7.17b}$$

where $\odot$ is the element-wise multiplication. It can be shown that the backward operator is related to inverse Gamma scale estimator. Similarly, RED-BSD in (7.14) can be simplified to

$$\boldsymbol{x}^k = \frac{\boldsymbol{x}^{k-1}}{\mathbf{1} + \gamma\, \boldsymbol{x}^{k-1} \odot (\nabla g + \tau\,(I - \mathsf{D}_{\boldsymbol{\theta}}))(\boldsymbol{x}^{k-1})}. \tag{7.18}$$

7.4 Numerical Illustration

7.4.1 Image Deblurring with Poisson Noise

We demonstrate the ability of our proposed algorithms PnP-BPGM and RED-BSD over their traditionally counterparts PGM and RED on Poisson linear inverse problems. We focus on image deblurring, where the forward model $\boldsymbol{A}$ corresponds to the blurring operator. Specifically, we follow a similar settings in [63, 154], and test our algorithms for Poisson noise with peaks 8 and 32 using two different blur kernels of size 9 by 9: (1) a Gaussian kernel with $\sigma = 1.6$, and (2) a uniform kernel, respectively. All the methods compared are trained in an unfolding fashion as illustrated in Fig. 7.1, where the end-to-end training seeks to compute the trainable parameters in $\mathsf{D}_{\boldsymbol{\theta}}$ by minimizing the ℓ_2 loss function between network output $\{\boldsymbol{x}^K\}$ and the ground-truth $\{\boldsymbol{x}\}$ over all training samples. We set $\boldsymbol{x}^0$ using the raw measurements $\boldsymbol{y}$ with a small white Gaussian perturbation. We unfold each algorithm with $K = 100$ iterations

Table 7.1: The PSNR (dB) results of different methods on the testing images with different peaks and kernels.*

Method	1	2	3	4	5	6	7	8	9	10	11	12	Average
Uniform kernel, peak = 8													
Corrupted	11.70	11.13	11.74	11.59	11.61	9.56	11.81	11.80	11.82	11.63	12.04	11.91	11.53
U-Net	20.89	23.11	21.28	20.79	19.79	19.15	19.63	24.76	21.96	22.70	23.37	22.65	21.67
U-P-PGM	19.57	22.74	20.89	20.59	19.20	19.15	19.04	24.38	21.79	22.43	23.19	22.44	21.28
U-R-SD	20.38	22.74	20.74	20.29	18.81	19.00	18.90	24.44	21.81	22.38	23.32	22.41	21.27
U-P-BPGM	21.00	24.12	21.27	20.72	20.10	19.17	19.81	25.21	22.13	22.65	23.82	22.62	21.89
U-R-BSD	20.97	23.97	21.14	20.82	20.25	19.28	19.78	25.08	22.13	22.70	23.75	22.66	21.88
Uniform kernel, peak = 32													
Corrupted	16.14	16.08	16.51	16.25	15.76	13.97	15.81	17.12	16.68	16.77	17.13	16.95	16.26
U-Net	21.50	24.66	22.12	21.71	21.01	19.97	20.23	26.19	22.56	23.38	24.67	23.40	22.62
U-P-PGM	20.90	23.76	22.03	21.54	20.54	19.71	19.85	25.37	22.22	23.26	24.01	23.32	22.21
U-R-SD	21.37	24.13	21.92	21.41	20.75	19.88	20.17	25.67	22.40	23.35	24.25	23.40	22.39
U-P-BPGM	21.58	25.01	22.15	21.81	21.64	20.23	20.57	26.33	22.64	23.45	24.90	23.46	22.81
U-R-BSD	21.57	25.04	22.17	21.64	21.48	20.22	20.34	26.44	22.65	23.35	24.91	23.41	22.77
Gaussian kernel, peak = 8													
Corrupted	11.98	11.25	12.01	11.86	12.01	9.71	12.32	11.89	11.89	11.79	12.17	12.07	11.75
U-Net	21.72	24.92	22.06	21.55	22.17	20.87	21.27	25.60	22.22	22.97	24.63	23.08	22.76
U-P-PGM	21.01	23.97	21.70	21.41	20.74	19.72	20.32	25.18	22.17	22.72	24.17	22.87	22.16
U-R-SD	21.18	23.30	21.88	21.26	20.65	19.79	20.15	24.86	21.96	22.97	23.69	23.03	22.06
U-P-BPGM	22.30	24.60	22.48	21.78	22.44	19.23	21.92	26.03	22.48	23.90	24.49	23.60	22.94
U-R-BSD	22.22	24.62	22.17	21.72	22.27	19.61	21.61	25.76	22.37	23.79	24.37	23.60	22.84
Gaussian kernel, peak = 32													
Corrupted	17.06	16.62	17.30	17.17	17.05	14.45	17.19	17.51	17.04	17.35	17.57	17.55	16.99
U-Net	22.63	26.74	23.13	23.13	23.83	21.69	22.51	27.14	22.89	24.00	25.95	24.12	23.98
U-P-PGM	22.12	24.50	23.61	23.10	22.54	19.53	21.81	26.03	22.60	24.27	24.77	24.22	23.26
U-R-SD	22.15	25.43	23.07	23.14	22.86	21.29	21.92	26.55	22.80	24.27	25.31	24.24	23.58
U-P-BPGM	23.41	26.85	23.79	23.54	24.41	20.82	23.30	27.86	23.13	25.03	26.03	24.83	24.42
U-R-BSD	23.12	26.79	23.41	23.27	24.46	21.02	23.05	27.88	23.11	24.96	26.04	24.75	24.32

* Due to the space limitation, we denote U-PnP-PGM, U-RED-SD, U-PnP-BPGM,U-RED-BSD as U-P-PGM, U-R-SD, U-P-BPGM and U-R-BSD in respective order in the table.
The best performance in each scenario is highlighted in blue.

for stable performance, where in each iteration, the network $\mathsf{D}_{\boldsymbol{\theta}}$ is realized using a 7−layer DnCNN [216] with shared weights across all iterations. The step-size parameter γ and the regularization parameter τ in RED and BRED are set as a learnable parameters, initialized with $\gamma = 5 \times 10^{-1}$ and $\tau = 1 \times 10^{-3}$. As a reference, we also report the image reconstruction performance of the end-to-end learning method where U-Net is trained end-to-end in the

Figure 7.2: Examples of image reconstruction results on *Babara* (top) and *Cameraman* (bottom) obtained by U-Net, U-PnP-PGM, U-RED-SD, U-PnP-BPGM, and U-RED-BSD. The first row is corresponding to the noise peak 8 with uniform kernel, and the second row is noiser peak 32 with Gaussian kernel. Each reconstruction is labeled with its PSNR (dB) value with respect to the Ground-truth image. Visual differences are highlighted using the rectangles drawn inside the images. Note U-PnP-BPGM and U-RED-BSD shows close performance one to another, outperforming other methods and providing the best visual results by recovering sharp edges and removing artifacts.

usual supervised fashion using the ℓ_2-loss [87, 155]. All networks are trained on public dataset BSD400 for 400 epochs, using the Adam solver [99] with an initial learning rate 1×10^{-4}. We select the models that achieved the best performance on the validation dataset BSD68. At test time, Set12 dataset is used to evaluate the performance of each algorithm.

The numerical results on the test dataset Set12 with respect to two scenarios are summarized in Table 7.1. Test images used for the quantitative performance labeled from 1 to 12 are: *Cameraman*, *House*, *Pepper*, *Starfish*, *Butterfly*, *Plane*, *Parrot*, *Lena*, *Barbara*, *Boat*, *Artist*, *Room*. For each image, the highest PSNR in each scenario is highlighted. We observe that the performances of U-PnP-BPGM and U-RED-BSD are very close to one another, providing the best performance compared to all the other methods, outperforming U-PnP-PGM and U-RED-SD. Fig. 7.2 shows visual examples for two images from Set12 in two different settings,

uniform kernel with peak 8 (top) and Gaussian kernel with peak 32 (bottom). Note that both U-PnP-PGM and U-RED-SD yield similar visual recovery performance with artifacts remaining in the images, U-PnP-BPGM and U-RED-BSD show much better reconstruction performance in removing artifacts and noise. The enlarged regions in the image suggest that U-PnP-BPGM and U-RED-BSD better recover the fine details and sharper edges compared to their counterparts and U-Net.

7.5 Summary

In this chapter, we propose generalizing plug-and-play priors (PnP) and regularization by denoising (RED) beyond squared Euclidean distance using the Bregman distance. The proposed Bregman-based methods are motivated by the recent progress in optimization, that have the potential to better align to specific non-Euclidean geometry of the loss function. Our numerical results show the potential of the proposed methods in Poisson linear inverse problems. This work can be considered as a first step towards extending widely-used PnP/RED to problems where there is a benefit of using non-Euclidean formulations of proximal and projection operators.

Part V

Developing Model-based Deep Learning Algorithms

Chapter 8

CoRRECT: A Deep Unfolding Framework for Motion-Corrected Quantitative R_2^* Recovery

We have analyzed PnP and RED in previous chapters. PnP/RED leverages the power of deep learning by using learning-based denoisers inside model-based optimization approaches. In this chapter, we focus on the deep unfolding (DU) frameworks, which are methods that integrate imaging model inside the end-to-end deep learning. In particular, we present CoRRECT as a new framework for recovering quantitative R_2^* maps from subsampled and artifact-corrupted MRI data using DU. Quantitative MRI (qMRI) refers to a class of techniques for quantifying the spatial distribution of biological tissue parameters using MRI. Traditional qMRI methods deal separately with artifacts arising from accelerated data acquisition, involuntary physical motion, and magnetic-field inhomogeneities, leading to suboptimal performance. This work addresses all three jointly by proposing

CoRRECT, a specialized DU method consisting of a model-based end-to-end neural network, an efficient motion-artifact simulator, and a self-supervised learning scheme. The network is trained to make the k-space data corresponding to the estimated R_2^* maps resemble the real data by accounting for motion and field inhomogeneities. When deployed, CoRRECT uses only the k-space data without any pre-computed parameters for motion or inhomogeneity correction. Our results on simulated and experimentally collected multi-Gradient-Recalled Echo (mGRE) MRI data show that CoRRECT recovers high-quality R_2^* maps in highly accelerated acquisition settings. This work opens the door to DU methods that can integrate information from physical measurement models, biophysical signal models, and learned prior models for high-quality quantitative MRI.

8.1 Introduction

The recovery of diagnostic-quality images from subsampled k-space measurements is fundamental to accelerated *magnetic resonance imaging (MRI)* [120]. The recovery is traditionally formulated as an *inverse problem*, where the unknown image is reconstructed by combining the MRI forward model and a regularizer [45, 54, 80, 156]. *Deep learning (DL)* has recently enabled a powerful data-driven paradigm for solving inverse problems, leading to new state-of-the-art MRI methods [100, 119, 127, 140, 187]. Instead of defining an explicit regularizer, the traditional DL methods are based on training *convolutional neural networks (CNNs)* to map the measured data to the desired high-quality image. Model-based DL methods—such as those based on PnP and DU—have extended the traditional DL to deep architectures that combine the MRI forward models and CNN regularizers [1, 3, 75, 79, 134, 205, 214].

Quantitative MRI (qMRI) refers to a class of techniques for quantifying the spatial distribution of biological tissue parameters from MRI data [77, 101, 150, 184, 192, 197, 218, 219]. qMRI

scans are relatively slow due to their reliance on acquisition sequences that require a large number k-space samples. Additionally, the recovered quantitative maps frequently suffer from undesirable imaging artifacts due to various sources of noise and corruption in the measurements. Three common sources of artifacts are the measurement noise, macroscopic B_0 magnetic field inhomogeneities, and involuntary physical motion of the object during signal acquisition. There is consequently a need for qMRI methods that can recover high-quality quantitative parameters from accelerated MRI data contaminated by measurement noise, field inhomogeneities, and motion artifacts.

Despite the rich literature on qMRI, the majority of methods consider the artifacts arising from accelerated data acquisition, involuntary physical motion, and magnetic-field inhomogeneities separately. In particular, it is common to view qMRI parameter estimation as a post-processing step decoupled from the MRI reconstruction. In this paper, we address this gap by proposing a new unified qMRI framework—referred to as *co-design of MRI reconstruction and* R_2^* *estimation with correction for motion (CoRRECT)*—for high-quality quantitative R_2^* mapping directly from noisy, subsampled, and artifact-corrupted MRI measurements. Inspired from the state-of-the-art performance of recent DU methods, we propose CoRRECT as a specialized DU method consisting of two core components: (a) a model-based end-to-end neural network, and (b) a self-supervised learning scheme for training without the ground-truth R_2^* maps. During training, the weights of the proposed network are updated to produce R_2^* maps with k-space data that resembles the real data while also accounting for object motion and magnetic field inhomogeneities. During testing, CoRRECT requires only the k-space data, without any pre-computed parameters related to motion or inhomogeneity correction, thus significantly simplifying and accelerating the imaging pipeline. We present numerical results on simulated and experimentally collected mGRE data showing that CoRRECT enables high-quality R_2^* mapping in highly accelerated data acquisition settings.

8.2 Background

8.2.1 Inverse Problem Formulation

In MRI, the relationship between the unknown complex-valued image $\boldsymbol{x} \in \mathbb{C}^n$ and its noisy k-space measurements $\boldsymbol{y} \in \mathbb{C}^m$ is commonly expressed as a linear system

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \ , \tag{8.1}$$

where $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is the measurement operator and $\boldsymbol{e} \in \mathbb{C}^m$ is the measurement noise, which is often modeled as an AWGN vector. In particular, in multi-coil parallel MRI, the measurement operator $\boldsymbol{A}$ consists of several operators representing the response of each coil [60]

$$\boldsymbol{A}^i = \boldsymbol{P}\boldsymbol{F}\boldsymbol{S}^i \ , \tag{8.2}$$

where $\boldsymbol{S}^i$ is the pixel-wise sensitivity map of the ith coil, $\boldsymbol{F}$ is the Fourier transform operator, $\boldsymbol{P}$ is the k-space sampling operator. When multiple gradient echos are used for qMRI, the sampling pattern $\boldsymbol{P}$ and the coil sensitivity maps $\{\boldsymbol{S}^i\}$ are assumed to be fixed for all echo times. We say that the MRI acquisition is "accelerated", when each coil collects $m < n$ measurements. It is common to formulate the reconstruction in accelerated MRI as the regularized optimization problem illustrated in Section 2.2.1

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{C}^n}{\mathsf{arg\,min}} \ f(\boldsymbol{x}) \quad \text{with} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + r(\boldsymbol{x}) \ , \tag{8.3}$$

where g is the data-fidelity term that quantifies consistency with the measured data $\boldsymbol{y}$ and h is a regularizer that enforces a prior knowledge on the unknown image $\boldsymbol{x}$. For example, two widely-used data-fidelity and regularization terms in accelerated MRI are the least-squares

and total variation (TV)

$$g(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\right\|^2 \quad \text{and} \quad r(\boldsymbol{x}) = \tau\left\|\boldsymbol{D}\boldsymbol{x}\right\|_1 \ , \tag{8.4}$$

where $\tau > 0$ controls the regularization strength and $\boldsymbol{D}$ is the discrete gradient operator [156].

8.2.2 Image Reconstruction using Deep Learning

In the past decade, DL has gained great popularity for solving MRI inverse problems due to its excellent performance (see reviews in [100, 212]). A widely-used supervised DL approach is to train an image reconstruction CNN $\mathsf{R}_{\boldsymbol{\theta}}$ by mapping a corrupted image $\boldsymbol{A}^{\dagger}\boldsymbol{y}$ to its clean target $\boldsymbol{x}$, where $\boldsymbol{A}^{\dagger}$ is an operator that maps the measurements back to the image domain. The training is formulated as an optimization problem over a training set consisting of desired ground-truth images $\{\boldsymbol{x}_j\}$ and their noisy subsampled measurements $\{\boldsymbol{y}_j\}$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{j=1}^{J} \mathcal{L}(\mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{A}_j^{\dagger}\boldsymbol{y}_j), \boldsymbol{x}_j) \ , \tag{8.5}$$

where $\mathcal{L}$ denotes the loss function that measures the discrepancy between the predictions of the CNN and the ground-truth. Popular choices for the CNN include U-Net [155] and for the loss function the ℓ_1 and ℓ_2 norms. For example, prior work on DL for accelerated MRI has considered trained the CNN by mapping the zero-filled images to the corresponding fully-sampled ground-truth images [73, 159, 189].

PnP [166, 186] is a widely-used framework that extend the traditional DL by enabling the integration of the physical measurement models and powerful DL denoisers as image priors to provide state-of-the-art reconstruction algorithms (see recent reviews of PnP in [4, 95]). As we have introduced in Section 2.3, the iterations of *regularization by denoising (RED)* [153],

which is a well-known PnP method, can be expressed as

$$\boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \left(\nabla g(\boldsymbol{x}^{k-1}) + \tau(\boldsymbol{x}^{k-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{k-1})) \right) , \tag{8.6}$$

where ∇g is the gradient of the data-fidelity term in (8.3), $\mathsf{D}_{\boldsymbol{\theta}}$ is the CNN denoiser parameterized by weights $\boldsymbol{\theta}$, and $\gamma, \tau > 0$ are the step size and the regularization parameters, respectively. The iterates of (8.6) seek an equilibrium between the physical measurement model and learned prior model. Remarkably, this heuristic of using CNNs not necessarily associated with any h within an iterative algorithm exhibited great empirical success [4, 166, 215, 217] and spurred a great deal of theoretical work on PnP [36, 158, 172, 199].

DU (also known as *deep unrolling* and *algorithm unrolling*) is another widely-used DL paradigm that was widely adopted in MRI due to its ability to provide a systematic connection between iterative algorithms and deep neural network architectures [1, 3, 75, 79, 134, 205, 207, 214]. PnP algorithms can be naturally turned into DU architectures by truncating the PnP algorithm to a fixed number of iterations and training the corresponding architecture end-to-end in a supervised fashion. By training the CNN $\mathsf{D}_{\boldsymbol{\theta}}$ jointly with the measurement model, DU leads to an image prior optimized for a given inverse problem.

In this work, we adopt the RED iteration (8.6) as the basis of our DU architecture. We will refer to this architecture as U-RED. The data-consistency layers of our U-RED architecture correspond to the gradient of the least-squares penalty (8.4)

$$\nabla g(\boldsymbol{x}) = \boldsymbol{A}^{\mathsf{H}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) , \tag{8.7}$$

where $\boldsymbol{A}^{\mathsf{H}}$ denotes the hermitian transpose of $\boldsymbol{A}$. We will introduce the details of our method in Section 8.3.

8.2.3 mGRE Sequences and Biophysical Model

The *multi-Gradient-Recalled Echo (mGRE)* sequences are used in different MRI applications to produce quantitative maps related to biological tissue microstructure in health and disease [77, 101, 150, 184, 192, 197, 218, 219]. Each reconstructed mGRE voxel can be interpreted using the following *biophysical model* [203]

$$x(t) = X_0 \cdot \exp(-R_2^* \cdot t - i\omega t) \cdot F(t), \tag{8.8a}$$

where t denotes the gradient echo time, $X_0 = x(0)$ is the signal intensity at $t = 0$, and ω is a local frequency of the MRI signal. The complex valued function $F(t)$ in (8.8) models the effect of macroscopic magnetic field inhomogeneities on the mGRE signal. The failure to account for such inhomogeneities is known to bias and corrupt the recovered R_2^* maps. The function $F(t)$ is traditionally computed using the *voxel spread function (VSF)* approach [204], based on evaluating the effects of macroscopic magnetic field inhomogeneities (background gradients) on formation of the complex-valued mGRE signal. The R_2^* maps, ω maps, and X_0 can be jointly estimated from 3D mGRE images acquired at different echo times t by fitting (8.8) with pre-calculated $F(t)$ on a voxel-by-voxel basis to experimental data using *non-linear least squares (NLLS).*

8.2.4 Deep qMRI Map Estimations

In practice, the traditional voxel-wise fitting methods such as NLLS are time consuming, and also sensitive to the artifacts (e.g. noise or motion artifacts) in MRI images. Recent work has shown the effectiveness and efficiency of using *deep neural networks (DNNs)* to estimate high-quality qMRI maps (see recent reviews in [59, 88, 110]). One straightforward

and effective application of DL in qMRI is to train a DNN to learn a direct mapping of qMRI maps from the MR images in a supervised fashion. The training can be guided by minimizing the loss between the outputs of the DNN and the qMRI map references estimated from the MR images using standard fitting methods. This end-to-end mapping strategy has been investigated in many qMRI applications, including T_2 [32], high quality susceptibility mapping (QSM) [22, 209], T_1 and $T_{1\rho}$ [108], $R2t^*$ and $R2'$ [89]. It has also been applied to help magnetic resonance fingerprinting (MRF) [143] with a better and more efficient generation of qMRI maps such as T_1 and T_2 [41, 57]. When it is challenging to obtain accurate and reliable qMRI map references, such end-to-end mapping can be further combined with biophysical models connecting MR images and qMRI maps to enable self-supervised learning where only MR images instead of the ground-truth qMRI maps are required for training [182, 201]. When measurement operator $\boldsymbol{A}$ is also available, it can be combined with biophysical models to enforce data consistency to the subsampled measurements in the DL pipeline [111, 213], leading to a model-based qMRI mapping. Other than these end-to-end qMRI mappings, some work also focused on developing DL-based image reconstruction methods to improve the qMRI estimation. The qMRI maps can be later computed from these reconstructions either using standard fitting method [66, 221] or DL-based mapping [194]. In analogy to these separated MRI reconstruction and qMRI map estimations, one can also combine the DL-based MRI reconstruction and DL-based qMRI estimation into one single pipeline and train it end-to-end [84].

DL-based qMRI estimation often leads to reliable qMRI mapping with better artifact-suppression capability and faster computation than standard fitting methods [32, 41, 84, 89, 111, 182, 201, 213]. In this work, we contribute to this growing deep qMRI estimation field by enabling the end-to-end estimation for R_2^* maps directly from k-space measurements, where the artifacts caused by subsampling, noise, motion and magnetic field inhomogeneities

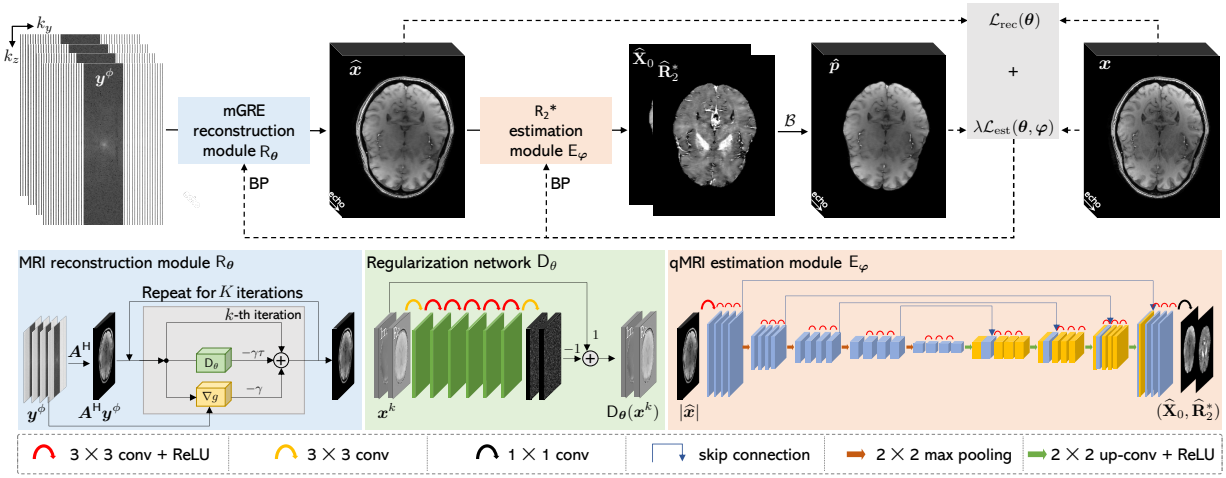

Figure 8.1: The overview of the proposed CoRRECT framework for training an end-to-end deep network consisting of two modules: $\mathsf{R}_{\boldsymbol{\theta}}$ for reconstructing mGRE MRI images and $\mathsf{E}_{\boldsymbol{\varphi}}$ for estimating corresponding R_2^* maps. The network takes input as subsampled, noisy, and motion-corrupted k-space measurements. $\mathsf{R}_{\boldsymbol{\theta}}$ is implemented as the unfolded U-RED architecture initialized using the zero-filled reconstruction. $\mathsf{E}_{\boldsymbol{\varphi}}$ is implemented as a customized U-Net architecture mapping the output of $\mathsf{R}_{\boldsymbol{\theta}}$ to the desired R_2^* map.The whole network is trained end-to-end using fully-sampled mGRE sequence data without any ground-truth quantitative (X_0, R_2^*) maps.

are considered and fixed together. We realize this idea by using a model-based MRI reconstruction module to first recover the mGRE images followed a biophysical-model-assisted R_2^* estimation module to compute motion- and $B0$-inhomogeneity-corrected R_2^* maps from those intermediately reconstructed mGRE images. By conducting joint training of these two modules, we achieve the accelerated, robust and accurate MRI reconstruction and R_2^* estimation simultaneously.

8.3 Proposed Method

The proposed CoRRECT framework seeks to jointly recover both mGRE images and R_2^* maps via end-to-end training. In this section, we present the technical details of CoRRECT.

8.3.1 Overall Architecture of CoRRECT

CoRRECT focuses on a motion-involved version of the imaging problem illustrated in Eq. (8.1). In particular, consider motion-affected subsampled measurements $\boldsymbol{y}^{\phi}$ obtained from the mGRE image $\boldsymbol{x}$ of N echo times as

$$\boldsymbol{y}^{\phi} = \boldsymbol{A}\phi(\boldsymbol{x}) + \boldsymbol{e}\ , \tag{8.9}$$

where the motion function ϕ represents the movements of the object during the scanning, which is assumed to be unknown. Our method aims to, without knowing ϕ, reconstruct both the motion-corrected mGRE image $\boldsymbol{x}$ and the R_2^* map by training a DNN on a set of ground-truth mGRE images $\{\boldsymbol{x}_j\}_{j=1}^J$ and their noisy subsampled measurements $\{\boldsymbol{y}_j^{\phi}\}_{j=1}^J$ given the measurement operator for each measurement as $\{\boldsymbol{A}\}_{j=1}^J$. Fig. 8.1 summarizes the details of the CoRRECT framework, where the sample index j of all variables are omitted for simplicity. We introduce the key idea of the CoRRECT as follows.

mGRE Reconstruction Module

Given the corrupted k-space measurements $\boldsymbol{y}_j$, a mGRE reconstruction module $\mathsf{R}_{\boldsymbol{\theta}}$ is first applied to reconstruct high quality mGRE images. We adopt a K-layer U-RED network as our mGRE reconstruction module $\mathsf{R}_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta} \in \mathbb{R}^p$ corresponds to the trainable parameters of the regularization network $\mathsf{D}_{\boldsymbol{\theta}}$. $\mathsf{R}_{\boldsymbol{\theta}}$ takes subsampled, noisy, and motion-corrupted k-space measurements $\boldsymbol{y}_j^{\phi}$ and the measurement operator $\boldsymbol{A}_j$ as its inputs, and produces artifact-corrected N-echo mGRE image $\widehat{\boldsymbol{x}}_j$ as its output, where

$$\widehat{\boldsymbol{x}}_j = \mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{y}_j^{\phi}; \boldsymbol{A}_j)\ . \tag{8.10}$$

As shown in Fig. 8.1, the input of U-RED is initialized with $\boldsymbol{A}_j^{\mathsf{H}}\boldsymbol{y}_j^{\phi}$. In our implementation, we set $K = 8$ and adopted a customized 7-layer DnCNN for the implementation of the regularization network $\mathsf{D}_{\boldsymbol{\theta}}$. The weights of $\mathsf{D}_{\boldsymbol{\theta}}$ are shared across all K steps for memory efficiency. To enable the reconstruction for complex mGRE data, the input of $\mathsf{D}_{\boldsymbol{\theta}}$ are split to 2 channels that consist of the real (denoted as Re) and imaginary (denoted as Im) parts.

