ADVANCING IMAGE RECONSTRUCTION AND RESTORATION THROUGH ROBUST SUPERVISED AND GENERATIVE MODELS

By

Shijun Liang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Biomedical Engineering—Doctor of Philosophy

2025

ABSTRACT

Medical imaging is integral to modern clinical workflows, extensively used in diagnosis, prognosis, and treatment planning for numerous diseases. Magnetic Resonance Imaging (MRI) is particularly valuable because it avoids ionizing radiation and provides excellent soft tissue contrast. However, MRI has limitations, such as prolonged scan times that raise imaging costs and increase susceptibility to artifacts like motion. Furthermore, challenges in medical imaging data acquisition and distribution arise due to patient privacy concerns and strict ethical and legal restrictions. Additionally, medical data is highly heterogeneous, varying across institutions, imaging devices, protocols, and patient demographics. Harmonizing data from diverse sources requires extensive preprocessing, complicating data acquisition.

This thesis presents algorithms that can be categorized into two main areas. The first area explores MRI reconstruction using limited or data-free machine learning techniques to bridge gaps in data acquisition. Specifically, for cases with limited training data, we propose the LONDN MRI method, which trains on a small set of adaptively chosen neighboring images that are similar to the target image. For data-free scenarios, we advanced the Deep Image Prior, introducing self-guided DIP, which is a self-regularization method leveraging a denoising regularization based on continuously perturbing inputs with random noise and applying network output smoothing to enhance generalization. Inspired by self-guided DIP, we further improve the method efficiency by building upon an insight into the impact of the DIP network input; we introduce Autoencoding Sequential DIP (aSeqDIP), which incorporates a U-Net architecture whose weights are updated sequentially with the input however simply updated in a feed-forward fashion with autoencoding regularization. Our findings indicate that LONDN MRI outperforms the supervised MoDL model by approximately 0.4 dB for MRI reconstruction from limited measurements. On a more challenging datasets such as the FSE MRI dataset, our method achieves a 0.8 dB improvement. Both the Self-Guided DIP and Autoencoding Sequential DIP (aSeqDIP) outperform the state of Art generaitve model Score MRI model by approximately 0.45 and 0.76 dB, respectively.

Secondly, we explore means to improve the reconstruction network's generalization capabilities.

We introduce an unrolling method (SMUG) combined with randomized smoothing to counteract effects of worst-case and other perturbations. This approach combines model-based deep unrolling with randomized smoothing, helping to mitigate worst-case perturbations as well as variations such as sampling pattern shifts, differing acceleration factors, and Gaussian noise. Furthermore, a pre-learned diffusion model can act as an effective purifier prior to an unrolled network by incrementally adding Gaussian noise and subsequently removing it, thus serving as a robust noise-removal method for image purification. We implemented the developed method Diffusion Purification in various experiments, particularly on biomedical lesion data, and found it outperforms common robustness approaches, such as adversarial training, randomized smoothing, and baseline methods without robustness enhancements. However, the primary drawback of diffusion models lies in their slow image generation and denoising processes, posing a significant challenge to balancing processing speed and output quality. To handle this issue. we proposed SITCOM that exploits three conditions for achieving measurement-consistent diffusion trajectories with expanded DDIM (DDIM). Building on these conditions, we propose a new optimization-based diffusion reverse sampling method that not only enforces the standard data manifold measurement consistency and forward diffusion consistency, as seen in previous studies, but also incorporates backward diffusion consistency that maintains a diffusion trajectory by optimizing over the input of the pre-trained model at every sampling step. It outperforms the state-of-the-art method DAPS for most of the tasks by 1.2 dB in terms of image quality.

To all warriors exploring in the darkness.

"There is only one heroism in the world: to see the world as it is and to love it."
— Romain Rolland

ACKNOWLEDGMENTS

First and foremost, I wish to express my gratitude to my advisor, Prof. Saiprasad Ravishankar, whose indefatigable mentorship and steady support have been pivotal in my pursuit of a doctoral degree. His guidance, patience, and unwavering dedication continually galvanized my resolve in the face of daunting challenges, and I am deeply indebted to his expertise throughout this academic journey. Moreover, I would like to extend special thanks to Michael T. McCann, whose insightful counsel in programming and mathematics anchored me during the early stages of my Ph.D. I am equally grateful to Zhishen Huang for easing my transition into doctoral life and offering invaluable advice and moral support. I also owe a debt of appreciation to Ismail Alkhouri, whose insights illuminated promising directions for diffusion model research and sharpened my writing skills. Finally, I would like to acknowledge Prof. Rongrong Wang and Prof. Qing Qu for their enriching collaborations, which broadened the horizons of my research.

I consider myself extraordinarily fortunate to have walked this path alongside a cohort of remarkable labmates who have profoundly shaped my personal growth. The thoughtful exchanges and steadfast encouragement of Avrajit Ghosh, Siddhant Gautam, Gabriel Maliakal, Evan Bell, Angqi Li, and Tiffany Owen have infused this journey with memories I will forever cherish. The synergy we cultivated has been a constant source of motivation, and I feel truly privileged to have had such a supportive team by my side

My gratitude likewise extends to my family, my mom, Yawen Chen, and my dad, Xiwen Liang, whose boundless love and unwavering support form the bedrock upon which all my achievements rest. To my parents, whose belief in my potential never wavered and whose sacrifices paved the way for every accomplishment I now claim—I am immeasurably thankful. Indeed, their quiet resilience and constant reassurance fueled my perseverance, reminding me of the values that guide my aspirations.

Lastly, I owe heartfelt thanks to the friends and extended family who steadfastly stood by me throughout this odyssey. Their presence offered respite from the most arduous trials, their laughter assuaged my worries, and their loyalty rekindled my resolve whenever it flickered. I am eternally

grateful for the solidarity and warmth they have shown, and their support will forever remain an integral part of my story.

TABLE OF CONTENTS

CHAPTER 1

OVERVIEW

1.1 Background

Magnetic Resonance Imaging (MRI) (Fessler, 2010) utilizes strong magnetic fields and radio-frequency (RF) waves to generate high-resolution, detailed images of tissues and anatomical structures within the body. It has gained widespread use in clinical practice due to advantages such as excellent soft tissue contrast, the absence of ionizing radiation, and the capability to capture a wide range of physiological phenomena through various imaging techniques. MRI is instrumental in diagnosing numerous disorders, including cerebral aneurysms, ocular and inner ear conditions, multiple sclerosis, spinal cord disorders, stroke, tumors, and traumatic brain injuries. However, a significant hurdle is the prolonged time required to acquire images. Some scanning procedures can last up to an hour, necessitating that the subject remains confined in a cramped space for extended periods. This not only poses discomfort for patients but also contributes to higher procedural costs, making MRI an expensive imaging option. Additionally, the lengthy acquisition times limit its applicability in situations requiring immediate diagnosis.

In MRI, measurements are acquired by sampling the transverse spins in the object after excitation by radiofrequency waves. The application of spatially varying magnetic field gradients allows only spins at specific resonant frequencies to be sampled during acquisition, enabling the localization of spins based on their spatial frequencies. As a result, raw MR measurements are obtained in the frequency domain, known as k-space, unlike other imaging modalities such as X-ray imaging where acquisition occurs in the image domain. The scan duration in MRI depends on the number of measurements collected in the frequency domain. One method to reduce scan time is by acquiring fewer k-space measurements than traditionally necessary, effectively adopting sub-Nyquist level sampling. Moreover, in dynamic MRI applications like cardiac imaging, acquiring fully sampled measurements may be impractical or impossible. However, aggressive undersampling can lead to the loss of critical information necessary for accurate reconstruction. Therefore, there is a pressing need to develop algorithms that can reconstruct high-quality images from limited measurements,

mitigating the trade-off between scan duration and image quality (Wen et al., 2023).

The rising demand for quantitative information over qualitative assessments in medical imaging has become increasingly apparent in recent years. Healthcare professionals are progressively relying on precise quantitative data regarding patient health, rather than generic indicators, to make more informed, case-specific decisions for monitoring, diagnosis, and treatment planning (Gatenby et al., 2013; Lee et al., 2008; Rosenthal et al., 1992; Ramani et al., 2006). This shift underscores the importance of advanced imaging techniques and reconstruction algorithms that can provide detailed, quantifiable insights while addressing the practical challenges associated with MRI.

On the other hand, Computed Tomography (CT) (Elbakri and Fessler, 2002) employs X-rays to produce cross-sectional images of the body, allowing clinicians to visualize internal structures with high spatial resolution. In CT imaging, an X-ray source rotates around the patient while a detector array captures the attenuated X-rays that pass through the body. This process generates multiple 2D X-ray projections, which are then reconstructed into a 3D image using advanced algorithms. CT's capability to visualize intricate anatomical details has led to its widespread use in diagnosing various conditions, including head trauma, fractures, pulmonary diseases, abdominal pathologies, and cardiovascular disorders (Hsieh, 2003; McCollough et al., 2009; Wintermark et al., 2015).

CT imaging offers several distinct advantages, primarily its speed and availability, making it a preferred modality in emergency settings where rapid diagnosis is essential. Additionally, it can capture high-resolution images within seconds, proving crucial in trauma cases, stroke evaluation, and other time-sensitive conditions. The relatively fast scan times also reduce the likelihood of motion artifacts caused by patient movement, which can otherwise degrade image quality. However, CT has its limitations, most notably its reliance on ionizing radiation, which poses a risk to patients, particularly with repeated exposure. The cumulative effects of radiation have led to efforts aimed at optimizing dose levels while maintaining image quality, a challenging balance that has driven advancements in CT technology and image reconstruction algorithms (McCollough et al., 2015).

CT measurements are typically acquired in the image domain by capturing the varying levels of X-ray attenuation through tissues of different densities. This enables clear differentiation between

structures such as bone, soft tissue, and air-filled cavities. However, because CT image quality is tied to radiation dose, there is an ongoing effort to minimize exposure by developing dose-reduction techniques, such as iterative reconstruction and artificial intelligence (AI)-based methods that enable diagnostic-quality images from low-dose scans. These techniques seek to reduce radiation while preserving, or even enhancing, the clarity and detail of the images.

Recently, the focus in CT has expanded beyond mere structural imaging to include functional imaging, where parameters like blood flow or tissue perfusion can provide valuable insights into physiological processes. Functional CT imaging, though in its early stages, holds promise for more comprehensive assessments of complex conditions, such as coronary artery disease and cancer. Like MRI, CT is increasingly driven by the demand for quantitative, rather than qualitative, data. This shift is evident in the growing emphasis on tools that can extract precise measurements from CT images, supporting more tailored and accurate clinical decision-making.

Our work in this thesis focuses on addressing some of the problems in both the aforementioned areas by developing dataless machine learning algorithms that allow for better reconstruction of MR images and CT images from limited sampling of measurements. Also, we want to improve the generlization of the MR images.

Chapter II briefly provides some background for the concepts that are pertinent to the algorithms that are formulated and developed in the subsequent chapters

Chapter III focuses on the how to use adaptive LOcal NeighborhooD-based Networks for MRI (LONDN-MRI) reconstruction to handle the Deep CNNs usually require enormous datasets for offline training to ensure adequate performance trade-offs. The approach efficiently learns reconstruction networks from small clusters in a training set, directly at reconstruction time. We show connections of this algorithm to a challenging bilevel optimization problem. Our algorithm for image reconstruction alternates between finding a small set of similar images to a current reconstruction, training the network locally on such neighbors, and updating the reconstruction. The proposed local learning approach is flexible and can be seamlessly integrated with various existing deep learning frameworks for MRI, such as unrolled networks and image-domain denoisers,

to enhance their performance. Our experimental results on multiple datasets (fastMRI, Stanford FSE, and fastMRI+) and across multiple k-space undersampling factors showed that the proposed local adaptation techniques surpass networks trained globally on larger datasets. We demonstrated improved performance against scan-specific deep learning methods such as deep image prior, RAKI, and LORAKI, even when using a small number of neighbors for training.

Chapter IV explores the direction from local learning to dateless learning and focuses on a self-guided DIP method, which eliminates the need for separate reference images (for network input) and gives much better image reconstruction quality than the prior reference-guided method as well as several other related and competing schemes. The proposed method relies on a crucial denoising-based regularization. Also, To gain a deeper understanding of image reconstruction using DIP, we conduct an analysis of gradient descent-trained CNNs in the over-parameterized regime. We employ a realistic imaging forward operator instead of a Gaussian measurement matrix for our analysis of the case of compressed sensing. Our primary finding is that as the number of gradient descent steps used to optimize the standard DIP objective function approaches infinity, the difference between the network estimate and the ground truth will reside in a subspace related to the null space of the forward operator and the network's neural tangent kernel. We empirically demonstrated that this method yields promising results for MRI reconstruction and image inpainting on different datasets. Notably, our approach does not involve any pre-training, and can thus readily handle changes in the measured data. Moreover, this self-guided method showed better performance than the same model trained in a supervised manner on a large dataset (with lengthy training times). This shows that highly adaptive learning approaches may have the potential to outperform traditional data-driven learning approaches in image reconstruction

Chapter V further improves the training data-free method unsupervised method self-guided- DIP by building upon an insight about the impact of the DIP network input; we introduce Autoencoding Sequential DIP (aSeqDIP), which incorporates a U-Net architecture whose weights are updated sequentially. These updates are based on objective functions that consist of an input-adaptive data consistency term and an autoencoding regularization term used for noise overfitting mitigation.

Our extensive experimental evaluations, in terms of standard image reconstruction metrics and required run-time, highlight the superior (or competitive) performance of aSeqDIP compared to DIP-based and leading DM-based methods for the tasks of MRI and CT reconstruction, denoising, in-painting, and non-linear deblurring.

Chapter VI proposed another direction of the thesis on generalization and robustness of deep learning based on MRI image reconstruction; in this direction, we proposed integrating the RS approach within the Model Based Deep Learning(MoDL) framework for the problem of MR image reconstruction. This is accomplished by using RS in each unrolling step and at the intermediate unrolled denoisers in MoDL. This strategy is underpinned by the 'pre-training + fine-tuning' technique. We empirically showed that this approach is effective. We provide an analysis and conditions under which the proposed smoothed unrolling (SMUG) technique is robust against perturbations. Furthermore, we introduce a novel weighted smoothed unrolling scheme that learns image-wise weights during smoothing, unlike conventional RS. This approach further improves the reconstruction performance. Furthermore, in this work, we evaluate worst-case additive perturbations in k-space or measurement space where image-space perturbations were considered.

Chapter VII tries to solve the problem of adversarial robustness perfectly; we introduce a general robustification framework designed to enhance the resilience of DL-based MRI reconstructors against a variety of instabilities and improve their generalization performance when faced with out-of-distribution samples. This is accomplished through integrating purification via pre-trained DMs into existing DL-based models. We present a novel approach to select a process-switching time step - a critical parameter within our DM-based purification method. This eliminates the necessity of treating it as a hyper-parameter.

Chapter VIII focuses on solving the problem of reverse sampling steps of Diffusion models such as diffusion purification. Reverse sampling steps in diffusion models, such as those used in diffusion purification, are often lengthy and computationally expensive. Moreover, errors tend to accumulate when appropriate conditions are not established. In this chapter, we identify three critical conditions necessary for achieving measurement-consistent diffusion trajectories.

Building upon these foundational conditions, we propose a novel optimization-based sampling method. Unlike traditional approaches, which primarily focus on ensuring data manifold measurement consistency and forward diffusion consistency, our method introduces an additional key element: backward diffusion consistency. This new element ensures the preservation of the diffusion trajectory by optimizing the input of the pre-trained model at every sampling step. This integrated approach significantly enhances the efficiency and accuracy of reverse sampling in diffusion models.

CHAPTER 2

BASIC IMAGE PROCESSING BACKGROUND

This chapter provides an overview of the core applications and foundational concepts used throughout this thesis. First, it reviews Magnetic Resonance Imaging (MRI) and the role of Compressive Sensing in MRI, along with an MRI-based technique for noninvasive perfusion quantification. It also introduces the Deep Image Prior and diffusion models, offering insight into the deep learning methods utilized in this work.

2.1 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) (Tran et al., 2021) utilizes a powerful magnetic field (denoted as B_0) to align the proton spins in hydrogen atoms (primarily found in water molecules within the body). By applying Radio Frequency (RF) waves, these spins are tipped, initiating a process known as precession around the B_0 field, which generates a detectable signal or voltage in a receiver coil. Unlike imaging methods such as X-ray radiography or photography, where measurements are captured in the image or spatial domain, MRI data is collected in the frequency domain, referred to as k-space. Spatial localization within k-space is achieved through magnetic field gradients along the x and y axes, known as frequency and phase encoding, respectively. In 2D imaging, gradients in the z-direction are employed to selectively excite spins within a specific cross-sectional "slab" of the subject. Once enough k-space data has been obtained, image reconstruction occurs through a Fourier transform that translates frequency domain data into the image domain (Donoho, 2006b). For 3D or volumetric imaging, the entire volume is excited, with additional spatial encoding achieved by incorporating phase encoding along the z-axis, complementing the standard phase and frequency encoding used in 2D acquisitions.

To ensure accurate image reconstruction, an ill-posed inverse problem can be formulated as:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathcal{R}(\boldsymbol{x}), \tag{2.1}$$

where $\boldsymbol{A}$ is a linear measurement operator, $\boldsymbol{y} \in \mathbb{R}^p$ are the measurements, and $\hat{\boldsymbol{x}} \in \mathbb{R}^q$ is the reconstructed image. The first term in the minimization is referred to as a data-fidelity function and

can also take on alternative forms depending on imaging setup. In classical image inpainting, $\boldsymbol{A}$ is a binary masking operator. For the task of reconstructing a multi-coil MRI image, represented by $\boldsymbol{x} \in \mathbb{C}^q$ the optimization problem is

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{c=1}^{N_c} \|\boldsymbol{A}_c \boldsymbol{x} - \boldsymbol{y}_c\|_2^2 + \lambda \mathcal{R}(\boldsymbol{x}), \tag{2.2}$$

where the k-space measurements taken from N_c coils are represented by $\boldsymbol{y}_c \in \mathbb{C}^p,\ c = 1, \ldots, N_c$. The coil-wise forward operator is denoted as $\boldsymbol{A}_c = \mathbf{M}\boldsymbol{\mathcal{F}}\mathbf{S}_c$, where $\mathbf{M} \in \{0, 1\}^{p \times q}$ is a masking operator that captures the data sampling pattern in k-space, $\boldsymbol{\mathcal{F}} \in \mathbb{C}^{q \times q}$ is the Fourier transform operator, and $\mathbf{S}_c \in \mathbb{C}^{q \times q}$ represents the cth coil-sensitivity map (a diagonal matrix) (Lustig et al., 2007).

An explicit regularizer $\mathcal{R}(\cdot)$ is employed to limit the solutions to the domain of desirable images. Various regularizers have been used in image reconstruction. For example, it can be the ℓ_1 penalty on wavelet coefficients, a total variation penalty, or patch-based sparsity in learned dictionaries (Wen et al., 2020; Ravishankar and Bresler, 2010). where the regularizer exploits the learned transform domain sparsity of reconstructed image patches or assumes that patches in the reconstructed image can be expressed as sparse linear combinations of the atoms of a learned dictionary:

$$R(x) = \min_{D,Z} \|Px - DZ\|_2^2 + \lambda \|Z\|_0, \tag{2.2}$$

or

$$R(x) = \min_{W,\alpha} \|WPx - \alpha\|_2^2 + \lambda \|\alpha\|_0. \tag{2.3}$$

Here, (2.2) and (2.3) correspond to the dictionary and transform-based regularization, respectively. P is a patch extraction operator, and W and D are the dictionary and transform matrices (typically, additional constraints or penalties are exploited for learning them). These can either be learned or fixed beforehand. α and Z represent sparse representation coefficients. Eqn. (2.2) adopts the synthesis model that tries to express each patch in the image as the sum of a few fundamental components, while Eqn. (2.3) adopts the analysis model that posits image patches can be decomposed into a few significant coefficients if an appropriate transform is applied.

The success of deep learning in domains like computer vision and image processing and the availability of pairwise training data (consisting of fully sampled reconstructions and corresponding undersampled reconstructions) has ushered in the use of deep-neural networks in compressed-sensing MRI, where a majority of techniques rely upon the richness of CNNs and GANs in their ability to learn features from training data. Typically, in these algorithms, the output of a deep network is used to regularize the MRI reconstruction problem, i.e., $R(x) = \|x - V_\theta(x_0)\|_2^2$, where V is a deep CNN whose weights are denoted by θ, and x_0 is an initial estimate of the image being reconstructed, like a zero-filled reconstruction. These deep networks are typically trained in a supervised fashion using pairwise training data consisting of a fully sampled ground truth reconstruction as the target, and the corresponding undersampled reconstruction as an input to the network (often, a zero-filled reconstruction is chosen for this purpose). Where such pairwise training data is not available, generative adversarial networks are often used instead. In these settings, $R(x) = \|x - G_\phi(x_0)\|_2^2$, where G is a CNN trained using unpaired or partially paired training data, and an (additional) adversarial objective, and ϕ are its weights, and x_0 is an initial estimate of the image being reconstructed, like a zero- filled reconstruction (Lei et al., 2020; Yang et al., 2017). Other than the reduced demands for fully-sampled training data, an advantage of using GANs for regularized reconstruction is that they yield images that have more realistic texture.

The category of supervised algorithms that have found the most success in reconstructing MR images from limited measurements are a class of algorithms called unrolled algorithms (Monga et al., 2021) . A trademark of such algorithms is that they usually extend iterative approaches to image reconstruction to incorporate pairwise training data. Usually, this involves replacing one or multiple stages in a single iteration of an image reconstruction algorithm by a deep CNN. While, unrolled loop algorithms have demonstrated their superiority amongst supervised algorithms, and are often treated as the replacement to traditional prior- based iterative reconstruction algorithms (Aggarwal et al., 2019a; Zhang and Ghanem, 2018; Hammernik et al., 2018; Schlemper et al., 2017), there has been little investigation into whether features learned by unrolled loop algorithms subsume those enforced in traditional priors like dictionary or transform learning priors, or even Total Variation

(TV)-based methods.

2.2 CT

For CT reconstruction (Shete and Jadhav, 2023) is shared the same problem setup as the MRI reconstruction where

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathcal{R}(\boldsymbol{x}), \tag{2.3}$$

$\boldsymbol{A}$ is the random transform which takes a function defined in a 2D space (often an image) and transforms it into a new function that represents the integrals of the original function over straight lines. For an arbitrary 2D object $f(x, y)$, the corresponding Radon transform with a parallel-beam X-ray CT imaging geometry can be written as follows:

$$p(s, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, \delta(x \cos\theta + y \sin\theta - s)\, dx\, dy. \tag{2.4}$$

Here, $p(s, \theta)$ denotes a Radon projection of $f(x, y)$ at a certain view angle θ. $\delta(\cdot)$ is the Dirac delta function, and s is the position of a detector unit relative to the geometry center of the X-ray imaging system. Thus, $f(x, y)\delta(x \cos\theta + y \sin\theta - s)$ represents the intersection of an X-ray beam with $f(x, y)$. The Radon projections are usually collected within a rotation interval of 180 degrees, namely, $0 \leq \theta \leq \pi$.

To reconstruct $f(x, y)$, one can use the FBP algorithm. Let's denote the two-dimensional Fourier transform of $f(x, y)$ as $F(\omega, \theta)$ and the one-dimensional Fourier transform of $p(s, \theta)$ as $P(\omega, \theta)$. According to the Inverse Fourier transform and the Central Slice Theorem, $f(x, y)$ can be expressed as follows:

$$f(x, y) = \int_0^{\pi} \int_0^{\infty} F(\omega, \theta)\, e^{2\pi i \omega (x \cos\theta + y \sin\theta)}\, \omega\, d\omega\, d\theta = \int_0^{\pi} \int_{-\infty}^{\infty} P(\omega, \theta) |\omega| e^{2\pi i \omega (x \cos\theta + y \sin\theta)}\, d\omega\, d\theta. \tag{2.5}$$

Here, $|\omega|$ is the transfer function of the ramp filter. Introducing $Q(\omega, \theta) = |\omega| P(\omega, \theta)$ and denoting the Inverse Fourier transform of $Q(\omega, \theta)$ as $q(s, \theta)$, the reconstruction of an arbitrary 2D object $f(x, y)$ via the FBP algorithm can be obtained with the following two steps:

1. Apply a ramp filtering operation to $p(s, \theta)$ with respect to the variable s in the Fourier domain, as follows:

$$q(s, \theta) = \mathcal{F}^{-1}\{|\omega| \cdot \mathcal{F}\{p(s, \theta)\}\}; \tag{2.6}$$

2. Back-project $q(s, \theta)$ to obtain the reconstruction, as follows:

$$f(x, y) = \int_0^{\pi} q(s, \theta)\big|_{s=x\cos\theta + y\sin\theta}\, d\theta. \tag{2.7}$$

Here, $s = x\cos\theta + y\sin\theta$ denotes a sinusoidal track, from which the Radon projection points are related to the reconstructed point (x, y). The reconstruction by the FBP algorithm (Kak and Slaney, 2001) might suffer from noise-induced artifacts due to the degradation of the Radon projections. To obtain a promising reconstruction, one can apply some off-the-shelf restoration algorithms in the Radon projections and/or image domain to further improve the FBP results. This indicates that the Radon inversion can be approximated by several successive operations, each of which is highly dependent on the results of the previous operation.

2.3 Deep Image Prior

Image reconstruction is an ill-posed inverse problem that seeks to recover an n-dimensional image $\mathbf{x}^*$ from an m-dimensional measurements vector $\mathbf{y}$, where $m < n$. The forward model can be formulated in different applications as $\mathbf{y} \approx \mathbf{A}\mathbf{x}^*$, where $\mathbf{A}$ is the forward operator. For multi-coil MRI, $\mathbf{A} = \mathbf{MFS}$, where $\mathbf{M}$ denotes coil-wise undersampling, $\mathbf{F}$ is the coil-by-coil Fourier transform, and $\mathbf{S}$ represents sensitivity encoding with multiple coils. For CT, we use a simplified forward operator to study the sparse-views setting: $\mathbf{A} = \mathbf{CR}$, where $\mathbf{C}$ selects specific projection views or angles, and $\mathbf{R}$ is the radon transform(corresponding to parallel beam CT).

Deep image prior (DIP) was introduced by (Aggarwal et al., 2018), showing that a U-Net generator network's architecture alone can capture substantial low-level image statistics even without prior learning. Specifically, the DIP image reconstruction is obtained through:

$$\hat{\theta} = \arg\min_{\theta} \ \|\mathbf{A} f_{\theta}(\mathbf{z}) - \mathbf{y}\|_2^2 \,, \quad \hat{\mathbf{x}} = f_{\hat{\theta}}(\mathbf{z}) \,, \tag{2.8}$$

where $\hat{\mathbf{x}}$ is the reconstructed image, and θ corresponds to the parameters of a network f. The input to the network, $\mathbf{z}$, is randomly selected and remains fixed during optimization. Although standard DIP performs well on many tasks, determining the optimal number of iterations is challenging, as the network may eventually fit noise in $\mathbf{y}$ or undesired images from the null space of $\mathbf{A}$.

To mitigate the problem of noise overfitting, previous studies considered different approaches such as regularization, early stopping (ES), and network pruning (Ghosh et al., 2024). For regularization-based methods, the work in (Liu et al., 2019b) enhanced the standard DIP by introducing a total variation (TV) regularization term for denoising and deblurring tasks, whereas the study in (Cheng et al., 2019) proposed combining DIP with stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011). The authors in (Wang et al., 2023a) use running variance as the criterion for ES, whereas the authors of (Li et al., 2021) propose combining self-validation and training to apply ES.

The input to the standard DIP (or Vanilla DIP) network is a random noise vector that, in most works, remains fixed during the optimization. Nevertheless, other works, such as those in (Zhao et al., 2020a) and (Tachella et al., 2021), have explored cases where the input contains some structure of the ground truth. The approach employed in reference-guided DIP (Ref-Guided DIP) (Zhao et al., 2020a) follows the same objective as standard DIP in (2.8). However, instead of using a fixed random noise vector as input, it utilizes a reference image closely resembling the one undergoing reconstruction. This method was applied to the task of MRI. This methodology proves particularly effective when datasets comprising structurally similar data points are available. The reference required here makes this method a data-dependent approach. In chapter V and VI , we introduce the solution of how to solve this problem properly.

2.4 Diffusion Model

Pre-trained Diffusion Models (DMs) generate images by applying a pre-defined iterative denoising process (Ho et al., 2020). In the Variance-Preserving Stochastic Differentiable Equations (SDEs)

setting (Song et al., 2021b,a), DMs are formulated using the forward and reverse processes

$$d\mathbf{x}_t = -\frac{\beta_t}{2}\mathbf{x}_t dt + \sqrt{\beta_t}d\mathbf{w}\,, \qquad d\mathbf{x}_t = -\beta_t\Big[\frac{1}{2}\mathbf{x}_t + \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\Big]dt + \sqrt{\beta_t}d\bar{\mathbf{w}}\,, \tag{2.9}$$

where $\beta : \{0,\dots,T\} \to (0,1)$ is a pre-defined function that controls the amount of additive perturbations at time t, $\mathbf{w}$ (resp. $\bar{\mathbf{w}}$) is the forward (resp. reverse) Weiner process (Anderson, 1982), $p_t(\mathbf{x}_t)$ is the distribution of $\mathbf{x}_t$ at t, and $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$ is the score function that is replaced by a neural network (typically a time-encoded U-Net (Ronneberger et al., 2015a)) $\boldsymbol{s} : \mathbb{R}^n \times \{0,\dots,T\} \to \mathbb{R}^n$, parameterized by θ. In practice, given the score function $\boldsymbol{s}_\theta$, the SDEs in (2.9) can be discretized as in (2.10) where $\boldsymbol{\eta}_t, \boldsymbol{\eta}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\eta}_{t-1}\,, \qquad \mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\Big[\mathbf{x}_t + \beta_t \boldsymbol{s}_\theta(\mathbf{x}_t, t)\Big] + \sqrt{\beta_t}\boldsymbol{\eta}_t\,. \tag{2.10}$$

When employed to solve inverse problems, the score function in (2.9) is replaced by a conditional score function which, by Bayes' rule, is $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t}\log p_t(\mathbf{y}|\mathbf{x}_t)$. Solving the SDE in (2.9) with the conditional score is referred to as *posterior sampling* (Chung et al., 2023b). As there doesn't exist a closed-form expression for the term $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{y}|\mathbf{x}_t)$ (which is termed as the measurements matching term in (Daras et al., 2024)), previous works have explored different approaches, which we will briefly discuss below. We refer the reader to the recent survey in (Daras et al., 2024) for an overview on DM-based methods for solving IPs.

A well-known method is Diffusion Posterior Sampling (DPS) (Chung et al., 2023b), which uses the approximation $p(\mathbf{y}|\mathbf{x}_t) \approx p(\mathbf{y}|\hat{\mathbf{x}}_0)$ where $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ (or simply $\hat{\mathbf{x}}_0$) is the estimated image at time t as a function of the pre-trained model and $\mathbf{x}_t$ (Tweedie's formula (Vincent, 2011)), given as

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\Big[\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big] =: f(\mathbf{x}_t; t, \boldsymbol{\epsilon}_\theta)\,, \tag{2.11}$$

where $\bar{\alpha}_t = \prod_{j=1}^{t}\alpha_j$ and $\alpha_t = 1-\beta_t$. We call the function f, defined in (2.11), as **'Tweedie-network denoiser'** (Chen et al., 2024). Here, $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = -\sqrt{1-\bar{\alpha}_t}\boldsymbol{s}_\theta(\mathbf{x}_t, t)$ (Luo, 2022) outputs the noise in $\mathbf{x}_t$. Tweedie's formula is also adopted in other DM-based IP solvers such as (Rout et al., 2023; Chung et al., 2023d; Wang et al., 2022). The drawback of these methods is that they require a large number of sampling steps.

The work in ReSample (Song et al., 2023a), solves an optimization problem on the estimated posterior mean in the latent space to enforce a step-wise measurement consistency, requiring many sampling and optimization steps.

The work in (Mardani et al., 2023) introduced RED-Diff, a variational Bayesian method that fits a Gaussian distribution to the posterior distribution of the clean image given the measurements. This approach involves solving an optimization problem using stochastic gradient descent (SGD) to minimize a data-fitting term while maximizing the likelihood of the reconstructed image under the denoising diffusion prior (as a regularizer). However, the SGD process requires multiple iterations, each involving evaluations of the pre-trained DM on a different noisy image at some randomly selected time. While RED-diff reduces the run-time, their qualitative results are not competitive on several image restoration tasks.

Recently, Decoupling Consistency with Diffusion Purification (DCDP) (Li et al., 2024) proposed separating diffusion sampling steps from measurement consistency by using DMs as diffusion purifiers (Nie et al., 2022a; Alkhouri et al., 2024), with the goal of reducing the run-time. However, for every task, DCDP requires tuning the number of forward diffusion steps for purification for each sampling step. Shortly after, Decoupled Annealing Posterior Sampling (DAPS) (Zhang et al., 2024a) introduced another decoupled approach, incorporating gradient descent noise annealing via Langevin dynamics. DAPS, similar to DPS, also requires a large number of sampling and optimization steps.

2.5 Robustness

Robustness in MRI reconstruction refers to the ability of a deep learning-based reconstruction model to maintain accurate and stable reconstructions under various perturbations and distribution shifts. These perturbations can arise due to (1) additive noise in k-space, (2) variations in sampling protocols such as changes in undersampling rates or k-space sampling locations, and (3) unseen anatomies and pathologies encountered at test time. A robust MRI reconstructor should generalize well across these conditions without requiring extensive retraining, ensuring reliable image reconstruction in real-world clinical settings.

2.5.0.1 K-space Additive Noise

Given a trained deep MRI reconstruction NN and an aliased image $\mathbf{z} = \mathbf{A}^H\mathbf{y}$, recent studies have shown that these NNs are not robust to additive perturbations $\boldsymbol{\delta}$ to $\mathbf{y}$ (Li et al., 2023). The study in (Jia et al., 2022b) presents an approach to generate worst-case additive noise that employs norm constraints, in line with the attack strategies utilized in image classification. This approach aims to produce a form of worst-case imperceptible additive noise against a reconstructor in the image domain. Given a perturbation budget $\epsilon > 0$, the worst-case additive perturbations can be obtained using the following optimization problem.

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \mathcal{L}\Big(\mathbf{CNN}_\theta(\mathbf{A}^H\mathbf{y}), \mathbf{CNN}_\theta(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}))\Big), \tag{2.12}$$

where CNN is the common neural network and $\|.\|_\infty$ is the ℓ_∞ norm and $\mathcal{L}$ is a differentiable loss function that computes the reconstruction loss. Given the original image $\mathbf{x}^*$, generating the perturbations can also be achieved by replacing the first argument of $\mathcal{L}$ in (2.12) with $\mathbf{x}^*$. A solution of (2.12) can be obtained using the Projected Gradient Descent (PGD) method (Madry et al., 2017). In this paper, we also use $\mathbf{z}_{\text{pert}} = \mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}) = \mathbf{A}^H\mathbf{y}_{\text{pert}}$ which relates perturbations in k-space and image space.

In addition to the worst-case perturbations, random/realistic additive measurement noise could also impact the performance of a reconstructor.

2.5.0.2 Training/Testing Sampling Protocol & Undersampling Rate Disparities

In addition to additive perturbations, the study presented in (Li et al., 2023) underscores an additional potential source of instability that MoDL (and other DL-based reconstructors) may face during testing. This source stems from changes in the measurement sampling rate, leading to perturbations in the sparsity of the sampling mask within $\mathbf{A}$ (Antun et al., 2020a). Furthermore, in this paper, we consider another variation that these NNs could encounter during the testing phase, involving a shift or variation in the k-space sampling locations within the matrix $\mathbf{M}$, resulting in the construction of a nonidentical forward operator for testing. For this case, $\mathbf{z}_{\text{pert}} = \mathbf{A}_{\text{test}}^H\mathbf{y}$, where $\mathbf{A}_{\text{test}} \neq \mathbf{A}$.

We remark that ensuring the robustness of a reconstruction model to variations in the sampling protocol, undersampling rate, scan contrast, etc., is crucial as it mitigates the need for re-training to all possible practical scenarios and variations, common in imaging. Re-training models for new setups is expensive. Moreover, the relatively limited training data availability (which requires fully-sampled measurements as labels in supervised learning) in reconstruction applications also warrant learning models that can still be significantly robust.

2.5.0.3 Unseen Anatomies & Pathologies at Testing Time

A lesion (or anatomy changes) denotes an anomaly, or impairment within a tissue or organ of the body, arising from diverse factors such as injuries, diseases, or pathological conditions. In the medical domain, the term commonly characterizes regions of abnormal or diseased tissue, observed through MR imaging. In this paper, we study the practical case where the DL-based image reconstructor is trained on some data points, but tested with measurements with unseen lesions.

Figure 2.1 illustrates reconstructed images from the instabilities and the generalization challenges

2.6 Definition for Common Metrics

For the performance evaluation, we employed three widely used metrics to assess the reconstruction quality of different methods. These metrics quantify the similarity between the reconstructed images and the ground truth images derived from fully sampled k-space data. The chosen metrics were:

- **Peak Signal-to-Noise Ratio (PSNR)**: PSNR, measured in decibels (dB), is a standard metric for evaluating image reconstruction quality. It is defined as:

$$\text{PSNR} = 10\log_{10}\left(\frac{\max(I_{\text{gt}})^2}{\text{MSE}}\right), \tag{2.13}$$

  where I_{gt} is the ground truth image, and MSE (Mean Squared Error) represents the average squared intensity differences between the reconstructed image and the ground truth. A higher PSNR value indicates better reconstruction quality, as it suggests lower error between the reconstructed and ground truth images.

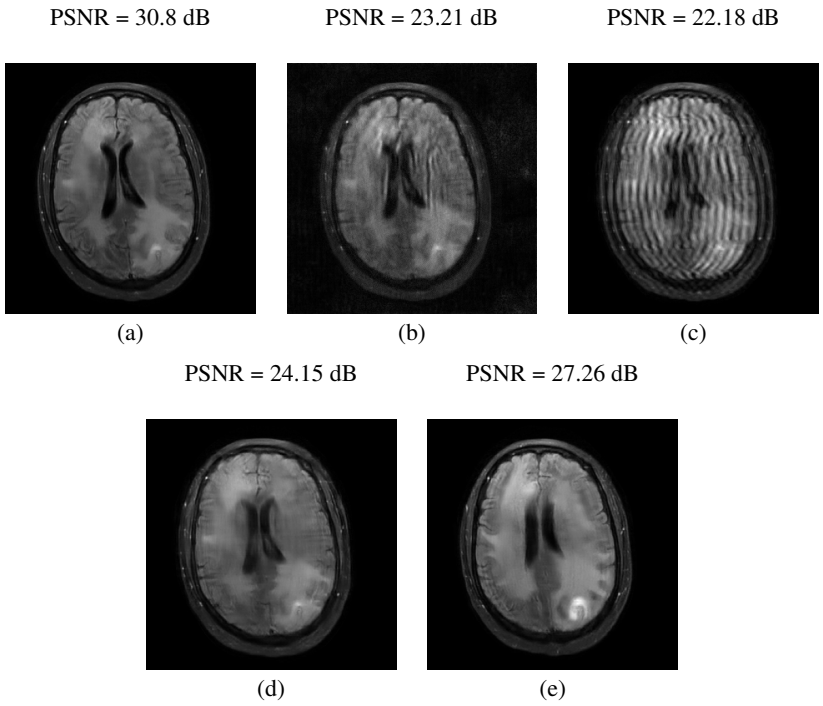


Figure 2.1 Here, we show the vulnerabilities and generalization challenges of DL-based MRI reconstruction models by evaluating a trained MoDL reconstructor (trained at 4x undersampling). (a) Reconstructed image from clean measurements. (b) Reconstructed image from measurements with worst-case additive perturbations (Equation (2.12) with $\epsilon = 0.02$). (c) Reconstructed image from measurements with 2x undersampling rate during testing. (d) Reconstructed image from a different test time sampling mask with 4x undersampling. (e) Reconstructed image from measurements with an unseen lesion during testing.

- **Structural Similarity Index (SSIM)**: SSIM is a perceptual metric that evaluates the similarity between two images by considering luminance, contrast, and structural information. It is computed as:

$$\text{SSIM}(I_r, I_{\text{gt}}) = \frac{(2\mu_{I_r}\mu_{I_{\text{gt}}} + C_1)(2\sigma_{I_r I_{\text{gt}}} + C_2)}{(\mu_{I_r}^2 + \mu_{I_{\text{gt}}}^2 + C_1)(\sigma_{I_r}^2 + \sigma_{I_{\text{gt}}}^2 + C_2)}, \tag{2.14}$$

  where I_r and I_{gt} denote the reconstructed and ground truth images, respectively, μ represents the mean intensity, σ denotes the standard deviation, and $\sigma_{I_r I_{\text{gt}}}$ is the covariance. C_1 and C_2 are small constants to stabilize the computation. SSIM values range from -1 to 1, with a value closer to 1 indicating higher structural similarity.

- **High Frequency Error Norm (HFEN)**: HFEN is designed to measure the preservation of high-frequency details, which are often crucial in medical image reconstruction. It quantifies

the error in fine details by comparing the reconstructed image and the ground truth after applying a Laplacian of Gaussian (LoG) filter. The HFEN metric is computed as:

$$\text{HFEN} = \frac{\|\text{LoG}(I_r) - \text{LoG}(I_{\text{gt}})\|_2}{\|\text{LoG}(I_{\text{gt}})\|_2}, \tag{2.15}$$

where $\text{LoG}(\cdot)$ represents the Laplacian of Gaussian filtering operation. The numerator measures the difference between LoG-filtered reconstructed and ground truth images using the ℓ_2-norm, while the denominator normalizes this difference by the ℓ_2-norm of the LoG-filtered ground truth. Lower HFEN values indicate better reconstruction performance, as they imply reduced high-frequency reconstruction errors.

These three metrics collectively provide a comprehensive assessment of reconstruction quality: PSNR captures overall signal fidelity, SSIM measures structural consistency, and HFEN evaluates the preservation of fine details.

CHAPTER 3

LONDN-MRI

3.1 Introduction

Recent medical image reconstruction techniques focus on generating high-quality medical images suitable for clinical use at the lowest possible cost and with the fewest possible adverse effects on patients. Recent works have shown significant promise for reconstructing MR images from sparsely sampled k-space data using deep learning. In this work, we propose a technique that rapidly estimates deep neural networks directly at reconstruction time by fitting them on small adaptively estimated neighborhoods of a training set. In brief, our algorithm alternates between searching for neighbors in a data set that are similar to the test reconstruction, and training a local network on these neighbors followed by updating the test reconstruction. Because our reconstruction model is learned on a dataset that is in some sense similar to the image being reconstructed rather than being fit on a large, diverse training set, it is more adaptive to new scans. It can also handle changes in training sets and flexible scan settings, while being relatively fast. Our approach, dubbed LONDN-MRI, was validated on multiple data sets using deep unrolled reconstruction networks. Reconstructions were performed at four fold and eight fold undersampling of k-space with 1D variable-density random phase-encode undersampling masks. Our results demonstrate that our proposed locally-trained method produces higher-quality reconstructions compared to models trained globally on larger datasets as well as other scan-adaptive methods.

3.2 Introduction

In applications like X-ray computed tomography (CT) (Elbakri and Fessler, 2002) and magnetic resonance imaging (MRI) (Fessler, 2010), reconstructing images from undersampled or corrupted observations is of critical importance. For example, this is necessary to reduce a patient's exposure to radiation in CT or reduce time spent acquiring MRI data. MRI scans involve sequential data acquisition resulting in long acquisition times that are not only a burden for patients and hospitals, but also make MRI susceptible to motion artifacts. Reconstructing images from limited measurements can speed up the MRI scan, but usually entails solving an ill-posed inverse problem. Recent

approaches to accelerating MRI acquisition such as compressed sensing (CS) (Donoho, 2006a) reduce scan time by collecting fewer measurements while preserving image quality by exploiting image priors or regularizers. Historically, regularization in CS-MRI has been based on sparsity of wavelet coefficients (Mihcak et al., 1999) or using total variation (Ma et al., 2008). While conventional CS assumes sparse or incoherent signals, approaches based on learned image models have been shown to be more effective for MRI reconstruction, starting with learned synthesis dictionaries (Ravishankar and Bresler, 2011; Lingala and Jacob, 2013). The dictionary parameters could be learned from unpaired clean image patches from a dataset and used for reconstruction or learned simultaneously with image reconstruction (Ravishankar et al., 2015; Xu et al., 2012; Ye et al., 2021; Ravishankar and Bresler, 2011).Additionally, recent advances in sparsifying transform learning have resulted in efficient or inexpensive data-adaptive sparsity-based reconstruction frameworks for MRI (Ravishankar and Bresler, 2012; Ravishankar et al., 2020; Wen et al., 2020). Other contemporary techniques could allow learning explicit regularizers in a supervised manner (Ghosh et al., 2022) for improved image restoration.

Deep learning (DL) has emerged as a potent methodology for tackling large-scale inverse problems, notably in enhancing image reconstruction techniques in MRI and CT. Predominantly, end-to-end CNN, as exemplified by the U-net model (Ronneberger et al., 2015b; Jin et al., 2017), have been employed to mitigate artifacts arising from undersampling in MRI datasets. Additionally, a plethora of alternative network models such as the Transformer (Feng et al., 2021), and Generative Adversarial Networks (GANs)(Lei et al., 2021), have demonstrated their effectiveness in MRI reconstruction, as detailed in comprehensive reviews like (Ravishankar et al., 2020). Furthermore, transfer learning (Dar et al., 2017) has also been used with neural networks for MRI reconstruction to achieve domain transfer.

To enhance both stability and performance, hybrid-domain approaches such as (Aggarwal et al., 2019a) enforce data consistency (i.e., the reconstruction is enforced to be consistent with the measurement model) all through training and reconstruction. Networks incorporating data consistency layers are pivotal in MR imaging, maintaining alignment between the reconstructed

image and the original data in k-space (Zheng et al., 2019; Schlemper et al., 2018). This category encompasses various methodologies, including deep unrolling-based methods (Yang et al., 2016; Hammernik et al., 2018)(which adapt traditional iterative algorithms to learn regularization parameters) regularization by denoising approaches (Romano et al., 2017), and plug-and-play methods (Buzzard et al., 2018), among others. Distinctively, the ADMM-CSNet (Yang et al., 2016) utilizes neural networks for the optimization of ADMM parameters, diverging from the ISTA-Net (Zhang and Ghanem, 2018), which focuses on refining CS reconstruction models grounded in the Iterative Shrinkage-Thresholding Algorithm. While these CNN-based reconstruction methods have demonstrated superiority over traditional CS techniques, concerns regarding their stability and interpretability persist, as highlighted in (Antun et al., 2020b).

Apart from algorithmic advances, another driving force behind deep learning-based reconstruction is the the rapid growth of publicly available training datasets. The availability of (paired or unpaired) training data sets made possible by efforts like OCMR (Chen et al., 2020) and fastMRI (et al, 2019) has enabled rapidly demonstrating the capacity of deep learning-based algorithms for improved image reconstruction or denoising quality in MRI applications.

However, one major drawback of these learned approaches is that they typically require large training datasets such as fully sampled MRI data to be effective. A recent scan-specific deep learning method is the deep image prior (Ulyanov et al., 2018), which has been applied to MRI (Darestani and Heckel, 2021) and learns a neural network for reconstruction in an unsupervised fashion from a single image's measurements. Other scan-adaptive methods include RAKI (Ak**c**cakaya et al., 2019), which is a nonlinear deep learning-based auto-regressive auto-calibrated reconstruction method. RAKI could be viewed as a deep neural network-based version of the parallel imaging scheme GRAPPA (Deshmane et al., 2012).

LORAKI (Kim et al., 2019) is another scheme that trains an autocalibrated recurrent neural network (RNN) to recover missing k-space data. The 1D deep low-rank and sparse network (ODLS) (Wang et al., 2023c) demonstrates enhanced robustness for 2D MR image reconstruction, particularly in scenarios characterized by a limited number of training samples. All these methods learn

scan-specific networks without requiring large datasets. A related approach dubbed self-supervised learning has also shown promise for MRI (Yaman et al., 2020) and uses a large unpaired data set.

3.2.1 Contributions

While deep learning approaches have gained popularity for MRI reconstruction due to their ability to model complex data sets, they often have difficulties generalizing to new data or distinct experimental situations at test time.

Deep CNNs usually require enormous datasets for offline training to ensure adequate performance trade-offs. In this work, we propose to learn adaptive LOcal NeighborhooD-based Networks for MRI (LONDN-MRI) reconstruction. The approach efficiently learns reconstruction networks from small clusters in a training set, directly at reconstruction time.

- The proposed models are trained using a small number of adaptively chosen neighbors that are in proximity (or are similar to in a sense) to the underlying (to be reconstructed) image (cf. (Lahiri et al., 2020) for a slightly related approach in the context of patch-based dictionary learning).
- We show connections of this algorithm to a challenging bilevel optimization problem. Our algorithm for image reconstruction alternates between finding a small set of similar images to a current reconstruction, training the network locally on such neighbors, and updating the reconstruction.
- The proposed local learning approach is flexible and can be seamlessly integrated with various existing deep learning frameworks for MRI, such as unrolled networks and image-domain denoisers, to enhance their performance.
- Our experimental results on multiple datasets (fastMRI, Stanford FSE, and fastMRI+) and across multiple k-space undersampling factors showed that the proposed local adaptation techniques surpass networks trained globally on larger datasets. We demonstrated improved performance against scan-specific deep learning methods such as deep image prior, RAKI, and LORAKI, even when using a small number of neighbors for training.

- We have shown the method's generalizability under different scenarios including different sampling patterns, and testing on data with artificial as well as natural lesions, when the training dataset didn't include such lesions. To establish clinical utility, we also conducted tests under different MR scan contrast settings and varying signal-to-noise ratios at test time, where the proposed method showed promise. Our study also encompassed an analysis of image quality vs. time consumption trade-offs when involving different networks and number of neighbors selected, and compared favorably with related approaches.

3.3 Method

Our approach relies on finding images in a data set that are in a sense similar to the one being reconstructed. The similarity may be defined using a metric such as Euclidean distance or other metrics. Assume we have a data set $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$ with N reference or ground-truth images $\boldsymbol{x}_n$ and their corresponding k-space measurements $\boldsymbol{y}_n$ (with multi-coil data), we use the distance metric d to find the k nearest neighbors to an (estimated/reconstructed) image $\boldsymbol{x}$ as follows:

$$\hat{C}_{\boldsymbol{x}} = \underset{C \in \mathcal{C}, |C|=k}{\arg\min} \sum_{r \in C} d(\boldsymbol{x}, \boldsymbol{x}_n), \tag{3.1}$$

where C is a set of cardinality k containing indices of feasible neighbors, and $\mathcal{C}$ denotes the set of all such sets with k elements. Different distance functions could produce a different set of similar neighbors, which could then affect the outcome of the reconstruction algorithm, as our network modeling is dependent on the choice of the local data set.

As a result, we used different metrics for evaluating our approach in this work. The distances serve as a proxy for data similarity, with nearby data considered similar and distant data considered dissimilar. We used the Euclidean distance, Manhattan distance, and normalized cross-correlation as distance metrics as follows.

$$d^{L1}(\boldsymbol{x}, \boldsymbol{x}_n) = \|\boldsymbol{x} - \boldsymbol{x}_n\|_1 \tag{3.2}$$

$$d^{L2}(\boldsymbol{x}, \boldsymbol{x}_n) = \|\boldsymbol{x} - \boldsymbol{x}_n\|_2 \tag{3.3}$$

$$d^{NCC}(\boldsymbol{x}, \boldsymbol{x}_n) = \frac{\left|\boldsymbol{x}^H \boldsymbol{x}_n\right|}{\|\boldsymbol{x}\|_2 \, \|\boldsymbol{x}_n\|_2} \tag{3.4}$$

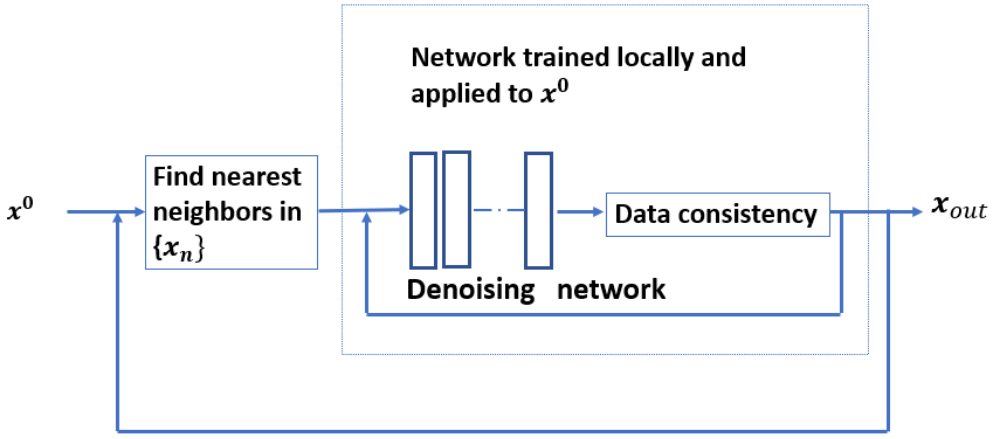


Figure 3.1 Flowchart of the proposed LONDN-MRI scheme with a specific unrolled reconstruction network. The denoising network could be for example a U-Net or the recent DIDN.

In all cases, we select the top k most similar neighbors from a set that corresponds to the k smallest distances in (3.1). The indices of the chosen images are in the set $\hat{C}_{\boldsymbol{x}}$, i.e., they are the minimizer in (3.1). These neighbors can be used to train the local model. These are expected to capture structures most similar to the image being reconstructed, enabling a highly effective reconstruction model to be learned.

3.3.1 Proposed Method

Our primary objective is to learn an adaptive neural network for MRI reconstruction, in which the model's free parameters are fitted using training data that are similar in a sense to the current scan. We emphasize that the proposed model is local in the sense that it changes in response to the input.

The advantage of the proposed method is that the model is fit for every scan and can thus be adaptive to the scan, readily handling changes in sampling masks, for example.

The algorithm begins by obtaining an initial estimate of the underlying image, denoted $\boldsymbol{x}^0$, from undersampled measurements $\boldsymbol{y}$. Our proposed strategy then alternates between computing the closest neighbors to the reconstruction in the training set and performing CNN-based supervised learning on the estimated local dataset. During supervised learning, the network weights could be randomly initialized or could be warm-started with the weights of a pre-trained (e.g., state-of-the-art)

network. In the latter case, the pre-trained network would adapt to the features of images similar to the one being reconstructed (akin to transfer learning (Dar et al., 2017)).

In each iteration, the nearest ground truth images in the training set are computed in relation to the reconstruction (estimate) predicted by the locally learned network, except in the first iteration, when the nearest neighbors are computed in relation to the (typically highly aliased) initial $\boldsymbol{x}^0$ (we used corresponding aliased images in the dataset for computing distances in the first iteration). In practice, pairwise distances to even a large number of images can be computed very efficiently (in parallel), after which the local network can be rapidly learned on a small set of neighbors (typically a shallow network or with early stopping). The network weights for deep reconstruction are constantly updated to map the initial images for the local data set to the target (ground truth) versions.

To demonstrate our approach, we used the state-of-the-art deep CNN reconstruction model MoDL (Aggarwal et al., 2019a), which is trained locally in our scheme. Additionally, we trained it globally, i.e., once on a larger dataset, in order to compare it to our on-the-fly neighborhood-based learning scheme. For completeness, we briefly recap the MoDL scheme in the following and discuss its local training within our framework. MoDL is similar to the plug-and-play approach, except that instead of pre-trained denoiser networks, end-to-end training is used to learn the shared network weights across iterations in the architecture.

3.3.2 Network Model and Training

The proposed approach is compatible with any network architecture. We use MoDL, which has shown promise for MR image reconstruction, and combines a denoising network with a data consistency (DC) module in each iteration of an unrolled architecture. MoDL unrolls alternating minimization for the following problem:

$$\mathrm{L}_a(\boldsymbol{z}, \boldsymbol{x}) := \nu \sum_{c=1}^{N_c} \|\boldsymbol{A}_c\boldsymbol{x} - \boldsymbol{y}_c\|_2^2 + \mathcal{R}(\boldsymbol{z}) + \mu\|\boldsymbol{x} - \boldsymbol{z}\|_2^2. \tag{3.5}$$

We denote the initial image in the process as $\boldsymbol{x}^0$, $\nu \geq 0$ weights the data-consistency term above, and $\mu \geq 0$ weights the proximity of $\boldsymbol{x}$ to $\boldsymbol{z}$. By decomposing the optimization into two subproblems over $\boldsymbol{z}$ and $\boldsymbol{x}$, the explicit regularizer-based update for $\boldsymbol{z}$ can be solved by replacing it with a CNN-based denoiser ($D_\theta(\cdot)$), and the denoised estimate is then used to update $\boldsymbol{x}$. The $\boldsymbol{x}$ update in the MoDL

scheme involves the data-consistency term and is performed using Conjugate Gradient (CG) descent. Thus, $\boldsymbol{z}$ is obtained as the output from a CNN-based denoiser (D_θ) and $\boldsymbol{x}$ is updated by CG.

This alternating scheme is repeated L times (unrolling), with the initial input image $\boldsymbol{x}^0$ being passed through L blocks of denoising CNN + CG updates. Now, if $\boldsymbol{S}_\theta^l(.)$ is the function capturing the lth iteration of the algorithm, then the MoDL output for the lth block is given as

$$\begin{aligned} &\boldsymbol{x}^{l+1} = \boldsymbol{S}_\theta^l(\boldsymbol{x}^l) = \boldsymbol{S}\big(\boldsymbol{x}^l, \theta, \nu_l, \{\boldsymbol{A}_c, \boldsymbol{y}_c\}_{c=1}^{N_c}\big), \text{ and} \\ &\boldsymbol{S}\big(\bar{\boldsymbol{x}}, \theta, \nu, \{\boldsymbol{A}_c, \boldsymbol{y}_c\}_{c=1}^{N_c}\big) \triangleq \\ &\arg\min_{\boldsymbol{x}} \nu \sum_{c=1}^{N_c} \|\boldsymbol{A}_c\boldsymbol{x} - \boldsymbol{y}_c\|_2^2 + \|\boldsymbol{x} - \boldsymbol{D}_\theta(\bar{\boldsymbol{x}})\|_2^2. \end{aligned} \tag{3.6}$$

After L iterations, the final output is

$$\boldsymbol{x}_{\text{supervised}} = \boldsymbol{x}^L = \left(\bigcirc_{l=0}^{L-1} \boldsymbol{S}_\theta^l\right)(\boldsymbol{x}^0) \triangleq {}_\theta(\boldsymbol{x}^0), \tag{3.7}$$

where $\bigcirc_{i=0}^{L-1} f^i$ represents the composition of L functions $f^{L-1} \circ f^{L-2} \circ \ldots \circ f^0$, and $\boldsymbol{x}^0$ is the initial image. The weights of the denoiser $\boldsymbol{D}_\theta$ are shared across the L blocks. The network parameters θ are learned in a supervised manner so that $\boldsymbol{x}_{\text{supervised}}$ matches known ground truths (in mean squared error or other metric) on a (large/global or local) training set. This involves the following optimization for training:

$$\begin{aligned} \hat{\theta} &= \arg\min_\theta \sum_{n \in S} C_\beta({}_\theta(\boldsymbol{x}_n^0); \boldsymbol{x}_{\text{n}}) \\ &= \arg\min_\theta \sum_{n \in S} \left(\left\|\boldsymbol{x}_n - {}_\theta(\boldsymbol{x}_n^0)\right\|_2^2\right), \end{aligned}$$

where n indexes the samples from the data set used for training, with $\boldsymbol{x}_{\text{n}}$ denoting the nth target (or ground truth) image reconstructed from fully-sampled k-space measurements and $\boldsymbol{x}_n^0$ denotes the initial image estimate from undersampled measurements. The cost $C_\beta(\hat{\boldsymbol{x}}_n; \boldsymbol{x}_n)$ denotes the training loss. The main difference between a globally learned and locally learned network is the choice of the set S of training indices. For the proposed local approach, we fit the network based on the k training samples closest to the current test image estimate, whereas the conventional (or global)

training would fit networks to a large dataset. The initial image estimate $\boldsymbol{x}_n^0$ is obtained from the undersampled measurements $\boldsymbol{y}_n$ using a simple analytical reconstruction scheme such as applying the adjoint of the forward model to the measurements.

In each iteration, the network is updated (Fig. 3.1), and the initial estimate of the underlying unknown image is passed through the network to obtain a new estimate. In Fig. 3.1, we illustrate the iterative process of neighbor fine-tuning and local network updating. Local learning may have the advantage of accommodating changes in experimental conditions (e.g., undersampling pattern) at test time, provided that such modified measurements and initial images for the small local training set can be easily simulated from the existing $\boldsymbol{x}_n$ or $\boldsymbol{y}_n$. Our overall algorithm is also summarized in Algorithm 3.1.

Algorithm 3.1 LONDN-MRI Algorithm

Require: Initial image $\boldsymbol{x}^0$, number of neighbors k, k-space undersampling mask $\mathbf{M}$, regularization parameters ν and μ, number of training epochs T, number of iterations of alternating algorithm S.

1: Initialize reconstruction network parameters θ with pre-learned network weights $\hat{\theta}$ or randomly initialized weights. Set $\boldsymbol{x} = \boldsymbol{x}^0$.
2: **for** Iteration < maximal iteration S **do**
3: Compute the set of k similar neighbors $\hat{C}_{\boldsymbol{x}}$ to the current reconstruction estimate $\boldsymbol{x}$ using metric d.
4: **for** epoch < maximal number T **do**
5: For each batch of neighbor data, compute the gradient of the training loss with respect to the network parameters θ and perform one update step on θ.
6: **end for**
7: Update $\boldsymbol{x} \leftarrow {}_\theta(\boldsymbol{x}^0)$
8: **end for**
9: **return** reconstruction $\boldsymbol{x}$ and learned net. parameters θ.

3.3.3 Regularization

In order to avoid over-fitting when training networks on small sets, we also adopted regularization of weights during training as follows:

$$\hat{\theta} = \arg\min_{\theta} \sum_{n \in S} \left\| \boldsymbol{x}_n - {}_\theta(\boldsymbol{x}_n^0) \right\|_2^2 + \lambda \mathcal{R}(\theta), \tag{3.8}$$

where $\mathcal{R}(\cdot)$ denotes the regularization term on network weights. We primarily used the ℓ_1 norm regularizer to enforce sparsity of the network weights to learn simpler models. We observed that

regularizing the local model enables it to converge more easily, and shrinks weights for less important or noisy features to zero. We provide more discussion in the experiments section.

3.3.4 Connections to Bilevel Optimization

The alternating algorithm for training involving a neighbor search step and a local network update step could be viewed as a heuristic algorithm for the following bilevel optimization problem:

$$\min_{C \in \mathcal{C},\, |C|=k} \sum_{i \in C} ||f_{\theta(C)}(\boldsymbol{y}) - \boldsymbol{x}_i||_2^2,$$
$$\text{s.t.} \quad \theta(C) = \arg\min_{\theta} \sum_{i \in C} ||\boldsymbol{x}_i - f_\theta(\boldsymbol{y}_i)||_2^2. \tag{3.9}$$

Here, $f_{\theta(C)}$ denotes a deep neural network learned on a subset C of a data set that maps the current k-space measurements $\boldsymbol{y}$ to a reconstruction. The network is akin to ${}_\theta(\boldsymbol{x}^0)$ shown earlier 3.8, but with $\boldsymbol{x}^0$ assumed to be generated from $\boldsymbol{y}$ (e.g., via the well-known sum of squares of coil-wise inverse Fourier transforms, or via SENSE reconstruction, etc.). Problem 3.9 aims to find the best neighborhood or cluster among the training data, where the reconstructed image belongs (with closest distances to neighbors – we assumed Euclidean distance here), with the network weights for reconstruction estimated on the data in that cluster. Problem 3.9 is a bilevel optimization problem with the cluster optimization forming the upper level cost and network optimization forming the lower level cost. Bilevel problems are known to be quite challenging (Crockett and Fessler, 2021; Ghosh et al., 2022). It is also a combinatorial problem because we would have to sweep through all possible choices of clusters of k training samples with reconstruction networks trained in each such cluster, to determine the best cluster choice.

The proposed algorithm is akin to optimizing the bilevel problem by optimizing for the network weights θ with the clustering C fixed (the lower level problem) and then optimizing for the clustering C (upper level minimization) with the network weights fixed. This is a heuristic because the optimized variables in each step are related, however, such an approach has been used in prior work (Ye et al., 2021) and shown to be approximately empirically convergent for the bilevel cost. In this work, we performed an empirical evaluation of convergence in the experiments section, where the alternating algorithm is shown to reduce the upper-level cost in (3.9).

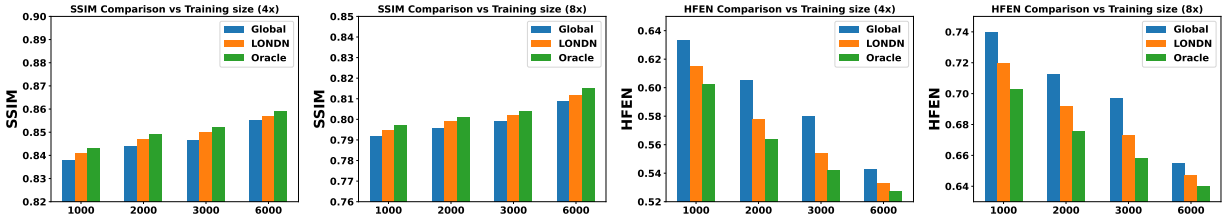


Figure 3.2 Comparison of MoDL with UNet denoiser trained globally vs. using the proposed LONDN-MRI scheme (1 iteration). Reconstruction metrics are shown across training set sizes at 4x and 8x undersampling.

3.4 Experiments

We first present the overall experimental setup in Section 3.4.1. Key results and comparisons are presented in Section 3.4.2. The intricacies and behavior of LONDN-MRI are analyzed in Section 3.4.3 and its generalizability is investigated in Section 3.4.4.

3.4.1 Experimental Setup

Datasets & Models: We evaluated the effectiveness of the proposed LONDN-MRI reconstruction method on multiple datasets: the multi-coil fastMRI knee and brain datasets (et al, 2019, 2020), the fastMRI+ dataset (which is just an annotated version of fastMRI indicating pathologies), and the Stanford 2D FSE (Cheng, 2019) dataset. The results obtained on the fastMRI knee dataset and the Stanford FSE data are described in Section 3.4.2. The fastMRI brain and fastMRI+ data are used in the studies in Section 3.4.4. For training, we randomly selected a subset of 3000 images from the fastMRI knee and brain datasets and the same for the fastMRI+ case. We used 2000 training images for the smaller Stanford FSE dataset. We used 15 or 20 images for testing in different scenarios, which were randomly chosen.