R_2^* Estimation Module

As shown in Fig. 8.1, the mGRE reconstruction module $\mathsf{R}_{\boldsymbol{\theta}}$ is followed by a R_2^* estimation module $\mathsf{E}_{\boldsymbol{\varphi}}$ to allow for the estimation of R_2^*. Our R_2^* estimation module $\mathsf{E}_{\boldsymbol{\varphi}}$ is built on the self-learning network LEARN-BIO [201] discussed in Sec. 8.2.4, which consists of a CNN customized from U-Net [155] with trainable parameters $\boldsymbol{\varphi} \in \mathbb{R}^q$. As its order-sensitive input, $\mathsf{E}_{\boldsymbol{\varphi}}$ accepts the magnitude of the reconstructed N-echo mGRE image $\widehat{\boldsymbol{x}}_j$ from $\mathsf{R}_{\boldsymbol{\theta}}$ as its input and produces qMRI maps $(\widehat{\mathbf{X}}_0, \widehat{\mathbf{R}}_2^*)_j$ as its output, where

$$(\widehat{\mathbf{X}}_0, \widehat{\mathbf{R}}_2^*)_j = \mathsf{E}_{\boldsymbol{\varphi}}(|\widehat{\boldsymbol{x}}_j|) \, . \tag{8.11}$$

Here we use $|\cdot|$ to represents the magnitude extraction operator, and $\widehat{\mathbf{X}}_0 \in \mathbb{R}^n, \widehat{\mathbf{R}}_2^* \in \mathbb{R}^n$ to denote the vectorized X_0 and R_2^* outputs from the estimation module, respectively. Once trained, CoRRECT allows the joint reconstruction of mGRE images and estimation of R_2^* maps. We introduce the training strategy of CoRRECT in the following section.

8.3.2 Training of CoRRECT

We adopt a *self-supervised learning* strategy for the end-to-end training of CoRRECT, where only the mGRE data instead of the ground-truth (X_0, R_2^*) maps are required. To explain this training procedure, let's consider the intermediate mGRE output $\widehat{\boldsymbol{x}}_j$ from the reconstruction

module $\mathsf{R}_{\boldsymbol{\theta}}$ in Eq. (8.10) and the final quantitative map output $(\widehat{\mathbf{X}}_0, \widehat{\mathbf{R}}_2^*)_j$ from the estimation module $\mathsf{E}_{\boldsymbol{\varphi}}$ in Eq. (8.11). The end-to-end training of CoRRECT is enabled by the joint minimization of two distinct loss functions with respect to these two outputs: the mGRE reconstruction loss $\mathcal{L}_{\text{rec}}(\boldsymbol{\theta})$ and the R_2^* estimation loss $\mathcal{L}_{\text{est}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$. Given sampled data $\boldsymbol{x}_j$ and $\boldsymbol{y}_j^{\phi}$, the mGRE reconstruction loss $\mathcal{L}_{\text{rec}}(\boldsymbol{\theta})_j$ computes the difference between the reconstructed mGRE image $\widehat{\boldsymbol{x}}_j$ and the ground truth mGRE image $\boldsymbol{x}_j$ as

$$\mathcal{L}_{\text{rec}}(\boldsymbol{\theta})_j = \mathcal{L}(\widehat{\boldsymbol{x}}_j, \boldsymbol{x}_j) \ . \tag{8.12}$$

The R_2^* estimation loss $\mathcal{L}_{\text{est}}(\boldsymbol{\theta}, \boldsymbol{\varphi})_j$, on the other hand, enforces the data consistency of the mGRE images synthesized by the estimated (X_0, R_2^*) maps to the ground-truth mGRE images. It uses the analytical biophysical model $\hat{\boldsymbol{p}}_j = \mathcal{B}((\widehat{\mathbf{X}}_0, \widehat{\mathbf{R}}_2^*)_j; \boldsymbol{f}_j)$ in Eq. (8.8) to relate the mGRE images and the quantitative R_2^* maps into the loss function

$$\mathcal{L}_{\text{est}}(\boldsymbol{\theta}, \boldsymbol{\varphi})_j = \mathcal{L}(|\boldsymbol{M}\hat{\boldsymbol{p}}_j|, |\boldsymbol{M}\boldsymbol{x}_j|), \tag{8.13}$$

where $\boldsymbol{f}_j \in \mathbb{C}^n$ denotes the vectorized $F(t)$ function pre-computed using the VSF approach [204] from ground-truth mGRE data $\boldsymbol{x}_j$ to compensate for the effect of macroscopic magnetic field inhomogeneities, and $\boldsymbol{M}$ denotes the voxel-wise region extraction mask (REM) where the biophysical model applies. REM $\boldsymbol{M}$ is only needed during the training to assist the computation of R_2^* estimation loss and is not necessary during the test. Given losses $\mathcal{L}_{\text{rec}}(\boldsymbol{\theta})_j$ and $\mathcal{L}_{\text{est}}(\boldsymbol{\theta}, \boldsymbol{\varphi})_j$, the training of CoRRECT is conducted by minimizing their combination over a training set consisting of J samples as

$$\boldsymbol{\theta}^*, \boldsymbol{\varphi}^* = \underset{\boldsymbol{\theta}, \boldsymbol{\varphi}}{\arg\min} \sum_{j=1}^{J} \left\{ \mathcal{L}_{\text{rec}}(\boldsymbol{\theta})_j + \lambda \mathcal{L}_{\text{est}}(\boldsymbol{\theta}, \boldsymbol{\varphi})_j \right\}, \tag{8.14}$$

where $\lambda > 0$ is a weight parameter and the optimized parameters $\boldsymbol{\theta}^*$ and $\boldsymbol{\varphi}^*$ are obtained through gradient-based back propagation (BP).

The key feature of CoRRECT is that it is fully self-supervised, in the sense that it does not need ground-truth quantitative R_2^* maps for training. Instead, it is trained using only the mGRE images and our knowledge of the biophysical model $\mathcal{B}$ connecting the mGRE signal with R_2^* that includes the contribution of magnetic field inhomogeneities described by $F(t)$. By using $F(t)$ during training, our estimation module $\mathsf{E}_{\boldsymbol{\varphi}}$ learns to compensate for macroscopic magnetic field inhomogeneities to produce motion-artifact-free and $B0$-inhomogeneity-corrected R_2^* maps. Therefore, at testing time, the information of $F(t)$ functions are not required, resulting in a fast computation of the quantitative maps. The joint training of the mGRE reconstruction module $\mathsf{R}_{\boldsymbol{\theta}}$ and R_2^* estimation module $\mathsf{E}_{\boldsymbol{\varphi}}$ also benefits the performance of CoRRECT. On one hand, with the assistance of the reconstruction module on artifact correction, the estimation module greatly releases its pressure in artifact removal and therefore can focus on the R_2^* fitting. On the other hand, the reconstruction module is also guided by the performance of our R_2^* estimation module via the minimization of the loss $\mathcal{L}_{\text{est}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$. And by reconciling the reconstruction module and the estimation module into one end-to-end pipeline, they learn how to collaborate with each other through joint training, resulting in a potential maximization of the overall performance.

Table 8.1: Average SNR and SSIM values over the testing dataset corrupted with random levels of synthetic motion. The table highlights that CoRRECT outperforms several well-known baseline methods for different accelerated subsampling rates.

Images	mGRE						R_2^*					
Metric	SNR(dB)			SSIM			SNR(dB)			SSIM		
Acceleration rate	x2	x4	x8	x2	x4	x8	x2	x4	x8	x2	x4	x8
Corrupted	16.72	14.73	14.00	0.90	0.86	0.85	6.70	6.30	6.17	0.85	0.82	0.82
TV	21.46	19.88	17.05	0.81	0.8	0.77	12.21	11.72	10.60	0.92	0.90	0.87
RED	21.49	20.10	17.49	0.92	0.90	0.87	12.16	11.70	10.59	0.91	0.90	0.87
U-Net	20.79	19.25	18.09	0.92	0.90	0.88	12.08	11.39	10.77	0.91	0.89	0.88
U-RED	21.53	20.36	19.08	0.93	0.91	0.90	12.20	11.79	11.15	0.92	0.90	0.89
CoRRECT (Ours)	**22.12**	**20.66**	**19.25**	**0.93**	**0.91**	**0.90**	**12.99**	**12.33**	**11.60**	**0.92**	**0.90**	**0.89**

The performance of CoRRECT is marked **bold** for achieving the best performance in each column.

8.4 Experimental Validation

CoRRECT is trained to directly provide high-quality R_2^* maps from subsampled, noisy, and motion-corrupted k-space measurements. It is trained on simulated data with synthetic motion and validated on both simulated and experimentally-collected data. Our results in this section show that the method trained *only* on simulated motion-corrupted data can achieve excellent performance on previously-unseen experimental data corrupted with real motion.

8.4.1 Dataset Preparation

To validate our method, we selected fully-sampled clean k-space data of the brain as our source to generate the synthetic subsampled, noisy and motion-corrupted measurements. These brain data were collected from 15 healthy volunteers using a Siemens 3T Trio MRI scanner and a 32-channel phased-array head coil. Studies were conducted with the approval of the local IRB of Washington University. All volunteers provided informed consent. The data were obtained using a 3D version of the mGRE sequence with $N = 10$ gradient echoes followed

by a navigator echo [191] used to reduce artifacts induced by physiological fluctuations during the scan. Sequence parameters were flip angle $FA = 30°$, voxel size of $1 \times 1 \times 2$ mm^3, first echo time $t_1 = 4$ ms, echo spacing $\Delta t = 4$ ms (monopolar readout), repetition time TR = 50 ms. The dimension of raw measurement for each subject from each coil at a single echo time was $N^{k_x} \times N^{k_y} \times N^{k_z}$ with k_x and k_y both being the frequency-encoding dimension and k_z being read-out dimension, respectively. In our data, $N^{k_x} = 72$, $N^{k_y} = 192$, and $N^{k_z} = 256$. For the sake of GPU memory, we converted 3D k-space data into 2D k-space slices after a 1D Fourier Transform along the k_x dimension and apply our method to 3D MRI reconstruction and R_2^* estimation in a slice-by-slice manner.

These 15 subjects were split into 10, 2, and 3 for training, validation, and testing, respectively. For each subject, we extracted the middle 25 to 56 slices (72 in total) that contains the most relevant regions of the brain to use. This yields 3100 images for training, 620 for validation, and 930 for testing in the simulation. For each slice, 10-echo mGRE images of fully-sampled, noise- and motion-free k-space data was used as the ground truth, corresponds to the target image $\boldsymbol{x}$ in Eq. (8.9). We corrupted the ground-truth images $\boldsymbol{x}$ with the synthetic motion function ϕ as well as the forward operator $\boldsymbol{A}$ and Gaussian noise $\boldsymbol{e}$ in Eq. (8.9) to generate the artifact-corrupted measurements $\boldsymbol{y}^{\phi}$. The data $\{\boldsymbol{x}, \boldsymbol{y}^{\phi}\}$ of all samples were used to serve the training and quantitative evaluation of our method. We explain details of such data simulation and pre-processing in Sec. 8.4.2. Additional experimental data with clear visible motion artifacts were used for evaluating the performance of our network trained on synthetic data. The coil sensitivity maps for each slice were estimated from its $1st$ echo of fully sampled k-space data using ESPIRiT [183] for both simulated and experimental data.

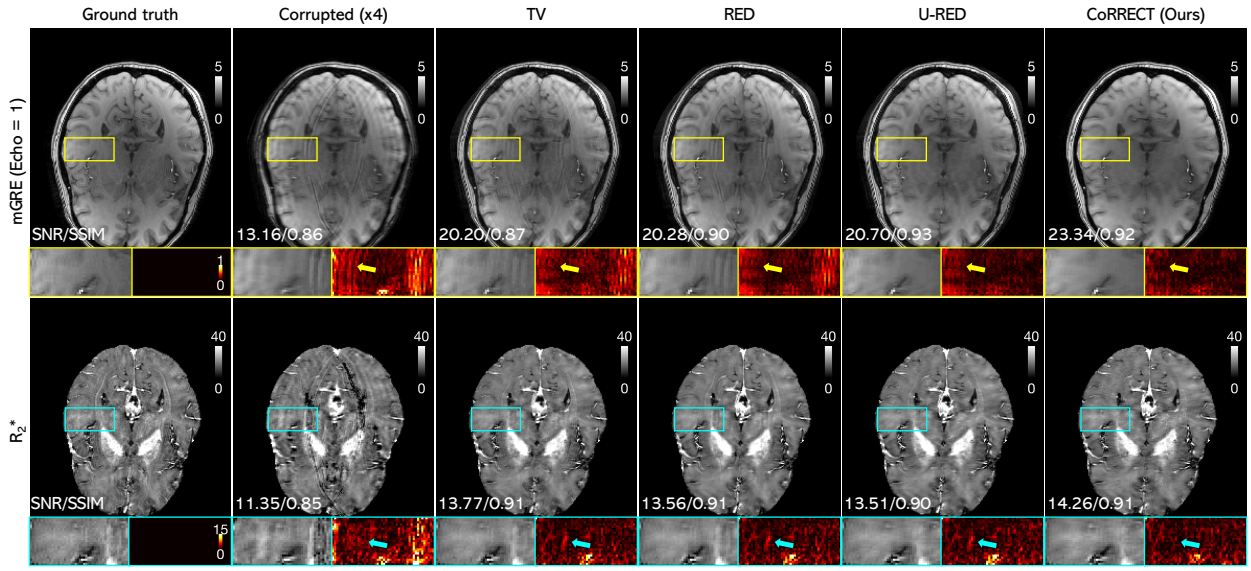

Figure 8.2: Performance of CoRRECT compared with different baseline methods on exemplar testing data corrupted with synthetic motion and subsampled with acceleration rate $\times 4$. The bottom-left corner of each image provides the SNR and SSIM values with respect to the ground-truth. Arrows in the zoomed-in plots highlight brain regions that are well reconstructed using CoRRECT. Note that the R_2^* estimation of *TV*, *RED*, *U-RED* are conducted by the motion-correction-enabled network LEARN-BIO for fixing the motion corruptions left in their reconstruction. This figure highlights that CoRRECT can achieve excellent quantitative and visual performance in both mGRE reconstruction and R_2^* estimation.

8.4.2 Data Simulation and Pre-processing

As aforementioned, the training and quantitative evaluation of our network requires the paired clean mGRE images $\boldsymbol{x}$ as its ground truth and corrupted k-space measurements $\boldsymbol{y}^{\phi}$ as its inputs. We obtained $\boldsymbol{x}$ from fully-sampled clean k-space measurements via inverse Fourier transformation, and synthesized the subsampled, noisy and motion-corrupted measurements $\boldsymbol{y}^{\phi}$ by corrupting the fully-sampled clean ones. We introduce the procedure of the simulation here.

Motion Simulation

We hypothesize the motion artifacts in the MR images are the consequence of a series of physical motions such as shifts or rotations that result in perturbations of blocks of k-space lines during corresponding motions. We therefore replace certain k-space lines of the ground-truth MR images $\boldsymbol{x}$ with those of their moved versions to synthesize motion artifacts. To generate a range of realistic and various motion artifacts for our simulated data, we set the number of motion movements, the duration of each movement, and the amplitude of each movements all as random numbers following the configuration used in [201].

Subsampling and Noise Corruption

The motion-affected k-space data were further corrupted by subsampling. In this work, we adopt a Cartesian sampling pattern that fully-samples along k_x and k_z dimension, and subsamples along the k_y dimension in the k-space. We experimented with three sampling rates 50%, 25% and 12.5%, which are referred to as acceleration rate $\times 2$, $\times 4$ and $\times 8$ respectively in the following context for simplicity. For each rate, we kept the central 60 out of 192 lines along k_y fully-sampled. The simulation of corrupted measurements $\boldsymbol{y}^{\phi}$ was finalized by adding AWGN corresponding to an input SNR of 40dB to the motion-corrupted and subsampled k-space data.

8.4.3 Experiments Setup

CoRRECT solves the mGRE reconstruction problem and R_2^* estimation problem simultaneously through joint training. To demonstrate the performance of CoRRECT on both problems and highlight the benefits of joint training, we compare our method against several approaches that decouple the image reconstruction and the R_2^* estimation.

Baseline Methods for mGRE Reconstruction

For the image reconstruction problem, we included *TV* [156], *U-Net* [155] and traditional *RED* [153]. We also included deep *U-RED* explained in Sec. 8.2.2 to illustrate the improvements due to joint training. TV is an iterative method that does not require training, while other methods are all DL-based with publicly available implementations. We trained our DL-based baseline methods on motion-free data to handle mGRE reconstruction. We used the same DnCNN [155] architecture used in our image reconstruction module as the AWGN denoiser for RED and trained those denoisers for AWGN removal at four noise levels corresponding to noise variance $\sigma \in \{1, 3, 5, 7\}$. For each experiment, we selected the denoiser achieving the highest SNR. U-RED shares the same setting as our image reconstruction module, except that it is not jointly trained with an attached R_2^* estimation module. We ran TV and RED both for 50 iterations. We fixed the step size $\gamma = 0.5$ for TV, RED, U-RED and CoRRECT. We used grid search to identify the optimal regularization parameters τ for TV, RED and learned its value through training for U-RED and CoRRECT.

Baseline Methods for R_2^* Estimation

We applied the DL-based R_2^* estimation method LEAR-BIO [201] to the reconstructed mGRE images from baseline methods to compute the corresponding R_2^* maps as comparisons to the ones from our end-to-end training. LEARN-BIO shares the same network structure as our R_2^* estimation module $\mathsf{E}_{\boldsymbol{\varphi}}$, except that it was not jointly trained with the mGRE reconstruction. In particular, we trained two LEARN-BIO networks, namely LEARN-BIO (clean) and LEARN-BIO (motion), to compute high-quality R_2^* maps. LEARN-BIO (clean) was trained on artifact-free mGRE images. We applied this network to ground-truth mGRE images to get ground-truth R_2^* references for quantitative evaluation. LEARN-BIO (motion) was trained on motion-corrupted mGRE images (generated with the same motion simulation

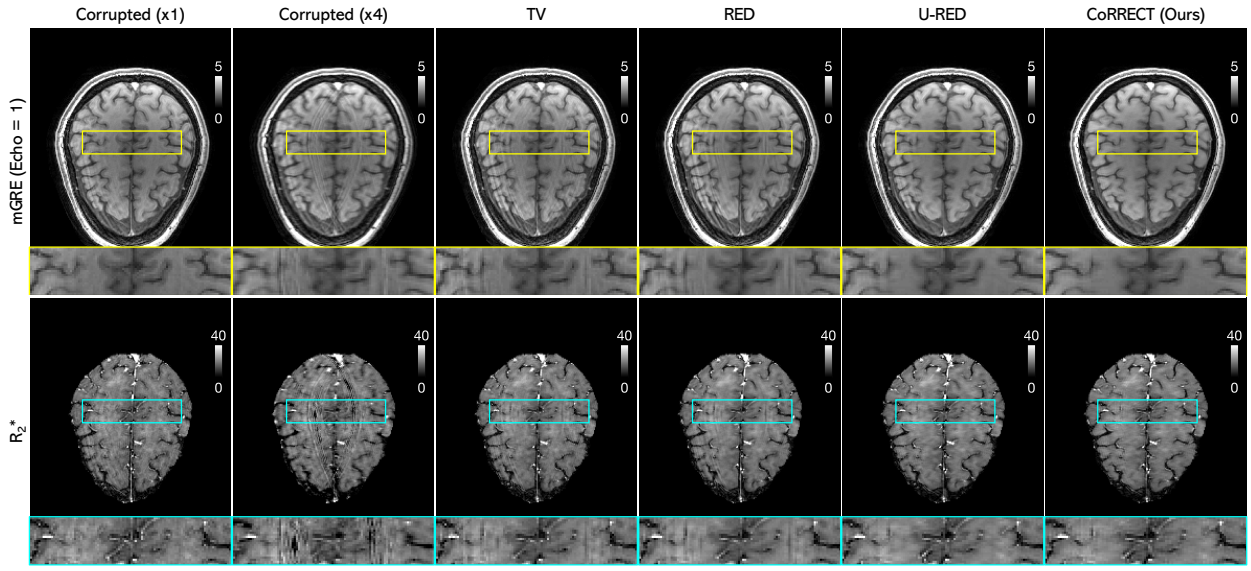

Figure 8.3: Performance of CoRRECT compared with different baseline methods on exemplar testing data corrupted with real motion and subsampled with acceleration rate $\times 4$. The mGRE image in the first column, denoted $\times 1$, is from the motion-corrupted but fully-sampled k-space data, while the one in the second column, denoted $\times 4$, is from the motion-corrupted and also subsampled k-space data. Note the excellent performance of our method in removing the comprehensive artifacts that remain in the results of all baseline methods. This demonstrates the capability of our network trained with synthetic motion in dealing with real motion artifacts.

configuration introduced in Sec. 8.4.2) to compute motion-corrected R_2^* maps. We applied this network to all reconstruction baseline methods to capture the motion residue in their mGRE reconstruction for high-quality R_2^* estimation. In addition, for the subsampled, noisy and motion-corrupted mGRE images reconstructed with zero-filling, we applied the traditional voxel-wise NLLS approach for their R_2^* estimation. As described in Sec. 8.2.3, NLLS is a standard iterative fitting method for computing R_2^* based on Eq. (8.8), where in each iteration, the regression is conducted by combining the data from different echo times t with their $F(t)$ values voxel by voxel. Since NLLS is a pure fitting method without artifact-fixing capability, this clearly shows how artifacts in mGRE images collapses the estimation of R_2^* maps. Prior to the NLLS fitting procedures, a brain extraction tool, implemented in the Functional Magnetic Resonance Imaging of the Brain Library(FMRIB), was used to generate

the REMs to mask out both skull and background voxels in all mGRE data [85], where the signal model defined in Eq. (8.8) doesn't apply. NLLS was run over only the set of unmasked voxels. Similarly, we applied the same REMs in the loss function Eq. (8.13) of our R_2^* estimation module as well as baseline method LEARN-BIO during their training. All the results of R_2^* presented in this paper were also processed by these masks before evaluation and visualization.

Implementation Details and Evaluation Metrics

Based on our empirical observation, we adopted the ℓ_2 loss for both loss functions $\mathcal{L}_{\text{rec}}(\boldsymbol{\theta})$ and $\mathcal{L}_{\text{est}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ and set the weighting parameter $\lambda = 1$. We set the learning rates of our network as 1×10^{-5}. We performed all our experiments on a machine equipped with 8 GeForce RTX 2080 GPUs. For quantitative evaluation, we adopted two widely-used quantitative metrics, *signal-to-noise ratio (SNR)*, measured in dB and *structural similarity index (SSIM)*, relative to the ground-truth. In experimental scenarios where ground-truth is not available, we applied our networks trained on synthetic data to experimental data and provided qualitative visual comparisons of different approaches.

8.4.4 Results on Simulated Data

We first test the performance of CoRRECT on simulated data with synthetic motion corruptions. We followed the configuration in Sec. 8.4 to add random motion to each data slice in our testing dataset to cover comprehensive motion levels. Table 8.1 summarizes quantitative results of all evaluated methods at different acceleration rates. As highlighted in Table 8.1, CoRRECT achieves the highest SNR and SSIM values compared to other methods over all considered configurations.

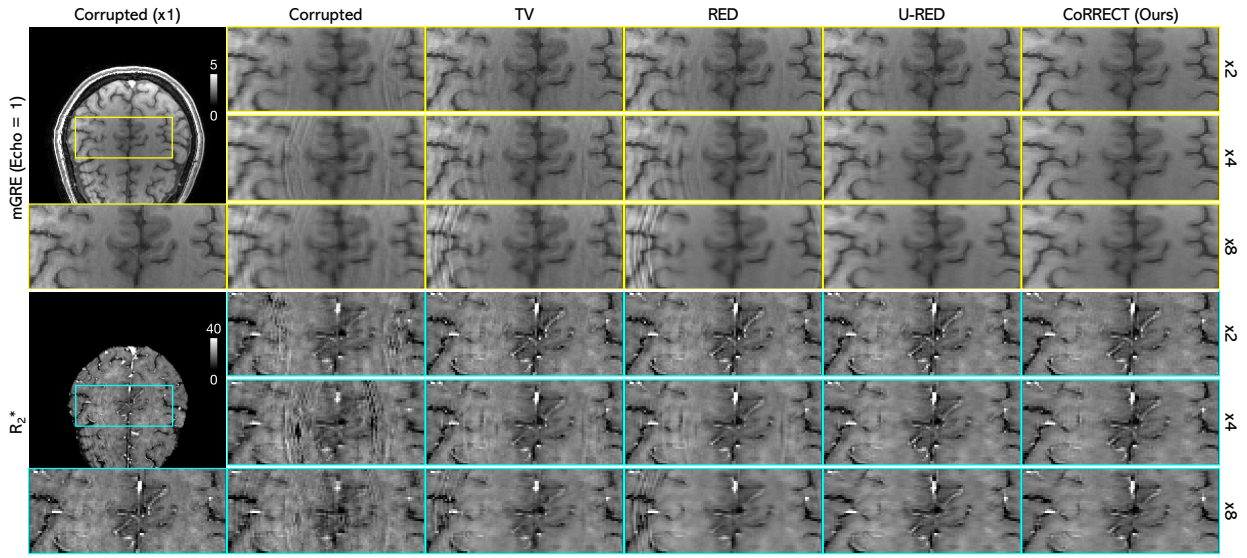

Figure 8.4: Performance of CoRRECT compared against different baseline methods on exemplar testing data corrupted with real motion and subsampled with acceleration rates $\{\times 2, \times 4, \times 8\}$. The mGRE image in the first column, denoted with $\times 1$, is from the motion-corrupted but fully-sampled k-space data, while the ones in the second column are from the motion-corrupted and also subsampled k-space data. Note the excellent performance of our method is demonstrated by its ability to remove comprehensive artifacts while maintaining structure details across different acceleration rates.

Fig. 8.2 visualizes the performance of CoRRECT compared with different baseline methods on exemplar simulated data. The $1st$ echo of a complex-valued mGRE image sequence is visualized as its normalized magnitude, where the normalization is done by dividing by the mean of the intensity in the $1st$ echo of the mGRE sequence. The corrupted image shows that subsampling and motion can severely degrade the quality of mGRE images by causing a significant amount of blurring and aliasing artifacts, and consequently collapses R_2^* estimation. Baseline methods TV and RED alleviate some of the artifacts in the corrupted image. However, due to their inability to capture the motion effects missed in the forward operator, a considerable amount of artifacts are still observed in mGRE reconstruction. Meanwhile, due to the existence of unknown motion, the forward operator $\boldsymbol{A}$ that only models the subsampling is no longer accurate. As a result, the reconstruction can get misled, even resulting in a degradation of artifacts (see the enhanced artifacts around the central region

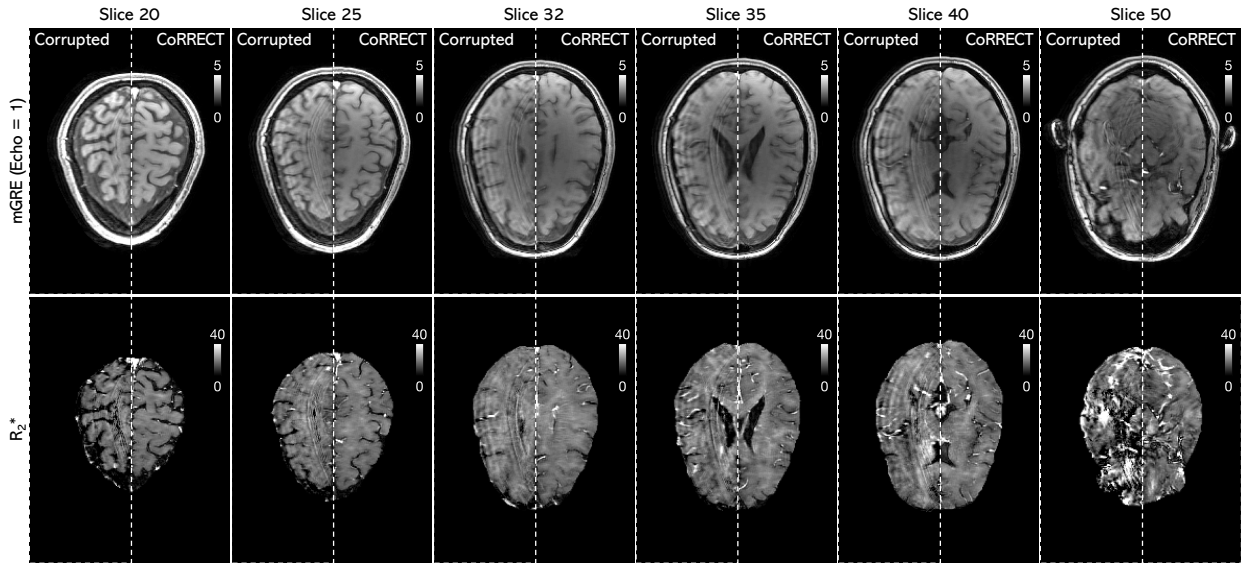

Figure 8.5: The performance of CoRRECT on experimental data corrupted with real motion and subsampled with acceleration rate $\times 4$. The first row shows the mGRE images across different slices in a whole brain volume of 72 slices, and the second row shows the corresponding R_2^* maps estimated from these mGRE images. For a given slice in each column of the first row, the image to the left of the dashed line is the mGRE image reconstructed by zero-filling from subsampled, noisy and motion-corrupted k-space data, and the image to the right is reconstructed by CoRRECT. In each column of the second row, the R_2^* to the left of the dashed line is estimated by applying NLLS to the corrupted mGRE image above it, and the right is produced by our method. This demonstrates the of capability of CoRRECT in removing artifacts for the whole brain volume.

of the brain). U-RED can further reduce the overall artifacts by using a CNN-embedded deep network to compensate for artifacts through end-to-end training, but is still suboptimal, showing visible artifacts in mGRE reconstruction. As for R_2^* estimation, although a significant improvement over the NLLS fitting is observed by using motion-correction-enabled LEARN-BIO on artifact-contaminated mGRE images from those baseline methods, the estimation still suffers from inaccuracy in the regions indicated by blue arrows. Our proposed method, CoRRECT, managed to achieve the best performance compared to all evaluated baseline methods in terms of sharpness, contrast, artifact removal and accuracy, thanks to joint training of mGRE reconstruction and R_2^* estimation.