In some experiments, we evaluated the effect of training set size, where we worked with fewer or more images in the training set. Coil sensitivity maps for model-based reconstruction were generated for each scan using the BART toolbox (Uecker, 2018). We tested obtaining these using either the fully-sampled k-space or only center of k-space data and noticed very little difference in reconstruction quality between the two approaches.

Since the proposed LONDN-MRI framework is quite general and can be combined with any supervised deep learning based reconstruction approach, we chose the recent popular model-based

deep learning (MoDL) reconstruction network and compared globally (over large set of training samples) and locally (over very small matched set of samples) learned versions of the model for different choices of deep denoisers in the network

We performed reconstructions at fourfold or 4x acceleration (25.0% sampling) as well as at eightfold or 8x acceleration (12.5% sampling) of the k-space acquisition. In all cases, variable density 1D random Cartesian (phase-encode) undersampling of k-space was performed. The initial image estimates for MoDL were obtained by applying the adjoint of the measurement operator to the subsampled k-space data, and were then used to train both local and global versions of MoDL networks. In our local versions (LONDN-MRI), we used 30 images for training (searched from e.g., 3000 images). while the global versions used the full subset of training images.

Network Architectures & Training: We trained two types of MoDL models at 4x and 8x k-space undersampling, respectively. One used the well-known UNet denoiser, with a two-channel input and two-channel output, where the real and imaginary parts of an image are separated into two channels. The network weights during training were initialized randomly (normally distributed). The ADAM optimizer was utilized for training the network weights. For LONDN-MRI, we used an initial learning rate of 6×10^{-5} with a multi-step learning rate scheduler, which decreases the learning rate at 100 and 150 epochs with learning rate decay 0.65. For training globally, we used an initial learning rate of 1×10^{-4} with 150 epochs of training and a multi-step learning rate scheduler that decreased the learning rate at 50 and 100 epochs with learning rate decay 0.6. For LONDN-MRI, MoDL with 5 iterations was used with a shallow UNet that had 2 layers in the encoder and decoder, respectively. We used a shallow network with dropout for the local model to avoid over-fitting to the very small training set. For the MoDL network trained globally (on large dataset) for making comparisons with, we utilized 4 layers in the decoder and encoder in UNet and 6 MoDL blocks. We used a batch size of 2 during training for both the global and local cases. Furthermore, for the data-consistency term, we used a tolerance of 10^{-5} in CG and a μ/ν ratio of 0.1. Also, we chose the regularization weight λ as 10^{-9} for LONDN-MRI, unless specified otherwise.

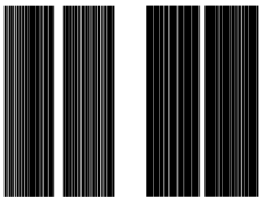
For the second MoDL architecture, we used the recent state-of-the-art denoising network

DIDN (Yu et al., 2019a; Lahiri et al., 2021). Due to the high complexity of the DIDN network, we first pre-trained it on the larger (global) dataset (learning rate, etc., similar to the UNet case) before adapting the weights within LONDN-MRI for each scan. This is an alternative to constructing shallower versions of a network for local adaptation.The ADAM optimizer was utilized for training, with a learning rate of 5×10^{-5} in LONDN-MRI. We used 6 iterations of MoDL with the DIDN denoiser for which we used 3 down-up blocks (DUBs). The number of epochs for training was 30 in LONDN-MRI. The remaining training parameters were chosen similarly as in the previous UNet-based case. Using a pre-trained state-of-the-art denoiser allows the local adaptation to converge faster.

Comparison to Scan-adaptive Methods: We compared the performance of our schemes to recent related scan-specific methods such as deep image prior (DIP) (Darestani and Heckel, 2021)(using the public package but additionally incorporating coil sensitivity maps), RAKI (Ak**c**cakaya et al., 2019)), SOUP-DIL (Ravishankar et al., 2015) (code extracted from publicly available package), and LORAKI(Kim et al., 2019) (modified from RAKI code). In our experiments, we used parameters specified in the authors' original implementations, which we observed worked well.

Sampling Masks & Performance Metrics: We used binary masks for fourfold and eightfold Cartesian undersampling of k-space. Fig. 3.3 shows the sampling masks primarily used in our experiments that include a fully-sampled central region (with 31 central lines at 4x acceleration and 15 central lines at 8x acceleration) and the remaining phase encode lines were sampled uniformly at random.

For the performance metrics, we used three common metrics to quantify the reconstruction quality of different methods. These were the peak signal-to-noise ratio (PSNR) in decibels (dB), structural similarity index (SSIM) (Wang et al., 2004), and the high frequency error norm (HFEN) (Ravishankar and Bresler, 2011), which were computed between the reconstruction and the ground truth obtained from fully-sampled k-space data. The HFEN was computed from the ℓ_2 norm of the difference between Laplacian of Gaussian (LoG) filtered reconstructed and ground truth images. This was normalized by the ℓ_2 norm of the LoG filtered ground truth.



(a) (b)

Figure 3.3 Undersampling masks used in our experiments: (a) fourfold undersampled 1D Cartesian phase-encoded; and (b) eightfold undersampled 1D Cartesian phase-encoded. The masks were zero-padded for slightly larger images.

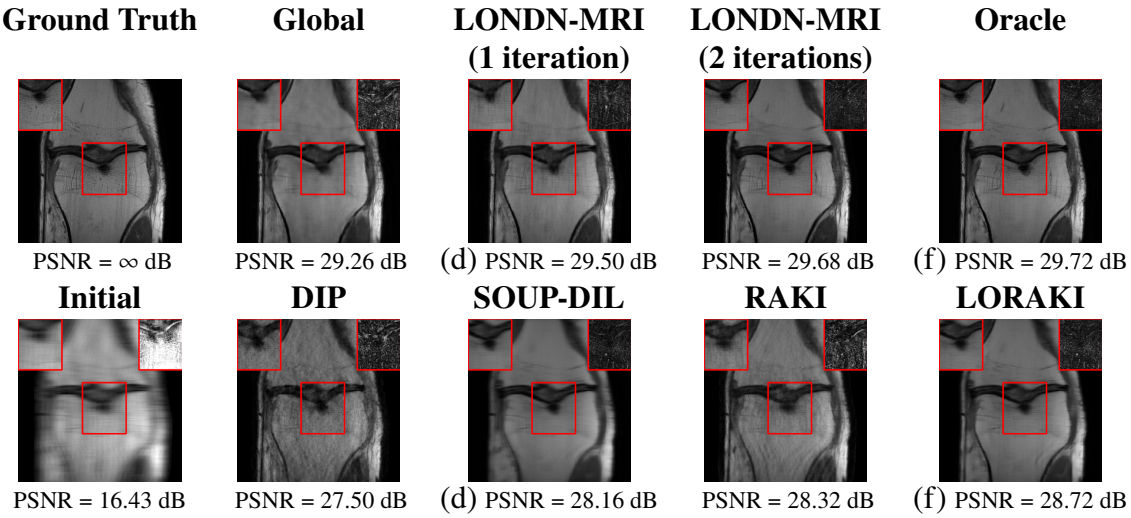


Figure 3.4 Comparison of image reconstructions with different methods at 8x undersampling. The global and LONDN-MRI methods use the MoDL architecture with UNet denoiser with 1000 training images. The inset panel on the top left in each image corresponds to a section of interest in the image (shown by the red bounding box), while the inset panel on the top right corresponds to the error map with respect to the ground truth.

3.4.2 Results and Comparisons

Results for the UNet-based Reconstructor: Table 3.1 compares the average PSNR values for reconstruction over the fastMRI knee testing set at both 4x and 8x undersampling. We varied the number of images in the training set for a more comprehensive study. We compare learning networks over a small set of similar images to learning networks over the larger datasets (global), as well as to an oracle LONDN scheme, where the neighbors in the training set were computed based on each ground truth test image. The oracle scheme would ideally provide an upper bound on the performance of the iterative LONDN-MRI scheme. Moreover, LONDN-MRI outperforms DIP, RAKI, SOUP-DIL, and LORAKI with U-Net (Table 3.1). Note that DIP, RAKI, SOUP-DIL,

and LORAKI do not use information beyond the test scan (scan-adaptive). Later, we show how LONDN-MRI performs when the overall dataset it uses is very limited.

When varying the size of the training set, the global approach was trained on the full set each time, whereas the local approach performed training on small subsets of 30 training pairs selected from the larger datasets. The iterations of the LONDN-MRI scheme quickly improve reconstruction performance, and even with only 2 LONDN-MRI alternations, the PSNR values begin approaching the oracle setting. The LONDN schemes (oracle or iterative) consistently outperform the globally trained networks across the different training set sizes considered.

We note that the results for the globally trained model with many (6000) training scans match closely the LONDN-MRI results, when LONDN-MRI uses a smaller overall training set (3000 scans) for neighbor search. This illustrates the potential of our approach with limited training data, when compared with models trained on larger sets. Figure 3.2 compares the SSIM and HFEN reconstruction metrics using bar graphs, where a similar trend is observed as with PSNR.

Figs. 3.4 and 3.5 show images reconstructed by different methods at 8x and 4x undersampling, respectively. The LONDN-MRI reconstructions (either iterative or oracle) show fewer artifacts, sharper features, and fewer errors than the global MoDL and initial aliased reconstructions. The iterative LONDN-MRI results are also quite close to the oracle result.

Ax	Data size	Global	LONDN-MRI (1 iteration)	LONDN-MRI (2 iterations)	Oracle	DIP	RAKI	LORAKI	SOUP DIL
4x	1000	32.63	32.78	**32.87**	32.99	30.1	30.25	31.35	30.97
	2000	33.00	33.28	**33.31**	33.35				
	3000	33.17	33.46	**33.51**	33.54				
	6000	33.48	33.58	**33.65**	33.69				
8x	1000	29.78	30.15	**30.26**	30.34	28.9	29.01	29.71	29.47
	2000	30.21	30.53	**30.58**	30.64				
	3000	30.47	30.76	**30.80**	30.85				
	6000	30.78	30.94	**31.04**	31.09				

Table 3.1 Average reconstruction PSNRs (in dB) for 15 images at 4x and 8x k-space undersampling. The proposed LONDN-MRI (with 1 or 2 alternations) is compared to training a global reconstructor for different training set sizes and another scan based method. We also compare to an oracle local reconstructor, where neighbors are found with respect to known ground truth test images.

Results for the DIDN-based Reconstructor: To demonstrate adaptability to different network architectures, Table 3.2 compares reconstruction performance on the test set with the DIDN denoiser-based MoDL architecture. Average PSNR values with LONDN-MRI are compared to those with

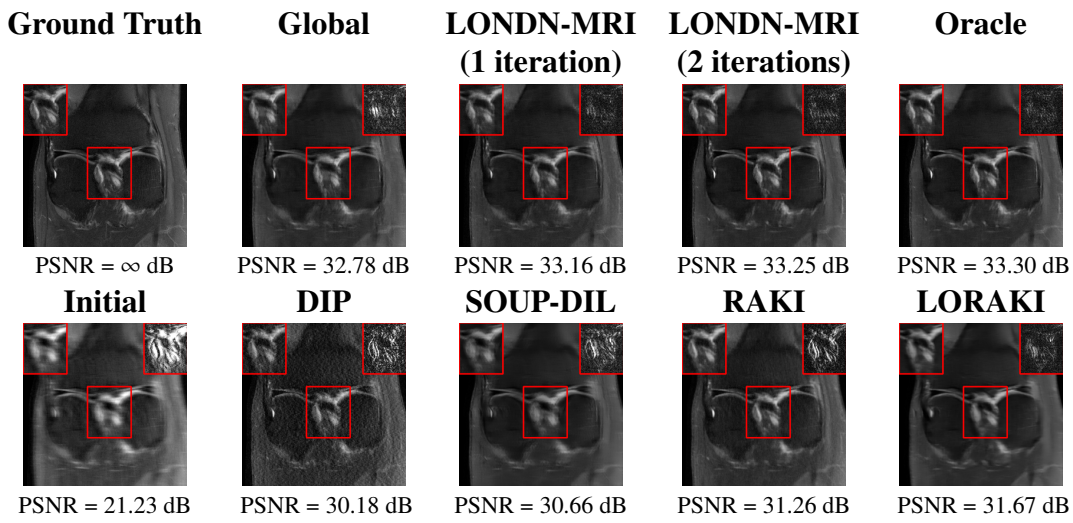


Figure 3.5 Same comparisons/setup as Fig. 3.4, but at 4x undersampling. The supervised methods used MoDL architecture with UNet denoiser (3000 training images).

Acceleration	Data Size	Global	LONDN-MRI (1 iteration)	Oracle
4x	1000	33.66	**33.92**	33.96
	2000	34.01	**34.23**	34.31
	3000	34.15	**34.39**	34.42
8x	1000	31.02	**31.33**	31.37
	2000	31.34	**31.64**	31.68
	3000	31.79	**32.08**	32.12

Table 3.2 Average reconstruction PSNR values (in dB) on the testing set at 4x and 8x undersampling for various training set sizes. MoDL reconstructor with DIDN denoiser is used.

networks trained globally at different training set sizes. We ran only 1 iteration of LONDN-MRI, where the reconstruction with a pre-trained (global) network was used to find neighbors. PSNR values for the oracle LONDN-MRI reconstructor are also shown. The overall performances with the DIDN-based architectures are better than with the UNet-based unrolled networks. The PSNRs for LONDN-MRI are consistently and similarly better than for the globally trained network across the different training set sizes considered, indicating potential for LONDN-MRI in improving state-of-the-art models. Fig. 3.6 visually compares reconstructions and reconstruction errors (in zoomed in region) for different methods. We can see that the LONDN reconstructors capture the original image features more sharply and accurately than the globally learned reconstruction.

Performance on the Stanford FSE Dataset: We also performed image reconstructions with the Stanford multi-coil FSE dataset, which is a smaller dataset. We used same settings for the networks

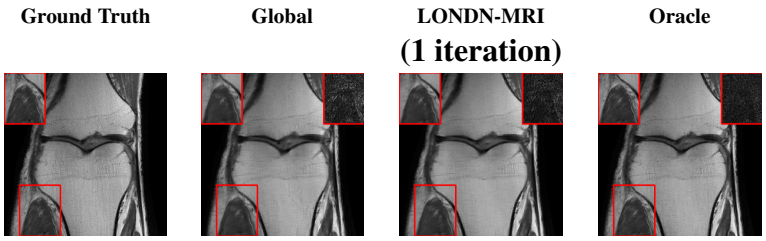


Figure 3.6 Comparison of image reconstructions at 4x undersampling for the MoDL network with DIDN denoiser and 3000 training images, when compared to LONDN-MRI. A region of interest and its error are also shown.

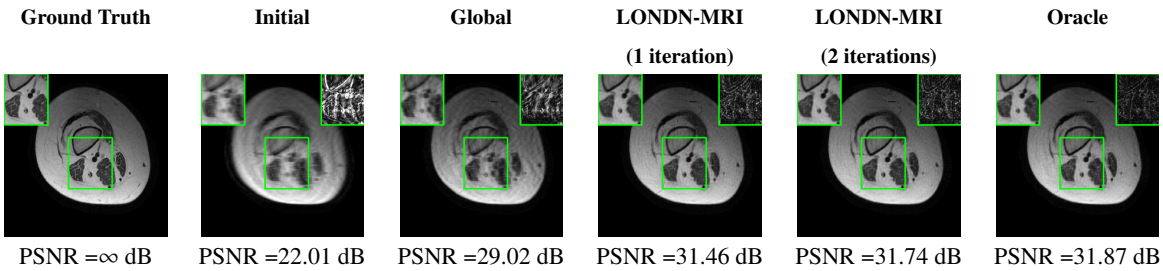


Figure 3.7 Comparison of image reconstructions with different methods at 4x undersampling using MoDL architecture with UNet denoiser with 2000 training scans. The test slice and training data were from the Stanford FSE dataset.

and training as in Section 3.4.1. Table 3.3 shows that LONDN-MRI significantly outperforms the globally learned MoDL network at both 4x and 8x acceleration. This indicates benefits for the proposed framework for smaller, more diverse datasets. Figs. 3.7 and 3.8 display visual comparisons that show the LONDN-MRI scheme recovering sharper features than the globally learned network.

Acceleration	Global	LONDN-MRI (1 iteration)	LONDN-MRI (2 iterations)	Oracle
4x	29.45	31.49	**31.56**	31.67
8x	27.25	29.35	**29.43**	29.60

Table 3.3 Average reconstruction PSNR values (in dB) for the Stanford FSE test set at 4x and 8x undersampling. The LONDN-MRI results are compared to a model globally trained on the FSE dataset.

3.4.3 Behavior of LONDN-MRI

Here, we explore the intricacies and workings of LONDNMRI in more detail.

Performance with Different Distance Metrics: To determine a suitable distance metric for

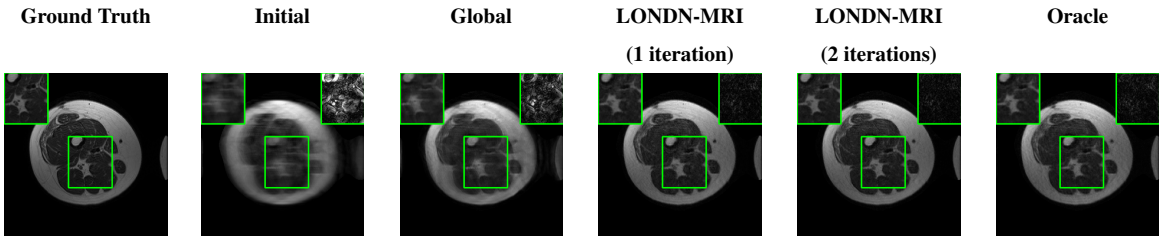


Figure 3.8 Same comparisons/setup as Fig. 3.7, but at 8x undersampling.

our method, we analyzed a few popular distance metrics. This study focused on evaluating their effectiveness in selecting the appropriate matching dataset for training in the context of LONDN-MRI (oracle scheme). We tested the performance of MoDL with UNet denoiser using L1 and L2 distance metrics as well as normalized cross-correlation (NCC), to find the matched training set from among 3000 images, which were all normalized. From the results in Table 3.4, we see that the different distance functions offer only slight differences in reconstruction performance, with NCC offering the best results with respect to all reconstruction metrics.

Acceleration	Reconstruction Metric	L1	L2	NCC
4x	SSIM	0.85	0.849	0.852
	PSNR (dB)	33.49	33.44	33.54
	HFEN	0.552	0.56	0.542
8x	SSIM	0.803	0.802	0.804
	PSNR (dB)	30.79	30.71	30.85
	HFEN	0.664	0.674	0.658

Table 3.4 Average PSNR, SSIM, and HFEN values over 15 testing images for LONDN-MRI with neighbor search peformed using L1 distance, L2 distance, and normalized cross-correlation (NCC).

Evaluating the Accuracy of Neighbor Search: Here, we study how the neighbor search proceeds across the iterations or alternations of LONDN-MRI. We are interested to know if our locally learned reconstructor can improve the neighbor finding process over iterations. We used all images from the test set. First, we find the k closest neighbors (in terms of Euclidean distance) for each ground truth test image amongst the ground truth training images. The set C_r^* contains the indices of these oracle neighbors for a test image indexed r. The set $\hat{C}_r$ contains the indices of closest neighbors from a certain iteration of LONDN-MRI. The neighbor matching accuracy (NMA) metric below computes

the average (over the test set indices $\mathcal{T}$) percentage match between the two sets:

$$\text{NMA} := \frac{100}{|C|} \sum_{r \in \mathcal{T}} \frac{|\hat{C}_r \cap C_r^*|}{k}, \tag{3.10}$$

The accuracy of the neighbor search at both 4x and 8x undersampling is shown in Fig. 3.9. The accuracy of the initial search (based on $\boldsymbol{x}^0$) and after 1 or 2 iterations of LONDN-MRI are shown. We find nearest neighbors for the initial highly aliased $\boldsymbol{x}^0$ with respect to the corresponding aliased images in the training set (based on the same k-space undersampling mask as at testing time), rather than based on the ground truth training images, because the latter resulted in lower neighbor search accuracy for $\boldsymbol{x}^0$. It is clear from Fig. 3.9 that the accuracy improves quickly and tapers off in few iterations.

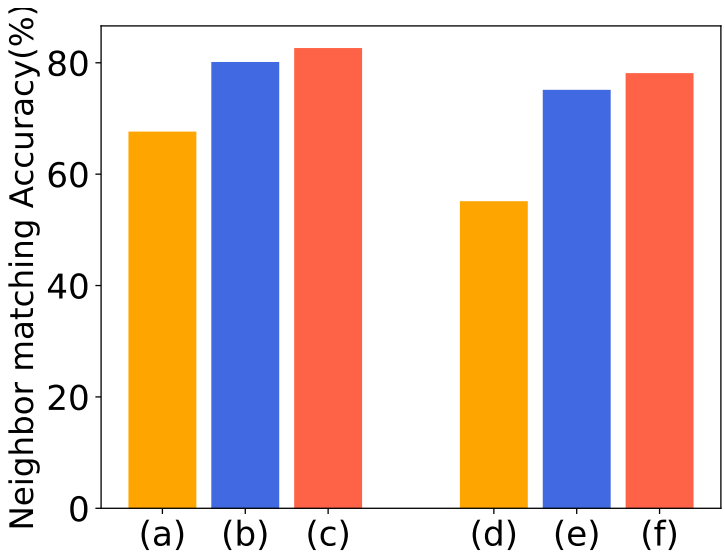


Figure 3.9 Average accuracy (over test set) of neighbor search in LONDN-MRI (MoDL with UNet denoiser) at 4x undersampling in (a) the first iteration (neighbors found with respect to the initial input images $\boldsymbol{x}^0$) and after the (b) first and (c) second iteration. (d)-(f) are corresponding results at 8x undersampling.

Effect of Weight Regularization in LONDN-MRI: Here, we vary the strength of the regularization penalty weight in (3.8) and run LONDN-MRI over the test set at 4x k-space undersampling. Fig. 3.10 plots the average PSNR as a function of the penalty weight for the MoDL network with UNet denoiser. The normalized cross-correlation distance was used during neighbor search, with other parameters as before. The result shows slight benefits for choosing the regularization weight carefully.

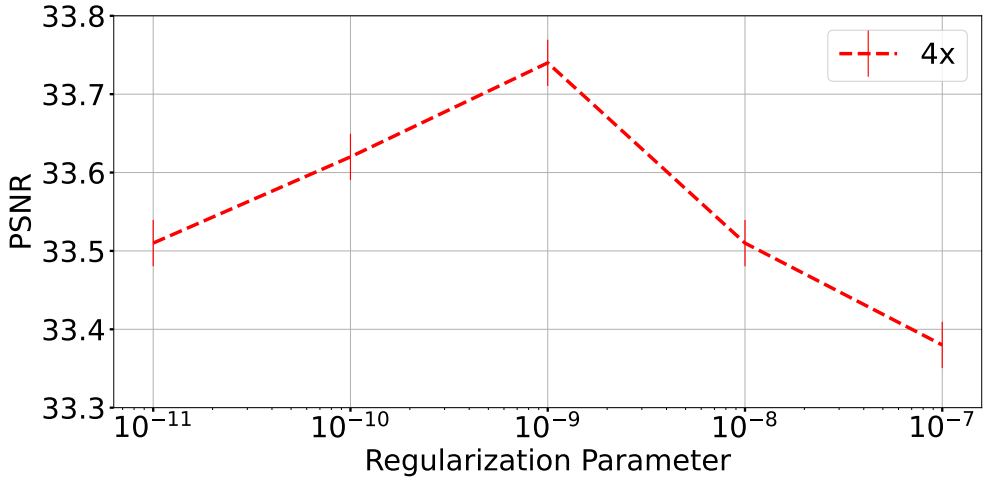


Figure 3.10 Average reconstruction PSNR on the test set at 4x undersampling for different regularization penalty parameters. We used ℓ_1 norm regularization of network weights for an MoDL network with UNet denoiser.

Convergence of Loss in Bilevel Optimization: Next, we study the behavior of the alternating LONDN-MRI algorithm as a heuristic for the bilevel optimization formulation in (3.9). Here, we used an MoDL network with the UNet denoiser and $k = 30$ training pairs were chosen (from 3000 cases) in the local dataset in each iteration of LONDN-MRI. The UNet weights were randomly initialized to begin with, and the neighbor search in the first iteration of LONDN-MRI was performed using $\boldsymbol{x}^0$ and correspondingly generated aliased training images. Fig. 3.11 plots the upper-level loss in (3.9) (in a root mean squared error form) after each iteration of LONDN-MRI for a test image. Here, we ran many iterations to verify convergence. We observe that the loss changes very little after a few iterations and stabilizes. This matches with the behavior of the neighbor search accuracy bar plots. The result indicates that the proposed alternating scheme could be a reasonable heuristic for reducing the loss in the challenging problem (3.9). Finally, we compare the loss values in Fig. 3.11 with an oracle loss, where the upper-level loss in (3.9) is computed using the ground truth test image and its k nearest neighbors. It is clear that the loss values in LONDN-MRI converge very close to the oracle loss, indicating the potential for our scheme.

Effect of Number of Nearest Neighbors on Image Quality: Again, we investigate how the LONDN-MRI algorithm behaves when the number of nearest neighbors is varied to see how it affects the effectiveness of the reconstruction. To test our method, we selected from 10 to 1000

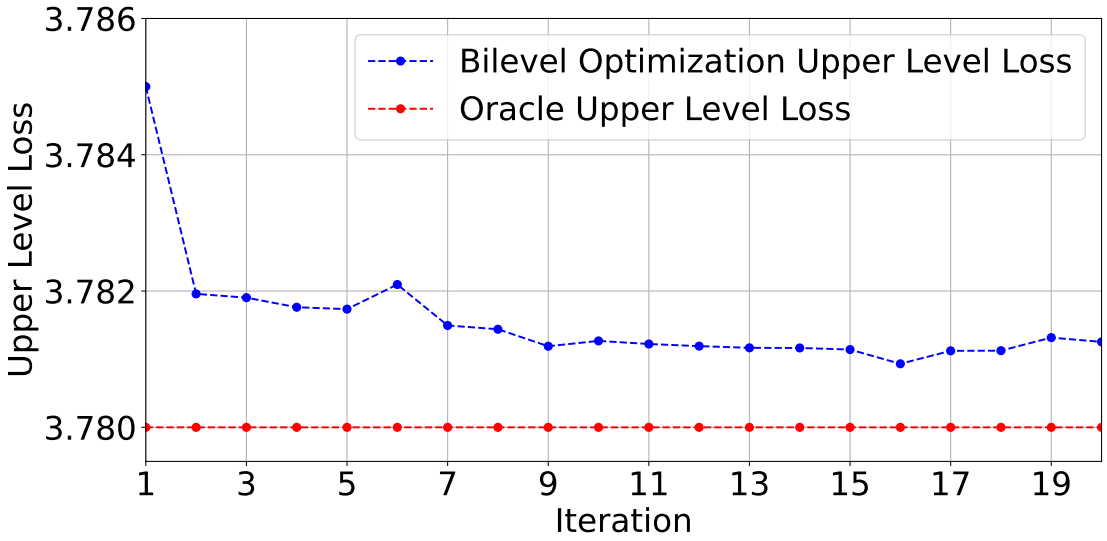


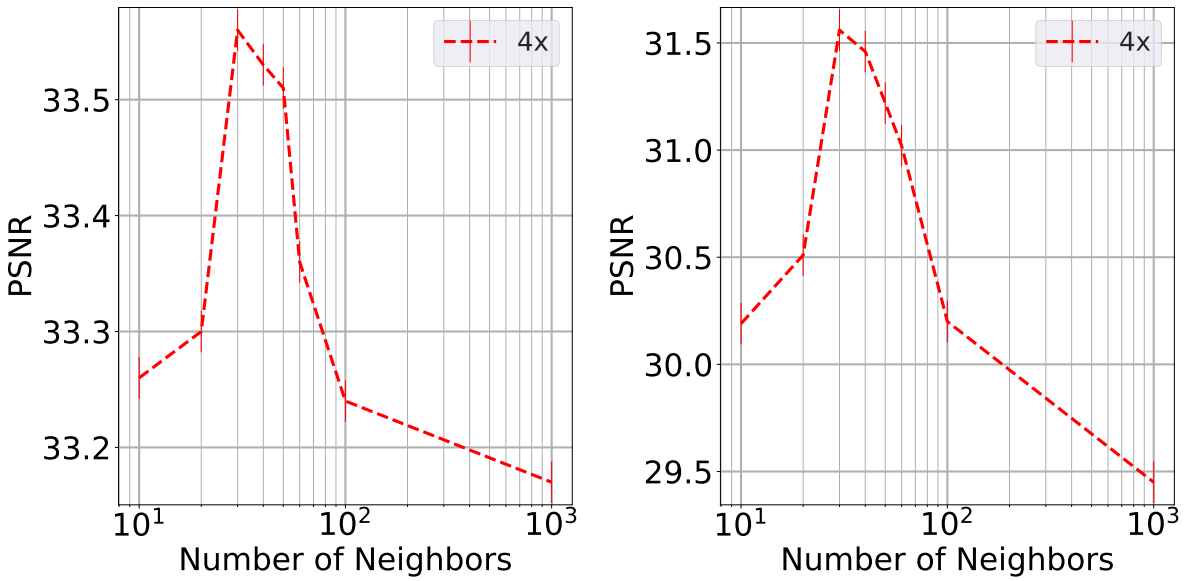
Figure 3.11 Upper-level loss in the bilevel optimization formulation (3.9) plotted over the iterations (after network update step) of the LONDN-MRI scheme at 4x undersampling. We used MoDL with a UNet denoiser and $k = 30$ for neighbor search. In addition, the red line shows an oracle upper-level loss computed using the ground truth test image and its k nearest neighbors.

images for the closest neighbors (with NCC metric). The average test reconstruction PSNR for different cases is shown in Fig. 3.12. Too few local neighbors can make the method prone to overfitting and too many neighbors lead to a lack of scan-specificity and worse performance. 30-50 neighbors provide similar performances.

Time Consumption Trade-offs: To further understand the time efficiency of our method across different neighborhood sizes for practical applicability, we conducted comparative analyses using three models: an image-domain UNet denoiser, MODL with UNet denoiser, and the MODL with DIDN denoiser. The experiments were run on an NVIDIA GeForce RTX A5000 GPU. The PSNR vs. runtime trade-offs depicted in Figure 3.13 shed light on the time consumption for each model configuration. It is observed that some decrease in the number of neighbors leads to reduced time consumption without significantly compromising image quality. In addition, the results show the effectiveness of starting with a pre-trained DIDN model to improved the reconstruction, as it enhances the efficiency of the reconstruction process, reducing it to order of seconds.

3.4.4 Generalizability of LONDN-MRI

Here, we present a series of studies to evaluate the generalizability of LONDN-MRI in diverse testing settings.



(a) Fast MRI case

(b) FSE MRI case

Figure 3.12 Average reconstruction PSNR on the fastMRI and FSE MRI test set at 4x undersampling for different numbers of nearest neighbors.

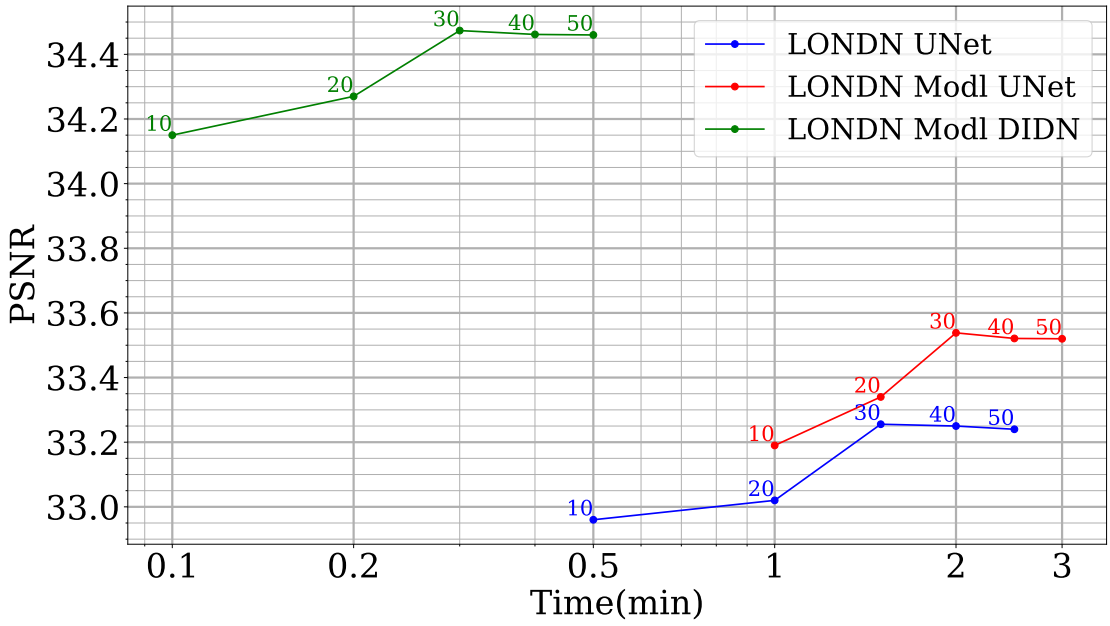


Figure 3.13 PSNR vs. runtime trade-offs of various LONDN-MRI models for the fastMRI knee dataset at 4x k-space undersampling. The models include MoDL networks with UNet or DIDN denoisers, as well as a standalone image-domain UNet. The performance was evaluated across different neighbor sizes, which are shown next to each data point. The processing time for these models ranged from 6 seconds to 3 minutes, depending on the neighbor size. Unrolled networks provided better image quality than the UNet denoiser.

Performance in the Presence of Planted Features: To assess the capability of LONDN-MRI for accurately reproducing image attributes not found in the training set (a common scenario when detecting pathologies, etc.), we embedded artificial features into a knee image from the fastMRI dataset, drawing inspiration from recent work (Lahiri et al., 2021). We performed 4x undersampling in k-space and reconstructed with the MoDL network (with UNet denoiser) that was trained using 3000 images. In Fig. 3.14, we observe that LONDN-MRI produces sharper reconstruction of image features and better PSNR compared to the globally trained network. The details or edges of the planted features are better preserved in LONDN-MRI. Moreover, LONDN-MRI provides similar image quality with and without the planted features (Fig. 3.5), whereas, the globally trained network degrades significantly. This indicates the relatively improved stability and generalizability of the proposed method.

Performance on Data with Lesions: While the previous experiment allowed comparing reconstruction quality with or without planted features, here we test our method on MRI scans with lesions, which are often regions of abnormal or diseased tissue. We utilize the annotated fastMRI+ data to evaluate our method's image reconstruction capabilities, and compare its outcomes with established baselines. For the training phase, the non-lesion dataset was employed for the global training approach with 3000 images whereas LONDN-MRI used 30 adaptively selected images for training (searched from 3000 images). In contrast, during the testing phase, we used 20 scans with lesions. The results, as displayed in Table 3.5, indicate that our method achieves substantially higher PSNR values in comparison to the globally trained baseline as well as the LORAKI method. Furthermore, visualizations in Figure 3.16 clearly demonstrate the superiority of our method, particularly in the nonspecific white matter lesion areas. Thus, both in terms of visual assessment and PSNR values, our approach outperforms the existing baselines and aligns more closely with the ground truth.

Acceleration	Global	LONDN-MRI (1 iteration)	LONDN-MRI (2 iterations)	Oracle	LORAKI
4x	34.37	34.89	**35.1**	35.21	32.89
8x	32.05	32.65	**32.72**	32.77	30.89

Table 3.5 Average reconstruction PSNR values (in dB) for the lesion fastMRI+ test set at 4x and 8x k-space undersampling. LONDN-MRI and the global model were trained on the non-lesion dataset.

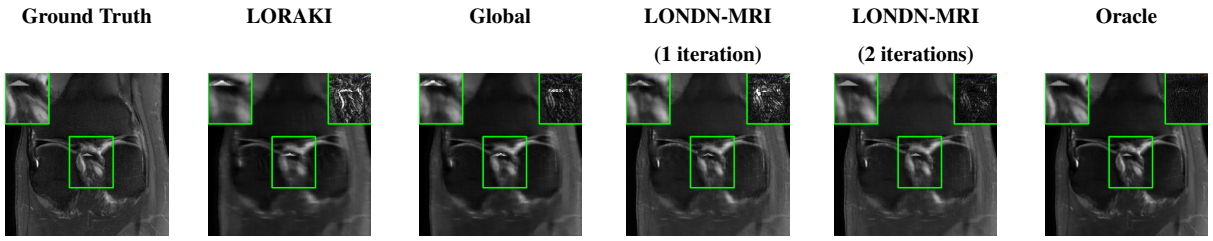


Figure 3.14 Visualization of ground truth and reconstructed images using different methods at 4x k-space undersampling. The central portion (with the planted feature) and its reconstruction error map are shown in the top panels in the images.

Performance without Well-Matched Neighbors: Another natural question is how sensitive is the proposed method to using a 'well matched' (to the test scan) subset of images in the global training set. One might consider this restrictive. To better evaluate the working of LONDN-MRI, we switched its training with the UNet denoiser from using the 30 closest neighbors to using the 31st to the 60th closest (or less similar) neighbors. Fig. 3.20 shows an example with the different near-neighbors that are chosen from the 3000 image global training set, ranked based on NCC distance. While the nearest neighbors look quite similar to the test image, the farther ones could be relatively dissimilar in practice. In this case, LONDN-MRI (with 1 iteration) using the 31st to the 60th closest neighbors still reconstructs the test scans well with an average PSNR of 33.34 dB (at 4x k-space undersampling and 15 test images), which is only slightly worse than when using the 30 closest neighbors (33.46 dB). This indicates the proposed approach may not be very sensitive to availability of highly visually matched training data. Indeed, the Stanford FSE data has more variability than fastMRI and our approach performs well on that dataset.

Evaluating Generalization with Limited Training Sets: To facilitate a fairer comparison with scan-adaptive methods such as DIP, LORAKI and RAKI, we conduct experiments utilizing much smaller subsets of the original fastMRI knee dataset, from which the neighbors in LONDN-MRI are selected. We randomly selected 5 to 100 slices for the overall training set in LONDN-MRI. These were chosen from a small random set of volumes/patients. The goal is to emulate comparisons with DIP, LORAKI, and RAKI when LONDN-MRI operates in a very limited dataset regime. For each overall training set size, we selected the top k similar neighbors at testing time, where k is adjusted

based on the dataset size. For example, for a dataset with 5 slices, we selected the top 3 similar scans at test time, and for a dataset with 100 samples, we selected the top 10 neighbors in the search.

The average reconstruction PSNR for the testing scans, plotted in Fig.3.15, reveals that although there is some decline in performance with decreasing dataset size, the results still surpass those achieved by DIP, RAKI and LORAKI, indicating potential for LONDN-MRI with very limited training sets. While DIP, RAKI and LORAKI adapt purely to the individual test scans without supervision, the LONDN-MRI approach wouldn't make sense in the 0-paired data regime. In future work, we plan to study hybrid methods leveraging both LONDN-MRI and DIP, i.e., adapting the network based on both similar paired data and the current test scan's measurements (as in DIP).

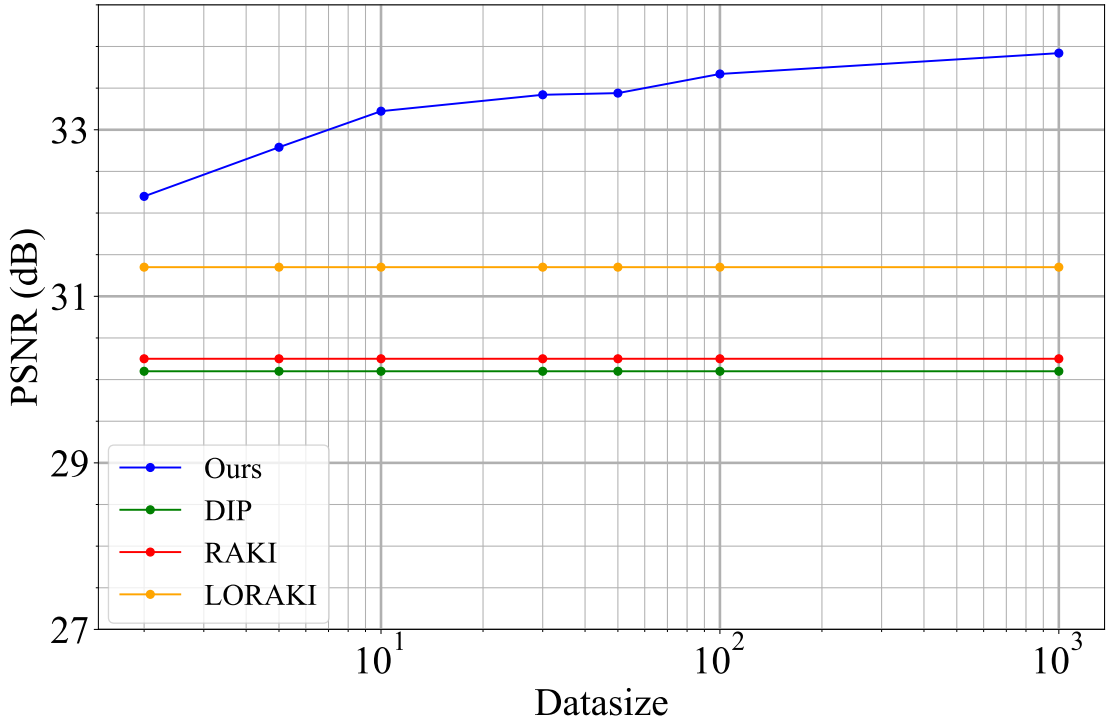


Figure 3.15 Average PSNR on test set (from fastMRI) for LONDN-MRI (MoDL network with UNet denoiser) at 4x k-space undersampling for various dataset sizes. Subsets of the dataset are chosen as neighbors in LONDN-MRI at test time. The average PSNR values with DIP, LORAKI, and RAKI, which require no training data are shown as horizontal lines.

Effect of Varying Scan Settings at Test Time: Since the reconstruction network in LONDN-MRI is trained for each scan, we would like to understand better the benefits this provides in terms of letting the network adapt to distinct scan settings. So we chose the MoDL reconstructor with UNet

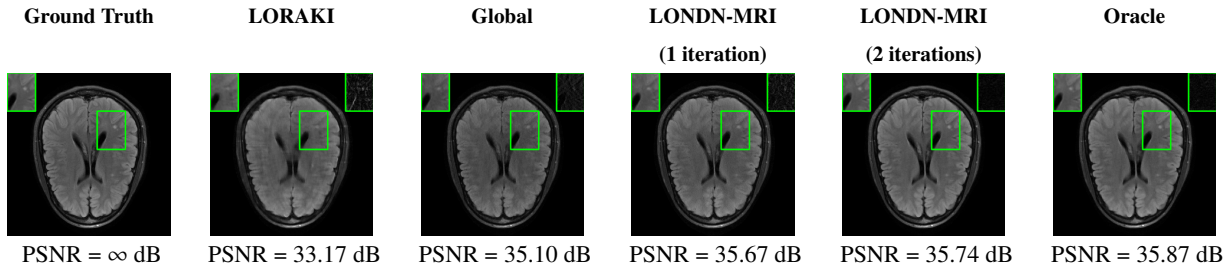


Figure 3.16 Visualization of ground truth and reconstructed images using different methods at 4x k-space undersampling for an annotated image from the fastMRI+ dataset, where the interest area is a nonspecific white matter lesion (in green box).

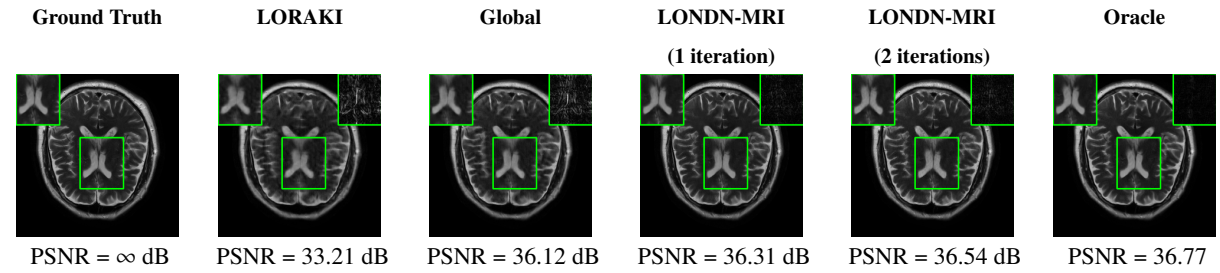


Figure 3.17 Visualization of ground-truth and reconstructed images using different methods at 4x k-space undersampling for a T2 contrast MRI scan (with training on T1 contrast scans). A region of interest (in green box) and its error map are also shown.

denoiser (with same hyperparameters for training as before) and trained it on the 3000 image set in two ways: with a fixed sampling mask across the images (the mask was padded with zeros to account for slight variations in matrix sizes), and with a different random sampling mask for each image. The first setting was used in previous subsections. For LONDN-MRI, here, we used a different random sampling mask for each test scan, but the network was adapted locally with the same mask used across each (small) local training set. Table 3.6 shows the average PSNR values on the test set with these different strategies as well as with the oracle LONDN-MRI scheme. It is clear that the globally learned model with a fixed sampling mask struggles to generalize to the different scan settings at test time. But training the global model with random sampling masks leads to improved reconstruction PSNRs. Importantly, the LONDN-MRI schemes that adapt the reconstruction model to the settings as well as the data for each scan provide marked improvements over both globally learned network settings.

Results with Different Contrasts: To delve deeper into clinical applicability of our method, we

Acceleration	Global Model trained with a fixed mask	Global Model trained with rand. masks	LONDN-MRI (2 iterations)	Oracle LONDN
4x	33.03	33.19	**33.56**	33.64
8x	30.62	30.84	**31.14**	31.22

Table 3.6 Average reconstruction PSNR values (in dB) on the test set at 4x and 8x undersampling. The LONDN-MRI results are compared to training a global model with a fixed sampling mask or with random masks.

conducted further tests to ascertain its adaptability to different contrasts or weightings in scans. Conventional deep learning reconstruction techniques may need consistency in contrast between training and testing to achieve optimal results and could struggle with generalization across varied experimental settings. Our method, being scan-specific, could offer some flexibility because of adaptivity to features in test scans. To further study this, we conducted a test, where the global model was trained exclusively on T1 MRI data at 4x and 8x undersampling using 3000 training scans. Subsequent testing was done on T2 contrast MRI data with 20 images. For LONDN-MRI, we used 30 images for local training (searched from 3000 images) for each test scan. The results, presented as box plots in Fig. 3.18 and visualized with one example in Fig 3.17, highlight our method's reconstruction performance in comparison to the globally trained MoDL network and the scan-specific LORAKI scheme. Our method exhibits notable better performance, underscoring its effectiveness in diverse imaging contexts.

Performance with Different Signal-to-Noise Ratios: To assess the performance of LONDN-MRI when the training and tested data have different signal-to-noise ratios (SNRs), we conducted tests on scans from the fastMRI knee dataset that were subjected to additive random Gaussian noise with a variance of 0.01 for the real and imaginary parts of the noise. The globally and locally trained models at 4x and 8x undersampling used data without added noise, and the training settings were the same as before in Section 3.4.1. Our findings revealed a general decline in reconstruction performance across all methods, attributable to the different SNRs between training and testing. Despite this, LONDN-MRI displays better capability in handling noise perturbations, with a wider

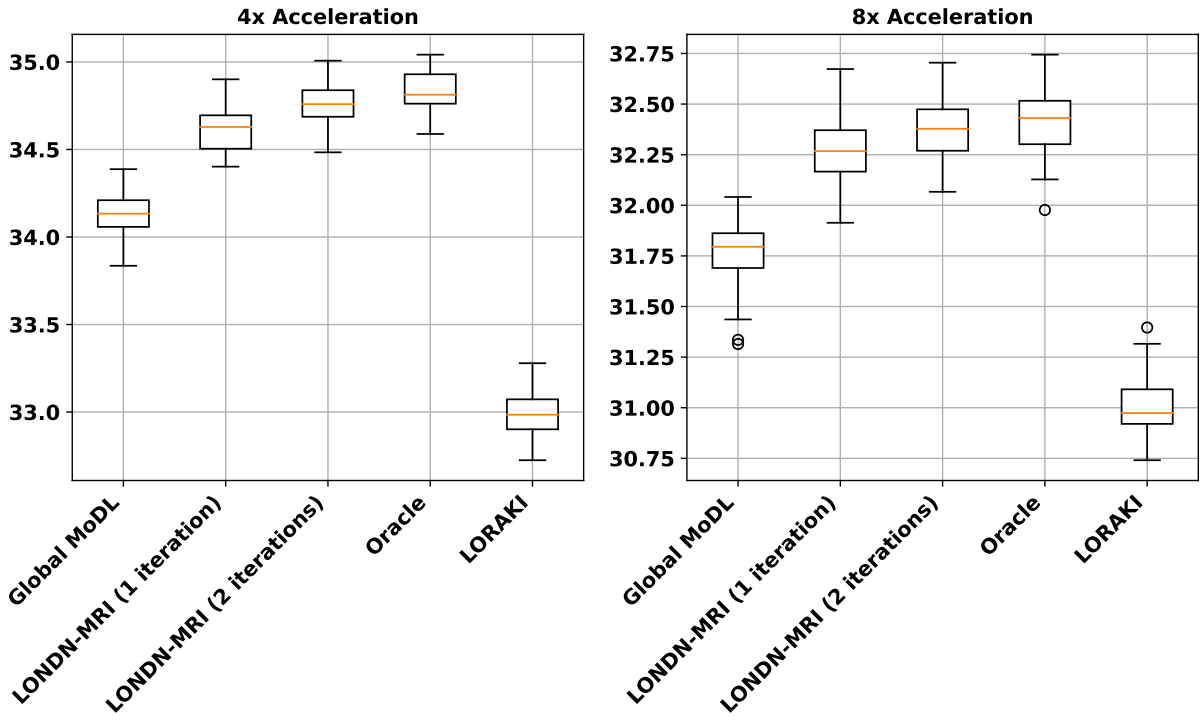


Figure 3.18 Box plots for average reconstruction PSNR values (in dB) for different methods for the T2 fastMRI brain test set at 4x and 8x undersampling. LONDN-MRI (trained on T1 contrast fastMRI dataset) results are compared to a model trained globally (on 3000 T1 contrast scans) and to LORAKI.

performance gap over the globally trained model. This is clear from the PSNR values depicted in the corresponding box plots in Fig. 3.19.

3.5 Discussion

We proposed a novel LONDN-MRI reconstruction technique that efficiently matches test reconstructions to a cluster of a dataset, where networks are adaptively estimated on images most related to a current scan. Our results on the multi-coil fastMRI brain and knee datasets, fastMRI+, and the Stanford FSE dataset showed promise for our patient-adaptive network estimation scheme. The approach does not require pre-training and can thus readily handle changes in the training set. Additionally, the networks in LONDN-MRI can be randomly initialized and trained adaptively on very small datasets, and such networks outperformed models trained globally on much larger datasets (with lengthy training times). For example, for fastMRI knee scans, LONDN-MRI with 2 alternations involving MoDL with a randomly initialized UNet denoiser took 5 minutes to run on a NVIDIA GeForce RTX A5000 GPU (with batchsize of 6 and 200 epochs each time to update

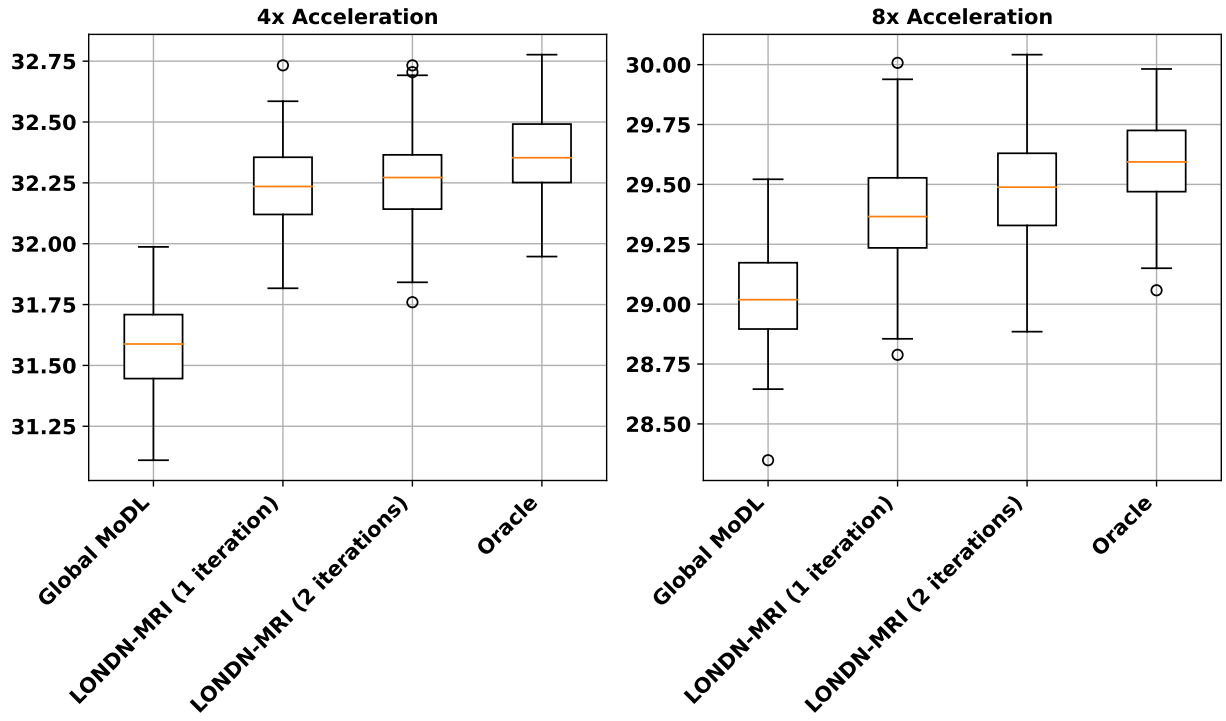


Figure 3.19 Box plots of average reconstruction PSNR values (in dB) for different methods on the fastMRI knee test set at 4x and 8x undersampling. For the test dataset, we added zero-mean Gaussian noise to the measurements with standard deviation $\sigma = 0.01$ for the real and imaginary parts of the noise. All training data used did not include additional noise.

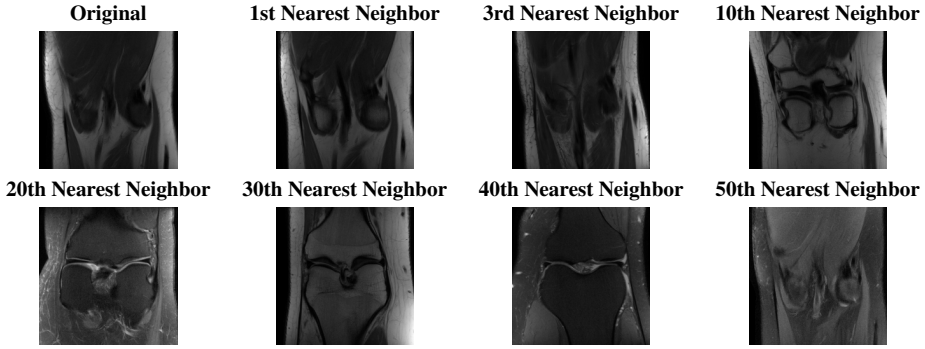


Figure 3.20 An image is shown along with different nearest neighbors from the fastMRI dataset.

networks locally). While LONDN-MRI outperformed the scan-adaptive methods such as DIP, RAKI, and LORAKI in image quality, the runtimes for the methods were somewhat similar. DIP takes about 5 minutes to reach peak performance (over iterations) with the same GPU, while RAKI and LORAKI took 3 mins and 4 mins, respectively. LONDN-MRI requires only a few images (e.g., 30) to train networks, with often 200-250 epochs for locally updating randomly initialized networks such as the UNet. Fewer epochs (often 10 suffices) of update were needed with pre-trained networks

such as the pre-trained DIDN, resulting in runtimes of only 18 seconds per iteration of LONDN-MRI (Fig. 3.13). Of course, a globally trained model would run faster at inference time. For example, MoDL with pre-trained DIDN denoiser takes 8 seconds on average to reconstruct fastMRI knee images. Note that the neighbor search process in the proposed method is highly efficient. We find 20-30 images from 3000 images to train the model in about 10 seconds, while the overall algorithm takes minutes. The neighbor search is also highly parallelizable.

When compared to the supervised global model, the proposed method offers consistently improved reconstruction quality in terms of PSNR, SSIM, and HFEN metrics. Additionally, we demonstrated that the local model adapts better to test time changes (such as changes to the sampling mask, scan contrast, SNR, presence of anomalies, etc.) compared to a globally learned (and fixed) model. Our approach produced marked improvements for the Stanford FSE dataset, and noticeable improvements for fastMRI/fastMRI+. Additionally, our study with different distance metrics revealed they have only slight effect on reconstruction quality. The NCC metric provided the best reconstruction quality and was thus used in our studies. We conjecture that a learned distance metric (Kaya and **c**S. bilge, 2019) could further enhance the performance of LONDN-MRI.

3.6 Conclusions

This paper examined supervised learning of deep unrolled networks at reconstruction time for MRI by exploiting training sets along with local modeling and clustering. We showed advantages for this approach at different k-space undersampling factors over networks learned in a global manner on larger data sets. The training may be connected to a bilevel optimization problem. We also compared different distance metrics for finding neighbors in our approach and regularization to reduce local overfitting. We intend to expand our studies in the future by incorporating non-Cartesian undersampling patterns, such as radial and spiral patterns, as well as deploying them to 3D settings and other imaging modalities. Additionally, the method's generalizability will be further examined, with a particular emphasis on heterogeneous datasets. To handle more extreme training-test data variations such as unseen anatomies, we plan to explore patch-based neighbors in local learning schemes for future work. We showed benefits for both randomly seeded training of simple models

and for fine tuning of sophisticated pre-trained models, and believe our methodology could be applied to a variety of deep learning-based tasks (even beyond image reconstruction) effectively to improve overall performance. Finally, metric learning (Kaya and **c**S. bilge, 2019) to improve local clustering and subsequent network adaptation will be an important future direction.

CHAPTER 4

SELF-GUIDED DEEP IMAGE PRIOR

4.1 Introduction

In last chapter we mentioned the local learning for the MRI reconstruction to avoid the data limit situation. But in the extreme case, we need to reconstruct the MRI image without any dataset. To avoid this kind of situation, we want to proposed the unsupervised learning with the zero shot method by using the deep image prior method. The ability of deep image prior (DIP) to recover high-quality images from incomplete or corrupted measurements has made it popular in inverse problems in image restoration and medical imaging including magnetic resonance imaging (MRI).

However, conventional DIP suffers from severe overfitting and spectral bias effects. In this work, we first provide an analysis of how DIP recovers information from undersampled imaging measurements by analyzing the training dynamics of the underlying networks in the kernel regime for different architectures. This study sheds light on important underlying properties for DIP-based recovery. Current research suggests that incorporating a reference image as network input can enhance DIP's performance in image reconstruction compared to using random inputs. However, obtaining suitable reference images requires supervision, and raises practical difficulties. In an attempt to overcome this obstacle, we further introduce a self-driven reconstruction process that concurrently optimizes both the network weights and the input while eliminating the need for training data. Our method incorporates a novel denoiser regularization term which enables robust and stable joint estimation of both the network input and reconstructed image. We demonstrate that our self-guided method surpasses both the original DIP and modern supervised methods in terms of MR image reconstruction performance and outperforms previous DIP-based schemes for image inpainting.

A recent study (Zhao et al., 2020b) demonstrated the effectiveness of incorporating additional guidance into DIP-based restoration by using a strategically chosen reference image as network input during training. This reference-guided technique considerably enhances reconstruction quality and stability while obviating the need for fully supervised training. Nevertheless, this approach

depends on the availability of an appropriate reference image, which may not always be the case. Additionally, it remains uncertain from (Zhao et al., 2020b) how to effectively select a suitable reference based solely on undersampled measurements of an unknown test image. Inspired by the ability of reference-based guidance to improve the performance of DIP reconstruction, we consider the setting where absolutely no reference or training data is available.

4.1.1 Contributions

We summarize the paper's main contributions as follows.

• To gain a deeper understanding of image reconstruction using DIP, we conduct an analysis of gradient descent-trained CNNs in the over-parameterized regime. We employ a realistic imaging forward operator instead of a Gaussian measurement matrix for our analysis of the case of compressed sensing. Our primary finding is that as the number of gradient descent steps used to optimize the standard DIP objective function approaches infinity, the difference between the network estimate and the ground truth will reside in a subspace related to the null space of the forward operator and the network's neural tangent kernel.

• The choice of network architecture significantly affects the ability of DIP to recover the image in compressed sensing tasks. We demonstrate both theoretically and empirically that certain generator architectures will have greater difficulty recovering missing information or frequencies than others.

• We propose a self-guided DIP method, which eliminates the need for separate reference images (for network input) and gives much better image reconstruction quality than the prior reference-guided method as well as several other related and competing schemes. The proposed method relies on a crucial denoising-based regularization.

4.1.2 Image reconstruction problem

To ensure accurate image reconstruction, an ill-posed inverse problem can be formulated as:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathcal{R}(\boldsymbol{x}), \tag{4.1}$$

where $\boldsymbol{A}$ is a linear measurement operator, $\boldsymbol{y} \in \mathbb{R}^p$ are the measurements, and $\hat{\boldsymbol{x}} \in \mathbb{R}^q$ is the reconstructed image. The first term in the minimization is referred to as a data-fidelity function and

can also take on alternative forms depending on imaging setup. In classical image inpainting, $\boldsymbol{A}$ is a binary masking operator. For the task of reconstructing a multi-coil MRI image, represented by $\boldsymbol{x} \in \mathbb{C}^q$ the optimization problem is

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{c=1}^{N_c} \|\boldsymbol{A}_c \boldsymbol{x} - \boldsymbol{y}_c\|_2^2 + \lambda \mathcal{R}(\boldsymbol{x}), \tag{4.2}$$

where the k-space measurements taken from N_c coils are represented by $\boldsymbol{y}_c \in \mathbb{C}^p,\ c = 1, \ldots, N_c$. The coil-wise forward operator is denoted as $\boldsymbol{A}_c = \mathbf{M}\boldsymbol{\mathcal{F}}\mathbf{S}_c$, where $\mathbf{M} \in \{0,1\}^{p \times q}$ is a masking operator that captures the data sampling pattern in k-space, $\boldsymbol{\mathcal{F}} \in \mathbb{C}^{q \times q}$ is the Fourier transform operator, and $\mathbf{S}_c \in \mathbb{C}^{q \times q}$ represents the cth coil-sensitivity map (a diagonal matrix).

An explicit regularizer $\mathcal{R}(\cdot)$ is employed to limit the solutions to the domain of desirable images. Various regularizers have been used in image reconstruction. For example, it can be the ℓ_1 penalty on wavelet coefficients, a total variation penalty, patch-based sparsity in learned dictionaries, or as in our technique, a denoising type regularization involving e.g., a convolutional neural network (CNN).

4.1.3 Deep Image Prior for Image Reconstruction

Image reconstruction using DIP is typically formulated as:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{A}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{y}\|_2^2, \quad \hat{\boldsymbol{x}} = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{z}), \tag{4.3}$$

Here, $\boldsymbol{f}$ is a neural network with parameters $\boldsymbol{\theta}$, and $\boldsymbol{z}$ is a typically fixed network input that is randomly chosen (e.g., a random Gaussian vector or tensor). We will refer to this formulation as “vanilla DIP" in this work.

4.1.4 Neural Tangent Kernel Analysis for Image Reconstruction

The Neural Tangent Kernel (NTK) (Jacot et al., 2018) is a mathematical tool used to analyze the training dynamics of neural networks, particularly in the infinite-width setting. It provides an approximation of the function space explored by a neural network during gradient-based training, such as gradient descent or stochastic gradient descent.

When trained with gradient descent, a network’s weights are updated according to the equation:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}_t), \tag{4.4}$$

where $\boldsymbol{w}$ are the trainable network parameters at a certain training iteration t, η is a step size parameter, and $\mathcal{L}$ represents the loss function to be minimized. Rearranging equation (4.4) then gives

$$\frac{\boldsymbol{w}_{t+1} - \boldsymbol{w}_t}{\eta} = -\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}_t). \tag{4.5}$$

If η is small, this approximates the differential equation

$$\frac{d\boldsymbol{w}}{dt} = -\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}). \tag{4.6}$$

Because the network input is fixed in the vanilla DIP setting, we can view the network output $\boldsymbol{z}$ as a function of $\boldsymbol{w}$. Applying the chain rule yields

$$\frac{d\boldsymbol{z}(\boldsymbol{w})}{dt} = \nabla \boldsymbol{z}(\boldsymbol{w})^T \frac{d\boldsymbol{w}}{dt}. \tag{4.7}$$

Substituting the loss from equation (4.3) into equations (4.6) and (4.7) gives

$$\frac{d\boldsymbol{z}(\boldsymbol{w})}{dt} = -\nabla \boldsymbol{z}(\boldsymbol{w})^T \nabla \boldsymbol{z}(\boldsymbol{w}) \boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{z}(\boldsymbol{w}) - \boldsymbol{y}). \tag{4.8}$$

The critical assumption of NTK theory is that the matrix $\boldsymbol{W} := \nabla \boldsymbol{z}(\boldsymbol{w})^T \nabla \boldsymbol{z}(\boldsymbol{w})$ – called the neural tangent kernel – remains fixed throughout training. In this regime, equation (4.8) can be rediscretized to show that the training dynamics of DIP for MRI reconstruction will reduce to

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \eta \boldsymbol{W} (\boldsymbol{A}^T \boldsymbol{y} - \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{z}_t). \tag{4.9}$$

We start gradient descent from a random initialization $\boldsymbol{\theta}_0 \sim \mathcal{N}(\boldsymbol{0}, \omega \boldsymbol{I})$.

In our analysis, we make the simplifying assumption that $\boldsymbol{z}_0 = \boldsymbol{0}$. This assumption is not unique to our analysis, and is consistent with prior literature (Tachella et al., 2020). To understand this assumption, we first note that since all network parameters $\boldsymbol{w}$ are initialized from mean $\boldsymbol{0}$ distributions, the initial output $\boldsymbol{z}_0$ is $\boldsymbol{0}$ in expectation over this initialization. However, it is possible that $\boldsymbol{z}_0$ may not be $\boldsymbol{0}$ for any particular instantiation of $\boldsymbol{w}$. To correct for this, we can consider a slightly modified network such that this assumption holds. Namely, for any particular $\boldsymbol{f}_{\boldsymbol{\theta}}$ with random input $\boldsymbol{z}$, it will have initial output $\boldsymbol{z}_{\text{init}} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})$. We can then define the slightly modified

network $\tilde{\boldsymbol{f}}_{\boldsymbol{\theta}}(\boldsymbol{z}) = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{z}_{\text{init}}$. This modification ensures that $\boldsymbol{z}_0 = \boldsymbol{0}$. Moreover, this modification has little other effect on the analysis, since $\boldsymbol{f}$ and $\tilde{\boldsymbol{f}}$ have the same NTK, because $\boldsymbol{z}_{\text{init}}$ is a constant.

With these preliminaries, we now state our first theorem on the training dynamics of DIP for image reconstruction.

Theorem 4.1.1. Let $\boldsymbol{A} \in \mathbb{R}^{p \times q}$ be a full row rank forward operator. Assume $\boldsymbol{z}_0 = \boldsymbol{0}$ and let $\boldsymbol{z}^\infty$ be the reconstruction as the number of training iterations approaches infinity. Let $\boldsymbol{x} \in \mathbb{R}^q$ be the ground truth image and let $\boldsymbol{W}$ be the NTK of the reconstructor network. Further suppose that the acquired measurements are free of noise so that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. If the learning rate satisfies $\eta < \frac{2}{\|\boldsymbol{B}\|}$, where $\boldsymbol{B} := \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{W}^{\frac{1}{2}}$, then

- If the NTK kernel $\boldsymbol{W}$ is non-singular, then the difference between $\boldsymbol{z}^\infty$ and $\boldsymbol{x}$ lies in the null space $N(\boldsymbol{A})$ of $\boldsymbol{A}$, i.e.,

$$\boldsymbol{z}^\infty - \boldsymbol{x} \in N(\boldsymbol{A}). \tag{4.10}$$

  Moreover, as long as $P_{N(\boldsymbol{A})}\boldsymbol{x} \neq 0$, the reconstruction error $\boldsymbol{z}^\infty - \boldsymbol{x} \neq \boldsymbol{0}$. Here, $P_{N(\boldsymbol{A})}$ is the projector onto the subspace $N(\boldsymbol{A})$.

- If the NTK $\boldsymbol{W}$ is singular, and $P_{N(\boldsymbol{A}) \cap R(\boldsymbol{W})}\boldsymbol{x} = \boldsymbol{0}$ with $R(\boldsymbol{W})$ denoting the column or range space of $\boldsymbol{W}$, then the difference $\boldsymbol{z}^\infty - \boldsymbol{x}$ will be linear in $P_{N(\boldsymbol{W})}\boldsymbol{x}$, which is the component of $\boldsymbol{x}$ lying in the null-space of $\boldsymbol{W}$,

$$\boldsymbol{z}^\infty - \boldsymbol{x} = -P_{N(\boldsymbol{W})}\boldsymbol{x} + \boldsymbol{W}^{\frac{1}{2}} (\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}})^\dagger \boldsymbol{A} P_{N(\boldsymbol{W})}\boldsymbol{x}. \tag{4.11}$$

- If the NTK $\boldsymbol{W}$ is singular, $P_{N(\boldsymbol{A}) \cap R(\boldsymbol{W})}\boldsymbol{x} = \boldsymbol{0}$ and $x \in R(\boldsymbol{W})$, then the reconstruction is exact or

$$\boldsymbol{z}^\infty = \boldsymbol{x}. \tag{4.12}$$

In practice, the NTK matrix isn't precisely singular (or low-rank). Nevertheless, the above theorem can be extended to the nearly singular case with some error correction terms. The principal message, however, remains consistent, as outlined below.

When the NTK kernel $\boldsymbol{W}$ is non-singular, it can result in inaccurate reconstruction (4.10) at convergence. On the other hand, if the NTK kernel $\boldsymbol{W}$ is singular or say low-rank, then surprisingly, there is a possibility of exact reconstruction. Specifically, (4.12) indicates that exact recovery is possible if the NTK operator effectively represents the underlying image, meaning $\boldsymbol{x} \in R(W)$ and if the measurement matrix $\boldsymbol{A}$ exhibits sufficient incoherence with the NTK, in the sense that $N(\boldsymbol{A}) \cap R(\boldsymbol{W}) = \emptyset$ or $P_{N(\boldsymbol{A})\cap R(\boldsymbol{W})}\boldsymbol{x} = \boldsymbol{0}$. An example of a situation that meets these criteria is that the true image $\boldsymbol{x}$ consists of a few non-bandlimited wavelet elements, the NTK kernel $\boldsymbol{W}$ is sufficient to represent this $\boldsymbol{x}$, and $\boldsymbol{A}$ includes a range of low-frequency Fourier modes. Provided that the wavelet elements constituting $\boldsymbol{x}$ cannot be linearly combined to form a band-constrained signal, they would not be included in the kernel of $\boldsymbol{A}$, which consists of such signals. Consequently, the condition $P_{N(\boldsymbol{A})\cap\text{range}(\boldsymbol{W})}\boldsymbol{x} = \boldsymbol{0}$ would be met.

Now consider the more practical scenario when the NTK kernel $\boldsymbol{W}$ is almost singular (but not exactly). We note that empirical studies suggest that the NTK of reasonably sized networks is generally poorly conditioned (with a condition number of 10^3 or greater) (Liu and Hui, 2023). Taking Theorem 4.1.1 into account, we can anticipate certain interesting outcomes. First of all, despite $\boldsymbol{W}$ being nearly singular, it retains full rank. Therefore, as per the result for the non-singular NTK outlined in equation (4.10), the reconstruction will incur a non-zero (likely non-negligible) error in $N(\boldsymbol{A})$ (e.g., MRI images invariably contain frequency content outside the sampled frequencies). However, this substantial reconstruction error will only emerge if the algorithm is allowed to converge fully over a sufficiently long duration. In the early stages of the iterations, however, the near singular nature of $\boldsymbol{W}$ and the use of the gradient descent algorithm imply that the larger elements of $\boldsymbol{W}$ will predominantly influence the gradient directions. As a result, the initial reconstructions will closely resemble those in scenarios with a singular or low-rank $\boldsymbol{W}$. According to the third statement of Theorem 4.1.1, under certain conditions, this leads to minimal reconstruction errors. Therefore, it is plausible to observe a pattern where reconstruction errors initially decrease significantly in the early iterations, before increasing (towards the level indicated by (4.10)) after a prolonged period. A full proof of the theorem is provided in Appendix A.1.

4.1.5 Extending Our Analysis to the Noisy Setting

The proof of Theorem 4.1.1 requires the assumption that the measurements $\boldsymbol{y}$ are free of noise, or that we have $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ exactly. In this section, we extend our analysis to the more realistic setting where the acquired imaging measurements are corrupted by noise, i.e., the acquired measurements are of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \mathbf{n}$ with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.

In this case, we estimate the mean squared error (MSE) of the reconstruction through the decomposition MSE = $||\mathbf{Bias}||_2^2$ + Variance (Tachella et al., 2020).

We first investigate the **Bias** term. In Appendix B, we use the recursion in equation (4.9) to show that

$$||\mathbf{Bias}_t||_2^2 = ||\mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t] - \boldsymbol{x}||_2^2 = ||(\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T\boldsymbol{A})^t \boldsymbol{x}||_2^2. \tag{4.13}$$

We also compute the covariance of $\boldsymbol{z}_t$, $\mathbf{Cov}_t$ and find that:

$$\mathbf{Cov}_t = \mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t \boldsymbol{z}_t^T] - \mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t]\mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t]^T \tag{4.14}$$

$$= \sigma^2 (\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T\boldsymbol{A})^t)\boldsymbol{A}^\dagger (\boldsymbol{A}^\dagger)^T (\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W})^t). \tag{4.15}$$

If we define $\boldsymbol{Q}_t := (\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T\boldsymbol{A})^t)\boldsymbol{A}^\dagger$, then we can write:

$$\mathrm{Var}_t = \mathrm{tr}(\mathbf{Cov}_t) = \sigma^2 \mathrm{tr}(\boldsymbol{Q}_t \boldsymbol{Q}_t^T) = \sigma^2 \sum_{i=1}^{p} \nu_{t,i}^2, \tag{4.16}$$

where $\nu_{t,i}$ are the singular values of $\boldsymbol{Q}_t$.

Theorem 4.1.2. Let $\boldsymbol{A} \in \mathbb{R}^{p \times q}$ be a full row rank measurement operator. Suppose that the acquired measurements are $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \mathbf{n}$, where $\boldsymbol{x}$ is the ground truth image and $\mathbf{n} \in \mathbb{R}^p$ with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Then the MSE for DIP based reconstruction at iteration t is given by

$$\mathrm{MSE}_t = ||(\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T\boldsymbol{A})^t \boldsymbol{x}||_2^2 + \sigma^2 \sum_{i=1}^{p} \nu_{t,i}^2, \tag{4.17}$$

where $\nu_{t,i}$ are the singular values of the matrix $(\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T\boldsymbol{A})^t)\boldsymbol{A}^\dagger$.

A full proof of the theorem is provided in Appendix A.2. As a corollary to Theorem 4.1.2, we consider a special case in the setting of MRI reconstruction. Since MR images are complex-valued, in practice it is common to use a network in DIP with real-valued input, real-valued weights, and a

two-channel output, representing the real and imaginary components of the reconstructed image. The following corollary considers this setting with single-coil MRI. Note that the typical MRI measurement operator mapping a complex-valued image to complex-valued measurements could readily be rewritten as a mapping from/to the stacked real and imaginary parts of the vectors.

Corollary 1. We consider the single-coil MRI forward operator $\boldsymbol{A} = \mathbf{M}\boldsymbol{\mathcal{F}}$. Suppose the network outputs vectors in $\mathbb{R}^{2q}$, representing the real and imaginary parts of the reconstruction concatenated together. Further suppose that the NTK, which we write as $\tilde{\boldsymbol{W}} \in \mathbb{R}^{2q \times 2q}$ has an eigendecomposition of the form:

$$\tilde{\boldsymbol{W}} = \tilde{\boldsymbol{\mathcal{F}}}^T \tilde{\boldsymbol{\Lambda}} \tilde{\boldsymbol{\mathcal{F}}}; \qquad \tilde{\boldsymbol{\Lambda}} = \begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} \end{bmatrix}, \tag{4.18}$$

where $\tilde{\boldsymbol{\mathcal{F}}} = \begin{bmatrix} \boldsymbol{\mathcal{F}}_R & -\boldsymbol{\mathcal{F}}_I \\ \boldsymbol{\mathcal{F}}_I & \boldsymbol{\mathcal{F}}_R \end{bmatrix}$, and $\boldsymbol{\mathcal{F}}_R$ and $\boldsymbol{\mathcal{F}}_I$ are the real and imaginary parts of the Fourier transform operator. Then the MSE at iteration t is given by:

$$MSE_t = \sum_{i=1}^{q} [(1 - \eta\lambda_i m_i)^{2t} |(\boldsymbol{\mathcal{F}}\boldsymbol{x})_i|^2 + \sigma^2 (1 - (1 - \eta\lambda_i m_i)^t)^2], \tag{4.19}$$

where λ_is are the diagonal entries of $\boldsymbol{\Lambda}$, m_i denotes the ith diagonal entry of $\mathbf{M}^T\mathbf{M}$, and $(\boldsymbol{\mathcal{F}}\boldsymbol{x})_i$ is the ith entry of $\boldsymbol{\mathcal{F}}\boldsymbol{x}$.

The above structure for $\tilde{\boldsymbol{W}}$ has a natural interpretation: applying $\tilde{\boldsymbol{W}}$ to a vector in $\mathbb{R}^{2q}$ can be seen to be equivalent to applying the matrix $\boldsymbol{W} = \boldsymbol{\mathcal{F}}^H \boldsymbol{\Lambda} \boldsymbol{\mathcal{F}}$ to the corresponding complex vector in $\mathbb{C}^q$. Thus, the setting corresponds to an equivalent circulant $\boldsymbol{W}$, whose eigenvectors are fully coherent with the Fourier forward operator.

Furthermore, we can interpret equation 4.19 in the limit as $t \to \infty$. In this limit, the first term in the sum will tend to 0 for all sampled frequencies, provided η is sufficiently small, and it is a constant $|(\boldsymbol{\mathcal{F}}\boldsymbol{x})_i|^2$ at nonsampled frequencies (a result of coherence between $\boldsymbol{W}$ and measurement operator $\boldsymbol{\mathcal{F}}$ similar to what is described in section 4.1.6). On the other hand, the second term is 0 for all of the *unsampled* frequencies, and will tend to σ^2 for the sampled frequencies. This behavior indicates that we expect a bias-variance tradeoff, where the bias decreases as $t \to \infty$, the variance

increases as $t \to \infty$, and the optimal performance is achieved for some intermediate t. A full proof of the corollary is provided in Appendix A.3.

4.1.6 Example of the Relationship Between the NTK and the Forward Operator

Theorem 4.1.2 and Corollary 1 show that the training dynamics of DIP for inverse problems such as MRI are largely governed by the relationship between the forward operator $\boldsymbol{A}$ and the NTK $\boldsymbol{W}$. In this section, we analyze a simple network architecture to theoretically demonstrate how this relationship affects the network's ability to recover missing frequency content.

In (Heckel and Soltanolkotabi, 2020), the authors analyze simple generator networks G of the form

$$G_{\boldsymbol{C}}(\,\cdot\,) = \mathrm{ReLU}(\boldsymbol{U}\boldsymbol{C}(\,\cdot\,))\boldsymbol{v}, \tag{4.20}$$

where $\boldsymbol{C} \in \mathbb{R}^{n\times k}$ is a weight matrix, $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is a convolution operator, and $\boldsymbol{v} \in \mathbb{R}^k$ is a vector with $\boldsymbol{v} = \frac{1}{\sqrt{k}}\left[1,\ldots,1,-1,\ldots,-1\right]^T$ with half of its entries 1 and half -1, which represent fixed last layer weights of the generator. It is then proven that in expectation

$$\mathbb{E}[\boldsymbol{W}] = \sum_{l=1}^{k} v_l^2 \mathbb{E}\left[\sigma'(\boldsymbol{U}\boldsymbol{c}^{(l)})\sigma'(\boldsymbol{U}\boldsymbol{c}^{(l)})^T\right] \odot \boldsymbol{U}\boldsymbol{U}^T, \tag{4.21}$$

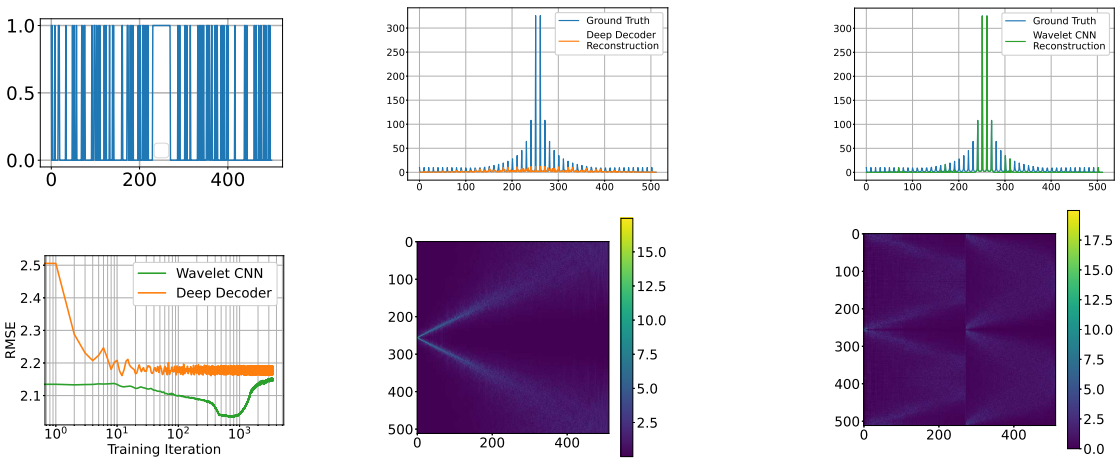
where σ' is the derivative of the ReLU activation, $\odot$ denotes the entry-wise product, v_l is the lth entry of $\boldsymbol{v}$, and $\boldsymbol{c}^{(l)}$ is the lth column of $\boldsymbol{C}$. It is then shown that

$$[\mathbb{E}[\boldsymbol{W}]]_{i,j} = \frac{1}{2}\left(1 - \cos^{-1}\left(\frac{\langle \mathbf{u}_i, \mathbf{u}_j\rangle}{\|\mathbf{u}_i\|_2\|\mathbf{u}_j\|_2}\right)/\pi\right), \tag{4.22}$$

where $\mathbf{u}_r$ denotes the rth row of $\boldsymbol{U}$. In this case, with circulant $\boldsymbol{U}$, it is possible to show that $\boldsymbol{W}$ is also circulant, and hence is diagonalized by Fourier operators. Thus, networks of this form are related to the case in Corollary 1 (in expectation).

4.1.7 Understanding DIP-MRI with Different Networks

In Section 4.1.6, we saw that Corollary 1 suggests that (in expectation) generator networks of the form (4.20) will not be able to effectively recover missing measurement frequencies when used for DIP reconstruction. To empirically validate this claim and compare network designs, we present a simple experiment comparing two neural network architectures for a 1D signal reconstruction task.



Magnitude of Fourier Transform of Deep Decoder NTK

Magnitude of Fourier Transform of Wavelet CNN NTK

Figure 4.1 Top row: The performance of reconstructing a 1D square signal using a Wavelet CNN and Deep Decoder. The top row shows the sampling mask applied in Fourier space (top left), and the Fourier transform of the signals recovered by the Deep Decoder (top middle), and Wavelet CNN (top right) where the y-axis is the magnitude and x-axis indexes the entries of the signal. Bottom row left: The RMSE of the reconstructed signals vs. the number of training iterations. Bottom row right: The two figures at the bottom display the Fourier transform of the left eigenvector matrix for both the Deep Decoder (bottom left) and the WCNN (bottom right).

We compare the Deep Decoder, a simple generator network described in (Heckel and Hand, 2019), and a U-Net architecture, where the upsampling and downsampling filters were replaced by wavelet transformations. The results of this experiment are shown in Figure 4.1. We found that the training on NTK regime in practice, the networks showed little change over training iterations. We also observed that the signal was very close to the subspace $R(\boldsymbol{W})$ when a low-rank approximation of $\boldsymbol{W}$ (obtained by truncating small singular values) was used.

We find that the reconstruction performance of the deep decoder quickly plateaus with the $\boldsymbol{A} = \mathbf{M}\boldsymbol{\mathcal{F}}$ sampling operator, and it is not able to recover significant missing frequency content. In contrast, the error of wavelet-based U-Net reconstruction slowly decreases, then increases after many training iterations because of overfitting. We also plot the magnitude of the Fourier transform of each network's NTK's eigenvectors. We can see that the deep decoder's NTK (the eigenvectors) is highly coherent with the Fourier basis, whereas the wavelet U-Net's NTK is less so. This experiment demonstrates that the analysis and discussion presented in Section 4.1.4 is based on reasonable assumptions, and our conclusions hold when applied to real networks.

4.1.8 Overfitting and Spectral Bias in Deep Image Prior

Because DIP typically uses corrupted and/or limited data for network training, any related distortions will inevitably manifest in the network's output if it is trained until the loss function reaches equilibrium. This issue impacts not only DIP's performance in well-researched areas like image denoising, but also in inverse problems such as MRI reconstruction, where forward operators may have high dimensional null spaces. Fig. 4.2 quantitatively demonstrates the overfitting phenomenon in MRI reconstruction. One can see that the reconstruction reaches peak performance quickly and then slowly diminishes as the training persists. This highlights the necessity for implementing an early stopping criterion when using vanilla DIP to solve inverse problems.

4.1.9 Understanding Spectral Bias and Overfitting for DIP MRI

To gain insights into the spectral bias inherent in vanilla DIP MRI image reconstruction, we utilize a frequency band metric to explore the disparity between the reconstructed frequencies and the actual ones. We compare the multi-coil k-space of the output image $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})$ at every stage of network training to the fully-sampled the k-space $\boldsymbol{y}_c, c = 1, ..., N_c$ of the ground truth image $\boldsymbol{x}$ to study how various frequency components converge (refer to Fig. 4.2). We execute this by calculating a normalized error metric for low, medium, and high-frequency bands:

$$\text{NMSE} := \frac{\sum_{c=1}^{N_c} \left\| \mathbf{M}_{\mathbf{freq}} \boldsymbol{\mathcal{F}} \mathbf{S}_c \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \mathbf{M}_{\mathbf{freq}} \boldsymbol{y}_c \right\|_2^2}{\sum_{c=1}^{N_c} \left\| \mathbf{M}_{\mathbf{freq}} \boldsymbol{y}_c \right\|_2^2} \tag{4.23}$$

where $\mathbf{M}_{\mathbf{freq}}$ is the frequency band mask. Intuitively, the above metric measures the consistency between the reconstructed image $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})$ and the true k-space $\boldsymbol{y}_c$ in the frequency domain. Fig. 4.2 plots this metric computed across three frequency bands for vanilla DIP MRI reconstruction. The result shows that the low frequencies are learned more quickly and with lower error, confirming that spectral bias is present in MRI reconstruction using DIP.

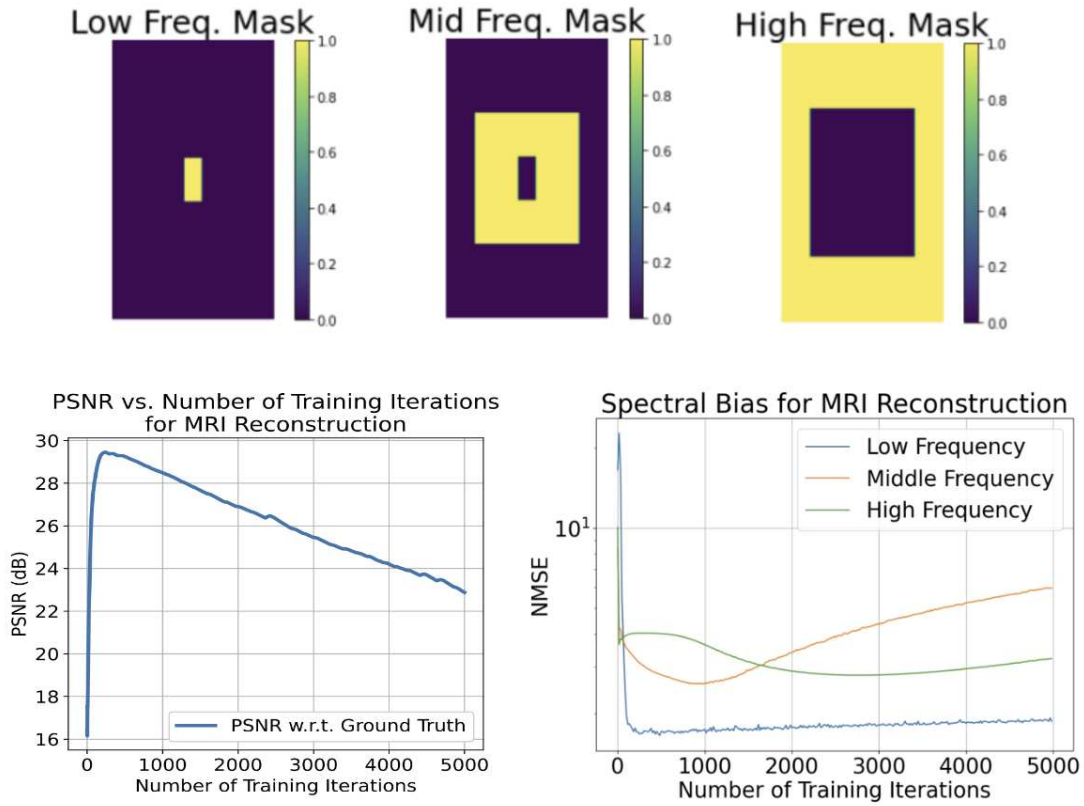


Figure 4.2 Top row: the three masks used to compute the frequency band-based metric. Bottom row: reconstruction PSNR plot on the left illustrates the overfitting issue that occurs during MRI reconstruction. Spectral bias also affects the performance of DIP for MRI reconstruction (right plot), as different frequency bands are reconstructed at different rates.

4.2 Methodology

To more effectively address the overfitting issue inherent in the vanilla deep image prior (DIP), certain methods have been introduced including using matched references. In contrast to the approach of using a reference image, we propose the introduction of a self-regulation method as an enhancement.

4.2.1 Reference-Guided DIP

The reference-guided DIP formulation was proposed in (Zhao et al., 2020b) as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{A}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{y}\|_2^2, \qquad \hat{\boldsymbol{x}} = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{z}). \tag{4.24}$$

This formulation is identical to the problem in (4.3), except that the input to the network is no longer fixed random noise, but is instead a reference image that is very similar to the one being

reconstructed. The input to the network introduces some additional structural information, and we can consider the network as essentially performing image refinement or style transfer rather than image generation from scratch. This method is quite reasonable in cases where a dataset of structurally similar images is available, and there is a systematic way to choose the network input image from the dataset based on only undersampled k-space observations at testing time.

In (Zhao et al., 2020b), the input image seems to be chosen by hand. As a more realistic modification of this method, we propose an approach similar to the recent LONDN-MRI (Liang et al., 2024b) method to search for the reference image (using a distance metric such as Euclidean distance or other metric) that is most similar to an estimated test reconstruction from undersampled data. In our experiments, we used $\boldsymbol{A}^H\boldsymbol{y}$ as estimated test image, and used corresponding versions of reference images to find the closest neighbor.

4.2.2 Self-Guided DIP

To circumvent the need for a prior chosen reference to guide DIP, we introduce the following method, which adaptively estimates such a reference that we call **self-guided DIP**:

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{\theta}, \boldsymbol{z}} \underbrace{\|\boldsymbol{A}\mathbb{E}_{\boldsymbol{\eta}}[\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}+\boldsymbol{\eta})] - \boldsymbol{y}\|_2^2}_{\text{data consistency}} + \alpha \underbrace{\|\mathbb{E}_{\boldsymbol{\eta}}[\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}+\boldsymbol{\eta})] - \boldsymbol{z}\|_2^2}_{\text{denoiser regularization}} \tag{4.25}$$

$$\hat{\boldsymbol{x}} = \mathbb{E}_{\boldsymbol{\eta}}\left[\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\hat{\boldsymbol{z}}+\boldsymbol{\eta})\right] \tag{4.26}$$

In this optimization, $\boldsymbol{z}$ is no longer a reference image, but is instead initialized appropriately and updated. The search space of $\boldsymbol{z}$ is not constrained, although $\boldsymbol{z}$ must have the same dimension as $\boldsymbol{x}$. For example, in multi-coil MRI, the initialization can be a zero-filled (for missing k-space) reconstruction $\sum_{c=1}^{N_c} \boldsymbol{A}_c^H\boldsymbol{y}$. Furthermore, $\boldsymbol{\eta}$ is random noise drawn from some distribution $P_{\boldsymbol{\eta}}$ (either uniform or Gaussian in our experiments). The first term in the optimization enforces data consistency, while the second term is a regularization penalty. The input $\boldsymbol{z}$ is optimized here, in contrast to both vanilla and reference-guided DIP. Hence, we call this method “self-guided" because

at each iteration (of an algorithm) the network's "reference" is updated, with the regularization also guiding the process. Another intriguing feature of this method that we have observed is that the optimal performance is obtained when the magnitude of $\boldsymbol{\eta}$ is quite large.

The proposed regularization smooths the network output over input perturbations. This strategy has been exploited in approaches such as randomized smoothing and makes the network mapping more stable. The regularizer attempts to match the smoothed output to the unperturbed input, mimicking a denoiser. A somewhat simpler form of the objective would place the expectation outside the norms rather than inside. In this case, the regularization term would push the network to act as a usual denoiser, i.e., ensure $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}+\boldsymbol{\eta}) \approx \boldsymbol{z}$. We place the expectation inside of the norms (with the reconstruction being $\mathbb{E}_{\eta}[\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}+\boldsymbol{\eta})]$). This offers the learned network some more flexibility and yielded slightly better image reconstructions in our studies. For example, with the expectation inside, the regularization loss would be 0 for zero-mean $\boldsymbol{\eta}$ if the network were a denoising autoencoder or even just an autoencoder.

The proposed loss is optimized using the Adam optimizer. In Fig. 4.3, the important role of the regularization component in the optimization process of a U-Net network (for multi-coil MRI with 4x and 8x undersampling mask) is underscored. In the absence of this element, $\boldsymbol{z}$ fails to be updated correctly, resulting in unstable training and inferior performance. This illustrates the efficiency of leveraging smoothing and denoising based regularization.

Next, we conduct an ablation study on the impact of additive noise in the network input. Specifically, for every optimization iteration, we add noise vectors η to the DIP network input with magnitude controlled by σ, i.e., $\eta \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$. Fig. 4.4 presents the results w.r.t. to different values of σ. The results demonstrate a correlation between the input noise magnitude and the reconstruction quality. Specifically, performance improves as the noise intensity increases, reaching an optimal point, after which further increases in noise lead to a gradual decline in performance.

Fig. 4.5 shows how the network's input $\boldsymbol{z}+\boldsymbol{\eta}$ evolves throughout the self-guided DIP optimization. It is observable that the input $\boldsymbol{z}$ progressively acquires more feature information, which facilitates the network's learning process, but the input continues to change because of the added noise $\boldsymbol{\eta}$. In

each iteration, the network and input are updated as in Fig. 4.6.

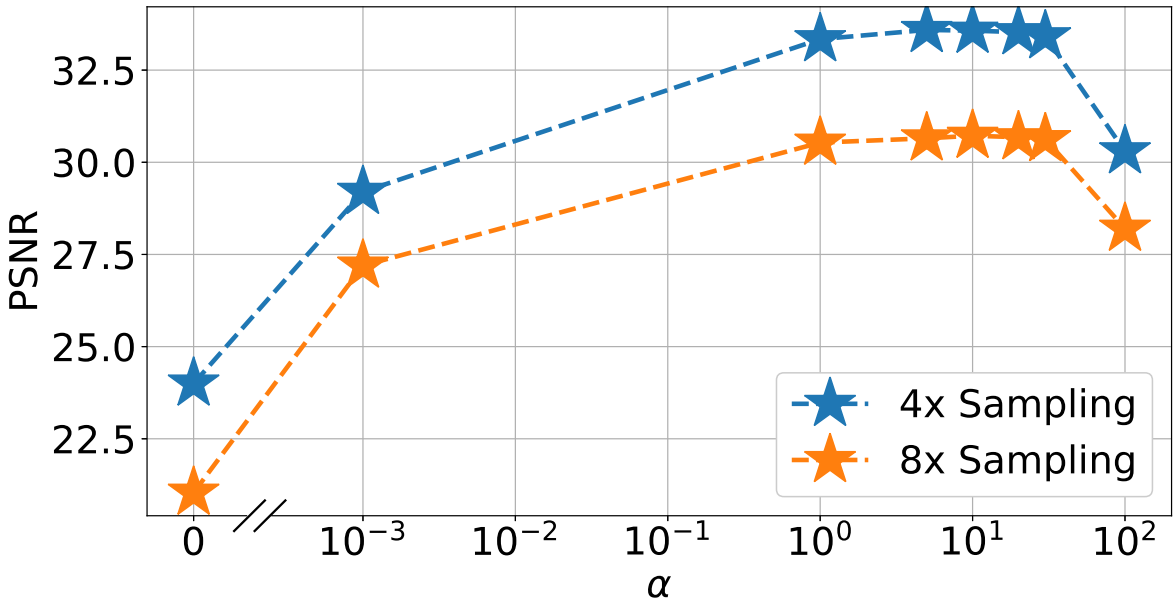


Figure 4.3 Self-guided deep image prior: effect of regularization.

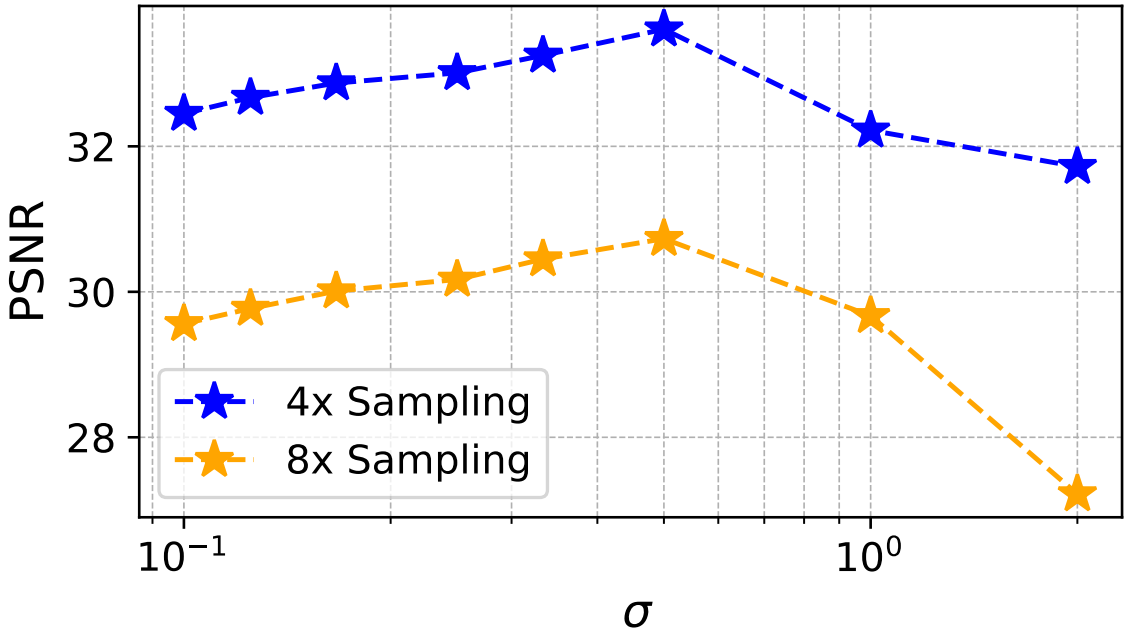


Figure 4.4 Self-guided deep image prior: effect of added noise in the network input.

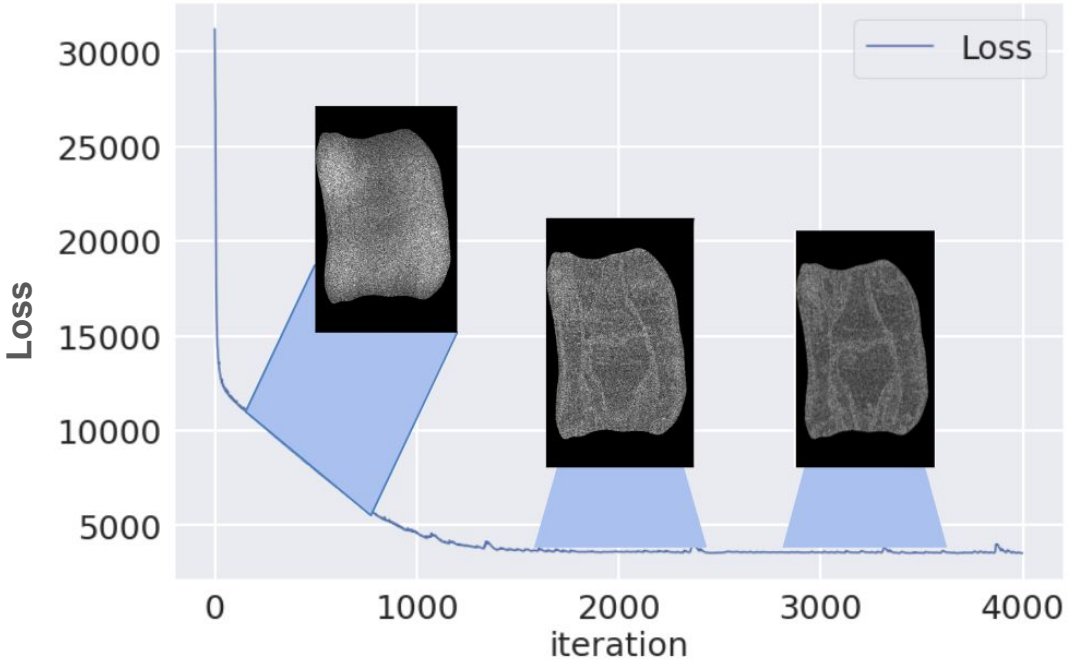


Figure 4.5 Evolution of the network input in self-guided DIP during training for MRI reconstruction at 4x undersampling. As the loss from (4.25) diminishes, the self-guided input supplies additional data, enabling the neural network to enhance its reconstruction capabilities.

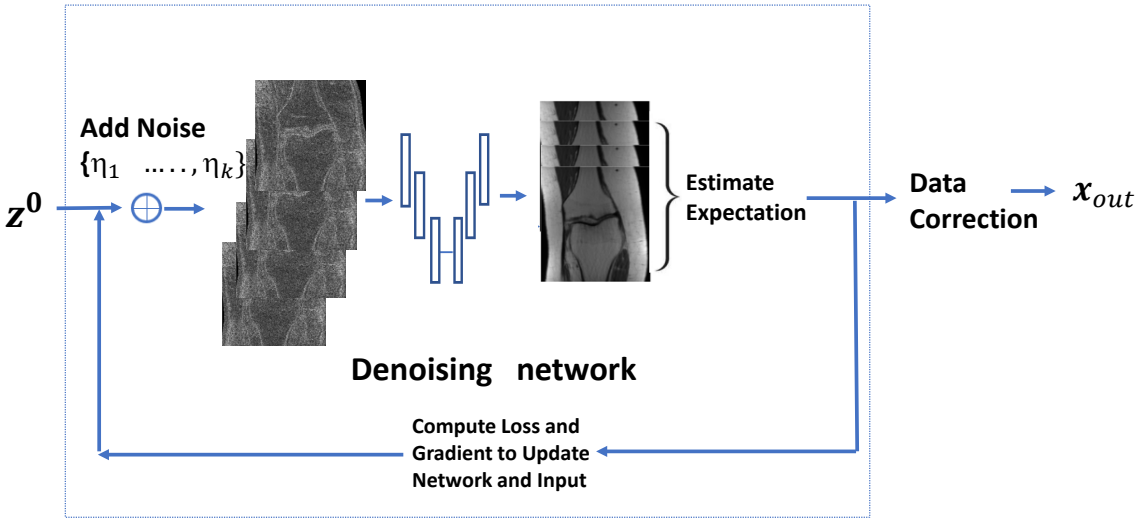


Figure 4.6 Flow chart of the proposed self-guided DIP algorithm.

4.2.3 Post-processing Data Correction

In some applications, it may be desirable to ensure that the reconstructed image is completely consistent with the acquired measurements. This could be the case in compressed sensing

problems, when signal-to-noise ratios are good. For example, consider $\boldsymbol{y} = \mathbf{M}\boldsymbol{\Psi}\boldsymbol{x}$, where $\mathbf{M} \in \mathbb{R}^{p\times q}$ is a subsampling matrix and $\boldsymbol{\Psi} \in \mathbb{C}^{q\times q}$ is a full measurement matrix. Then the matrix $\mathbf{M}' := \mathbf{M}^T\mathbf{M} \in \mathbb{R}^{q\times q}$, subsamples the same measurements, but has zero rows for measurements that are not sampled. Define $\overline{\mathbf{M}} = \boldsymbol{I} - \mathbf{M}'$. Then, for any reconstruction $\hat{\boldsymbol{x}}$, we can construct new, "fully sampled" measurements $\boldsymbol{y}_{\text{new}} \in \mathbb{C}^q$ as $\boldsymbol{y}_{\text{new}} = \mathbf{M}^T\boldsymbol{y} + \overline{\mathbf{M}}\boldsymbol{\Psi}\hat{\boldsymbol{x}}$. Then with these measurements, we can obtain a corrected (data consistent) reconstruction by solving $\hat{\boldsymbol{x}}_{\text{corrected}} = \arg\min_{\boldsymbol{x}} ||\boldsymbol{\Psi}\boldsymbol{x} - \boldsymbol{y}_{\text{new}}||_2^2$.

For multi-coil MRI, assuming appropriately normalized coil sensitivity maps ($\sum_{c=1}^{N_c} \mathbf{S}_c^H\mathbf{S}_c = \boldsymbol{I}$) yields

$$\boldsymbol{y}_{c_{\text{new}}} = \mathbf{M}^T\boldsymbol{y}_c + \overline{\mathbf{M}}\boldsymbol{\mathcal{F}}\mathbf{S}_c\hat{\boldsymbol{x}}, \quad \hat{\boldsymbol{x}}_{\text{corrected}} = \sum_{c=1}^{N_c} \mathbf{S}_c^H\boldsymbol{\mathcal{F}}^H\boldsymbol{y}_{c_{\text{new}}}.$$

4.3 Experiments and Results

We tested the proposed method and alternatives for MRI reconstruction from undersampled measurements and image inpainting.

Dataset. We tested methods for MRI reconstruction using the multi-coil fastMRI knee and brain datasets (et al, 2019, 2020) and the Stanford 2D FSE (Cheng, 2019) dataset. The coil sensitivity maps for all cases were obtained using the BART toolbox (Uecker, 2018). The sensitivity maps were estimated from under-sampled center of k-space data. We also tested our method on image inpainting using the CBSD68 dataset (Roth and Black, 2005) which is show on the supplement material.

Training setup. In our experiments, we compare to related reconstruction methods, which include vanilla DIP, RAKI(Ak**c**cakaya et al., 2019) which is a nonlinear deep learning-based auto-regressive auto-calibrated reconstruction method, reference-guided DIP, DIP with total variation (TV) regularization (Liu et al., 2019a), self-guided DIP, compressed sensing with wavelet regularization, ZS-SSL (zero-shot self-supervised learning) , TRPA (Truncated Residual Based Plug-and-Play ADMM) and a neural network trained in an end-to-end supervised manner (on a set of 3000 images). For compressed sensing MRI, we used the SigPy package, and the regularization parameter was tuned and set as $\lambda = 10^{-6}$. During training, network weights were initialized randomly (normally

distributed). For all of the deep network methods, the network architecture used was a deep U-Net ($\sim 3 \times 10^8$ parameters). The network parameters were optimized using Adam with a learning rate of 3×10^{-4}. For TV-regularized DIP, the parameters used are the same as those in the original paper (Liu et al., 2019a), which worked well. For the ZS-SSL (Yaman et al., 2022), we used the settings in the original paper, with 300 epochs, 10 unrolling blocks, 10 CG iterations and learning rate of 5×10^{-4}. Finally, we compared to the plug-and-play method TRPA (Hou et al., 2022) using the default settings and training the network using 3000 images.

For the self-guided method, we observed that the noise $\boldsymbol{\eta}$ can be drawn from different distributions such as the normal or uniform distribution with essentially identical performance. For our experiments, we drew $\boldsymbol{\eta}$ from $U(0, m)$, where m is $\frac{1}{2}$ of the maximum value of the magnitude of any real or imaginary component of $\boldsymbol{z}$. In this case, $\boldsymbol{z}$ is also optimized using Adam with a learning rate of 1×10^{-1}. At each iteration, we estimated the expectation inside the loss function using 4 realizations of $\boldsymbol{\eta}$. For all unsupervised methods besides compressed sensing, the data correction outlined in Section 4.2.3 was applied. Among supervised methods, we tested with the U-Net and the unrolled MoDL network (Aggarwal et al., 2019a), for which no post-processing was undertaken, as it did not yield significant improvements.

Evaluation. We tested each of the MRI reconstruction methods at 4x acceleration (25.0% sampling) and 8x acceleration (12.5% sampling). Variable density 1-D random Cartesian (phase-encode) undersampling was performed in most cases, unless uniform sampling is specified. We quantified the reconstruction quality of the different methods using the peak signal-to-noise ratio (PSNR) in decibels (dB). We also computed the frequency band metric using equation (4.23) to study the spectral bias and overfitting in each method.

4.3.1 Reconstruction Results for fastMRI Dataset

Table 4.1 provides a quantitative comparison of the average PSNR values for knee (test) data with 4x and 8x sampling acceleration. The proposed self-guided DIP outperforms vanilla DIP, reference-guided DIP, compressed sensing reconstruction, and a corresponding supervised model that was trained on a paired dataset. We also compare to the zero-shot self-supervised learning

method ZS-SSL (Yaman et al., 2022) and plug-and-play based method TRPA (Hou et al., 2022), and find that self-guided DIP yields better performance. A similar comparison for the fastMRI brain dataset can be found in Table 4.2. The benefits of self-guided DIP are also evident in the visual comparisons in Figs. 4.9, 4.10, and 4.11, which show qualitative comparisons for 8x and 4x accelerated knee images, and a 4x accelerated brain image.

Ax	Vanilla DIP	RAKI	TV DIP	Ref-Guided	CS Recon	**Ours**	ZS SSL	Supervised U-Net	TRPA
4x	30.22	30.47	30.52	33.18	29.32	**33.61**	33.01	33.17	33.21
8x	28.77	29.03	28.98	30.24	27.82	**30.73**	30.34	30.28	30.31

Table 4.1 Average reconstruction PSNR values (in dB) for 25 images from the fastMRI knee dataset at 4x and 8x undersampling or acceleration (Ax) including the ZS-SSL method.

Ax	Vanilla DIP	RAKI	TV DIP	Ref-Guided	CS Recon	**Ours**	ZS SSL	Supervised U-Net	TRPA
4x	30.72	30.99	31.04	33.56	29.84	**34.12**	33.44	33.74	33.64
8x	29.03	29.27	29.25	30.54	28.12	**31.04**	30.34	30.57	30.45

Table 4.2 Average reconstruction PSNR values (in dB) for 25 images from the fastMRI brain dataset at 4x and 8x undersampling or acceleration (Ax).

We also evaluate the performance of the reconstruction methods using uniform sampling masks. A quantitative comparison in this setting for fastMRI knee data is given in Table 4.3. The reconstruction results with the uniform undersampling mask indicate that our method significantly outperforms the reference-guided DIP and compared self-supervised and supervised approaches.

Ax	Vanilla DIP	RAKI	TV DIP	Ref-Guided	CS Recon	**Ours**	ZS SSL	Supervised U-Net	TRPA
4x	30.34	30.55	30.67	33.24	29.45	**34.45**	33.67	33.74	33.45
8x	29.25	29.56	29.75	30.84	28.32	**31.88**	30.74	30.87	30.78

Table 4.3 Average reconstruction PSNR values (in dB) for 25 images from the fastMRI knee dataset using 4x and 8x uniform undersampling masks.

Ax	Vanilla DIP	RAKI	TV DIP	Ref-Guided	CS Recon	**Ours**	ZS SSL	Supervised U-Net	TRPA
4x	1.12	1.45	1.15	2.02	0.45	2.05	15.22	0.24	0.56

Table 4.4 Average run-time (minutes) for 30 images from the fastMRI knee dataset at 4x undersampling.

Furthermore, we observed that our method requires slightly more computation time than both the vanilla DIP and the supervised approach in Table 4.4. We note that the run-time for Supervised

U-Net and TRPA exclude training time, i.e., it is inference time only. However, our method demonstrates significantly better performance than vanilla DIP and operates without the need for any training data (dataless) that supervised methods require.

In this subsection, we conducted experiments to understand the reconstruction of different frequencies across the three DIP-based methods. To do this, we used the same frequency band metric introduced previously. We computed this metric over 25 images for 4x k-space undersampling, and the average metric is shown in Fig. 4.7. We observe that the self-guided method shows reduced spectral bias (high frequencies are reconstructed sooner and more accurately), and shows less overfitting in both frequency bands considered, especially compared to vanilla DIP.

To further compare the presence of overfitting in the vanilla, reference-guided, and self-guided methods, Fig. 4.8 shows the average of the reconstruction PSNR for 25 images throughout training. The self-guided DIP shows essentially no overfitting, compared to the vanilla DIP and the reference-guided DIP. The PSNR increases a bit more gradually for self-guided DIP due to its reference/input optimization. However, it quickly outperforms the other compared DIP methods. Our hypothesis is that as the input undergoes continuous optimization, it accrues more high-frequency details (see Fig. 4.5). This enrichment facilitates the network's ability to better assimilate high-frequency details in the output without overfitting.

4.3.2 Reconstruction Results for the Stanford FSE Dataset

Here, we evaluate the reconstruction performance on the Stanford multi-coil FSE dataset(Cheng, 2019). The FSE dataset is a relatively more challenging dataset because it has more diversity in terms of anatomical structures when compared to fastMRI. However, the number of samples is relatively smaller than fastMRI. As illustrated in Table 4.5, the self-guided DIP method outperforms other methods including the well-known unrolling-based MoDL (Aggarwal et al., 2019a) reconstructor. We note that the U-Net in MoDL was trained with supervision on a set of 2000 scans, comprising most of the dataset. We used 6 iterations/unrollings within MoDL. For the End-to-End VarNet, we used a sigmoid with a slope of 10 and 5 cascades, which is the default setting in the paper. We note that MoDL and End-to-End VarNet were each trained separately for the 4x and the 8x

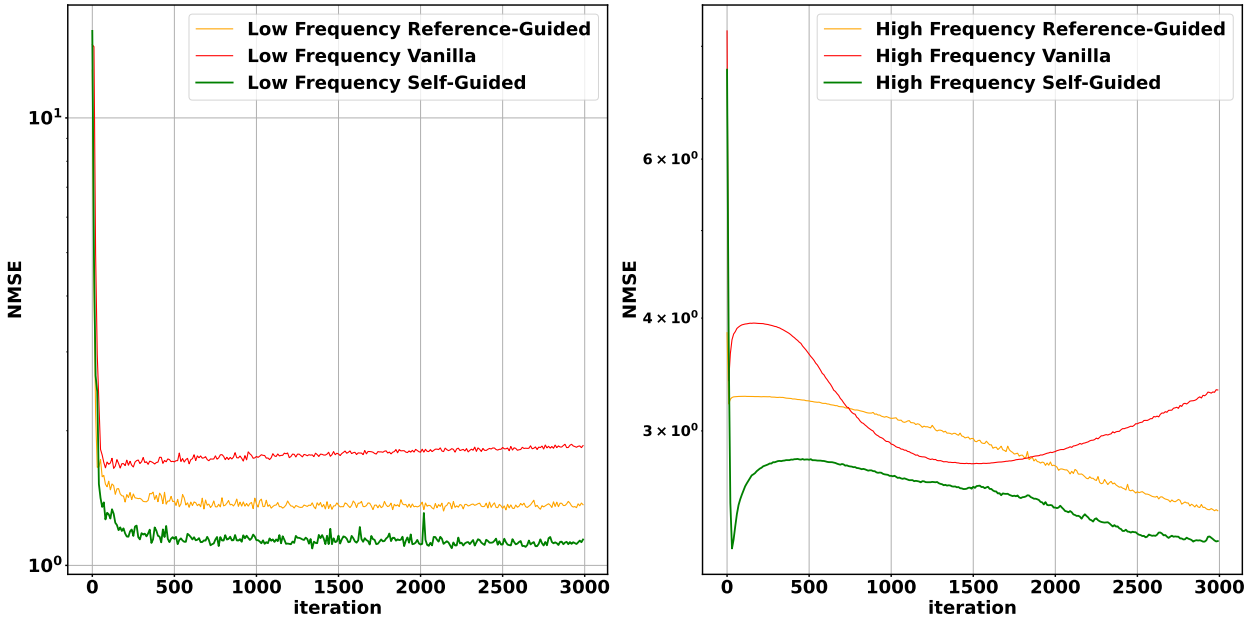


Figure 4.7 Error in the low and high frequencies of the reconstructions, with different methods plotted over iterations at 4x undersampling.

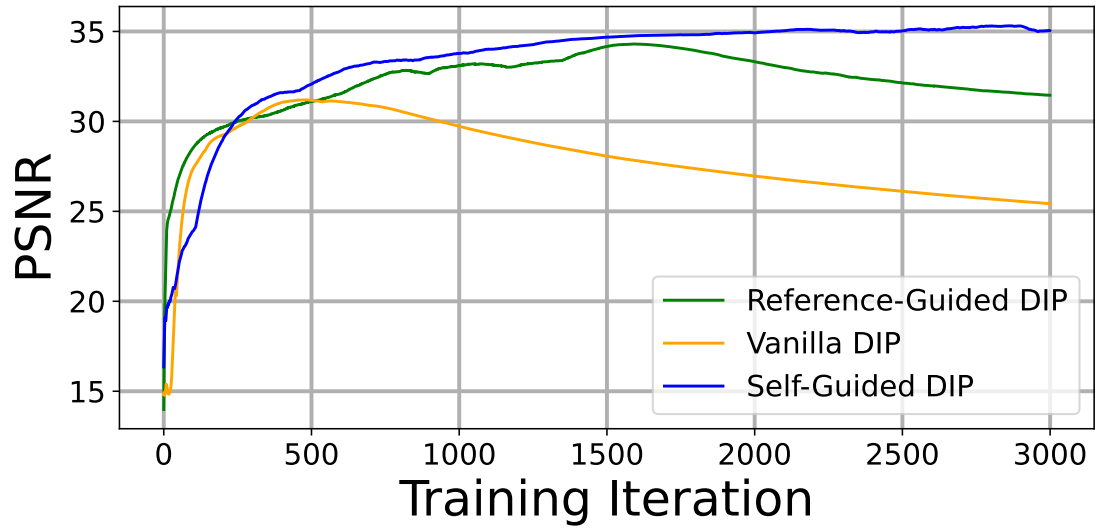


Figure 4.8 PSNR plotted over iterations at 4x undersampling.

acceleration factors. A visual comparison is presented in Fig. 4.12. The results demonstrate the ability of self-guided DIP to restore sharper features compared to the MoDL reconstructor, despite using no training data or references.

4.3.3 Generalization of Self-Guided DIP

Because self-guided DIP trains a network that can accepts different kind of inputs(and noise perturbed), we anticipate that a network trained in this manner will exhibit superior generalization

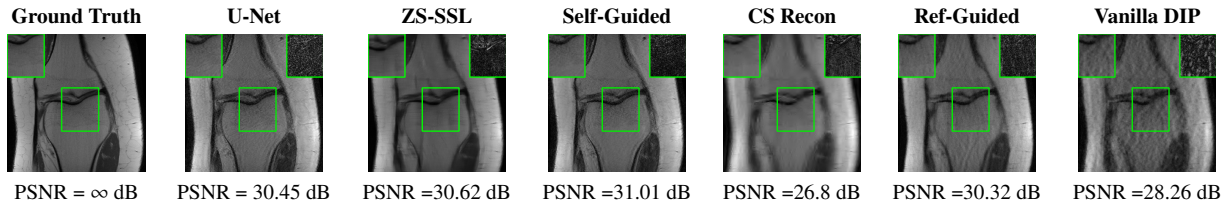


Figure 4.9 Comparison of reconstructions of a knee image using the proposed self-guided DIP method at 8x k-space undersampling or acceleration compared to supervised learning, vanilla DIP, compressed sensing, ZS-SSL and reference-guided DIP reconstruction. A region of interest is shown with the green box and its error (magnitude) is shown in the panel on the top right.

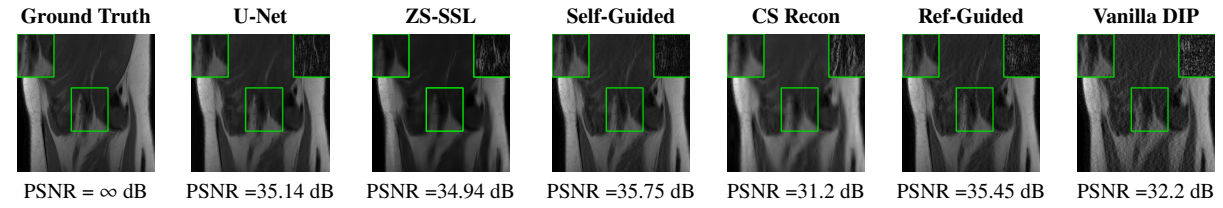


Figure 4.10 Same comparisons/setup as Fig. 4.9, but for 4x acceleration.

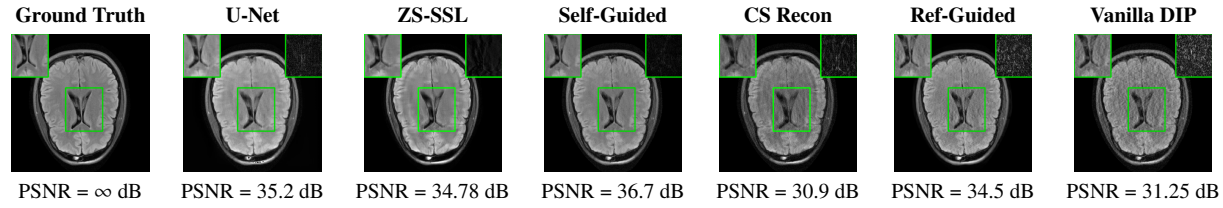


Figure 4.11 Comparison of reconstructions of a brain image using the proposed self-guided method at 4x acceleration versus supervised learning, vanilla DIP, compressed sensing, ZS-SSL and reference-guided reconstruction.

Ax	Vanilla DIP	RAKI	Reference-Guided	CS Recon	Self-Guided DIP	Supervised VarNet	Supervised MoDL
4x	29.9	30.11	32.57	29.75	**33.15**	32.78	32.89
8x	27.45	27.56	29.81	28.1	**30.45**	30.12	29.88

Table 4.5 Average PSNR values (in dB) on the Stanford FSE test set at 4x and 8x undersampling for 15 images for different reconstructors.

to unseen data compared to a network trained using the conventional DIP method. To test this hypothesis, we train the network to reconstruct the nearest neighbor (in terms of ℓ_2 distance) of the target for both vanilla DIP and self-guided DIP. Subsequently, we optimize only the network *input* while keeping its network parameters fixed (i.e., to the network that was trained on the nearest neighbor) using the loss function from (4.24) for DIP and (4.25) for self-guided DIP to reconstruct

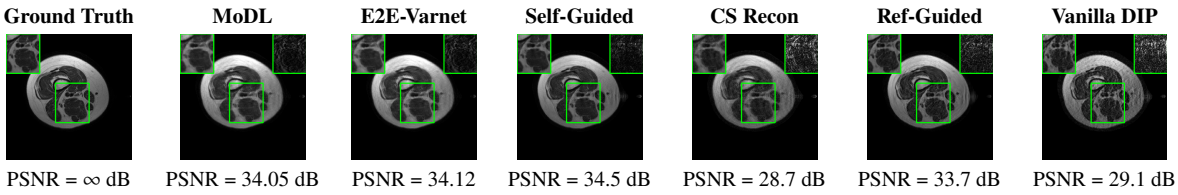


Figure 4.12 Comparison of reconstructions of a FSE dataset image from fourfold undersampled data using the proposed self-guided method versus supervised learning, vanilla DIP, compressed sensing, E2E-Varnet and reference-guided DIP. A region of interest and its error are also shown.

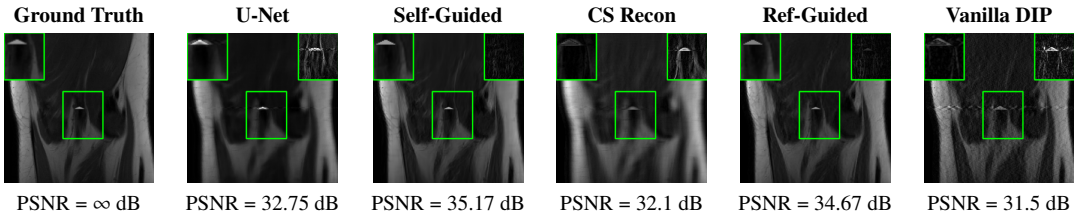


Figure 4.13 Comparison of image reconstructions at 4x k-space undersampling. The methods shown are the proposed self-guided method, supervised learning, vanilla DIP, compressed sensing, and reference-guided reconstruction.

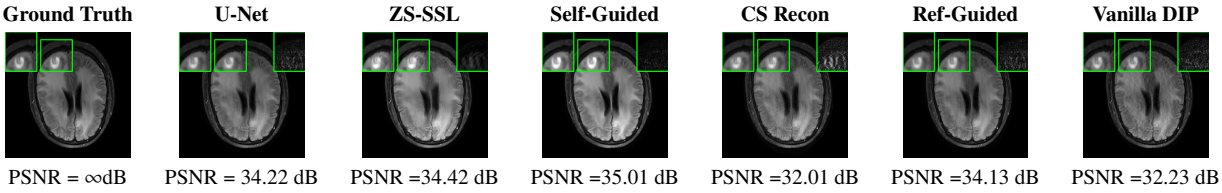


Figure 4.14 Visualization of ground truth and reconstructed images using different methods at 4x k-space undersampling for an annotated image from the fastMRI+ dataset, where the interest area is a nonspecific white matter lesion (in green box). Self-Guided DIP produces sharper image features with reduced artifacts compared to other methods. The top right box shows the error (magnitude) of each reconstruction in the region of interest.

the target. We executed the same experiment for 4x and 8x data undersampling scenarios for 15 fastMRI knee images. The results in Table 4.6 show that self-guided DIP displays much higher benefits in terms of generalization compared to the conventional DIP.

Ax	Vanilla DIP Generalized	Self-Guided DIP Generalized
4x	28.2	31.77
8x	26.65	29.11

Table 4.6 Average reconstruction PSNR values (in dB) for 15 images from the fastMRI knee test set at 4x and 8x undersampling.

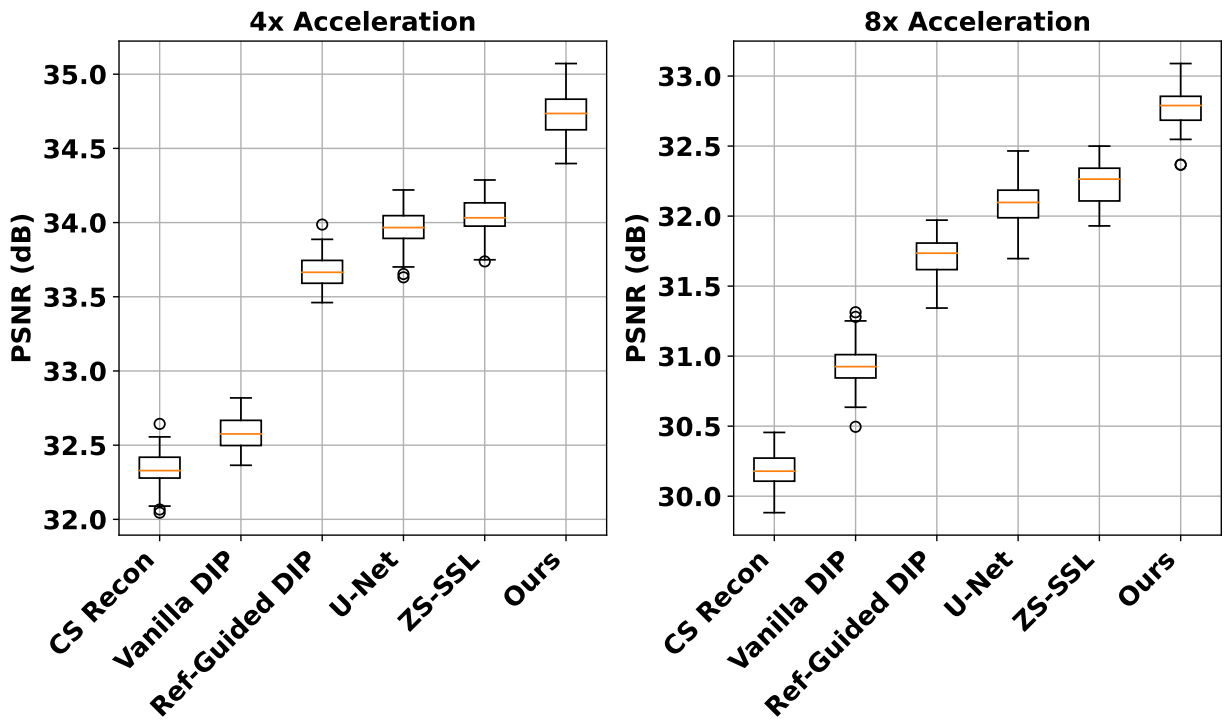


Figure 4.15 Box plots of reconstruction PSNR values (in dB) for different methods for the fastMRI lesion test set at 4x and 8x undersampling. Our (self-guided DIP) results are compared to vanilla DIP, reference-guided DIP, ZS-SSL, and a supervised U-Net trained on 3000 non-lesion scans, and CS reconstruction.

Furthermore, to evaluate the ability of self-guided DIP to accurately reconstruct fine image details, especially in common scenarios like pathology detection, we incorporated some features into a knee image from the fast MRI dataset. This is similar to recent work (Lahiri et al., 2021). By undersampling at a 4x rate in k-space and using the U-Net, we observe (Fig. 4.13) that the self-guided DIP renders a clearer image with better PSNR compared to supervised learning techniques. The intricacies and boundaries of the added features were more effectively maintained with the self-guided DIP scheme. Distinctively, the image quality offered by self-guided DIP remains similar, irrespective of whether the features are included or not (see Fig. 4.10). In contrast, the quality using the supervised method dipped notably, highlighting the superior stability and adaptability of a robust DIP-based approach.

In Figure 4.14, we provide an additional comparison of these methods for reconstructing an image from the fastMRI+ dataset that contains a real brain lesion. This comparison shows the

superiority of our method in reconstructing the white matter lesion. For the training phase of the supervised U-Net, a non-lesion dataset was employed with 3000 scans with 4x and 8x undersampling (as in section 4.3.1).Also, the ZS-SSL was employed using the same setting as the previous section To provide a quantitative comparison, we tested the methods on 15 scans with lesions. The results, displayed in boxplots in Figure 4.15, show that our method also achieves higher PSNR values on this data compared to other methods, including the supervised U-Net.

4.4 Discussion of Results

We have introduced a novel self-guided image reconstruction method requiring no training data that iteratively optimizes the reconstructor network and its input. This approach is completely unsupervised and instance-adaptive, and demonstrated strong reconstruction performance on the multi-coil fastMRI knee and brain datasets and the Stanford FSE dataset. The approach does not require pre-training and can easily accommodate variations in most MRI reconstruction settings. Additionally, it was found to outperform supervised methods like image-domain U-Net and hybrid-domain MoDL and E2E Varnet, especially on smaller, more diverse datasets. We note that given enough matched training data, these powerful supervised methods should outperform self-guided DIP, although the required number of samples may be very large. Indeed, previous studies (Klug and Heckel, 2023) have demonstrated significantly diminishing returns for datasets larger than a few thousand images for medical image reconstruction. Since acquiring large paired datasets is challenging, particularly in medical imaging, we emphasize the importance of developing effective zero-shot methods. We also showed that the networks learned in self-guided DIP demonstrate better stability and generalizability compared to those learned in vanilla DIP. Finally, we demonstrated the effectiveness of self-guided DIP for image inpainting on the CBSD68 dataset.

While the self-guided DIP algorithm does require more optimization steps than vanilla DIP because the network's input must also be optimized, this additional cost is not detrimental. For example, self-guided DIP with a randomly initialized U-Net took about 2 minutes to run on an NVIDIA GeForce RTX A5000 GPU (with a batch size of 2 and 1500 training iterations), whereas vanilla DIP took about 1 minute.

4.5 Conclusions

In this study, we first presented theoretical results that help explain the training dynamics of unsupervised neural networks for general image reconstruction. We empirically validated our findings using some simple example problems.

We then proposed a novel self-guided deep image prior based MRI reconstruction technique that iteratively optimizes the network input while also training the model to be robust to large random perturbations of its input. This was achieved by introducing a new regularization term that encourages the reconstructor to act as a denoiser.