8.4.5 Results on Experimental Data

We further validate the performance of our network trained on simulated data using experimental data with real motion corruptions.

Figure 8.3 visualizes the performance of CoRRECT compared with different baseline methods on exemplar experimental data corrupted with real motion and subsampled with acceleration rate $\times 4$. Note that the corrupted mGRE image in the first column, denoted with acceleration rate $\times 1$, corresponds to the corrupted mGRE image of motion-affected but fully-sampled k-space data. The corresponding R_2^*, which is estimated using LEARN-BIO (clean), consequently suffers from these motion corruptions as well. While such motion artifacts in this experimental data might not follow our simulation model, we do observe similar results to our synthetic experiments. It can be seen that CoRRECT outperforms the evaluated baseline methods in both mGRE reconstruction and R_2^* estimation in terms of removing artifacts and maintaining sharpness. This shows CoRRECT is capable of handling real motion artifacts while still keeping detailed structural information. Figure 8.4 shows comprehensive results across different acceleration rates for the same data sample, where consistently outstanding and robust performance of CoRRECT is observed.

Fig. 8.5 further demonstrates the performance of our method across different data slices in a whole brain volume, where each slice, in principle, is corrupted with different and random motions during the scan. For each slice, we show the side-to-side comparison between the results of CoRRECT and the corrupted images, including the zero-filled mGRE images reconstructed from subsampled, noisy and motion-corrupted k-space data and their NLLS-estimated R_2^* maps. The constant success of CoRRECT on different brain slices proves that our network can work on the whole spectrum of brain volume, highlighting the effectiveness and adaptability of our method.

8.5 Discussion and Conclusion

In this manuscript, we proposed CoRRECT, a codesign of MRI reconstruction and R_2^* estimation with correction for unknown motion. Our method realizes MR image reconstruction and end-to-end R_2^* estimation simultaneously and directly from subsampled, noisy and motion-corrupted k-space data. It is realized by integrating a MRI reconstruction module that produces clean mGRE images and a qMRI estimation module that computes R_2^* maps together. Both modules are DL-based to allow for joint end-to-end training, where the MRI reconstruction module adopts the popular deep U-RED framework, and the qMRI estimation module adopts the powerful U-Net structure. Our network is trained in a self-supervised fashion, where ground truth mGRE images instead of the quantitative R_2^* maps are used. Such learning is enabled by embedding the biophysical model that connects the mGRE images and R_2^* maps into the loss function. We train our network on simulated data, and validated it on both simulated data with synthetic motion and experimental data with real motion. Our results show that CoRRECT achieves the best performance in different scenarios compared to other popular methods, showing its effectiveness and potential in practical applications.

Despite the excellent performance of our method obtained using our current design, both the reconstruction module and estimation module in CoRRECT are compatible with other potential architectures. Therefore, any improvements on MRI reconstruction and qMRI estimation, in principle, can also be adopted to further improve our performance. Also, although in this work we focused on R_2^* estimation, by changing biophysical models used, our method can be modified for the prediction of many other qMRI maps.

Part VI

Conclusion

Chapter 9

Conclusion

In this dissertation, we introduced extensions, analysis, and applications of three popular image reconstruction frameworks: plug-and-play priors (PnP), regularized by denoising (RED), and deep unfolding (DU). In the first section below, we compare the major features of these three frameworks. In the second section, we summarize the results and contributions of our work. In the last section, we discuss the potential areas of interest for future research related to our work.

9.1 Summary of PnP, RED and DU

As popular computational imaging algorithms for image reconstruction, PnP, RED and DU integrate the imaging model and deep learning to achieve both data consistency guarantees and advanced prior representation. PnP and RED realize the integration by using learning-based denoisers inside model-based optimization, while DU by including the imaging models inside end-to-end deep learning. These three methods are closely related to but also distinct

from each other in many ways. Here we summarize their key features of them to highlight their similarities and differences.

PnP. PnP [166, 186] is flexible methodology that embed image denoisers D_σ with denoising strength σ as priors inside model-based optimization algorithms for image reconstruction. PnP algorithms alternate between imposing data consistency by minimizing a data-fidelity term and imposing a statistical prior by applying an AWGN denoiser. The use of advanced learning-based denoisers specified through pre-trained deep neural nets enhanced PnP with the power of deep learning. The key advantage of PnP is that it can impose statistical priors without explicitly forming an objective function, which on the other hand, also introduces challenges for theoretical analysis of PnP. Nevertheless, as presented in Chapter 5, the iterates of PnP can be related to the minimization of some objective function for certain type of denoisers such as MMSE denoisers.

RED. RED [153] is a closely related approach to PnP that also enables integration of denoisers D_σ as priors for inverse problems. The key difference between RED and PnP is that RED was initially derived as an optimization problem where it can lead to an explicit denoiser-embedded regularization function $r(\boldsymbol{x}) = (\tau/2)\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{x}-\mathsf{D}_\sigma(\boldsymbol{x}))$, under certain conditions on D_σ: 1) local homogeneity and 2) symmetric Jacobian. However, due to the impracticalness of such constrains, the use of RED usually abandons the explicit regularization function and relies on the operator $\tau(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))$ instead. Subsequent analysis showed that the gradient-based RED variants can be interpreted as an interpretable fixed-point iterations [149], which is more appropriate for practical denoisers. Another attractive advantage of RED is that it can adjust the strength between data-fidelity and the prior imposed by a denoiser through the parameter τ. The missing of such a weighting parameter in PnP was addressed in Chapter 4 with our denoiser scaling technique.

DU. DU [1, 3, 75, 79, 134, 205, 214] is a related approach to both PnP and RED that interprets the iterations of an image recovery algorithm as layers of a neural network and trains it end-to-end in a supervised fashion. In each layer, it usually contains a un-trainable data-consistency module and a deep learning module. Many PnP/RED algorithms have been turned into DU architectures by truncating the algorithm to a fixed number of iterations. But unlike in PnP/RED that specify the prior with AWGN denoisers D_σ, the CNN in DU is trained jointly with the measurement model, leading to an image prior $\mathsf{D}_{\boldsymbol{\theta}}$ optimized for a given inverse problem via the trainable parameter $\boldsymbol{\theta}$. Such task-specific priors obtained trough end-to-end training usually lead DU to outperform PnP and RED where the denoiser is trained separately from the task. Besides the use in the context of DL, such task-dependent prior $\mathsf{D}_{\boldsymbol{\theta}}$ pre-trained using DU can also be plugged into PnP or RED algorithms as an artifact-removal (AR) operator. Equipped with the power of artifact removal, the sophisticated AR operators yield significantly improved results relative to an AWGN denoiser [114]. Other than advanced image priors, the end-to-end training also benefits the application of DU in a sense that it allows DU to be embedded into even larger frameworks for complex imaging applications. Our work in Chapter 8 presented such an example where DU was used as an image reconstruction submodule inside an end-to-end learning framework for the accelerated estimation of quantitative MRI maps.

9.2 Summary of Our Work

In this dissertation, we have presented our work on computational imaging algorithms based on the PnP, RED, and DU frameworks. We summarize the key results and contribution of our work as follows.

The statistical interpretation of PnP. In Chapter 4, we presented a denoiser scaling technique for improving the performance of PnP algorithms. We theoretical justified the denoiser scaling from the perspectives of proximal optimization, statistical estimation, and consensus equilibrium. In Chapter 5, we established the first theoretical convergence result for PnP-PGM algorithm with MMSE denoisers. We showed that the iterates produced by PnP-PGM with an MMSE denoiser converge to a stationary point of some global cost function. These two chapters together provided several new insights into the PnP methodology by giving statistical interpretations for PnP through the analysis of the denoisers.

An incremental PnP-ADMM algorithm. In Chapter 6, we addressed the limitation of PnP algorithms in dealing with a large number measurements. Our proposed incremental PnP-ADMM algorithm can effectively scale to datasets that are too large for traditional batch processing by using a single element or a small subset of the dataset at a time. We theoretically analyzed the convergence of the algorithm under a set of explicit assumptions, extending recent theoretical results in the area. This work successfully adapted PnP to large-scale imaging problems and validated its effectiveness with stochastic data processing.

Bregman PnP algorithms. In Chapter 7, we generalized the PnP/RED algorithms to the Bregman-PnP/Bregman-RED algorithms based on the more general Bregman distance beyond the classical Euclidean distance . We presented a theoretical convergence result for PnP-BPGM and demonstrated its effectiveness on Poisson linear inverse problems using DU. This work bypassed the Lipschitz gradient assumption on the data-fidelity term and broadened the family of PnP algorithms to the non-Euclidean setting.

A novel DU framework for quantitative MRI. In Chapter 8, we presented CoRRECT as a new framework for recovering quantitative R_2^* maps from subsampled and artifact-corrupted MRI data using DU. Our method jointly addressed three problems: image reconstruction,

motion correction and quantitative map estimation. Our results on simulated and experimentally collected multi-Gradient-Recalled Echo (mGRE) MRI data showed the potential of CoRRECT to enable high-qualityR_2^* mapping in highly accelerated acquisition settings. This work opened the door to DU methods that can integrate information from physical measurement models, biophysical signal models, and learned prior models for high-quality quantitative MRI.

9.3 Outlook

Based on our work presented in this dissertation, we discuss the potential future directions for computational imaging algorithms as follows.

Understanding the solutions of PnP/RED. The current theoretical analysis for PnP and RED has established many results for the solution of PnP/RED algorithms in terms of fixed points. However, the interpretation to such fixed-point solutions are still open questions. Meanwhile, there are also questions about how to relate different denoisers to optimization problem and which denoisers provide guaranteed convergence.

Learning more sophisticated priors for imaging.: The use of AWGN denoisers especially the learning-based ones in PnP/RED opened a new door to the exploration of advanced image priors that unnecessarily represented by a regularizer function. AR operators further boost the performance of PnP/RED by adopting task-optimized priors [114]. A recent line of work has also investigated the priors specified by generative adversarial networks (GANs) [23, 82, 83, 106, 145, 162]. It is a interesting question to ask how to develop even better priors that can assist the image reconstruction and how to establish convergence analysis for these advanced denoisers.

Enabling the learning without ground truth. Most of current model-based learning frameworks require the ground truth for training. It is appealing to investigate the unsupervised, self-supervised and half-supervised learning where the access to full ground truth data is relaxed. Recent work on neural fields representation provides an option for representing and rendering 3D scenes using coordinate-based deep neural networks without access to the full view of the scenes [132, 164, 165]. Such internal learning scheme have been adopted to enables the interpolation for measurements where the ground-truth data is not required [171]. Future end-to-end learning framework that combines the internal learning with advanced priors worth discovering.

Training memory-efficient DU networks. The success of DU highly benefits from the end-to-end training of imaging model and prior integrated networks. DU architectures, however, are usually limited to a small number of unfolded iterations due to the high computational and memory complexity of training. Recent work has addressed this issue with the deep equilibrium model (DEQ) [9, 68, 113], which is a method for training infinite-depth networks by analytically backpropagating through the fixed points using implicit differentiation. Both theoretical and practical extensions of such DEQ-assisted DU frameworks can be discussed to enable the efficient training of DU networks for large-scale imaging problems.

References

[1] J. Adler and O. Öktem. "Learned Primal-Dual Reconstruction". In: *IEEE Trans. Med. Imag.* 37.6 (June 2018), pp. 1322–1332.

[2] M. V. Afonso, J. M.Bioucas-Dias, and M. A. T. Figueiredo. "Fast Image Recovery Using Variable Splitting and Constrained Optimization". In: *IEEE Trans. Image Process.* 19.9 (Sept. 2010), pp. 2345–2356.

[3] H. K. Aggarwal, M. P. Mani, and M. Jacob. "MoDL: Model-based Deep Learning Architecture for Inverse Problems". In: *IEEE Trans. Med. Imag.* 38.2 (Feb. 2019), pp. 394–405. DOI: 10.1109/TMI.2018.2865356.

[4] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter. "Plug-and-Play Methods for Magnetic Resonance Imaging: Using Denoisers for Image Recovery". In: *IEEE Signal Process. Mag.* 37.1 (2020), pp. 105–116.

[5] Abdullah H Al-Shabili, Hassan Mansour, and Petros T Boufounos. "Learning Plug-and-Play Proximal Quasi-Newton Denoisers". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 8896–8900.

[6] Abdullah H. Al-Shabili, Xiaojian Xu, Ivan Selesnick, and Ulugbek S. Kamilov. *Bregman Plug-and-Play Priors*. Feb. 2022. DOI: 10.48550/arXiv.2202.02388. arXiv: 2202.02388 [cs, eess].

[7] M. S. C. Almeida and M. A. T. Figueiredo. "Deconvolving Images with Unknown Boundaries Using the Alternating Direction Method of Multipliers". In: *IEEE Trans. Ima* 22.8 (Aug. 2013), pp. 3074–3086.

[8] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. "Multi-Task Feature Learning". In: *Advances in Neural Information Processing Systems*. Vol. 19. Vancouver, British Columbia, Canada, Dec. 2006, pp. 41–48.

[9] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. "Deep Equilibrium Models". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[10] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. "Clustering with Bregman Divergences". In: *J. Mach. Learn. Res.* 6.Oct (2005), pp. 1705–1749.

[11] H. H. Bauschke, J. Bolte, and M. Teboulle. "A Descent Lemma beyond Lipschitz Gradient Continuity: First-Order Methods Revisited and Applications". In: *Math. Oper. Res.* 42.2 (2017), pp. 330–348.

[12] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Second. New York, NY, USA: Springer, 2017.

[13] H. H. Bauschke, R. Goebel, Y. Lucet, and X. Wang. "The Proximal Average: Basic Theory". In: *SIAM J. Optim.* 19.2 (2008), pp. 766–785.

[14] Heinz H Bauschke, Jonathan M Borwein, et al. "Legendre Functions and the Method of Random Bregman Projections". In: *J. Convex Anal.* 4.1 (1997), pp. 27–67.

[15] A. Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. SIAM, 2017.

[16] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM J. Imaging Sci.* 2.1 (2009), pp. 183–202.

[17] A. Beck and M. Teboulle. "Convex Optimization in Signal Processing and Communications". In: Cambridge, 2009. Chap. Gradient-Based Algorithms with Applications to Signal Recovery Problems, pp. 42–88.

[18] A. Beck and M. Teboulle. "Fast Gradient-Based Algorithm for Constrained Total Variation Image Denoising and Deblurring Problems". In: *IEEE Trans. Image Process.* 18.11 (Nov. 2009), pp. 2419–2434.

[19] J. Bect, L. Blanc-Feraud, G. Aubert, and A. Chambolle. "A $_1$-Unified Variational Framework for Image Restoration". In: *Proc. Euro. Conf. Comp. Vis. (ECCV)*. Vol. 3024. New York, 2004, pp. 1–13.

[20] D. P. Bertsekas. "Incremental Proximal Methods for Large Scale Convex Optimization". In: *Math. Program. Ser. B* 129 (2011), pp. 163–195.

[21] S. Biswas, H. K. Aggarwal, and M. Jacob. "Dynamic MRI Using Model-based Deep Learning and SToRM Priors: MoDL-SToRM". In: *Magn. Reson. Med.* 82.1 (July 2019), pp. 485–494.

[22] Steffen Bollmann et al. "DeepQSM - Using Deep Learning to Solve the Dipole Inversion for Quantitative Susceptibility Mapping". In: *NeuroImage* 195 (July 2019), pp. 373–383. DOI: 10.1016/j.neuroimage.2019.03.060.

[23] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. "Compressed Sensing Using Generative Models". In: *Proc. 34th Int. Conf. Machine Learning (ICML)*. Vol. 70. International Convention Centre, Sydney, Australia, Aug. 2017, pp. 537–546.

[24] A. Borji and L. Itti. "CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research". In: *Proc IEEE Conf Comput Vis Pattern Recognit Workshop Future Datasets* (June 2015).

[25] L. Bottou. “Neural Networks: Tricks of the Trade”. In: *Neural Networks: Tricks of the Trade*. Second. Springer, Sept. 2012. Chap. Stochastic Gradient Descent Tricks, pp. 421–437.

[26] L. Bottou, F. E. Curtis, and J. Nocedal. “Optimization Methods for Large-Scale Machine Learning”. In: *SIAM Rev.* 60.2 (2018), pp. 223–311.

[27] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2004.

[28] S. Boyd and L. Vandenberghe. *Subgradients*. Stanford University, Stanford, CA, USA, Apr. 2008.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. In: *Found. Trends Mach. Learn.* 3.1 (July 2011), pp. 1–122.

[30] Lev M Bregman. “The Relaxation Method of Finding the Common Point of Convex Sets and Its Application to the Solution of Problems in Convex Programming”. In: *USSR Comput. Math. Math. Phys.* 7.3 (1967), pp. 200–217.

[31] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman. “Plug-and-Play Unplugged: Optimization Free Reconstruction Using Consensus Equilibrium”. In: *SIAM J. Imaging Sci.* 11.3 (Sept. 2018), pp. 2001–2020.

[32] Congbo Cai et al. “Single-Shot T2 Mapping Using Overlapping-Echo Detachment Planar Imaging and a Deep Convolutional Neural Network”. In: *Magn Reson Med* 80.5 (Nov. 2018), pp. 2202–2214. DOI: 10.1002/mrm.27205.

[33] E. J. Candès, J. Romberg, and T. Tao. “Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information”. In: *IEEE Trans. Inf. Theory* 52.2 (Feb. 2006), pp. 489–509.

[34] A. Chambolle. “An Algorithm for Total Variation Minimization and Applications”. In: *J Math Imag Vis* 20.1 (2004), pp. 89–97.

[35] S. H. Chan, X. Wang, and O. A. Elgendy. “Plug-and-Play ADMM for Image Restoration: Fixed-point Convergence and Applications”. In: *IEEE Trans. Comp. Imag.* 3.1 (Mar. 2017), pp. 84–98.

[36] Stanley H Chan, Xiran Wang, and Omar A Elgendy. “Plug-and-Play ADMM for Image Restoration: Fixed-point Convergence and Applications”. In: *IEEE Trans. Comput. Imag.* 3.1 (2016), pp. 84–98.

[37] Tony F. Chan, Jianhong Shen, and Hao-Min Zhou. “Total Variation Wavelet Inpainting”. In: *J Math Imaging Vis* 25.1 (July 2006), pp. 107–125. DOI: 10.1007/s10851-006-5257-3.

[38] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert. “Self-Supervised Learning for Medical Image Analysis Using Image Context Restoration”. In: *Medical Image Analysis* 58 (Dec. 2019), p. 101539. DOI: 10.1016/j.media.2019.101539.

[39] Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. "Theoretical Linear Convergence of Unfolded ISTA and Its Practical Weights and Thresholds". In: *ArXiv Prepr. ArXiv180810038* (2018). arXiv: 1808.10038.

[40] Y. Chen and T. Pock. "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.6 (June 2017), pp. 1256–1272.

[41] Ouri Cohen, Bo Zhu, and Matthew S. Rosen. "MR Fingerprinting Deep RecOnstruction NEtwork (DRONE)". In: *Magn Reson Med* 80.3 (Sept. 2018), pp. 885–894. DOI: 10.1002/mrm.27198.

[42] P. L. Combettes and J.-C. Pesquet. "Proximal Thresholding Algorithm for Minimization over Orthonormal Bases". In: *SIAM J. Optim.* 18.4 (2007), pp. 1351–1376.

[43] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. "Color Image Denoising via Sparse 3D Collaborative Filtering with Grouping Constraint in Luminance-Chrominance Space". In: *Proc. IEEE Int. Conf. Image Proc. (ICIP 2017)*. San Antonio, TX, USA, 2007.

[44] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering". In: *IEEE Trans. Image Process.* 16.16 (Aug. 2007), pp. 2080–2095.

[45] A. Danielyan, V. Katkovnik, and K. Egiazarian. "BM3D Frames and Variational Image Deblurring". In: *IEEE Trans. Image Process.* 21.4 (Apr. 2012), pp. 1715–1728.

[46] I. Daubechies, M. Defrise, and C. De Mol. "An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint". In: *Commun. Pure Appl. Math.* 57.11 (Nov. 2004), pp. 1413–1457.

[47] A. Dempster, N. M. Laird, and D. B. Rubin. "Maximum-Likelihood from Incomplete Data via the EM Algorithm". In: *J Roy Stat. Soc* 39 (1977), pp. 1–17.

[48] Nicolas Dey, Laure Blanc-Feraud, Christophe Zimmer, Pascal Roux, Zvi Kam, Jean-Christophe Olivo-Marin, and Josiane Zerubia. "Richardson–Lucy Algorithm with Total Variation Regularization for 3D Confocal Microscope Deconvolution". In: *Microsc. Res. Tech.* 69.4 (2006), pp. 260–266.

[49] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu. "Denoising Prior Driven Deep Neural Network for Image Restoration". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41.10 (Oct. 2019), pp. 2305–2318. DOI: 10.1109/TPAMI.2018.2873610.

[50] D. L. Donoho. "Compressed Sensing". In: *IEEE Trans. Inf. Theory* 52.4 (Apr. 2006), pp. 1289–1306.

[51] D. L. Donoho, A. Maleki, and A. Montanari. "Message-Passing Algorithms for Compressed Sensing". In: *Proc. Nat. Acad. Sci.* 106.45 (Nov. 2009), pp. 18914–18919.

[52] François-Xavier Dupé, Jalal M Fadili, and Jean-Luc Starck. "A Proximal Iteration for Deconvolving Poisson Noisy Images Using Sparse Representations". In: *IEEE Trans. Image Process* 18.2 (2009), pp. 310–321.

[53] Jonathan Eckstein and Dimitri P. Bertsekas. "On the Douglas—Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators". In: *Mathematical Programming* 55.1 (Apr. 1992), pp. 293–318. DOI: 10.1007/BF01581204.

[54] M. Elad and M. Aharon. "Image Denoising via Sparse and Redundant Representations over Learned Dictionaries". In: *IEEE Trans. Image Process.* 15.12 (Dec. 2006), pp. 3736–3745.

[55] N. Eslahi and A. Foi. "Anisotropic Spatiotemporal Regularization in Compressive Video Recovery by Adaptively Modeling the Residual Errors as Correlated Noise". In: *IEEE Image, Video, and Multidimensional Signal Processing Workshop*. 2018.

[56] F. Knoll *et al.* "fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning". In: *Radiol. Artif. Intell.* 2.1 (2020), e190007.

[57] Zhenghan Fang, Yong Chen, Mingxia Liu, Lei Xiang, Qian Zhang, Qian Wang, Weili Lin, and Dinggang Shen. "Deep Learning for Fast and Spatially Constrained Tissue Quantification From Highly Accelerated Data in Magnetic Resonance Fingerprinting". In: *IEEE Trans Med Imaging* 38.10 (Oct. 2019), pp. 2364–2374. DOI: 10.1109/TMI.2019.2899328.

[58] M. Fazlyab, A. Robey, Hassani. H., M. Marari, and G. Pappas. "Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks". In: *Proc. Adv. Neural Inf. Process. Syst.* Vancouver, BC, Canada, Dec. 8, pp. 11427–11438.

[59] Li Feng, Dan Ma, and Fang Liu. "Rapid MR Relaxometry Using Deep Learning: An Overview of Current Techniques and Emerging Trends". In: *NMR Biomed* 35.4 (Apr. 2022), e4416. DOI: 10.1002/nbm.4416.

[60] J. A. Fessler. "Optimization Methods for Magnetic Resonance Image Reconstruction". In: *IEEE Signal Process. Mag.* 1.37 (Jan. 2020), pp. 33–40.

[61] M. A. T. Figueiredo and R. D. Nowak. "An EM Algorithm for Wavelet-Based Image Restoration". In: *IEEE Trans. Image Process.* 12.8 (Aug. 2003), pp. 906–916.

[62] M. A. T. Figueiredo and R. D. Nowak. "Wavelet-Based Image Estimation: An Empirical Bayes Approach Using Jeffreys' Noninformative Prior". In: *IEEE Trans. Image Process.* 10.9 (Sept. 2001), pp. 1322–1331.

[63] Mário AT Figueiredo and José M Bioucas-Dias. "Restoration of Poissonian Images Using Alternating Direction Optimization". In: *IEEE Trans. Image Process* 19.12 (2010), pp. 3133–3145.

[64] Alyson K Fletcher, Parthe Pandit, Sundeep Rangan, Subrata Sarkar, and Philip Schniter. "Plug-in Estimation in High-Dimensional Linear Inverse Problems: A Rigorous Analysis". In: *Proc. Advances in Neural Information Processing Systems (NIPS)*. Montreal, QC, Canada, Dec. 2018, pp. 7451–7460.

[65] Y. Gandelsman, A. Shocher, and M. Irani. "'Double-DIP': Unsupervised Image Decomposition via Coupled Deep-Image-Priors". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019, pp. 11026–11035.

[66] Yang Gao, Martijn Cloos, Feng Liu, Stuart Crozier, G. Bruce Pike, and Hongfu Sun. "Accelerating Quantitative Susceptibility and R2* Mapping Using Incoherent Undersampling and Deep Neural Network Reconstruction". In: *NeuroImage* 240 (Oct. 2021), p. 118404. DOI: 10.1016/j.neuroimage.2021.118404.

[67] R. G. Gavaskar and K. N. Chaudhury. "Plug-and-Play ISTA Converges with Kernel Denoisers". In: *IEEE Signal Process. Lett.* 27 (2020), pp. 610–614.

[68] Davis Gilton, Gregory Ongie, and Rebecca Willett. "Deep Equilibrium Architectures for Inverse Problems in Imaging". In: *IEEE Trans. Comput. Imaging* 7 (2021), pp. 1123–1133. DOI: 10.1109/TCI.2021.3118944.

[69] K. Gregor and Y. LeCun. "Learning Fast Approximation of Sparse Coding". In: *Proc. 27th Int. Conf. Machine Learning (ICML)*. Haifa, Israel, June 2010, pp. 399–406.

[70] R. Gribonval. "Should Penalized Least Squares Regression Be Interpreted as Maximum a Posteriori Estimation?" In: *IEEE Trans. Signal Process.* 59.5 (May 2011), pp. 2405–2410.

[71] R. Gribonval and P. Machart. "Reconciling "Priors" & "Priors" without Prejudice?" In: *Proc. Advances in Neural Information Processing Systems 26*. Lake Tahoe, NV, USA, Dec. 2013, pp. 2193–2201.

[72] S. S. Gurbani, S. Sheriff, A. A. Maudsley, H. Shim, and Lee A. D. Cooper. "Incorporation of a Spectral Model in a Convolutional Neural Network for Accelerated Spectral Fitting". In: *Magn Reson Med* 81.5 (2019), pp. 3346–3357. DOI: 10.1002/mrm.27641.