We empirically demonstrated that this method yields promising results for MRI reconstruction and image inpainting on different datasets. Notably, our approach does not involve any pre-training, and can thus readily handle changes in the measured data. Moreover, this self-guided method showed better performance than the same model trained in a supervised manner on a large dataset (with lengthy training times). This shows that highly adaptive learning approaches may have the potential to outperform traditional data-driven learning approaches in image reconstruction. In the future, we hope to carry out more theoretical analyses to better understand the performance of self-guided DIP for image reconstruction and analyze how the optimization of the network's input improves reconstruction performance. We also plan to study whether similar self-guided schemes could improve the performance of DIP for other imaging modalities and restoration tasks such as deblurring and super-resolution.

CHAPTER 5

AUTOENCODING SEQUENTIAL DEEP IMAGE PRIOR

5.1 Introduction

In the previous chapter, we introduced Self-Guided Deep Image Prior (Self-Guided DIP) to alleviate the overfitting issue commonly encountered in Deep Image Prior (DIP) methods. While Self-Guided DIP significantly enhances reconstruction quality by guiding the optimization with a carefully designed self-supervisory signal, it also introduces additional computational overhead. This longer runtime partly stems from the extra steps required for the self-guidance mechanism to refine the network's predictions.

Our goal, therefore, is to preserve the benefits of Self-Guided DIP—namely, its ability to discourage overfitting and improve reconstruction fidelity—while addressing its slower convergence. Drawing inspiration from the progressive denoising strategy found in recent diffusion-based generative models, we propose a novel approach, Autoencoding Sequential DIP (aSeqDIP), to achieve more efficient image reconstruction. Compared to diffusion models, our method does not require training data and outperforms other DIP-based methods in mitigating noise overfitting while maintaining a similar number of parameter updates as Vanilla DIP. Through extensive experiments, we validate the effectiveness of our method in various image reconstruction tasks, such as MRI and CT reconstruction, as well as in image restoration tasks like image denoising, inpainting, and non-linear deblurring.

5.1.1 Related Work

DIP-based Methods: Deep Image Prior (DIP) was first introduced by (Ulyanov et al., 2018). The authors demonstrated that the architecture of a generator network alone is capable of capturing a significant amount of low-level image statistics even before any learning takes place. Specifically, the DIP image reconstruction is obtained through the minimization of the following objective:

$$\hat{\theta} = \arg\min_{\theta} \ \|\mathbf{A} f_{\theta}(\mathbf{z}) - \mathbf{y}\|_2^2 \,, \quad \hat{\mathbf{x}} = f_{\hat{\theta}}(\mathbf{z}) \,, \tag{5.1}$$

where $\hat{\mathbf{x}}$ is the reconstructed image, and θ corresponds to the parameters of network $f : \mathbb{R}^n \to \mathbb{R}^n$, which is typically implemented using a U-Net architecture (Ronneberger et al., 2015a). The input to the network, $\mathbf{z} \in \mathbb{R}^n$, is randomly chosen and remains fixed throughout the optimization process. While standard DIP was shown to perform well in many tasks, selecting the number of iterations to optimize objective (5.1) poses a challenge as the network would eventually fit the noise present in $\mathbf{y}$ or could fit to undesired images based on the null space of $\mathbf{A}$.

To mitigate the problem of noise overfitting, previous studies considered different approaches such as regularization, early stopping (ES), and network pruning (Ghosh et al., 2024). For regularization-based methods, the work in (Liu et al., 2019b) enhanced the standard DIP by introducing a total variation (TV) regularization term for denoising and deblurring tasks, whereas the study in (Cheng et al., 2019) proposed combining DIP with stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011). The authors in (Wang et al., 2023a) use running variance as the criterion for ES, whereas the authors of (Li et al., 2021) propose combining self-validation and training to apply ES.

The input to the standard DIP (or Vanilla DIP) network is a random noise vector that, in most works, remains fixed during the optimization. Nevertheless, other works, such as those in (Zhao et al., 2020a) and (Tachella et al., 2021), have explored cases where the input contains some structure of the ground truth. The approach employed in reference-guided DIP (Ref-Guided DIP) (Zhao et al., 2020a) follows the same objective as standard DIP in (5.1). However, instead of using a fixed random noise vector as input, it utilizes a reference image closely resembling the one undergoing reconstruction. This method was applied to the task of MRI. This methodology proves particularly effective when datasets comprising structurally similar data points are available. The reference required here makes this method a data-dependent approach.

Inspired by the departure from using a random fixed input, the authors in (Liang et al., 2024a) recently introduced Self-Guided DIP. Unlike Ref-Guided DIP (Zhao et al., 2020a), a prior image that closely resembles the unknown (to be estimated) image is not needed, and the optimization occurs simultaneously with respect to both the input and the parameters of the network. Specifically,

Self-Guided DIP employs the following objective:

$$\hat{\theta}, \hat{\mathbf{z}} = \arg\min_{\theta,\mathbf{z}} \|\mathbf{A}\mathbb{E}_{\boldsymbol{\eta}}[f_\theta(\mathbf{z}+\boldsymbol{\eta})] - \mathbf{y}\|_2^2 + \alpha\|\mathbb{E}_{\boldsymbol{\eta}}[f_\theta(\mathbf{z}+\boldsymbol{\eta})] - \mathbf{z}\|_2^2\,, \tag{5.2}$$

where $\boldsymbol{\eta}$ is random noise, and α is a regularization parameter. The first (resp. second) term is used for data consistency (resp. denoising regularization) and final reconstruction is obtained as $\hat{\mathbf{x}} = \mathbb{E}_{\boldsymbol{\eta}}[f_{\hat{\theta}}(\hat{\mathbf{z}}+\boldsymbol{\eta})]$. aSeqDIP is different from Self-Guided DIP as our method does not require gradient-based updates for the input, making it computationally less expensive. Self-Guided DIP has demonstrated superior performance compared to Vanilla DIP, TV-DIP, and SGLD-DIP, thus serving as a primary baseline for comparison.

DM-based Methods: In recent years, there has been an abundance of DM-based methods proposed to address inverse imaging problems (Chung and Ye, 2022; Chung et al., 2022, 2023c; Li et al., 2024; Song et al., 2024; Daras et al., 2024). A well-known method for natural images is Diffusion Posterior sampling (DPS) (Chung et al., 2023c). DPS incorporates a gradient step into the reverse sampling process of pre-trained DMs, ensuring data consistency and enabling sampling from the conditional distribution. In the context of image reconstruction and restoration tasks, numerous diffusion-based approaches have emerged, as evidenced by works such as (Xie and Li, 2022; Güngör et al., 2023; Peng et al., 2022). Notably, the authors in (Chung and Ye, 2022) and (Chung et al., 2022) introduced a SOTA DM-based approach for addressing the MRI and CT reconstruction inverse problems, respectively. They propose incorporating the predictor-corrector sampling algorithm (Song et al., 2021c) for data consistency, akin to DPS, thereby facilitating the sampling from a conditional distribution.

One clear distinction between aSeqDIP and DM-based methods is that our approach does not necessitate pre-trained models. For our experiments in MRI, CT, and denoising (as well as in-painting and deblurring) tasks, we will utilize Score-MRI (Chung and Ye, 2022), Manifold Constrained Gradient (MCG) (Chung et al., 2022), and DPS (Chung et al., 2023c), respectively, as DM-based baselines.

5.2 Method

In this section, we begin by investigating the impact of the input on DIP. Then, we introduce our method, aSeqDIP. We note that while we consider linear and non-linear inverse problems, in our formulations, we use a linear forward model to simplify notation.

5.2.1 Motivation of aSeqDIP: The Impact of the Network Input in Vanilla DIP

Here, we aim to address the question: *How does employing a noisy version of the ground truth image, which retains some structure of the ground truth, as the fixed input to the Vanilla DIP objective in* (5.1)*, affect performance?* To investigate, we conduct the following experiment.

Consider the MRI task defined as $\mathbf{y} \approx \mathbf{A}\mathbf{x}^*$. Let the input to the standard DIP objective in (5.1) be denoted as $\mathbf{z} = \mathbf{x}^* + \boldsymbol{\delta}$, where $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Here, σ controls the magnitude of the perturbations added to the ground truth image, indicating that a larger σ results in a greater deviation between $\mathbf{z}$ and $\mathbf{x}^*$. We optimize (5.1) for various values of σ, recording the best possible PSNR compared to the ground truth, i.e., prior to the start of the noise overfitting decay.

Figure 5.1 displays the average results for 8 images. Notably, for all images, a closer similarity of the DIP network input to $\mathbf{x}^*$, as indicated by σ, corresponds to higher reconstruction quality, measured by PSNR. Larger variance in the standard Gaussian distribution corresponds to larger additive perturbations even for the case of $\mathbf{x}^* = \mathbf{0}$ (the red curve). We conjecture that this still leads to larger distances from the ground truth and hence worse performance.

Based on this discussion, a notable insight emerges:

The proximity of the DIP network input to the ground truth correlates with the quality of the reconstruction. This promotes the question: *Can we develop an input-adaptive DIP method that mitigates noise overfitting?* We proceed to address this question by proposing our method, which we refer to as Autoencoding Sequential DIP (aSeqDIP). In Appendix B.1, we provide a case study and theory on the impact of the DIP network input through the lens of the Neural Tangent Kernel in residual networks. The onset of severe noise overfitting therein is delayed for better inputs (Appendix B.1.2).

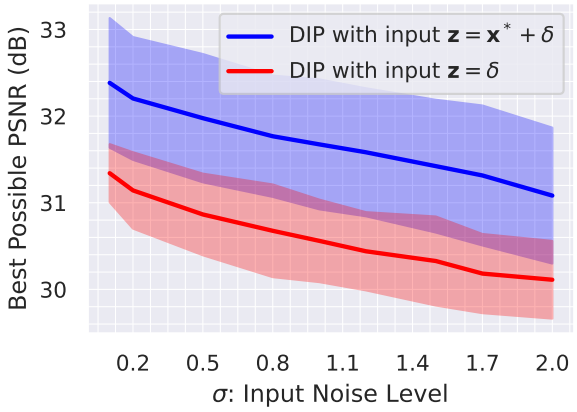


Figure 5.1 Average best possible PSNR values (in dB) obtained from standard DIP in (5.1) for 8 MRI (with 4x acceleration factor) scans (y-axis), where the network input $\mathbf{z}$ is either a perturbed version of the ground truth or pure noise. The noise is a zero-mean additive Gaussian noise with strength determined by σ (x-axis).

5.2.2 The Proposed aSeqDIP Algorithm

Consider that we have a U-Net architecture defined by $f : \mathbb{R}^n \to \mathbb{R}^n$ whose weights are given by ϕ_k, where $k \in [K]$, and $[K] := \{1, \ldots, K\}$. Each set of parameters in f_{ϕ_k} takes an input $\mathbf{z}_k$ and outputs $f_{\phi_k}(\mathbf{z}_k)$. Based on the insight from the previous subsection, we initially set $\mathbf{z}_0$ to $\mathbf{y}$ (resp. $\mathbf{A}^H\mathbf{y}$) for denoising, in-painting, and deblurring (resp. MRI and CT). The initialization of ϕ_1 follows the same initialization as any other DIP-based method. The parameters in f_{ϕ_k}, and the input, $\mathbf{z}_k$, are then updated sequentially through

$$\phi_k \leftarrow \arg\min_{\phi_k} \|\mathbf{A} f_{\phi_k}(\mathbf{z}_{k-1}) - \mathbf{y}\|_2^2 + \lambda \|f_{\phi_k}(\mathbf{z}_{k-1}) - \mathbf{z}_{k-1}\|_2^2 \,, \tag{5.3}$$

$$\mathbf{z}_k \leftarrow f_{\phi_k}(\mathbf{z}_{k-1}) \,, \tag{5.4}$$

where $\lambda \in \mathbb{R}_+$ is a regularization parameter, and the initialization of ϕ_k is the optimized ϕ_{k-1} in (5.3). The final reconstruction is given as:

$$\hat{\mathbf{x}} = \mathbf{z}_K = f_{\phi_K}(\mathbf{z}_{K-1}) \,. \tag{5.5}$$

The proposed procedure outlined in (5.3) and (5.4) consists of two key components. First, the optimization of each set of weights in f_{ϕ_k} using an objective that consists of the data consistency

term and the second autoencoding term that aims to alleviate noise overfitting. Second, the update of the input, $\mathbf{z}_k$, after optimizing each set of weights f_{ϕ_k}, so that our method is *input-adaptive*.

Algorithm 5.1 presents the procedure of our proposed approach. As inputs, the algorithm takes $\mathbf{y}$, $\mathbf{A}$, K, N, λ, and the learning rate β. Apart from the measurements and the forward operator, the remaining parameters are considered hyper-parameters, typical in most DIP-based methods. The parameters in f_{ϕ_k} are set to ϕ_{k-1} (step 2) and subsequently optimized for N iterations using a gradient-based optimizer, such as gradient descent (as depicted in Algorithm 5.1) or Adam (Kingma and Ba, 2014). A block diagram of our proposed aSeqDIP method is presented in Figure 5.2.

In the following remarks, we provide insights into our proposed aSeqDIP method.

Remark 1 (Differences from Vanilla DIP (Ulyanov et al., 2018))**.** *Assume that the iterates of* (5.3) *and* (5.4) *converge, i.e., as* $k \to \infty$, $\mathbf{z}_k \to \mathbf{z}^*$ *and* $\phi_k \to \phi^*$. *Then, according to* (5.4)*, for a continuous mapping* f*, we have* $\mathbf{z}^* = f_{\phi^*}(\mathbf{z}^*)$. *Substituting this into* (5.3) *in the limit, we get*

$$\phi^* = \{\arg\min_{\phi} \|\mathbf{A}f_\phi(\mathbf{z}^*) - \mathbf{y}\|_2^2 \ : \ f_\phi(\mathbf{z}^*) = \mathbf{z}^*\}, \tag{5.6}$$

which corresponds to the minimizer of

$$\min_{\phi} \|\mathbf{A}f_\phi(\mathbf{z}) - \mathbf{y}\|_2^2 \quad s.t. \quad \mathbf{z} = f_\phi(\mathbf{z}). \tag{5.7}$$

The limit points of aSeqDIP correspond to the solution of a constrained version of the Vanilla DIP objective in (5.1)*. The constraint enforces additional prior that could alleviate overfitting. While its not straightforward to use a gradient-based algorithm for* (5.7) *given the hard constraint, the aSeqDIP scheme's limit points nevertheless minimize* (5.7)*. Furthermore, aSeqDIP automatically estimates the network input by a sequential feed forward process without needing expensive updates. The main point is to show that aSeqDIP is solving the optimization problem in* (5.7)*, which is different than Vanilla DIP in* (5.1)*.*

Remark 2 (Differences from Self-Guided DIP (Liang et al., 2024a))**.** *While both aSeqDIP and Self-Guided DIP (Liang et al., 2024a) update the input and network parameters simultaneously, there exist fundamental differences. Firstly, Self-Guided DIP solves the optimization problem in* (5.2)

Algorithm 5.1 **A**utoencoding **Seq**uential **D**eep **I**mage **P**rior (**aSeqDIP**).

Input: Measurements $\mathbf{y}$, forward operator $\mathbf{A}$, number of input updates K, number of gradient updates N per input update, regularization parameter λ, and learning rate β.

Output: Reconstructed image $\hat{\mathbf{x}}$.

Initialization: $\mathbf{z}_0 = \mathbf{A}^H\mathbf{y}$; $\phi_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

1: **For each** $k \in [K]$

2: **Initialize** $\phi_k^{(0)} \leftarrow \phi_{k-1}^{(N)}$ for $k \in \{2, \ldots, K\}$, and $\phi_k^{(0)} \leftarrow \phi^{(0)}$ for $k = 1$.

3: **For each** $i \in [N]$. (Network parameters update)

4: $$\phi_k^{(i)} = \phi_k^{(i-1)} - \beta \nabla_{\phi_k} \left[\|\mathbf{A} f_{\phi_k}(\mathbf{z}_{k-1}) - \mathbf{y}\|_2^2 + \lambda \|f_{\phi_k}(\mathbf{z}_{k-1}) - \mathbf{z}_{k-1}\|_2^2 \right] \Big|_{\phi_k = \phi_k^{(i-1)}}.$$

5: **Obtain** $\mathbf{z}_k := f_{\phi_k^{(N)}}(\mathbf{z}_{k-1})$. (Network input update)

6: **Reconstructed image:** $\hat{\mathbf{x}} = \mathbf{z}_K = f_{\phi_K^{(N)}}(\mathbf{z}_{K-1})$

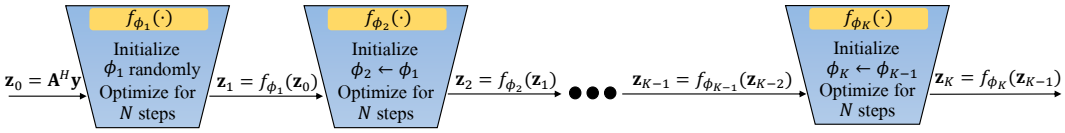


Figure 5.2 Illustrative block diagram of the proposed aSeqDIP procedure. Each trapezoid corresponds to the updates of f_{ϕ_k} that takes $\mathbf{z}_{k-1}$ as input and is initialized with the optimized parameters ϕ_{k-1} for $k \in \{2, \ldots, K\}$ or randomly for $k = 1$. The optimization for each set of weights takes place based on (5.3) and is run for N steps. The final reconstruction is $f_{\phi_K}(\mathbf{z}_{K-1})$.

which does not strictly enforce the auto-encoder constraint $\mathbf{z} = f_\phi(\mathbf{z})$ *as in* (5.7)*. Secondly, aSeqDIP only requires a network forward pass to update* $\mathbf{z}$*, resulting in significantly fewer computations as will further be demonstrated in our experimental results. Thirdly, the second term in the aSeqDIP objective does not require computing the expectation, as it is an auto-encoder rather than a denoiser that results in higher resistance to noise overfitting. Lastly, our method does not require initializing* $\mathbf{z}$ *randomly and generating random vectors (*$\boldsymbol{\eta}$ *in* (5.2)*). The selection of* $\boldsymbol{\eta}$ *introduces an additional hyper-parameter that we avoid, focusing solely on selecting* NK *(total number of iterations) and* λ *(regularization strength), which are necessary in most DIP-based methods.*

Remark 3 (Computational Requirements)**.** *The computational requirements of aSeqDIP are determined by two factors: (i) the* NK *gradient-based parameter updates, and (ii) the number of function evaluations necessary for updating* $\mathbf{z}$*, which is* K*. In our experiments, we have found that setting* $N = 2$ *and* $K = 2000$ *is generally sufficient. This configuration makes aSeqDIP nearly as efficient as Vanilla DIP.*

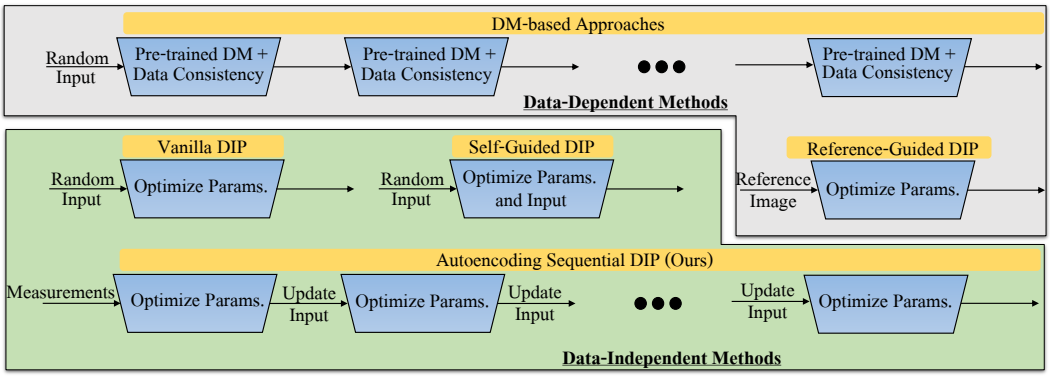


Figure 5.3 **An overview of differences between aSeqDIP and prior arts** in terms of data dependency, network architecture(s), and procedural requisites. '**Data-Dependency**' here indicates whether a method depend on a prior reference image or pre-trained models.

Remark 4 (Relationship to DMs)**.** *aSeqDIP bears resemblance to the reverse process in DMs due to their shared gradual denoising steps. However, despite these similarities, several distinctions emerge. Firstly, unlike the DM network, aSeqDIP does not require encoding a scalar representing time t. Secondly, and perhaps most significantly, aSeqDIP operates without requiring any training data or pre-trained networks. Thirdly, aSeqDIP operates in a truly sequential manner in terms of time, whereas in DMs, whether it's training (e.g., denoising score matching (Vincent, 2011)) or sampling, the prevalent technique involves sampling from time* $t \sim \mathcal{U}[0,1]$ *(uniform distribution), which allows for non-sequential time points.*

Figure 5.3 illustrates how different approaches compare to aSeqDIP.

5.2.2.1 Mitigating Noise Overfitting in aSeqDIP

In DIP-based approaches, noise overfitting occurs as the network attempts to fit its output to the noisy or subsampled measurements, $\mathbf{y}$, as k increases during training. However, the specific value of k at which this PSNR decay begins is uncertain and varies across tasks and even among images within the same task and distribution. In aSeqDIP, when the output of network f_{ϕ_k} improves compared to that of $f_{\phi_{k-1}}$, the autoencoder term enforces similarity between the input and output of the network, thus delaying the onset of noise overfitting decay. This occurs because we are not only enforcing the network output to be measurement-consistent, but also enforcing that the output and input become similar. Consequently, as k increases, noise fitting is delayed, and utilizing the

autoencoder provides regularization against noise overfitting. In Section 5.3, we will demonstrate how the proposed autoencoding term effectively regulates noise overfitting.

One might expect that incorporating the autoencoder could negatively impact reconstruction quality. However, empirical observations reveal that not only is noise overfitting delayed with the autoencoder term, but also image reconstruction quality is enhanced. To further support this statement, in Appendix B.1.3, we investigate whether a trained autoencoder on clean images can act as a reconstructor at testing time by optimizing the input.

5.3 Experimental Results

5.3.1 Settings, Datasets, and Baselines

In our experiments, we consider five tasks: MRI reconstruction from undersampled measurements, sparse-view CT image reconstruction, denoising, non-linear deblurring and in-painting. For MRI, we use the fastMRI dataset. The forward model is $\mathbf{y} \approx \mathbf{A}\mathbf{x}^*$. The multi-coil data is obtained using 15 coils and is cropped to a resolution of 320×320 pixels. To simulate undersampling of the MRI k-space, we use a Cartesian mask with 4x and 8x accelerations. Sensitivity maps for the coils are obtained using the BART toolbox (Tamir et al., 2016). For CT, we use the AAPM dataset. For parallel beam CT, the input image with 512×512 pixels is transformed into its sinogram representation using a Radon transform (the operator $\mathbf{A}$). The forward model assuming a monoenergetic source and no scatter, noise is $y_i = I_0 e^{-[\mathbf{A}\mathbf{x}^*]_i}$, with I_0 denoting the number of incident photons per ray (assumed to be 1 for simplicity) and i indexing the ith measurement or detector pixel. We use the post-log measurements for reconstruction. We use a full set of 180 projection angles and simulate two different sparse view acquisition scenarios (with equispaced angles). Specifically, we created cases with 18 and 30 angles/views. The image resolution is kept at a fixed size.

For the tasks of denoising, in-painting, and non-linear deblurring, we use the CBSD68 dataset. For each task, we use 20 measurements/corrupted images. To evaluate the reconstruction quality, we use the Peak Signal to Noise Ratio (PSNR), and the Structural SIMilarity (SSIM) index (Wang et al., 2004). For experimental settings and baselines, see Table 5.1 and its caption. Note that we consider data-dependent and data-independent baselines as shown in the third and fourth columns

Task	Setting	Data-independent baselines	Data-dependent baselines
MRI	Ax $\in \{4x, 8x\}$	Vanilla DIP (Ulyanov et al., 2018), ES-DIP (Wang et al., 2023a), TV-DIP (Liu et al., 2019b), Self-Guided DIP (Liang et al., 2024a)	Ref-Guided DIP (Zhao et al., 2020a) Score-MRI (Chung and Ye, 2022)
CT	views $\in \{18, 30\}$	Vanilla DIP (Ulyanov et al., 2018), Self-Guided DIP (Liang et al., 2024a), Filter Back Projection (FBP) (Zeng, 2020)	Ref-Guided DIP (Zhao et al., 2020a) MCG (Chung et al., 2022)
Denoising	$\sigma_{\text{d}} \in \{15, 30\}$	Vanilla DIP (Ulyanov et al., 2018), ES-DIP (Wang et al., 2023a), Self-Guided DIP (Liang et al., 2024a), TV-DIP (Liu et al., 2019b), Rethinking-DIP (Jo et al., 2021), SGLD-DIP (Cheng et al., 2019)	DPS (Chung et al., 2023c)
In-painting	HIAR $\in \{0.1, 0.25\}$	Vanilla DIP (Ulyanov et al., 2018), ES-DIP (Wang et al., 2023a), Self-Guided DIP (Liang et al., 2024a), SGLD-DIP (Cheng et al., 2019), TV-DIP (Liu et al., 2019b)	DPS (Chung et al., 2023c)
Deblurring	BKSE (Tran et al., 2021)	Self-Guided DIP (Liang et al., 2024a) SGLD-DIP (Cheng et al., 2019)	DPS (Chung et al., 2023c)

Table 5.1 Tasks, settings, and baselines considered in our experiments. For MRI, we consider two Acceleration (Ax) factors, 4x and 8x, that determine the subsampling of the measurements. For 2D CT (parallel beam geometry), we use two sparse view settings: 18 and 30 views. For denoising, we perturb the ground truth images using two noise levels determined by σ_{d}. In in-painting, we use two hole-to-image area ratios (HIAR), 0.1 and 0.25. For non-linear deblurring, we use the Blurring Kernel Space Exploring (BKSE) setting (Tran et al., 2021), described in Equations (56) to (59) of (Chung et al., 2023c). Each baseline that utilizes pre-trained models or a reference image is considered data-dependent. Further details are provided in Appendix B.2.

of Table 5.1. All the experiments are conducted on a single RTX5000 GPU machine. Further implementation details are provided in Appendix B.2.

For the proposed aSeqDIP method in Algorithm 5.1, we use the Adam optimizer with learning rate of $\beta = 0.0001$. Furthermore, the regularization parameter is set to $\lambda = 1$ following the ablation study in Appendix B.2.4. We select $N = 2$ and $K = 2000$ following the ablation study in Appendix B.2.5.

5.3.2 Impact of the Autoencoding term on Noise Overfitting

In this subsection, we showcase the impact of the proposed autoencoding regularization in aSeqDIP on noise or null space (nuisance) overfitting.

We conducted experiments using 20 MRI scans and 20 CT scans, considering two cases of aSeqDIP as outlined in Algorithm 5.1. The first case sets $\lambda = 1$, consistent with the remainder of the paper, while the second case sets $\lambda = 0$, effectively disabling the autoencoding regularization term in (5.3). Additionally, for comparison, we report results for Vanilla DIP and Self-Guided DIP. The average PSNR results for these cases are depicted in Figure 5.4.

As observed, when the autoencoder term is disabled in aSeqDIP (black dashed lines), noise overfitting in MRI, akin to Self-Guided DIP, begins after nearly 1600 iterations. For CT, we note that aSeqDIP without regularization starts noise overfitting at around iteration 3800, whereas

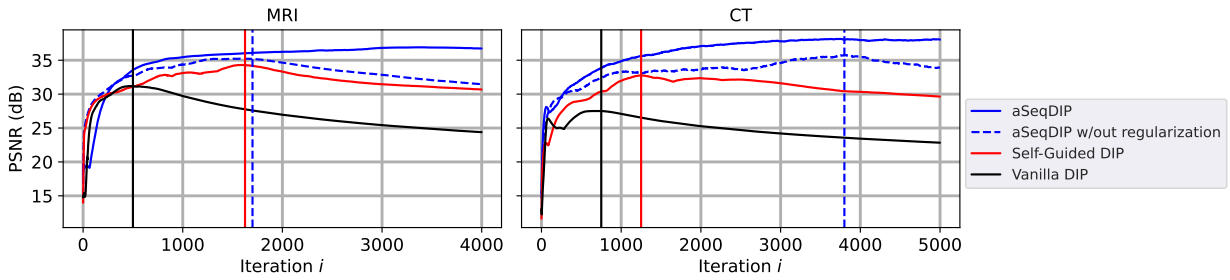


Figure 5.4 Average PSNR results w.r.t. iteration i of 20 MRI (with 4x) scans (*left*) and 20 CT (with 18 views) scans (*right*) to show the impact of the proposed autoencoding regularization term on noise overfitting in aSeqDIP. Furthermore, average results of Vanilla DIP and Self-Guided DIP are also reported for comparison. For aSeqDIP, iteration $i \in [NK]$, where $N = 2$. Vertical lines approximately indicate the start of the PSNR decay for every case. In Appendix B.2.1, we include the PSNR curves of aSeqDIP and other DIP-based methods for the task of denoising.

Self-Guided DIP experiences PSNR decay earlier, after approximately 1250 iterations. Importantly, when the autoencoding term is utilized (black solid lines), not only does the decay in noise overfitting not commence until after iteration 4000, but the reconstruction quality (measured by PSNR) also improves.

As expected, PSNR decay in Vanilla DIP begins early, at around iteration 500 and 750 for MRI and CT, respectively. In Appendix B.2.4, we provide an ablation study to better show the impact of the value of λ in aSeqDIP.

5.3.3 Main Results

Here, we present our primary results regarding the reconstruction quality, measured by PSNR and SSIM, as well as the associated run-time. Table 5.2 presents the results for the considered tasks in this paper. Column 3 indicates whether the baselines depend on prior data or pre-trained models. The last three columns provide the PSNR, SSIM, and run-time results where the arrows indicate favorable results. For PSNR and SSIM, the settings correspond to the second column of Table 5.1. The black (resp. black) text corresponds to the first (resp. second) setting. Values after the ± sign indicate standard deviation. Subsequently, we offer observations on the main results.

Compared to data-independent methods, i.e., the baselines that do not depend on a reference image or pre-trained models, aSeqDIP demonstrates improved PSNR and SSIM scores. For example, aSeqDIP, apart from Self-Guided DIP, shows nearly a 1dB improvement for MRI 8X acceleration

Task	Method	Data Independency	**PSNR** (dB) (↑) (Setting 1, Setting 2)	**SSIM** $\in [0,1]$ (↑) (Setting 1, Setting 2)	**Run-time** (↓) (minutes)
MRI	Score-MRI	×	(31.51±0.45, 29.61±0.44)	(0.891±0.012, 0.862±0.014)	6.2±0.12
	Ref-Guided DIP	×	(33.17±0.27, 30.23±0.24)	(0.912±0.021, 0.873±0.016)	2.5±0.2
	TV-DIP	✓	(30.52±0.25, 29.20±0.37)	(0.872±0.022, 0.852±0.022)	2.5±0.1
	ES-DIP	✓	(31.02±0.34, 29.44±0.45)	(0.882±0.031, 0.858±0.028)	1.56±0.34
	Vanilla DIP	✓	(30.21±0.42, 28.75±0.33)	(0.865±0.02, 0.842±0.022)	**1.5±0.12**
	Self-Guided DIP	✓	(33.6±0.23, 30.75±0.25)	(0.922±0.008, 0.874±0.006)	4.5±0.67
	aSeqDIP (Ours)	✓	(**34.08±0.41**, **31.34±0.47**)	(**0.929±0.008**, **0.887±0.009**)	2.2±0.12
CT	MCG	×	(32.82±0.52, 31.35±0.49)	(0.912±0.08, 0.852±0.09)	6.4±0.2
	FBP	✓	(22.92±0.22, 19.52±0.32)	(0.75±0.021, 0.68±0.023)	**0.2±0.01**
	Ref-Guided DIP	×	(31.21±0.24, 28.31±0.42)	(0.892±0.023, 0.842±0.021)	2.5±0.42
	Vanilla DIP	✓	(26.21±0.12, 24.31±0.34)	(0.791±0.021, 0.772±0.012)	1.5±0.21
	Self-Guided DIP	✓	(33.95±0.32, 31.95±0.32)	(0.918±0.02, 0.872±0.031)	4.5±0.56
	aSeqDIP (Ours)	✓	(**34.88±0.36**, **33.09±0.39**)	(**0.941±0.026**, **0.92±0.022**)	2.2±0.42
Denoising	DPS	×	(31.02±0.25, 28.2±0.31)	(0.912±0.02, 0.882±0.021)	2.5±0.17
	Vanilla DIP	✓	(30.48±0.28, 27.84±0.32)	(0.905±0.021, 0.871±0.030)	**1.5±0.22**
	SGLD DIP	✓	(30.58±0.34, 28.12±0.42)	(0.908±0.021, 0.877±0.017)	3.2±0.24
	TV-DIP	✓	(30.57±0.31, 28.47±0.26)	(0.914±0.022, 0.882±0.014)	2.5±0.24
	Rethinking-DIP	✓	(30.98±0.31, 28.67±0.25)	(0.912±0.02, 0.887±0.03)	2.5±0.34
	ES-DIP	✓	(31.11±0.23, 28.12±0.41)	(0.914±0.017, 0.886±0.024)	1.45±0.44
	Self-Guided DIP	✓	(31,21±0.26, 28.31±0.35)	(0.916±0.02, 0.891±0.03)	3.5±0.45
	aSeqDIP (Ours)	✓	(**31.51±0.34**, **28.97±0.44**)	(**0.926±0.021**, **0.908±0.031**)	2.4±0.45
In-Painting	DPS	×	(23.9±0.45, 22.03±0.36)	(0.817±0.023, 0.762±0.021)	2.5±0.3
	Vanilla DIP	✓	(22.56±0.31, 21.32±0.67)	(0.754±0.023, 0.721±0.012)	**1.5±0.35**
	SGLD DIP	✓	(23.09±0.55, 21.41±0.45)	(0.772±0.023, 0.732±0.041)	2.5±0.45
	TV-DIP	✓	(22.87±0.45, 21.64±0.51)	(0.774±0.04, 0.742±0.042)	2.5±0.31
	ES-DIP	✓	(23.33±0.44, 21.89±0.28)	(0.781±0.034, 0.745±0.041)	1.25±0.55
	Self-Guided DIP	✓	(23.84±0.43, 21.78±0.52)	(0.792±0.042, 0.752±0.064)	3.5±0.45
	aSeqDIP (Ours)	✓	(**24.56±0.45**, **22.57±0.47**)	(**0.838±0.051**, **0.778±0.045**)	2.4±0.54
Deblurring	DPS	×	(23.40±0.56)	(0.776±0.032)	2.24±0.65
	SGLD DIP	✓	(19.80±0.43)	(0.720±0.03)	3.24±0.55
	Self-Guided DIP	✓	(20.34±0.55)	(0.732±0.025)	3.4±1.02
	aSeqDIP (Ours)	✓	(**23.89±0.40**)	(**0.792±0.033**)	2.5±0.78

Table 5.2 Average PSNR, SSIM, and run-time results reported by our method against the selected baselines for the tasks of MRI reconstruction, CT reconstruction, image denoising, in-painting, and non-linear deblurring. '**Data-Independency**' in column 3 indicates whether the methods depend on prior data or pre-trained models. Setting 1 and Setting 2, in the fourth and fifth columns correspond to the scenarios in the second column of Table 5.1. For tasks with two settings, the run-time results are averaged over the two settings. Values past ± represent the standard deviation. See Appendix B.2.2 and Appendix B.2.3 for more comparison results.

compared to conventional methods. For the task of 30-views CT, aSeqDIP reports SSIM score of 0.92 which is 5% more than the second best, which is Self-Guided DIP with SSIM of 0.872. Although improvements against Self-Guided DIP are generally marginal in terms of reconstruction quality, our method proves to be 2X faster for MRI and CT reconstruction and requires 1 minute less than Self-Guided DIP for denoising and in-painting. This speed-up is attributed to updating

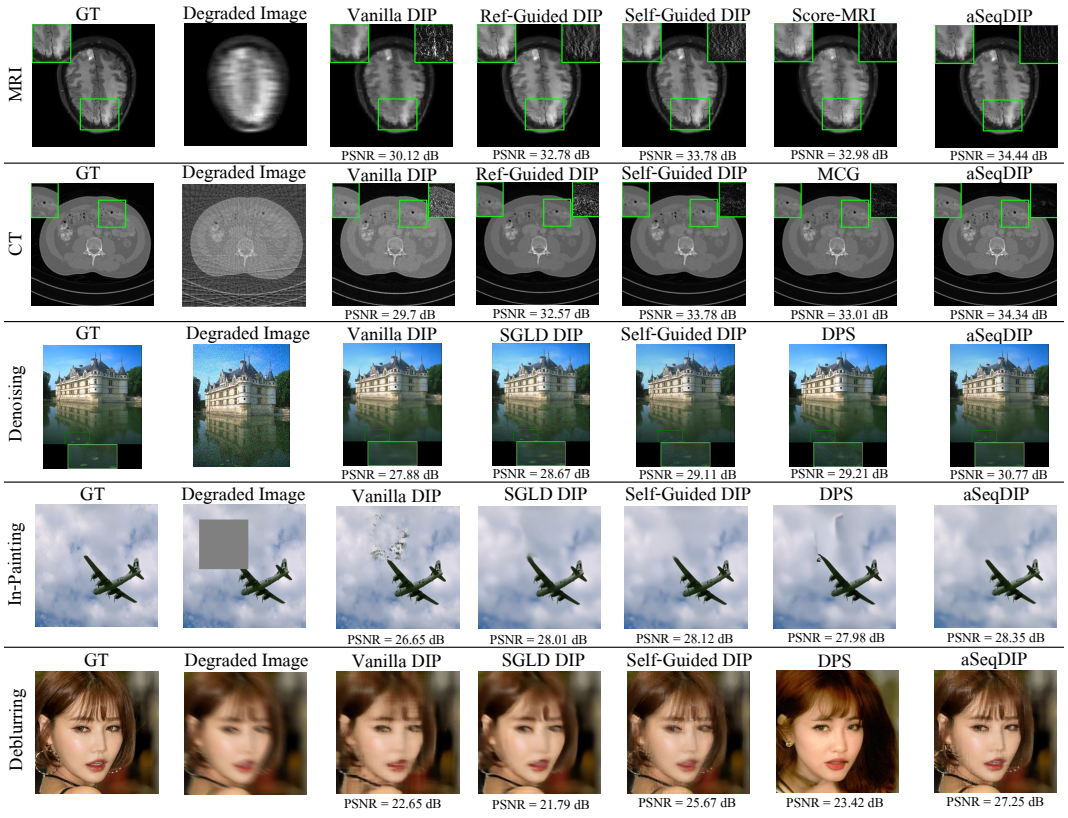


Figure 5.5 Reconstructed/recovered images using our proposed approach, aSeqDIP, and the baselines for the considered tasks. The ground truth (GT) and degraded images are shown in the first and second columns, respectively, followed by three or four baselines per task. The last column presents our method. PSNR results are given at the bottom of each reconstructed image. For MRI (8x undersampling) and CT (18 views), the top right box shows the absolute difference between the center region box of the reconstructed image and the same region in the GT image. Denoising and in-painting used $\sigma_d = 25$ and HIAR = 0.25. For the task of Deblurring, aSeqDIP contains artifacts when compared to DPS. However, DPS generates a perceptually different image when compared to the GT. For all other tasks, aSeqDIP reconstructions contain sharper and clearer image features than other methods.

the input using one forward pass of the trained network at each iteration k, instead of computing gradients with respect to the input for the update. Compared to Vanilla DIP, our method, on average, only requires an additional 30 to 60 seconds. When compared to ES-DIP (Wang et al., 2023a), our method requires longer time, but on average achieves better reconstruction results across three tasks and different settings.

In comparison to data-dependent methods such as Score-MRI and MCG, our approach not only yields the best PSNR and SSIM but also requires reduced run-time, all without requiring any training data or pre-trained models. For instance, on average, aSeqDIP achieves nearly a 2dB improvement

in 30-views CT compared to MCG while being 2X faster. In comparison to DPS, on average, our method report higher SSIM. Our method requires slightly less run-time on average but enhances the PSNR by approximately 0.6dB for both denoising and in-painting. Notably, our method is an optimization-based approach, whereas DM-based methods only require function evaluations. However, the generally larger run-time reported for DM-based methods is due to the necessity of running a large number of reverse sampling steps. When compared to Ref-Guided DIP, our method achieves higher PSNR and SSIM results without the need for any prior (or reference image) image.

5.3.4 Visualizations

Figure 5.5 shows reconstructed images for the five considered tasks using aSeqDIP and the other baselines. Each row corresponds to a task. The first column displays the ground truth (GT) image whereas the second column shows the degraded image. Column 3 to the column before last present the reconstructed images by the baselines, while the last column shows the reconstructed images by aSeqDIP. PSNR values are provided at the bottom of each reconstructed image.

As observed, aSeqDIP achieves the highest PSNR scores. Additionally, the top right green boxes, which show the difference between the central region of the reconstructed and GT images, indicate that for MRI and CT, our method visually exhibits the least difference, making it the closest to the GT.

A similar observation is seen for the denoising task for the zoomed in bottom box. For inpainting, we note that aSeqDIP introduces the fewest unwanted artifacts as observed in the clouds (for DPS), and the left wing of the plane. While aSeqDIP contains artifacts for the task of Deblurring when compared to DPS, the latter generates a perceptually different image when compared to the GT. Similar observation are noticed with the additional visualizations provided in Appendix B.3.

5.4 Conclusions & Future Work

In this paper, we introduced Autoencoding Sequential Deep Image Prior (aSeqDIP), a new unsupervised image recovery algorithm. Notably, aSeqDIP operates without the need of pre-trained models, relying solely on a sequential update of network parameters. These parameters are optimized using an input-adaptive data consistency objective combined with autoencoding regularization,

effectively mitigating noise overfitting. Our experimental results across various tasks highlight the competitive performance of the proposed algorithm, matching (or outperforming) diffusion-based methods in terms of reconstruction quality and required run time, all without the need for pre-trained models.

For future directions, we aim to explore the applicability of aSeqDIP to other image recovery problems, thereby expanding its versatility and potential impact across diverse domains. Additionally, we are interested in investigating the integration of a network input update mechanism to dynamically adjust the autoencoding regularization parameter and the number of gradient updates per iteration.

CHAPTER 6

MRI RECONSTRUCTION BY SMOOTHED UNROLLING

6.1 Introduction

After the last charpter, we will explore the another direction of my research that focus on enhancing the robustness of the deep learning method. As the popularity of deep learning (DL) in the field of magnetic resonance imaging (MRI) continues to rise, recent research has indicated that DL-based MRI reconstruction models might be excessively sensitive to minor input disturbances, including worst-case or random additive perturbations. This sensitivity often leads to unstable aliased images. This raises the question of how to devise DL techniques for MRI reconstruction that can be robust to these variations. To address this problem, we propose a novel image reconstruction framework, termed Smoothed Unrolling (SMUG), which advances a deep unrolling-based MRI reconstruction model using a randomized smoothing (RS)-based robust learning approach. RS, which improves the tolerance of a model against input noise, has been widely used in the design of adversarial defense approaches for image classification tasks. Yet, we find that the conventional design that applies RS to the entire DL-based MRI model is ineffective. In this paper, we show that SMUG and its variants address the above issue by customizing the RS process based on the unrolling architecture of DL-based MRI reconstruction models. We theoretically analyze the robustness of our method in the presence of perturbations. Compared to vanilla RS and other recent approaches, we show that SMUG improves the robustness of MRI reconstruction with respect to a diverse set of instability sources, including worst-case and random noise perturbations to input measurements, varying measurement sampling rates, and different numbers of unrolling steps.

6.2 Preliminaries and Problem Statement

6.2.1 Setup of MRI Reconstruction

Many medical imaging approaches involve ill-posed inverse problems such as the work in (Donoho, 2006a), where the aim is to reconstruct the original signal $\mathbf{x} \in \mathbb{C}^q$ (vectorized image) from undersampled k-space (Fourier domain) measurements $\mathbf{y} \in \mathbb{C}^p$ with $p < q$. The imaging system in MRI can be modeled as a linear system $\mathbf{y} \approx \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ may take on different

forms for single-coil or parallel (multi-coil) MRI, etc. For example, in the single coil Cartesian MRI acquisition setting, $\mathbf{A} = \mathbf{MF}$, where $\mathbf{F}$ is the 2D discrete Fourier transform and $\mathbf{M}$ is a masking operator that implements undersampling. With the linear observation model, MRI reconstruction is often formulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \mathcal{R}(\mathbf{x}), \tag{6.1}$$

where $\mathcal{R}(\cdot)$ is a regularization function (*e.g.*, ℓ_1 norm in the wavelet domain to impose a sparsity prior (Mihcak et al., 1999)), and $\lambda > 0$ is the regularization parameter.

MoDL (Aggarwal et al., 2019a) is a recent popular supervised deep learning approach inspired by the MR image reconstruction optimization problem in (6.1). MoDL combines a denoising network with a data-consistency (DC) module in each iteration of an unrolled architecture. In MoDL, the hand-crafted regularizer, $\mathcal{R}$, is replaced by a learned network-based prior $\|\mathbf{x} - \mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2$ involving a network $\mathcal{D}_{\boldsymbol{\theta}}$.

MoDL attempts to optimize this loss by initializing $\mathbf{x}^0 = \mathbf{A}^H\mathbf{y}$, and then iterating the following process for a number of unrolling steps indexed by $n \in \{0, \ldots, N-1\}$. Specifically, MoDL iterations are given by

$$\boldsymbol{x}^{n+1} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^n)\|_2^2. \tag{6.2}$$

After N iterations, we denote the final output of MoDL as $\boldsymbol{x}^N = \boldsymbol{F}_{\text{MoDL}}(\boldsymbol{x}^0)$. The weights of the denoiser are shared across the N blocks and are learned in an end-to-end supervised manner (Aggarwal et al., 2019a).

6.2.2 Lack of Robustness of DL-based Reconstructors

In (Antun et al., 2020a), it was demonstrated that deep learning-based MRI reconstruction can exhibit instability, when confronted with subtle, nearly imperceptible input perturbations. These perturbations are commonly referred to as 'adversarial perturbations' and have been extensively investigated in the context of DL-based image classification tasks, as outlined in (I et al., 2015). In the context of MRI, these perturbations represent the worst-case additive perturbations, which can be used to evaluate method sensitivity and robustness (Antun et al., 2020a; Jia et al., 2022a).

Let $\boldsymbol{\delta}$ denote a small perturbation of the measurements that falls in an ℓ_∞ ball of radius ϵ, *i.e.*, $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$. Adversarial disturbances then correspond to the worst-case input perturbation vector $\boldsymbol{\delta}$ that maximizes the reconstruction error, *i.e.*,

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \|\boldsymbol{F}_{\text{MoDL}}(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta})) - \mathbf{t}\|_2^2, \tag{6.3}$$

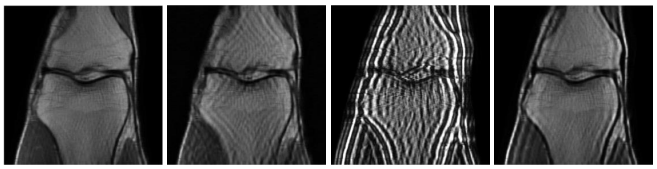
where $\mathbf{t}$ is a ground truth target image from the training set (*i.e.*, label). The operator $\mathbf{A}^H$ transforms the measurements $\mathbf{y}$ to the image domain, and $\mathbf{A}^H\mathbf{y}$ is the input (aliased) image to the reconstruction model. The optimization problem in (6.3) can be effectively solved using the iterative projected gradient descent (PGD) method (Madry et al., 2017). In **Fig. 6.1-(a)** and **(b)**, we show reconstructed images using MoDL originating from a benign (i.e., undisturbed) input and a PGD scheme-perturbed input, respectively. It is evident that the worst-case input disturbance significantly deteriorates the quality of the reconstructed image. While one focus of this work is to enhance robustness against input perturbations, **Fig.6.1-(c)** and **(d)** highlight two additional potential sources of instability that the reconstructor (MoDL) can encounter during testing: variations in the measurement sampling rate (resulting in "perturbations" to the sparsity of the sampling mask in $\mathbf{A}$) (Antun et al., 2020a), and changes in the number of unrolling steps (Gilton et al., 2021a). In scenarios where the sampling mask (**Fig.6.1-(c)**) or number of unrolling steps (**Fig.6.1-(d)**) deviate from the settings used during MoDL training, we observe a significant degradation in performance compared to the original setup (**Fig.6.1-(a)**), even in the absence of additive measurement perturbations. In Section 6.4, we demonstrate how our method improves the reconstruction robustness in the presence of different types of perturbations, including those in **Fig.6.1**.

6.2.3 Randomized Smoothing (RS)

Randomized smoothing, introduced in (Cohen et al., 2019), enhances the robustness of DL models against noisy inputs. It is implemented by generating multiple randomly modified versions of the input data and subsequently calculating an averaged output from this diverse set of inputs.

Given some function $f(\mathbf{x})$, RS formally replaces f with a smoothed version

$$g(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\eta}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}\left[f(\mathbf{x}+\boldsymbol{\eta})\right], \tag{6.4}$$



(a) (b) (c) (d)

Figure 6.1 MoDL's instabilities resulting from perturbations to input data, the measurement sampling rate, and the number of unrolling steps used at testing phase shown on an image from the `fastMRI` dataset (Zbontar et al., 2018). We refer readers to Section 6.4 for further details about the experimental settings. (a) MoDL reconstruction from benign (*i.e.*, without additional noise/perturbation) measurements with 4× acceleration (*i.e.*, 25% sampling rate) and 8 unrolling steps. (b) MoDL reconstruction from disturbed input with perturbation strength $\epsilon = 0.02$ (see Section 6.4.1). (c) MoDL reconstruction from clean measurements with 2× acceleration (*i.e.*, 50% sampling), and using 8 unrolling steps. (d) MoDL reconstruction from clean or unperturbed measurements with 4× acceleration and 16 unrolling steps. In (b), (c), and (d), the network trained in (a) is used.

where $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ denotes a Gaussian distribution with zero mean and element-wise variance σ^2, and $\mathbf{I}$ denotes the identity matrix of appropriate size. Prior research has shown that RS has been effective as an adversarial defense approach in DL-based image classification tasks (Cohen et al., 2019; Salman et al., 2020; Zhang et al., 2022). However, the question of whether RS can significantly improve the robustness of MoDL and other image reconstructors has not been thoroughly explored. A preliminary investigation in this area was conducted by (Wolf, 2019), which demonstrated the integration of RS into MR image reconstruction in an end-to-end (E2E) setting. We can formulate image reconstruction using RS-E2E as

$$\boldsymbol{x}_{\text{RS-E2E}} = \mathbb{E}_{\boldsymbol{\eta}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}[\boldsymbol{F}_{\text{MoDL}}(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\eta}))]. \tag{RS-E2E}$$

This formulation aligns with the one used in (Wolf, 2019), where the random noise vector $\boldsymbol{\eta}$ is directly added to $\mathbf{y}$ in the frequency domain (complex-valued), followed by multiplication with $\mathbf{A}^H$ to obtain the input image for MoDL. The noisy measurements are also utilized in each iteration in MoDL. RS-E2E can be identically formulated for alternative reconstruction models.

Fig. 6.2 shows a block diagram of RS-E2E-backed MoDL. This RS-integrated MoDL is trained with supervision in the standard manner. Although RS-E2E represents a straightforward application of RS to MoDL, it remains unclear if the formulation in (RS-E2E) is the most effective method to incorporate RS into unrolled algorithms such as MoDL, considering the latter's specialties, e.g., the

involved denoising and the data-consistency (DC) steps.

As such, for the rest of the paper, we focus on studying the following questions **(Q1)**–**(Q4)**.

- **(Q1)**: *How should RS be integrated into an unrolled algorithm such as MoDL?*
- **(Q2)**: *How do we learn the network $\mathcal{D}_{\boldsymbol{\theta}}(\cdot)$ in the presence of RS operations?*
- **(Q3)**: *Can we prove the robustness of SMUG in the presence of data perturbations?*
- **(Q4)**: *Can we further improve the RS operation in SMUG for enhanced image quality or sharpness?*

6.3 Methodology

In this section, we address questions **(Q1)**–**(Q4)** by taking the unrolling characteristics of MoDL into the design of an RS-based MRI reconstruction. The proposed novel integration of RS with MoDL is termed Smoothed Unrolling (**SMUG**). We also explore an extension to SMUG. We note that while we develop our methods based on MoDL, in the last subsection, we discuss incorporating our approaches within other unrolling methods such as ISTA-Net.

6.3.1 Solution to (Q1): RS at intermediate unrolled denoisers

As illustrated in **Fig.,6.2**, the RS operation in RS-E2E is typically applied to MoDL in an end-to-end manner. This does not shed light on which component of MoDL needs to be made more robust. Here, we explore integrating RS at each intermediate unrolling step of MoDL.

In this subsection, we present SMUG, which applies RS to the denoising network. This seemingly simple modification is related to a robustness certification technique known as "denoised smoothing" (Salman et al., 2020). In this technique, a smoothed denoiser is used, proving to be sufficient for establishing robustness in the model. We use $\boldsymbol{x}_{\mathrm{S}}^{n}$ to denote the n-th iteration of SMUG. Starting from $\boldsymbol{x}_{\mathrm{S}}^{0} = \boldsymbol{A}^{H}\boldsymbol{y}$, the procedure is given by

$$\boldsymbol{x}_{\mathrm{S}}^{n+1} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x} - \mathbb{E}_{\boldsymbol{\eta}}\left[\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\mathrm{S}}^{n} + \boldsymbol{\eta})\right]\|_2^2 , \tag{6.5}$$

where $\boldsymbol{\eta}$ is drawn from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. After $N-1$ iterations, the final output of SMUG is denoted by $\boldsymbol{x}_{\mathrm{S}}^{N} = \boldsymbol{F}_{\mathrm{SMUG}}(\boldsymbol{x}^{0})$. **Fig. 6.3** presents the architecture of SMUG.

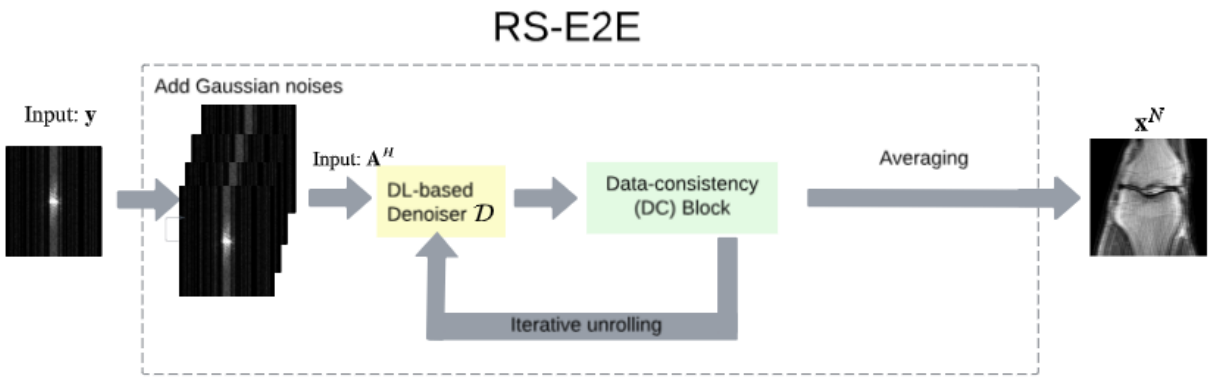


Figure 6.2 A schematic overview of RS-E2E. Here, iterative unrolling takes place between the data consistency and denoising blocks for multiple noisy versions of the input.

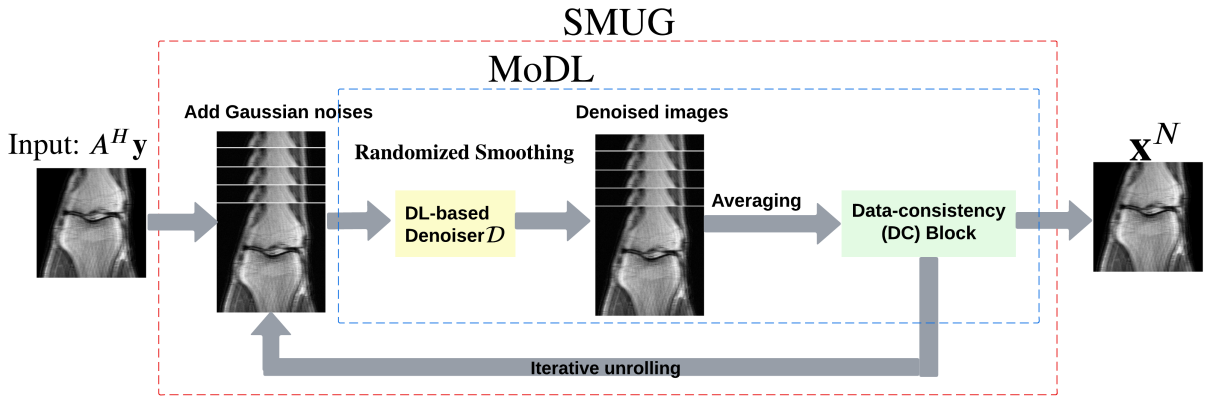


Figure 6.3 Architecture of SMUG. Here, for every unrolling step, after applying the denoiser on each noisy version of the input, the data consistency is applied on the average of the denoised images.

6.3.2 Solution to (Q2): SMUG's pre-training & fine-tuning

In this subsection, we develop the training scheme of SMUG. Inspired by the currently celebrated "pre-training + fine-tuning" technique (Zoph et al., 2020; Salman et al., 2020), we propose to train SMUG following this learning paradigm. Our rationale is that pre-training can provide a robustness-aware initialization of the DL-based denoising network for fine-tuning. To pre-train the denoising network $\mathcal{D}_{\boldsymbol{\theta}}$, we consider a mean squared error (MSE) loss that measures the Euclidean distance between images denoised by $\mathcal{D}_{\boldsymbol{\theta}}$ and the target (ground truth) images, denoted by $\mathbf{t}$. This leads to the **pre-training** step:

$$\boldsymbol{\theta}_{\text{pre}} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{t}\in\mathcal{T}}\left[\mathbb{E}_{\boldsymbol{\eta}}||\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{t}+\boldsymbol{\eta}) - \mathbf{t}||_2^2\right], \tag{6.6}$$

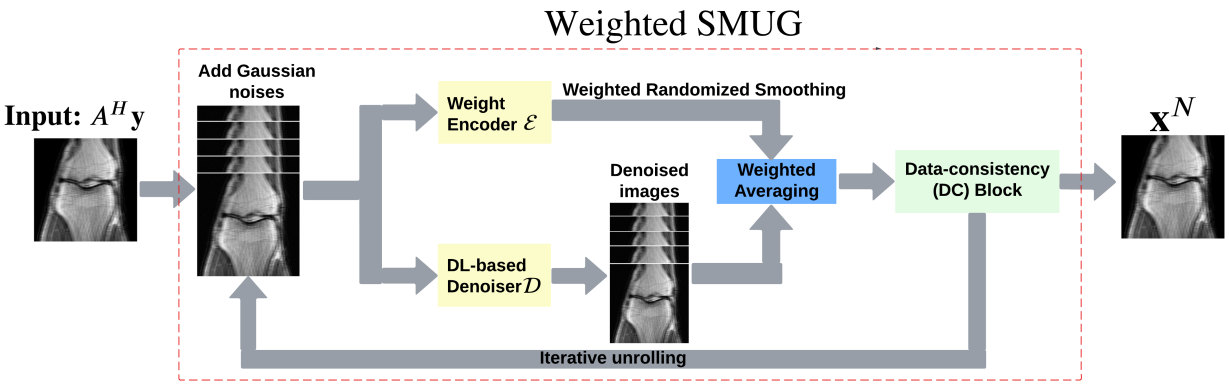


Figure 6.4 Architecture of weighted SMUG. Here, we extend SMUG by including the weight encoder and the use of weighted randomized smoothing.

where $\mathcal{T}$ is the set of ground truth images in the training dataset. Next, we develop the fine-tuning scheme to improve $\boldsymbol{\theta}_{\text{pre}}$ based on the labeled/paired MRI dataset. Since RS in SMUG (**Fig. 6.3**) is applied to every unrolling step, we propose an *unrolled stability (**UStab**) loss* for fine-tuning $\mathcal{D}_{\boldsymbol{\theta}}$:

$$\ell_{\text{UStab}}(\boldsymbol{\theta};\mathbf{y},\mathbf{t}) = \sum_{n=0}^{N-1} \mathbb{E}_{\boldsymbol{\eta}} ||\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}^n + \boldsymbol{\eta}) - \mathcal{D}_{\boldsymbol{\theta}}(\mathbf{t})||_2^2 \,. \tag{6.7}$$

The UStab loss in (6.7) relies on the target images, bringing in a key benefit: the denoising stability is guided by the reconstruction accuracy of the ground-truth image, yielding a graceful trade-off between robustness and accuracy.

Integrating the UStab loss, defined in (6.7), with the standard reconstruction loss, we obtain the **fine-tuned** $\boldsymbol{\theta}$ by minimizing $\mathbb{E}_{(\mathbf{y},\mathbf{t})}[\ell(\boldsymbol{\theta};\mathbf{y},\mathbf{t})]$, where

$$\ell(\boldsymbol{\theta};\mathbf{y},\mathbf{t}) = \ell_{\text{UStab}}(\boldsymbol{\theta};\mathbf{y},\mathbf{t}) + \lambda_\ell \|\boldsymbol{F}_{\text{SMUG}}(\mathbf{A}^H\mathbf{y}) - \mathbf{t}\|_2^2, \tag{6.8}$$

with $\lambda_\ell > 0$ representing a regularization parameter to strike a balance between the reconstruction error (for accuracy) and the denoising stability (for robustness) terms. We initialize $\boldsymbol{\theta}$ as $\boldsymbol{\theta}_{\text{pre}}$ when optimizing (6.8) using standard optimizers such as Adam (Kingma and Ba, 2015a).

In practice, the same dataset is used for fine-tuning as pre-training because the pre-trained model is initially trained solely as a denoiser, while the fine-tuning process aims at integrating the entire regularization strategy applied to the MoDL framework. This approach ensures that the fine-tuning optimally adapts the model to the specific enhancements introduced by our robustification strategies.

6.3.3 Answer to (Q3): Analyzing the robustness of SMUG in the presence of data perturbations

The following theorem discusses the robustness (i.e., sensitivity to input perturbations) achieved with SMUG. Note that all norms on vectors (resp. matrices) denote the ℓ_2 norm (resp. spectral norm) unless indicated otherwise.

Theorem 6.3.1. Assume the denoiser network's output is bounded in norm. Given the initial input image $\mathbf{A}^H\mathbf{y}$ obtained from measurements $\mathbf{y}$, let the SMUG reconstructed image at the n-th unrolling step be $\mathbf{x}_{\mathrm{S}}^n(\mathbf{A}^H\mathbf{y})$ with RS variance of σ^2. Let $\boldsymbol{\delta}$ denote an additive perturbation to the measurements $\mathbf{y}$. Then,

$$\|\mathbf{x}_{\mathrm{S}}^n(\mathbf{A}^H\mathbf{y}) - \mathbf{x}_{\mathrm{S}}^n(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}))\| \leq C_n\|\boldsymbol{\delta}\|, \tag{6.9}$$

where $C_n = \alpha\|\mathbf{A}\|_2\left(\frac{1-\left(\frac{M\alpha}{\sqrt{2\pi}\sigma}\right)^n}{1-\frac{M\alpha}{\sqrt{2\pi}\sigma}}\right) + \|\mathbf{A}\|_2\left(\frac{M\alpha}{\sqrt{2\pi}\sigma}\right)^n$, with $\alpha = \|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2$ and

$$M = 2\max_{\boldsymbol{x}}(\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x})\|)$$

. The proof is provided in the Appendix. Note that the output of SMUG $\mathbf{x}_{\mathrm{S}}^n(\cdot)$ depends on both the initial input (here $\mathbf{A}^H\mathbf{y}$) and the measurements $\mathbf{y}$. We abbreviated it to $\mathbf{x}_{\mathrm{S}}^n(\mathbf{A}^H\mathbf{y})$ in the theorem and proof for notational simplicity. The constant C_n depends on the number of iterations or unrolling steps n as well as the RS standard deviation parameter σ. For large σ, the robustness error bound for SMUG clearly decreases as the number of iterations n increases. In particular, if $\sigma > M\alpha/\sqrt{2\pi}$, then as $n \to \infty$, $C_n \to \alpha\|\mathbf{A}\|_2 / \left(1 - \frac{M\alpha}{\sqrt{2\pi}\sigma}\right)$. Furthermore, as $\sigma \to \infty$, $C_n \to C \triangleq \alpha\|\mathbf{A}\|_2$. Clearly, if $\alpha \leq 1$ and $\|\mathbf{A}\|_2 \leq 1$ (normalized), then $C \leq 1$.

Thus, for sufficient smoothing, the error introduced in the SMUG output due to input perturbation never gets worse than the size of the input perturbation. Therefore, the output is stable with respect to (w.r.t.) perturbations. These results corroborate experimental results in Section 6.4 on how SMUG is robust (whereas other methods, such as vanilla MoDL, breakdown) when increasing the number of unrolling steps at test time, and is also more robust for larger σ (with good accuracy-robustness trade-off).

The only assumption in our analysis is that the denoiser network output is bounded in norm. This consideration is handled readily when the denoiser network incorporates bounded activation functions such as the sigmoid or hyperbolic tangent. Alternatively, if we expect image intensities to lie within a certain range, a simple clipping operation in the network output would ensure boundedness for the analysis.

A key distinction between SMUG and prior works, such as RS-E2E (Wolf, 2019), is that smoothing is performed in every iteration. Moreover, while (Wolf, 2019) assumes the end-to-end mapping is bounded, in MoDL or SMUG, it clearly isn't because the data-consistency step's output is unbounded as $\mathbf{y}$ grows.

We remark that our intention with Theorem 1 is to establish a baseline of robustness intrinsic to models with unrolling architectures.

6.3.4 Solution to (Q4): Weighted Smoothing

In this subsection, we present a modified formulation of randomized smoothing to improve its performance in SMUG. Randomized smoothing in practice involves uniformly averaging images denoised with random perturbations. This can be viewed as a type of mean filter, which leads to oversmoothing of structural information in practice. As such, we propose weighted randomized smoothing, which employs an encoder to assess a weighting (scalar) for each denoised image and subsequently applies the optimal weightings while aggregating images to enhance the reconstruction performance. Our method not only surpasses the SMUG technique but also excels in enhancing image sharpness across various types of perturbation sources. This allows for a more versatile or flexible and effective approach for improving image quality under different conditions.