[73] Y. S. Han, J. Yoo, and J. C. Ye. "Deep Learning with Domain Adaptation for Accelerated Projection Reconstruction MR". In: *Magn Reson Med* 80.3 (Sept. 2017), pp. 1189–1205.

[74] Zachary T Harmany, Roummel F Marcia, and Rebecca M Willett. "This Is SPIRAL-TAP: Sparse Poisson Intensity Reconstruction Algorithms—Theory and Practice". In: *IEEE Trans. Image Process* 21.3 (2011), pp. 1084–1096.

[75] A. Hauptmann et al. "Model-Based Learning for Accelerated, Limited-View 3-D Photoacoustic Tomography". In: *IEEE Trans. Med. Imag.* 37.6 (2018), pp. 1382–1393.

[76] Heng Huang and C. Ding. "Robust Tensor Factorization Using R1 Norm". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. June 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587392.

[77] Diego Hernando, Karl K. Vigen, Ann Shimakawa, and Scott B. Reeder. "R2* Mapping in the Presence of Macroscopic B0 Field Variations". In: *Magn Reson Med* 68.3 (Sept. 2012), pp. 830–840. DOI: 10.1002/mrm.23306.

[78] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer Science & Business Media, 2012.

[79] S. A. Hosseini, B. Yaman, S. Moeller, M. Hong, and M. Akcakaya. "Dense Recurrent Neural Networks for Accelerated MRI: History-Cognizant Unrolling of Optimization Algorithms". In: *IEEE J. Sel. Topics Signal Process.* 14.6 (Oct. 2020), pp. 1280–1291.

[80] Y. Hu, S. G. Lingala, and M. Jacob. "A Fast Majorize-Minimize Algorithm for the Recovery of Sparse and Low-Rank Matrices". In: *IEEE Trans. Image Process.* 21.2 (Feb. 2012), pp. 742–753.

[81] F. Huang, S. Chen, and H. Huang. "Faster Stochastic Alternating Direction Method of Multipliers for Nonconvex Optimization". In: *Proc. 36th Int. Conf. Machine Learning (ICML)*. Long Beach, CA, USA, June 2019, pp. 2839–2848.

[82] R. Hyder, V. Shah, C. Hegde, and M. S. Asif. "Alternating Phase Projected Gradient Descent with Generative Priors for Solving Compressive Phase Retrieval". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.* Brighton, UK, May 2019, pp. 7705–7709.

[83] A. Jalal, L. Liu, A. G Dimakis, and C. Caramanis. "Robust Compressed Sensing Using Generative Models". In: *Adv. Neural Inf. Process. Syst.* 33 (2020).

[84] Haris Jeelani, Yang Yang, Ruixi Zhou, Christopher M. Kramer, Michael Salerno, and Daniel S. Weller. "A Myocardial T1-Mapping Framework with Recurrent and U-Net Convolutional Neural Networks". In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. Apr. 2020, pp. 1941–1944. DOI: 10.1109/ISBI45749.2020.9098459.

[85] Mark Jenkinson, Mickael Pechaud, and Stephen Smith. "BET2: MR-based Estimation of Brain, Skull and Scalp Surfaces". In: *Eleventh Annual Meeting of the Organization for Human Brain Mapping*. Vol. 17. Toronto, Ontario, Canada, June 2005, p. 167.

[86] Wenhao Jiang, Feiping Nie, and Heng Huang. "Robust Dictionary Learning with Capped L_1-Norm". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, July 2015, pp. 3590–3596.

[87] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. "Deep Convolutional Neural Network for Inverse Problems in Imaging". In: *IEEE Trans Image Process* 26.9 (Sept. 2017), pp. 4509–4522. DOI: 10.1109/TIP.2017.2713099.

[88] Woojin Jung, Steffen Bollmann, and Jongho Lee. "Overview of Quantitative Susceptibility Mapping Using Deep Learning: Current Status, Challenges and Opportunities". In: *NMR Biomed.* 35.4 (2022), e4292. DOI: 10.1002/nbm.4292.

[89] Sayan Kahali, Satya V.V.N. Kothapalli, Xiaojian Xu, Ulugbek S. Kamilov, and Dmitriy A. Yablonskiy. "Deep-Learning-Based Accelerated and Noise-Suppressed Estimation (DANSE) of Quantitative Gradient Recalled Echo (qGRE) MRI Metrics Associated with Human Brain Neuronal Structure and Hemodynamic Properties". In: *bioRxiv* (2021). DOI: 10.1101/2021.09.10.459810. eprint: https://www.biorxiv.org/content/early/2021/09/11/2021.09.10.459810.full.pdf.

[90] U. S. Kamilov. "A Parallel Proximal Algorithm for Anisotropic Total Variation Minimization". In: *IEEE Trans. Image Process.* 26.2 (Feb. 2017), pp. 539–548.

[91] U. S. Kamilov, H. Mansour, and B. Wohlberg. "A Plug-and-Play Priors Approach for Solving Nonlinear Imaging Inverse Problems". In: *IEEE Signal. Proc. Let.* 24.12 (Dec. 2017), pp. 1872–1876.

[92] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, D. Psaltis, and M. Unser. "Isotropic Inverse-Problem Approach for Two-Dimensional Phase Unwrapping". In: *J. Opt. Soc. Am. A* 32.6 (June 2015), pp. 1092–1100.

[93] U. S. Kamilov, P. Pad, A. Amini, and M. Unser. "MMSE Estimation of Sparse Lévy Processes". In: *IEEE Trans. Signal Process.* 61.1 (Jan. 2013), pp. 137–147.

[94] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis. "Optical Tomographic Image Reconstruction Based on Beam Propagation and Sparse Regularization". In: *IEEE Trans. Comp. Imag.* 2.1 (Mar. 2016), pp. 59–70.

[95] Ulugbek S. Kamilov, Charles A. Bouman, Gregery T. Buzzard, and Brendt Wohlberg. *Plug-and-Play Methods for Integrating Physical and Learned Models in Computational Imaging*. Mar. 2022. DOI: 10.48550/arXiv.2203.17061. arXiv: 2203.17061 [eess].

[96] A. Kazerouni, U. S. Kamilov, E. Bostan, and M. Unser. "Bayesian Denoising: From MAP to MMSE Using Consistent Cycle Spinning". In: *IEEE Signal Process. Lett.* 20.3 (Mar. 2013), pp. 249–252.

[97] M. Kellman, K. Zhang, E. Markley, J. Tamir, E. Bostan, M. Lustig, and L. Waller. "Memory-Efficient Learning for Large-Scale Computational Imaging". In: *IEEE Trans. Comput. Imag.* 6 (2020), pp. 1403–1414.

[98] B. Kim and J. C. Ye. "Mumford–Shah Loss Functional for Image Segmentation with Deep Learning". In: *IEEE Trans Image Process* 29 (2020), pp. 1856–1866. DOI: 10.1109/TIP.2019.2941265.

[99] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, May 2015, pp. 1–13.

[100] F. Knoll, K. Hammernik, C. Zhang, S. Moeller, T. Pock, D. K. Sodickson, and M. Akcakaya. "Deep-Learning Methods for Parallel Magnetic Resonance Imaging Reconstruction: A Survey of the Current Approaches, Trends, and Issues". In: *IEEE Signal Process. Mag.* 37.1 (Jan. 2020), pp. 128–140.

[101] Satya V. V. N. Kothapalli et al. *Quantitative Gradient Echo MRI Identifies Dark Matter as a New Imaging Biomarker of Neurodegeneration That Precedes Tissue Atrophy in Early Alzheimer Disease*. Apr. 2021. DOI: 10.1101/2021.04.27.21256098.

[102] A. Krizhevsky, I. Sutskevar, and G. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings Advances in Neural Information Processing Systems 25 (NeurIPS)*. Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.

[103] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová. "Statistical-Physics-Based Reconstruction in Compressed Sensing". In: *Phys. Rev. X* 2 (June 2012), p. 021005.

[104] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. "Reconnet: Non-iterative Reconstruction of Images from Compressively Sensed Measurements". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Caesars Palace, Las Vegas, NV, June 2016, pp. 449–458. DOI: 10.1109/CVPR.2016.55.

[105] K. Lange, D. R. Hunter, and I. Yang. "Optimization Transfer Using Surrogate Objective Functions". In: *J. Comput. Graph. Statist.* 9 (2000), pp. 1–20.

[106] F. Latorre, A. Eftekhari, and V. Cevher. "Fast and Provable ADMM for Learning with Generative Priors". In: *Proc. Adv. Neural Inf. Process. Syst.* Vancouver, BC, USA, Dec. 8-14, 2019, pp. 12027–12039.

[107] Emanuel Laude, Peter Ochs, and Daniel Cremers. "Bregman Proximal Mappings and Bregman–Moreau Envelopes under Relative Prox-Regularity". In: *J. Optim. Theory Appl.* 184.3 (2020), pp. 724–761.

[108] Hongyu Li et al. "Ultra-Fast Simultaneous T1rho and T2 Mapping Using Deep Learning". In: *ISMRM Annual Meeting*. 2020.

[109] R. Ling, W. Tahir, H.-Y. Lin, H. Lee, and L. Tian. "High-Throughput Intensity Diffraction Tomography with a Computational Microscope". In: *Biomed. Opt. Express* 9.5 (May 2018), pp. 2130–2141. DOI: 10.1364/BOE.9.002130.

[110] Fang Liu. "Improving Quantitative Magnetic Resonance Imaging Using Deep Learning". In: *Semin Musculoskelet Radiol* 24.4 (Aug. 2020), pp. 451–459. DOI: 10.1055/s-0040-1709482.

[111] Fang Liu, Li Feng, and Richard Kijowski. "MANTIS: Model-Augmented Neural neTwork with Incoherent k-Space Sampling for Efficient MR Parameter Mapping". In: *Magn Reson Med* 82.1 (July 2019), pp. 174–188. DOI: 10.1002/mrm.27707.

[112] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov. "Image Restoration Using Total Variation Regularized Deep Image Prior". In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, UK, May 2019, pp. 7715–7719. DOI: 10.1109/ICASSP.2019.8682856.

[113] Jiaming Liu, Xiaojian Xu, Weijie Gan, Shirin Shoushtari, and Ulugbek S. Kamilov. *Online Deep Equilibrium Learning for Regularization by Denoising*. May 2022. arXiv: 2205.13051 [cs, eess].

[114] Jiaming Liu, Salman Asif, Brendt Wohlberg, and Ulugbek Kamilov. "Recovery Analysis for Plug-and-Play Priors Using the Restricted Eigenvalue Condition". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 5921–5933. arXiv: 2106.03668.

[115] Jiaming Liu, Yu Sun, Weijie Gan, Xiaojian Xu, Brendt Wohlberg, and Ulugbek S. Kamilov. "SGD-Net: Efficient Model-Based Deep Learning With Theoretical Guarantees". In: *IEEE Trans. Comput. Imaging* 7 (2021), pp. 598–610. DOI: 10.1109/TCI.2021.3085534.

[116] A. X. Lu, O. Z. Kraus, S. Cooper, and A. M. Moses. "Learning Unsupervised Feature Representations for Single Cell Microscopy Images with Paired Cell Inpainting". In: *PLoS Comput Biol* 15.9 (Sept. 2019), e1007348. DOI: 10.1371/journal.pcbi.1007348.

[117] Cewu Lu, Jiaping Shi, and Jiaya Jia. "Online Robust Dictionary Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. June 2013, pp. 415–422.

[118] H. Lu, R.M. Freund, and Y. Nesterov. "Relatively Smooth Convex Optimization by First-Order Methods, and Applications". In: *SIAM J. Optim.* 28.1 (2018), pp. 333–354.

[119] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. "Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods". In: *IEEE Signal Process. Mag.* 35.1 (Jan. 2018), pp. 20–36.

[120] M. Lustig, D. L. Donoho, and J. M. Pauly. "Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging". In: *Magn. Reson. Med.* 58.6 (Dec. 2007), pp. 1182–1195.

[121] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. "Waterloo Exploration Database: New Challenges for Image Quality Assessment Models". In: *IEEE Trans. Image Process.* 26.2 (2016), pp. 1004–1016.

[122] J. Mairal. "Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning". In: *SIAM J. Optim.* 25.2 (Jan. 2015), pp. 829–855.

[123] Markku Makitalo and Alessandro Foi. "Optimal Inversion of the Anscombe Transformation in Low-Count Poisson Image Denoising". In: *IEEE Trans. Image Process* 20.1 (2010), pp. 99–109.

[124] 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. IEEE Int. Conf. Comp. Vis. (ICCV)*. Vancouver, Canada, July 2001, pp. 416–423.

[125] Gary Mataev, Peyman Milanfar, and Michael Elad. "DeepRED: Deep Image Prior Powered by RED". In: *ICCV 2019 Workshop on Learning for Computational Imaging*. Seoul, Korea, Oct. 2019.

[126] A. Matakos, S. Ramani, and J. A. Fessler. "Accelerated Edge-Preserving Image Restoration without Boundary Artifacts". In: *IEEE Trans. Image Process.* 22.5 (May 2013), pp. 2019–2029.

[127] M. T. McCann, K. H. Jin, and M. Unser. "Convolutional Neural Networks for Inverse Problems in Imaging: A Review". In: *IEEE Signal Process. Mag.* 34.6 (2017), pp. 85–95.

[128] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers. "Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems". In: *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*. 2017, pp. 1799–1808.

[129] C. Metzler, P. Schniter, A. Veeraraghavan, and R. Baraniuk. "prDeep: Robust Phase Retrieval with a Flexible Deep Network". In: *Proc. 35th Int. Conf. Machine Learning (ICML)*. Stockholmsmässan, Stockholm Sweden, July 2018, pp. 3501–3510.

[130] C. A. Metzler, A. Maleki, and R. Baraniuk. "BM3D-PRGAMP: Compressive Phase Retrieval Based on BM3D Denoising". In: *Proc. IEEE Int. Conf. Image Proc. (ICIP)*. Phoenix, AZ, USA, Sept. 2016, pp. 2504–2508.

[131] C. A. Metzler, A. Maleki, and R. G. Baraniuk. "From Denoising to Compressed Sensing". In: *IEEE Trans. Inf. Theory* 62.9 (Sept. 2016), pp. 5117–5144. DOI: 10.1109/TIT.2016.2556683.

[132] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. 2020.

[133] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. "Spectral Normalization for Generative Adversarial Networks". In: *Int. Conf. on Learn. Representations*. Apr. 30.

[134] Subhadip Mukherjee, Marcello Carioni, Ozan Öktem, and Carola-Bibiane Schönlieb. "End-to-End Reconstruction Meets Data-Driven Regularization for Inverse Problems". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 21413–21425.

[135] Pravin Nair, Ruturaj G. Gavaskar, and Kunal Narayan Chaudhury. "Fixed-Point and Objective Convergence of Plug-and-Play Algorithms". In: *IEEE Trans. Comput. Imaging* 7 (2021), pp. 337–348. DOI: 10.1109/TCI.2021.3066053.

[136] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[137] Y. E. Nesterov. "A Method for Solving the Convex Programming Problem with Convergence Rate O($1/K^2$)". In: *Dokl. Akad. Nauk* 269 (1983), pp. 543–547.

[138] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. "Efficient and Robust Feature Selection via Joint 2,1-Norms Minimization". In: *In NIPS*. Vancouver, British Columbia, Canada, Dec. 2010, pp. 1813–1821.

[139] J. Nocedal and S. J. Wright. *Numerical Optimization*. Second. Springer, 2006.

[140] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. "Deep Learning Techniques for Inverse Problems in Imaging". In: *IEEE J. Sel. Areas Inf. Theory* 1.1 (2020), pp. 39–56.

[141] S. Ono. "Primal-Dual Plug-and-Play Image Restoration". In: *IEEE Signal. Proc. Let.* 24.8 (Aug. 2017), pp. 1108–1112.

[142] H. Ouyang, N. He, L. Q. Tran, and A. Gray. "Stochastic Alternating Direction Method of Multipliers". In: *Proc. 30th Int. Conf. Machine Learning (ICML)*. Atlanta, GA, USA, 16, pp. 80–88.

[143] Ananya Panda, Bhairav B. Mehta, Simone Coppo, Yun Jiang, Dan Ma, Nicole Seiberlich, Mark A. Griswold, and Vikas Gulani. "Magnetic Resonance Fingerprinting-An Overview". In: *Curr Opin Biomed Eng* 3 (Sept. 2017), pp. 56–66. DOI: 10.1016/j.cobme.2017.11.001.

[144] N. Parikh and S. Boyd. "Proximal Algorithms". In: *Found. Trends Optim.* 1.3 (2014), pp. 123–231.

[145] A. Raj, Y. Li, and Y. Bresler. "GAN-based Projector for Faster Recovery in Compressed Sensing with Convergence Guarantees". In: *Proc. IEEE Int. Conf. Comput. Vis.* Seoul, South Korea, Oct. 2019, pp. 5601–5610.

[146] S. Ramani and J. A. Fessler. "A Splitting-Based Iterative Algorithm for Accelerated Statistical X-ray CT Reconstruction". In: *IEEE Trans. Med. Imaging* 31.3 (Mar. 2012), pp. 677–688.

[147] S. Rangan. "Generalized Approximate Message Passing for Estimation with Random Linear Mixing". In: *Proc. IEEE Int. Symp. Information Theory*. St. Petersburg, Russia, July 2011, pp. 2168–2172.

[148] M. Razaviyayn, M. Hong, and Z.-Q. Luo. "A Stochastic Successive Minimization Method for Nonsmooth Nonconvex Optimization". In: (2013).

[149] E. T. Reehorst and P. Schniter. "Regularization by Denoising: Clarifications and New Interpretations". In: *IEEE Trans. Comput. Imag.* 5.1 (Mar. 2019), pp. 52–67.

[150] Nathan T. Roberts, Louis A. Hinshaw, Timothy J. Colgan, Takanori Ii, Diego Hernando, and Scott B. Reeder. "B0 and B1 Inhomogeneities in the Liver at 1.5 T and 3.0 T". In: *Magn Reson Med* 85.4 (2021), pp. 2212–2220. DOI: 10.1002/mrm.28549.

[151] R. T. Rockafellar. "Convex Analysis". In: Princeton, NJ: Princeton Univ. Press, 1970. Chap. Conjugate Saddle-Functions and Minimax Theorems, pp. 388–398.

[152] R. T. Rockafellar and R. J-B Wets. *Variational Analysis*. Springer, 1998.

[153] Y. Romano, M. Elad, and P. Milanfar. "The Little Engine That Could: Regularization by Denoising (RED)". In: *SIAM J. Imaging Sci.* 10.4 (2017), pp. 1804–1844.

[154] Arie Rond, Raja Giryes, and Michael Elad. "Poisson Inverse Problems by the Plug-and-Play Scheme". In: *J. Vis. Commun. Image Represent.* 41 (2016), pp. 96–108.

[155] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Munich, Germany, Oct. 2015, pp. 234–241.

[156] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear Total Variation Based Noise Removal Algorithms". In: *Physica D* 60.1–4 (Nov. 1992), pp. 259–268.

[157] E. K. Ryu and S. Boyd. "A Primer on Monotone Operator Methods". In: *Appl. Comput. Math.* 15.1 (2016), pp. 3–43.

[158] 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: *Proc. 36th Int. Conf. Machine Learning (ICML)*. Vol. 97. Long Beach, CA, USA, June 2019, pp. 5546–5557.

[159] 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 Trans. Med. Imag.* 37.2 (Feb. 2018), pp. 491–503.

[160] H. Sedghi, V. Gupta, and P. M. Long. "The Singular Values of Convolutional Layers". In: *International Conference on Learning Representations (ICLR)*. 2019.

[161] O. Senouf, S. Vedula, T. Weiss, A. Bronstein, O. Michailovich, and M. Zibulevsky. "Self-Supervised Learning of Inverse Problem Solvers in Medical Imaging". In: *DART/MIL3ID@MICCAI*. Shenzen, China, Oct. 2019, pp. 111–119. arXiv: 1905.09325.

[162] V. Shah and C. Hegde. "Solving Linear Inverse Problems Using GAN Priors: An Algorithm with Provable Guarantees". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.* Calgary, AB, Canada, Apr. 2018, pp. 4609–4613.

[163] W. Shi, F. Jiang, S. Liu, and D. Zhao. "Scalable Convolutional Neural Network for Image Compressed Sensing". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, June 2019, pp. 12282–12291. DOI: 10.1109/CVPR.2019.01257.

[164] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[165] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. "Implicit Neural Representations with Periodic Activation Functions". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 7462–7473.

[166] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman. "Plug-and-Play Priors for Bright Field Electron Tomography and Sparse Interpolation". In: *IEEE Trans. Comput. Imaging* 2.4 (Dec. 2016), pp. 408–423.

[167] J-L Starck and Fionn Murtagh. "Astronomical Image and Data Analysis". In: (2007).

[168] Y. Sun, B. Wohlberg, and U. S. Kamilov. "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction". In: *IEEE Trans. Comput. Imaging* 5.3 (Sept. 2019), pp. 395–408.

[169] Y. Sun, S. Xu, Y. Li, L. Tian, B. Wohlberg, and U. S. Kamilov. "Regularized Fourier Ptychography Using an Online Plug-and-Play Algorithm". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*. Brighton, UK, May 2019, pp. 7665–7669. DOI: 10.1109/ICASSP.2019.8683057.

[170] Yu Sun, Jiaming Liu, and Ulugbek Kamilov. "Block Coordinate Regularization by Denoising". In: *Adv. Neural Inf. Process. Syst. NeurIPS* 32 (2019).

[171] Yu Sun, Jiaming Liu, Mingyang Xie, Brendt Wohlberg, and Ulugbek S. Kamilov. "CoIL: Coordinate-Based Internal Learning for Tomographic Imaging". In: *IEEE Trans. Comput. Imaging* 7 (2021), pp. 1400–1412. DOI: 10.1109/TCI.2021.3125564.

[172] Yu Sun, Zihui Wu, Xiaojian Xu, Brendt Wohlberg, and Ulugbek S. Kamilov. "Scalable Plug-and-Play ADMM With Convergence Guarantees". In: *IEEE Trans. Comput. Imaging* 7 (2021), pp. 849–863. DOI: 10.1109/TCI.2021.3094062.

[173] T. Suzuki. "Dual Averaging and Proximal Gradient Descent for Online Alternating Direction Multiplier Method". In: *Proc. 30th Int. Conf. Machine Learning (ICML)*. Atlanta, GA, USA, June 2013, pp. 392–400.

[174] J. Tan, Y. Ma, and D. Baron. "Compressive Imaging via Approximate Message Passing with Image Denoising". In: *IEEE Trans. Signal Process.* 63.8 (Apr. 2015), pp. 2085–2092.

[175] M. Teboulle. "A Simplified View of First Order Methods for Optimization". In: *Math. Program.* 170.1 (2018), pp. 67–96.

[176] A. Teodoro, J. M. Bioucas-Dias, and M. A. T. Figueiredo. "Scene-Adapted Plug-and-Play Algorithm with Convergence Guarantees". In: *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing*. Tokyo, Japan, Sept. 2017, pp. 1–6.

[177] A. M. Teodoro, J. M. Bioucas-Dias, and M. A. T. Figueiredo. "A Convergent Image Fusion Algorithm Using Scene-Adapted Gaussian-Mixture-Based Denoising". In: *IEEE Trans. Image Process.* 28.1 (Jan. 2019), pp. 451–463.

[178] M. Terris, A. Repetti, J.-C. Pesquet, and Y. Wiaux. "Building Firmly Nonexpansive Convolutional Neural Networks". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.* Barcelona, Spain, May 2020, pp. 8658–8662.

[179] L. Tian, Z. Liu, L. Yeh, M. Chen, J. Zhong, and L. Waller. "Computational Illumination for High-Speed in Vitro Fourier Ptychographic Microscopy". In: *Optica* 2.10 (2015), pp. 904–911.

[180] R. Tibshirani. "Regression and Selection via the Lasso". In: *J R Stat Soc Ser. B Methodol.* 58.1 (1996), pp. 267–288.

[181] T. Tirer and R. Giryes. "Image Restoration by Iterative Denoising and Backward Projections". In: *IEEE Trans. Image Process.* 28.3 (Mar. 2019), pp. 1220–1234.

[182] M. Torop, S. V. V. N. Kothapalli, Y. Sun, J. Liu, S. Kahali, D. A. Yablonskiy, and U. S. Kamilov. "Deep Learning Using a Biophysical Model for Robust and Accelerated Reconstruction of Quantitative, Artifact-Free and Denoised R2* Images". In: *Magn Reson Med* 84.6 (Dec. 2020), pp. 2932–2942. DOI: 10.1002/mrm.28344.

[183] Martin Uecker, Peng Lai, Mark J. Murphy, Patrick Virtue, Michael Elad, John M. Pauly, Shreyas S. Vasanawala, and Michael Lustig. "ESPIRiT — An Eigenvalue Approach to Autocalibrating Parallel MRI: Where SENSE Meets GRAPPA". In: *Magn Reson Med* 71.3 (Mar. 2014), pp. 990–1001. DOI: 10.1002/mrm.24751.

[184] Xialing Ulrich and Dmitriy A. Yablonskiy. "Separation of Cellular and BOLD Contributions to T2* Signal Relaxation". In: *Magn Reson Med* 75.2 (Feb. 2016), pp. 606–615. DOI: 10.1002/mrm.25610.

[185] D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Deep Image Prior". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, June 2018, pp. 9446–9454.

[186] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. "Plug-and-Play Priors for Model Based Reconstruction". In: *Proc. IEEE Global Conf. Signal Process. and Inf. Process.* Austin, TX, USA, Dec. 2013, pp. 945–948.

[187] G. Wang, J. C. Ye, and B. De Man. "Deep Learning for Tomographic Image Reconstruction". In: *Nat. Mach. Intell.* 2.12 (2020), pp. 737–748.

[188] H. Wang and A. Banerjee. "Online Alternating Direction Method". In: *Proc. 29th Int. Conf. Machine Learning (ICML)*. Edinburgh, Scotland, UK, June 2012, pp. 1699–1706.

[189] S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang. "Accelerating Magnetic Resonance Imaging via Deep Learning". In: *Proc. Int. Symp. Biomedical Imaging*. Apr. 2016, pp. 514–517. DOI: 10.1109/ISBI.2016.7493320.

[190] Kaixuan Wei, Angelica I. Avilés-Rivero, Jingwei Liang, Y. Fu, C. Schönlieb, and H. Huang. "Tuning-Free Plug-and-Play Proximal Algorithm for Inverse Imaging Problems". In: *ICML*. 2020, 13–18 Jul, pp. 10158–10169.

[191] Jie Wen, Anne H. Cross, and Dmitriy A. Yablonskiy. "On the Role of Physiological Fluctuations in Quantitative Gradient Echo MRI – Implications for GEPCI, QSM and SWI". In: *Magn Reson Med* 73.1 (Jan. 2015), pp. 195–203. DOI: 10.1002/mrm.25114.

[192] Jie Wen, Manu S. Goyal, Serguei V. Astafiev, Marcus E. Raichle, and Dmitriy A. Yablonskiy. "Genetically Defined Cellular Correlates of the Baseline Brain MRI Signal". In: *PNAS* 115.41 (Oct. 2018), E9727–E9736. DOI: 10.1073/pnas.1808121115.

[193] B. Wohlberg. "Efficient Algorithms for Convolutional Sparse Representations". In: *IEEE Trans. Image Process.* 25.1 (Jan. 2016), pp. 301–315.

[194] Yan Wu, Yajun Ma, Jiang Du, and Lei Xing. "Accelerating Quantitative MR Imaging with the Incorporation of B1 Compensation Using Deep Learning". In: *Magnetic Resonance Imaging* 72 (Oct. 2020), pp. 78–86. DOI: 10.1016/j.mri.2020.06.011.

[195] Z. Wu, Y. Sun, J. Liu, and U. S. Kamilov. "Online Regularization by Denoising with Applications to Phase Retrieval". In: *The IEEE International Conference on Computer Vision (ICCV) Workshops*. Oct. 2019.

[196] Z. Wu, Y. Sun, A. Matlock, J. Liu, L. Tian, and U. S. Kamilov. "SIMBA: Scalable Inversion in Optical Tomography Using Deep Denoising Priors". In: *IEEE J. Sel. Topics Signal Process.* (2020), pp. 1–1.

[197] B Xiang, W Jie, R Schmidt, D Yablonskiy, and A Cross. "Quantitative Assessment of Multiple Sclerosis Tissue Damage and Partial Repair in a Biopsy Proven Demyelinating Brain Lesion Using Gradient Recalled Echo Imaging". In: *Multiple Sclerosis Journal*. Vol. 26. SAGE PUBLICATIONS LTD 1 OLIVERS YARD, 55 CITY ROAD, LONDON EC1Y 1SP, ENGLAND. London, England, May 2020, pp. 93–94.

[198] X. Xu and U. S. Kamilov. "SignProx: One-bit Proximal Algorithm for Nonconvex Stochastic Optimization". In: *IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*. Brighton, UK, May 2019, pp. 7800–7804. DOI: 10.1109/ICASSP.2019.8682059.

[199] X. Xu, Y. Sun, J. Liu, B. Wohlberg, and U. S. Kamilov. "Provable Convergence of Plug-and-Play Priors with MMSE Denoisers". In: *IEEE Signal Process. Lett.* 27 (2020), pp. 1280–1284.

[200] Xiaojian Xu, Jiaming Liu, Yu Sun, Brendt Wohlberg, and Ulugbek S. Kamilov. "Boosting the Performance of Plug-and-Play Priors via Denoiser Scaling". In: *54th Asilomar Conf. on Signals, Systems, and Computers*. 2020, pp. 1305–1312. DOI: 10.1109/IEEECONF51394.2020.9443410.

[201] Xiaojian Xu, Satya V. V. N. Kothapalli, Jiaming Liu, Sayan Kahali, Weijie Gan, Dmitriy A. Yablonskiy, and Ulugbek S. Kamilov. "Learning-Based Motion Artifact Removal Networks for Quantitative $R2_*$ Mapping". In: *Magn. Reson. Med.* 88.1 (2022), pp. 106–119. DOI: 10.1002/mrm.29188. arXiv: 2109.01622.

[202] Xiaojian Xu, Oussama Dhifallah, Hassan Mansour, Petros T. Boufounos, and Philip V. Orlik. "Robust 3D Tomographic Imaging of the Ionospheric Electron Density". In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. Sept. 2020, pp. 437–440. DOI: 10.1109/IGARSS39084.2020.9324189.

[203] D. A. Yablonskiy. "Quantitation of Intrinsic Magnetic Susceptibility-Related Effects in a Tissue Matrix. Phantom Study". In: *Magn Reson Med* 39.3 (Mar. 1998), pp. 417–428. DOI: 10.1002/mrm.1910390312.

[204] D. A. Yablonskiy, A. L. Sukstanskii, J. Luo, and X. Wang. "Voxel Spread Function Method for Correction of Magnetic Field Inhomogeneity Effects in Quantitative Gradient-Echo-Based MRI". In: *Magn Reson Med* 70.5 (Nov. 2013), pp. 1283–1292. DOI: 10.1002/mrm.24585.

[205] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya. "Self-Supervised Learning of Physics-Guided Reconstruction Neural Networks without Fully Sampled Reference Data". In: *Magn Reson Med* (July 2020).

[206] 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: *2020 IEEE International Symposium on Biomedical Imaging (ISBI)*. Iowa City, Iowa, USA: IEEE Computer Society, Apr. 2020, pp. 921–925. DOI: 10.1109/ISBI45749.2020.9098514.

[207] Y. Yang, J. Sun, H. Li, and Z. Xu. "Deep ADMM-Net for Compressive Sensing MRI". In: *Advances in Neural Information Processing Systems 29*. 2016, pp. 10–18.

[208] Z. Yang and M. Jacob. "Nonlocal Regularization of Inverse Problems: A Unified Variational Framework". In: *IEEE Trans. Image Process.* 22.8 (Aug. 2013), pp. 3192–3203.

[209] Jaeyeon Yoon et al. "Quantitative Susceptibility Mapping Using Deep Neural Network: QSMnet". In: *NeuroImage* 179 (Oct. 2018), pp. 199–206. DOI: 10.1016/j.neuroimage.2018.06.030.

[210] Di You, Jingfen Xie, and Jian Zhang. "ISTA-Net++: Flexible Deep Unfolding Network for Compressive Sensing". In: *2021 IEEE International Conference on Multimedia and Expo (ICME)*. 2021, pp. 1–6. DOI: 10.1109/ICME51207.2021.9428249. arXiv: 2103.11554.

[211] Y. Yu. "Better Approximation and Faster Algorithm Using the Proximal Average". In: *Neural Information Processing Systems*. Lake Tahoe, CA, USA, Dec. 2013, pp. 458–466.

[212] Gushan Zeng, Yi Guo, Jiaying Zhan, Zi Wang, Zongying Lai, Xiaofeng Du, Xiaobo Qu, and Di Guo. "A Review on Deep Learning MRI Reconstruction without Fully Sampled K-Space". In: *BMC Medical Imaging* 21.1 (Dec. 2021), p. 195. DOI: 10.1186/s12880-021-00727-9.

[213] Wei Zha, Sean B Fain, Richard Kijowski, and Fang Liu. "Relax-MANTIS: REference-free LAtent Map-eXtracting MANTIS for Efficient MR Parametric Mapping with Unsupervised Deep Learning". In: *27th ISMRM Annual Meeting*. 2019.

[214] J. Zhang and B. Ghanem. "ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 1828–1837. DOI: 10.1109/CVPR.2018.00196.

[215] K. Zhang, W. Zuo, and L. Zhang. "Deep Plug-and-Play Super-Resolution for Arbitrary Blur Kernels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, USA, 2019, pp. 1671–1681.

[216] 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 Trans. Image Process* 26.7 (July 2017), pp. 3142–3155.

[217] K. Zhang, W. Zuo, S. Gu, and L. Zhang. "Learning Deep CNN Denoiser Prior for Image Restoration". In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. July 2017, pp. 3929–3938.

[218] Yue Zhao et al. “In Vivo Detection of Microstructural Correlates of Brain Pathology in Preclinical and Early Alzheimer Disease with Magnetic Resonance Imaging”. In: *Neuroimage* 148 (Mar. 2017), pp. 296–304. DOI: 10.1016/j.neuroimage.2016.12.026.

[219] Yue Zhao, Jie Wen, Anne H. Cross, and Dmitriy A. Yablonskiy. “On the Relationship between Cellular and Hemodynamic Properties of the Human Brain Cortex throughout Adult Lifespan”. In: *Neuroimage* 133 (June 2016), pp. 417–429. DOI: 10.1016/j.neuroimage.2016.03.022.

[220] W. Zhong and J. Kwok. “Fast Stochastic Alternating Direction Method of Multipliers”. In: *Proc. 31th Int. Conf. Machine Learning (ICML)*. Bejing, China, June 2014, pp. 46–54.

[221] Marcelo V. W. Zibetti, Patricia M. Johnson, Azadeh Sharafi, Kerstin Hammernik, Florian Knoll, and Ravinder R. Regatte. “Rapid Mono and Biexponential 3D-T1ρ Mapping of Knee Cartilage Using Variational Networks”. In: *Sci Rep* 10 (Nov. 2020), p. 19144. DOI: 10.1038/s41598-020-76126-x.

Appendix A

Declaration

A.1 Declaration of Previous Publications and Contribution

Most of the work presented in this dissertation has been published, with each being a result of collaboration with other researchers. Here we list the references to those previous publications and describe the contribution of each author to the work.

Chapter 4 is based on our paper [200]. In this work, Xu was the main author who conducted the experiments and analyzed the results. Xu and Kamilov collaboratively wrote the paper. Liu helped with the denoiser training. Sun contributed to the discussion. Wohlberg and Kamilov revised the paper.

Chapter 5 is based on our paper [199]. In this work, Xu was the main author who conducted the experiments and analyzed the results. Xu and Kamilov collaboratively wrote the paper. Liu and Sun contributed to the discussion. Wohlberg and Kamilov revised the paper.

Chapter 6 is based on our paper [172]. In this work, Sun, Wu and Xu shared equal contribution. Sun and Wu conducted the experiments and analyzed the results. Sun, Wu and Kamilov drafted the original paper. Wohlberg helped revise the paper. Xu conducted some additional suggested experiments, revised and rearranged the paper for the final publication.

Chapter 7 is based on our paper [6]. In this work, Al-Shabili and Xu shared equal contribution. Al-Shabili and Selesnick proposed the original idea and Al-Shabili performed theoretical analysis. Xu conducted the experiments and analyzed the results. Al-Shabili and Xu collaboratively wrote the paper. Kamilov revised the paper.

Chapter 8 is an unpublished work upon on the completion of this dissertation. In this work, Xu was the main author who conducted the experiments, analyzed the results and drafted the manuscript. Gan helped with the discussion. Yablonskiy provided the raw k-space data for experiments. Kamilov revised the manuscript.

Appendix B

Background Material

We briefly introduce the definitions and propositions in monotone operator theory that are related to our analysis in the dissertation. We note that the content we present in this section are well-known results from the optimization literature that can be found in different forms in standard textbooks [12, 27, 136, 152].

B.1 Properties of Monotone Operators

Definition B.1. An operator T is Lipschitz continuous with constant $\lambda > 0$ if

$$\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\| \leq \lambda \|\boldsymbol{x} - \boldsymbol{y}\| \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n \ .$$

When $\lambda = 1$, we say that T is nonexpansive. When $\lambda < 1$, we say that T is a contraction.

Definition B.2. T is monotone if

$$(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq 0 \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n \ .$$

We say that it is strongly monotone or coercive with parameter $\mu > 0$ if

$$(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \mu\|\boldsymbol{x} - \boldsymbol{y}\|^2 \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n \ .$$

Definition B.3. T is cocoercive with constant $\beta > 0$ if

$$(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \beta\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|^2 \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n \ .$$

When $\beta = 1$, we say that T is firmly nonexpansive.

The following results are derived from the definition above.

Proposition B.1. *Consider* $\mathsf{R} = \mathsf{I} - \mathsf{T}$ *where* $\mathsf{T} : \mathbb{R}^n \to \mathbb{R}^n$.

$$\mathsf{T} \textit{ is nonexpansive } \Leftrightarrow \mathsf{R} \textit{ is } (1/2)\textit{-cocoercive} \ .$$

Proof. First suppose that R is $1/2$ cocoercive. Let $\boldsymbol{h} \coloneqq \boldsymbol{x} - \boldsymbol{y}$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$. We then have

$$\frac{1}{2}\|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\|^2 \leq (\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} = \|\boldsymbol{h}\|^2 - (\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} \ .$$

We also have that

$$\frac{1}{2}\|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\|^2 = \frac{1}{2}\|\boldsymbol{h}\|^2 - (\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} + \frac{1}{2}\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|^2 \ .$$

By combining these two and simplifying the expression

$$\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\| \leq \|\boldsymbol{h}\| \ .$$

The converse can be proved by following this logic in reverse.

Proposition B.2. *Consider* $\mathsf{R} = \mathsf{I} - \mathsf{T}$ *where* $\mathsf{T} : \mathbb{R}^n \to \mathbb{R}^n$.

$$\begin{aligned} & \mathsf{T} \text{ is Lipschitz continuous with constant } \lambda < 1 \\ \Rightarrow \quad & \mathsf{R} \text{ is } (1-\lambda)\text{-strongly monotone}\,. \end{aligned}$$

Proof. By using the Cauchy-Schwarz inequality, we have for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$\begin{aligned} & (\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \\ & = \|\boldsymbol{x} - \boldsymbol{y}\|^2 - (\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \\ & \geq \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|\|\boldsymbol{x} - \boldsymbol{y}\| \\ & \geq \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \lambda\|\boldsymbol{x} - \boldsymbol{y}\|^2 \geq (1-\lambda)\|\boldsymbol{x} - \boldsymbol{y}\|^2\,. \end{aligned}$$

Definition B.4. For a constant $\alpha \in (0, 1)$, we say that T is α-averaged, if there exists a nonexpansive operator N such that $\mathsf{T} = (1-\alpha)\mathsf{I} + \alpha\mathsf{N}$.

The following characterization is often convenient.

Proposition B.3. *For a nonexpansive operator* T*, a constant* $\alpha \in (0, 1)$*, and the operator* $\mathsf{R} \coloneqq \mathsf{I} - \mathsf{T}$*, the following are equivalent*

(a) T *is* α*-averaged*

(b) $(1 - 1/\alpha)\mathsf{I} + (1/\alpha)\mathsf{T}$ *is nonexpansive*

(c) $\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|^2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \left(\frac{1-\alpha}{\alpha}\right)\|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\|^2$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

Proof. See Proposition 4.35 in [12].

Proposition B.4. *Consider* $\mathsf{T} : \mathbb{R}^n \to \mathbb{R}^n$ *and* $\beta > 0$*. Then, the following are equivalent*

(a) T *is* β*-cocoercive*

(b) $\beta\mathsf{T}$ *is firmly nonexpansive*

(c) $\mathsf{I} - \beta\mathsf{T}$ *is firmly nonexpansive.*

(d) $\beta\mathsf{T}$ *is* $(1/2)$*-averaged.*

(e) $\mathsf{I} - 2\beta\mathsf{T}$ *is nonexpansive.*

Proof. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, let $\boldsymbol{h} \coloneqq \boldsymbol{x} - \boldsymbol{y}$. The equivalence between (a) and (b) is readily observed by defining $\mathsf{P} \coloneqq \beta\mathsf{T}$ and noting that

$$
\begin{aligned}
&(\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} = \beta(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} \quad \text{and} \\
&\|\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y}\|^2 = \beta^2\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\| \ .
\end{aligned}
$$

Define $\mathsf{R} \coloneqq \mathsf{I} - \mathsf{P}$ and suppose (b) is true, then

$$
\begin{aligned}
&(\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} \\
&= \|\boldsymbol{h}\|^2 - (\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} \\
&= \|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\|^2 + (\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h} - \|\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y}\|^2 \\
&\geq \|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\|^2 \ .
\end{aligned}
$$

By repeating the same argument for $\mathsf{P} = \mathsf{I} - \mathsf{R}$, we establish the full equivalence between (b) and (c).

The equivalence of (b) and (d) can be seen by noting that

$$\begin{aligned} & 2\|\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y}\|^2 \leq 2(\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y})^\mathsf{T}\boldsymbol{h} \\ \Leftrightarrow \quad & \|\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y}\|^2 \leq 2(\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y})^\mathsf{T}\boldsymbol{h} - \|\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y}\|^2 \\ & = \|\boldsymbol{h}\|^2 - (\|\boldsymbol{h}\|^2 - 2(\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y})^\mathsf{T}\boldsymbol{h} + \|\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y}\|^2) \\ & = \|\boldsymbol{h}\|^2 - \|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\|^2 \, . \end{aligned}$$

To show the equivalence with (e), first suppose that $\mathsf{N} \coloneqq \mathsf{I} - 2\mathsf{P}$ is nonexpansive, then $\mathsf{P} = \frac{1}{2}(\mathsf{I} + (-\mathsf{N}))$ is 1/2-averaged, which means that it is firmly nonexpansive. On the other hand, if P is firmly nonexpansive, then it is 1/2-averaged, which means that from Proposition B.3(b) we have that $(1-2)\mathsf{I} + 2\mathsf{P} = 2\mathsf{P} - \mathsf{I} = -\mathsf{N}$ is nonexpansive. This directly means that N is nonexpansive.

B.2 Convex Functions, Subdifferentials, and Proximal Operators

Definition B.5. A continuously differentiable function f is called convex on $\mathbb{R}^n$ if

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) \tag{B.1}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$. If $-f$ is convex, we call f concave.

The definition is also known as the first-order convexity inequality, for which more details can be found in Section 2.1 of [136].

Proposition B.5. *Let f be convex and differentiable function with ∇f that is L-Lipschitz continuous. Then,*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$.

Proof. The proof is a minor variation of the one presented in Section 2.1 of [136]. □

Proposition B.6. *Let f be a proper, closed, and convex function. Then for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{g} \in \partial f(\boldsymbol{x})$, and $\boldsymbol{h} \in \partial f(\boldsymbol{y})$, ∂f is a monotone operator*

$$(\boldsymbol{g} - \boldsymbol{h})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq 0 \ .$$

Additionally if f is strongly convex with constant $\mu > 0$, then ∂f is strongly monotone with the same constant.

$$(\boldsymbol{g} - \boldsymbol{h})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \mu\|\boldsymbol{x} - \boldsymbol{y}\|^2 \ .$$

Proof. Consider a strongly convex function f with a constant $\mu \geq 0$. Then, we have that

$$\begin{aligned} &\begin{cases} f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 \\ f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \boldsymbol{h}^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \end{cases} \\ \Rightarrow \quad &(\boldsymbol{g} - \boldsymbol{h})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \mu\|\boldsymbol{x} - \boldsymbol{y}\|^2 \ . \end{aligned}$$

The proof for a weakly convex f is obtained by considering $\mu = 0$ in the inequalities above.

It is well-known that the proximal operator is firmly nonexpansive.

Proposition B.7. *Proximal operator* $\mathsf{prox}_{\gamma f}$ *of a proper, closed, and convex* f *is firmly nonexpansive.*

Proof. Denote with $\boldsymbol{x}_1 = \mathsf{G}\boldsymbol{z}_1 = \mathsf{prox}_{\gamma f}(\boldsymbol{z}_1)$ and $\boldsymbol{x}_2 = \mathsf{G}\boldsymbol{z}_2 = \mathsf{prox}_{\gamma f}(\boldsymbol{z}_2)$, then

$$
\begin{aligned}
&\begin{cases} (\boldsymbol{z}_1 - \boldsymbol{x}_1) \in \gamma \partial f(\boldsymbol{x}_1) \\ (\boldsymbol{z}_2 - \boldsymbol{x}_2) \in \gamma \partial f(\boldsymbol{x}_2) \end{cases} \\
\Rightarrow \quad & (\boldsymbol{z}_1 - \boldsymbol{x}_1 - \boldsymbol{z}_2 + \boldsymbol{x}_2)^{\mathsf{T}} (\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq 0 \\
\Rightarrow \quad & (\mathsf{G}\boldsymbol{z}_1 - \mathsf{G}\boldsymbol{z}_2)^{\mathsf{T}} (\boldsymbol{z}_1 - \boldsymbol{z}_2) \geq \|\mathsf{G}\boldsymbol{z}_1 - \mathsf{G}\boldsymbol{z}_2\|^2 .
\end{aligned}
$$

The following proposition is sometimes referred to as *Moreau-Rockafellar theorem*. It establishes that for functions defined over all of $\mathbb{R}^n$, we have that $\partial f = \partial f_1 + \cdots + \partial f_m$.

Proposition B.8. *Consider* $f = f_1 + \cdots + f_m$*, where* $f_1, \ldots, f_m$ *are proper, closed, and convex functions on* $\mathbb{R}^n$*. Then*

$$
\partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}) \subset \partial f(\boldsymbol{x}) \quad \boldsymbol{x} \in \mathbb{R}^n .
$$

Moreover, suppose that convex sets $\mathsf{ri}(\mathsf{dom}\, f_i)$ *have a point in common, then we also have*

$$
\partial f(\boldsymbol{x}) \subset \partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}) \quad \boldsymbol{x} \in \mathbb{R}^n .
$$

Proof. See Theorem 23.8 in [151].

Proposition B.9. *Let f be a convex function, then we have that*

$$\begin{aligned} & f \text{ is Lipschitz continuous with constant } L > 0 \\ \Leftrightarrow \quad & \|\boldsymbol{g}(\boldsymbol{x})\| \leq L, \quad \boldsymbol{g}(\boldsymbol{x}) \in \partial f(\boldsymbol{x}) \quad \boldsymbol{x} \in \mathbb{R}^n \ . \end{aligned}$$

Proof. First assume that $\|\boldsymbol{g}(\boldsymbol{x})\| \leq L$ for all subgradients. Then, from the definition of subgradient

$$\begin{aligned} & \begin{cases} f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \boldsymbol{g}(\boldsymbol{y})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \\ f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) \end{cases} \\ \Leftrightarrow \quad & \boldsymbol{g}(\boldsymbol{y})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \leq f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \boldsymbol{g}(\boldsymbol{x})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \ . \end{aligned}$$

Then, from Cauchy-Schwarz inequality, we obtain

$$\begin{aligned} -L\|\boldsymbol{x} - \boldsymbol{y}\| &\leq -\|\boldsymbol{g}(\boldsymbol{y})\|\|\boldsymbol{x} - \boldsymbol{y}\| \\ &\leq f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \|\boldsymbol{g}(\boldsymbol{x})\|\|\boldsymbol{x} - \boldsymbol{y}\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\| \ . \end{aligned}$$

Now assume that g is L-Lipschitz continuous. Then, we have for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$\boldsymbol{g}(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) \leq f(\boldsymbol{y}) - f(\boldsymbol{x}) \leq L\|\boldsymbol{y} - \boldsymbol{x}\| \ .$$

Consider $\boldsymbol{v} = \boldsymbol{y} - \boldsymbol{x} \neq \boldsymbol{0}$, then we have that

$$\boldsymbol{g}(\boldsymbol{x})^\mathsf{T}\left(\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\right) \leq L \ .$$

Since, this must be true for any $\boldsymbol{v} \neq \boldsymbol{0}$, we directly obtain $\|\boldsymbol{g}(\boldsymbol{x})\| \leq L$.

Appendix C

Supplement for Chapter 4

C.1 Proof of Proposition 4.1

The proof below is a direct consequence of definition (2.14). It is similar to the derivation of other properties of the proximal operator that have been extensively described in the literature (see for example [15, Ch. 6]). For any $\boldsymbol{z} \in \mathbb{R}^n$, we have that

$$
\begin{aligned}
&\mathsf{D}_\mu(\boldsymbol{z}) = (1/\mu) \cdot \mathsf{prox}_r(\mu \cdot \boldsymbol{z}) \\
&= (1/\mu) \cdot \underset{\boldsymbol{x}}{\mathsf{arg\,min}} \left\{ \frac{1}{2}\|\boldsymbol{x} - \mu\boldsymbol{z}\|_2^2 + r(\boldsymbol{x}) \right\} \\
&= (1/\mu) \cdot \underset{\boldsymbol{x}}{\mathsf{arg\,min}} \left\{ \frac{1}{2}\,\|(\boldsymbol{x}/\mu) - \boldsymbol{z}\|_2^2 + (1/\mu^2) r(\boldsymbol{x}) \right\} \\
&= (1/\mu) \cdot \mu \cdot \underset{\boldsymbol{u}}{\mathsf{arg\,min}} \left\{ \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2 + (1/\mu^2) r(\mu\boldsymbol{u}) \right\} \\
&= \mathsf{prox}_{(1/\mu^2) r(\mu \cdot)}(\boldsymbol{z}),
\end{aligned}
$$

where in the second and the last lines we used the definition of the proximal operator, and in the fourth line we performed the variable change $\boldsymbol{u} = \boldsymbol{x}/\mu$.

C.2 Proof of Proposition 4.2

This result is a direct consequence of the definition of the MMSE denoiser. For any $\boldsymbol{z} \in \mathbb{R}^n$, we have that

$$\begin{aligned}
&\mathsf{D}_\mu(\boldsymbol{z}) = (1/\mu) \cdot \mathsf{D}(\mu \cdot \boldsymbol{z}) \\
&= (1/\mu) \cdot \frac{\int_{\mathbb{R}^n} \boldsymbol{x} \phi_1(\boldsymbol{x} - \mu \boldsymbol{z}) p_{\boldsymbol{x}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}}{\int_{\mathbb{R}^n} \phi_1(\boldsymbol{x} - \mu \boldsymbol{z}) p_{\boldsymbol{x}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}} \\
&= (1/\mu) \cdot \frac{\int_{\mathbb{R}^n} \boldsymbol{x} \phi_{(1/\mu^2)}(\boldsymbol{x}/\mu - \boldsymbol{z}) p_{\boldsymbol{x}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}}{\int_{\mathbb{R}^n} \phi_{(1/\mu^2)}(\boldsymbol{x}/\mu - \boldsymbol{z}) p_{\boldsymbol{x}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}} \\
&= (1/\mu) \cdot \frac{\int_{\mathbb{R}^n} (\mu \boldsymbol{u}) \phi_{(1/\mu^2)}(\boldsymbol{u} - \boldsymbol{z}) p_{\boldsymbol{x}}(\mu \boldsymbol{u}) \, \mathrm{d}\boldsymbol{u}}{\int_{\mathbb{R}^n} \phi_{(1/\mu^2)}(\boldsymbol{u} - \boldsymbol{z}) p_{\boldsymbol{x}}(\mu \boldsymbol{u}) \, \mathrm{d}\boldsymbol{u}},
\end{aligned}$$

where we defined the probability density function of AWGN of variance $\nu > 0$ as

$$\phi_\nu(\boldsymbol{x}) \coloneqq \frac{1}{\sqrt{2\pi\nu}} \mathrm{e}^{-\frac{\|\boldsymbol{x}\|^2}{2\nu}}.$$

The final line corresponds to the MMSE estimate of a random variable $\boldsymbol{u} \sim p_{\boldsymbol{x}}(\mu \cdot)$ from AWGN of variance $1/\mu^2$.

C.3 Proof of Proposition 4.3

It has already been shown that for any continuous denoiser both PnP-ADMM and PnP-PGM have the same set of fixed points [168]. Consider any fixed point $\boldsymbol{x}$ of PnP-PGM

$$\boldsymbol{x} = \mathsf{D}_{\mu}(\boldsymbol{x} - \gamma \nabla g(\boldsymbol{x})) \quad \Leftrightarrow \quad \mu\boldsymbol{x} = \mathsf{D}(\mu\boldsymbol{x} - \gamma\mu \nabla g(\boldsymbol{x}))$$

$$\Leftrightarrow \begin{cases} \mu\boldsymbol{x} = \mathsf{D}(\mu\boldsymbol{x} - \boldsymbol{z}) \\ \boldsymbol{z} = \gamma\mu \nabla g(\boldsymbol{x}) \end{cases} \Leftrightarrow \begin{cases} \mu\boldsymbol{x} = \mathsf{D}(\mu\boldsymbol{x} - \boldsymbol{z}) \\ \mu\boldsymbol{x} = \mathsf{prox}_{(\gamma\mu^2) g(\cdot/\mu)}(\mu\boldsymbol{x} + \boldsymbol{z}), \end{cases}$$

which directly leads to the result. To see the last equivalence, assume that g is a smooth and convex function and note that

$$\begin{aligned} \mu\boldsymbol{x} &= \mathsf{prox}_{(\mu^2\gamma) g(\cdot/\mu)}\left(\mu\boldsymbol{x} + \boldsymbol{z}\right) \\ &= \underset{\boldsymbol{u}}{\mathsf{arg\,min}} \left\{ \frac{1}{2} \left\| \boldsymbol{u} - (\mu\boldsymbol{x} + \boldsymbol{z}) \right\|_2^2 + (\gamma\mu^2) g(\boldsymbol{u}/\mu) \right\} \\ &\Leftrightarrow \quad \mu\boldsymbol{x} - (\mu\boldsymbol{x} + \boldsymbol{z}) + (\gamma\mu^2) \cdot (1/\mu) \cdot \nabla g(\mu\boldsymbol{x}/\mu) = \boldsymbol{0} \\ &\Leftrightarrow \quad \boldsymbol{z} = \gamma\mu \nabla g(\boldsymbol{x}), \end{aligned}$$

where we used the optimality conditions.

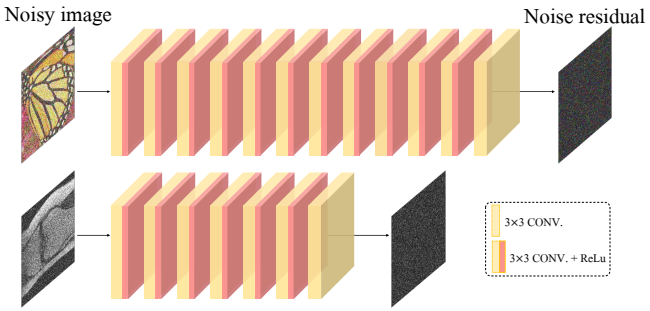

Figure C.1: The architecture of two variants of DnCNN* we use in our simulations. DnCNN* (top) is applied on natural images, and DnCNN* (bottom) is applied on the medical knee images. Both neural nets are trained to predict the AWGN from the input. The final desired denoiser D is obtained by simply subtracting the predicted noise from the input $\mathsf{D}(\boldsymbol{x}) = \boldsymbol{x} - \text{DnCNN}^*(\boldsymbol{x})$.