The weighted randomized smoothing operation applied on a function $f(\cdot)$ is as follows:

$$g_{\mathrm{w}}(\boldsymbol{x}) := \frac{\mathbb{E}_{\boldsymbol{\eta}}[w(\mathbf{x}+\boldsymbol{\eta})f(\mathbf{x}+\boldsymbol{\eta})]}{\mathbb{E}_{\boldsymbol{\eta}}[w(\mathbf{x}+\boldsymbol{\eta})]}\,, \tag{6.10}$$

where $w(\cdot)$ is an input-dependent weighting function.

Based on the weighted smoothing in (6.10), we introduce **Weighted SMUG**. This approach involves applying weighted RS at each denoising step, and the weighting encoder is trained in conjunction with the denoiser during the fine-tuning stage. For the weighting encoder in

our experiments, we use a simple architecture consisting of five successive convolution, batch normalization, and ReLU activation layers followed by a linear layer and Sigmoid activation. Specifically, in the n-th unrolling step, we use a weighting encoder $\mathcal{E}_{\phi}$, parameterized by ϕ, to learn the weight of each image used for (weighted) averaging. Here, we use $\boldsymbol{x}_{\mathrm{W}}^{n}$ to denote the output of the n-th block. Initializing $\boldsymbol{x}_{\mathrm{W}}^{0} = \boldsymbol{A}^{H}\boldsymbol{y}$, the output of Weighted SMUG w.r.t. n is

$$\begin{aligned} \boldsymbol{x}_{\mathrm{W}}^{n+1} &= \arg\min_{\boldsymbol{x}} \ \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \\ &\lambda \left\| \boldsymbol{x} - \frac{\mathbb{E}_{\boldsymbol{\eta}}[\mathcal{E}_{\phi}(\mathbf{x}_{\mathrm{W}}^{n}+\boldsymbol{\eta})\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_{\mathrm{W}}^{n}+\boldsymbol{\eta})]}{\mathbb{E}_{\boldsymbol{\eta}}[\mathcal{E}_{\phi}(\mathbf{x}_{\mathrm{W}}^{n}+\boldsymbol{\eta})]} \right\|_2^2 . \end{aligned} \tag{6.11}$$

After N iterations, the final output of Weighted SMUG is $\boldsymbol{x}_{\mathrm{W}}^{N} = \boldsymbol{F}_{\mathrm{wSMUG}}(\boldsymbol{x}^{0})$.

Figure 6.4 illustrates the block diagram of weighted SMUG.

Furthermore, we extend the "pre-training + fine-tuning" approach proposed in Section 6.3.2 to the Weighted SMUG method.

In this case, we obtain the **fine-tuned** $\boldsymbol{\theta}$ and ϕ by using

$$\min_{\boldsymbol{\theta},\phi} \mathbb{E}_{(\mathbf{y},\mathbf{t})}[\lambda_l \|F_{\mathrm{wSMUG}}(\mathbf{A}^{H}\mathbf{y}) - \mathbf{t}\|_2^2 + \ell_{\mathrm{UStab}}(\boldsymbol{\theta};\mathbf{y},\mathbf{t})]. \tag{6.12}$$

6.3.5 Integrating RS into Other Unrolled Networks

In this subsection, we further discuss the extension of our SMUG schemes for other unrolling based reconstructors, using ISTA-Net (Zhang and Ghanem, 2018) as an example. The goal is to demonstrate the generality of our proposed approaches for deep unrolled models.

ISTA-Net uses a training loss function composed of discrepancy and constraint terms. In particular, it performs the following for N unrolling steps:

$$\boldsymbol{r}^{n} = \boldsymbol{x}^{n-1} - \lambda^{(n)}\boldsymbol{A}^{H}(\boldsymbol{A}\boldsymbol{x}^{n-1} - \boldsymbol{y}) \tag{6.13}$$

$$\boldsymbol{x}^{n} = \hat{\mathcal{F}}^{n}(\mathbf{Soft}(\mathcal{F}^{n}(\boldsymbol{r}^{n}), \theta^{n})) \ , \tag{6.14}$$

where $\hat{\mathcal{F}}$ and $\mathcal{F}$ involve two linear convolutional layers (without bias terms) separated by ReLU activations, and $\hat{\mathcal{F}}^{n} \circ \mathcal{F}^{n}$ are constrained close to the identity operator. The function **Soft** performs soft-thresholding with parameter θ^{n} (Zhang and Ghanem, 2018).

Similar to SMUG for MoDL, we integrate RS into the network-based regularization (denoising) component of ISTA-Net. This results in the following modification to (6.14):

$$\boldsymbol{x}^n = \mathbb{E}_{\boldsymbol{\eta}}[\hat{\mathcal{F}}^n(\mathbf{Soft}(\mathcal{F}^n(\boldsymbol{r}^n + \boldsymbol{\eta}), \theta^n))] \,, \tag{6.15}$$

where $\boldsymbol{\eta}$ is drawn from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. For weighted SMUG, (6.14) becomes

$$\boldsymbol{x}^n = \frac{\mathbb{E}_{\boldsymbol{\eta}}[\mathcal{E}_\phi(\boldsymbol{r}^n + \boldsymbol{\eta})\hat{\mathcal{F}}^n(\mathbf{Soft}(\mathcal{F}^n(\boldsymbol{r}^n + \boldsymbol{\eta}), \theta^n))]}{\mathbb{E}_{\boldsymbol{\eta}}[\mathcal{E}_\phi(\boldsymbol{r}^n + \boldsymbol{\eta})]} \,. \tag{6.16}$$

6.4 Experiments

6.4.1 Experimental Setup

6.4.1.1 Models & Sampling Masks

For the MoDL architecture, we use the recent state-of-the-art denoising network Deep iterative Down Network, which consists of 3 down-up blocks (DUBs) and 64 channels (Yu et al., 2019b). Additionally, for MoDL, we use $N = 8$ unrolling steps with denoising regularization parameter $\lambda = 1$. The conjugate gradient method (Aggarwal et al., 2019b), with a tolerance level of 10^{-6}, is utilized to execute the DC block. We used variable density Cartesian random undersampling masks in k-space, one for each undersampling factor that include a fully-sampled central k-space region and the remaining phase encode lines were sampled uniformly at random. The coil sensitivity maps for all scenarios were generated with the BART toolbox (Tamir et al., 2016). Extension to the ISTA-Net model is discussed in Section 6.4.7.

6.4.1.2 Baselines

We consider two robustification approaches: first is the RS-E2E method (Jia et al., 2022a) presented in (RS-E2E), and the second is Adversarial Training (AT) (Jia et al., 2022b). Furthermore, we consider other recent reconstruction models, specifically, the Deep Equilibrium (Deep-Eq) method (Gilton et al., 2021b) and a leading diffusion-based MRI reconstruction model from (Chung and Ye, 2022), which we denote as Score-MRI.

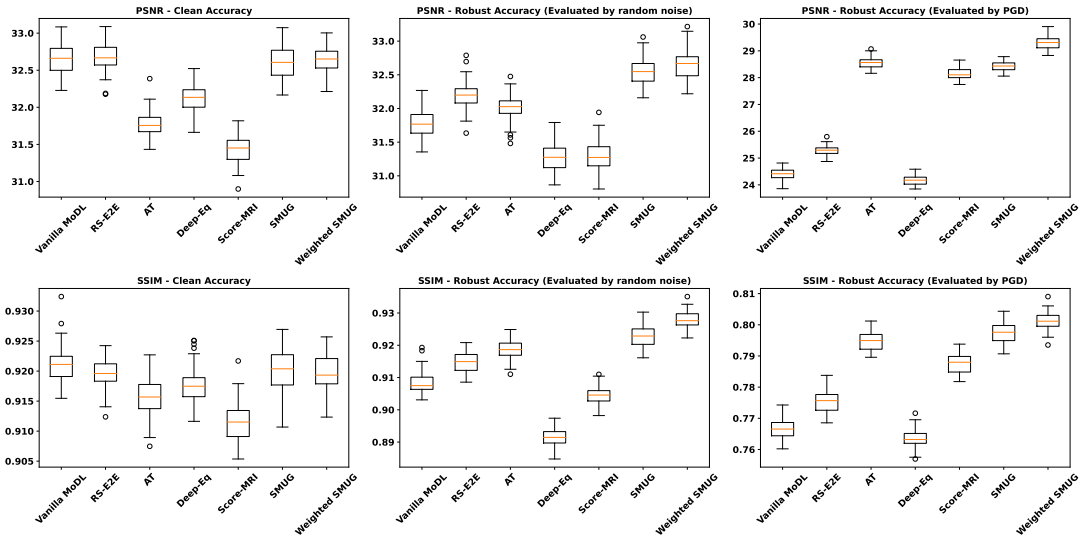


Figure 6.5 Reconstruction accuracy box plots for the fastMRI **brain** dataset with 4x acceleration factor. The additive random Gaussian noise of the second column plots is obtained using standard deviation of 0.01. The worst-case additive noise of the third column is obtained using the PGD method with $\epsilon = 0.02$.

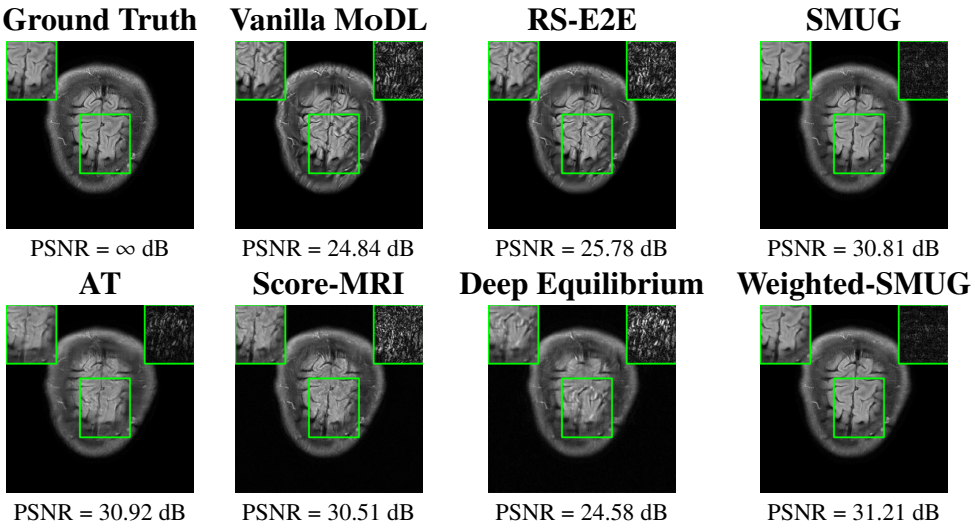


Figure 6.6 Visualization of ground truth and reconstructed images using different methods for 4x k-space undersampling, evaluated on PGD-generated worst-case inputs of perturbation strength $\epsilon = 0.02$.

6.4.1.3 Datasets & Training

For our study, we execute two experimental cases. For the first case, we utilize the `fastMRI` knee dataset, with 32 scans for validation and 64 unseen scans/slices for testing. In the second case, we employ our method for the `fastMRI` brain dataset. We used 3000 training scans in both cases. The k-space data are normalized so that the real and imaginary components are in the range $[-1, 1]$.

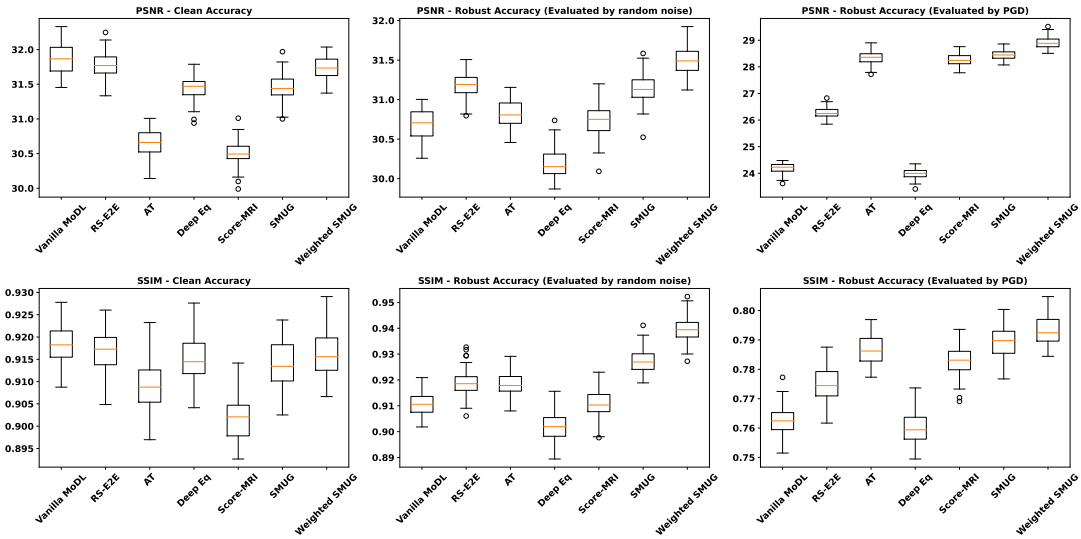


Figure 6.7 Reconstruction accuracy box plots for the fastMRI **knee** dataset with 4x Acceleration factor. The additive random Gaussian noise of the second column plots is obtained using a standard deviation of 0.01. The worst-case additive noise of the third column is obtained using the PGD method with $\epsilon = 0.02$.

We use a batch size of 2 and 60 training epochs. The experiments are run using two A5000 GPUs. The ADAM optimizer (Kingma and Ba, 2014) is utilized for training the network weights with momentum parameters of $(0.5, 0.999)$ and learning rate of 10^{-4}. The stability parameter λ_ℓ in (6.8) (and (6.12)) is tuned so that the standard accuracy of the learned model is comparable to vanilla MoDL. For RS-E2E, we set the standard deviation of Gaussian noise to $\sigma = 0.01$, and use 10 Monte Carlo samplings to implement the smoothing operation. Note that in our experiments, Gaussian noise and corruptions are added to real and imaginary parts of the data with the indicated σ. For AT, we implemented a 30-step PGD procedure within its minimax formulation with $\epsilon = 0.02$. For Score MRI, we used 150 steps for the reverse diffusion process with the pre-trained model. We fine-tuned a pre-trained Deep-Eq model with the same data as the proposed schemes. Unless specified, training parameters were similar across the compared methods.

6.4.1.4 Testing

We evaluate our methods on clean data (without additional perturbations), data with randomly injected noise, and data contaminated with worst-case additive perturbations. The worst-case disturbances allow us to see worst-case method sensitivity and are generated by the ℓ_∞-norm based PGD scheme with 10 steps (Antun et al., 2020a) corresponding to $\|\delta\|_\infty \leq \epsilon$, where ϵ is set

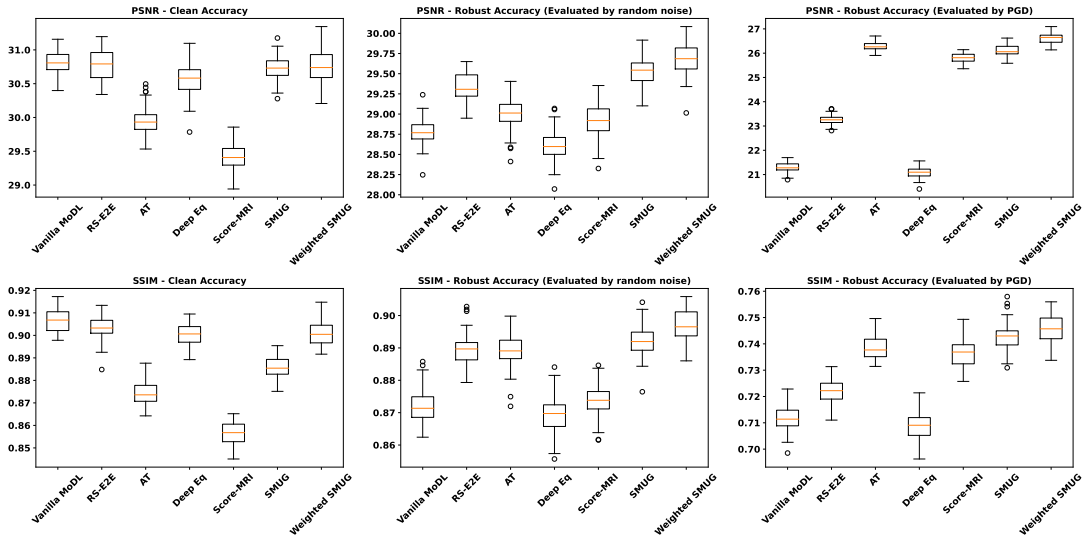


Figure 6.8 Reconstruction accuracy box plots for the fastMRI **knee** dataset with 8x Acceleration factor. The additive random Gaussian noise in the second column plots is obtained using a standard deviation of 0.01. The worst-case additive noise in the third column is obtained using the PGD method with $\epsilon = 0.02$.

nominally as the maximum underlying k-space real and imaginary part magnitude scaled by 0.05. We will indicate the scaling for ϵ (e.g., 0.05) in the results and plots that follow. The quality of reconstructed images is measured using peak signal-to-noise ratio (PSNR) and structure similarity index measure (SSIM)(Wang et al., 2004). In addition to the worst-case perturbations and random noise, we evaluate the performance of our methods in the presence of additional instability sources such as (i) different undersampling rates, and (ii) different numbers of unrolling steps.

6.4.2 Robustness with Additive Perturbations

In this subsection, we present the robustness results of the proposed approaches w.r.t. additive noise. In particular, the evaluation is conducted on the clean, noisy (with added Gaussian noise), and worst-case perturbed (using PGD for each method) measurements. **Fig. 6.5** presents testing set PSNR and SSIM values as box plots for different smoothing architectures, along with vanilla MoDL and the other baselines using the brain dataset. The clean accuracies of Weighted SMUG and SMUG are similar to vanilla MoDL indicating a good clean accuracy vs. robustness trade-off. As indicated by the PSNR and SSIM values, we observe that weighted SMUG, on average, outperforms all other baselines in robust accuracy (the second and third set of box plots of the two rows in **Fig. 6.5**). This observation is consistent with the visualization of reconstructed images for the brain

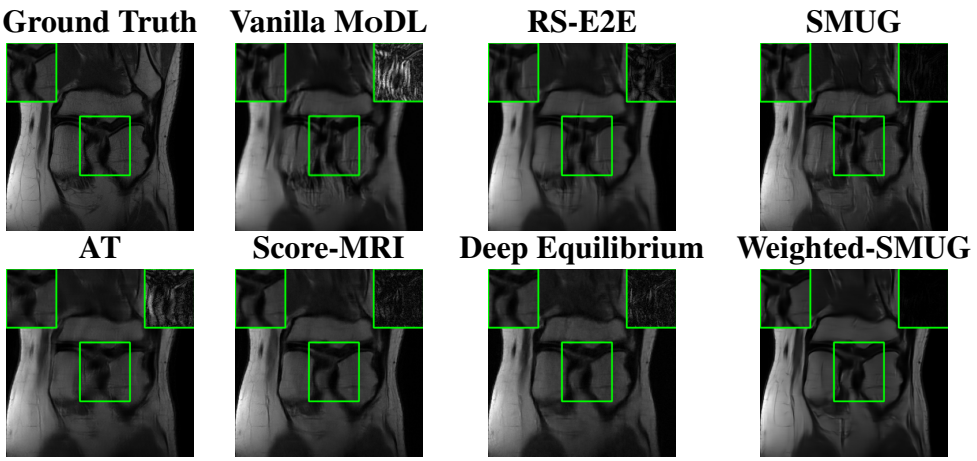


Figure 6.9 Visualization of ground-truth and reconstructed images using different methods for 4x k-space undersampling, evaluated on PGD-generated worst-case inputs of perturbation strength $\epsilon = 0.02$.

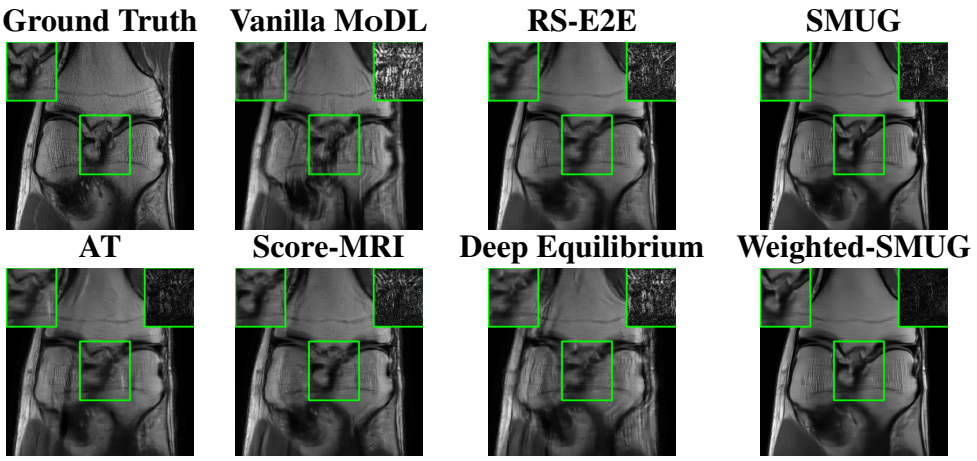


Figure 6.10 Visualization of ground truth and reconstructed images using different methods for 8x k-space undersampling, evaluated on PGD-generated worst-case inputs of perturbation scaling $\epsilon = 0.02$.

dataset in **Fig. 6.6**. We note that weighted SMUG requires longer time for training, which represents a trade-off. When comparing to AT, we observe that AT is comparable to SMUG in the case of robust (or worst-case noise) accuracy. However, the drop in clean accuracy (without perturbations) for AT is significantly larger than for SMUG. Furthermore, AT takes a much longer training time as it requires to solve an optimization problem (PGD) for every training data sample at every iteration to obtain the worst-case perturbations. Furthermore, we observe that its effectiveness is degraded for other perturbations including random noise as well as modified sampling rates shown in the next subsection. Importantly, the proposed SMUG and Weighted SMUG are not trained to be robust to

any specific perturbations or instabilities, but are nevertheless effective for several scenarios.

In comparison to the diffusion based Score-MRI, the proposed methods perform better in terms of both clean accuracy and random noise accuracy. Although for worst-case perturbations, the PSNR values of Score-MRI are only slightly worse than SMUG, it is important to note that not only the training of diffusion-based models takes longer than our method, but also the inference time is longer as Score-MRI requires to perform nearly 150 sampling steps to process one scan and takes nearly 5 minutes with a single RTX5000 GPU, whereas our method takes only about 25 seconds per scan. The SMUG schemes also substantially outperform the deep equilibrium model in the presence of perturbations.

In **Fig 6.7** and **Fig 6.8**, we report PSNR and SSIM results of different methods at two sampling acceleration factors for the knee dataset. Therein, we observe quite similar outcomes to those reported in **Fig 6.5**.

Figs. 6.9 and **6.10** show reconstructed images by different methods for knee scans at 4x and 8x undersampling, respectively. We observe that SMUG and Weighted SMUG show fewer artifacts, sharper features, and fewer errors when compared to Vanilla MoDL and other baselines in the presence of the worst-case perturbations.

Fig. 6.11 presents average PSNR results over the test dataset for the considered models under different levels of worst-case perturbations (*i.e.*, attack strength ϵ). We used the knee dataset for this experiment. We observe that SMUG and weighted SMUG outperform RS-E2E, vanilla MoDL, and Deep-Eq across all perturbation strengths. When compared to Score-MRI and AT, our proposed methods consistently maintain higher PSNR values for moderate to large perturbations (less than $\epsilon = 0.08$). For instance, when $\epsilon = 0.02$, weighted SMUG reports more than 1 dB improvement over AT and Score-MRI.

6.4.3 Robustness for Varying Sampling Rates and Unrolling Steps

In this subsection, we evaluate the robustness of our proposed approaches and the considered baselines at varying sampling rates and unrolling steps.

For our first experiment, during training, a k-space undersampling or acceleration factor of 4x

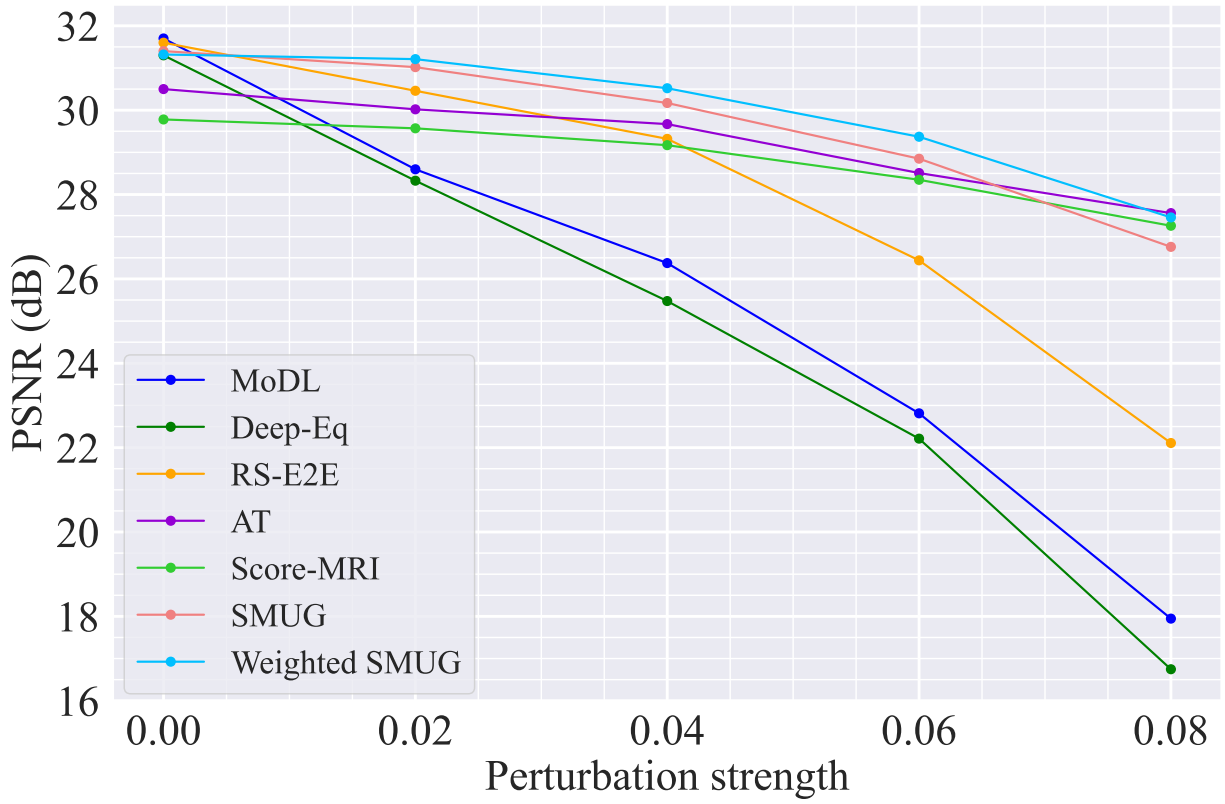


Figure 6.11 PSNR of baseline methods and the proposed method versus perturbation strength (i.e., scaling) ϵ used in PGD-generated worst-case examples at testing time with 4x k-space undersampling. $\epsilon = 0$ corresponds to clean accuracy.

is used for our methods and the considered baselines. At testing time, we evaluate performance (in terms of PSNR) with acceleration factors ranging from 2x to 8x. The results are presented in **Fig. 6.12**. It is clear that when the acceleration factor during testing matches that of the training phase (4x), all methods achieve their highest PSNR results. Conversely, performance generally declines when the acceleration factors differ. For acceleration factors 3x to 8x (ignoring 4x where models were trained), we observe that our methods outperform all the considered baselines. For the 2x case, our methods report higher PSNR values compared to RS-E2E, vanilla MoDL, and Deep-Eq and slightly underperform AT, while Score-MRI shows more resilience at 2x.

For the second experiment, we study the performance of varying unrolling steps. More specifically, during training, we utilize 8 unrolling steps to train our methods and the baselines. At testing time, we report the results of utilizing 1 to 16 unrolling steps. The PSNR results of all

the considered cases are given in **Fig. 6.13**. The results show that both SMUG and Weighted SMUG maintain performance comparable to the Deep Equilibrium model. Furthermore, when using different unrolling steps and faced with additive measurement perturbations, the SMUG methods' PSNR values are stable and close to the unperturbed case (indicating robustness), whereas the other methods see more drastic drop in performance. This behavior for SMUG also agrees with the theoretical bounds in Section 6.3.

Although we do not intentionally design our method to mitigate MoDL's instabilities against different sampling rates and unrolling steps, the SMUG approaches nevertheless provide improved PSNRs over other baselines. This indicates broader value for the robustification strategies incorporated in our schemes.

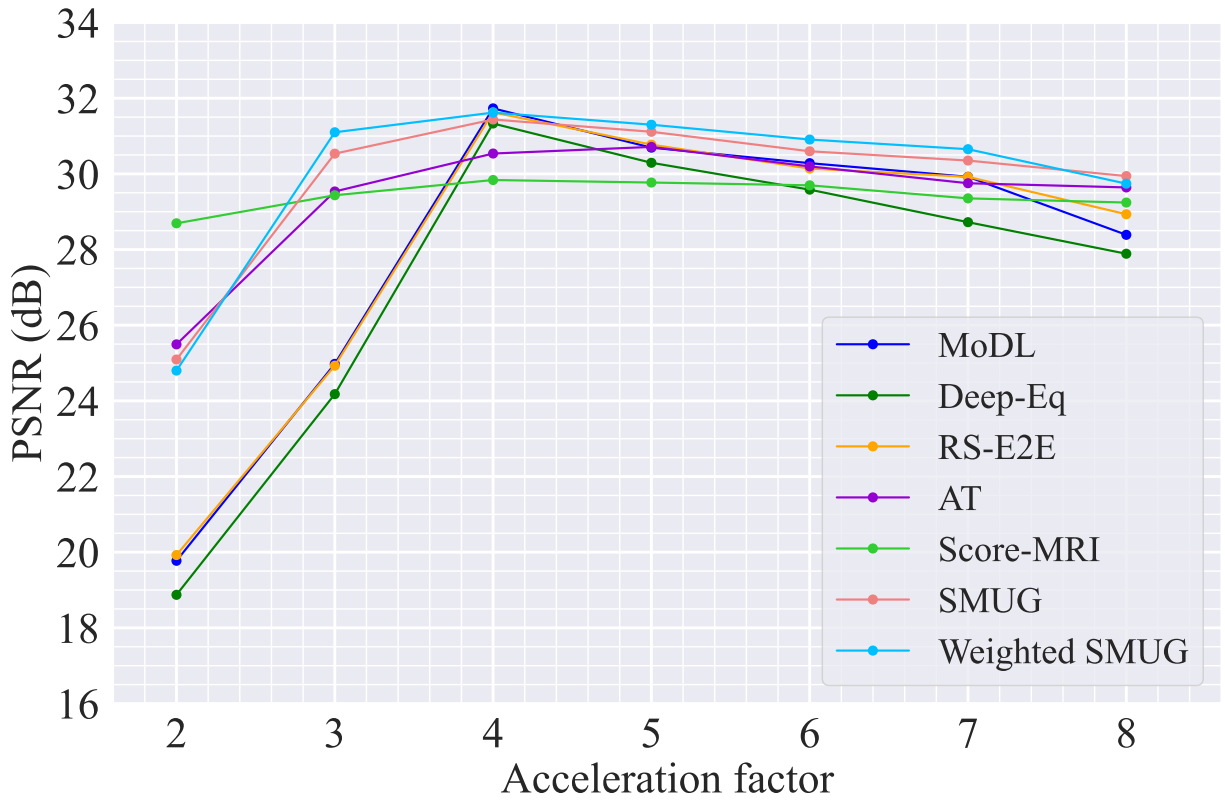


Figure 6.12 PSNR results for different MRI reconstruction methods versus different measurement sampling rates (models trained at 4× acceleration).

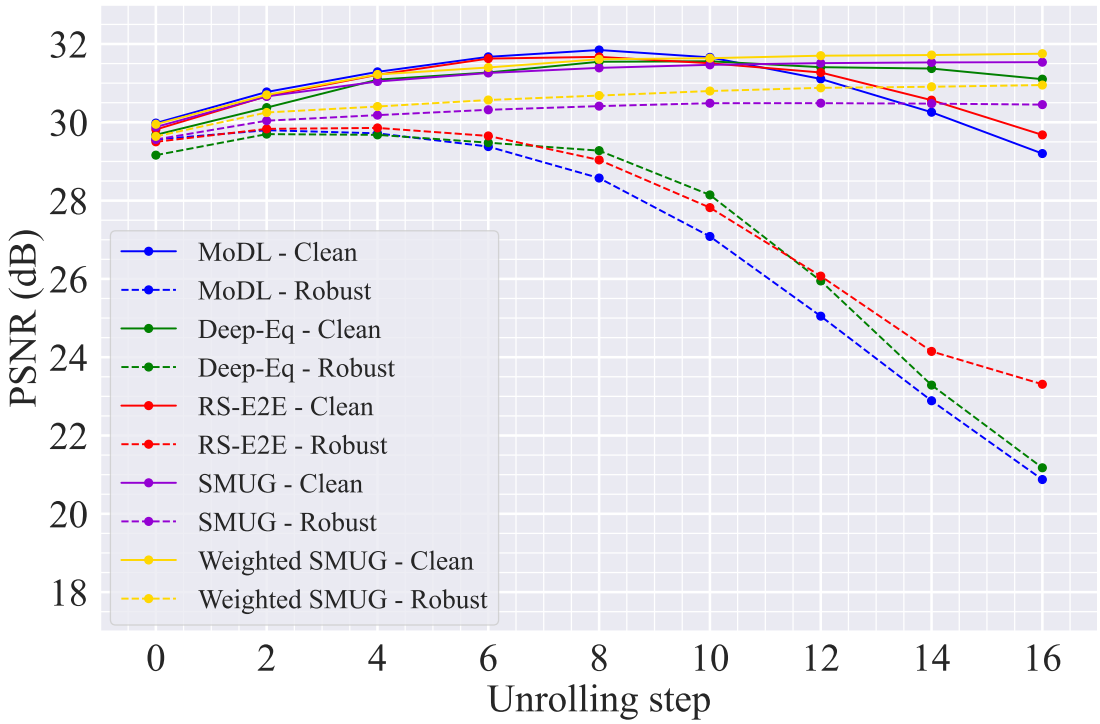


Figure 6.13 PSNR results for different MRI reconstruction methods at 4x k-space undersampling versus number of unrolling steps (8 steps used in training). “Clean" and "Robust" denote the cases without and with added worst-case (for each method) measurement perturbations.

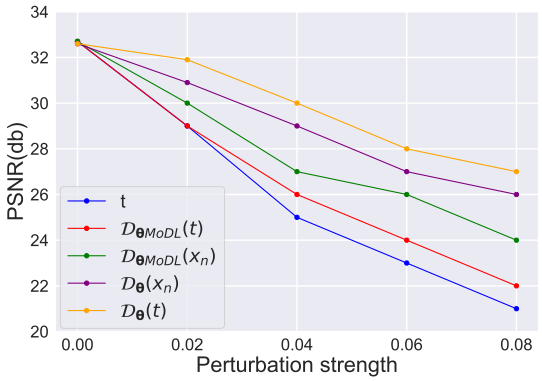


Figure 6.14 PSNR vs. worst-case perturbation strength (ϵ) for SMUG for different configurations of UStab loss (6.7).

6.4.4 Importance of the Ustab Loss

We conduct additional studies on the unrolled stability loss in our scheme to show the importance of integrating target image denoising into SMUG’s training pipeline in (6.7). **Fig. 6.14** presents PSNR values versus perturbation strength/scaling (ϵ) when using different alternatives to $\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{t})$ in (6.7), including $\mathbf{t}$ (the original target image), $\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_n)$ (denoised output of each unrolling step),

and variants when using the fixed, vanilla MoDL's denoiser $\mathcal{D}_{\boldsymbol{\theta}_{\text{MoDL}}}$ instead. As we can see, the performance of SMUG varies when the UStab loss (6.7) is configured differently. The proposed $\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{t})$ outperforms other baselines. A possible reason is that it infuses supervision of target images in an adaptive, denoising-friendly manner, *i.e.*, taking the influence of $\mathcal{D}_{\boldsymbol{\theta}}$ into consideration.

6.4.5 Impact of the Noise Smoothing

To comprehensively assess the influence of the introduced noise during smoothing, denoted as $\boldsymbol{\eta}$, on the efficacy of the suggested approaches, we undertake an experiment involving varying noise standard deviations σ. The outcomes, documented in terms of RMSE, are showcased in **Fig.**6.15. The accuracy (reconstruction quality w.r.t. ground truth) and robustness error (error between with and without measurement perturbation cases) are shown for both SMUG and RS-E2E. We notice a notable trend: as the noise level σ increases, the accuracy for both methods improves before beginning to degrade. Importantly, SMUG consistently outperforms end-to-end smoothing. Furthermore, the robustness error continually drops as σ increases (corroborating with our analysis/bound in Section 6.3), with more rapid decrease for SMUG.

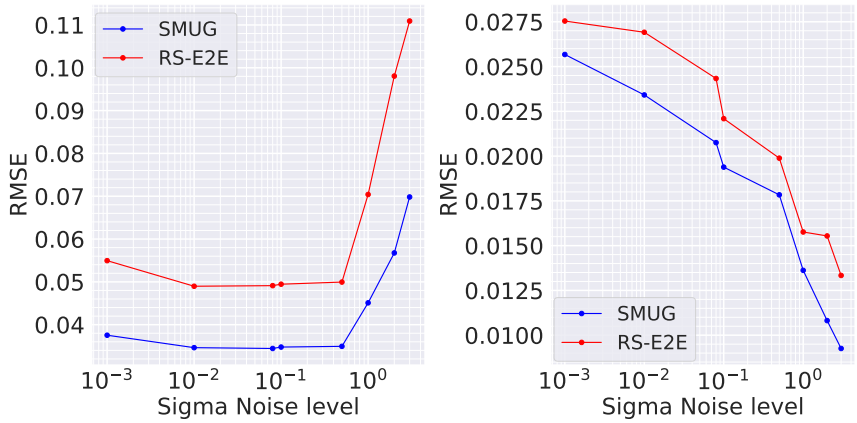


Figure 6.15 Left: Norm of difference between SMUG and RS-E2E reconstructions and the ground truth for different choices of σ in the smoothing process. A worst-case PGD perturbation $\boldsymbol{\delta}$ computed at $\epsilon = 0.01$ was added to the measurements in all cases. Right: Robustness error for SMUG and RS-E2E at various σ, i.e., norm of difference between output with the perturbation $\boldsymbol{\delta}$ and without it.

6.4.6 Empirical Analysis of the behavior of Weighted SMUG

In subsequent final study, we analyze the behavior of the Weighted SMUG algorithm. We delve into the nuances of weighted smoothing, which can assign different weights to different images

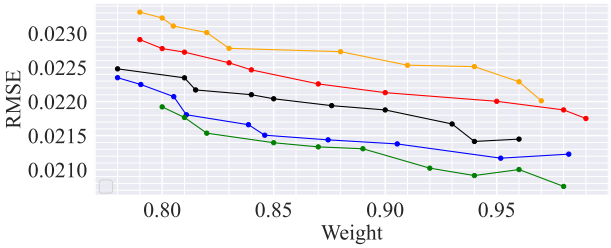


Figure 6.16 Weights predicted by the weight encoder network in Weighted SMUG (from final layer of unrolling) plotted against root mean squared error (RMSE) of the corresponding denoised images for 5 randomly selected scans (with 4x undersampling).

during the smoothing process. The aim is to gauge how the superior performance of Weighted SMUG arises from the variations in learned weights. Our findings indicate that among the 10 Monte Carlo samplings implemented for the smoothing operation, those with lower denoising RMSE when compared to the ground truth images generally receive higher weights, as illustrated in **Fig. 6.16**.

6.4.7 Results of Applying Our Methods to ISTA-NET

In our concluding study, we investigate whether our robustification methods can be effective with an alternative unrolling technique, ISTA-Net (Zhang and Ghanem, 2018). For ISTA-Net, we adopted the default architecture, utilizing the ADAM optimizer with a learning rate of 10^{-4}. The network was configured with 9 phases (unrolling iterations) and trained on the fastMRI knee dataset comprising 3000 scans at 4x undersampling and with 100 epochs for training. Similar to previous experiments, we used 64 scans for testing. Other settings for training the vanilla ISTA-Net were set to default values. Other settings for the RS-E2E version, and the SMUG and Weighted SMUG versions of ISTA-Net were similar to the MoDL case. The results, as presented in Figure 6.17, demonstrate that the clean accuracy performance of SMUG and weighted SMUG versions of ISTA-Net are comparable to vanilla ISTA-Net. Notably, under conditions of random noise (Gaussian noise with $\sigma = 0.01$ added) and PGD attack (30 steps with $\epsilon = 0.02$) perturbed measurements, our method surpasses both the original ISTA-Net and the RS-E2E version. The comparative results reveal that the performance closely aligns with the outcomes previously observed when unrolling smoothing was combined with the MoDL network.

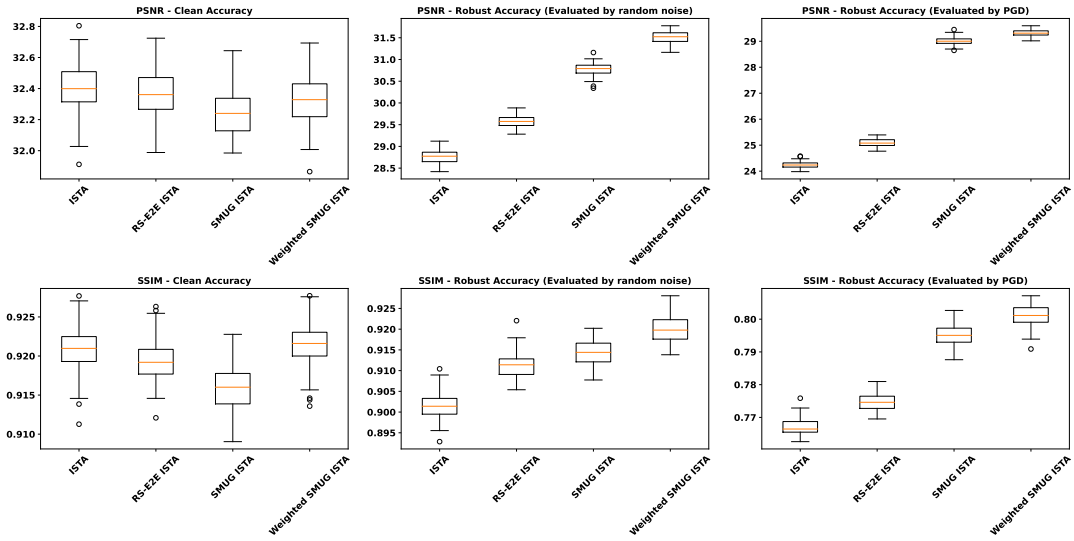


Figure 6.17 Reconstruction accuracy box plots for the fastMRI **knee** dataset with 4x acceleration factor for the case of **ISTA-Net**. The additive random Gaussian noise in the second column plots is obtained using a standard deviation of 0.01. The worst-case additive noise in the third column is obtained using the PGD method with $\epsilon = 0.02$.

6.5 Discussion and Conclusion

In this work, we proposed a scheme for improving the robustness of DL-based MRI reconstruction. In particular, we investigated deep unrolled reconstruction's weaknesses in robustness against worst-case or noise-like additive perturbations, sampling rates, and unrolling steps. To improve the robustness of the unrolled scheme, we proposed SMUG with a novel unrolled smoothing loss. We also provided a theoretical analysis on the robustness achieved by our proposed method. Compared to the vanilla MoDL approach and other schemes, we empirically showed that our approach is effective and can significantly improve the robustness of a deep unrolled scheme against a diverse set of external perturbations. We also further improved SMUG's robustness by introducing weighted smoothing as an alternative to conventional RS, which adaptively weights different images when aggregating them. In future work, we hope to apply the proposed schemes to other imaging modalities and evaluate robustness against additional types of realistic perturbations. While we theoretically characterized the robustness error for SMUG, we hope to further analyze its accuracy-robustness trade-off with perturbations.

CHAPTER 7

MRI RECONSTRUCTION VIA DIFFUSION PURIFICATION

7.1 Introduction

In the last chapter, we introduced several methods to improve model generalization and robustness. However, a primary weakness of SMUG is that it can be difficult to integrate into models other than the unrolling model. Furthermore, SMUG does not demonstrate significantly better performance than adversarial training or other existing methods. To address these shortcomings, we now aim to refine and enhance this approach. Insipried by a recent study conducted by Nie et al. (Nie et al., 2022b) has introduced a robustification strategy that effectively mitigates the impact of additive worst-case perturbations, harnessing the power of diffusion models (DMs) (Chung and Ye, 2022; Chung et al., 2023c; Karras et al., 2022). Drawing inspiration from this methodology and benefiting from the generalization capabilities of DMs, we investigate the application of a similar approach to enhance the resilience of DL-based MRI reconstruction. Our approach centers on the application of pre-trained diffusion models as noise purifiers. More precisely, this purification process entails a gradual introduction of noise, followed by the refinement of the noise through the utilization of the pre-trained DM

7.1.1 Contributions

- We introduce a general robustification framework designed to enhance the resilience of DL-based MRI reconstructors against a variety of instabilities, and improve their generalization performance when faced with out of distribution samples. This is accomplished through integrating purification via pre-trained DMs into existing DL-based models.
- We prove that the perturbed and clean images' distributions (and conditional distributions) get closer to each other as the time increases in the forward diffusion stage.
- We present a novel approach to select a process-switching time step - a critical parameter within our DM-based purification method. This eliminates the necessity of treating it as a hyper-parameter.
- We use fine-tuning to further improve the DL-based reconstructors' performance, which, unlike well-known state-of-the-art (SOTA) robustification method AT, neither requires solving a minimax

problem nor involves generating worst-case examples.

- In our experimental results, we demonstrate the effectiveness of our proposed approach by assessing it against standard evaluation metrics, surpassing the performance of AT, RS, and diffusion-based MRI reconstruction in the presence of several sources of instabilities. Furthermore, we illustrate that after being trained on the knee fastMRI dataset, the purification process using DMs extends its benefits to other MRI datasets, including a brain MRI dataset or data with unseen lesions. Additionally, we show that our robustification approach can be applied to multiple DL-based supervised methods such as the well-known MoDL and the recent Recurrent Variational Network (RecurrentVarNet) (Yiasemis et al., 2022).

7.2 Lack of Robustness in DL-based MRI Reconstruction & Score-based DMs

In this section, we first introduce the inverse problem formulation for Deep MRI reconstruction. Second, we shed light on the lack of robustness in these models. Then, we present the formulation of the score-based DM used in this paper.

7.2.1 DL-based MRI Reconstruction

MRI reconstruction is a challenging ill-posed inverse problem (Donoho, 2006a). Its objective is to recover the original signal $\mathbf{x} \in \mathbb{C}^n$ from observed measurements $\mathbf{y} \in \mathbb{C}^m$, with $m < n$. For multi-coil MRI, this task can be formulated as a linear inverse problem denoted as $\mathbf{y} \approx \mathbf{A}\mathbf{x}$, where $\mathbf{A} = \mathbf{MFS}$ with $\mathbf{S}$ denoting the sensitivity encoding with multiple coils, $\mathbf{F}$ denoting coil-by-coil Fourier transform, and $\mathbf{M}$ denoting coil-wise undersampling.

Typically, the reconstruction process involves solving the optimization problem $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \mathcal{R}(\mathbf{x})$, where $\mathcal{R}(\cdot)$ (resp. $\lambda > 0$) is a regularization term (resp. parameter).

There are several methods that use unrolling steps to train Deep MRI image reconstruction. While for the major part of this paper we focus on the popular MoDL framework (Aggarwal et al., 2018), our proposed method can be applied to other DL-based reconstruction models, as illustrated in the last subsection of the experimental results. In MoDL, the traditional regularization term is substituted with a denoising Neural Network (NN) represented as $f : \mathbb{C}^n \to \mathbb{C}^n$, parameterized by θ.

This denoising NN is trained in a supervised learning framework using a dataset of multiple pairs of measurements $\mathbf{y}$ and their corresponding ground truth images $\mathbf{x}$.

For each pair $(\mathbf{y}, \mathbf{x})$ in the training set D, the MoDL training process initializes $\mathbf{x}_0$ (e.g., as $\mathbf{A}^H\mathbf{y}$) and then iterates through the subsequent steps for a specified number of unrolling iterations indexed by $j \in \{0, \ldots, N-1\}$. This process can be described as follows:

$$\mathbf{z}_j \leftarrow f_\theta(\mathbf{x}_j)\,, \tag{7.1}$$

$$\mathbf{x}_{j+1} \leftarrow \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \lambda\|\mathbf{x} - \mathbf{z}_j\|^2\,. \tag{7.2}$$

The parameters of f_θ are updated end-to-end in a supervised manner following (Aggarwal et al., 2018).

Equation (7.1) corresponds to the denoising step, while Equation (7.2) pertains to the data consistency (DC). Equation (7.2) has a closed-form solution given by $\mathbf{x}_{j+1} \leftarrow (\mathbf{A}^H\mathbf{A} + \lambda\mathbf{I})^{-1}(\mathbf{A}^H\mathbf{y} + \lambda\mathbf{z}_j)$.

During the testing phase, when presented with an aliased image (e.g., $\mathbf{A}^H\mathbf{y}$), a trained MoDL model reconstructs $\mathbf{x}$ by applying the procedure described in Equations (7.1) and (7.2) for a specified number of unrolling steps. For the remainder of this paper, we use $\text{MoDL}_\theta(\mathbf{A}^H\mathbf{y})$ to denote the image reconstructed from MoDL.

7.2.2 Vulnerabilities & Challenges of DL-based MRI Reconstructors

7.2.2.1 K-space Additive Noise

Given a trained deep MRI reconstruction NN and an aliased image $\mathbf{z} = \mathbf{A}^H\mathbf{y}$, recent studies have shown that these NNs are not robust to additive perturbations $\boldsymbol{\delta}$ to $\mathbf{y}$ (Li et al., 2023). The study in (Jia et al., 2022b) presents an approach to generate worst-case additive noise that employs norm constraints, in line with the attack strategies utilized in image classification. This approach aims to produce a form of worst-case imperceptible additive noise against a reconstructor in the image domain. Given a perturbation budget $\epsilon > 0$, the worst-case additive perturbations can be obtained using the following optimization problem.

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \mathcal{L}\Big(\text{MoDL}_\theta(\mathbf{A}^H\mathbf{y}), \text{MoDL}_\theta(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}))\Big)\,, \tag{7.3}$$

where $\|.\|_\infty$ is the ℓ_∞ norm and $\mathcal{L}$ is a differentiable loss function that computes the reconstruction loss. Given the original image $\mathbf{x}^*$, generating the perturbations can also be achieved by replacing the first argument of $\mathcal{L}$ in (7.3) with $\mathbf{x}^*$. A solution of (7.3) can be obtained using the Projected Gradient Descent (PGD) method (Madry et al., 2017). In this paper, we also use $\mathbf{z}_{\text{pert}} = \mathbf{A}^H(\mathbf{y} + \boldsymbol{\delta}) = \mathbf{A}^H\mathbf{y}_{\text{pert}}$ which relates perturbations in k-space and image space.

In addition to the worst-case perturbations, random/realistic additive measurement noise could also impact the performance of a reconstructor.

7.2.2.2 Training/Testing Sampling Protocol & Undersampling Rate Disparities

In addition to additive perturbations, the study presented in (Li et al., 2023) underscores an additional potential source of instability that MoDL (and other DL-based reconstructors) may face during testing. This source stems from changes in the measurement sampling rate, leading to perturbations in the sparsity of the sampling mask within $\mathbf{A}$ (Antun et al., 2020a). Furthermore, in this paper, we consider another variation that these NNs could encounter during the testing phase, involving a shift or variation in the k-space sampling locations within the matrix $\mathbf{M}$, resulting in the construction of a nonidentical forward operator for testing. For this case, $\mathbf{z}_{\text{pert}} = \mathbf{A}_{\text{test}}^H\mathbf{y}$, where $\mathbf{A}_{\text{test}} \neq \mathbf{A}$.

We remark that ensuring the robustness of a reconstruction model to variations in the sampling protocol, undersampling rate, scan contrast, etc., is crucial as it mitigates the need for re-training to all possible practical scenarios and variations, common in imaging. Re-training models for new setups is expensive. Moreover, the relatively limited training data availability (which requires fully-sampled measurements as labels in supervised learning) in reconstruction applications also warrant learning models that can still be significantly robust.

7.2.2.3 Unseen Anatomies & Pathologies at Testing Time

A lesion (or anatomy changes) denotes an anomaly, or impairment within a tissue or organ of the body, arising from diverse factors such as injuries, diseases, or pathological conditions. In the medical domain, the term commonly characterizes regions of abnormal or diseased tissue,

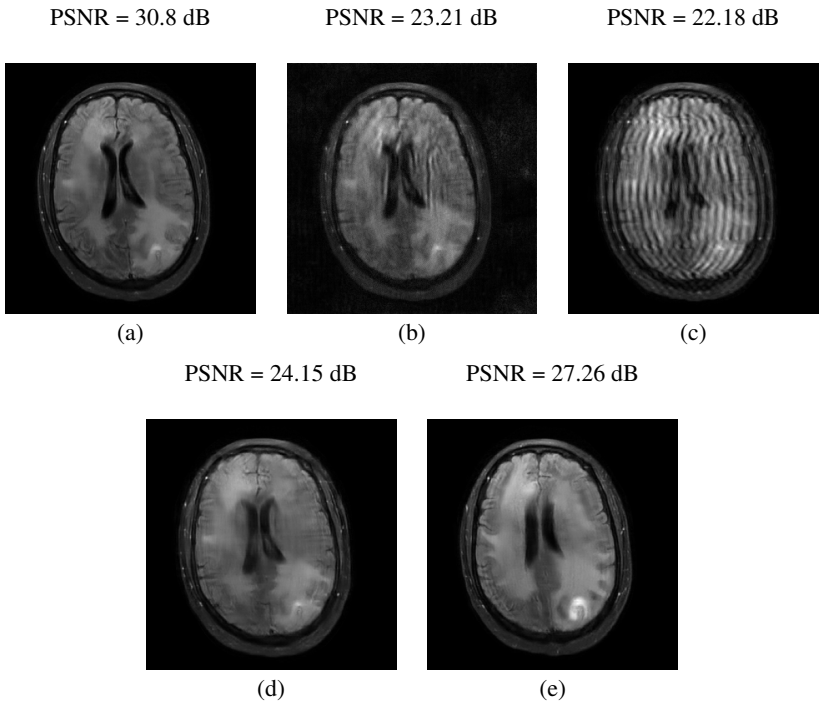


Figure 7.1 Here, we show the vulnerabilities and generalization challenges of DL-based MRI reconstruction models by evaluating a trained MoDL reconstructor (trained at 4x undersampling) with the considered cases in Section 7.2.2. (a) Reconstructed image from clean measurements. (b) Reconstructed image from measurements with worst-case additive perturbations (Equation (7.3) with $\epsilon = 0.02$). (c) Reconstructed image from measurements with 2x undersampling rate during testing. (d) Reconstructed image from a different test time sampling mask with 4x undersampling. (e) Reconstructed image from measurements with an unseen lesion during testing.

observed through MR imaging. In this paper, we study the practical case where the DL-based image reconstructor is trained on some data points, but tested with measurements with unseen lesions.

Figure 7.1 illustrates reconstructed images from the instabilities and the generalization challenges considered in this paper.

7.2.3 Score-based Diffusion Models

Diffusion models (DMs) have shown great potential for solving many hard computer vision tasks and recently extended to medical imaging applications.

The Bayesian framework of DMs, introduced in (Ho et al., 2020; Sohl-Dickstein et al., 2015), consists of a discrete Markov Chain. The forward direction is constructed by sampling from $p(\mathbf{z}_i \mid \mathbf{z}_{i-1}) = \mathcal{N}(\mathbf{z}_i\,;\, \sqrt{1-\beta_i}\mathbf{z}_{i-1}, \beta_i \mathbf{I})$, where $\beta_i \in (0,1)$ is an entry of a sequence of monotonically increasing positive noise scales w.r.t. i.

Score-based DM was introduced in (Song et al., 2021c) and was shown to be equivalent to the Bayesian framework. Score-based DMs can be formulated by the following forward and reverse Stochastic Differential Equations (SDEs).

$$d\mathbf{z} = \mathbf{f}(\mathbf{z}, t)dt + g(t)d\mathbf{w}\,, \tag{7.4}$$

$$d\mathbf{z} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{z}}\log p_t(\mathbf{z})\right]dt + g(t)d\bar{\mathbf{w}}\,, \tag{7.5}$$

where $\mathbf{f}$ and g are the drift and diffusion coefficients, respectively. t spans the interval $[0, 1]$ and represents the time index. $d\mathbf{w}$ and $d\bar{\mathbf{w}}$ represent standard Brownian motion evolving forward and backward in time, respectively. The term $p_t(\mathbf{z})$ denotes the distribution of $\mathbf{z}$ at time t, while $\nabla_{\mathbf{z}}\log p_t(\mathbf{z})$ represents the score function. By employing the formulation of the Variance Exploding (VE) SDE (VE-SDE) (Song et al., 2021c), for which $\mathbf{f} = \mathbf{0}$ and $g(t) = \sqrt{d\sigma^2(t)/dt}$, we can re-write the forward and reverse SDEs as

$$d\mathbf{z} = \sqrt{\frac{d\sigma^2(t)}{dt}}d\mathbf{w}\,, \tag{7.6}$$

$$d\mathbf{z} = -\frac{d\sigma^2(t)}{dt}\nabla_{\mathbf{z}}\log p_t(\mathbf{z})dt + \sqrt{\frac{d\sigma^2(t)}{dt}}d\bar{\mathbf{w}}\,. \tag{7.7}$$

In Equations (7.6) and (7.7), function $\sigma(t) = \sigma_l(\sigma_u/\sigma_l)^t$ is a monotonically increasing function w.r.t. t, where $\sigma_l \in (0, 1)$ and $\sigma_u > 1$ are constants.

The score function is in practice replaced by a neural network denoted as $s : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$, parameterized by ϕ, which is trained using the denoising score matching technique (Chung and Ye, 2022) as

$$\min_{\phi} \mathbb{E}\left[\left\|\sigma(t)s_\phi(\mathbf{z}(t), t) - \frac{\mathbf{z}(t) - \mathbf{z}}{\sigma(t)}\right\|^2\right]. \tag{7.8}$$

The expectation in (7.8) is taken over $t \sim U[0, 1]$, $\mathbf{z} \sim p(\mathbf{z})$, and $\mathbf{z}(t) \sim \mathcal{N}(\mathbf{z}, \sigma(t)\mathbf{I})$, where $p(\mathbf{z}) = p_0(\mathbf{z})$ is the distribution of the training data.

Having obtained a trained DM with parameters ϕ, the task of sampling $\hat{\mathbf{z}}(0)$ at the time instant $t = 0$ is realized through the solution of the reverse process SDE in (7.7). In this step, the score function is substituted with the learned function s_ϕ. There exist various techniques for sampling

Algorithm 7.1 Predictor-Corrector Sampling with DC (Chung and Ye, 2022)

Input: Image $\mathbf{z} = \mathbf{A}^H\mathbf{y}$, trained DM s_ϕ, discretized time step N_r, and noise schedule ϵ_i.
Function: $\hat{\mathbf{z}} = \text{PCDC}\left(s_\phi(\mathbf{z}(N_r), N_r), \mathbf{y}, \mathbf{A}, N_r, 0\right)$.

Initialize $\mathbf{z}(N_r) \sim \mathcal{N}(0, \sigma^2(N_r)\mathbf{I})$.
For $i \in \{N_r - 1, \ldots, 0\}$ \\Prediction
  $\mathbf{z}'(i) \leftarrow \mathbf{z}(i+1) + (\sigma^2(i+1) - \sigma^2(i))s_\phi(\mathbf{z}(i+1), i+1)$
  $\mathbf{z}(i) \leftarrow \mathbf{z}'(i) + \sqrt{\sigma^2(i+1) - \sigma^2(i)}\eta,\ \eta \sim \mathcal{N}(0, \mathbf{I})$
  $\mathbf{z}(i) \leftarrow \mathbf{z}(i) + \mathbf{A}^H(\mathbf{y} - \mathbf{A}\mathbf{z}(i))$ \\Data Consistency
  For M_r steps **do** \\Correction
    $\mathbf{z}'(i) \leftarrow \mathbf{z}(i) + \epsilon_i s_\phi(\mathbf{z}(i), i)$
    $\mathbf{z}'(i) \leftarrow \mathbf{z}'(i) + \sqrt{2\epsilon_i}\,\eta,\ \eta \sim \mathcal{N}(0, \mathbf{I})$
    $\mathbf{z}(i) \leftarrow \mathbf{z}'(i) + \mathbf{A}^H(\mathbf{y} - \mathbf{A}\mathbf{z}(i))$ \\Data Consistency
$\hat{\mathbf{z}} = \mathbf{z}(0)$

from DMs, which involve solving the reverse SDE in (7.7). In this paper, the Euler method (Platen and Bruti-Liberati, 2010) and the Predictor-Corrector (PC) scheme (Allgower and Georg, 2012) are used. Following the work in (Chung and Ye, 2022), a data consistency step is considered to allow sampling from the conditional distribution $p(\mathbf{z}|\mathbf{y})$, In practice, the continuous time index $t \in [0, 1]$ is discretized into $i \in [N_r]$, where $[N_r] := \{1, \ldots, N_r\}$. The PC sampling technique consists of N_r prediction reverse steps. In each prediction iteration, M_r correction steps are required (Song et al., 2021c). The full procedure is outlined in Algorithm 7.1.

7.3 Diffusion Purification for Robust DL-based MRI Reconstruction

In this section, we begin by outlining the key components of the proposed Diffusion Purification (DP) pipeline. Subsequently, we introduce our approach for obtaining the PST step. Following that, we elaborate on our fine-tuning strategy for MoDL, leveraging the purified samples.

7.3.1 DM-based Purification

Here, we present our DP approach, which consists of the following two stages.

Diffusion Stage: Given measurements $\mathbf{y}$, let $\mathbf{z}_{\text{pert}}$ denote the perturbed version of $\mathbf{z} = \mathbf{A}^H\mathbf{y}$. As illustrated in the previous section, this perturbed version can be due to various reasons such as random measurement noise, not well-modeled noise and artifacts (e.g., it may make sense to consider worst-case additive noise (from (7.3))), and different k-space undersampling factors or sampling patterns/masks at testing time.

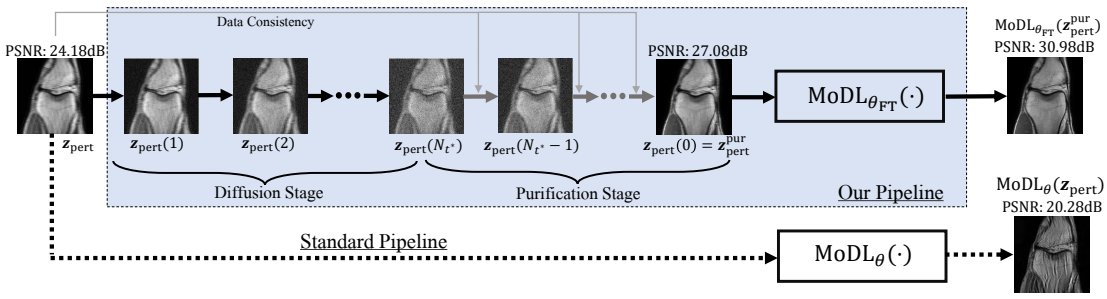


Figure 7.2 A schematic block diagram illustrating the standard pipeline and our proposed reconstruction pipeline. The functions $\mathrm{MoDL}_{\theta}(\cdot)$ and $\mathrm{MoDL}_{\theta_{\mathrm{FT}}}(\cdot)$ represent the application of the standard pre-trained MoDL procedure and our 'pre-trained+fine-tuned' robust MoDL procedure, respectively. Here, MoDL can be replaced with other DL-based reconstruction models.

The first stage of the DP approach involves diffusing $\mathbf{z}(0) = \mathbf{z}_{\mathrm{pert}}$ from $t = 0$ to $t = t^*$, where $t^* \in (0, 1)$ indicates the diffusion time index at which the forward process stops. We term t^* as the Process-Switching Time (PST) step. The PST step and $\sigma(\cdot)$ control the amount of noise added to $\mathbf{z}_{\mathrm{pert}}$. This stage corresponds to

$$\mathbf{z}_{\mathrm{pert}}(t^*) = \mathbf{z}_{\mathrm{pert}} + \sqrt{\sigma^2(t^*) - \sigma^2(0)}\eta_{t^*}\,, \eta_{t^*} \sim \mathcal{N}(0, \mathbf{I})\,. \tag{7.9}$$

Purification Stage: After obtaining the diffused perturbed image, denoted as $\mathbf{z}_{\mathrm{pert}}(t^*)$, the objective of the second step is to derive the purified sample, denoted as $\mathbf{z}^{\mathrm{pur}}_{\mathrm{pert}}$, from $\mathbf{z}_{\mathrm{pert}}(t^*)$. This is achieved by employing the PC reverse process with data consistency (DC). In other words, we use the PC with DC procedure in Algorithm 7.1 as:

$$\mathbf{z}^{\mathrm{pur}}_{\mathrm{pert}}(0) = \mathrm{PCDC}\left(s_\phi(\mathbf{z}_{\mathrm{pert}}(t^*), t^*), \mathbf{y}_{\mathrm{pert}}, \mathbf{A}, t^*, 0\right)\,. \tag{7.10}$$

In practice, we use N_{t^*}, which represents the discrete PST step. We remark that N_{t^*} is less than the total number of available steps in standard sampling reverse process N_r. Algorithm 7.2 illustrates the diffusion purification procedure.

Intuition: Starting with a perturbed image $\mathbf{z}_{\mathrm{pert}}$, which is assumed to be drawn from distribution $q(\mathbf{z})$, our approach initiates with $\mathbf{z}(0) = \mathbf{z}_{\mathrm{pert}}$ and gradually introduces noise. If the aliased image $\mathbf{z}$ follows a distribution $p(\mathbf{z})$, then as $t \to 1$, these two distributions will get closer. This signifies that the perturbations are progressively diminishing due to the incremental noise

Algorithm 7.2 Diffusion Purification

Input: Perturbed measurements $\mathbf{y}_{\text{pert}}$, operator $\mathbf{A}$, trained DM s_ϕ, and PST step N_{t^*}
Function: $\mathbf{z}_{\text{pert}}^{\text{pur}} = \text{DP}_\phi(\mathbf{y}_{\text{pert}}, \mathbf{A}, N_{t^*})$.