C.4 Architecture of the DnCNN* denoiser

Two variants of the residual DnCNN*, shown in Figure C.1, are used in our simulations. The DnCNN* of 12 convolutional layers is used for natural images. The DnCNN* of 7 layers from [169] is used for the knee images from the NYU fastMRI dataset [56]. The latter has a bounded Lipschitz constant $L = 2$, providing a necessary but not sufficient condition for D to be a nonexpansive denoiser. As discussed in [169], the Lipschitz constant is controlled via spectral-normalization [160].

Appendix D

Supplement for Chapter 5

D.1 MMSE Denoising as Proximal Operator

The relationship between MMSE estimation and regularized optimization has been established by Gribonval in [70] and has been discussed in other contexts [71, 96]. Our contribution is to formally connect this relationship to PnP algorithms, leading to their new interpretation for MMSE denoisers.

It is well known that the estimator (5.5) can be compactly expressed using the *Tweedie's formula*

$$\mathsf{D}_\sigma(\boldsymbol{z}) = \boldsymbol{z} - \sigma^2 \nabla r_\sigma(\boldsymbol{z}) \quad \text{with} \quad r_\sigma(\boldsymbol{z}) = -\mathsf{log}(p_{\boldsymbol{z}}(\boldsymbol{z})), \tag{D.1}$$

which can be obtained by differentiating (5.5) using the expression for the probability distribution

$$p_{\boldsymbol{z}}(\boldsymbol{z}) = (p_{\boldsymbol{x}} * \phi_\sigma)(\boldsymbol{z}) = \int_{\mathbb{R}^n} \phi_\sigma(\boldsymbol{z} - \boldsymbol{x}) p_{\boldsymbol{x}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{D.2}$$

where

$$\phi_\sigma(\boldsymbol{x}) \coloneqq \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}\right).$$

Since ϕ_σ is infinitely differentiable, so are $p_{\boldsymbol{z}}$ and D_σ. By differentiating D_σ, one can show that the Jacobian of D_σ is positive definite (see Lemma 2 in [70])

$$\mathsf{J}\mathsf{D}_\sigma(\boldsymbol{z}) = \mathbf{I} - \sigma^2 \mathsf{H} r_\sigma(\boldsymbol{z}) \succ 0, \quad \boldsymbol{z} \in \mathbb{R}^n, \tag{D.3}$$

where $\mathsf{H}r_\sigma$ denotes the Hessian matrix of the function r_σ. Finally, Assumption 5.1 also implies that D_σ is a *one-to-one* mapping from $\mathbb{R}^n$ to $\mathcal{X} = \mathsf{Im}(\mathsf{D}_\sigma)$, which means that $\mathsf{D}^{-1} : \mathcal{X} \to \mathbb{R}^n$ is well defined and also infinitely differentiable over $\mathcal{X}$ (see Lemma 1 in [70]). This directly implies that the regularizer r in (5.6) is also infinitely differentiable for any $\boldsymbol{x} \in \mathcal{X}$.

We will now show that

$$\begin{aligned} \mathsf{D}_\sigma(\boldsymbol{z}) &= \mathsf{prox}_{\gamma r}(\boldsymbol{z}) \\ &= \operatorname*{\mathsf{arg\,min}}_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|^2 + \gamma r(\boldsymbol{x}) \right\} \end{aligned} \tag{D.4}$$

where r is a (possibly nonconvex) function defined in (5.6). Our aim is to show that $\boldsymbol{u}^* = \boldsymbol{z}$ is the unique stationary point and global minimizer of

$$\varphi(\boldsymbol{u}) \coloneqq \frac{1}{2}\|\mathsf{D}_\sigma(\boldsymbol{u}) - \boldsymbol{z}\|^2 + \gamma r(\mathsf{D}_\sigma(\boldsymbol{u})), \quad \boldsymbol{u} \in \mathbb{R}^n.$$

By using the definition of r in (5.6) and the Tweedie's formula (D.1), we get

$$\begin{aligned} \varphi(\boldsymbol{u}) &= \frac{1}{2}\|\mathsf{D}_\sigma(\boldsymbol{u}) - \boldsymbol{z}\|^2 - \frac{1}{2}\|\mathsf{D}_\sigma(\boldsymbol{u}) - \boldsymbol{u}\|^2 + \sigma^2 r_\sigma(\boldsymbol{u}) \\ &= \frac{1}{2}\|\mathsf{D}_\sigma(\boldsymbol{u}) - \boldsymbol{z}\|^2 - \frac{\sigma^4}{2}\|\nabla r_\sigma(\boldsymbol{u})\|^2 + \sigma^2 r_\sigma(\boldsymbol{u}). \end{aligned}$$

The gradient of φ is then given by

$$\begin{aligned}
&\nabla\varphi(\boldsymbol{z}) \\
&= [\mathsf{J}\mathsf{D}_\sigma(\boldsymbol{u})](\mathsf{D}_\sigma(\boldsymbol{u}) - \boldsymbol{z}) + \sigma^2[\mathbf{I} - \sigma^2\mathsf{H}r_\sigma(\boldsymbol{u})]\nabla r_\sigma(\boldsymbol{u}) \\
&= [\mathsf{J}\mathsf{D}_\sigma(\boldsymbol{u})]\left(\mathsf{D}_\sigma(\boldsymbol{u}) + \sigma^2\nabla r_\sigma(\boldsymbol{u}) - \boldsymbol{z}\right) \\
&= [\mathsf{J}\mathsf{D}_\sigma(\boldsymbol{u})](\boldsymbol{u} - \boldsymbol{z}),
\end{aligned}$$

where we used (D.3) in the second line and (D.1) in the third line. Now consider a scalar function $q(\nu) = \varphi(\boldsymbol{z} + \nu\boldsymbol{u})$ and its derivative

$$q'(\nu) = \nabla\varphi(\boldsymbol{z} + \nu\boldsymbol{u})^\mathsf{T}\boldsymbol{u} = \nu\boldsymbol{u}^\mathsf{T}[\mathsf{J}\mathsf{D}_\sigma(\boldsymbol{z} + \nu\boldsymbol{u})]\boldsymbol{u}.$$

From the positive definiteness of the Jacobian (D.3), we have $q'(\nu) < 0$ and $q'(\nu) > 0$ for $\nu < 0$ and $\nu > 0$, respectively. This implies that $\nu = 0$ is the global minimizer of q. Since $\boldsymbol{u} \in \mathbb{R}^n$ is an arbitrary vector, we have that φ has no stationary point beyond $\boldsymbol{u}^* = \boldsymbol{z}$ and that $\varphi(\boldsymbol{z}) < \varphi(\boldsymbol{u})$ for any $\boldsymbol{u} \neq \boldsymbol{z}$.

D.2 Convergence Analysis

Prior work has analyzed the convergence of PnP algorithms for contractive, nonexpansive, or bounded denoisers [36, 67, 158, 166, 168, 177]. Our analysis extends the prior work on PnP by analyzing convergence for MMSE denoisers without any assumptions on convexity of g and r or on nonexpansiveness of D_σ. We adopt *majorization-minimization (MM) strategy* widely used in nonconvex optimization [16, 47, 105, 122, 148].

Consider the following approximation of f at $\boldsymbol{s} \in \mathbb{R}^n$

$$\begin{aligned}\mu(\boldsymbol{x}, \boldsymbol{s}) &= g(\boldsymbol{s}) + \nabla g(\boldsymbol{s})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{s}) + \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{s}\|^2 + r(\boldsymbol{x}) \\ &= g(\boldsymbol{s}) + \frac{1}{2\gamma}\|\boldsymbol{x} - (\boldsymbol{s} - \gamma\nabla g(\boldsymbol{s})\|^2 + r(\boldsymbol{x}) - \frac{\gamma}{2}\|\nabla g(\boldsymbol{s})\|^2.\end{aligned}$$

Assumption 5.2 implies that for any $0 < \gamma \leq 1/L$, we have

$$\mu(\boldsymbol{x}, \boldsymbol{s}) \geq f(\boldsymbol{x}) \quad \text{and} \quad \mu(\boldsymbol{s}, \boldsymbol{s}) = f(\boldsymbol{s}), \quad \boldsymbol{x}, \boldsymbol{s} \in \mathbb{R}^n. \tag{D.5}$$

We express (5.3) in the MM format

$$\boldsymbol{x}^k = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\mathsf{arg\,min}}\, \mu(\boldsymbol{x}, \boldsymbol{x}^{k-1}) = \mathsf{D}_\sigma(\boldsymbol{x}^{k-1} - \gamma\nabla g(\boldsymbol{x}^{k-1})), \tag{D.6}$$

where from Appendix D.1, we know that $\mathsf{D}_\sigma = \mathsf{prox}_{\gamma r}$. Therefore, from (D.5) and (D.6), we directly have that

$$f(\boldsymbol{x}^k) \leq \mu(\boldsymbol{x}^k, \boldsymbol{x}^{k-1}) \leq \mu(\boldsymbol{x}^{k-1}, \boldsymbol{x}^{k-1}) = f(\boldsymbol{x}^{k-1}).$$

From Assumption 5.3, we know that f is bounded from below; therefore, the *monotone convergence theorem* implies that the sequence $\{f(\boldsymbol{x}^k)\}_{k \geq 0}$ converges.

Consider the residual function v between μ and f

$$\begin{aligned}p(\boldsymbol{x}) &= \mu(\boldsymbol{x}, \boldsymbol{s}) - f(\boldsymbol{x}) \\ &= g(\boldsymbol{s}) + \nabla g(\boldsymbol{s})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{s}) + \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{s}\|^2 - g(\boldsymbol{x}).\end{aligned}$$

The definition of v implies that $p(\boldsymbol{s}) = 0$ and $\nabla p(\boldsymbol{s}) = \mathbf{0}$. Additionally, we have for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$\begin{aligned}\|\nabla p(\boldsymbol{x}) - \nabla p(\boldsymbol{y})\| &= \|(1/\gamma)(\boldsymbol{x} - \boldsymbol{y}) - (\nabla g(\boldsymbol{x}) - \nabla g(\boldsymbol{y}))\| \\ &\leq (1/\gamma)\|\boldsymbol{x} - \boldsymbol{y}\| + \|\nabla g(\boldsymbol{x}) - \nabla g(\boldsymbol{y})\| \\ &\leq (1/\gamma + L)\|\boldsymbol{x} - \boldsymbol{y}\| \leq (2/\gamma)\|\boldsymbol{x} - \boldsymbol{y}\|,\end{aligned}$$

where we used $0 < \gamma \leq 1/L$. The last inequality implies that ∇v is Lipschitz continuous with constant $2/\gamma$.

Denote by f^* the infimum of f and by

$$p_k(\boldsymbol{x}) = \mu(\boldsymbol{x}, \boldsymbol{x}^{k-1}) - f(\boldsymbol{x}) \geq 0, \quad \boldsymbol{x} \in \mathbb{R}^n,$$

the residual at iteration $k \geq 1$. Then,

$$\begin{aligned}& p_k(\boldsymbol{x}^k) = \mu(\boldsymbol{x}^k, \boldsymbol{x}^{k-1}) - f(\boldsymbol{x}^k) \leq f(\boldsymbol{x}^{k-1}) - f(\boldsymbol{x}^k) \\ \Rightarrow \quad & \sum_{t=1}^{\infty} p_k(\boldsymbol{x}^k) \leq (f(\boldsymbol{x}^0) - f^*),\end{aligned}$$

where we used the fact that $f^* \leq \mathsf{lim}_{k\to\infty} f(\boldsymbol{x}^k)$. This implies that $p_k(\boldsymbol{x}^k) \to 0$ as $k \to \infty$.

Since ∇p_k is $(2/\gamma)$-Lipschitz continuous, we have that

$$\begin{aligned}& \boldsymbol{u} \coloneqq \boldsymbol{x}^k - \frac{\gamma}{2}\nabla p_k(\boldsymbol{x}^k) \\ \Rightarrow \quad & p_k(\boldsymbol{u}) \leq p_k(\boldsymbol{x}^k) - \frac{\gamma}{4}\|\nabla p_k(\boldsymbol{x}^k)\|^2.\end{aligned}$$

Since $p_k(\boldsymbol{x}) \geq 0$, for all $\boldsymbol{x} \in \mathbb{R}^n$, we have

$$\|\nabla p_k(\boldsymbol{x}^k)\|^2 \leq \frac{4}{\gamma}(p_k(\boldsymbol{x}^k) - p_k(\boldsymbol{u})) \leq \frac{4}{\gamma} p_k(\boldsymbol{x}^k) \to 0,$$

as $k \to \infty$.

Finally, consider the gradient of f at $\boldsymbol{x}^k \in \mathcal{X} = \mathsf{Im}(\mathsf{D}_\sigma)$

$$\|\nabla f(\boldsymbol{x}^k)\| = \|\nabla_{\boldsymbol{x}} \mu(\boldsymbol{x}^k, \boldsymbol{x}^{k-1}) - \nabla p_k(\boldsymbol{x}^k)\| = \|\nabla p_k(\boldsymbol{x}^k)\| \to 0,$$

as $k \to \infty$, where we used the fact that $\boldsymbol{x}^k$ is the minimizer of $\mu(\boldsymbol{x}, \boldsymbol{x}^{k-1})$. This concludes the proof.

Appendix E

Supplement for Chapter 6

We adopt monotone operator theory [12, 157] for a unified analysis of IPA. The preliminary results of monotone operator theory related to our proofs are summarized in Section B. In Section E.1, we present the convergence analysis of IPA. In Appendix E.2, we analyze the convergence of the algorithm for strongly convex data-fidelity terms and contractive denoisers. In Section E.3, we discuss interpretation of IPA's fixed-points from the perspective of monotone operator theory. For completeness, in Section E.4, we discuss the convergence results for traditional PnP-ADMM [158]. In Section E.5, we summarize the major similarities and differences of variations of PnP and RED algorithms. In Section E.6, we provide technical details and additional validation. For the sake of simplicity, we use $\|\cdot\|$ to denote the standard ℓ_2-norm in $\mathbb{R}^n$. We will also use $\mathsf{D}(\cdot)$ instead of $\mathsf{D}_\sigma(\cdot)$ to denote the denoiser, thus dropping the explicit notation for σ.

E.1 Convergence Analysis of IPA

In this section, we present one of the main results in this paper, namely the convergence analysis of IPA. A fixed-point convergence of averaged operators is well-known under the name of Krasnosel'skii-Mann theorem (see Section 5.2 in [12]) and was recently applied to the analysis of PnP-SGD [168]. Additionally, PnP-ADMM was analyzed for strongly convex data-fidelity terms g and contractive residual denoisers R_σ [158]. Our analysis extends these results to IPA by providing an explicit upper bound on the convergence. In Appendix E.1.1, we present the main steps of the proof, while in Appendix E.1.2 we prove two technical lemmas useful for our analysis.

E.1.1 Proof of Theorem 6.1

Appendix E.3.3 establishes that S defined in (6.6) is firmly nonexpansive. Consider any $\boldsymbol{v}^* \in \mathsf{zer}(\mathsf{S})$ and any $\boldsymbol{v} \in \mathbb{R}^n$, then we have

$$\begin{aligned} &\|\boldsymbol{v} - \boldsymbol{v}^* - \mathsf{S}\boldsymbol{v}\|^2 \\ &= \|\boldsymbol{v} - \boldsymbol{v}^*\| - 2(\mathsf{S}\boldsymbol{v} - \mathsf{S}\boldsymbol{v}^*)^\mathsf{T}(\boldsymbol{v} - \boldsymbol{v}^*) + \|\mathsf{S}\boldsymbol{v}\|^2 \\ &\leq \|\boldsymbol{v} - \boldsymbol{v}^*\|^2 - \|\mathsf{S}\boldsymbol{v}\|^2 \,, \end{aligned} \tag{E.1}$$

where we used the firm nonexpansiveness of S and $\mathsf{S}\boldsymbol{v}^* = \boldsymbol{0}$. The direct consequence of (E.1) is that

$$\|\boldsymbol{v} - \boldsymbol{v}^* - \mathsf{S}\boldsymbol{v}\| \leq \|\boldsymbol{v} - \boldsymbol{v}^*\| \,.$$

We now consider the following two equivalent representations of IPA for some iteration $k \geq 1$

$$\begin{cases} \boldsymbol{z}^k = \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1}) \\ \boldsymbol{x}^k = \mathsf{D}(\boldsymbol{z}^k - \boldsymbol{s}^{k-1}) \\ \boldsymbol{s}^k = \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k, \end{cases} \quad \Leftrightarrow \quad \begin{cases} \boldsymbol{x}^{k-1} = \mathsf{D}(\boldsymbol{v}^{k-1}) \\ \boldsymbol{z}^k = \mathsf{G}_{i_k}(2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1}) \\ \boldsymbol{v}^k = \boldsymbol{v}^{k-1} + \boldsymbol{z}^k - \boldsymbol{x}^{k-1} \end{cases} \tag{E.2}$$

where i_k is a random variable uniformly distributed over $\{1, \ldots, b\}$, $\mathsf{G}_i = \mathsf{prox}_{\gamma g_i}$ is the proximal operator with respect to g_i, and D is the denoiser. the left and the right sides of (E.2), , simply introduce the variable $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ into the right side of (E.2) [158]. It is straightforward to verify that the right side of (E.2) can also be rewritten as

$$\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}(\boldsymbol{v}^{k-1}) \quad \text{with} \quad \mathsf{S}_{i_k} \coloneqq \mathsf{D} - \mathsf{G}_{i_k}(2\mathsf{D} - \mathsf{I}) \; . \tag{E.3}$$

Then, for any $\boldsymbol{v}^* \in \mathsf{zer}(\mathsf{S})$, we have that

$$\begin{aligned} &\|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 = \|(\boldsymbol{v}^{k-1} - \boldsymbol{v}^* - \mathsf{S}\boldsymbol{v}^{k-1}) + (\mathsf{S}\boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}\boldsymbol{v}^{k-1})\|^2 \\ &= \|\boldsymbol{v}^{k-1} - \boldsymbol{v}^* - \mathsf{S}\boldsymbol{v}^{k-1}\|^2 + 2(\mathsf{S}\boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}\boldsymbol{v}^{k-1})^\mathsf{T}(\boldsymbol{v}^{k-1} - \boldsymbol{v}^* - \mathsf{S}\boldsymbol{v}^{k-1}) + \|\mathsf{S}\boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|^2 \\ &\leq \|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|^2 - \|\mathsf{S}\boldsymbol{v}^{k-1}\|^2 + 2\|\mathsf{S}\boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\| + \|\mathsf{S}\boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|^2 \\ &\leq \|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|^2 - \|\mathsf{S}\boldsymbol{v}^{k-1}\|^2 + 2(R + 2\gamma L)\|\mathsf{S}\boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\| + \|\mathsf{S}\boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|^2, \end{aligned}$$

where in the first inequality we used Cauchy-Schwarz and (E.1), and in the second inequality we used Lemma E.2 in Appendix E.1.2. By taking the conditional expectation on both sides, invoking Lemma E.1 in Appendix E.1.2, and rearranging the terms, we get

$$\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2 \leq \|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|^2 - \mathbb{E}\left[\|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 \mid \boldsymbol{v}^{k-1}\right] + 4\gamma LR + 12\gamma^2 L^2 \; .$$

Hence, by averaging over $t \geq 1$ iterations and taking the total expectation, we obtain

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t}\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2\right] \leq \frac{(R+2\gamma L)^2}{t} + 4\gamma LR + 12\gamma^2 L^2 \ .$$

The final result is obtained by noting that

$$4\gamma LR + 12\gamma^2 L^2 \leq \mathsf{max}\{\gamma, \gamma^2\}(4LR + 12L^2) \ .$$

E.1.2 Lemmas Useful for the Proof of Theorem 6.1

This section presents two technical lemmas used in our analysis in Appendix E.1.1.

Lemma E.1. *Assume that Assumptions 6.1-6.3 hold and let i_k be a uniform random variable over $\{1, \ldots, b\}$. Then, we have that*

$$\mathbb{E}\left[\|\mathsf{S}_{i_k}\boldsymbol{v} - \mathsf{S}\boldsymbol{v}\|^2\right] \leq 4\gamma^2 L^2, \quad \boldsymbol{v} \in \mathbb{R}^n \ .$$

Proof. Let $\boldsymbol{z}_i = \mathsf{G}_i(\boldsymbol{x})$ and $\boldsymbol{z} = \mathsf{G}(\boldsymbol{x})$ for any $1 \leq i \leq b$ and $\boldsymbol{x} \in \mathbb{R}^n$. From the optimality conditions for each proximal operator

$$\mathsf{G}_i\boldsymbol{x} = \mathsf{prox}_{\gamma g_i}(\boldsymbol{x}) = \boldsymbol{x} - \gamma \boldsymbol{g}_i(\boldsymbol{z}_i), \quad \boldsymbol{g}_i(\boldsymbol{z}_i) \in \partial g_i(\boldsymbol{z}_i)$$

and

$$\mathsf{G}\boldsymbol{x} = \mathsf{prox}_{\gamma g}(\boldsymbol{x}) = \boldsymbol{x} - \gamma \boldsymbol{g}(\boldsymbol{z})$$

such that

$$\boldsymbol{g}(\boldsymbol{z}) = \frac{1}{b}\sum_{i=1}^{b}\boldsymbol{g}_i(\boldsymbol{z}) \in \partial g(\boldsymbol{z}) \ ,$$

where we used Proposition B.8 in Appendix B.2. By using the bound on all the subgradients (due to Assumption 6.1 and Proposition B.9 in Appendix B.2), we obtain

$$\|\mathsf{G}_i(\boldsymbol{x}) - \mathsf{G}(\boldsymbol{x})\| = \|\mathsf{prox}_{\gamma g_i}(\boldsymbol{x}) - \mathsf{prox}_{\gamma g}(\boldsymbol{x})\| = \gamma\|\boldsymbol{g}_i(\boldsymbol{z}_i) - \boldsymbol{g}(\boldsymbol{z})\| \leq 2\gamma L \;,$$

where $L > 0$ is the Lipschitz constant of all g_is and g. This inequality directly implies that

$$\|\mathsf{S}\boldsymbol{v} - \mathsf{S}_i\boldsymbol{v}\| = \|\mathsf{G}(2\mathsf{D}\boldsymbol{v} - \boldsymbol{v}) - \mathsf{G}_i(2\mathsf{D}\boldsymbol{v} - \boldsymbol{v})\| \leq 2\gamma L \;.$$

Since, this inequality holds for every i, it also holds in expectation.

Lemma E.2. *Assume that Assumptions 6.1-6.3 hold and let the sequence* $\{\boldsymbol{v}^k\}$ *be generated via the iteration* (E.3)*. Then, for any* $k \geq 1$*, we have that*

$$\|\boldsymbol{v}^k - \boldsymbol{v}^*\| \leq (R + 2\gamma L) \quad \textit{for all} \quad \boldsymbol{v}^* \in \mathsf{zer}(\mathsf{S}) \;.$$

Proof. The optimality of the proximal operator in (E.3) implies that there exists $\boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \in \partial g_{i_k}(\boldsymbol{z}^k)$ such that

$$\begin{aligned} & \boldsymbol{z}^k = \mathsf{G}_{i_k}(2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1}) \\ \Leftrightarrow \quad & 2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1} - \boldsymbol{z}^k = \gamma\boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \;. \end{aligned}$$

By applying $\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}(\boldsymbol{v}^{k-1}) = \boldsymbol{v}^{k-1} + \boldsymbol{z}^k - \boldsymbol{x}^{k-1}$ to the equality above, we obtain

$$\boldsymbol{x}^{k-1} - \boldsymbol{v}^k = \gamma\boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \quad \Leftrightarrow \quad \boldsymbol{v}^k = \boldsymbol{x}^{k-1} - \gamma\boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \;.$$

Additionally, for any $\boldsymbol{v}^* \in \mathsf{zer}(\mathsf{S})$ and $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*)$, we have that

$$\mathsf{S}(\boldsymbol{v}^*) = \mathsf{D}(\boldsymbol{v}^*) - \mathsf{G}(2\mathsf{D}(\boldsymbol{v}^*) - \boldsymbol{v}^*) = \boldsymbol{x}^* - \mathsf{G}(2\boldsymbol{x}^* - \boldsymbol{v}^*) = \mathbf{0}$$
$$\Rightarrow \quad \boldsymbol{x}^* - \boldsymbol{v}^* = \gamma \boldsymbol{g}(\boldsymbol{x}^*) \quad \text{for some} \quad \boldsymbol{g}(\boldsymbol{x}^*) \in \partial g(\boldsymbol{x}^*) \; .$$

Thus, by using Assumption 6.3 and the bounds on all the subgradients (due to Assumption 6.1 and Proposition B.9 in Appendix B.2), we obtain

$$\|\boldsymbol{v}^k - \boldsymbol{v}^*\| = \|\boldsymbol{x}^{k-1} - \gamma \boldsymbol{g}_{i_k}(\boldsymbol{z}^k) - \boldsymbol{x}^* - \gamma \boldsymbol{g}(\boldsymbol{x}^*)\| \leq \|\boldsymbol{x}^{t-1} - \boldsymbol{x}^*\| + 2\gamma L \leq (R + 2\gamma L) \; .$$

E.2 Analysis of IPA for Strongly Convex Functions

In this section, we perform analysis of IPA under a different set of assumptions, namely under the assumptions adopted in [158].

Assumption E.1. Each g_i is proper, closed, strongly convex with constant $M_i > 0$, and Lipschitz continuous with constant $L_i > 0$. We define the smallest strong convexity constant as $M = \mathsf{min}\{M_1, \ldots, M_b\}$ and the largest Lipschitz constant as $L = \mathsf{max}\{L_1, \ldots, L_b\}$.

This assumption further restricts Assumption 6.1 to strongly convex functions.

Assumption E.2. The residual $\mathsf{R}_\sigma \coloneqq \mathsf{I} - \mathsf{D}_\sigma$ of the denoiser D_σ is a contraction. It thus satisfies

$$\|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\| \leq \epsilon \|\boldsymbol{x} - \boldsymbol{y}\| \; ,$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ for some constant $0 < \epsilon < 1$.

This assumption replaces Assumption 6.2 by assuming that the residual of the denoiser is a contraction. Note that this can be practically imposed on deep neural net denoisers via spectral normalization [133]. We can then state the following.

Theorem E.1. *Run IPA for $t \geq 1$ iterations with random i.i.d. block selection under Assumptions 6.3-E.2 using a fixed penalty parameter $\gamma > 0$. Then, the iterates of IPA satisfy*

$$\mathbb{E}\left[\|\boldsymbol{x}^k - \boldsymbol{x}^*\|\right] \leq \eta^k(2R + 4\gamma L) + \frac{4\gamma L}{1-\eta}, \quad 0 < \eta < 1 \, .$$

Proof. It was shown in Theorem 2 of [158] that under Assumptions E.1 and E.2, we have that

$$\|(\mathsf{I} - \mathsf{S})\boldsymbol{x} - (\mathsf{I} - \mathsf{S})\boldsymbol{y}\| \leq \eta \|\boldsymbol{x} - \boldsymbol{y}\| \tag{E.4}$$

with

$$\eta \coloneqq \left(\frac{1 + \epsilon + \epsilon\gamma M + 2\epsilon^2\gamma M}{1 + \gamma M + 2\epsilon\gamma M}\right) \, ,$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, where S is given in (6.6). Hence, when

$$\frac{\epsilon}{\gamma M(1 + \epsilon - 2\epsilon^2)} < 1 \, ,$$

the operator $(\mathsf{I} - \mathsf{S})$ is a contraction. Using the reasoning in Appendix E.1, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ can be written as

$$\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}(\boldsymbol{v}^{k-1}) \quad \text{with} \quad \mathsf{S}_{i_k} \coloneqq \mathsf{D} - \mathsf{G}_{i_k}(2\mathsf{D} - \mathsf{I}) \, . \tag{E.5}$$

Then, for any $\boldsymbol{v}^* \in \mathsf{zer}(\mathsf{S})$, we have that

$$
\begin{aligned}
&\|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 \\
&= \|(\mathsf{I}-\mathsf{S})\boldsymbol{v}^{k-1} - (\mathsf{I}-\mathsf{S})\boldsymbol{v}^*\|^2 + 2((\mathsf{I}-\mathsf{S})\boldsymbol{v}^{k-1} - (\mathsf{I}-\mathsf{S})\boldsymbol{v}^*)^{\mathsf{T}}((\mathsf{I}-\mathsf{S}_{i_k})\boldsymbol{v}^{k-1} - \\
&\quad (\mathsf{I}-\mathsf{S})\boldsymbol{v}^{k-1}) + \|(\mathsf{I}-\mathsf{S}_{i_k})\boldsymbol{v}^{k-1} - (\mathsf{I}-\mathsf{S})\boldsymbol{v}^{k-1}\|^2 \\
&\leq \eta^2\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|^2 + 2\eta\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|\|\mathsf{S}_{i_k}\boldsymbol{v}^{k-1} - \mathsf{S}\boldsymbol{v}^{k-1}\| + \|\mathsf{S}_{i_k}\boldsymbol{v}^{k-1} - \mathsf{S}\boldsymbol{v}^{k-1}\|^2 \ ,
\end{aligned}
$$

where we used the Cauchy-Schwarz inequality and the fact that $(\mathsf{I}-\mathsf{S})$ is η-contractive. By taking the conditional expectation on both sides, invoking Lemma E.1 in Appendix E.1.2, and completing the square, we get

$$
\mathbb{E}\left[\|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 | \boldsymbol{v}^{k-1}\right] \leq \left(\eta\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\| + 2\gamma L\right)^2 \ .
$$

Then, by applying the Jensen inequality and taking the total expectation, we get

$$
\mathbb{E}\left[\|\boldsymbol{v}^k - \boldsymbol{v}^*\|\right] \leq \eta\mathbb{E}\left[\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|\right] + 2\gamma L \ .
$$

By iterating this result and invoking Lemma E.2 from Appendix E.1.2, we obtain

$$
\mathbb{E}\left[\|\boldsymbol{v}^k - \boldsymbol{v}^*\|\right] \leq \eta^k(R + 2\gamma L) + (2\gamma L)/(1-\eta) \ .
$$

Finally by using the nonexpansiveness of $(1/(1+\epsilon))\mathsf{D}$ (see Lemma 9 in [158]) and the fact that $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*)$, we obtain

$$
\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{x}^k - \boldsymbol{x}^*\|\right] &\leq (1+\epsilon)\left[\eta^k(R + 2\gamma L) + \frac{2\gamma L}{1-\eta}\right] \\
&\leq \eta^k(2R + 4\gamma L) + \frac{4\gamma L}{1-\eta} \ .
\end{aligned}
$$

This concludes the proof.

E.3 Fixed Point Interpretation

Fixed points of PnP algorithms have been extensively discussed in the recent literature [31, 128, 158]. Our goal in this section is to revisit this topic in a way that leads to a more intuitive equilibrium interpretation of PnP. Our formulation has been inspired from the classical interpretation of ADMM as an algorithm for computing a zero of a sum of two monotone operators [53].

E.3.1 Equilibrium Points of PnP Algorithms

It is known that a fixed point $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{s}^*)$ of PnP-ADMM (and of all PnP algorithms [128]) satisfies

$$\boldsymbol{x}^* = \mathsf{G}(\boldsymbol{x}^* + \boldsymbol{s}^*) \quad \text{and} \quad \boldsymbol{x}^* = \mathsf{D}(\boldsymbol{x}^* - \boldsymbol{s}^*) \,, \tag{E.6}$$

with $\boldsymbol{x}^* = \boldsymbol{z}^*$, where $\mathsf{G} = \mathsf{prox}_{\gamma g}$. Consider the *inverse* of D at $\boldsymbol{x} \in \mathbb{R}^n$, which is a set-valued operator $\mathsf{D}^{-1}(\boldsymbol{x}) \coloneqq \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}_\sigma(\boldsymbol{z})\}$. Note that the inverse operator exists even when D is not a bijection (see Section 2 of [157]). Then, from the definition of D^{-1} and optimality conditions of the proximal operator, we can equivalently rewrite (E.6) as follows

$$\boldsymbol{s}^* \in \gamma \partial g(\boldsymbol{x}^*) \quad \text{and} \quad -\boldsymbol{s}^* \in \mathsf{D}^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^* \,.$$

This directly leads to the following equivalent representation of PnP fixed points

$$\boldsymbol{0} \in \mathsf{T}(\boldsymbol{x}^*) \coloneqq \gamma \partial g(\boldsymbol{x}^*) + (\mathsf{D}^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^*) \,. \tag{E.7}$$

Hence, a vector $\boldsymbol{x}^*$ computed by PnP can be interpreted as an equilibrium point between two terms with $\gamma > 0$ explicitly influencing the balance.

E.3.2 Equivalence of Zeros of T and S

Define $\boldsymbol{v}^* \coloneqq \boldsymbol{z}^* - \boldsymbol{s}^*$ for a given fixed point $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{s}^*)$ of PnP-ADMM and consider the operator

$$\mathsf{S} = \mathsf{D} - \mathsf{G}(2\mathsf{D} - \mathsf{I}) \quad \text{with} \quad \mathsf{G} = \mathsf{prox}_{\gamma g} \,,$$

which was defined in (6.6). Note that from (E.6), we also have $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*)$ and $\boldsymbol{v}^* = \boldsymbol{x}^* - \boldsymbol{s}^*$ (due to $\boldsymbol{z}^* = \boldsymbol{x}^*$). We then have the following equivalence

$$\begin{aligned}
&\mathbf{0} \in \mathsf{T}(\boldsymbol{x}^*) = \gamma \partial g(\boldsymbol{x}^*) + (\mathsf{D}^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^*) \\
\Leftrightarrow \quad &\begin{cases} \boldsymbol{x}^* = \mathsf{G}(\boldsymbol{x}^* + \boldsymbol{s}^*) \\ \boldsymbol{x}^* = \mathsf{D}(\boldsymbol{x}^* - \boldsymbol{s}^*) \end{cases} \\
\Leftrightarrow \quad &\begin{cases} \boldsymbol{x}^* = \mathsf{G}(2\boldsymbol{x}^* - \boldsymbol{v}^*) \\ \boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*) \end{cases} \\
\Leftrightarrow \quad &\mathsf{S}(\boldsymbol{v}^*) = \mathsf{D}(\boldsymbol{v}^*) - \mathsf{G}(2\mathsf{D}(\boldsymbol{v}^*) - \boldsymbol{v}^*) = \mathbf{0} \,,
\end{aligned}$$

where we used the optimality conditions of the proximal operator G. Hence, the condition that $\boldsymbol{v}^* = \boldsymbol{z}^* - \boldsymbol{s}^* \in \mathsf{zer}(\mathsf{S})$ is equivalent to $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*) \in \mathsf{zer}(\mathsf{T})$.

E.3.3 Firm Nonexpansiveness of S

We finally would like to show that under Assumptions 6.1-6.3, the operator S is firmly nonexpansive. Assumption 6.2 and Proposition B.7 in Appendix B.2 imply that D and G

are firmly nonexpansive. Then, Proposition B.4 in Appendix B.1 implies that $(2\mathsf{D}-\mathsf{I})$ and $(2\mathsf{G}-\mathsf{I})$ are nonexpansive. Thus, the composition $(2\mathsf{G}-\mathsf{I})(2\mathsf{D}-\mathsf{I})$ is also nonexpansive and

$$(\mathsf{I}-\mathsf{S})=\frac{1}{2}\mathsf{I}+\frac{1}{2}(2\mathsf{G}-\mathsf{I})(2\mathsf{D}-\mathsf{I}) \tag{E.8}$$

is (1/2)-averaged. Then, Proposition B.4 in Appendix B.1 implies that S is firmly nonexpansive.

E.4 Convergence Analysis of PnP-ADMM

The following analysis has been adopted from [158]. For completeness, we summarize the key results useful for our own analysis by restating them under the assumptions in Section 6.4.

E.4.1 Equivalence between PnP-ADMM and PnP-DRS

An elegant analysis of PnP-ADMM emerges from its interpretation as the Douglas–Rachford splitting (DRS) algorithm [158]. This equivalence is well-known and has been extensively studied in the context of convex optimization [53]. Here, we restate the relationship for completeness.

Consider the following DRS (left) and ADMM (right) sequences

$$\begin{cases} \boldsymbol{x}^{k-1}=\mathsf{D}(\boldsymbol{v}^{k-1}) \\ \boldsymbol{z}^{k}=\mathsf{G}(2\boldsymbol{x}^{k-1}-\boldsymbol{v}^{k-1}) \\ \boldsymbol{v}^{k}=\boldsymbol{v}^{k-1}+\boldsymbol{z}^{k}-\boldsymbol{x}^{k-1} \end{cases} \quad \Leftrightarrow \quad \begin{cases} \boldsymbol{z}^{k}=\mathsf{G}(\boldsymbol{x}^{k-1}+\boldsymbol{s}^{k-1}) \\ \boldsymbol{x}^{k}=\mathsf{D}(\boldsymbol{z}^{k}-\boldsymbol{s}^{k-1}) \\ \boldsymbol{s}^{k}=\boldsymbol{s}^{k-1}+\boldsymbol{x}^{k}-\boldsymbol{z}^{k}, \end{cases} \tag{E.9}$$

where $\mathsf{G} \coloneqq \mathsf{prox}_{\gamma g}$ is the proximal operator and D is the denoiser. To see the equivalence between them, simply introduce the variable change $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ into DRS. Note also the DRS sequence can be equivalently written as

$$\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}(\boldsymbol{v}^{k-1}) \quad \text{with} \quad \mathsf{S} \coloneqq \mathsf{D} - \mathsf{G}(2\mathsf{D} - \mathsf{I}) \; .$$

To see this simply rearrange the terms in DRS as follows

$$\begin{aligned} \boldsymbol{v}^k &= \boldsymbol{v}^{k-1} + \mathsf{G}(2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1}) - \boldsymbol{x}^{k-1} \\ &= \boldsymbol{v}^{k-1} - \left[\mathsf{D}(\boldsymbol{v}^{k-1}) - \mathsf{G}(2\mathsf{D}(\boldsymbol{v}^{k-1}) - \boldsymbol{v}^{k-1})\right] \; . \end{aligned}$$

E.4.2 Convergence Analysis of PnP-DRS and PnP-ADMM

It was established in Appendix E.3.3 that S defined in (6.6) is firmly nonexpansive.

Consider a single iteration of DRS $\boldsymbol{v}^+ = \boldsymbol{v} - \mathsf{S}\boldsymbol{v}$. Then, for any $\boldsymbol{v}^* \in \mathsf{zer}(\mathsf{S})$, we have

$$\begin{aligned} \|\boldsymbol{v}^+ - \boldsymbol{v}^*\|^2 &= \|\boldsymbol{v} - \boldsymbol{v}^*\|^2 - 2(\mathsf{S}\boldsymbol{v} - \mathsf{S}\boldsymbol{v}^*)^{\mathsf{T}}(\boldsymbol{v} - \boldsymbol{v}^*) + \|\mathsf{S}\boldsymbol{v}\|^2 \\ &\leq \|\boldsymbol{v} - \boldsymbol{v}^*\|^2 - \|\mathsf{S}\boldsymbol{v}\|^2 \; , \end{aligned}$$

where we used $\mathsf{S}\boldsymbol{v}^* = \boldsymbol{0}$ and firm nonexpansiveness of S. By rearranging the terms, we obtain the following upper bound at iteration $k \geq 1$

$$\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2 \leq \|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|^2 - \|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 \; . \tag{E.10}$$

Table E.1: Overview of several existing PnP/RED algorithms

Algorithms	**Nonsmooth**	**Online**
PnP-ADMM [35, 158, 166, 186]	✓	✗
PnP-PGM/PnP-APGM [91, 128, 168]	✗	✗
PnP-SPGM [168]	✗	✓
RED-SD [153]	✗	✗
RED-ADMM [149, 153]	✓	✗
prDeep [129]	✓	✗
RED-PG/RED-APG [149]	✓	✗
SIMBA/On-RED [195, 196]	✗	✓
IPA (proposed)	✓	✓

By averaging the inequality (E.10) over $t \geq 1$ iterations, we obtain

$$\frac{1}{t}\sum_{k=1}^{t} \|\mathsf{S}\boldsymbol{v}^{k-1}\|^2 \leq \frac{\|\boldsymbol{v}^0 - \boldsymbol{v}^*\|^2}{t} \leq \frac{(R+2\gamma L)^2}{t}$$

where used the bound on $\|\boldsymbol{v}^0 - \boldsymbol{v}^*\| \leq (R + 2\gamma L)$ that can be easily obtained by following the steps in Lemma E.2 in Appendix E.1.2.

This result directly implies that $\|\mathsf{S}\boldsymbol{v}^t\| \to 0$ as $t \to 0$. Additionally, Krasnosel'skii-Mann theorem (see Section 5.2 in [12]) implies that $\boldsymbol{v}^t \to \mathsf{zer}(\mathsf{S})$. Then, from continuity of D, we have that $\boldsymbol{x}^t = \mathsf{D}(\boldsymbol{v}^t) \to \mathsf{zer}(\mathsf{T})$ (see also Appendix E.3.2). This completes the proof.

E.5 Variants of PnP/RED Algorithms

Several variants of PnP/RED algorithms are summarized in Table E.1, focusing on two properties (a) the ability to handle nonsmooth data-fidelity terms, and (b) the ability to

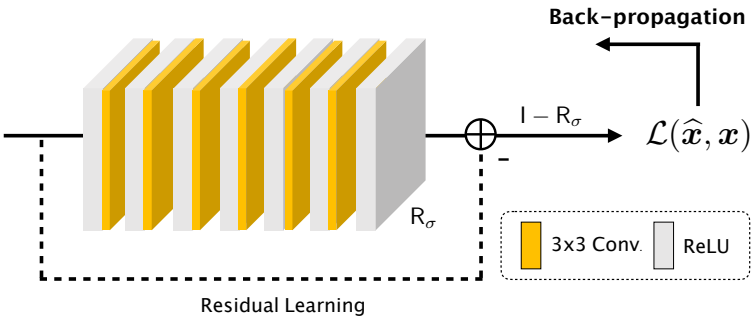

Figure E.1: Illustration of the architecture of DnCNN used in all experiments. Vectors $\widehat{\boldsymbol{x}}$ and $\boldsymbol{x}$ denote the denoised image and ground truth, respectively. The neural net is trained to remove the AWGN from its noisy input image. We also constrains the Lipschitz constant of R_σ to be smaller than 1 by using the spectral normalization technique in [160]. This provides a necessary condition for the satisfaction of Assumption 6.2.

handle online/minibatch processing of the measurements. The table highlights the way IPA complements existing work by addressing both (a) and (b).

E.6 Additional Technical Details

In this section, we present several technical details of our experiments.

E.6.1 Architecture and Training of the DnCNN Prior

Fig. E.1 illustrates the architectural details of the DnCNN prior used in our experiments. In total, the network contains 7 layers, of which the first 6 layers consist of a convolutional layer and a rectified linear unit (ReLU), while the last layer is just a convolution. A skip connection from the input to the output is implemented to enforce residual learning. The output images of the first 6 layers have 64 feature maps while that of the last layer is a single-channel image. We set all convolutional kernels to be 3×3 with stride 1, so that intermediate images have the same spatial size as the input image. We generated 11101 training examples by adding AWGN to 400 images from the BSD400 dataset [124] and extracting patches of 128×128

Table E.2: Per-iteration memory usage specification for reconstructing 512×512 images

Algorithms		IPA (60)		PnP-ADMM (300)		PnP-ADMM (600)	
Variables		size	memory	size	memory	size	memory
$\{\boldsymbol{A}_i\}$	real	$512\times512\times60$	0.23 GB	$512\times512\times300$	1.17 GB	$512\times512\times600$	2.34 GB
	imaginary	$512\times512\times60$	0.23 GB	$512\times512\times300$	1.17 GB	$512\times512\times600$	2.34 GB
$\{\boldsymbol{y}_i\}$		$512\times512\times60$	0.47 GB	$512\times512\times300$	2.34 GB	$512\times512\times600$	4.69 GB
others combined		—	0.03 GB	—	0.03 GB	—	0.03 GB
Total			**0.97 GB**		**4.72 GB**		**9.41 GB**

pixels with stride 64. We trained DnCNN to optimize the *mean squared error* by using the Adam optimizer [99].

We use the spectral normalization technique in [160] to control the global Lipschitz constant (LC) of DnCNN. In the training, we constrain the residual network R_σ to have LC smaller than 1. Since the firm non-expansiveness implies non-expansiveness, this provides a *necessary* condition for R_σ to satisfy Assumption 6.2. The training of DnCNN *with* and *without* spectral normalization takes 4 and 1.82 hours, respectively, on the same hardware. Thus, for about 2× increase in the denoiser pre-training time, one can make IPA/PnP-ADMM convergent.

E.6.2 Computation of Proximal Operators

In the CS experiments, the measurement matrix $\boldsymbol{A}$ is a random matrix, and the data-fidelity term is based on the ℓ_1-norm: $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_1$. While closed form solution of the proximal operator is inaccessible in this setting, we can efficiently approximate the proximal solution in the dual domain by using projected gradient method (PGM) [18]. Note that the closed-form solution is also unavailable for other ℓ_1-based proximal operators [18, 90]. The stopping criteria for the PGM algorithm are that either that the total iterations exceeds 200, or that the relative change between two iterates is below 1×10^{-4}.

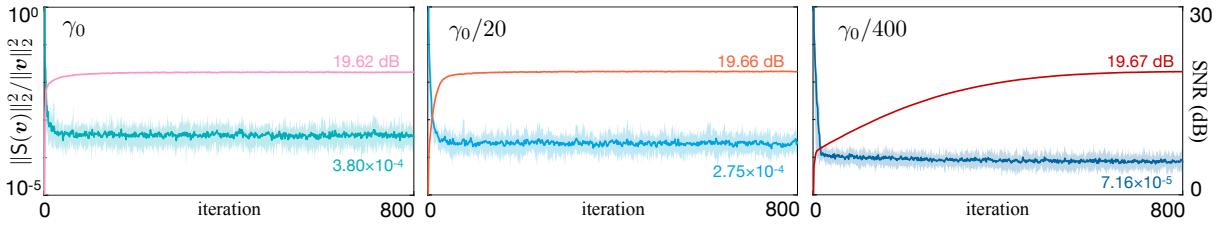

Figure E.2: Illustration of the convergence of IPA for a DnCNN prior under drastically changed γ values. The average normalized distance to $\mathsf{zer}(\mathsf{S})$ and SNR (dB) are plotted against the iteration number with the shaded areas representing the range of values attained over 12 test images. In practice, the convergence speed improves with larger values of γ. However, IPA still can achieve same level of SNR results for a wide range of γ values.

For intensity diffraction tomography (IDT), we adopted the linearized forward model developed in [109], which is based on the Fourier transform. For the i^{th} measurement, the forward model for the 2-dimensional case is described as $\boldsymbol{A}_i = \boldsymbol{F}^{\mathsf{H}}\boldsymbol{H}_i\boldsymbol{F}$, where $\boldsymbol{F}$ and $\boldsymbol{F}^{\mathsf{H}}$ denote the discrete Fourier transform and its inverse, respectively, and $\boldsymbol{H}_i$ corresponds to light transfer function of the i^{th} illumination. Under the ℓ_2-norm, we can directly derive the closed-form solution of the proximal operator in the Fourier space [2, 193].

E.6.3 Extra Details and Validations for Optical Tomography

All experiments were run on the machine equipped with an Intel Core i7 Processor that has 6 cores of 3.2 GHz and 32 GBs of DDR memory. We trained all neural nets using NVIDIA RTX 2080 GPUs. We define the SNR (dB) used in the experiments as

$$\mathrm{SNR}(\hat{\boldsymbol{x}}, \boldsymbol{x}) \triangleq \max_{a,b\in\mathbb{R}} \left\{ 20\log_{10}\left(\frac{\|\boldsymbol{x}\|}{\|\boldsymbol{x} - a\hat{\boldsymbol{x}} + b\|}\right)\right\},$$

where $\hat{\boldsymbol{x}}$ is the estimate and $\boldsymbol{x}$ is the ground truth.

For intensity diffraction tomography, we implemented an epoch-based selection rule due to the large size of data. We randomly divide the measurements (along with the corresponding

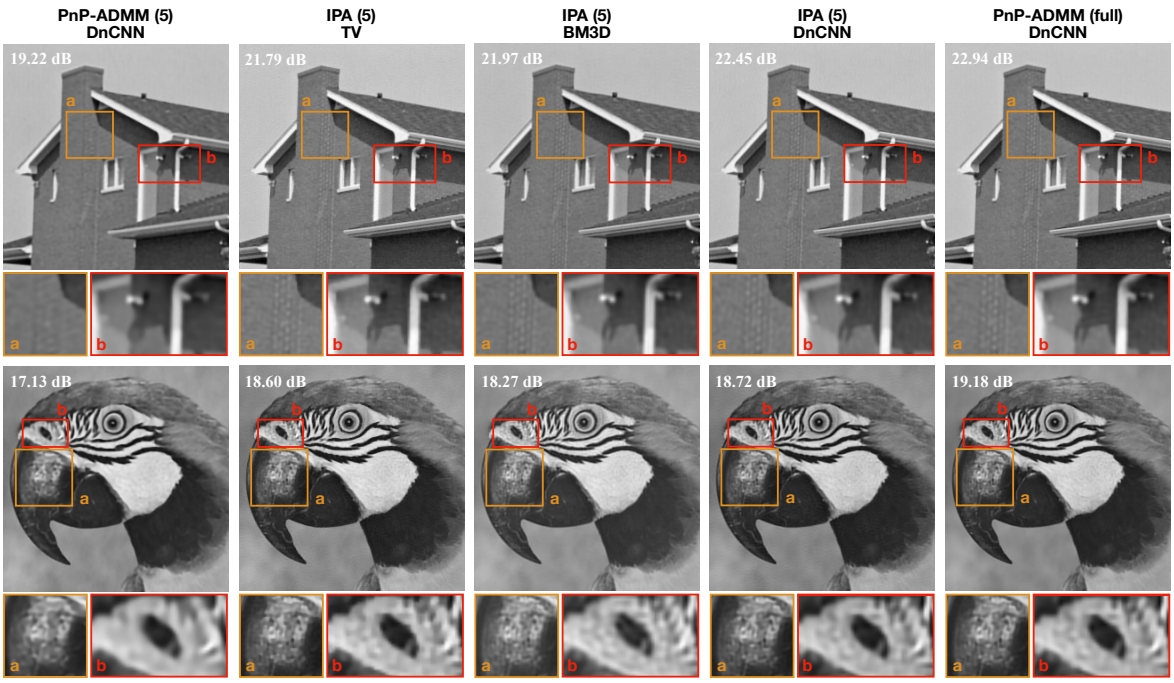

Figure E.3: Visual examples of the reconstructed House (upper) and Parrot (bottom) images by IPA and PnP-ADMM. The first and last columns correspond to PnP-ADMM under DnCNN with 5 fixed measurements and with the full 60 measurements, respectively. The second, third, and fourth column correspond to IPA with a small minibatch of size 5 under TV, BM3D, and DnCNN, respectively. Each image is labeled by its SNR (dB) with respect to the original image, and the visual difference is highlighted by the boxes underneath. Note that IPA recovers the details lost by the batch algorithm with the same computational cost and achieves the same high-quality results as the full batch algorithm.

forward operators) into non-overlapping chunks of size 60 and save these chunks on the hard drive. At every iteration, IPA loads only a single random chunk into the memory while the full-batch PnP-ADMM loads all chunks sequentially and process the full set of measurements. This leads to the lower per iteration cost and less memory usage of IPA than PnP-ADMM. Table E.2 shows extra examples of the memory usage specification for reconstructing 512×512 pixel permittivity images. These results follow the same trend observed in Table 6.2. We also conduct some extra validations that provide additional insights into IPA. In these simulations, we use images of size 254×254 pixels from *Set 12* as test examples. We assume real permittivity functions with the total number of measurement $b = 60$.

Table E.3: Optimized SNR (dB) obtained by IPA under different priors for images from *Set12* from [216]

Algorithms	PnP-ADMM (Fixed 5)	IPA (**Ours**) (Random 5 from full 60)			PnP-ADMM (Full 60)
Denoisers	DnCNN	TV	BM3D	DnCNN	DnCNN
Cameraman	15.95	17.45	17.38	18.16	18.34
House	19.22	21.79	21.97	22.45	22.94
Pepper	17.06	18.68	19.55	20.60	21.11
Starfish	18.20	19.29	20.29	21.64	22.22
Monarch	17.70	19.81	18.66	20.85	21.60
Aircraft	17.15	18.67	18.83	19.28	19.54
Parrot	17.13	18.60	18.27	18.72	19.18
Lenna	15.41	16.48	16.32	16.94	17.13
Barbara	13.63	16.00	17.53	16.58	16.85
Boat	17.98	19.35	20.21	20.95	21.34
Pirate	17.93	19.36	19.45	19.88	20.10
Couple	15.40	17.31	17.53	18.24	18.57
Average	**16.90**	**18.57**	**18.83**	**19.52**	**19.91**

Fig. E.2 illustrates the evolution of the convergence of IPA for different values of the penalty parameter. We consider three different values of $\gamma \in \{\gamma_0, \gamma_0/20, \gamma/400\}$ with $\gamma_0 = 20$. The average normalized distance $\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2/\|\boldsymbol{v}^k\|_2^2$ and SNR are plotted against the iteration number and labeled with their respective final values. The shaded areas represent the range of values attained across all test images. IPA randomly selects 5 measurements in every iteration to impose the data-consistency. Fig. E.2 complements the results in Fig 6.1 by showing the fast convergence speed in practice with larger values of γ. On the other hand, this plot further demonstrates that IPA is stable in terms of the SNR results for a wide range of γ values.

Our final simulation compares the reconstruction performance of IPA using TV, BM3D, and DnCNN. Since TV has a proximal operator, it serves as a baseline. The reconstruction performance of IPA on *House* and *Parrot* are presented in Fig. E.3, while average SNR values for additional images are presented in Table E.3. We include the results of PnP-ADMM using 5 fixed measurements and the full batch as reference. First, note the significant improvement

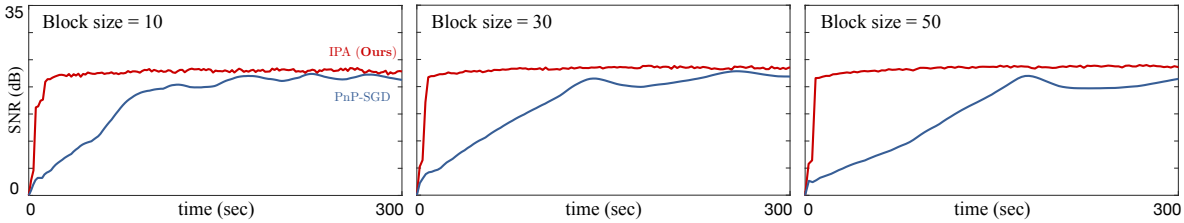

Figure E.4: Comparison between IPA and PnP-SGD for block sizes 10, 30, and 50.

of IPA over PnP-ADMM under the same computational budget. Second, using learned priors in IPA leads to better reconstruction than other priors. For instance, DnCNN outperforms TV and BM3D by 0.7 dB in SNR. Finally, the agreement between IPA and the full batch PnP-ADMM highlights the nearly optimal performance of IPA at a lower computational cost and memory usage.

E.6.4 IPA with Different Block Sizes

The block size in IPA (and other online/minibatch methods), is a free parameter that must be adjusted to achieve the best overall convergence speed. In our experiments on IDT, we used block size $= 60$ due to its excellent empirical performance. Here we provide additional experiments to show the influence of block sizes $\ll 60$ on both IPA and PnP-SGD (with Nesterov acceleration). The setup is identical to the one in the leftmost plot of Figure 2 in the Chapter 6, where $b = 300$ and $n = 512 \times 512$. Figure E.4 plots the average SNR in dB against the time in seconds for both methods under block sizes $\in \{10, 30, 50\}$, highlighting the relative advantage of IPA over PnP-SGD for smaller values of block size.

Appendix F

Supplement for Chapter 7

F.1 Properties of the Bregman Proximal Operator

The following proposition addresses the statistical interpretation of the backward step in the BPGM, that is the left Bregman Proximal Operator (BPO).