Initialize $\mathbf{z}(0) = \mathbf{z}_{\text{pert}}$
For $i \in \{1, \ldots, N_{t^*}\}$ \\Diffusion steps
  Obtain $\mathbf{z}(i) \leftarrow \mathbf{z}(i-1) + \sqrt{\sigma^2(i) - \sigma^2(i-1)}\,\eta, \eta \sim \mathcal{N}(0, \mathbf{I})$
For $i \in \{N_{t^*}, \ldots, 1\}$ \\Purification steps
  Obtain $\mathbf{z}(i-1) \leftarrow \text{PCDC}(s_\phi(\mathbf{z}(i), i), \mathbf{y}_{\text{pert}}, \mathbf{A}, i, i-1)$
Obtain $\mathbf{z}_{\text{pert}}^{\text{pur}} = \mathbf{z}(0)$.

incorporated during the forward process of (7.9). To emphasize this point, we present the following Theorem, whose proof is deferred to the Appendix.

Theorem 7.3.1. Let $p_t(\mathbf{z})$ and $p_{0t}(\mathbf{z}(t) \mid \mathbf{z})$ be the distribution and the conditional distribution of $\mathbf{z}(t)$ given the VE-SDE forward process of (7.6) starts at the unperturbed image $\mathbf{z}$. Similarly, let $q_t(\mathbf{z})$ and $q_{0t}(\mathbf{z}(t) \mid \mathbf{z}_{\text{pert}})$ be the distribution and the conditional distribution of $\mathbf{z}(t)$ given the VE-SDE forward process of (7.6) starts at the perturbed image $\mathbf{z}_{\text{pert}} = \mathbf{A}^H\mathbf{y}_{\text{pert}} = \mathbf{A}^H(\mathbf{y} + \boldsymbol{\delta})$. Then, as t moves forward from $t = 0$ to $t = 1$:

1. The KL divergence between p_{0t} and q_{0t}, defined in (7.11), monotonically decreases.

$$D_{\text{KL}}(p_{0t} \,||\, q_{0t}) = \frac{\|\mathbf{A}^H\boldsymbol{\delta}\|^2}{2(\sigma^2(t) - \sigma^2(0))}\,, t \in (0, 1]\,. \tag{7.11}$$

2. The KL divergence between p_t and q_t monotonically decreases, i.e.,

$$\frac{dD_{\text{KL}}(p_t \,||\, q_t)}{dt} \leq 0\,. \tag{7.12}$$

It is important to highlight that our Theorem uses the VE-SDE, where the probability distributions are from the standard Bayesian framework of DMs (Ho et al., 2020).

7.3.2 Selection of the Process-Switching Time Step

Here, we present an approximate method to obtain $t^* < 1$ (or $N_{t^*} < N_r$) based on the Maximum Mean Discrepancy (MMD) metric (Gretton et al., 2006). The MMD metric measures the dissimilarity between two distributions by comparing their mean embedding in a reproducing kernel Hilbert

space. It is commonly employed in machine learning and statistics for various tasks, including domain adaptation (Guan and Liu, 2021) and kernel methods (Hofmann et al., 2008).

We utilize the MMD metric to approximately quantify the empirical distribution shift between the original distribution $p(\mathbf{z})$ and the perturbed images' distribution $q(\mathbf{z})$. During the forward diffusion process, let $Z(i)$ and $Z_{\mathrm{p}}(i)$ (with $|Z(i)| = |Z_{\mathrm{p}}(i)|$) represent the set of unperturbed and perturbed images, respectively, at discrete time step i, where $|\cdot|$ denotes the cardinality of a set. Since we lack access to the exact distributions, we can approximate $\mathrm{MMD}(p_i, q_i)$ using empirical distributions as follows:

$$\begin{aligned}\mathrm{MMD}(p_i, q_i) \approx C\Big(&\sum_{\mathbf{z}(i),\mathbf{z}'(i)\in Z(i),\mathbf{z}(i)\neq\mathbf{z}'(i)} k(\mathbf{z}(i), \mathbf{z}'(i)) + \\ &\sum_{\mathbf{z}(i),\mathbf{z}'(i)\in Z_{\mathrm{p}(i)},\mathbf{z}(i)\neq\mathbf{z}'(i)} k(\mathbf{z}(i), \mathbf{z}'(i))\Big) \\ -\frac{2}{|Z(i)|^2} &\sum_{\mathbf{z}(i)\in Z(i),\mathbf{z}'(i)\in Z_{\mathrm{p}}(i)} k(\mathbf{z}(i), \mathbf{z}'(i)) \,,\end{aligned} \tag{7.13}$$

where $C = 1/(|Z(i)|(|Z(i)| - 1))$ is used for brevity, and $k(\mathbf{z}(i), \mathbf{z}'(i)) = \exp(-\|\mathbf{z}(i) - \mathbf{z}'(i)\|^2/2v^2)$ is the Gaussian kernel parameterized by $v > 0$.

Considering the balance between purifying additive perturbations (achieved with a larger t^*) and preserving global structures (achieved with a smaller t^*) within perturbed samples, there exists an ideal value of t^* that yields a robust reconstruction accuracy. In the case of the worst-case additive perturbations, the changes are usually small and can be rectified with a small t^*. It was shown in (Nie et al., 2022b) that the most efficient choice of t^* related to adversarial robustness tends to be on the smaller side. As such, our objective is to find the minimum value of $i \in [N_r]$ for which $\mathrm{MMD}(p_i, q_i) \approx 0$. Consequently, we formulate the following optimization problem to determine the near-optimal discrete PST step, N_{t^*}.

$$N_{t^*} := \left\{\arg\min_{i\in[N_r]} i \;\text{ s.t. }\; \mathrm{MMD}(p_i, q_i) = 0\right\} . \tag{7.14}$$

In order to obtain the solution of (7.14), it is required to perform the forward diffusion (steps 2 and 3 in Algorithm 7.2) on the unperturbed and perturbed samples until the constraint is satisfied.

Algorithm 7.3 Our Robust MoDL Pipeline

Input: Perturbed measurements $\mathbf{y}_{\text{pert}}$, operator $\mathbf{A}$, trained DM s_ϕ, PST step N_{t^*}, number of unrolling steps N, and fine-tuned MoDL parameters θ_{FT}.

Output: Reconstructed image after purification $\mathbf{x}$.

1: **Obtain** $\mathbf{z}_{\text{pert}}^{\text{pur}} = \text{DP}_\phi(\mathbf{y}_{\text{pert}}, \mathbf{A}, N_{t^*})$.
2: **Initialize** MoDL reconstructed image as $\mathbf{x}_0 = \mathbf{z}_{\text{pert}}^{\text{pur}}$
3: **For** $j \in \{0, \ldots, N-1\}$ \\MoDL unrolling steps
4: **Obtain** $\mathbf{z}_j \leftarrow f_{\theta_{\text{FT}}}(\mathbf{x}_j)$
5: **Obtain** $\mathbf{x}_{j+1} \leftarrow (\mathbf{A}^H\mathbf{A} + \lambda\mathbf{I})^{-1}(\mathbf{z}_{\text{pert}}^{\text{pur}} + \lambda\mathbf{z}_j)$
6: **Obtain** $\mathbf{x} \leftarrow \mathbf{x}_N$

Since we have knowledge of the source of perturbations that allows us to obtain Z_{p} from Z, we remark that the PST step selection method we propose can be applied to any diffusion purification task.

7.3.3 Fine-tuning with Purified Perturbed Examples

In this subsection, drawing inspiration from the widely used 'pre-training + fine-tuning' approach (Zoph et al., 2020; Salman et al., 2020), we propose fine-tuning the parameters of MoDL, which are obtained through the process outlined in Section 7.2.A, using contaminated purified examples.

We start with pre-trained parameters θ, and utilize noised purified examples for fine-tuning. Let θ_{FT} represent the fine-tuned parameters specific to MoDL. Initially, we set θ_{FT} equal to θ. Then, for each measurement $\mathbf{y}$ within dataset D, we generate a noisy version of the aliased reconstruction, $\mathbf{A}^H(\mathbf{y} + \mathbf{v})$, where $\mathbf{v}$ is drawn from a normal distribution $\mathcal{N}(0, \sigma_{\text{FT}}\mathbf{I})$. Subsequently, for every $(\mathbf{y}, \mathbf{x})$, we follow the procedure outlined in (Aggarwal et al., 2018), while initializing $\mathbf{x}_0$ as

$$\mathbf{x}_0 = \text{DP}_\phi(\mathbf{y} + \mathbf{v}, \mathbf{A}, N_{t^*}) \,. \tag{7.15}$$

Having trained θ_{FT} that maps $\mathbf{x}_0$ to fully-sampled reconstructions, at the testing phase, the robust MoDL MRI reconstruction using diffusion purification is represented in Algorithm 7.3. A block diagram of the proposed approach is given in Figure 7.2.

We emphasize that while our primary focus is on the formulation of MoDL under which we develop our proposed approach, in the last subsection of our experimental results, we demonstrate the versatility of our approach by showcasing its applicability to other DL-based supervised MRI

reconstruction models.

7.4 Experimental Results

In this section, we start by illustrating our experimental setup, baselines, and the instability sources and the generalization challenges considered in this work. Subsequently, we present results for the process-switching time (PST) step selection through our MMD-based method. Following this, we present the primary results showcasing the robustness of our approach. Furthermore, we present visualizations illustrating knee and brain MRI reconstructions.

7.4.1 Experimental Setup

In the case of MoDL, we employ a configuration with $N = 6$ unrolling steps and a regularization parameter $\lambda = 1$. The architecture of f_θ is selected as the Deep Iterative Down-Up Network (Yu et al., 2019b). Additionally, we set the convergence threshold for the conjugate gradient optimization used in the data consistency step of (7.2) to 10^{-6}. In the DM setting, $t \in [0, 1]$ is discretized into 500 steps. We adopt a pre-trained DM model from (Chung and Ye, 2022), where $\sigma(i)$ is a geometric series selected as $\sigma(i) = 0.01(37800)^{\frac{i}{N_r - 1}}$. We note that the DM model was trained on the knee training dataset. We conduct our experiments on the fastMRI dataset (Zbontar et al., 2018), using 3000 purified images for fine-tuning the pre-trained MoDL network. Additionally, 20 images are reserved for validation, and 64 images are used for testing. Moreover, we use $\sigma_{\text{FT}} = 0.01$. The multi-coil image data is acquired using 15 coils and is cropped to a resolution of 320×320 pixels for MRI reconstruction. To simulate undersampling of the MRI k-space, we adopt a Cartesian mask with 4x acceleration (equivalent to a 25% sampling rate). Sensitivity maps for the coils, which are incorporated into the operator $\mathbf{A}$ for all scenarios, are obtained using the BART toolbox (Tamir et al., 2016). Rather than employing the root-sum-of-squares reconstruction method, we apply sensitivity map-based reconstruction. The quality of the reconstructed images is evaluated using the Peak Signal-to-Noise Ratio (PSNR) in dB, and the Structural Similarity Index Measure (SSIM), which returns values in $[0, 1]$ with 1 indicating identical images. All the experiments are conducted on a single RTX5000 GPU machine.

Baselines: Here, we list the baselines used in our experiments.

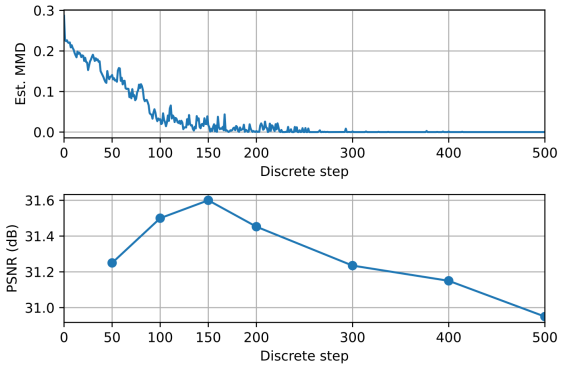


Figure 7.3 Selection of the PST step. Estimated MMD (using (7.13)) w.r.t. the discrete steps $i \in [N_r]$ (*top*). Ablation study by comparing with the ground truth (*bottom*).

7.4.1.1 Vanilla DL-based MRI Reconstructors

Here, we consider standalone MoDL and Recurrent VarNet. These are also incorporated within our proposed framework..

7.4.1.2 Adversarial Training

In AT, we implemented a 30-step PGD procedure within its minimax formulation.

7.4.1.3 E2E Randomized Smoothing

For E2E-RS, we introduced Gaussian noise with a standard deviation of 0.01, and to perform the smoothing operation, we employed 10 Monte Carlo samplings.

7.4.1.4 Score-MRI

We compare our proposed approach with a diffusion-based method, namely the score-MRI work in (Chung et al., 2023c).

7.4.1.5 Standalone Diffusion Purification

We report results of using only the diffusion purifier with data-consistency (Algorithm 2). We use 'DP' to refer to this case.

7.4.1.6 LORAKI

LORAKI is an unsupervised recurrent neural network tailored for MRI reconstruction in k-space, representing a scan-specific method. Here, we utilize the modification in (Ak**c**cakaya et al., 2019), where a publicly available code is used. For generating worst-case additive noise, we use the same approach as in (7.17).

7.4.2 Implementation Details for the Sources of Instabilities & Generalization Settings

7.4.2.1 k-space Additive Noise

Here, we consider additive perturbations applied to the measurements $\mathbf{y}$. Recall that for example in the unrolled MoDL, this is both an input and is used in the conjugate gradients (CG) scheme in the data consistency step. We consider two types of additive noise: a zero-mean complex Gaussian random vector with a variance of 0.01, and worst-case additive perturbations. For the latter, we employed two gradient-based optimization techniques. The first method is the conventional ℓ_∞-norm PGD (Madry et al., 2017) with 30 iterations and a perturbation budget of $\epsilon = 0.004$. The second approach utilizes the advanced momentum-based AUTO attack (Croce and Hein, 2020), configured similarly to PGD. To generate perturbations using PGD or AUTO, it is necessary to calculate the gradients w.r.t. the input of our model.

In this paper, we consider an additional case where we apply the method from (Nie et al., 2022b) and calculate the gradients to propagate through both MoDL and the SDE of the DP. This represents the worst-case additive perturbations w.r.t. the DP and MoDL. In this case, the perturbations are generated as:

$$\begin{aligned}\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \mathcal{L}\Big(&\mathrm{MoDL}_{\theta_{\mathrm{FT}}}(\mathrm{DP}_\phi(\mathbf{A}^H\mathbf{y}, N_{t^*})), \\ &\mathrm{MoDL}_{\theta_{\mathrm{FT}}}(\mathrm{DP}_\phi(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}), N_{t^*}))\Big)\,.\end{aligned} \tag{7.16}$$

Worst-case additive noise for AT, E2E-RS, Recurrent Var Net and LORAKI are generated using the optimization problem in (7.3), with changing the structure of the network.

For score-MRI and standalone diffusion purification, we use Equations (7.17) and (7.18), respectively, which are modified versions of (7.3).

$$\begin{aligned}\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \mathcal{L}\Big(&\mathrm{PCDC}\left(s_\phi(\mathbf{z}(N_r), N_r), \mathbf{y}+\boldsymbol{\delta}, \mathbf{A}, N_r, 0\right), \\ &\mathrm{PCDC}\left(s_\phi(\mathbf{z}(N_r), N_r), \mathbf{y}, \mathbf{A}, N_r, 0\right)\Big)\,.\end{aligned} \tag{7.17}$$

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \mathcal{L}\big(\mathrm{DP}_\phi(\mathbf{A}^H\mathbf{y}, N_{t^*}), \mathrm{DP}_\phi(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}), N_{t^*})\big)\,. \tag{7.18}$$

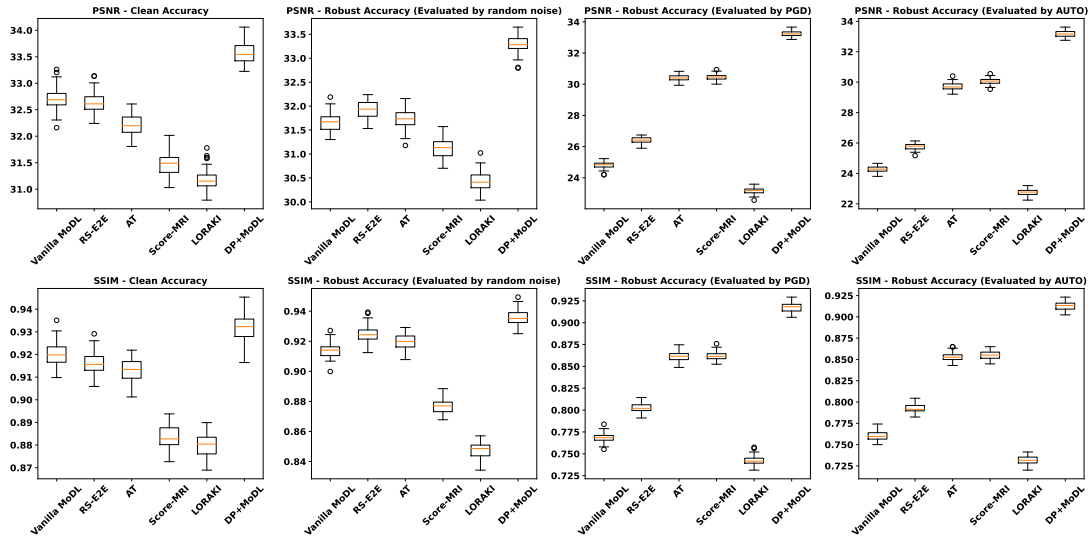


Figure 7.4 Reconstruction accuracy box plots for the **knee** fastMRI dataset with 4x Acceleration factor. The additive Gaussian random noise of the second column plots is obtained using variance of 0.01. The worst-case additive noise of the third and fourth columns are obtained using PGD and AUTO methods with $\epsilon = 0.02$.

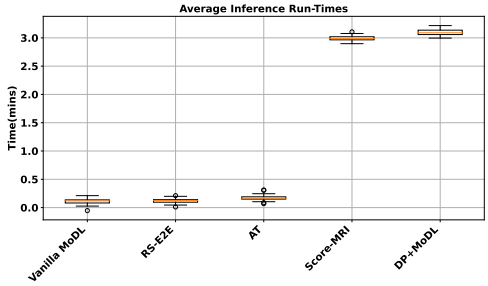


Figure 7.5 Average inference run-time of our proposed approach and the baselines for the experiment setting of the top right box plot of Figure 7.4.

7.4.2.2 Training/Testing Sampling Protocol and Undersampling Rate Disparities

Here, we consider two variations in the construction of the forward operator $\mathbf{A}$ between the training and testing phases. In other words, we train MoDL with $\mathbf{A}$ and evaluate it with different $\mathbf{A}_{\text{test}}$. The first variation involves using a different acceleration factor (sampling rate), while the second involves shifts in the locations of the k-space samples. In particular, for the first variation, we train MoDL with 4x undersampling, and test it with {2x,3x,4x,5x,6x,7x,8x}. For the second variation, we train MoDL using a 4x mask and then evaluate it using various shifted versions of the original mask. Specifically, the central part of the mask (low frequencies) remains constant, whereas the higher frequency phase encodes are shifted by {5%,10%,15%,20%,25%}.

7.4.2.3 Unseen Anatomies & Pathologies at Testing Phase

We evaluate our method's performance in the presence of white non-specific lesions using the fastMRI+ dataset. In particular, the DL-based image reconstructor is trained on the lesion-free fastMRI dataset and evaluated on the fastMRI+ dataset.

Furthermore, we evaluate the performance of the proposed method with testing brain measurements but wherein the diffusion purifier was pre-trained on knee data (i.e., different anatomy).

7.4.3 Selection of the PST Step

In this section, we conduct an experiment to evaluate the effectiveness of the proposed MMD-based method in determining the near-optimal PST step, denoted as N_{t^*}. The experiment is depicted in Figure 7.3 (*top*), where we present the MMD values computed using (7.13). Additionally, Figure 7.3 (*bottom*) displays the results obtained when applying various values of N_{t^*} within our pipeline, with corresponding PSNR values compared to ground truth images. In this experiment, we calculate the MMD values by setting the Gaussian kernel v as the mean of the magnitude of the images in set Z, which comprises images $\mathbf{A}^H\mathbf{y}$ for 20 scans $\mathbf{y} \in D$. For the perturbed images, we utilize the worst-case additive perturbations, denoted as $\boldsymbol{\delta}$, calculated from (7.3). Consequently, the set Z_{p} encompasses $\mathbf{A}^H(\mathbf{y} + \boldsymbol{\delta})$ for the same measurements used in Z.

The results of Figure 7.3 (*bottom*) show that, in comparison to the ground truth, the optimal PSNR result is achieved at $N_{t^*} = 150$, consistent with the observed approximate MMD value in Figure 7.3 (*top*). Furthermore, it is evident that although the MMD values for N_{t^*} in the range $(150, 500]$ are also close to zero, PSNR values begin to deteriorate. This observation aligns with the intuition that increasing the value of N_{t^*} effectively removes perturbations but runs the risk of losing image structure. Consequently, for the remainder of this paper, we adopt $N_{t^*} = 150$ as our chosen setting.

Furthermore, we remark that the number of reverse (purification) process steps chosen for our robustification task, which is 150, is notably lower than the requirement in the diffusion-based image reconstruction task presented in (Chung and Ye, 2022), where 500 steps were used.

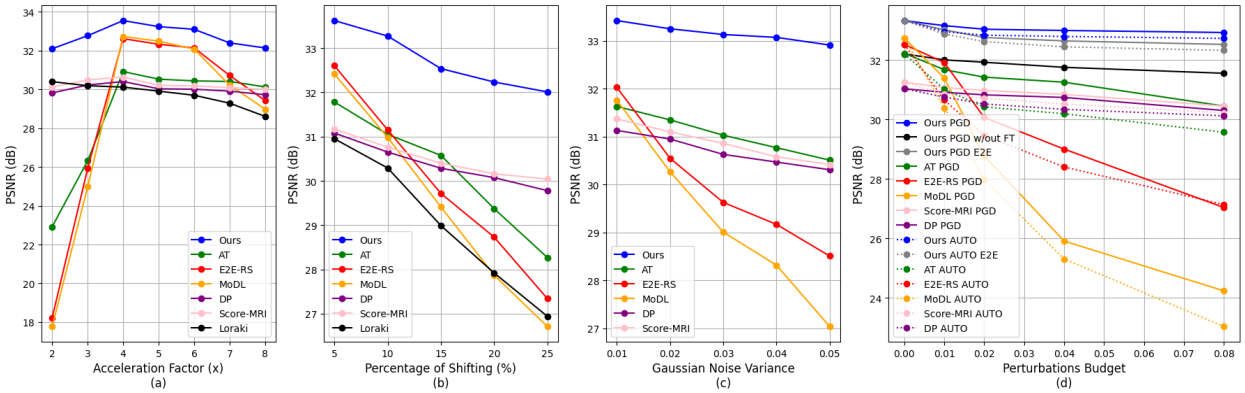


Figure 7.6 Robustness evaluation against variations in: (a) acceleration factors, (b) locations of k-space sampling, (c) variance level of the Gaussian random additive noise, and (d) perturbation budget of the worst-case additive disturbances generated by PGD and AUTO methods. The 'PGD E2E' and 'AUTO E2E' in (d) correspond to the cases of generating end-to-end perturbations while calculating gradients through propagating the DP and MoDL. Furthermore, 'Ours PGD w/out FT' corresponds to the case where no MoDL fine-tuning is applied. This figure is best viewed in color.

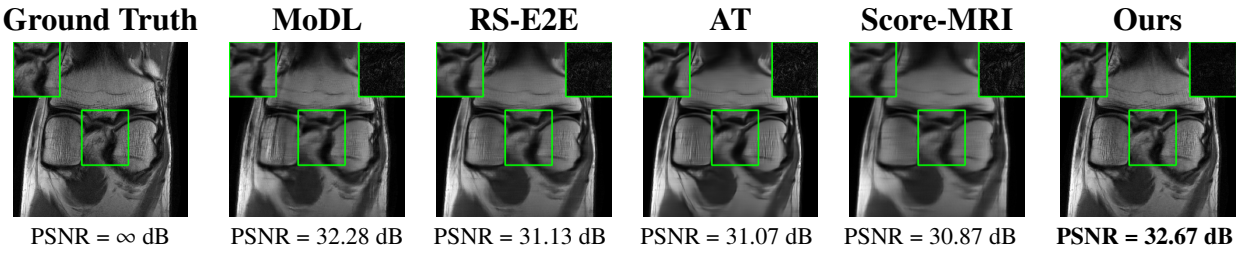


Figure 7.7 Visualization of ground-truth and reconstructed images using different methods, evaluated by the knee fastMRI testing set with 8x acceleration factor.

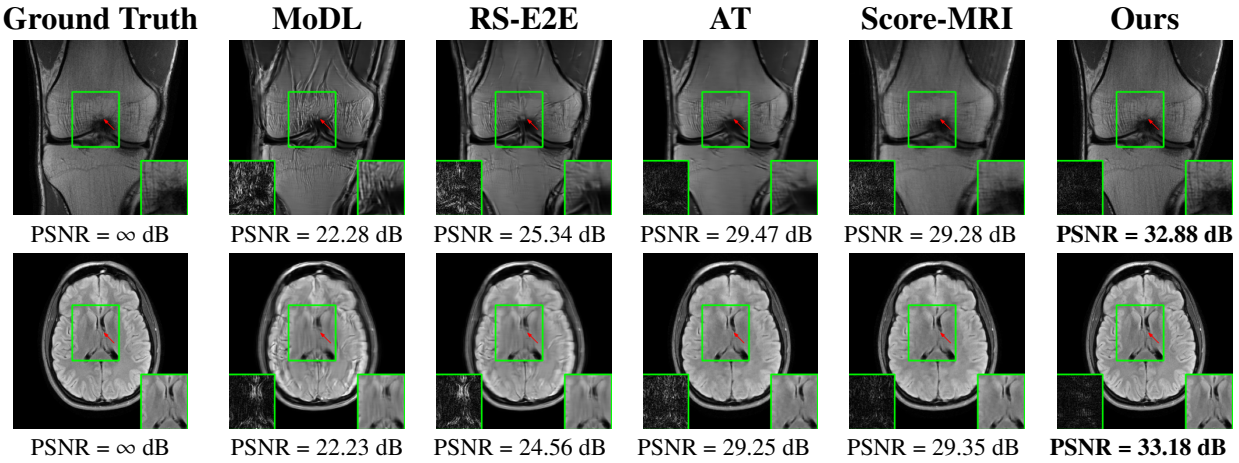


Figure 7.8 Visualization of ground-truth and reconstructed images using different methods, evaluated by PGD-based worst-case additive perturbations with $\epsilon = 0.02$.

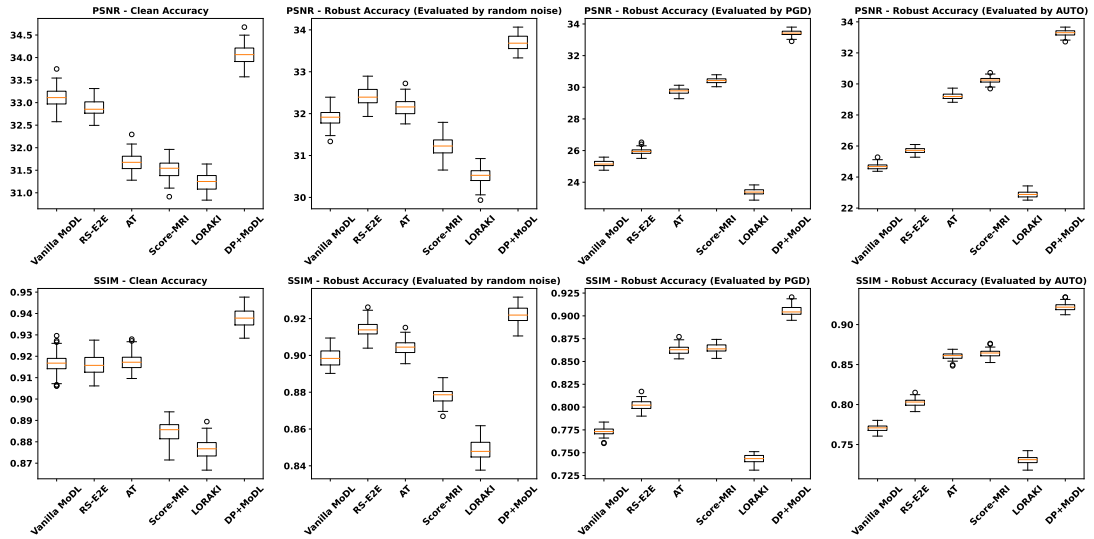


Figure 7.9 Reconstruction accuracy box plots of the **brain** fastMRI dataset with 4x Acceleration factor. The additive Gaussian random noise of the second column plots are obtained from using variance of 0.01. The worst-case additive noise of the third and fourth columns are obtained using PGD and AUTO methods with $\epsilon = 0.02$.

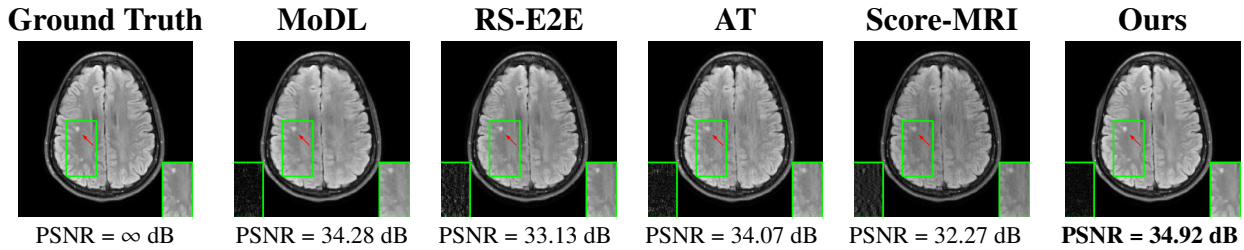


Figure 7.10 Visualization of ground-truth and reconstructed images using different methods, trained with fastMRI (without lesions), and evaluated by the fastMRI+ dataset (with lesions).

7.4.4 Robustness Results

7.4.4.1 Robustness to Additive Perturbations

Figure 7.4 presents box plots for a comprehensive view of the performance of our robustification method, as well as that of Vanilla MoDL, AT, E2E-RS, LORAKI, and Score-MRI, assessed through PSNR (top) and SSIM (bottom) metrics using the knee dataset. We evaluate these methods across multiple scenarios, including benign aliased images (top and bottom first plots), images subjected to additive random Gaussian noise with variance of 0.01 (top and bottom second plots), and images with additive worst-case perturbations generated using PGD and AUTO methods with $\epsilon = 0.02$ (top and bottom last two plots).

While AT, RS, and score-MRI show improvements when compared to vanilla MoDL, we observe

that, on average, our robustification approach reports the highest values of PSNR and SSIM. For the example of the rightmost plot, our method achieves an average PSNR that is approximately 3 dB more than score-MRI and nearly 9 dB more than Vanilla MoDL. Additionally, the PSNR and SSIM results in the first plots (top and bottom) indicate an improvement of our proposed approach (DP+MoDL), even in the absence of any perturbations. It is important to highlight that although our proposed approach reports the highest PSNR values in terms of reconstruction, it requires larger inference run-time compared to AT, RS, and vanilla MoDL. In Figure 7.5, we present inference run-times for the setting of the top right box plot of Figure 7.4. As observed, on average, our method and score-MRI need nearly 3 minutes per image, whereas other methods require only 60 seconds or less. The increased run-time is attributed to the application of the proposed diffusion purification prior to DL-reconstructor, representing a trade-off.

In Figure 7.6 (c), we present the PSNR values of AT, E2E-RS, DP, score-MRI, and our approach, evaluated under different levels of added Gaussian noise during testing. Notably, as the noise level (indicated by the variance) increases, the reported PSNR values decrease for all methods. However, our approach consistently reports higher PSNR values when compared to the other baselines across all tested noise levels. For instance, when faced with a variance of 0.05, our method reports nearly 33 dB whereas the second best (in this case AT) reports a PSNR of 30.5 dB.

In Figure 7.6 (d), we present the PSNR performance of our approach and the considered baselines, evaluated under varying perturbation budgets given by the values of ϵ. The evaluation encompasses both PGD and AUTO methods. Additionally, we explore the PGD E2E and AUTO E2E scenarios, which involve generating end-to-end perturbations using (7.16). As the perturbation budget increases, all methods experience a decline in their PSNR values, which is expected. However, we observe that our approach consistently returns the highest PSNR values across the entire range of perturbation budgets. We also observe that employing the E2E attack results in slightly lower PSNR values compared to the case of generating perturbations solely w.r.t. MoDL. Finally, we observe that the AUTO results are marginally lower than those of PGD, which aligns with expectations since AUTO represents a more advanced approach in generating worst-case additive noise.

Moreover, in Figure 7.6 (d), we illustrate the effect of fine-tuning on the robustness of our method. Specifically, we compare PSNR values for our approach when exposed to PGD-based worst-case additive perturbations under two scenarios: with fine-tuning MoDL using perturbed purified training samples (i.e., $f_{\theta_{\text{FT}}}$) and without fine-tuning, relying solely on the pre-trained MoDL (i.e., f_{θ}). These two cases are represented by the solid blue and blue plots in Figure 7.6 (d). The results clearly highlight that the pre-trained+fine-tuned MoDL enhances robustness, as evidenced by the higher PSNR values compared to pre-trained MoDL. We also note that the results obtained without fine-tuning are slightly higher than those achieved using AT (see the solid green curve in Figure 7.6 (d)). This indicates that MoDL+DP without fine-tuning still exhibits improvements when compared to AT, vanilla MoDL, and RS-E2E

Figure 7.8 presents visual comparison of image reconstructions and their associated reconstruction errors within a closely examined region. Each image in the figure includes two inset panels in the bottom-left and bottom-right corners. The bottom-left inset panel, enclosed within a green bounding box, serves as a reference for the region of interest in the image. In contrast, the bottom-right inset panel depicts an error map in relation to the ground truth. Notably, our method stands out in its ability to capture more features from the original image, surpassing the performance of alternative methods (as also evident from the reported PSNR values).

7.4.4.2 Robustness to Different Sampling Protocols & Undersampling Rates

In Figure 7.6 (a), we illustrate the performance across different acceleration factors. During training, a k-space undersampling or acceleration factor of 4x was employed. However, during testing, we assess performance with various acceleration factors ranging from 2x to 8x. It is evident that when the acceleration factor matches the training phase (4x), all methods exhibit their highest PSNR results compared to when different acceleration factors are used. Nevertheless, when compared to the other methods, our approach consistently reports the highest PSNR values when tested with acceleration factors other than 4x. For instance, at 2x acceleration, AT and E2E-RS report PSNR values of 21 dB or lower, while our approach achieves nearly 32 dB. Additionally, in Figure 7.6 (a), we report results of using LORAKI with different acceleration factors. As observed,

Models	Accuracy		Training
Metrics	PSNR ↑	SSIM ↑	Acceleration Factor
Vanilla MoDL	33.25	0.920	8x
E2E-RS	33.12	0.917	8x
AT	32.17	0.913	8x
Score-MRI	33.5	0.899	4x
DP+MoDL	33.67	0.922	4x

Table 7.1 Reconstruction accuracy for fastMRI knee data using the testing portion of the dataset with acceleration of **8x**.

LORAKI reports lower PSNR values when compared to our proposed approach.

Figure 7.6 (b) shows the PSNR values of our proposed approach and the considered baselines, assessed under varying percentages of shifts in the location of the k-space sampling during testing. The shifts were applied to high-frequency phase encode locations in the original sampling pattern or mask. This is to help understand reconstruction robustness when the sampling masks change a lot at a fixed k-space undersampling factor. We observe that as the percentage of shifts increases, the reported PSNR values decrease across all methods. However, we observe that, our method consistently outperforms the other approaches across all tested percentages, exhibiting the highest PSNR values. For instance, when the mask at testing time contains 25% shift when compared to the training mask, our method achieves 32 dB whereas all other methods report PSNR values of 31.2 dB or less.

To further underscore the generalization and robustness of our proposed approach, we designed an experiment with different training and testing settings across different methods. Specifically, we trained vanilla MoDL, AT, and RS models using an 8x acceleration factor, while our method and score-MRI were trained with a 4x acceleration factor. Subsequently, we subjected benign measurements to testing with an 8x acceleration factor, aligning with the training settings of MoDL, AT, and RS, rather than 4x. The results, given in Table 7.1, showcase that our method, despite undergoing testing with a different acceleration setting, reports slightly higher PSNR (33.67 dB) and SSIM (0.922) values when compared to other methods. Moreover, the visualizations in Figure 7.7 show that when tested with an 8x acceleration factor despite being trained on 4x, our proposed approach outperforms the considered baselines under conditions where both training and testing acceleration factors are 8x.

Models Metrics	MRI Reconstruction Accuracy PSNR ↑	 SSIM ↑
Vanilla MoDL	31.25	0.915
E2E-RS	31.12	0.912
AT	30.87	0.910
Score-MRI	30.22	0.885
DP+MoDL (Ours)	32.4	0.919

Table 7.2 Brain fastMRI+ (with lesion) results.

Models	Clean Accuracy		Robust Accuracy (Evaluated by random noise)		Robust Accuracy (Evaluated by PGD)	
Metrics	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Vanilla RecurrentVarNet	33.78	0.925	32.89	0.91	26.5	0.793
AT+RecurrentVarNet	33.19	0.919	33.01	0.914	31.67	0.892
E2E-RS+RecurrentVarNet	33.67	0.922	33.12	0.915	30.20	0.875
DP+RecurrentVarNet (Ours)	34.33	0.941	34.07	0.938	33.64	0.935

Table 7.3 **Brain** dataset reconstruction accuracy using **Recurrent Variational Network** as our DL-based image reconstructor.

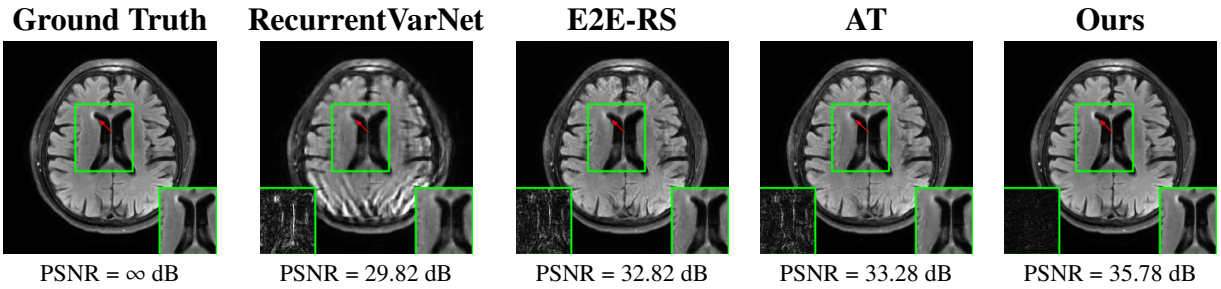


Figure 7.11 Visualization of ground-truth and reconstructed images using RecurrentVarNet and RecurrentVarNet+DP (Ours) methods, evaluated by PGD-based worst-case additive perturbations with $\epsilon = 0.02$.

7.4.4.3 Robustness to Anatomical Variations

In Figure 7.9, we replicate the experiment conducted in Figure 7.4, this time utilizing the brain dataset. Notably, MoDL underwent fine-tuning using perturbed purified examples sourced from the training set of the brain dataset. When comparing the results of our proposed method with other approaches, we find that the observations of Figure 7.4 remain consistent. For the PGD case (third column), our method reports an average SSIM of nearly 0.91 whereas Vanilla MoDL (the DL-reconstructor considered in this experiment) reports an average SSIM of approximately 0.775. An important point to highlight is that the pre-trained DM employed in our purification stage for this experiment was originally trained exclusively on knee data, without any exposure to brain data. This underscores the robust generalization capabilities of the diffusion purification process within our

approach, extending its effectiveness to previously unseen MRI datasets.

It's worth mentioning that similar diffusion model generalization capabilities were also observed in the study conducted by Chung et al. (Chung and Ye, 2022). However, further thorough investigation is required to precisely determine the limitations of these generalization capabilities, and this remains a promising direction for future research.

Here, we employ the fastMRI+ dataset to assess our approach's image reconstruction capability, contrasting the outcomes with relevant baselines. For the training phase, we employ the original fastMRI brain dataset, which excludes lesion cases, as the basis for training all methods. During the testing phase, however, we utilize the lesion dataset. Table 7.2 shows the results, where our method reports the highest PSNR and SSIM values compared to other baselines. It is important to highlight that, unlike the cases of additive k-space noise and training/testing sampling protocol and undersampling rate disparities, the improvements observed from utilizing our method with unseen lesions are somewhat marginal as seen from the average PSNR and SSIM results (at least 1.2 dB PSNR improvement when compared to the 2nd best results). Additionally, visualizations are provided Figure 7.10 where we highlight the nonspecific white matter lesion area. As observed, both visually and in terms of PSNR values, our approach reports improved results when compared to the other baselines.

7.4.5 Applying Our Method to Other DL-based MRI Reconstruction Models

Here, we demonstrate the applicability of our diffusion purification strategy to other DL-based supervised MRI reconstructors. Specifically, we explore the Recurrent Variational Network (RecurrentVarNet) (Yiasemis et al., 2022), presenting results both with and without perturbations, as well as with and without the integration of our diffusion purification technique. The results are summarized in Table 7.3. As depicted in the table, when the standalone RecurrentVarNet (or RecurrentVarNet integrated with AT and/or RS) encounters additive worst-case perturbations in the measurement space, the reported PSNR and SSIM scores (last two columns of the first three rows) experience a significant drop (for example, the Vanilla RecurrentVarNet encounters a PSNR drop of nearly 7 dB). However, upon employing our diffusion purification (last row), we observe only a

marginal decrease in performance (of 0.69 dB). These findings illustrate that our strategy can be integrated well with general DL-based reconstructors. The visualizations in Figure 7.11 provide additional support for our claim.

7.5 Conclusion

Recent studies have unmasked vulnerabilities in DL-based MRI reconstruction methods—namely, susceptibility to additive perturbations and variations in training/testing settings, such as acceleration factors and k-space sampling patterns. This paper has addressed these challenges by harnessing the power of diffusion models. Our innovative robustification strategy enhanced the resilience of DL-based MRI reconstruction models by integrating pre-trained diffusion models as noise purifiers. Unlike conventional robustification techniques like adversarial training (AT), our method eliminated the need for complex minimax optimization problems. Instead, it simply requires fine-tuning on perturbed purified examples. Our extensive experiments have illustrated the remarkable efficacy of our approach in mitigating different instabilities when compared to utilizig diffusion-based MRI reconstructors and leading robustification methods, including AT and randomized smoothing. We also evaluated the robustness of our approach using an MRI dataset with lesions. Moreover, we illustrated the adaptability of our strategy to multiple reconstruction models. These findings underscore the promise of leveraging diffusion models to enhance the robustness and reliability of DL-based MRI reconstruction, paving the way for more dependable and accurate medical imaging technologies in the future.

7.6 Proof of Theorem 1

Proof of Theorem 1: For the first part, we begin by establishing the results in (7.11). Utilizing the VE-SDE formulation of DMs, the conditional distributions p_{0t} and q_{0t} are expressed as per the following equations (Song et al., 2021c).

$$p_{0t}(\mathbf{z}(t) \mid \mathbf{z}) = \mathcal{N}(\mathbf{z}(t); \mathbf{z}, (\sigma^2(t) - \sigma^2(0))\mathbf{I}) \,, \tag{7.19a}$$

$$q_{0t}(\mathbf{z}(t) \mid \mathbf{z}_{\text{pert}}) = \mathcal{N}(\mathbf{z}(t); \mathbf{z}_{\text{pert}}, (\sigma^2(t) - \sigma^2(0))\mathbf{I}) \,. \tag{7.19b}$$

Notably, these two distributions have different means, but share the same covariance. Consequently, the D_{KL} can be obtained as

$$D_{\mathrm{KL}}(p_{0t} \,||\, q_{0t}) = \frac{1}{2}\Bigg(\log\Big(\frac{\det(\sigma^2\mathbf{I})}{\det(\sigma^2\mathbf{I})}\Big) + \mathrm{Tr}\Big((\sigma^2\mathbf{I})^{-1}(\sigma^2\mathbf{I})\Big) + (\mathbf{z}_{\mathrm{pert}} - \mathbf{z})^T(\sigma^2\mathbf{I})^{-1}(\mathbf{z}_{\mathrm{pert}} - \mathbf{z}) - n\Bigg),$$

where $\det(\cdot)$ (resp. $\mathrm{Tr}(\cdot)$) denotes the determinant (resp. trace) of a matrix, and $\sigma^2 = \sigma^2(t) - \sigma^2(0)$ is used for brevity. Since $\log(1) = 0$, the first term is zero. Given the definition of the trace and the identity matrix properties, the second term reduces to n and cancels the last term. Since $\mathbf{A}^H\boldsymbol{\delta} = \mathbf{z}_{\mathrm{pert}} - \mathbf{z}$, and $(\mathbf{A}^H\boldsymbol{\delta})^T\mathbf{A}^H\boldsymbol{\delta} \geq 0$, then Equation (7.11) holds.

Subsequently, the numerator in (7.11) is more than or equal to 0 (can only be zero if $\boldsymbol{\delta} = 0$), and is not a function of t. Moreover, since $\sigma(t) = \sigma_l(\sigma_u/\sigma_l)^t$, where $\sigma_l \in (0, 1)$ and $\sigma_u > 1$ are constants, it is evident that the denominator monotonically increases as t increases.

In conclusion, the rate of change of $D_{\mathrm{KL}}(p_{0t} \,||\, q_{0t})$ w.r.t. t (as long as $\boldsymbol{\delta} \neq 0$) is less than 0. Given the derivative of $\sigma(t)$ w.r.t. t is $\frac{d\sigma(t)}{dt} = \sigma_l \log(\sigma_u/\sigma_l)(\sigma_u/\sigma_l)^t$, this is supported by

$$\frac{dD_{\mathrm{KL}}(p_{0t} \,||\, q_{0t})}{dt} = \frac{-\|\mathbf{A}^H\boldsymbol{\delta}\|^2\sigma_l \log(\sigma_u/\sigma_l)(\sigma_u/\sigma_l)^{2t}}{\left(\sigma^2(t) - \sigma_l^2\right)^2} < 0\,.$$

This inequality establishes that $D_{\mathrm{KL}}(p_{0t} \,||\, q_{0t})$ monotonically decreases as time travels from $t = 0$ to $t = 1$ while employing the forward process defined in (7.6). Consequently, the proof of the first part is complete.

The proof of the second part follows from (Song et al., 2021c) and (Nie et al., 2022b). Using the Fokker-Planck-Kolmogorov representation (Särkkä and Solin, 2019) for the forward process in (7.6), we write

$$\frac{dp_t(\mathbf{z})}{dt} = \frac{1}{2}\nabla_{\mathbf{z}} \cdot \Big(p_t(\mathbf{z})\frac{d\sigma^2(t)}{dt}\nabla_{\mathbf{z}}\log p_t(\mathbf{z})\Big)\,, \tag{7.20a}$$

$$\frac{dq_t(\mathbf{z})}{dt} = \frac{1}{2}\nabla_{\mathbf{z}} \cdot \Big(q_t(\mathbf{z})\frac{d\sigma^2(t)}{dt}\nabla_{\mathbf{z}}\log q_t(\mathbf{z})\Big)\,. \tag{7.20b}$$

Employing the definition of the KL divergence, Equation (7.20), integration by parts, and assuming the smoothness and fast decay of $p_t(\mathbf{z})$ and $q_t(\mathbf{z})$, we can derive the derivative of the KL

divergence w.r.t. t:

$$\frac{dD_{\mathrm{KL}}(p_t \,||\, q_t)}{dt} = -\frac{1}{2}\frac{d\sigma^2(t)}{dt} D_{\mathrm{F}}(p_t \,||\, q_t) \leq 0\,, \tag{7.21}$$

where

$$D_{\mathrm{F}}(p_t \,||\, q_t) = \int p_t(\mathbf{z}) \|\nabla_{\mathbf{z}} \log p_t(\mathbf{z}) - \nabla_{\mathbf{z}} \log q_t(\mathbf{z})\|^2 d\mathbf{z} \geq 0\,,$$

denotes the Fisher divergence. Given that $\frac{d\sigma^2(t)}{dt} > 0$, the proof of the second part is thereby established.

CHAPTER 8

STEP-WISE TRIPLE-CONSISTENT DIFFUSION SAMPLING

8.1 Introduction

In the previous chapter, we introduced the diffusion model as the purifier of the image reconstruction in order to handle the different kinds of noise. However, a key bottleneck in DMs is their computational speed, as they are slower than other generative models due to the large number of sampling steps. Although various methods have been proposed to reduce sampling frequency (e.g., (Song et al., 2023b)), these improvements have yet to be fully realized for DMs applied to IPs. Most existing methods still require dense sampling, which continues to pose speed challenges.

Contributions: In this chapter, we: (*i*) identify key issues in accelerating DMs for IPs, (*ii*) propose three conditions that could fully leverage the information from the measurements and the pre-trained diffusion model to effectively address these issues, and (*iii*) present a new optimization-based method in the pixel space that satisfies these conditions. We refer to our accelerated sampling method as **S**tep-w**i**se **T**riple-**Co**nsistent Sa**m**pling (SITCOM). We evaluate our method on several image restoration tasks: Super Resolution, Box In-painting, Random In-painting, Motion Deblurring, Gaussian Deblurring, Non-linear Deblurring, High Dynamic Range, and Phase Retrieval. Compared to leading baselines, our approach consistently achieves either state-of-the-art or highly competitive quantitative results, while also reducing the number of sampling steps and, consequently, the computational time. See Figure 8.1 for examples.

8.2 Background: Diffusion Models & Their Usage in Solving IPs

Pre-trained Diffusion Models (DMs) generate images by applying a pre-defined iterative denoising process (Ho et al., 2020). In the Variance-Preserving Stochastic Differentiable Equations (SDEs) setting (Song et al., 2021b,a), DMs are formulated using the forward and reverse processes

$$d\mathbf{x}_t = -\frac{\beta_t}{2}\mathbf{x}_t dt + \sqrt{\beta_t} d\mathbf{w}\,, \qquad d\mathbf{x}_t = -\beta_t\left[\frac{1}{2}\mathbf{x}_t + \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\right]dt + \sqrt{\beta_t} d\bar{\mathbf{w}}\,, \tag{8.1}$$

where $\beta : \{0, \dots, T\} \to (0, 1)$ is a pre-defined function that controls the amount of additive perturbations at time t, $\mathbf{w}$ (resp. $\bar{\mathbf{w}}$) is the forward (resp. reverse) Weiner process (Anderson, 1982),

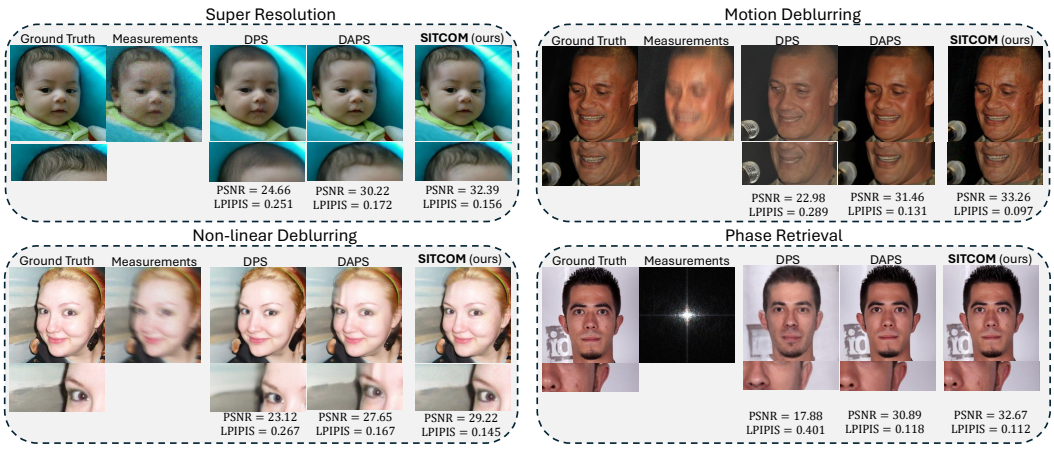


Figure 8.1 **Qualitative results on the FFHQ dataset** on two linear tasks (*top*) and two non-linear tasks (*bottom*) under measurement noise of $\sigma_{\mathbf{y}} = 0.05$. The PSNR and LPIPS values are given below each restored image. Zoomed-in regions show how SITCOM captures greater image details when compared to two general (non)linear DM-based methods (DPS (Chung et al., 2023b) and DAPS (Zhang et al., 2024a)).

$p_t(\mathbf{x}_t)$ is the distribution of $\mathbf{x}_t$ at t, and $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is the score function that is replaced by a neural network (typically a time-encoded U-Net (Ronneberger et al., 2015a)) $\boldsymbol{s} : \mathbb{R}^n \times \{0, \dots, T\} \to \mathbb{R}^n$, parameterized by θ. In practice, given the score function $\boldsymbol{s}_\theta$, the SDEs in (8.1) can be discretized as in (8.2) where $\boldsymbol{\eta}_t, \boldsymbol{\eta}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\eta}_{t-1}\,, \qquad \mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\Big[\mathbf{x}_t + \beta_t \boldsymbol{s}_\theta(\mathbf{x}_t, t)\Big] + \sqrt{\beta_t}\boldsymbol{\eta}_t\,. \tag{8.2}$$

When employed to solve inverse problems, the score function in (8.1) is replaced by a conditional score function which, by Bayes' rule, is $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$. Solving the SDE in (8.1) with the conditional score is referred to as *posterior sampling* (Chung et al., 2023b). As there doesn't exist a closed-form expression for the term $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$ (which is termed as the measurements matching term in (Daras et al., 2024)), previous works have explored different approaches, which we will briefly discuss below. We refer the reader to the recent survey in (Daras et al., 2024) for an overview on DM-based methods for solving IPs.

A well-known method is Diffusion Posterior Sampling (DPS) (Chung et al., 2023b), which uses the approximation $p(\mathbf{y}|\mathbf{x}_t) \approx p(\mathbf{y}|\hat{\mathbf{x}}_0)$ where $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ (or simply $\hat{\mathbf{x}}_0$) is the estimated image at time t as a function of the pre-trained model and $\mathbf{x}_t$ (Tweedie's formula (Vincent, 2011)), given as

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left[\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right] =: f(\mathbf{x}_t; t, \boldsymbol{\epsilon}_\theta)\,, \tag{8.3}$$

where $\bar{\alpha}_t = \prod_{j=1}^{t} \alpha_j$ and $\alpha_t = 1 - \beta_t$. We call the function f, defined in (8.3), as **'Tweedie-network denoiser'** (also termed as 'posterior mean predictor' in (Chen et al., 2024)). Here, $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = -\sqrt{1-\bar{\alpha}_t}\boldsymbol{s}_\theta(\mathbf{x}_t, t)$ (Luo, 2022) outputs the noise in $\mathbf{x}_t$. Tweedie's formula, like in our method, is also adopted in other DM-based IP solvers such as (Rout et al., 2023; Chung et al., 2023d; Wang et al., 2022). The drawback of these methods is that they require a large number of sampling steps.

The work in ReSample (Song et al., 2023a), solves an optimization problem on the estimated posterior mean in the latent space for many steps to enforce measurement consistency, requiring many sampling and optimization steps.

The work in (Mardani et al., 2023) introduced RED-Diff, a variational Bayesian method that fits a Gaussian distribution to the posterior distribution of the clean image conditional on the measurements. This approach involves solving an optimization problem using stochastic gradient descent (SGD) to minimize a data-fitting term while maximizing the likelihood of the reconstructed image under the denoising diffusion prior (as a regularizer). However, the SGD process requires multiple iterations, each involving evaluations of the pre-trained DM on a different noisy image at some randomly selected time, making it quite computationally expensive.

Recently, Decoupling Consistency with Diffusion Purification (DCDP) (Li et al., 2024) proposed separating diffusion sampling steps from measurement consistency by using DMs as diffusion purifiers (Nie et al., 2022a; Alkhouri et al., 2024), with the goal of reducing the run-time. However, DCDP requires tuning the number of forward diffusion steps for purification. Shortly after, Decoupled Annealing Posterior Sampling (DAPS) (Zhang et al., 2024a) introduced another decoupled approach, incorporating gradient descent noise annealing via Langevin dynamics. DAPS, similar to DPS and RED-Diff, also requires a large number of sampling and optimization steps. Under measurement noise, DCDP achieves SOTA run-time across various linear restoration tasks, while DAPS sets the SOTA in restoration quality. Both will serve as primary baselines in our experiments.

8.3 SITCOM: Step-wise Triple-Consistent Sampling

8.3.1 Motivation: Addressing the Challenges in Applying DMs to IPs

Most inverse problems are ill-conditioned and undersampled. DMs, when trained on a dataset that closely resembles the target image, can provide critical information to alleviate ill-conditioning and improve recovery. Despite various previous efforts, a key challenge remains: How to *efficiently* integrate DMs into the framework of inverse problems? We will now elaborate on this challenge in detail.

The standard reverse sampling procedure in DMs consists of applying the backward discrete steps in (8.2) for $t \in \{T, T-1, \dots, 1\}$, forming the standard diffusion trajectory for which $\mathbf{x}_0$ is the generated image. To incorporate the measurement $\mathbf{y}$ into these steps, a common approach adopted in previous works that demonstrate superior performance (e.g., (Song et al., 2023a; Zhang et al., 2024a; Li et al., 2024)) is to the $\hat{\mathbf{x}}_0$ computed via (8.3) as follows:

$$\hat{\mathbf{x}}_0'(\mathbf{x}_t) = \arg\min_{\mathbf{x}} \|\mathcal{A}(\mathbf{x}) - \mathbf{y}\|^2 + \lambda \|\mathbf{x} - \hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2 \,, \tag{8.4}$$

where $\lambda \in \mathbb{R}_+$ is a regularization parameter. The $\hat{\mathbf{x}}_0'(\mathbf{x}_t)$ obtained from (8.4) is close to $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ while also remaining consistent with the measurements. When using $\hat{\mathbf{x}}_0'(\mathbf{x}_t)$ to sample $\mathbf{x}_{t-1}$, the second formula in (8.2) can be rewritten as in (8.5), where the derivation is provided in Appendix D.1.

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t) + \sqrt{\beta_t}\boldsymbol{\eta}_t \,. \tag{8.5}$$

By substituting $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ into (8.5) with the measurement-consistent $\hat{\mathbf{x}}_0'(\mathbf{x}_t)$, the modified sampling formula becomes:

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0'(\mathbf{x}_t) + \sqrt{\beta_t}\boldsymbol{\eta}_t \,. \tag{8.6}$$

While this approach effectively ensures data consistency at each step, it inevitably causes $\hat{\mathbf{x}}_0'$ to deviate from the diffusion trajectory, leading to two major issues:

(I1) The image $\hat{\mathbf{x}}_0(\mathbf{x}_t)$, initially constructed through Tweedie's formula, usually appears quite natural (e.g., columns 3 to 5 of Figure 8.2); however, the modified version, $\hat{\mathbf{x}}_0'(\mathbf{x}_t)$, is likely to exhibit severe artifacts (e.g., columns 6 to 8 of Figure 8.2).

(I2) Since the DM network, ϵ_θ, is trained via minimizing the objective function $\mathbb{E}_{\mathbf{x}_0,\epsilon}\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2$ (denoising score matching (Vincent, 2011)) on a finite dataset, it performs best on noisy images lying in the high-density regions of the training distribution $\mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. We define an algorithm as **forward-consistent** if it likely applies ϵ_θ only to in-distribution inputs (i.e., those from the same distribution used for training). For example, if the forward diffusion used to train ϵ_θ adds Gaussian noise, the in-distribution input to ϵ_θ should ideally be sampled from a Gaussian with specific parameters. If Poisson noise is used in the forward process, inputs drawn from suitable Poisson distributions are more likely to fall within the well-trained region of the network. In summary, forward consistency requires that inputs to ϵ_θ during sampling align with the forward process. While the $\mathbf{x}_{t-1}$ generated from (8.5) is forward-consistent by design, the one generated from the modified formula (8.6) is not. Therefore, in the latter case, the DM network, ϵ_θ, may be applied to many out-of-distribution inputs, leading to degraded performance.

We pause to verify our claimed Issue **(I1)** through a box-inpainting experiment. Columns 3 to 5 of Figure 8.2 show $\hat{\mathbf{x}}_0'(\mathbf{x}_t)$ at various t. The results clearly demonstrate successful enforcement of data consistency, as the region outside the box aligns with the original image. However, this enforcement compromises the natural appearance of the image, introducing significant artifacts in the reconstructed area inside the box. Details about the setting of the results in Figure 8.2 are given in Section D.3.

Issue **(I2)** was previously observed in (Lugmayr et al., 2022), which proposed a remedy known as '*resampling*'. In this approach, the sampling formula in (8.6) is replaced by

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\eta}_t \,. \tag{8.7}$$

Provided $\hat{\mathbf{x}}_0$ is close to the ground truth $\mathbf{x}_0$, $\mathbf{x}_{t-1}$ generated this way will stay in-distribution with high probability. For a more detailed explanation of the rationale behind this remedy, we refer the reader to (Lugmayr et al., 2022). This method has since been adopted by subsequent works, such as (Song et al., 2023a; Zhang et al., 2024a), and we will also employ it to address **(I2)**.

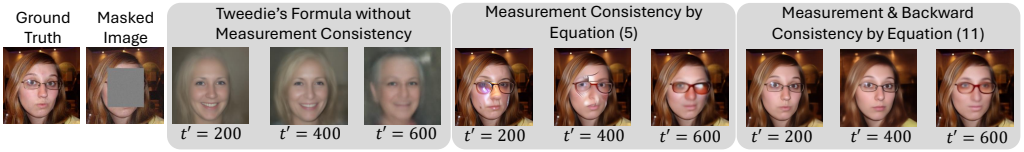


Figure 8.2 Effects of enforcing backward-consistency in box-inpainting: Results of using Tweedie's formula without measurement consistency (columns 3 to 5), enforcing measurement-consistency via (8.4) (columns 6 to 9), and enforcing both measurement-consistency and backward-consistency via (8.11) (columns 10 to 12) at different time steps t'. Experimental details are given in Appendix D.3.

8.3.2 Network Regularization & Backward Diffusion Consistency

Previous studies, such as (Song et al., 2023a; Zhang et al., 2024a), mitigate issue **(I1)** by using a large number of sampling steps, which inevitably increases the computational burden. In contrast, this paper proposes employing a network regularization to resolve issue **(I1)**. This approach not only accelerates convergence but also enhances reconstruction quality. Let's first clarify the underlying intuition.

It is widely observed that the U-Net architecture or trained transformers exhibit an effective image bias (Ulyanov et al., 2018; Liang et al., 2024a; Ghosh et al., 2024; Hatamizadeh et al., 2024). From columns 3 to 5 of Figure 8.2, we observe that without enforcing data consistency, the reconstructed $\hat{\mathbf{x}}_0$, derived directly from Tweedie-network denoiser $f(\mathbf{x}_t; t, \boldsymbol{\epsilon}_\theta)$ for each time t, exhibits natural textures. This indicates that the reconstruction using the combination of Tweedie's formula and the DM network has a natural regularizing effect on the image. By definition, the output of $f(\mathbf{x}_t; t, \boldsymbol{\epsilon}_\theta)$ in (8.3) represents the denoised version of $\mathbf{x}_t$ at time t using the Tweedie's formula and the DM denoiser $\boldsymbol{\epsilon}_\theta$. Due to the implicit bias of $\boldsymbol{\epsilon}_\theta$, this denoised image tends to align with the clean image manifold, even if $\mathbf{x}_t$ does not correspond to a training image, as shown in columns 3 to 5 of Figure 8.2. We refer to this regularization effect of $f(\mathbf{x}_t; t, \boldsymbol{\epsilon}_\theta)$, which arises from network bias, as "network regularization".

By employing network regularization, we can address **(I1)** by ensuring that the data-consistent $\hat{\mathbf{x}}'_0$ is also network-consistent. We refer to the latter condition as **Backward Consistency** and define it formally as follows.

Definition 1 (Backward Consistency). We say an $\hat{\mathbf{x}}'_0$ is backward-consistent with Tweedie's formula and the DM neural network $\boldsymbol{\epsilon}_\theta$ at time t if there exists some $\mathbf{v}_t$ such that $\hat{\mathbf{x}}'_0 = f(\mathbf{v}_t; t, \boldsymbol{\epsilon}_\theta)$. In other

words, backward consistency requires $\hat{\mathbf{x}}'_0$ to be a 'denoised version' of some noisy image $\mathbf{v}_t$ via the Tweedie-network denoiser f at time t.

The subset of images that are in the range of the function f (i.e., backward-consistent) is denoted by C_t and defined as

$$C_t := \{f(\mathbf{v}_t; t, \boldsymbol{\epsilon}_\theta) : \mathbf{v}_t \in \mathbb{R}^n\} . \tag{8.8}$$

Enforcing $\hat{\mathbf{x}}'_0$ to be both measurement- and backward-consistent involves solving the following optimization problem.

$$\hat{\mathbf{x}}'_0, \hat{\mathbf{v}}_t := \arg\min_{\mathbf{v}'_t, \mathbf{x}'_0} \left\{ \|\mathcal{A}(\mathbf{x}'_0) - \mathbf{y}\|_2^2 \quad \text{subject to} \quad \mathbf{x}'_0 = f(\mathbf{v}'_t; t, \boldsymbol{\epsilon}_\theta) \right\} . \tag{8.9}$$

However, (8.9) may violate forward consistency, as $\hat{\mathbf{v}}_t$ could possibly be far from $\mathbf{x}_t$. Therefore, we propose adding a regularization term, for which (8.9) becomes

$$\hat{\mathbf{x}}'_0, \hat{\mathbf{v}}_t := \arg\min_{\mathbf{v}'_t, \mathbf{x}'_0} \left\{ \|\mathcal{A}(\mathbf{x}'_0) - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}_t - \mathbf{v}'_t\|_2^2 \quad \text{subject to} \quad \mathbf{x}'_0 = f(\mathbf{v}'_t; t, \boldsymbol{\epsilon}_\theta) \right\} . \tag{8.10}$$

During the reverse sampling process, at each time t, with the given $\mathbf{x}_t$, we seek a $\mathbf{v}'_t$ in the nearby region (i.e., $\|\mathbf{x}_t - \mathbf{v}'_t\|$ is small), such that $\mathbf{v}'_t$ can be denoised by f to produce a clean image $\mathbf{x}'_0$ (i.e., $\mathbf{x}'_0 = f(\mathbf{v}'_t; t, \boldsymbol{\epsilon}_\theta)$), which is also consistent with the measurements $\mathbf{y}$ (i.e., $\|\mathcal{A}(\mathbf{x}'_0) - \mathbf{y}\|_2^2$ is small). We need to identify such a $\mathbf{v}'_t$ because $\mathbf{x}_t$ itself cannot be directly denoised by f to yield an image consistent with the measurements. By substituting the constraint into the objective function, the optimization problem in (8.10) is reduced to

$$\hat{\mathbf{v}}_t := \arg\min_{\mathbf{v}'_t} \left\{ \|\mathcal{A}(f(\mathbf{v}'_t; t, \boldsymbol{\epsilon}_\theta)) - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}_t - \mathbf{v}'_t\|_2^2 \right\}, \quad \hat{\mathbf{x}}'_0 = f(\hat{\mathbf{v}}_t; t, \boldsymbol{\epsilon}_\theta). \tag{8.11}$$

The benefit of the considered backward consistency constraint is shown in columns 6 to 8 of Figure 8.2. After obtaining $\hat{\mathbf{x}}'_0$, the resampling formula in (8.7) is used to obtain $\mathbf{x}_{t-1}$.

8.3.3 Triple Consistency Conditions

We now summarize the three key conditions that apply at each sampling step.

(C1) **Measurement Consistency**: The reconstruction $\hat{\mathbf{x}}'_0$ is consistent with the measurements This means that $\mathcal{A}(\hat{\mathbf{x}}'_0) \approx \mathbf{y}$.

C2 Backward Consistency: The reconstruction $\hat{\mathbf{x}}_0'$ is a denoised image produced by the Tweedie-network denoiser f. More generally, we define the backward consistency to include any form of DM network regularization (e.g., using the DM probability-flow (PF) ODE (Karras et al., 2022)) applied to $\hat{\mathbf{x}}_0'$.

C3 Forward Consistency: The pre-trained DM network $\boldsymbol{\epsilon}_\theta$ is provided with in-distribution inputs with high probability. To ensure this, we apply the resampling formula in (8.7) and enforce that $\hat{\mathbf{v}}_t$ remains close to $\mathbf{x}_t$.

We emphasize that **C1**-**C3** aim to ensure that all intermediate reconstructions $\hat{\mathbf{x}}_0'(\mathbf{x}_t)$ (with $t > 0$) are as accurate as possible, allowing us to effectively reduce the number of sampling steps. If reducing sampling steps is not necessary, these conditions become less critical, as the final reconstruction at $t = 0$ can still be accurate with a large number of sampling steps, even if the intermediate reconstructions are less precise. Previous works, such as (Song et al., 2023a; Zhang et al., 2024a), enforce measurement consistency by applying $\mathcal{A}(\hat{\mathbf{x}}_0) = \mathbf{y}$ exactly, whereas DPS (Chung et al., 2023b) does not ensure consistency along the diffusion trajectory.

8.3.4 The Proposed Sampler

Given $\mathbf{x}_t$, $\boldsymbol{\epsilon}_\theta$, and towards satisfying the above conditions, our method, at sampling time t, consists of the following three steps:

$$\hat{\mathbf{v}}_t := \arg\min_{\mathbf{v}_t'} \; \|\mathcal{A}\big(\tfrac{1}{\sqrt{\bar{\alpha}_t}}\big[\mathbf{v}_t' - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\mathbf{v}_t', t)\big]\big) - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}_t - \mathbf{v}_t'\|_2^2 \tag{S$_1$}$$

$$\hat{\mathbf{x}}_0' = f(\hat{\mathbf{v}}_t; t, \boldsymbol{\epsilon}_\theta) \equiv \tfrac{1}{\sqrt{\bar{\alpha}_t}}\big[\hat{\mathbf{v}}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\hat{\mathbf{v}}_t, t)\big] \tag{S$_2$}$$

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0' + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\eta}_t\,, \quad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,. \tag{S$_3$}$$

The minimization in the first step optimizes over the input $\mathbf{v}_t'$ of the pre-trained diffusion model at time t, where the first term of the objective enforces measurement consistency for the posterior mean estimated image, satisfying condition **C1**. The second term serves as a regularization term, implicitly

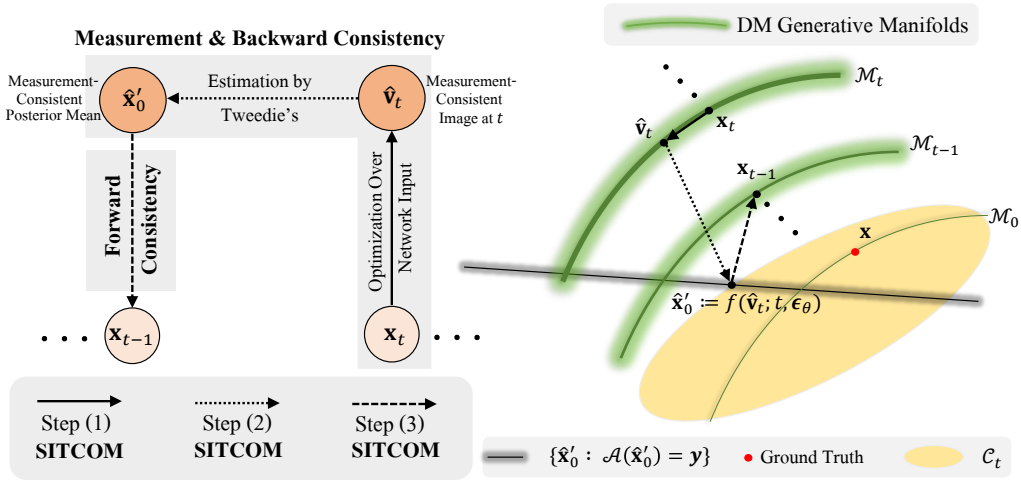


Figure 8.3 Illustrative diagram of the proposed procedure in SITCOM (*left*). Conceptual illustration of SITCOM, where $\mathcal{M}_t$ is the DM generative manifold at time t and C_t is the subset of images that are backward-consistent, defined in (8.8) (*right*). Step (1) (solid arrow), Step (2) (dotted arrow), and Step (3) (dashed arrow) correspond to ($\mathtt{S}_1$), ($\mathtt{S}_2$), and ($\mathtt{S}_3$), respectively.

promoting closeness between $\hat{\mathbf{v}}_t$ and $\mathbf{x}_t$ (i.e., condition **C3**), with $\lambda > 0$ acting as the regularization parameter. The argument of the forward operator in ($\mathtt{S}_1$) and the second step in ($\mathtt{S}_2$) enforce that $\hat{\mathbf{v}}_t$ and $\hat{\mathbf{x}}_0'$, respectively, maintain the diffusion trajectory through obeying Tweedie's formula, thereby satisfying the backward consistency condition, **C2**. After obtaining the measurement-consistent estimate, $\hat{\mathbf{x}}_0'$, as given in ($\mathtt{S}_2$), it must be mapped back to time $t-1$ to generate $\mathbf{x}_{t-1}$. This is achieved through the forward diffusion step in ($\mathtt{S}_3$) as outlined in the forward consistency condition, **C3**. A diagram of SITCOM procedure is provided in Figure 8.3 (*left*).

Remark 5. *Obtaining the estimated image at time* 0 *given some* $\mathbf{x}_t$ *using the standard DM PF-ODE (Karras et al., 2022) is more accurate compared to the one-step Tweedie's formula. However, since PF-ODE is an iterative procedure, it requires more computational time. In SITCOM, PF-ODE could replace Tweedie's formula in* ($\mathtt{S}_2$). *Nevertheless, we chose not to use it, as this would increase the run time, and our empirical results are already highly competitive using Tweedie's formula.*

A conceptual illustration of SITCOM is shown in Figure 8.3 (*right*).

The DM generative manifold, $\mathcal{M}_t$, is defined as the set of all $\mathbf{x}_t$ sampled from $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, and $\mathbf{x}_0 \sim p_0(\mathbf{x})$. This set coincides with the entire space $\mathbb{R}^n$ equipped with the probability measure induced by the distribution of $\mathbf{x}_t$, which we denote as $\mathcal{P}_t$. In Figure 8.3 (*right*), the variation of color around each $\mathcal{M}_t$ indicates the concentration of the measure $\mathcal{P}_t$, with

darker colors representing higher concentration.

SITCOM's Step (1) and Step (2) enforce measurement consistency and backward consistency, thus map $\mathbf{x}_t$ to $\hat{\mathbf{x}}_0' = f(\hat{\mathbf{v}}_t; t, \boldsymbol{\epsilon}_\theta)$ which lies within the intersection of (i) measurement-consistent set $\{\hat{\mathbf{x}}_0' : \mathcal{A}(\hat{\mathbf{x}}_0') \approx \mathbf{y}\}$ (the shaded black line) and (ii) the backward-consistent set C_t (the yellow ellipsoid) defined in (8.8). Subsequently, $\mathbf{x}_{t-1}$ is generated by inserting $\hat{\mathbf{x}}_0'$ into the resampling formula, which enforces the forward consistency.

Handling Measurement Noise: To avoid the case where the first term of the objective in $(\mathtt{S}_1)$ reaches small values yielding noise overfitting (i.e., when additive Gaussian noise is considered, $\sigma_{\mathbf{y}} > 0$), we propose refraining from enforcing strict measurement fitting $\mathcal{A}(\mathbf{x}) = \mathbf{y}$. Instead, we use the stopping criterion $\left\|\mathcal{A}\left(\frac{1}{\sqrt{\bar{\alpha}_t}}\left[\mathbf{v}_t' - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\mathbf{v}_t', t)\right]\right) - \mathbf{y}\right\|_2^2 < \delta^2$,

where $\delta \in \mathbb{R}_+$ is a hyper-parameter that indicates the level of tolerance for noise and helps prevent overfitting. This is equivalent to enforcing an ℓ_2 constraint, and is in spirit similar to (Wang et al., 2024). Since the noise level cannot be accurately estimated, in our experiments, we use δ that is slightly larger than the actual level of noise in the measurements, i.e., $\delta > \sigma_{\mathbf{y}}\sqrt{m}$.

8.3.5 SITCOM with Arbitrary Stepsizes

In this subsection, we explain how to apply SITCOM with a large stepsize and present the final algorithm. The pre-trained DM is trained with T diffusion steps. Given that our method is designed to satisfy measurement and diffusion consistency, SITCOM requires $N \ll T$ sampling iterations, using a step size of $\Delta t := \lfloor \frac{T}{N} \rfloor$. Thus, we introduce the index i instead of t with a relation $t = i\Delta t$.

The procedure of SITCOM is outlined in Algorithm 8.1. As inputs, SITCOM takes $\mathbf{y}$, $\mathcal{A}(\cdot)$, $\boldsymbol{\epsilon}_\theta$, the number of sampling steps N, $\bar{\alpha}_i$ for all $i \in \{1, \ldots, N\}$, the number of optimization steps K per sampling step, stopping criteria δ, and the learning rate γ.

Starting with initializing $\mathbf{v}_i^{(0)}$ as $\mathbf{x}_i$ (satisfying condition **C3**), lines 3 through 6 correspond to the first step of SITCOM, where $(\mathtt{S}_1)$ is solved via either gradient descent (as shown in the algorithm), or the ADAM optimizer (Kingma and Ba, 2015b). In lines 5 and 6, the stopping criterion is applied to prevent strict data fidelity (avoiding noise overfitting). Following the gradient updates in the inner loop, $\hat{\mathbf{v}}_i$ is obtained in line 7, which is then used in line 8 to obtain $\hat{\mathbf{x}}_0'$ as specified in $(\mathtt{S}_2)$,

Algorithm 8.1 **S**tep-wi**s**e **T**riple-**Co**nsistent Sa**m**pling (**SITCOM**).

Input: Measurements $\mathbf{y}$, forward operator $\mathcal{A}(\cdot)$, pre-trained DM $\boldsymbol{\epsilon}_\theta(\cdot\,,\cdot)$, number of diffusion steps N, DM noise schedule $\bar{\alpha}_i$ for $i \in \{1,\dots,N\}$, number of gradient updates K, stopping criterion δ, learning rate γ, and regularization parameter λ.

Output: Restored image $\hat{\mathbf{x}}$.

Initialization: $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, $\Delta t = \lfloor \frac{T}{N} \rfloor$

1: **For each** $i \in \{N, N-1,\dots,1\}$. (Reducing diffusion sampling steps)

2: **Initialize** $\mathbf{v}_i^{(0)} \leftarrow \mathbf{x}_i$. (Initialization to ensure Closeness: **C3**)

3: **For each** $k \in \{1,\dots,K\}$. (Gradient updates for measurement & backward consistency: **C1**, **C2**)

4: $\mathbf{v}_i^{(k)} = \mathbf{v}_i^{(k-1)} - \gamma \nabla_{\mathbf{v}_i} \Big[\Big\| \mathcal{A}\Big(\frac{1}{\sqrt{\bar{\alpha}_i}} \big[\mathbf{v}_i - \sqrt{1-\bar{\alpha}_i}\, \boldsymbol{\epsilon}_\theta(\mathbf{v}_i, i\Delta t) \big] \Big) - \mathbf{y} \Big\|_2^2 + \lambda \|\mathbf{x}_i - \mathbf{v}_i\|_2^2 \Big] \Big|_{\mathbf{v}_i = \mathbf{v}_i^{(k-1)}}$.

5: **If** $\big\| \mathcal{A}\big(\frac{1}{\sqrt{\bar{\alpha}_i}} \big[\mathbf{v}_i^{(k)} - \sqrt{1-\bar{\alpha}_i}\, \boldsymbol{\epsilon}_\theta(\mathbf{v}_i^{(k)}, i\Delta t) \big] \big) - \mathbf{y} \big\|_2^2 < \delta^2$. (Stopping criterion)

6: **Break** the **For** loop in step 3. (Preventing noise overfitting)

7: **Assign** $\hat{\mathbf{v}}_i \leftarrow \mathbf{v}_i^{(k)}$. (Backward diffusion consistency of $\hat{\mathbf{v}}_i$: **C2**)

8: **Obtain** $\hat{\mathbf{x}}_0' = f(\hat{\mathbf{v}}_i; t, \theta) = \frac{1}{\sqrt{\bar{\alpha}_i}} \big[\hat{\mathbf{v}}_i - \sqrt{1-\bar{\alpha}_i}\, \boldsymbol{\epsilon}_\theta(\hat{\mathbf{v}}_i, i\Delta t) \big]$. (Backward consistency of $\hat{\mathbf{x}}_0'$: **C2**)

9: **Obtain** $\mathbf{x}_{i-1} = \sqrt{\bar{\alpha}_{i-1}}\hat{\mathbf{x}}_0' + \sqrt{1-\bar{\alpha}_{i-1}}\boldsymbol{\eta}_i,\ \boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. (Forward diffusion consistency: **C3**)

10: **Restored image:** $\hat{\mathbf{x}} = \mathbf{x}_0$.

satisfying condition **C2**. Note that line 8 requires no additional computation, as the $\hat{\mathbf{x}}_0'$ calculated here was already obtained while checking the stopping condition in line 6. After obtaining the double-consistent $\hat{\mathbf{x}}_0'$, the resampling is applied to map the image back to time $t-1$ while ensuring $\mathbf{x}_{t-1}$ to be in-distribution, as indicated in line 9 of the algorithm. In the next iteration, the requirement that $\hat{\mathbf{v}}_{t-1}$ is close to $\mathbf{x}_{t-1}$ ensures that the input $\hat{\mathbf{v}}_{t-1}$ to the DM network, $\boldsymbol{\epsilon}_\theta$, is also in-distribution, thus satisfying the forward-consistency (condition **C3**).

The computational requirements of SITCOM are determined by (*i*) the number of sampling steps N and (*ii*) the number of gradient steps K required for each sampling iteration. Given the proposed stopping criterion, this results in at most NK Number of Function Evaluations (NFEs) of the pre-trained model (forward pass), NK backward passes through the pre-trained model, and NK applications each for the forward operator and its adjoint to solve the optimization problem in ($\mathtt{S}_1$). With early stopping, the computational cost is lower. For example, for a linear operator $\mathcal{A}$ with dimensions $m \times n$, the cost of applying it (or its adjoint) to a vector is $O(mn)$. For a network with width M and depth L, the cost for making a forward pass is $O(LM^2)$. The gradients are computed w.r.t. the input of the DM network, requiring an additional backward pass. This backward pass has

the same computational cost as the forward pass. Consequently, this procedure is significantly more efficient than network training, where the network weights are updated instead of the input.

8.3.6 Relation with Existing Approaches

While SITCOM and DPS (Chung et al., 2023b) both use Tweedie's formula, there are two major differences. First, DPS does not enforce backward consistency. Specifically, it only considers one gradient descent step of the optimization in ($\mathtt{S}_1$), whereas our method perform multiple steps, initializing with $\mathbf{x}_t$. Second, DPS does not enforce the forward diffusion consistency, namely, it does not use resampling ($\mathtt{S}_3$). This means that DPS does not enforce a step-wise **C1**-**C3**.

Both SITCOM and the works in (Song et al., 2023a; Zhang et al., 2024a) are optimization-based methods that modify the sampling steps to enforce measurement consistency, and both involve mapping back to time $t-1$ (as in step 3 of SITCOM). However, there is a major difference between them: The optimization variable in these works is the estimated image at time t (the output of the DM network), whereas in SITCOM, it is the noisy image at time t (the input of the network). This means that these studies enforce **C1** and **C3**, but not **C2**.

8.4 Experimental Results

Tasks: Our experimental setup for IPs and noise levels used largely follows DPS (Chung et al., 2023b). For linear IPs, we evaluate five tasks: super resolution, Gaussian deblurring, motion deblurring, box inpainting, and random inpainting. For Gaussian deblurring and motion deblurring, we use 61×61 kernels with standard deviations of 3 and 0.5, respectively. In the super-resolution task, a bicubic resizer downscales images by a factor of 4. For box inpainting, a random 128×128 box is applied to mask image pixels, and for random inpainting, the mask is generated with each pixel masked with a probability of 0.7, as described in (Song et al., 2023a). For nonlinear IP tasks, we consider three tasks: phase retrieval, high dynamic range (HDR) reconstruction, and nonlinear (non-uniform) deblurring. For phase retrieval, an oversampling rate of 2 is applied in frequency domain, and we report the best result out of four independent samples, consistent with (Chung et al., 2023b; Zhang et al., 2024a) (see Appendix D.4 for more discussion on phase retrieval). In HDR reconstruction, the goal is to restore a higher dynamic range image from a lower dynamic

range image (with a factor of 2). Nonlinear deblurring follows the setup in (Tran et al., 2021). For measurement noise, we use $\sigma_{\mathbf{y}} \in \{0.01, 0.05\}$ for all tasks.

Baselines & Datasets: For baselines, in this section, we use DPS (Chung et al., 2023b), DDNM (Wang et al., 2022), DCDP (Li et al., 2024), and DAPS (Zhang et al., 2024a). The selection criteria is based on these baselines' competitive performance on several linear and non-linear inverse problems under measurement noise. Additionally, we provide comparison results with three other baselines in Table D.2 of Appendix D.5. We evaluate SITCOM and baselines using 100 test images from the validation set of FFHQ (Karras et al., 2019) and 100 test images from the validation set of ImageNet (Deng et al., 2009) for which the FFHQ-trained and ImageNet-trained DMs are given in (Chung et al., 2023b) and (Dhariwal and Nichol, 2021), respectively, following the previous convention. For evaluation metrics, we use PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018).

SITCOM Settings: For Algorithm 8.1, we set $N = 20$ and $K = 30$ for most tasks. We show the impact of N and K in Appendix D.6.1. The parameter λ is set to 0 for all tasks other than phase retrieval where we use $\lambda = 1$, following the ablation study in Appendix D.6.2. The impact of the stopping criterion under the noisy setting is given in Appendix D.6.3. The learning rate for $(\mathtt{S}_1)$ is set to $\gamma = 0.01$ across all measurements noise levels, datasets, and tasks. Table D.5 in Appendix D.6.4 lists all the hyper-parameters used for every task. We note that the exact set of hyper-parameters is used for the FFHQ and ImageNet datasets. Our code is available online.

Main Results: In Table 8.1, we present the quantitative results in terms of the average PSNR, SSIM, LPIPS, and run-time (minutes). Columns 3 to 6 correspond to the FFHQ dataset, while columns 7 to 10 reflect results for the ImageNet dataset. The table covers 8 tasks, 4 evaluation metrics, and 2 datasets, totaling 64 results. Among these, SITCOM reports the best performance in 58 out of 64 cases.

On average, SITCOM demonstrates strong reconstruction capabilities across most tasks. For the FFHQ dataset, SITCOM reports a PSNR improvement of over 1 dB in Super Resolution, random In-painting, and Gaussian Deblurring compared to the second-best method. On ImageNet, we

Task	Method	FFHQ				ImageNet			
		PSNR (↑)	SSIM (↑)	LPIPS (↓)	Run-time (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	Run-time (↓)
Super Resolution 4×	DPS	24.44±0.56	0.801±0.032	0.26±0.022	1.26±0.52	23.86±0.34	0.76±0.041	0.357±0.069	2.38±1.02
	DAPS	29.24±0.42	0.851±0.024	**0.135**±0.039	1.24 ±0.22	25.67±0.73	0.802±0.045	0.256±0.067	2.16±0.45
	DDNM	28.02±0.78	0.842±0.034	0.197±0.034	1.07±0.42	23.96±0.89	0.767±0.045	0.475±0.044	1.27±0.55
	DCDP	27.88±1.34	0.825±0.07	0.211±0.05	0.52±0.34	24.12±1.24	0.772±0.000	0.351±0.00	1.45±0.00
	SITCOM (ours)	**30.68**±1.02	**0.867**±0.045	0.142±0.056	**0.45**±0.58	**26.35**±1.21	**0.812**±0.021	**0.232**±0.038	**1.12**±0.52
Box In-Painting	DPS	23.20±0.89	0.754±0.023	0.196±0.032	1.57±0.55	19.78±0.78	0.691±0.052	0.312±0.025	2.28 ±1.02
	DAPS	24.17±1.02	0.787±0.032	0.135±0.032	1.35±0.45	21.43±0.40	0.736±0.020	0.218±0.021	2.54±1.02
	DDNM	24.37±0.45	0.792±0.024	0.232±0.026	1.02±0.032	21.64±0.66	0.732±0.028	0.319±0.015	1.45±1.02
	DCDP	23.66±1.67	0.762±0.07	0.144±0.05	0.56±0.25	20.45±1.22	0.712±0.07	0.298±0.04	1.127±0.25
	SITCOM (ours)	**24.68**±0.78	**0.801**±0.042	**0.121**±0.08	**0.35**±0.25	**21.88**±0.92	**0.742**±0.032	**0.214**±0.021	**1.12**±0.35
Random In-Painting	DPS	28.39±0.82	0.844±0.042	0.194±0.021	1.52±0.30	24.26±0.42	0.772±0.02	0.326±0.034	2.27±0.25
	DAPS	31.02±0.45	0.902±0.015	0.098±0.017	1.56±0.40	28.44±0.45	0.872±0.024	0.135±0.052	2.14±0.45
	DDNM	29.93±0.67	0.889±0.032	0.122±0.056	1.45±0.35	29.22±0.55	0.912±0.034	0.191±0.048	1.54±0.52
	DCDP	28.59±0.95	0.852±0.06	0.202±0.04	0.55±0.25	26.22±1.13	0.791±0.06	0.289±0.03	1.44±0.34
	SITCOM (ours)	**32.05**±1.02	**0.909**±0.09	**0.095**±0.025	**0.45**±0.50	**29.60**±0.78	**0.915**±0.028	**0.127**±0.039	**1.14**±0.45
Gaussian Deblurring	DPS	25.52±0.78	0.826±0.052	0.211±0.017	1.50±0.50	21.86±0.45	0.772±0.08	0.362±0.034	2.55±0.45
	DAPS	29.22±0.50	0.884±0.056	0.164±0.032	1.40±0.52	26.12±0.78	0.832±0.092	0.245±0.022	2.23±0.52
	DDNM	28.22±0.52	0.867±0.056	0.216±0.042	1.56±0.45	**28.06**±0.52	**0.879**±0.072	0.278±0.089	1.75±0.63
	DCDP	26.67±0.78	0.835±0.08	0.196±0.04	0.56±0.23	23.24±1.18	0.781±0.06	0.343±0.04	1.34±0.43
	SITCOM (ours)	**30.25**±0.89	**0.892**±0.032	**0.135**±0.078	**0.46**±0.25	27.40±0.45	0.854±0.045	**0.236**±0.039	**1.10**±0.42
Motion Deblurring	DPS	23.40±1.42	0.737±0.024	0.270±0.025	2.40±0.55	21.86±2.05	0.724±0.022	0.357±0.032	2.56±0.40
	DAPS	29.66±0.50	0.872±0.027	0.157±0.012	1.86±0.12	27.86±1.20	0.862±0.032	0.196±0.021	2.3±0.45
	SITCOM (ours)	**30.34**±0.67	**0.902**±0.037	**0.148**±0.041	**0.5**±0.45	**28.65**±0.34	**0.876**±0.021	**0.189**±0.036	**1.48**±0.35
Phase Retrieval	DPS	17.34±2.67	0.67±0.045	0.41±0.08	1.50±0.34	16.82±1.22	0.64±0.08	0.447±0.032	2.17±0.24
	DAPS	30.67±3.12	0.908±0.041	0.122±0.084	1.34±0.78	**25.76**±2.33	0.797±0.045	0.255±0.095	2.24±0.25
	DCDP	28.52±2.50	0.892±0.19	0.167±0.92	3.30±0.45	24.25±2.25	0.778±0.14	0.287±0.089	3.49±0.52
	SITCOM (ours)	**30.97**±3.10	**0.915**±0.064	**0.112**±0.102	**0.52**±0.34	25.45±2.78	**0.808**±0.065	**0.246**±0.088	**1.40**±0.40
Non-Uniform Deblurring	DPS	23.42±2.15	0.757±0.042	0.279±0.067	1.55±0.44	22.57±0.67	0.778±0.067	0.310±0.102	2.35±0.45
	DAPS	28.23±1.55	0.833±0.052	0.155±0.041	1.42±0.41	27.65±1.2	0.822±0.056	0.169±0.044	2.14±0.45
	DCDP	28.78±1.44	0.827±0.08	0.162±0.04	3.30±0.45	26.56±1.09	0.803±0.06	0.182±0.05	3.70±0.36
	SITCOM (ours)	**30.12**±0.68	**0.902**±0.042	**0.145**±0.037	**0.52**±0.45	**28.78**±0.79	**0.832**±0.056	**0.16**±0.048	**1.25**±0.45
High Dynamic Range	DPS	22.88±1.25	0.722±0.056	0.264±0.089	1.45±0.34	19.33±1.45	0.688±0.067	0.503±0.132	2.42±0.46
	DAPS	27.12±0.89	0.825±0.056	0.166±0.078	1.25±0.35	26.30±1.02	0.792±0.046	0.177±0.089	2.18±0.55
	SITCOM (ours)	**27.98**±1.06	**0.832**±0.052	**0.158**±0.032	**0.52**±0.30	**26.97**±0.87	**0.821**±0.045	**0.167**±0.052	**1.54**±0.35

Table 8.1 Average PSNR, SSIM, LPIPS, and run-time (minutes) of SITCOM and baselines using 100 test images from the **FFHQ** dataset (columns 3 to 7) and 100 test images from the **ImageNet** dataset with a **measurement noise level of** $\sigma_{\mathbf{y}} = 0.05$. The results for the $\sigma_{\mathbf{y}} = 0.01$ case are given in Table D.1 of Appendix D.5. The first five tasks are linear, while the last three tasks are non-linear (underlined). For each task and dataset combination, the best results are bolded, and the second-best results are underlined. Values after ± represent the standard deviation. All results were obtained using a **single RTX5000 GPU** machine. For phase retrieval, the run-time is reported for the best result out of four independent runs. This is applied for SITCOM and baselines. More discussion about phase retrieval is given in Appendix D.4.

observe more than a 1 dB improvement in random In-painting. Other than ImageNet Gaussian Deblurring and ImageNet Phase Retrieval, for which we under-perform by 0.66 dB and 0.31 dB, respectively, our PSNR improvement when compared to the second-best results are less than 1 dB. However, in terms of run-time, SITCOM consistently requires less computational time across all tasks. For FFHQ, SITCOM is over 3× faster in Box In-painting and motion Deblurring, and more than 2× faster in the remaining tasks, whereas on ImageNet, the run-time improvement ranges from

36 seconds (for HDR) to 62.4 seconds (for Super Resolution), when compared to DPS, DDNM, and DAPS.

For linear tasks, SITCOM requires slightly less run-time than DCDP on both datasets. However, across the two datasets, SITCOM achieves PSNR improvements of more than 1 dB, 2 dB, and 3 dB for the tasks of super resolution, box in-painting, and random in-painting (and Gaussian Deblurring), respectively, as compared to DCDP. For non-linear tasks, SITCOM not only provides PSNR improvements over DCDP but also significantly reduces run-time.

In summary, the results in Table 8.1 demonstrate that SITCOM either provides a notable improvement in restoration quality (e.g., cases where we report PSNR improvements of over 1 dB) or delivers comparable results to the baselines, all while significantly reducing computation time.

In Appendix D.5, we present the results with $\sigma_{\mathbf{y}} = 0.01$ case (Table D.1). Additionally, Table D.2 includes quantitative results for three more baselines. In addition to the FFHQ restored images in Figure 8.1, we also provide additional samples from both datasets in the figures found in Appendix D.8.

8.5 Conclusion

In this paper, we proposed three conditions to achieve measurement- and diffusion-consistent trajectories for linear and non-linear inverse imaging problems using diffusion models (DMs) as priors. These conditions form the basis of our unique optimization-based sampling method, which optimizes the input of the diffusion model at each step. This approach allows for greater control over the diffusion process and enhances data consistency with the given measurements. Through extensive experiments across eight image restoration tasks, we evaluated the effectiveness of our method. The results showed that our sampler consistently delivers improved or comparable quantitative performance against state-of-the-art baselines, even with measurement noise. Notably, our method is efficient, requiring significantly less run-time than leading baselines, making it practical for real-world applications.

CHAPTER 9

CONCLUSION

This chapter lists some of the possible extensions to the work presented in the thesis.

- In Chapter 3, we examined supervised learning of deep unrolled networks at reconstruction time for MRI by exploiting training sets along with local modeling and clustering. We intend to expand our studies in the future by incorporating non-Cartesian undersampling patterns, such as radial and spiral patterns, as well as deploying them to 3D settings and other imaging modalities.

- Additionally, the method's generalizability will be further examined, with a particular emphasis on heterogeneous datasets. To handle more extreme training-test data variations, such as unseen anatomies, we plan to explore patch-based neighbors in local learning schemes for future work.

- In Chapter 4, we introduced a self-guided deep image prior-based MRI reconstruction technique that iteratively optimizes the network input while also training the model to be robust to large random perturbations of its input. This was achieved by introducing a new regularization term that encourages the reconstructor to act as a denoiser. However, the main disadvantage is the time costs associated with gradient updates for the network input.

- In Chapter 5, to solve the problem of the previous chapter, we proposed the aSeq DIP, which relies solely on a sequential update of network parameters. These parameters are optimized using an input-adaptive data consistency objective combined with autoencoding regularization, effectively mitigating noise overfitting. For future directions, we aim to explore the applicability of aSeqDIP to other image recovery problems, thereby expanding its versatility and potential impact across diverse domains. Additionally, we are interested in investigating the integration of a network input update mechanism to dynamically adjust the autoencoding regularization parameter and the number of gradient updates per iteration.

Also, we want to have an analysis of the convergence of self-guided DIP and aSeqDIP is also needed and left for future work

- In Chapter 6, we proposed a scheme for improving the robustness of DL-based MRI reconstruction. In particular, we investigated deep unrolled reconstruction's weaknesses in robustness against worst-case or noise-like additive perturbations, sampling rates, and unrolling steps. To improve the robustness of the unrolled scheme, we proposed SMUG with a novel unrolled smoothing loss. In future work, we hope to apply the proposed schemes to other imaging modalities and evaluate robustness against additional types of realistic perturbations. While we theoretically characterized the robustness error for SMUG, we hope to further analyze its accuracy-robustness trade-off with perturbations.

- In chapter 7, we addressed challenges of unseen noise by harnessing the power of diffusion models. Our innovative robustification strategy enhanced the resilience of DL-based MRI reconstruction models by integrating pre-trained diffusion models as noise purifiers.

- In Chapter 8, we improve the diffusion purifier speed and improve performance by applying a better reverse sampling method by introducing the triple consistency regularization.

- In the future, we will focus on how to prune the unnecessary network weight for the diffusion model in order to improve the computation speed further

- In conclusion, this thesis addresses the dual challenges of deep learning model-based approaches, namely data scarcity and limited robustness. To alleviate data scarcity, we introduce three methods—LONDN-MRI, Self-Guided DIP, and aSeq DIP—that employ adaptive strategies to work effectively with limited datasets. In parallel, we tackle robustness issues through SMUG and diffusion purification, which mitigate vulnerabilities such as noise and adversarial perturbations. Furthermore, to improve the efficiency of diffusion models, we propose SITCOM, an approach that accelerates the reverse sampling process without

sacrificing result quality. Collectively, these contributions push the boundaries of deep learning in constrained settings while strengthening the reliability of model-based solutions

BIBLIOGRAPHY

Aggarwal, H. K., Mani, M. P., and Jacob, M. (2018). Modl: Model-based deep learning architecture for inverse problems. IEEE transactions on medical imaging, 38(2):394–405.

Aggarwal, H. K., Mani, M. P., and Jacob, M. (2019a). MoDL: model-based deep learning architecture for inverse problems. IEEE Trans. Med. Imaging, 38(2):394–405.

Aggarwal, H. K., Mani, M. P., and Jacob, M. (2019b). MoDL: Model-based deep learning architecture for inverse problems. IEEE Transaction on Medical Imaging, 38(2):394–405.

Akccakaya, M., Moeller, S., Weingartner, S., and Ugurbil, K. (2019). Scan-specific robust artificial-neural-networks for k-space interpolation (raki) reconstruction: Database-free deep learning for fast imaging. Magnetic Resonance in Medicine, 81(2):439–453.

Alkhouri, I., Liang, S., Wang, R., Qu, Q., and Ravishankar, S. (2024). Diffusion-based adversarial purification for robust deep mri reconstruction. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 12841–12845. IEEE.

Allgower, E. L. and Georg, K. (2012). Numerical continuation methods: an introduction, volume 13. Springer Science & Business Media.

Anderson, B. D. (1982). Reverse-time diffusion equation models. Stochastic Processes and their Applications, 12(3):313–326.

Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen, A. (2020a). On instabilities of deep learning in image reconstruction and the potential costs of AI. Proceedings of the National Academy of Sciences, 117(48):30088–30095.

Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen., A. C. (2020b). On instabilities of deep learning in image reconstruction and the potential costs of ai. Proceedings of the National Academy of Sciences, 117(48):30088–30095.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. (2019). On exact computation with an infinitely wide neural net. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Buzzard, G. T., Chan, S. H., Sreehari, S., and Bouman, C. A. (2018). Plug-and-play unplugged: optimization-free reconstruction using consensus equilibrium. SIAM J. Imaging Sci., 11(3):2001–20.

Chan, S. H., Wang, X., and Elgendy, O. A. (2016). Plug-and-play admm for image restoration: Fixed-point convergence and applications. IEEE Transactions on Computational Imaging, 3(1):84–98.

Chen, C., Liu, Y., Schniter, P., Tong, M., Zareba, K., Simonetti, O., Potter, L., and Ahmad, R. (2020). Ocmr (v1.0)–open-access multi-coil k-space dataset for cardiovascular magnetic resonance imaging. arXiv preprint arXiv:2008.03410.

Chen, G., Zhu, F., and Ann Heng, P. (2015). An efficient statistical method for image noise level estimation. In Proceedings of the IEEE International Conference on Computer Vision, pages 477–485.

Chen, S., Zhang, H., Guo, M., Lu, Y., Wang, P., and Qu, Q. (2024). Exploring low-dimensional subspaces in diffusion models for controllable image editing. arXiv preprint arXiv:2409.02374.

Cheng, J. (2019). Stanford 2D FSE.

Cheng, Z., Gadelha, M., Maji, S., and Sheldon, D. (2019). A bayesian perspective on the deep image prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5443–5451.

Chung, H., Kim, J., Kim, S., and Ye, J. C. (2023a). Parallel diffusion models of operator and image for blind inverse problems. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6059–6069.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023b). Diffusion posterior sampling for general noisy inverse problems. In The Eleventh International Conference on Learning Representations.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023c). Diffusion posterior sampling for general noisy inverse problems. In The Eleventh International Conference on Learning Representations.

Chung, H., Kim, J., and Ye, J. C. (2023d). Direct diffusion bridge using data consistency for inverse problems. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, Advances in Neural Information Processing Systems, volume 36, pages 7158–7169. Curran Associates, Inc.

Chung, H., Sim, B., Ryu, D., and Ye, J. C. (2022). Improving diffusion models for inverse problems using manifold constraints. Advances in Neural Information Processing Systems, 35:25683–25696.

Chung, H. and Ye, J. C. (2022). Score-based diffusion models for accelerated mri. Medical image analysis, 80:102479.

Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. In International Conference on Machine Learning, pages 1310–1320. PMLR.

Croce, F. and Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of

diverse parameter-free attacks.

Crockett, C. and Fessler, J. A. (2021). Bilevel methods for image reconstruction. arXiv preprint arXiv:2109.09610.

Dar, S. U. H., Özbey, M., **c**Catlı, A. B., and **c**Cukur, T. (2017). A Transfer-Learning Approach for Accelerated MRI using Deep Neural Networks. arXiv preprint arXiv:1710.02615.

Daras, G., Chung, H., Lai, C.-H., Mitsufuji, Y., Ye, J. C., Milanfar, P., Dimakis, A. G., and Delbracio, M. (2024). A survey on diffusion models for inverse problems. arXiv preprint arXiv:2410.00083.

Darestani, M. Z. and Heckel, R. (2021). Accelerated MRI with un-trained neural networks. arXiv preprint arXiv:2007.02471.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee.

Deshmane, A., Gulani, V., Griswold, M. A., and N, S. (2012). Parallel mr imaging. Journal of magnetic resonance imaging, 36(1):55–72.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. Advances in neural information processing systems, 34:8780–8794.

Donoho, D. (2006a). Compressed sensing. IEEE Transactions on Information Theory, 52(4):1289–1306.

Donoho, D. L. (2006b). Compressed sensing. IEEE Transactions on information theory, 52(4):1289–1306.

Elbakri, I. A. and Fessler, J. A. (2002). Statistical image reconstruction for polyenergetic X-ray computed tomography. IEEE Transactions on Medical Imaging, 21(2):89–99.

et al, F. K. (2020). fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning. Radiology: Artificial Intelligence, 2(1):e190007.

et al, J. Z. (2019). fastMRI: An Open Dataset and Benchmarks for Accelerated MRI. arXiv preprint arXiv:1811.08839.

Feng, C., Yan, Y., Fu, H., Chen, L., and Xu, Y. (2021). Task Transformer Network for Joint MRI Reconstruction and Super-Resolution. In International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI).

Fessler, J. A. (2010). Model-Based Image Reconstruction for MRI. IEEE Signal Processing

Magazine, 27(4):81–89.

Gatenby, R. A., Grove, O., and Gillies, R. J. (2013). Quantitative imaging in cancer evolution and ecology. Radiology, 269(1):8–14.

Ghosh, A., Mccann, M., and Ravishankar, S. (2022). Bilevel learning of l1 regularizrs with closed-form gradients (blorc). In ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1491–1495.

Ghosh, A., Zhang, X., Sun, K. K., Qu, Q., Ravishankar, S., and Wang, R. (2024). Optimal eye surgeon: Finding image priors through sparse generators at initialization. In Forty-first International Conference on Machine Learning.

Gilton, D., Ongie, G., and Willett, R. (2021a). Deep equilibrium architectures for inverse problems in imaging. IEEE Transactions on Computational Imaging, 7:1123–1133.

Gilton, D., Ongie, G., and Willett, R. (2021b). Deep equilibrium architectures for inverse problems in imaging. IEEE Transactions on Computational Imaging, 7:1123–1133.

Gretton, A., Borgwardt, K., Raschand, M., Schölkopf, B., and Smola, A. (2006). A kernel method for the two-sample-problem. Advances in neural information processing systems, 19.

Guan, H. and Liu, M. (2021). Domain adaptation for medical image analysis: a survey. IEEE Transactions on Biomedical Engineering, 69(3):1173–1185.

Güngör, A., Dar, S. U., Öztürk, **c**S., Korkmaz, Y., Bedel, H. A., Elmas, G., Ozbey, M., and **c**Cukur, T. (2023). Adaptive diffusion priors for accelerated MRI reconstruction. Medical Image Analysis, page 102872.

Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., and Knoll, F. (2018). Learning a variational network for reconstruction of accelerated MRI data. Magnetic resonance in medicine, 79(6):3055–3071.

Hatamizadeh, A., Song, J., Liu, G., Kautz, J., and Vahdat, A. (2024). Diffit: Diffusion vision transformers for image generation. In European Conference on Computer Vision, pages 37–55. Springer.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 1026–1034.

Heckel, R. and Hand, P. (2019). Deep decoder: Concise image representations from untrained non-convolutional networks. In ICLR.

Heckel, R. and Soltanolkotabi, M. (2020). Denoising and regularization via exploiting the structural

bias of convolutional generators. In ICLR.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851.

Hofmann, T., Schölkopf, B., and Smola, A. (2008). Kernel methods in machine learning. The Annals of Statistics, 36(3):1171–1220.

Hou, R., Li, F., and Zhang, G. (2022). Truncated residual based plug-and-play admm algorithm for MRI reconstruction. IEEE Transactions on Computational Imaging, 8:96–108.

Hsieh, J. (2003). Computed tomography: principles, design, artifacts, and recent advances. SPIE Journal of Medical Imaging, 42(6):1234–1245.

I, G., J, S., and C, S. (2015). Explaining and harnessing adversarial examples. 2015 ICLR, arXiv preprint arXiv:1412.6572.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: convergence and generalization in neural networks. In Proceedings of the 32nd Neurips, NIPS'18, page 8580–8589. Curran Associates Inc.

Jia, J., Hong, M., Y.Zhang, M.Akcakaya, and Liu, S. (2022a). On the robustness of deep learning-based MRI reconstruction to image transformations. In Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022.

Jia, J., Hong, M., Zhang, Y., Akcakaya, M., and Liu, S. (2022b). On the robustness of deep learning-based mri reconstruction to image transformations. In Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022.

Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. (2017). Deep convolutional neural network for inverse problems in imaging. IEEE Trans. Im. Proc., 26(9):4509–22.

Jo, Y., Chun, S. Y., and Choi, J. (2021). Rethinking deep image prior for denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5087–5096.

Kak, A. C. and Slaney, M. (2001). Principles of computerized tomographic imaging. SIAM.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems, 35:26565–26577.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4401–4410.

Kawar, B., Elad, M., Ermon, S., and Song, J. (2022). Denoising diffusion restoration models.

Advances in Neural Information Processing Systems, 35:23593–23606.

Kaya, M. and **c**S. bilge, H. (2019). Deep metric learning: A survey. Symmetry, 11(9).

Kim, T. H., Garg, P., and Haldar., J. P. (2019). LORAKI: Autocalibrated Recurrent Neural Networks for Autoregressive MRI Reconstruction in k-Space. arXiv preprint arXiv:1904.09390.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kingma, D. P. and Ba, J. (2015a). Adam: A method for stochastic optimization. 2015 ICLR, arXiv preprint arXiv:1412.6980.

Kingma, D. P. and Ba, J. (2015b). Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR).

Klug, T. and Heckel, R. (2023). Scaling laws for deep learning based image reconstruction. In ICLR.

Lahiri, A., Ravishankar, S., and Fessler, J. A. (2020). Combining supervised and semi-blind dictionary (Super-BReD) learning for MRI reconstruction. In Proc. Intl. Soc. Mag. Res. Med., page 3456.

Lahiri, A., Wang, G., Ravishankar, S., and Fessler, J. (2021). Blind Primed Supervised (BLIPS) Learning for MR Image Reconstruction. IEEE Transactions on Medical Imaging, 40(11):3113–3124.

Lakshmanan, H., De, F., and Daniela, P. (2008). Decentralized resource allocation in dynamic networks of agents. SIAM Journal on Optimization, 19(2):911–940.

Lee, S. S., Byun, J. H., Park, B. J., Park, S. H., Kim, N., Park, B., Kim, J. K., and Lee, M.-G. (2008). Quantitative analysis of diffusion-weighted magnetic resonance imaging of the pancreas: usefulness in characterizing solid pancreatic masses. Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine, 28(4):928–936.

Lei, K., Mardani, M., M.Pauly, J., and Vasanawala, S. (2021). Wasserstein gans for mr imaging: From paired to unpaired training. IEEE Transactions on Medical Imaging, 40(1):105–115.

Lei, K., Mardani, M., Pauly, J. M., and Vasanawala, S. S. (2020). Wasserstein gans for mr imaging: from paired to unpaired training. IEEE transactions on medical imaging, 40(1):105–115.

Li, H., Jia, J., Liang, S., Yao, Y., Ravishankar, S., and Liu, S. (2023). Smug: Towards robust mri reconstruction by smoothed unrolling. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE.

Li, T., Zhuang, Z., Liang, H., Peng, L., Wang, H., and Sun, J. (2021). Self-validation: Early stopping for single-instance deep generative priors. In Proceedings of the British Machine Vision Conference (BMVC), 2021.

Li, X., Kwon, S. M., Alkhouri, I. R., Ravishankar, S., and Qu, Q. (2024). Decoupled data consistency with diffusion purification for image restoration. arXiv preprint arXiv:2403.06054.

Liang, S., Bell, E., Qu, Q., Wang, R., and Ravishankar, S. (2024a). Analysis of deep image prior and exploiting self-guidance for image reconstruction. arXiv preprint arXiv:2402.04097.

Liang, S., Lahiri, A., and Ravishankar, S. (2024b). Adaptive local neighborhood-based neural networks for MR image reconstruction from undersampled data. IEEE Transactions on Computational Imaging. to appear.

Lingala, S. G. and Jacob, M. (2013). Blind compressive sensing dynamic MRI. IEEE Transactions on Medical Imaging, 32(6):1132–1145.

Liu, C., Freeman, W., Szeliski, R., and Kang, S. B. (2006). Noise estimation from a single image. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 1, pages 901–908.

Liu, C. and Hui, L. (2023). ReLU soothes the NTK condition number and accelerates optimization for wide neural networks. arXiv e-prints, pages arXiv–2305.

Liu, J., Sun, Y., Xu, X., and Kamilov, U. (2019a). Image restoration using total variation regularized deep image prior. ICASSP 2019 - 2019 IEEE International Conference on ICASSP.

Liu, J., Sun, Y., Xu, X., and Kamilov, U. S. (2019b). Image restoration using total variation regularized deep image prior. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7715–7719. Ieee.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11461–11471.

Luo, C. (2022). Understanding diffusion models: A unified perspective. arXiv preprint arXiv:2208.11970.

Lustig, M., Donoho, D., and Pauly, J. M. (2007). Sparse mri: The application of compressed sensing for rapid mr imaging. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, 58(6):1182–1195.

Ma, S., Yin, W., Zhang, Y., and Chakraborty, A. (2008). An efficient algorithm for compressed MR imaging using total variation and wavelets. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

Mardani, M., Song, J., Kautz, J., and Vahdat, A. (2023). A variational perspective on solving inverse problems with diffusion models. arXiv preprint arXiv:2305.04391.

McCollough, C. H., Leng, S., Yu, L., and Fletcher, J. G. (2015). Dual-and multi-energy ct: principles, technical approaches, and clinical applications. Radiology, 276(3):637–653.

McCollough, C. H., Primak, A. N., Braun, N., Kofler, J., Yu, L., and Christner, J. (2009). Strategies for reducing radiation dose in ct. Radiologic Clinics, 47(1):27–40.

Mihcak, M. K., Kozintsev, I., Ramchandran, K., and Moulin, P. (1999). Low-complexity image denoising based on statistical modeling of wavelet coefficients. IEEE Signal Processing Letters, 6(12):300–303.

Monga, V., Li, Y., and Eldar, Y. C. (2021). Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. IEEE Signal Processing Magazine, 38(2):18–44.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. (2022a). Diffusion models for adversarial purification. In International Conference on Machine Learning, pages 16805–16827. PMLR.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. (2022b). Diffusion models for adversarial purification. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 16805–16827. PMLR.

Peng, C., Guo, P., Zhou, S. K., Patel, V. M., and Chellappa, R. (2022). Towards performant and reliable undersampled MR reconstruction via diffusion model sampling. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 623–633. Springer.

Platen, E. and Bruti-Liberati, N. (2010). Numerical solution of stochastic differential equations with jumps in finance, volume 64. Springer Science & Business Media.

Ramani, A., Jensen, J. H., and Helpern, J. A. (2006). Quantitative mr imaging in alzheimer disease. Radiology, 241(1):26–44.

Ravishankar, S. and Bresler, Y. (2010). Mr image reconstruction from highly undersampled k-space data by dictionary learning. IEEE transactions on medical imaging, 30(5):1028–1041.

Ravishankar, S. and Bresler, Y. (2011). MR image reconstruction from highly undersampled k-space data by dictionary learning. IEEE Transactions on Medical Imaging, 30(5):1028–1041.

Ravishankar, S. and Bresler, Y. (2012). Learning sparsifying transforms. IEEE Transactions on Signal Processing, 61(5):1072–1086.

Ravishankar, S., Nadakuditi, R. R., and Fessler, J. A. (2015). Efficient sum of sparse outer products dictionary learning (SOUP-DIL). CoRR, abs/1511.06333.

Ravishankar, S., Ye, J. C., and A.Fessler, J. (2020). Image reconstruction: From sparsity to data-adaptive methods and machine learning. Proceedings of the IEEE, 108(1):86–109.

Romano, Y., Elad, M., and Milanfar, P. (2017). The Little Engine That Could: Regularization by Denoising (RED). SIAM Journal on Imaging Sciences, 10(4):1804–1844.

Ronneberger, O., Fischer, P., and Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pages 234–241. Springer.

Ronneberger, O., Fischer, P., and Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pages 234–241.

Rosenthal, D. I., Barton, N. W., McKusick, K. A., Rosen, B., Hill, S., Castronovo, F., Brady, R., Doppelt, S., and Mankin, H. (1992). Quantitative imaging of gaucher disease. Radiology, 185(3):841–845.

Roth, S. and Black, M. J. (2005). Fields of experts: a framework for learning image priors. In 2005 IEEE Computer Society Conference on CVPR'05, volume 2, pages 860–867 vol. 2.

Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A. G., and Shakkottai, S. (2023). Solving linear inverse problems provably via posterior sampling with latent diffusion models. arXiv preprint arXiv:2307.00619.

Salman, H., Sun, M., Yang, G., Kapoor, A., and Kolter, J. Z. (2020). Denoised smoothing: A provable defense for pretrained classifiers. Advances in Neural Information Processing Systems, 33.

Särkkä, S. and Solin, A. (2019). Applied stochastic differential equations, volume 10. Cambridge University Press.

Schlemper, J., Caballero, J., Hajnal, J. V., Price, A., and Rueckert, D. (2017). A deep cascade of convolutional neural networks for mr image reconstruction. In International Conference on Information Processing in Medical Imaging, pages 647–658. Springer.

Schlemper, J., Caballero, J., Hajnal, J. V., Price, A., and Rueckert, D. (2018). A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction. IEEE Transactions on

Medical Imaging, 37(2):491–503.

Shete, M. M. and Jadhav, C. R. (2023). Advancements in ct image reconstruction: An exploration of conventional and deep learning-driven approaches. In International Conference on Computational Intelligence, pages 77–88. Springer.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In International conference on machine learning, pages 2256–2265. PMLR.

Song, B., Kwon, S. M., Zhang, Z., Hu, X., Qu, Q., and Shen, L. (2023a). Solving inverse problems with latent diffusion models via hard data consistency. In The Twelfth International Conference on Learning Representations.

Song, B., Kwon, S. M., Zhang, Z., Hu, X., Qu, Q., and Shen, L. (2024). Solving inverse problems with latent diffusion models via hard data consistency. In The Twelfth International Conference on Learning Representations.

Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In International Conference on Learning Representations.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. (2023b). Consistency models. In International Conference on Machine Learning, pages 32211–32252. PMLR.

Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021b). Maximum likelihood training of score-based diffusion models. Advances in neural information processing systems, 34:1415–1428.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021c). Score-based generative modeling through stochastic differential equations. In International Conference on Learning Representations.

Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., Knoll, F., and Johnson, P. (2020). End-to-end variational networks for accelerated mri reconstruction. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23, pages 64–73. Springer.

Tachella, J., Tang, J., and Davies, M. (2021). The neural tangent link between cnn denoisers and non-local filters. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8618–8627.

Tachella, J., Tang, J., and Davies, M. E. (2020). CNN denoisers as non-local filters: The neural tangent denoiser. CoRR, abs/2006.02379.

Tamir, J. I., Ong, F., Cheng, J. Y., Uecker, M., and Lustig, M. (2016). Generalized magnetic resonance image reconstruction using the berkeley advanced reconstruction toolbox. In ISMRM

Workshop on Data Sampling & Image Reconstruction, Sedona, AZ.

Tran, P., Tran, A. T., Phung, Q., and Hoai, M. (2021). Explore image deblurring via encoded blur kernel space. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11956–11965.

Uecker, M. (2018). mrirecon/bart: version 0.4.03.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2018). Deep image prior. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9446–9454.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. Neural computation, 23(7):1661–1674.

Wang, H., Li, T., Zhuang, Z., Chen, T., Liang, H., and Sun, J. (2023a). Early stopping for deep image prior. Transactions on Machine Learning Research.

Wang, H., Zhang, X., Li, T., Wan, Y., Chen, T., and Sun, J. (2024). Dmplug: A plug-in method for solving inverse problems with diffusion models. arXiv preprint arXiv:2405.16749.

Wang, Y., Yu, J., and Zhang, J. (2022). Zero-shot image restoration using denoising diffusion null-space model. arXiv preprint arXiv:2212.00490.

Wang, Y., Yu, J., and Zhang, J. (2023b). Zero-shot image restoration using denoising diffusion null-space model. In The Eleventh International Conference on Learning Representations.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612.

Wang, Z., Qian, C., Guo, D., Sun, H., Li, R., Zhao, B., and Qu, X. (2023c). One-dimensional deep low-rank and sparse network for accelerated mri. IEEE Transactions on Medical Imaging, 42(1):79–90.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 681–688. Citeseer.

Wen, B., Li, Y., and Bresler, Y. (2020). Image recovery via transform learning and low-rank modeling: The power of complementary regularizers. IEEE Transactions on Image Processing, 29:5310–5323.

Wen, B., Ravishankar, S., Pfister, L., and Bresler, Y. (2020). Transform learning for magnetic resonance image reconstruction: From model-based learning to building neural networks. IEEE Signal Processing Magazine, 37(1):41–53.

Wen, B., Ravishankar, S., Zhao, Z., Giryes, R., and Ye, J. C. (2023). Physics-driven machine learning for computational imaging [from the guest editor]. IEEE Signal Processing Magazine, 40(1):28–30.

Wintermark, M., Sanelli, P. C., Anzai, Y., Tsiouris, A. J., Whitlow, C. T., Druzgal, T. J., Gean, A. D., Lui, Y. W., Norbash, A. M., Raji, C., et al. (2015). Imaging evidence and recommendations for traumatic brain injury: conventional neuroimaging techniques. Journal of the American College of Radiology, 12(2):e1–e14.

Wolf, A. (2019). Making medical image reconstruction adversarially robust. Online Report: https://cs229.stanford.edu/proj2019spr/report/97.pdf.

Xie, Y. and Li, Q. (2022). Measurement-conditioned denoising diffusion probabilistic model for under-sampled medical image reconstruction. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 655–664. Springer.

Xu, Q., Yu, H., Mou, X., Zhang, L., Hsieh, J., and Wang, G. (2012). Low-dose x-ray ct reconstruction via dictionary learning. IEEE Transactions on Medical Imaging, 31(9):1682–1697.

Yaman, B. ., Hosseini, S. A. H., Moeller, S., Ellermann, J., Ugurbil, K., and Akcakay, M. (2020). Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. Magnetic Resonance in Medicine, 84(6):3172–3191.

Yaman, B., Hosseini, S. A. H., and Akcakaya, M. (2022). Zero-shot self-supervised learning for MRI reconstruction. In International Conference on Learning Representations.

Yang, G., Yu, S., Dong, H., Slabaugh, G., Dragotti, P. L., Ye, X., Liu, F., Arridge, S., Keegan, J., Guo, Y., et al. (2017). Dagan: deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. IEEE transactions on medical imaging, 37(6):1310–1321.

Yang, Y., Sun, J., Li, H., and Xu, Z. (2016). Deep ADMM-Net for compressive sensing MRI. In Advances in Neural Information Processing Systems, pages 10–18.

Ye, S., Li, Z., McCann, M. T., Long, Y., and Ravishankar, S. (2021). Unified Supervised-Unsupervised (SUPER) Learning for X-Ray CT Image Reconstruction. IEEE Transactions on Medical Imaging, 40(11):2986–3001.

Yiasemis, G., Sonke, J.-J., Sánchez, C., and Teuwen, J. (2022). Recurrent variational network: a deep learning inverse problem solver applied to the task of accelerated mri reconstruction. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 732–741.

Yu, S., Park, B., and Jeong, J. (2019a). Deep iterative down-up cnn for image denoising. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 2095–2103.

Yu, S., Park, B., and Jeong, J. (2019b). Deep iterative down-up CNN for image denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0.

Zbontar, J., Knoll, F., Sriram, A., Murrell, T., Huang, Z., Muckley, M. J., Defazio, A., Stern, R., Johnson, P., Bruno, M., et al. (2018). fastmri: An open dataset and benchmarks for accelerated mri. arXiv preprint arXiv:1811.08839.

Zeng, G. L. (2020). Fast filtered backprojection algorithm for low-dose computed tomography. Journal of radiology and imaging, 4(7):45.

Zhang, B., Chu, W., Berner, J., Meng, C., Anandkumar, A., and Song, Y. (2024a). Improving diffusion inverse problem solving with decoupled noise annealing. arXiv preprint arXiv:2407.01521.

Zhang, H., Zhou, J., Lu, Y., Guo, M., Wang, P., Shen, L., and Qu, Q. (2024b). The emergence of reproducibility and consistency in diffusion models. In Forty-first International Conference on Machine Learning.

Zhang, J. and Ghanem, B. (2018). ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing. arXiv preprint arXiv:1706.07929.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 586–595.

Zhang, Y., Yao, Y., Jia, J., Yi, J., Hong, M., Chang, S., and Liu, S. (2022). How to robustify black-box ML models? a zeroth-order optimization perspective. In International Conference on Learning Representations.

Zhao, D., Zhao, F., and Gan, Y. (2020a). Reference-driven compressed sensing mr image reconstruction using deep convolutional neural networks without pre-training. Sensors, 20(1):308.

Zhao, D., Zhao, F., and Gan, Y. (2020b). Reference-driven compressed sensing mr image reconstruction using deep convolutional neural networks without pre-training. Sensors, 20(1).

Zheng, H., Fang, F., and Zhang, G. (2019). Cascaded dilated dense network with two-step data consistency for MRI reconstruction. In NeurIPS.

Zoph, B., G.Ghiasi, Lin, T., Cui, Y., Liu, H., Cubuk, E. . D., and Le, Q. (2020). Rethinking pre-training and self-training. Advances in Neural Information Processing Systems, 33.

APPENDIX A

APPENDIX FOR SELF-GUIDED DIP

In the Appendix, we provide additional intuition and a detailed explanation for Theorems 4.1.1 and 4.1.2, as well as the Corollary introduced in the previous section.

A.1 Proof of Theorem 4.1.1

We start from the following update step for the estimate $\boldsymbol{z}_t$:

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \eta \boldsymbol{W}(\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{z}_t). \tag{A.1}$$

We make a change of variables $\boldsymbol{p}_t(\boldsymbol{w}) = \boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{z}_t(\boldsymbol{w})$, where $\boldsymbol{W}^{-\frac{1}{2}}$ is the pseudo-inverse of $\boldsymbol{W}^{1/2}$, and $\boldsymbol{W}^{1/2}$ is the positive semidefinite matrix whose square is equal to $\boldsymbol{W}$. With this change of variables, the recursion formula (A.1) becomes

$$\begin{aligned}
&\boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \eta \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{p}_t) \\
&= (\boldsymbol{I} - \eta \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}})\boldsymbol{p}_t + \eta \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{x}_\perp + \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x})) \\
&= (\boldsymbol{I} - \eta \boldsymbol{B})\boldsymbol{p}_t + \eta \boldsymbol{B}(\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x}) + \eta \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_\perp,
\end{aligned}$$

where $\boldsymbol{z}_t(\boldsymbol{w}) = \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{p}_t(\boldsymbol{w})$ because $\boldsymbol{z}_0 = \boldsymbol{0}$ and so $\boldsymbol{z}_t(\boldsymbol{w}) \in R(\boldsymbol{W}) = R(\boldsymbol{W}^{\frac{1}{2}})$ here. We have also set $\boldsymbol{B} := \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}}$, and $\boldsymbol{x}_\perp := P_{N(\boldsymbol{W})}\boldsymbol{x}$. We have also used the decomposition $\boldsymbol{x} = \boldsymbol{x}_\perp + P_{R(\boldsymbol{W})}\boldsymbol{x}$. Since we hope $\boldsymbol{z}_t$ to converge to $\boldsymbol{x}$, then $\boldsymbol{p}_t$ is expected to converge to $\tilde{\boldsymbol{x}} := \boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x}$. Now we keep track of the errors $\varepsilon_t := \boldsymbol{p}_t - \tilde{\boldsymbol{x}}$ and $\boldsymbol{e}_t := \boldsymbol{z}_t - \boldsymbol{x}$. By subtracting $\tilde{\boldsymbol{x}}$ from the above recursion for $\boldsymbol{p}_t$, we obtain

$$\varepsilon_{t+1} = (\boldsymbol{I} - \eta \boldsymbol{B})\varepsilon_t + \eta \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_\perp,$$

This implies the following result. The proof for the second equality is included in Appendix D in the supplement.

$$\begin{aligned}
\varepsilon_t &= (\boldsymbol{I} - \eta \boldsymbol{B})^t\varepsilon_0 + \eta \left[\sum_{k=0}^{t-1}(\boldsymbol{I} - \eta \boldsymbol{B})^k\right] \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_\perp \\
&= (\boldsymbol{I} - \eta \boldsymbol{B})^t\varepsilon_0 + \boldsymbol{B}^\dagger(\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{B})^t)\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_\perp.
\end{aligned} \tag{A.2}$$

Invoking the relation between $\boldsymbol{p}_t$ and $\boldsymbol{z}_t$, we can derive the following useful relation between the errors $\boldsymbol{e}_t$ and ε_t:

$$\begin{aligned}
\boldsymbol{e}_t &= \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{p}_t - \boldsymbol{x} && \text{(A.3)}\\
&= \boldsymbol{W}^{\frac{1}{2}}(\varepsilon_t + \tilde{\boldsymbol{x}}) - \boldsymbol{x} && \text{(A.4)}\\
&= \boldsymbol{W}^{\frac{1}{2}}(\varepsilon_t + \boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x}) - \boldsymbol{x} && \text{(A.5)}\\
&= \boldsymbol{W}^{\frac{1}{2}}\varepsilon_t + \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x} - \boldsymbol{x} && \text{(A.6)}\\
&= \boldsymbol{W}^{\frac{1}{2}}\varepsilon_t + P_{R(\boldsymbol{W})}\boldsymbol{x} - \boldsymbol{x} && \text{(A.7)}\\
&\overset{(*)}{=} \boldsymbol{W}^{\frac{1}{2}}\varepsilon_t - P_{N(\boldsymbol{W})}(\boldsymbol{x}) && \text{(A.8)}\\
&\overset{(**)}{=} \boldsymbol{W}^{\frac{1}{2}}\varepsilon_t + P_{N(\boldsymbol{W})}(\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{p}_t - \boldsymbol{x}) && \text{(A.9)}\\
&= \boldsymbol{W}^{\frac{1}{2}}\varepsilon_t + P_{N(\boldsymbol{W})}\boldsymbol{e}_t, && \text{(A.10)}
\end{aligned}$$

where $P_{N(\boldsymbol{W})}$ denotes projection onto the null space of $\boldsymbol{W}$. Most steps in the derivation above follow from the definitions of the quantities $\boldsymbol{p}_t$, $\boldsymbol{e}_t$, and $\tilde{\boldsymbol{x}}$. To obtain $(*)$, we have used the symmetry of $\boldsymbol{W}$, and to obtain $(**)$ we have used the fact that $R(\boldsymbol{W}^{\frac{1}{2}})$ is equal to $R(\boldsymbol{W})$, which is orthogonal to $N(\boldsymbol{W})$. In what follows, for simplicity of notation, we use $P_{\boldsymbol{W}}$ to denote the projection onto the range of $\boldsymbol{W}$. Using the above relation and the fact that $R(\boldsymbol{W})$ and $N(\boldsymbol{W})$ are orthogonal for symmetric $\boldsymbol{W}$, we can write:

$$\begin{aligned}
&P_{\boldsymbol{W}}\boldsymbol{e}_t = \boldsymbol{W}^{\frac{1}{2}}\varepsilon_t\\
&= \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t\varepsilon_0 + \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{B}^{\dagger}(\boldsymbol{I} - (\boldsymbol{I} - \eta\boldsymbol{B})^t)\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_{\perp}\\
&= \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{e}_0\\
&+ \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{B}^{\dagger}(\boldsymbol{I} - (\boldsymbol{I} - \eta\boldsymbol{B})^t)\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_{\perp}.
\end{aligned}$$

On the other hand, subtracting $\boldsymbol{x}$ from (A.1) and then projecting both sides of the resulting equation to $N(\boldsymbol{W})$ yields

$$P_{N(\boldsymbol{W})}\boldsymbol{e}_t = P_{N(\boldsymbol{W})}\boldsymbol{e}_{t-1} = \cdots = P_{N(\boldsymbol{W})}\boldsymbol{e}_0.$$

Summing the above two equations yields

$$\begin{aligned} \boldsymbol{z}_t - \boldsymbol{x} &= \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x}) + P_{N(\boldsymbol{W})}(\boldsymbol{z}_0 - \boldsymbol{x}) \\ &\quad + \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{B}^{\dagger}(\boldsymbol{I} - (\boldsymbol{I} - \eta\boldsymbol{B})^t)\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_{\perp}. \end{aligned} \tag{A.11}$$

If $\boldsymbol{W}$ is of full rank, then (A.11) reduces to

$$\boldsymbol{z}_t - \boldsymbol{x} = \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x}) \tag{A.12}$$

(A.12) can be further rewritten as

$$\begin{aligned} \boldsymbol{z}_t - \boldsymbol{x} &= \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t P_{R(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x}) \\ &\quad + \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t P_{N(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x}) \\ &= \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t P_{R(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x}) \\ &\quad + \boldsymbol{W}^{\frac{1}{2}}P_{N(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x}). \end{aligned} \tag{A.13}$$

In order to make sure the operator $\boldsymbol{I} - \eta\boldsymbol{B}$ is non-expansive, we need to require the learning rate η to satisfy $\eta < \frac{2}{\|\boldsymbol{B}\|}$, where $\|\boldsymbol{B}\|$ is the spectral norm of $\boldsymbol{B}$. Under this assumption, $||\boldsymbol{I} - \eta\boldsymbol{B}\| \leq \rho := \max\{1 - \eta\sigma_{\min}(\boldsymbol{B}), \eta\|\boldsymbol{B}\| - 1\} < 1$, meaning that the operator $\boldsymbol{I} - \eta\boldsymbol{B}$ is contractive on the range of $\boldsymbol{B}$. Then as $t \to \infty$, the first term on the right-hand side in (A.13) converges to 0, since

$$\|\boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t P_{R(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x})\|_2^2 \leq \kappa(\boldsymbol{W})\rho^{2t}\|\boldsymbol{z}_0 - \boldsymbol{x}\|_2^2$$

where $\kappa(\boldsymbol{W})$ is the condition number of $\boldsymbol{W}$[1]. Therefore, (A.13) implies that

$$\begin{aligned} \boldsymbol{z}_\infty - \boldsymbol{x} &= \boldsymbol{W}^{\frac{1}{2}}P_{N(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}(\boldsymbol{z}_0 - \boldsymbol{x}) \\ &= -\boldsymbol{W}^{\frac{1}{2}}P_{N(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x}, \end{aligned} \tag{A.14}$$

where the last equality used the assumption $\boldsymbol{z}_0 = \boldsymbol{0}$.