Proposition F.1. *Let the reference function h be Legendre function. Then, the left BPO in Eq.* (7.12) *can be rewritten as*

$$\mathsf{prox}_{\gamma r}^{h}(\boldsymbol{z}) = \left(\nabla h^* \circ \mathcal{E}_\gamma \circ \nabla h\right)(\boldsymbol{z}) \tag{F.1a}$$

$$\textit{with} \quad \mathcal{E}_\gamma(\boldsymbol{w}) \coloneqq \operatorname*{\mathsf{arg\,min}}_{\boldsymbol{x}\in\mathbb{R}^n} \left\{B_{h^*}(\boldsymbol{w},\boldsymbol{x}) + \gamma r \circ \nabla h^*(\boldsymbol{x})\right\}. \tag{F.1b}$$

Proof. The proof is based on the dual symmetry property [14] of the Bregman distance

$$B_h\left(\boldsymbol{x},\boldsymbol{z}\right) = B_{h^*}\left(\nabla h(\boldsymbol{z}), \nabla h(\boldsymbol{x})\right) \tag{F.2}$$

for all $\boldsymbol{z}, \boldsymbol{x} \in \operatorname{int} \operatorname{dom} h$. Then, we have

$$\begin{aligned}
\mathsf{prox}_{\gamma r}^{h}(\boldsymbol{z}) &= \operatorname*{\mathsf{arg\,min}}_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ B_h(\boldsymbol{x}, \boldsymbol{z}) + \gamma\, r(\boldsymbol{x}) \right\} \\
&= \operatorname*{\mathsf{arg\,min}}_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ B_{h^*}\left(\nabla h(\boldsymbol{z}), \nabla h(\boldsymbol{x})\right) + \gamma\, r(\boldsymbol{x}) \right\} \\
&= \nabla h^* \circ \operatorname*{\mathsf{arg\,min}}_{\boldsymbol{w} \in \mathbb{R}^n} \left\{ B_{h^*}(\nabla h(\boldsymbol{z}), \boldsymbol{w}) + \gamma\, r(\nabla h^*(\boldsymbol{w})) \right\}
\end{aligned}$$

where we used $\boldsymbol{x} = \nabla h^{-1}(\boldsymbol{w}) = \nabla h^*(\boldsymbol{w})$. Similar result can be found in [107]. □

It was established that there is a unique Bregman distance corresponding to every *regular exponential family (REF)* distribution [10]. Consequently, given a valid reference function h, the operator (F.1b) can be interpreted as mean MAP estimator of a REF distribution. In case h is the squared Euclidean, we have $\mathsf{prox}_{\gamma r}^{h} = \mathcal{E}_{\gamma}$. That is, the BPO (F.1a) generalize the Gaussian denoiser into the REF distributions.

F.2 Proof of Theorem 7.1

The proof of Theorem 7.1 requires the following two lemmas.

Lemma F.1. *[78, Theorem 4.2.2] If h is μ-strongly convex function, then ∇h^* is $1/\mu$-Lipschitz, that is*

$$\|\nabla h^*(\boldsymbol{x}) - \nabla h^*(\boldsymbol{y})\| \leq \frac{1}{\mu} \|\boldsymbol{x} - \boldsymbol{y}\| \tag{F.3}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \operatorname{int} \operatorname{dom} h$.

Lemma F.2. *Assume that the reference function h is μ_h-strongly convex with L_h-Lipschitz continuous gradient. Assume g is μ_g-strongly convex function with L_g-Lipschitz continuous*

gradient. Then

$$\mathsf{F}(\boldsymbol{x}) = (\nabla h - \gamma \nabla g)(\boldsymbol{x}) \tag{F.4}$$

is Lipschitz continuous with constant

$$\rho(\gamma) = \mathsf{max}\{|\mu_h - \gamma L_g|, |L_h - \gamma \mu_g|\}. \tag{F.5}$$

With the lemmas above, we establish following proof for Theorem 7.1.

Proof. Using Lemma F.1 and F.2, the operator $\mathsf{T} = \mathsf{D}\left(\nabla h^*(\nabla h - \gamma \nabla g)\right)$ representing the updates in (7.13) is Lipschitz with coefficient

$$L = M \ \mathsf{max}\{|1 - \gamma L_g/\mu_h|, |L_h/\mu_h - \gamma \mu_g/\mu_h|\}. \tag{F.6}$$

Considering fixed point convergence is achieved if $L < 1$, the result is obtained with elementary algebra. In the case where $h(\boldsymbol{x}) = 1/2\|\boldsymbol{x}\|^2$, the result reduces to [158, Theorem 1]. □

Appendix G

Supplement for Chapter 8

This chapter provides additional technical details and experimental results of the proposed method CoRRECT. Section G.1 reports the architectures and training details of the our network. Section G.2 provides additional validation on the performance of CoRRECT.

G.1 Network Architectures and Training

G.1.1 Network Architectures

The regularization network $\mathsf{D}_{\boldsymbol{\theta}}$ used in our reconstruction module is a customized version of the original DnCNN. It consists of 7 layers, where the first and the last is a convolution (conv) layer with a kernel size of 3 followed by rectified linear unit (ReLU), and the middle ones are just conv. Filters of all convs are set to 32. $\mathsf{D}_{\boldsymbol{\theta}}$ is implemented using the strategy of residue learning, where its outputs are the artifacts in the inputs, and the clean predictions are obtained by subtracting those artifacts from the inputs.

The CNN used in our R_2^* estimation module is modified based on the popular U-Net architecture [155]. It consists of five encoder blocks, four decoder blocks with skip connections, and an output block. These connection increase the effective receptive field of the network as the input goes deeper in the network. For each block in encoder and decoder blocks, it consists of convs with a kernel size of 3 followed by ReLU. Filters of all convs are set to 64.

G.1.2 Network Training

We use a warm-up strategy to initialize the reconstruction and estimation modules in our network for joint training. During the warm-up stage, both modules are first trained separately for their own task on our simulated data. The reconstruction module is trained to recover mGRE images from the subsampled, noisy and motion-corrupted k-space data, while the R_2^* module is trained to estimate high-quality R_2^* from motion-corrupted mGRE images. Both modules are trained for 400 epochs when stable convergence is observed. We then plug the two warmed-up modules into our CoRRECT framework and train them jointly for another 50 epochs. Final network instances used in our experiments are finalized based on the performance on our validation dataset.

G.2 Supporting Materials and Additional Validation

In this section, we provide more supporting materials and experimental validation for CoRRECT, which was trained to directly produce high-quality mGRE images and R_2^* maps from subsampled, noisy, and motion-corrupted k-space measurements. These materials include the sampling masks used in our experiments, the performance of our methods at different echoes, and the comparison of artifacts caused by motion, sampling, and their

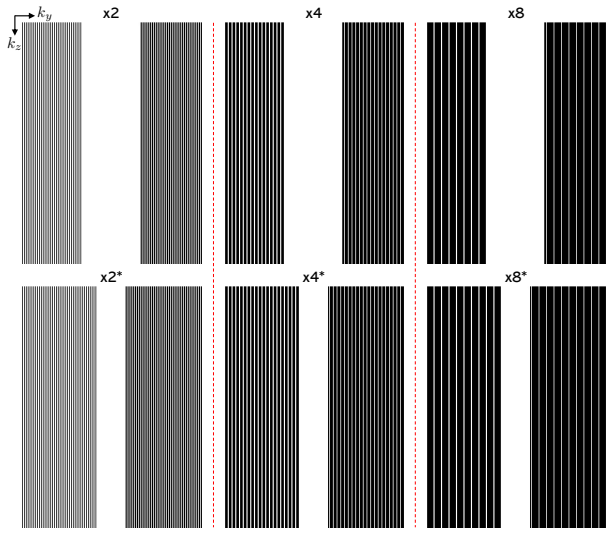

Figure G.1: Different subsampling masks used in our experiments. The masks in the first row are the ones used in our main manuscript, where the center 60 out of 192 lines are fully sampled while the other parts are subsampled with rates 50%, 25% and 12.5%, denoted as acceleration rate $\times 2$, $\times 4$, and $\times 8$ respectively. The masks in the second row provide more challenging sampling patterns used in this appendix for additional validation. These sampling patterns keep the center 30 out of 192 lines fully sampled while the other parts are subsampled with rates 50%, 25% and 12.5%, denoted as acceleration rate $\times 2^*$, $\times 4^*$, and $\times 8^*$ respectively.

combination. Additional validation using data from different subjects and more challenging sampling patterns than the ones used in our main manuscript are also provided.

G.2.1 Supporting Materials

Configuration of Motion Simulation

To generate a range of realistic motion artifacts for our simulated data, we followed the configuration in [201] for motion simulation, introducing various levels of motion artifacts to our training, validation and testing dataset. We explain the details of such configurations here. We selected the total number of motions occurring during data acquisition as a random number in the range from 1 to 10. For each motion, we simulated random in-plane shifts

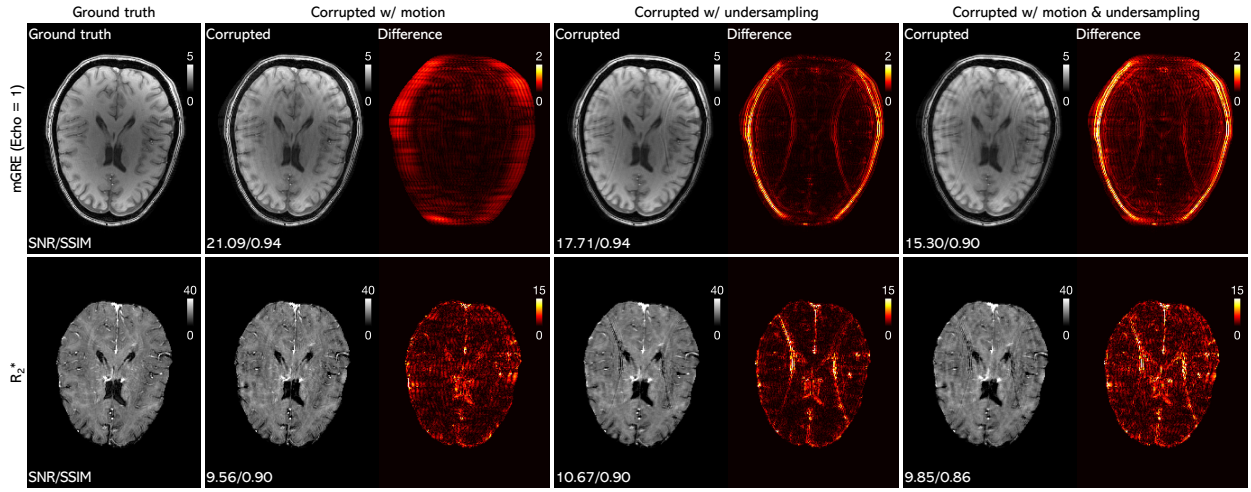

Figure G.2: The visualization of artifacts caused by different synthetic corruptions in k-space data. The effects of motion, subsampling and their combination are shown in columns 2, 4, and 6, and their difference with respect to the ground truth in the first column are shown in columns 3, 5, and 7, respectively. Note the motion and subsampling cause different artifacts in our mGRE images, where the former leads to the ring-shape artifacts near the skull, and the latter adds to the overall blurry and aliasing effects in the central region.

within the range of 0 to 15 voxels in each dimension followed by a random rotation within the range of 0° to 15° relative to the center of a 2D mGRE data slice. The time at which each motion occurred and the duration it lasted were randomly generated as well. In particular, all motions were assumed to occur randomly throughout the whole examination process, and each of them is assumed to last for a random duration from about 3 seconds to 30 seconds, which would be equivalent to disturbing about 1 to 10 k-space lines in a single 2D slice. All random numbers mentioned above were uniformly generated in the given range, introducing various levels of motion artifacts to our training, validation and testing dataset. Considering the fact that k-space scanning in the echo direction is much faster than the physical movement, we assume that all 10-echo images of a data slice suffer from the same motion effects. While the simulation setting above yields excellent performance in our experimental data, it can be adjusted for different applications.

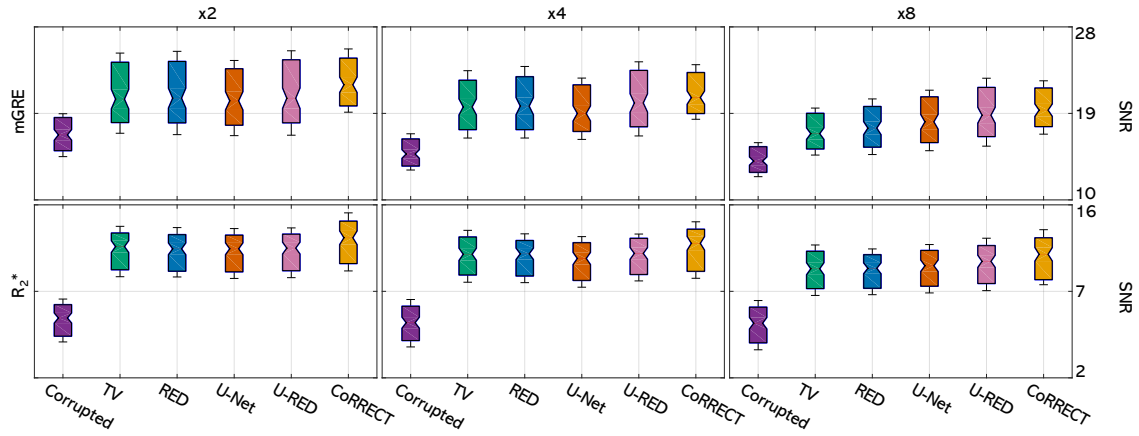

Figure G.3: The statistical analysis of SNR values obtained over the testing dataset corrupted with random levels of synthetic motion. Results highlight the performance of CoRRECT in both mGRE reconstruction and R_2^* estimation against different approaches.

Sampling Masks

Fig. G.1 shows the subsampling masks used in our experiments. Note that the masks in the first row are the ones used in the experiments in our main manuscript. Those sampling masks keep the center 60 out of 192 lines fully sampled while the other parts subsampled with rates to 50%, 25% and 12.5%, denoted as acceleration rate $\times 2$, $\times 4$, and $\times 8$ respectively. The masks in the second row provide more challenging subsampling patterns that are used in this appendix for additional validation. These sampling masks keep the center 30 out of 192 lines fully-sampled while the other parts are subsampled with rates of 50%, 25% and 12.5%, denoted as acceleration rate $\times 2^*$, $\times 4^*$, and $\times 8^*$ respectively. The results of our method with these more challenging masks are shown in Fig. G.7.

Effects of Different Corruptions

Fig. G.2 illustrates the artifacts in mGRE images and their R_2^* estimation caused by different synthetic corruptions in the k-space data. Corrupted mGRE images are obtained by zero-filling

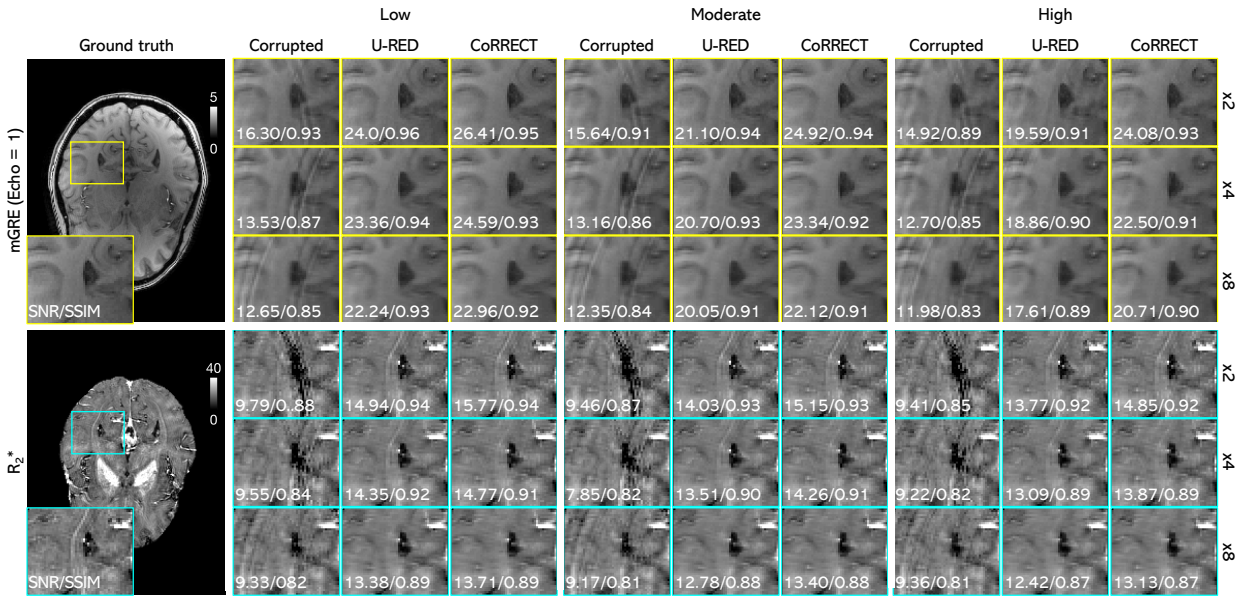

Figure G.4: Performance of CoRRECT compared against baseline method *U-RED* on exemplar testing data with synthetic motion of levels $\{low, moderate, high\}$ and acceleration rates $\{\times 2, \times 4, \times 8\}$. The bottom-left corner of each image provides the SNR and SSIM values (on the full-size image) with respect to the ground-truth. Note that while the baseline method U-RED gradually collapses along with the increase of motion levels, CoRRECT maintains a much more stable performance in terms of artifact removal and detail maintenance, which highlights the robustness of our method.

from corrupted k-space measurements, and the corresponding R_2^* for each is obtained using NLLS fitting (which has no artifact correction capabilities). Note that motion movements mostly introduce ring-shape artifacts near the skull, while subsampling causes the additional overall blurring and aliasing effects in the central region. The combination of both results in comprehensive artifacts in mGRE images and therefore collapses the estimation of R_2^*. Our method, CoRRECT, was developed and shown to be able to fix such comprehensive artifacts in both mGRE reconstruction and R_2^* estimation. Though those synthetic artifacts may not model the real ones perfectly, our method trained on such synthetic data resulted in excellent performance on our real-world experimental data, as shown in the extensive performance validation in both our main manuscript and this appendix.

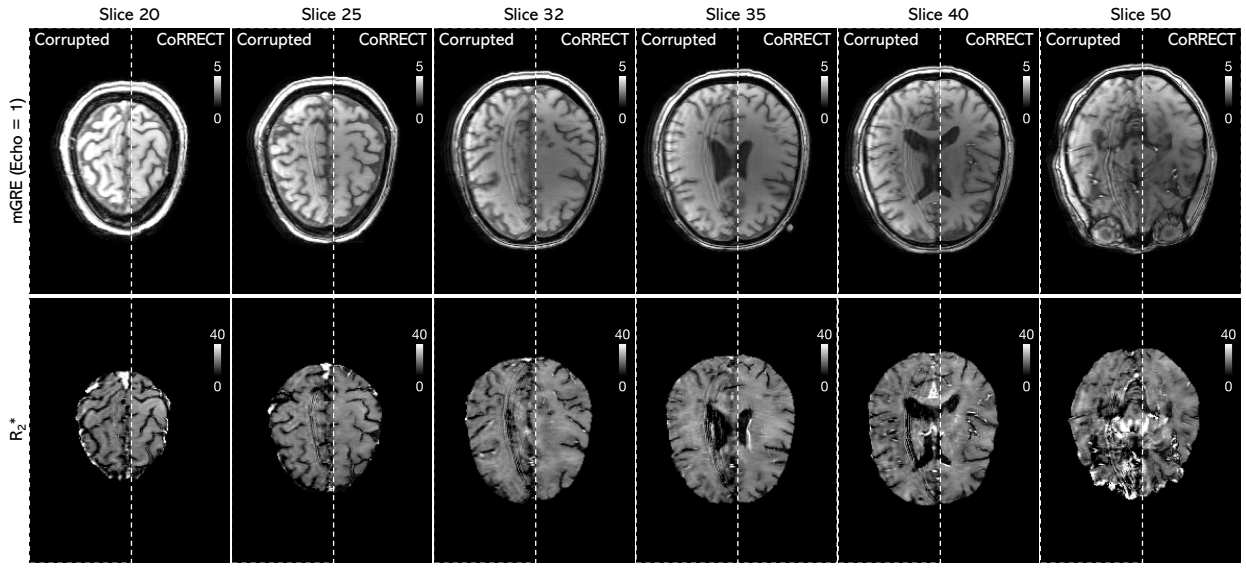

Figure G.5: The performance of CoRRECT on additional experimental data corrupted with real motion and subsampled with acceleration rate $\times 4$. The first row shows the mGRE images across different slices in a whole brain volume of 72 slices, and the second row shows the corresponding R_2^* maps estimated from these mGRE images. For a given slice in each column of the first row, the image to the left of the dashed line is the mGRE image reconstructed by zero-filling from subsampled, noisy and motion-corrupted k-space data, and the image to the right is reconstructed by CoRRECT. In each column of the second row, the R_2^* to the left of the dashed line is estimated by applying NLLS to the corrupted mGRE image above it, and the right is produced by our method. This demonstrates the capability of CoRRECT to remove artifacts for the whole brain volume.

G.2.2 Additional validation on Simulated Data

Fig. G.3 provides statistical analysis for the results shown in Table 8.1 in our main manuscript. Fig. G.3 visualizes the statistical significance compared to the baseline methods in both mGRE reconstruction and R_2^* estimation, thanks to the joint training of our the mGRE reconstruction and R_2^* estimation module.

To evaluate the robustness of our network trained on random motion across different levels of motion artifacts, we specially synthesized three motion levels such that the artifacts introduced by each are representative of different levels of motion corruption that appear

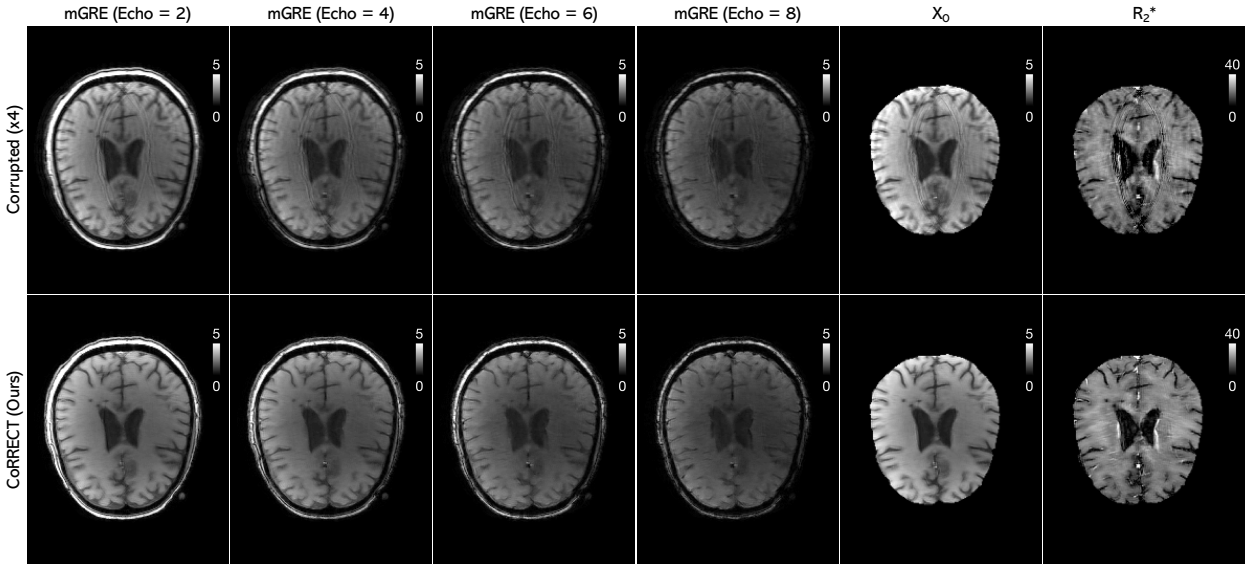

Figure G.6: The performance of CoRRECT for different echoes on experimental data corrupted with real motion and subsampled with acceleration rate $\times 4$. The different echoes of the 10-echo corrupted mGRE and its NLLS-based (X_0, R_2^*) estimation are shown in the first row, and the CoRRECT reconstructed mGRE images and (X_0, R_2^*) estimation are shown in the second row. The results validate the performance of our method on the whole mGRE sequence.

in our experimental data. We name the motion levels generated through each of these settings as *light*, *moderate* and *heavy*, where each manipulate 3%, 4% 6% of the k-space data, respectively. Fig. G.4 comprehensively illustrates the performance of CoRRECT across those different motion levels as well as different acceleration rates. It can be seen that while the baseline method U-RED gradually collapses with increased motion level and acceleration rate, our method results in rather stable performance in terms of artifact correction and detail maintenance. This shows the robustness of our network over different levels of motion.

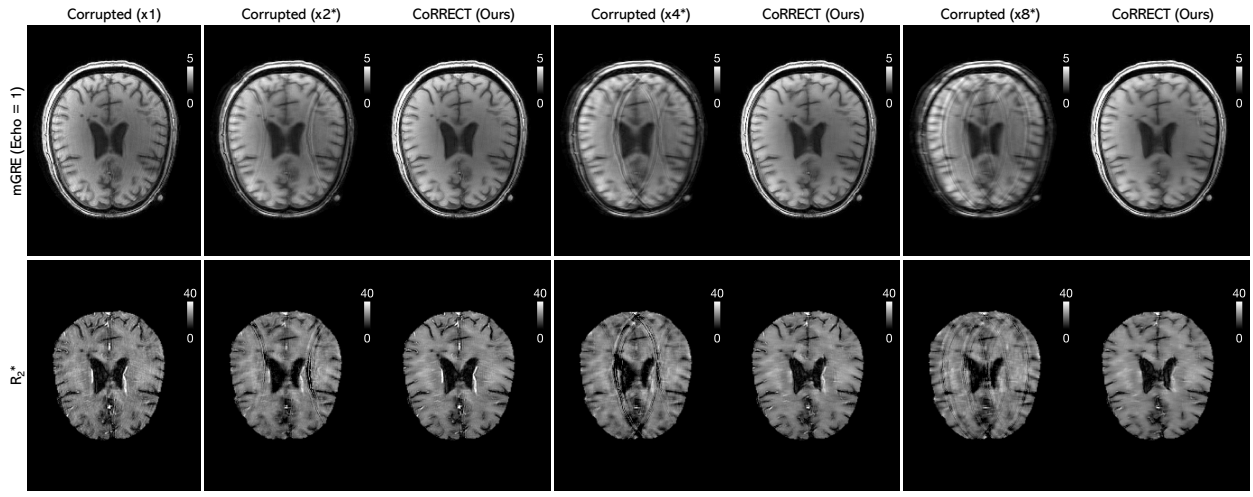

Figure G.7: Performance of CoRRECT on exemplar testing data corrupted with real motion and challenging acceleration rates $\{\times 2^*, \times 4^*, \times 8^*\}$. The mGRE image in column 1, denoted with $\times 1$, is from the motion-corrupted but fully-sampled k-space data, while the ones in column 2, 4, and 6 are from the motion-corrupted and subsampled k-space data. Note that our method can successfully remove the artifacts in mGRE reconstruction and produce high quality R_2^* maps even in such challenging scenarios, as shown in column 3, 5, and 7.

G.2.3 Additional Validation on Experimental Data

Performance on Additional Experimental Data

Fig. G.5 further demonstrates the performance of our method with a different subject than the ones used in our main manuscript. Similar to the results in Fig. 8.5 in our main manuscript, this figure shows the performance of CoRRECT across different data slices in a whole brain volume, where each slice, in principle, is corrupted with different and random motions during the scan. For each slice, we show a side-to-side comparison of the results of CoRRECT and the corrupted images, including the zero-filled mGRE images reconstructed from subsampled, noisy and motion-corrupted k-space data and their NLLS-estimated R_2^* maps. The constant success of CoRRECT on different brain slices shows that our network can work on the whole spectrum of brain volume, highlighting the effectiveness and adaptability of our method.

Performance of X_0 Estimation and Reconstruction of Different Echoes

Fig. G.6 shows the performance of CoRRECT across different echoes on experimental data. We show the comparison between the reconstructed mGRE images of our method and the corrupted mGRE images at different echoes in a 10-echo mGRE image sequence. We also show the (X_0, R_2^*) maps estimated from each sequence. One can observe that CoRRECT successfully removes the artifacts for different echoes and produce high-quality (X_0, R_2^*) maps.

Performance on More Challenging Sampling Patterns

Fig. G.7 shows the performance of CoRRECT with more challenging subsampling masks than the ones used in our main manuscripts. These subsampling masks are shown in the second row of Fig. G.1. Notice the great performance of CoRRECT even in such challenging scenarios highlights the artifact correction capability of our method.