[1] If $\boldsymbol{W}$ is low-rank, then its condition number is defined as the ratio of the maximal and minimal non-zero singular values.

Let $\mathbf{v} := P_{N(\boldsymbol{B})}\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x}$. By this definition, we have $\mathbf{v} \in N(\boldsymbol{B})$ which is equivalent to $\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}}\mathbf{v} = 0$ or $\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}}\mathbf{v} = 0$. The latter implies $\boldsymbol{W}^{\frac{1}{2}}\mathbf{v} \in N(\boldsymbol{A})$. This when combined with the equation $\boldsymbol{z}_\infty - \boldsymbol{x} = -\boldsymbol{W}^{\frac{1}{2}}\mathbf{v}$ (A.14), yields $\boldsymbol{z}_\infty - \boldsymbol{x} \in N(\boldsymbol{A})$.

Moreover, for $\boldsymbol{z}_\infty - \boldsymbol{x}$ to be $\mathbf{0}$, it is necessary that $\mathbf{v} = \mathbf{0}$, which means $\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x}$ has to be orthogonal to $N(\boldsymbol{B})$. Consequently, this necessitates that $\boldsymbol{x}$ be orthogonal to $N(\boldsymbol{A})$, or $P_{N(\boldsymbol{A})}\boldsymbol{x} = 0$. This completes the proof for the full-rank portion of the theorem.

To prove the result for the singular $\boldsymbol{W}$ case, we rewrite the quantity $\boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t\boldsymbol{W}^{-\frac{1}{2}}$ in (A.11) as follows. Here, for simplicity of notation, we use $P_{\boldsymbol{B}}$ to denote the projection onto the range of $\boldsymbol{B}$, and $P_{\boldsymbol{B}^\perp}$ to denote the projection onto the kernel of $\boldsymbol{B}$.

$$\begin{aligned}\boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t\boldsymbol{W}^{-\frac{1}{2}} =& \boldsymbol{W}^{\frac{1}{2}}P_{\boldsymbol{B}}(\boldsymbol{I} - \eta\boldsymbol{B})^t P_{\boldsymbol{B}}\boldsymbol{W}^{-\frac{1}{2}} \\ &+ \boldsymbol{W}^{\frac{1}{2}}(P_{\boldsymbol{B}^\perp}P_{\boldsymbol{W}}P_{\boldsymbol{B}^\perp})^t\boldsymbol{W}^{-\frac{1}{2}}. \end{aligned} \tag{A.15}$$

The detailed proof of the above result is given in Supplement Appendix E.

Taking $t \to \infty$ in (A.15), we obtain that

$$\begin{aligned}&\lim_{t\to\infty} \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{I} - \eta\boldsymbol{B})^t\boldsymbol{W}^{-\frac{1}{2}} \\ &= \boldsymbol{W}^{\frac{1}{2}} \lim_{t\to\infty} P_{\boldsymbol{B}}(\boldsymbol{I} - \eta\boldsymbol{B})^t P_{\boldsymbol{B}}\boldsymbol{W}^{-\frac{1}{2}} \\ &+ \boldsymbol{W}^{\frac{1}{2}} \lim_{t\to\infty} (P_{\boldsymbol{B}^\perp}P_{\boldsymbol{W}}P_{\boldsymbol{B}^\perp})^t\boldsymbol{W}^{-\frac{1}{2}} \\ &= \mathbf{0} + \boldsymbol{W}^{\frac{1}{2}}P_{N(\boldsymbol{B})\cap R(\boldsymbol{W})}\boldsymbol{W}^{-\frac{1}{2}},\end{aligned}$$

where the last equality used the fact that $\lim_{n\to\infty}(P_{\mathbf{A}}P_{\mathbf{B}}P_{\mathbf{A}})^n = P_{\mathbf{A}\cap\mathbf{B}}$. Then (A.11) implies that

$$\begin{aligned}\boldsymbol{z}_\infty - \boldsymbol{x} &= -\boldsymbol{W}^{\frac{1}{2}}P_{N(\boldsymbol{B})\cap R(\boldsymbol{W})}\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x} - \boldsymbol{x}_\perp \\ &\quad + \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{B}^\dagger\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_\perp \\ &= -\boldsymbol{W}^{\frac{1}{2}}P_{N(\boldsymbol{B})\cap R(\boldsymbol{W})}\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x} - \boldsymbol{x}_\perp \\ &\quad + \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}})^\dagger\boldsymbol{A}\boldsymbol{x}_\perp, \end{aligned} \tag{A.16}$$

where the last equality is based on the fact that $(\mathbf{C}\mathbf{C}^H)^\dagger\mathbf{C} = (\mathbf{C}^H)^\dagger$ for any tall matrix $\mathbf{C}$.

Now if

$$P_{N(\boldsymbol{B})\cap R(\boldsymbol{W})}\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x} = \mathbf{0}, \tag{A.17}$$

then the first term in the RHS of (A.16) is $\mathbf{0}$, and then

$$\boldsymbol{z}_\infty - \boldsymbol{x} = -\boldsymbol{x}_\perp + \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{A}\boldsymbol{W}^{\frac{1}{2}})^\dagger \boldsymbol{A}\boldsymbol{x}_\perp. \tag{A.18}$$

Given that the condition expressed in equation (A.17) can be inferred from the condition (A.19) below, it follows that (A.19) also implies (A.18) as stated in the theorem.

$$P_{N(\boldsymbol{A})\cap R(\boldsymbol{W})}\boldsymbol{x} = \mathbf{0}. \tag{A.19}$$

The rationale behind (A.19) being a sufficient condition of (A.17) is

$$\begin{aligned}
&P_{N(\boldsymbol{A})\cap R(\boldsymbol{W})}\boldsymbol{x} = \mathbf{0} \Rightarrow \boldsymbol{x} \perp N(\boldsymbol{A}) \cap R(\boldsymbol{W}) \\
&\Rightarrow \langle \boldsymbol{x}, \mathbf{a}\rangle = 0, \forall \mathbf{a} \in N(\boldsymbol{A}) \cap R(\boldsymbol{W}) \\
&\Rightarrow \langle \boldsymbol{x}, \mathbf{a}\rangle = 0, \forall \mathbf{a} \in R(\boldsymbol{W}), \boldsymbol{A}\mathbf{a} = \mathbf{0} \\
&\Rightarrow \langle \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{x}, \boldsymbol{W}^{-\frac{1}{2}}\mathbf{a}\rangle = 0, \forall \mathbf{a} \in R(\boldsymbol{W}), \boldsymbol{A}\boldsymbol{W}^{1/2}\boldsymbol{W}^{-1/2}\mathbf{a} = \mathbf{0} \\
&\Rightarrow \langle \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{x}, \mathbf{b}\rangle = 0, \forall \mathbf{b} \in R(\boldsymbol{W}), \boldsymbol{A}\boldsymbol{W}^{1/2}\mathbf{b} = \mathbf{0} \\
&\Rightarrow \langle \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{x}, \mathbf{b}\rangle = 0, \forall \mathbf{b} \in R(\boldsymbol{W}), \boldsymbol{B}\mathbf{b} = \mathbf{0} \\
&\Rightarrow \boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x} \perp N(\boldsymbol{B}) \cap R(\boldsymbol{W}) \\
&\Rightarrow P_{N(\boldsymbol{B})\cap R(\boldsymbol{W})}\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{x} = \mathbf{0},
\end{aligned}$$

where $\mathbf{b} = \boldsymbol{W}^{-\frac{1}{2}}\mathbf{a}$.

Furthermore, if aside from (A.19) we also have $\boldsymbol{x} \in R(\boldsymbol{W})$, then (A.18) reduces to

$$\boldsymbol{z}_\infty - \boldsymbol{x} = \mathbf{0}$$

which completes the proof of Theorem 1.

A.2 Proof of Theorem 4.1.2

In this case, we suppose that the acquired measurements are $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \mathbf{n}$, where $\mathbf{n} \in \mathbb{R}^p$ with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, and $\boldsymbol{A} \in \mathbb{R}^{p\times q}$ is full row rank. We first note that this can equivalently be written

as $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n})$, since $\boldsymbol{A}$ has full row rank, so $\boldsymbol{A}\boldsymbol{A}^\dagger \mathbf{n} = \mathbf{n}$. Then, we start with the recursion (A.1), which in this case gives:

$$\begin{aligned}\boldsymbol{z}_{t+1} &= \boldsymbol{z}_t + \eta \boldsymbol{W}(\boldsymbol{A}^T \boldsymbol{A}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) - \boldsymbol{A}^T \boldsymbol{A}\boldsymbol{z}_t) \\ &= \boldsymbol{z}_t + \eta \boldsymbol{W}\boldsymbol{A}^T \boldsymbol{A}((\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) - \boldsymbol{z}_t).\end{aligned}$$

We set $\boldsymbol{z}_0 = \mathbf{0}$ and define $\mathbf{K} := \eta \boldsymbol{W}\boldsymbol{A}^T \boldsymbol{A}$ to ease notation. We can use this to derive a useful closed-form for $\boldsymbol{z}_t$:

$$\begin{aligned}\boldsymbol{z}_t &= (\boldsymbol{I} - \mathbf{K})\boldsymbol{z}_{t-1} + \mathbf{K}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) \\ &= (\boldsymbol{I} - \mathbf{K})((\boldsymbol{I} - \mathbf{K})\boldsymbol{z}_{t-2} + \mathbf{K}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n})) + \mathbf{K}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) \\ &= (\boldsymbol{I} - \mathbf{K})^2 \boldsymbol{z}_{t-2} + (\boldsymbol{I} + (\boldsymbol{I} - \mathbf{K}))\mathbf{K}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) \\ &= (\boldsymbol{I} - \mathbf{K})^2((\boldsymbol{I} - \mathbf{K})\boldsymbol{z}_{t-3} + \mathbf{K}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n})) \\ &\quad + (\boldsymbol{I} + (\boldsymbol{I} - \mathbf{K}))\mathbf{K}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) \\ &= (\boldsymbol{I} - \mathbf{K})^3 \boldsymbol{z}_{t-3} + (\boldsymbol{I} + (\boldsymbol{I} - \mathbf{K}) + (\boldsymbol{I} - \mathbf{K})^2)\mathbf{K}(\boldsymbol{x} + A^\dagger \mathbf{n}) \\ &\vdots \\ &= \underbrace{(\boldsymbol{I} - \mathbf{K})^t \boldsymbol{z}_0}_{\mathbf{0}} + \sum_{i=0}^{t-1} (\boldsymbol{I} - \mathbf{K})^i \mathbf{K}(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) \\ &\overset{(*)}{=} (\boldsymbol{I} - (\boldsymbol{I} - \mathbf{K})^t)(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}) \\ &= (\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T \boldsymbol{A})^t)(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n}).\end{aligned}$$

To obtain the equality $(*)$, we have used the algebraic identity $\sum_{i=0}^{t-1} \mathbf{M}^i(\boldsymbol{I} - \mathbf{M}) = \boldsymbol{I} - \mathbf{M}^t$, which holds for any square matrix $\mathbf{M}$. In this case, replacing $\mathbf{M}$ with $\boldsymbol{I} - \mathbf{K}$ yields the identity used in $(*)$.

We can express the squared norm of the bias at iteration t as:

$$\begin{aligned}||\mathbf{Bias}_t||_2^2 &= ||\mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t] - \boldsymbol{x}||_2^2 \\ &= ||\mathbb{E}_{\mathbf{n}}[(\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T \boldsymbol{A})^t)(\boldsymbol{x} + \boldsymbol{A}^\dagger \mathbf{n})] - \boldsymbol{x}||_2^2 \\ &\overset{(**)}{=} ||(\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T \boldsymbol{A})^t)\boldsymbol{x} - \boldsymbol{x}||_2^2 \\ &= ||(\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T \boldsymbol{A})^t \boldsymbol{x}||_2^2,\end{aligned}$$

where $(**)$ follows by linearity of expectation and the assumption that $\mathbf{n}$ is zero mean. Next we compute the covariance matrix of $\boldsymbol{z}_t$ as:

$$\mathbf{Cov}_t = \mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t \boldsymbol{z}_t^T] - \mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t]\mathbb{E}_{\mathbf{n}}[\boldsymbol{z}_t]^T \tag{A.20}$$

To simplify notation, we define the matrix $\boldsymbol{R}_t := (\boldsymbol{I} - (\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{A}^T\boldsymbol{A})^t)$, so we get:

$$\begin{aligned}
\mathbf{Cov}_t &= \mathbb{E}_{\mathbf{n}}[\boldsymbol{R}_t(\boldsymbol{x} + \boldsymbol{A}^\dagger\mathbf{n})(\boldsymbol{x} + \boldsymbol{A}^\dagger\mathbf{n})^T\boldsymbol{R}_t^T] \\
&\quad - \mathbb{E}_{\mathbf{n}}[\boldsymbol{R}_t(\boldsymbol{x} + \boldsymbol{A}^\dagger\mathbf{n})]\mathbb{E}_{\mathbf{n}}[\boldsymbol{R}_t(\boldsymbol{x} + \boldsymbol{A}^\dagger\mathbf{n})]^T \\
&= \mathbb{E}_{\mathbf{n}}[\boldsymbol{R}_t(\boldsymbol{x}\boldsymbol{x}^T + \boldsymbol{A}^\dagger\mathbf{n}\mathbf{n}^T(\boldsymbol{A}^\dagger)^T)\boldsymbol{R}_t^T] - (\boldsymbol{R}_t\boldsymbol{x})(\boldsymbol{R}_t\boldsymbol{x})^T \\
&= \mathbb{E}_{\mathbf{n}}[\boldsymbol{R}_t\boldsymbol{A}^\dagger\mathbf{n}\mathbf{n}^T(\boldsymbol{A}^\dagger)^T\boldsymbol{R}_t^T] \\
&= \sigma^2\boldsymbol{R}_t\boldsymbol{A}^\dagger(\boldsymbol{A}^\dagger)^T\boldsymbol{R}_t^T \\
&= \sigma^2\boldsymbol{Q}_t\boldsymbol{Q}_t^T,
\end{aligned}$$

where we have defined $\boldsymbol{Q}_t := \boldsymbol{R}_t\boldsymbol{A}^\dagger = (\boldsymbol{I} - (\boldsymbol{I} - \eta\boldsymbol{W}\boldsymbol{A}^T\boldsymbol{A})^t)\boldsymbol{A}^\dagger$. Then, to compute the variance, we take the trace of $\mathbf{Cov}_t$, and use the fact that the trace of a matrix is the sum of its eigenvalues. However, the eigenvalues of $\boldsymbol{Q}_t\boldsymbol{Q}_t^T$ are exactly the squares of the singular values of $\boldsymbol{Q}_t$. This gives us that:

$$\mathrm{Var}_t = \sigma^2 \sum_{i=1}^{p} \nu_{t,i}^2, \tag{A.21}$$

where $\nu_{t,i}$ are the singular values of $\boldsymbol{Q}_t$. Summing these expressions for the bias and variance of the estimate exactly give equation (4.17).

A.3 Proof of Corollary 1

We now consider the single-coil MRI forward operator, $\boldsymbol{A} = \mathbf{M}\boldsymbol{\mathcal{F}}$, where $\boldsymbol{\mathcal{F}}$ is the usual Fourier operator. Since $\boldsymbol{A} \in \mathbb{C}^{p\times q}$, we introduce an equivalent $\tilde{\boldsymbol{A}} \in \mathbb{R}^{2p\times 2q}$ (that maps between stacked real and imaginary parts of vectors) to ensure that everything is real-valued. Throughout, we use subscripts R and I to denote the real and imaginary parts of vectors or operators. We define the matrices $\tilde{\mathbf{M}} \in \mathbb{R}^{2p\times 2q}$ and $\tilde{\boldsymbol{\mathcal{F}}} \in \mathbb{R}^{2q\times 2q}$ by:

$$\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}; \qquad \tilde{\boldsymbol{\mathcal{F}}} = \begin{bmatrix} \boldsymbol{\mathcal{F}}_R & -\boldsymbol{\mathcal{F}}_I \\ \boldsymbol{\mathcal{F}}_I & \boldsymbol{\mathcal{F}}_R \end{bmatrix}.$$

We note that $\tilde{\boldsymbol{\mathcal{F}}}$ is orthogonal, i.e. $\tilde{\boldsymbol{\mathcal{F}}}^T\tilde{\boldsymbol{\mathcal{F}}} = \tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{\mathcal{F}}}^T = \boldsymbol{I}$. Thus, we define $\tilde{\boldsymbol{A}} = \tilde{\mathbf{M}}\tilde{\boldsymbol{\mathcal{F}}}$. It is straightforward to verify that applying $\tilde{\boldsymbol{A}}$ to a vector with stacked real and imaginary components is equivalent to applying $\boldsymbol{A}$ to a complex vector. We also rewrite $\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_R \\ \boldsymbol{x}_I \end{bmatrix}$ and $\tilde{\mathbf{n}} = \begin{bmatrix} \mathbf{n}_R \\ \mathbf{n}_I \end{bmatrix}$. Supposing that $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$, we then have that $\mathbf{n}_R, \mathbf{n}_I \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \frac{\sigma^2}{2}\boldsymbol{I})$.

We consider a network with a 2 channel output, i.e., a network that outputs $\tilde{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{z}_R \\ \boldsymbol{z}_I \end{bmatrix} \in \mathbb{R}^{2q}$, so that its NTK is $\tilde{\boldsymbol{W}} \in \mathbb{R}^{2q \times 2q}$. We now suppose that $\tilde{\boldsymbol{W}}$ is diagonalized by $\tilde{\boldsymbol{\mathcal{F}}}$ with the following structure:

$$\tilde{\boldsymbol{W}} = \tilde{\boldsymbol{\mathcal{F}}}^T\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{\mathcal{F}}}; \qquad \tilde{\boldsymbol{\Lambda}} = \begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} \end{bmatrix}.$$

With this structure, applying $\tilde{\boldsymbol{W}}$ to a vector with its real and imaginary parts concatenated is equivalent to applying the circulant matrix $\boldsymbol{W} = \boldsymbol{\mathcal{F}}^H\boldsymbol{\Lambda}\boldsymbol{\mathcal{F}}$ to a complex vector.

With this reformulation, the update equation for $\tilde{\boldsymbol{z}}_t$ becomes:

$$\begin{aligned}
\tilde{\boldsymbol{z}}_t &= (\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{W}}\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})^t)(\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}^\dagger\tilde{\mathbf{n}}) \\
&= (\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\mathcal{F}}}^T\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{\mathcal{F}}}(\tilde{\mathbf{M}}\tilde{\boldsymbol{\mathcal{F}}})^T\tilde{\mathbf{M}}\tilde{\boldsymbol{\mathcal{F}}})^t)(\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}^\dagger\mathbf{n}) \\
&= (\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\mathcal{F}}}^T\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}}\tilde{\boldsymbol{\mathcal{F}}})^t)(\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}^\dagger\tilde{\mathbf{n}}) \\
&= \tilde{\boldsymbol{\mathcal{F}}}^T(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)\tilde{\boldsymbol{\mathcal{F}}}(\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}^\dagger\tilde{\mathbf{n}}).
\end{aligned}$$

In this case, the bias becomes:

$$
\begin{aligned}
&||\mathbf{Bias}_t||_2^2 \\
&= ||\mathbb{E}_{\tilde{\mathbf{n}}}[\tilde{\boldsymbol{z}}_t] - \tilde{\boldsymbol{x}}||_2^2 \\
&= ||\mathbb{E}_{\tilde{\mathbf{n}}}[(\tilde{\boldsymbol{\mathcal{F}}}^T(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)\tilde{\boldsymbol{\mathcal{F}}}(\tilde{\boldsymbol{x}} + \boldsymbol{A}^\dagger\tilde{\mathbf{n}})] - \tilde{\boldsymbol{x}}||_2^2 \\
&= ||\tilde{\boldsymbol{\mathcal{F}}}^T(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)\tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}||_2^2 \\
&= ||\tilde{\boldsymbol{\mathcal{F}}}^T(\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t\tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{x}}||_2^2 \\
&= ||(\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)\tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{x}}||_2^2 \\
&= \sum_{i=1}^{2q}(1 - \eta\tilde{\lambda}_i\tilde{m}_i)^{2t}|(\tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{x}})_i|^2 \\
&= \sum_{i=1}^{q}(1 - \eta\lambda_i m_i)^{2t}|(\boldsymbol{\mathcal{F}}\boldsymbol{x})_i|^2,
\end{aligned}
$$

where $\tilde{\lambda}_i$ are the diagonal entries of $\tilde{\boldsymbol{\Lambda}}$, $\tilde{m}_i$ are the diagonal entries of $\tilde{\mathbf{M}}^T\tilde{\mathbf{M}}$, and $(\tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{x}})_i$ is the ith entry of $\tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{x}}$.

The computation of the covariance is similar to Theorem 2 with small modifications. First, we now have $\boldsymbol{R}_t = \tilde{\boldsymbol{\mathcal{F}}}^T(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)\tilde{\boldsymbol{\mathcal{F}}}$. Also, we define $\boldsymbol{Q}_t = \boldsymbol{R}_t\tilde{\boldsymbol{A}}^T$, which is valid since we have the identity $\tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^T\tilde{\mathbf{n}} = \tilde{\mathbf{n}}$. We also note the additional factor of $\frac{1}{2}$ introduced by separating $\mathbf{n}$ into $\mathbf{n}_R$ and $\mathbf{n}_I$. Thus, we have:

$$
\begin{aligned}
\text{Var}_t &= \text{tr}(\mathbf{Cov}_t) \\
&= \frac{\sigma^2}{2}\text{tr}(\tilde{\boldsymbol{\mathcal{F}}}^T(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t) \\
&\tilde{\boldsymbol{\mathcal{F}}}\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}}\tilde{\boldsymbol{\mathcal{F}}}^T(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)\tilde{\boldsymbol{\mathcal{F}}}) \\
&= \frac{\sigma^2}{2}\text{tr}((\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t) \\
&\tilde{\mathbf{M}}^T\tilde{\mathbf{M}}(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)) \\
&= \frac{\sigma^2}{2}\text{tr}(\tilde{\mathbf{M}}^T\tilde{\mathbf{M}}(\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)^2) \\
&= \frac{\sigma^2}{2}\text{tr}((\boldsymbol{I} - (\boldsymbol{I} - \eta\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^t)^2) \\
&= \frac{\sigma^2}{2}\sum_{i=1}^{2q}(1 - (1 - \eta\tilde{\lambda}_i\tilde{m}_i)^t)^2 \\
&= \sigma^2\sum_{i=1}^{q}(1 - (1 - \eta\lambda_i m_i)^t)^2.
\end{aligned}
$$

Summing these expressions for the bias and variance yields the result of Corollary 1.

APPENDIX B

APPENDIX FOR AUTOENCODING SEQUENTIAL DEEP IMAGE PRIOR

In this Appendix, we first shed more light on the impact of the DIP network input by studying the training dynamics using the neural tangent kernel for CNNs with residual connections. Next, we show how trained autoencoders on clean images can be used as reconstructors at testing time. Lastly, we provide additional experimental results and visualizations.

B.1 Case Study: Impact of the DIP Network Input through lens of Neural Tangent Kernel in Residual Networks

We show the impact of the DIP input through the lens of the Neural Tangent Kernel (NTK) (Tachella et al., 2021; Jacot et al., 2018) for residual networks[1]. The NTK is a tool used to analyze the training dynamics of neural networks in the infinite width limit, where for CNNs the network width corresponds to the number of channels. In this limit, the change of any individual parameter during training becomes very small, which means that the change in the network's output during training can be accurately approximated by a first order Taylor expansion around its initialization. In the context of DIP, we consider training a neural network f with parameters θ and a fixed input z using gradient descent. At each training iteration, the network parameters are updated according to:

$$\theta^{(t+1)} = \theta^{(t)} - \beta \nabla_\theta \mathcal{L}(f_{\theta^{(t)}}(z)), \tag{B.1}$$

where $\mathcal{L}$ is the loss function, and β is the learning rate. We also consider the resulting change in the network's output due to this parameter update using the first order Taylor expansion:

$$f_{\theta^{(t+1)}}(z) \approx f_{\theta^{(t)}}(z) + \nabla_\theta f_\theta(z)\big|_{\theta=\theta^{(t)}} (\theta^{(t+1)} - \theta^{(t)}). \tag{B.2}$$

Substituting (B.1) into (B.2) and applying the chain rule to write:

$$\nabla_\theta \mathcal{L}(f_{\theta^{(t)}}(z)) = (\nabla_\theta f_\theta(z)\big|_{\theta=\theta^{(t)}})^T (\nabla_{f_{\theta^{(t)}}(z)} \mathcal{L}(f_{\theta^{(t)}}(z))) \tag{B.3}$$

[1]We note that skip and residual connections are not exactly the same as skip represents concatenation (typically from encoder to decoder end) and residual represents adding the input to the output. However, both operations correspond to sending initial input or features of a network to its latter portion or output.

yields the equation:

$$f_{\theta^{(t+1)}}(\boldsymbol{z}) \approx f_{\theta^{(t)}}(\boldsymbol{z}) - \beta \underbrace{(\nabla_\theta f_\theta(\boldsymbol{z})\big|_{\theta=\theta^{(t)}})(\nabla_\theta f_\theta(\boldsymbol{z})\big|_{\theta=\theta^{(t)}})^T}_{\boldsymbol{\Theta}^{(t)}}(\nabla_{f_{\theta^{(t)}}(\boldsymbol{z})}\mathcal{L}(f_{\theta^{(t)}}(\boldsymbol{z}))) \,. \tag{B.4}$$

In the infinite width limit, NTK theory states that the matrix $\boldsymbol{\Theta}^{(t)} := (\nabla_\theta f_\theta(\boldsymbol{z})\big|_{\theta=\theta^{(t)}})(\nabla_\theta f_\theta(\boldsymbol{z})\big|_{\theta=\theta^{(t)}})^T$ stays fixed throughout training, so that $\boldsymbol{\Theta}^{(t)} = \boldsymbol{\Theta}^{(0)}$ for all t. This matrix is called the neural tangent kernel, and we denote it as $\boldsymbol{\Theta}$. Moreover, because the parameters $\theta^{(0)}$ are initialized randomly, in the infinite width limit, the NTK $\boldsymbol{\Theta}$ becomes deterministic (as a function of $\boldsymbol{z}$) due to the law of large numbers (Tachella et al., 2021), and does not depend on the specific instantiation of $\theta^{(0)}$. In DIP, the loss function is the least squares loss given in (2.8). For simplicity, we consider the denoising case, where the forward operator $\mathbf{A} = \mathbf{I}$. Then, substituting the gradient of the loss into (B.4) shows explicitly how the output of deep image prior evolves during training:

$$f_{\theta^{(t+1)}}(\boldsymbol{z}) = f_{\theta^{(t)}}(\boldsymbol{z}) + \beta\boldsymbol{\Theta}(\mathbf{y} - f_{\theta^{(t)}}(\boldsymbol{z})). \tag{B.5}$$

Using this recursion relation, one can derive a closed form of the network output at iteration t in terms of the initial output and NTK (Liang et al., 2024a; Tachella et al., 2021). The reconstruction at iteration t is given by:

$$f_{\theta^{(t)}}(\boldsymbol{z}) = \mathbf{y} - (\mathbf{I} - \beta\boldsymbol{\Theta})^t(\mathbf{y} - f_{\theta^{(0)}}(\boldsymbol{z})). \tag{B.6}$$

It is evident from (B.6) that the initial reconstruction of the network, $f_{\theta^{(0)}}(\boldsymbol{z})$, has important effects on the training dynamics of DIP. Furthermore, networks used in DIP often feature skip connections from earlier layers to later ones, and it is natural to believe that these connections may cause the input $\boldsymbol{z}$ to have a large effect on $f_{\theta^{(0)}}(\boldsymbol{z})$. In the following theorem, we analyze the training dynamics of CNNs with a very similar architectural modification: a residual connection that adds the input directly to the network output.

Theorem B.1.1 (Dynamics of DIP with Residual Connections). Let g be a convolutional neural network with parameters θ. We consider the complementary residual network f defined by $f_\theta(\boldsymbol{z}) = \boldsymbol{z} + g_\theta(\boldsymbol{z})$. Suppose that f is trained using gradient descent with the loss $\mathcal{L}(f_\theta(\boldsymbol{z})) = \frac{1}{2}||f_\theta(\boldsymbol{z}) - \mathbf{y}||_2^2$.

Then, in the infinite width limit (number of channels), in expectation over the initialization of the parameters θ, we have that the output at training iteration t is given by:

$$f_{\theta^{(t)}}(\boldsymbol{z}) = \mathbf{y} - (\mathbf{I} - \beta\boldsymbol{\Theta})^t(\mathbf{y} - \boldsymbol{z}). \tag{B.7}$$

The proof of Theorem B.1.1 is provided in Appendix B.1.1, along with a precise statement of the assumptions on the network architecture and parameter initialization. Additionally, in Appendix B.1.2 we provide a simple experiment to validate that the training dynamics given in (B.7) hold for real networks.

Remark 6. *Theorem B.1.1 can be used to understand how the choice of network input affects the performance of DIP for image denoising. To gain intuition, we consider two special cases. First, we consider using the noisy image* $\mathbf{y}$ *as the input* $\boldsymbol{z}$*. In this case, equation* (B.7) *simplifies to* $f_{\theta^{(t)}}(\mathbf{y}) = \mathbf{y}$ *for all iterations* t*. In this case absolutely no denoising occurs. On the other hand, we consider the oracle case where the clean image* $\mathbf{x}$ *is used as* $\boldsymbol{z}$*. This gives us* $f_{\theta^{(t)}}(\mathbf{x}) = \mathbf{y} - (\mathbf{I} - \beta\boldsymbol{\Theta})^t(\mathbf{y} - \mathbf{x})$*. We see that at initialization (*$t = 0$*), we already expect perfect denoising, since* $f_{\theta^{(0)}}(\mathbf{x}) = \mathbf{y} - (\mathbf{I} - \beta\boldsymbol{\Theta})^0(\mathbf{y} - \mathbf{x}) = \mathbf{x}$*. These two cases support intuition that using a network input closer to the true image could result in better performance in fewer training iterations.*

B.1.1 Proof of Theorem B.1.1

Setting of Theorem B.1.1. We first precisely state the conditions of Theorem B.1.1, in particular the network architectures considered and the corresponding parameter initializations. The present setting is very similar to the setting considered in (Tachella et al., 2021), but we provide the details here for completeness. We consider an L layer CNN with c_{in} input channels and c_{out} output channels, with c hidden channels in all intermediate layers. We assume that $c_{\text{in}}, c_{\text{out}} << c$. We assume all convolutions have a filter size of r. For simplicity, the network input and output are vectorized, so convolutions of any dimension are treated identically. For example, for a 2D CNN with 5×5 kernels, $r = 25$. Written explicitly, a network g with this architecture takes the form

$$g_\theta(\boldsymbol{z}) = C_L(\varphi(C_{L-1}(\varphi(\cdots\varphi(C_1(\boldsymbol{z})))))),$$

where the operators C_i represent convolutions with an additive bias, and φ is a pointwise activation function such as ReLU. In this section, we also consider the residual network architecture defined by $f_\theta(z) = z + g_\theta(z)$.

We assume that the parameters are initialized using the He initialization (He et al., 2015). With this initialization, the first layer convolutional filter weights are drawn from $\mathcal{N}(0, \frac{\sigma_w^2}{c_{\text{in}} r})$, and the filter weights for all other layers are drawn from $\mathcal{N}(0, \frac{\sigma_w^2}{cr})$, where the variance σ_w^2 depends on the non-linearity used in the network. For ReLU networks, $\sigma_w^2 = 2$ (He et al., 2015). All biases are initialized to 0.

Proof. In the setting described above, the NTK emerges in the limit $c \to \infty$. A body of existing theory (Tachella et al., 2021; Jacot et al., 2018; Arora et al., 2019) establishes that in this limit the NTK is a *deterministic* matrix as a function of the network input z. This theory does not consider residual connections, but applies immediately to both g and f. For g, the NTK is given by

$$\mathbf{\Theta} := (\nabla_\theta g_\theta(z)\big|_{\theta=\theta^{(0)}})(\nabla_\theta g_\theta(z)\big|_{\theta=\theta^{(0)}})^T . \tag{B.8}$$

However, we can see that $\nabla_\theta g_\theta(z)\big|_{\theta=\theta^{(0)}} = \nabla_\theta f_\theta(z)\big|_{\theta=\theta^{(0)}}$.

Therefore, the linearization given in equation (B.4) holds for f using the same kernel $\mathbf{\Theta}$, and equation (B.6) describes the training dynamics of f.

Using equation (B.6), we can write:

$$f_{\theta^{(t)}}(z) = \mathbf{y} - (\mathbf{I} - \beta\mathbf{\Theta})^t(\mathbf{y} - f_{\theta^{(0)}}(z)) \tag{B.9}$$

$$= \mathbf{y} - (\mathbf{I} - \beta\mathbf{\Theta})^t(\mathbf{y} - z - g_{\theta^{(0)}}(z)) \tag{B.10}$$

To prove Theorem B.1.1, we consider the output $f_{\theta^{(t)}}(z)$ in expectation over the initialization $\theta^{(0)}$. Since all parameters $\theta^{(0)}$ are drawn from mean 0 gaussian distributions, we find that $\mathbb{E}_{\theta^{(0)}}[g_{\theta^{(0)}}(z)] = 0$ for any input z. Since $\mathbf{\Theta}$ is deterministic in the limit $c \to \infty$, in expectation over $\theta^{(0)}$ equation (B.10) reduces to $f_{\theta^{(t)}}(z) = \mathbf{y} - (\mathbf{I} - \beta\mathbf{\Theta})^t(\mathbf{y} - z)$, which proves Theorem B.1.1. □

B.1.2 Example to support the results of Theorem B.1.1

We now provide a simple example using real networks to support the validity of Theorem B.1.1. This experiment substantiates both of the special cases considered in Remark 6. Additionally, it shows that using the ground truth as the network input greatly inhibits overfitting. Indeed, in this experiment, we find that a residual network trained with the ground truth as input takes approximately 25 times more training iterations to completely learn the noisy signal than a residual network trained using a random noise input.

We use DIP for denoising a 1D sinusoidal signal. We denote the clean signal $\mathbf{x}$. The noisy signal is $\mathbf{y} = \mathbf{x}+\mathbf{n}$, where $\mathbf{n} \sim \mathcal{N}(0, \mathbf{I})$. The network used is a five layer ReLU CNN with a residual connection. The full architecture can be written as $f(\boldsymbol{z}) = \boldsymbol{z} + C_5(\text{ReLU}(C_4(\cdots \text{ReLU}(C_2(\text{ReLU}(C_1(\boldsymbol{z})))))))$, where each C_i represents a convolution (with bias). The signal has a size of 100, and the convolutions each have a filter size of 3, with 64 hidden channels. In all cases, the network is trained using gradient descent with a learning rate of 5×10^{-4}. The same seed was used to initialize the network in all cases.

We trained this network using three different inputs: the true signal $\mathbf{x}$, the noisy signal $\mathbf{y}$, and noise $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$. The results of training the network using these three inputs are shown in Figure B.1. We find that the results for training this real, reasonably sized network show the behavior predicted by Theorem B.1.1, which was obtained in the infinite width limit. Indeed, equation (B.7) predicts that when $\mathbf{x}$ is used as the input, the initial error will be small, with eventual overfitting to the noisy signal. This is observed in Figure B.1, where the error is lowest at initialization, and it steadily increases throughout training. With this input, the network is highly resistant to overfitting, requiring approximately 10^5 training iterations to completely fit the noisy signal. We also see that when $\mathbf{y}$ is used as the input, the error curve obtained is essentially flat and converges quickly to the error of the noisy signal. This agrees with the expectation that the network output will be $\mathbf{y}$ for all iterations t when $\mathbf{y}$ is used as the input. Finally, when random noise $\boldsymbol{z}$ is used as the input, the typical DIP behavior emerges.

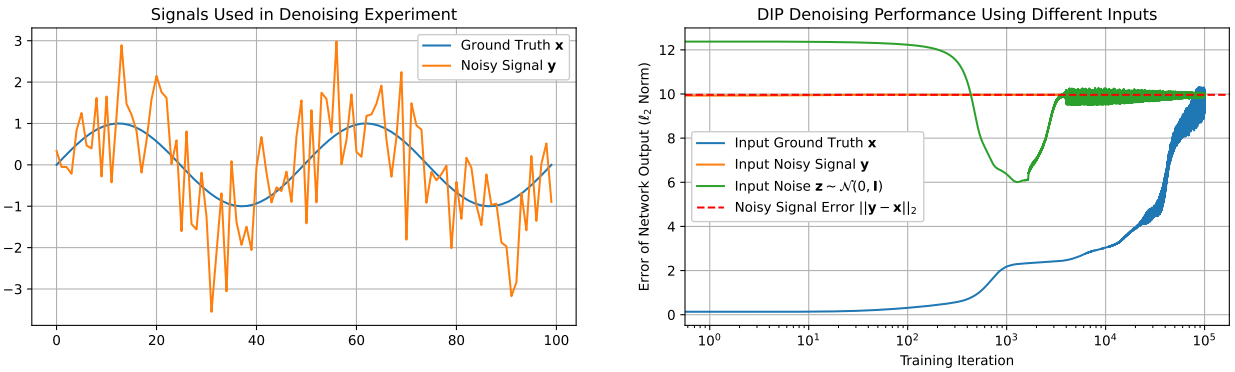


Figure B.1 Ground truth signal and measurements (*left*), and the results of the denoising experiment in Appendix B.1.2 (*right*) to support the claims in Theorem B.1.1 and Remark 6.

B.1.3 Trained Autoencoders as Reconstructors

Here, we investigate how the autoencoder term in aSeqDIP is improving the reconstruction quality while mitigating the impact of noise overfitting. In particular, we try to answer the question: *Can an autoencoder trained on clean images operate as a reconstructor at testing time?*

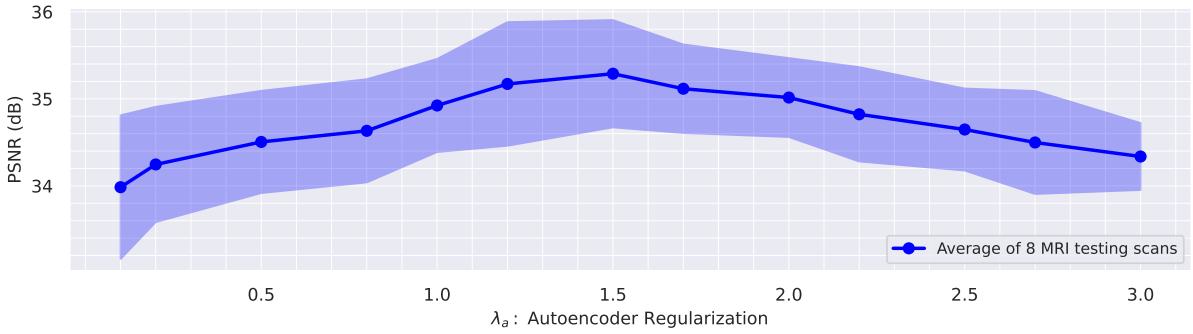


Figure B.2 Average PSNR (y-axis) of 8 MRI images (with 4x undersampling) obtained by optimizing the input of a trained autoencoder (using (B.11)) w.r.t. different values of the regularization parameter λ_a in (B.12) (x-axis).

To address this question, we perform the following steps: (*i*) train an autoencoder on fully sampled measurements or clean data images and (*ii*) utilize the trained autoencoder with unseen subsampled or corrupted measurements, optimizing over the input using the DIP objective with the autoencoder term. This enables the autoencoder to function as an image reconstructor. Specifically, given a training dataset, $\mathcal{D}$, comprising unperturbed images or fully sampled MRI/CT data, denoted by $\mathbf{x}$, we train an autoencoder U-Net $g : \mathbb{R}^n \to \mathbb{R}^n$ with parameters ψ. The training process seeks to

obtain $\hat{\psi}$ as

$$\hat{\psi} = \arg\min_{\psi} \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}\in\mathcal{D}} \|g_{\psi}(\mathbf{x}) - \mathbf{x}\|_2^2 \,. \tag{B.11}$$

Subsequently, given unseen measurements $\mathbf{y}$ and the learned autoencoder's parameters $\hat{\psi}$, we test the reconstruction of

$$\mathbf{z} \leftarrow \arg\min_{\mathbf{z}} \|\mathbf{A}g_{\hat{\psi}}(\mathbf{z}) - \mathbf{y}\|_2^2 + \lambda_a \|g_{\hat{\psi}}(\mathbf{z}) - \mathbf{z}\|_2^2 \,, \tag{B.12}$$

where $\lambda_a \in \mathbb{R}_+$ is a regularization parameter. We perform this experiment by training ψ using 3000 fully sampled scans from the fastMRI dataset (Zbontar et al., 2018). We then evaluate the reconstruction quality of the trained encoder using 8 scans from the fastMRI testing set. The average PSNR results for different values of λ_a are depicted in Figure B.2. As observed, a trained autoencoder effectively serves as a reconstructor as evidenced by the achieved PSNR. Thus, we deduce that the autoencoder term in aSeqDIP not only mitigates noise overfitting but also enhances the reconstruction quality as an important prior.

B.2 Additional Experiments

B.2.1 Robustness to Noise Overfitting for the Denoising Task

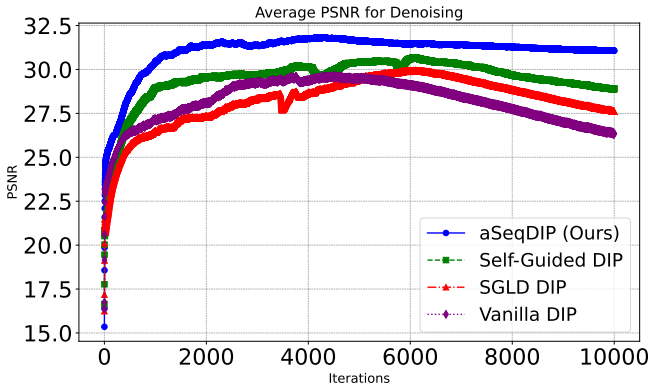


Figure B.3 Average PSNR results of 20 images w.r.t. to the optimization iteration using aSeqDIP and DIP-based baselines.

In this subsection, we illustrate aSeqDIP's robustness to the noise overfitting issue for the denoising task. The average PSNR for 20 images from the CBSD68 dataset for denoising using aSeqDIP and other DIP-based methods are given in Figure B.3. We observe two key points. First, in addition to higher PSNR, aSeqDIP shows higher robustness against noise overfitting compared to

other DIP-based methods, consistent with MRI and CT results in Figure 5.4. Second, unlike MRI and CT, the onset of noise overfitting occurs earlier, but the subsequent decay is very small.

B.2.2 Comparison with VarNet: An End-to-End MRI Supervised Method

Here, we compare aSeqDIP with the End-to-End (E2E) MRI supervised model (Sriram et al., 2020) that uses the variational network (VarNet). Results are given in Table B.1. As observed, we slightly under-perform when compared to VarNet (trained on 8000 data points) for the task of MRI reconstruction all without requiring any labeled training data. It is important to note that, at inference, E2E models only require a few unrolling steps, whereas aSeqDIP is an optimization method that requires to train the network parameters for each new set of measurements.

Task	Method	Data Independency	PSNR (↑)
MRI	E2E VarNet (trained on 8000 data points from fastMRI) (Sriram et al., 2020)	×	**34.89**
	E2E VarNet (trained on 3000 data points from fastMRI) (Sriram et al., 2020)	×	33.78
	aSeqDIP (Ours)	✓	<u>34.08</u>

Table B.1 Average PSNR results (over 20 MRI scans at 4x undersampling from the testing set of fastMRI) reported by our method against E2E VarNet (Sriram et al., 2020) (pre-trained on fastMRI) for the task of MRI reconstruction.

B.2.3 Comparison with DM-based Methods on the FFHQ Dataset

Task	Method	Data Independency	PSNR (↑)
Denoising	DPS (trained on FFHQ) (Chung et al., 2023c)	×	31.45
	DDNM (trained on FFHQ) (Wang et al., 2023b)	×	31.65
	aSeqDIP (Ours)	✓	**31.77**
Random In-Painting	DPS (trained on FFHQ) (Chung et al., 2023c)	×	24.54
	DDNM (trained on FFHQ) (Wang et al., 2023b)	×	25.54
	aSeqDIP (Ours)	✓	**25.76**
Deblurring	DPS (trained on FFHQ) (Chung et al., 2023c)	×	23.67
	DDNM (trained on FFHQ) (Wang et al., 2023b)	×	23.88
	aSeqDIP (Ours)	✓	**24.02**
Box In-Painting	DPS (trained on FFHQ) (Chung et al., 2023c)	×	22.67
	DDNM (trained on FFHQ) (Wang et al., 2023b)	×	**22.89**
	aSeqDIP (Ours)	✓	22.3

Table B.2 Average PSNR results reported by our method against DPS as well as a more recent leading method DDNM (Wang et al., 2023b) for four image restoration tasks: Denoising, Random In-Painting, non-linear Deblurring, and Box In-Painting.

Here, we present average PSNR results (averaged over 20 images) for the tasks of denoising , random inpainting (97% missing pixels), box-in-painting (with HIAR of 0.25), and non-linear deblurring of our method versus Denoising Diffusion Null-Space Model (DDNM) (Wang et al.,

2023b) and DPS (Chung et al., 2023c) on the FFHQ testing dataset. For DPS and DDNM, we used a pre-trained model that was trained on the training set of FFHQ. As observed, our training-data-free method achieves competitive or slightly improved results when compared to data-intensive methods on all tasks other than box-inpainting (for which we under-perform by less than 1 dB), all without requiring a pre-trained model.

B.2.4 Ablation Study on the Regularization Parameter in aSeqDIP

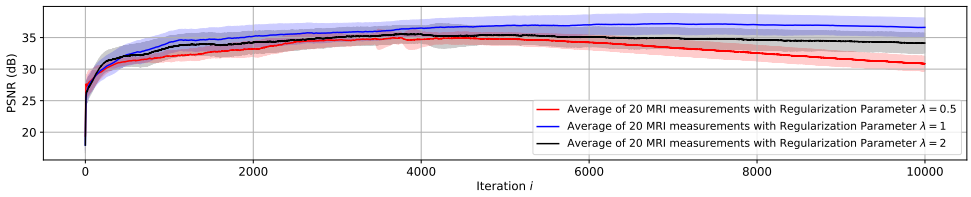


Figure B.4 Average PSNR results of 20 MRI (with 4x undersampling) scans in aSeqDIP for the cases where $\lambda \in \{0.5, 1, 2\}$ and $i \in [10000]$.

In this section, we conduct an ablation study on the choice of the autoencoding regularization parameter, λ, in (5.3). Specifically, we conduct an experiment using 20 MRI scans to examine the impact of λ on the reconstruction quality and noise overfitting in aSeqDIP. We note that while the main results in Section 5.3 use $NK = 4000$, in these experiments, we run our algorithm for an extended number of iterations to investigate the onset of noise overfitting. We set $K = 5000$ and $N = 2$ for this purpose.

In this experiment, we run aSeqDIP with values of $\lambda \in \{0.5, 1, 2\}$. Average PSNR results are given in Figure B.4. It is evident that, on average, using $\lambda = 1$ yields the most favorable results in terms of PSNR values, which is our selected choice. Furthermore, we observe that for $\lambda = 0.5$ (red), the start of the PSNR decay (the onset of noise overfitting) precedes that of $\lambda = 1$ and $\lambda = 2$ (blue and black).

B.2.5 Ablation Study on N and K in aSeqDIP

In this section, we conduct an ablation study to investigate the impact of the number of gradient updates (N) per one set of parameter (K) in aSeqDIP. Specifically, we report the PSNR results across the tasks of MRI, CT, denoising, and in-painting for the case of $NK = 4000$, considering combinations of (N, K) as $(1, 4000)$, $(2, 2000)$, and $(4, 1000)$. The results, presented in Figure B.5,

reveal that across all tasks considered, the combination of $N = 2$ and $K = 2000$ consistently yields the most favorable results in terms of PSNR.

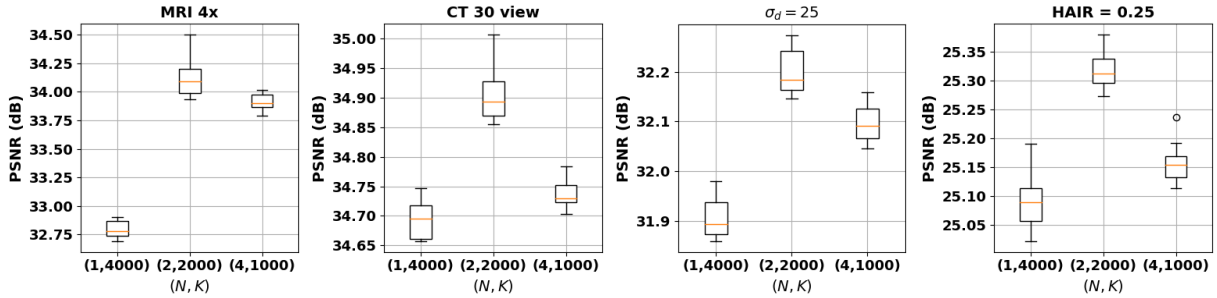


Figure B.5 Ablation study on the choice of the number of gradient updates N per U-Net, and the number of U-Nets K in terms of PSNR using the four considered tasks for the case of $NK = 4000$.

B.2.6 Additional Implementation Details

For denoising, each noisy RGB image is 512×512 and is generated by adding an additive Gaussian white noise with two noise levels as descired in Table 5.1. For in-painting, we consider a central region mask, and we evaluate two hole-to-image area ratios (HIAR) with image size 512×512.

For Vanilla DIP, Self-Guided DIP, Reference-guided DIP, TV-DIP, Rethinking DIP, and SGLD DIP, we use 4000 iterations. For TV-DIP, we set the regularization parameter to 1. For ES-DIP, we use the default configuration provided by the authors for the three considered tasks.

The reference image in Reference-guided DIP is chosen as (using a distance metric such as Euclidean distance or other metric) that which is most similar to an estimated test reconstruction from undersampled data or sparse-view data.

For DM-based approaches, we use the codes attached to the authors' papers. Specifically, Score-MRI (Chung and Ye, 2022), MCG (Chung et al., 2022), and DPS (Chung et al., 2023c). For our experiments with natural images (denoising, in-painting, and non-linear deblurring) in Table 5.2, we used the CBSD68 dataset. As such, for DPS (the DM-based method), we utilized a pre-trained model that was trained on a very large and diverse dataset which is ImageNet 128×128, 256×256, and 512×512. This pre-trained model is much more generalizable when compared to the other option which was trained on FFHQ (a dataset of faces). According to (Zhang et al., 2024b),

the ImageNet pre-trained model has high generalizability. For the FFHQ comparison results in Appendix B.2.3, we used an FFHQ-pre-trained DM for DPS and DDNM.

For MRI, the pre-trained model used in Score-MRI was originally trained on natural images then fine-tuned using the training set of fastMRI. Similar approach was used for the CT pre-trained model used in MCG.

B.3 Additional Visualizations

Figures B.6 and B.7 present additional MRI visualisations, whereas Figures B.8 and B.9 present CT visualizations. Samples from the natural image restoration tasks are given in Figure B.10, Figure B.11, and Figure B.12 for box-inpainting, denoising, and non-linear deblurring, respectively.

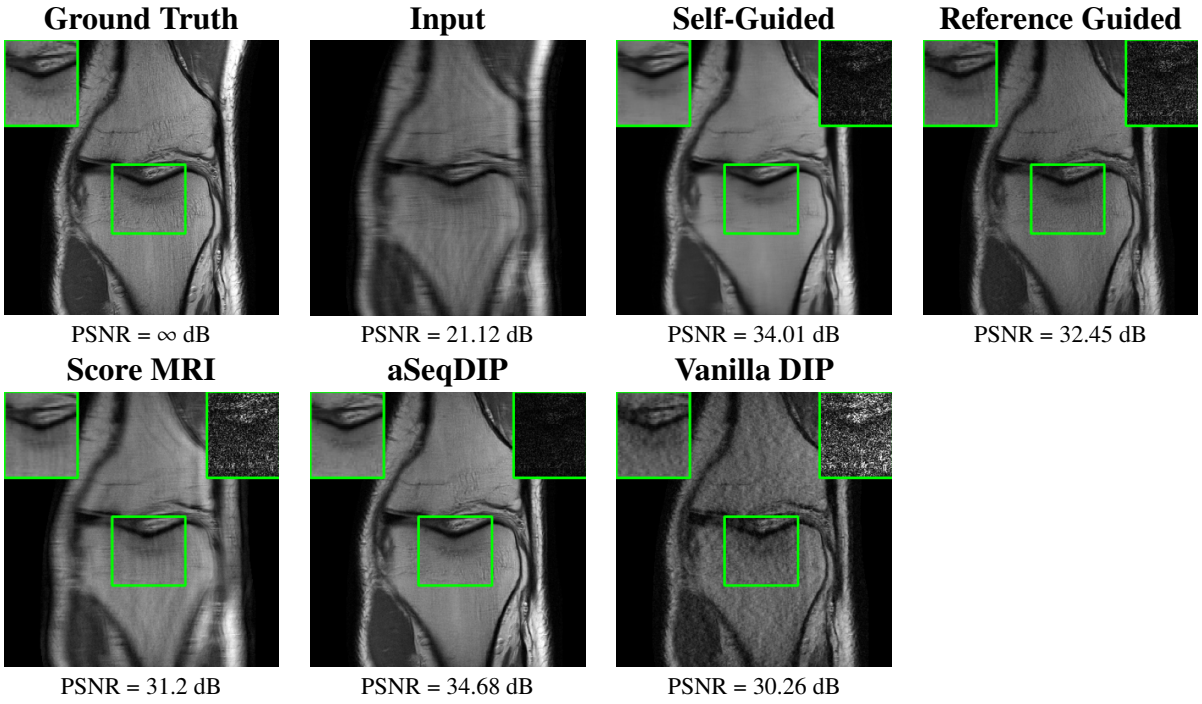


Figure B.6 Visualization of ground-truth and reconstructed images using different methods of a knee image from the fastMRI dataset with 4x k-space undersampling. A region of interest is shown with a green box and its error (magnitude) is shown in the panel on the top right. aSeqDIP provides the sharpest and clearest reconstruction of image features.

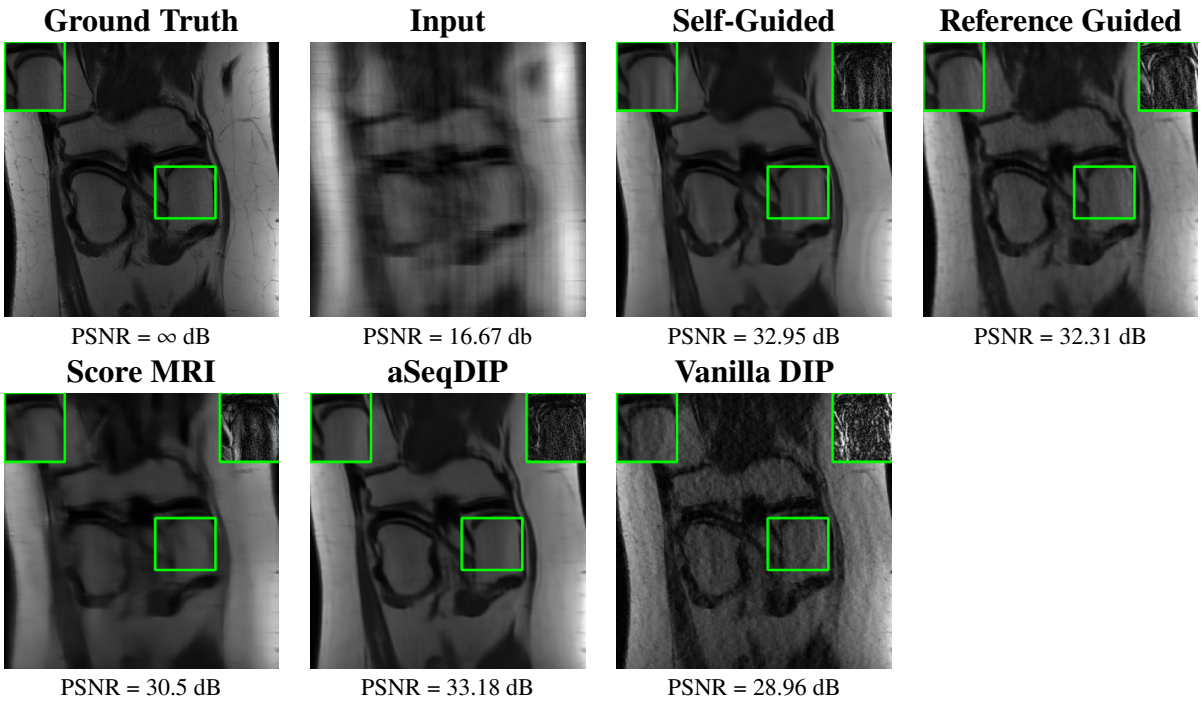


Figure B.7 Visualization of ground-truth and reconstructed images using different methods of a knee image from the fastMRI dataset with 8x k-space undersampling. A region of interest is shown with a green box and its error (magnitude) is shown in the panel on the top right. aSeqDIP provides the clearest reconstruction of image features amongst the methods.

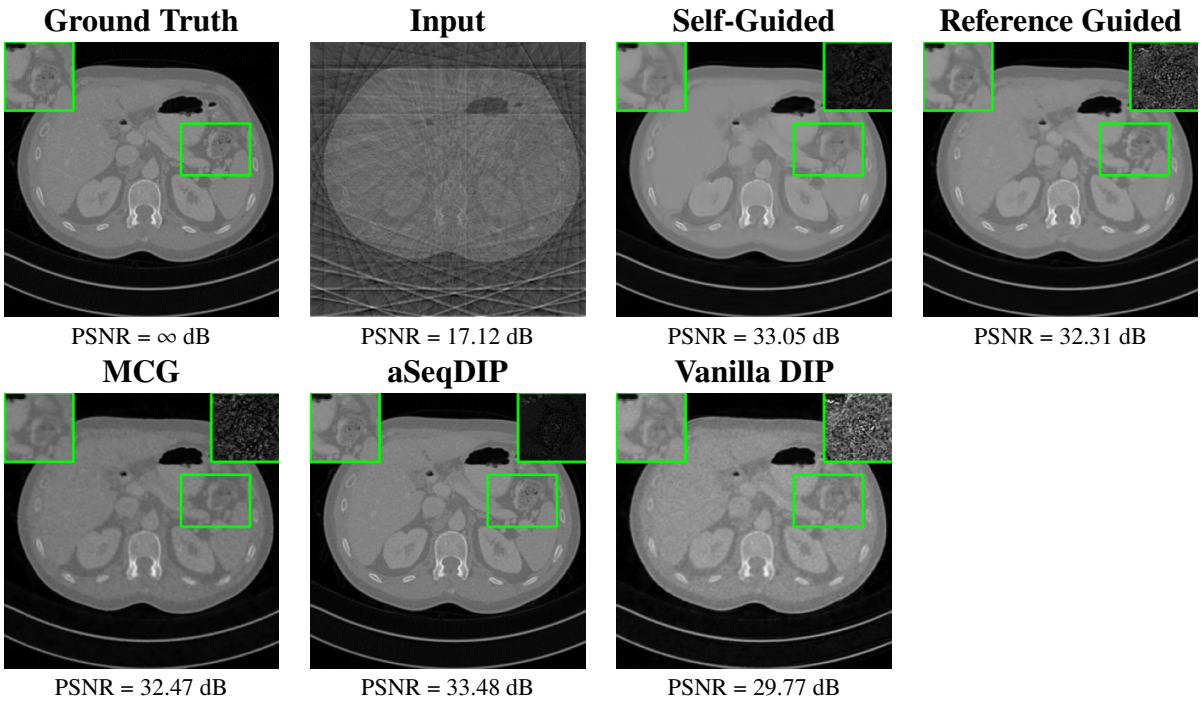


Figure B.8 Visualization of ground-truth and reconstructed images using different methods of a CT scan from the AAPM dataset with 18 views. A region of interest is shown with a green box and its error (magnitude) is shown in the panel on the top right. aSeqDIP provides the sharpest and clearest reconstruction of image features.

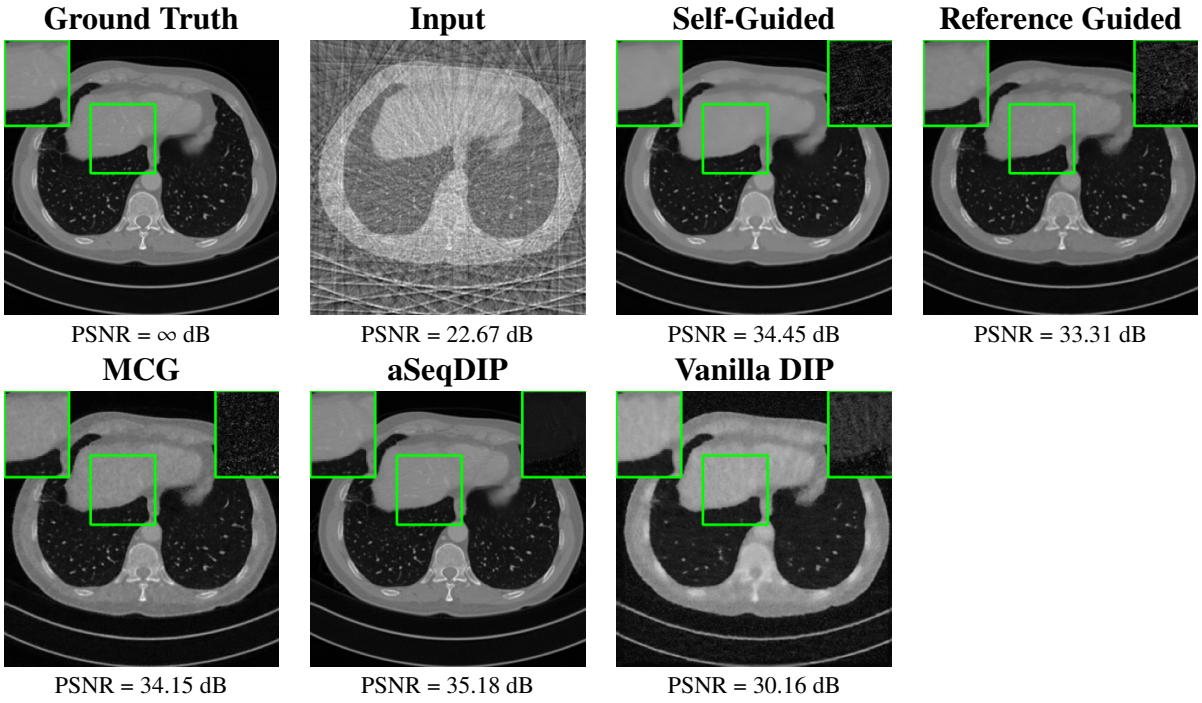


Figure B.9 Visualization of ground-truth and reconstructed images using different methods of a CT scan from the AAPM dataset with with 30 views. A region of interest is shown with a green box and its error (magnitude) is shown in the panel on the top right. aSeqDIP provides better reconstruction of small and low-contrast image features.

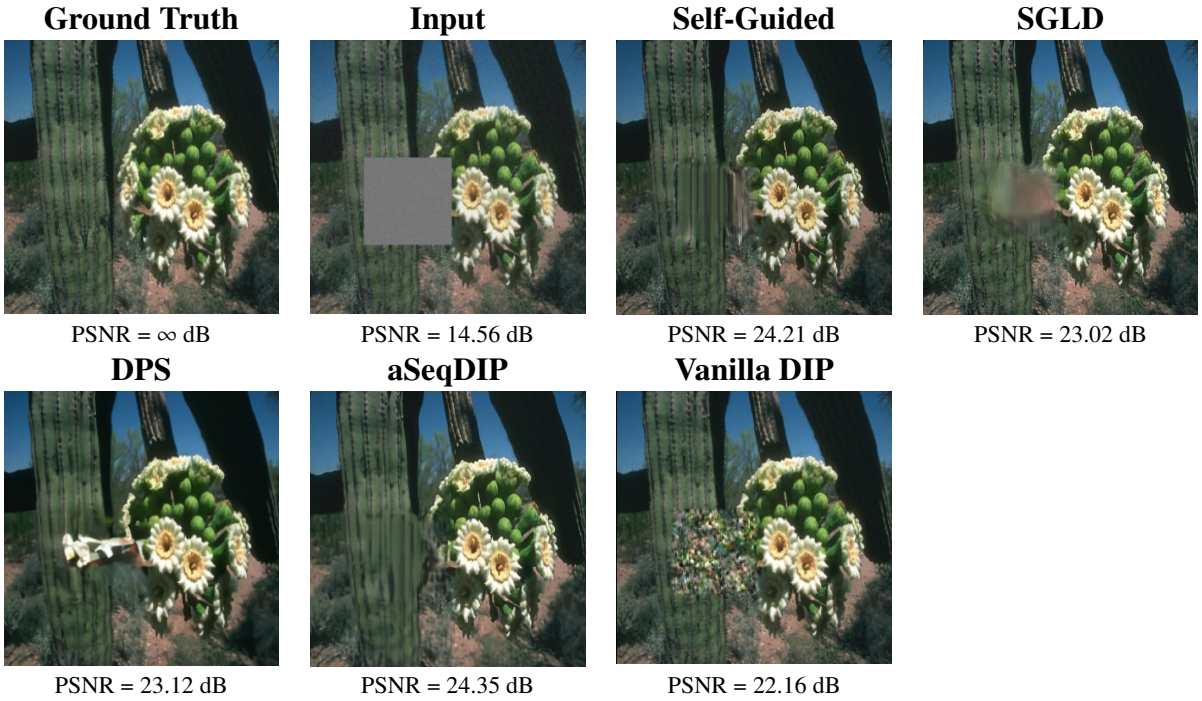


Figure B.10 In-painting example with 0.1 HIAR where image restorations of different methods are given using an example from the CBSD68 dataset. The diffusion-based DPS produces spurious (although sharp) content in the hole region while aSeqDIP much better preserves features in the original ground truth.

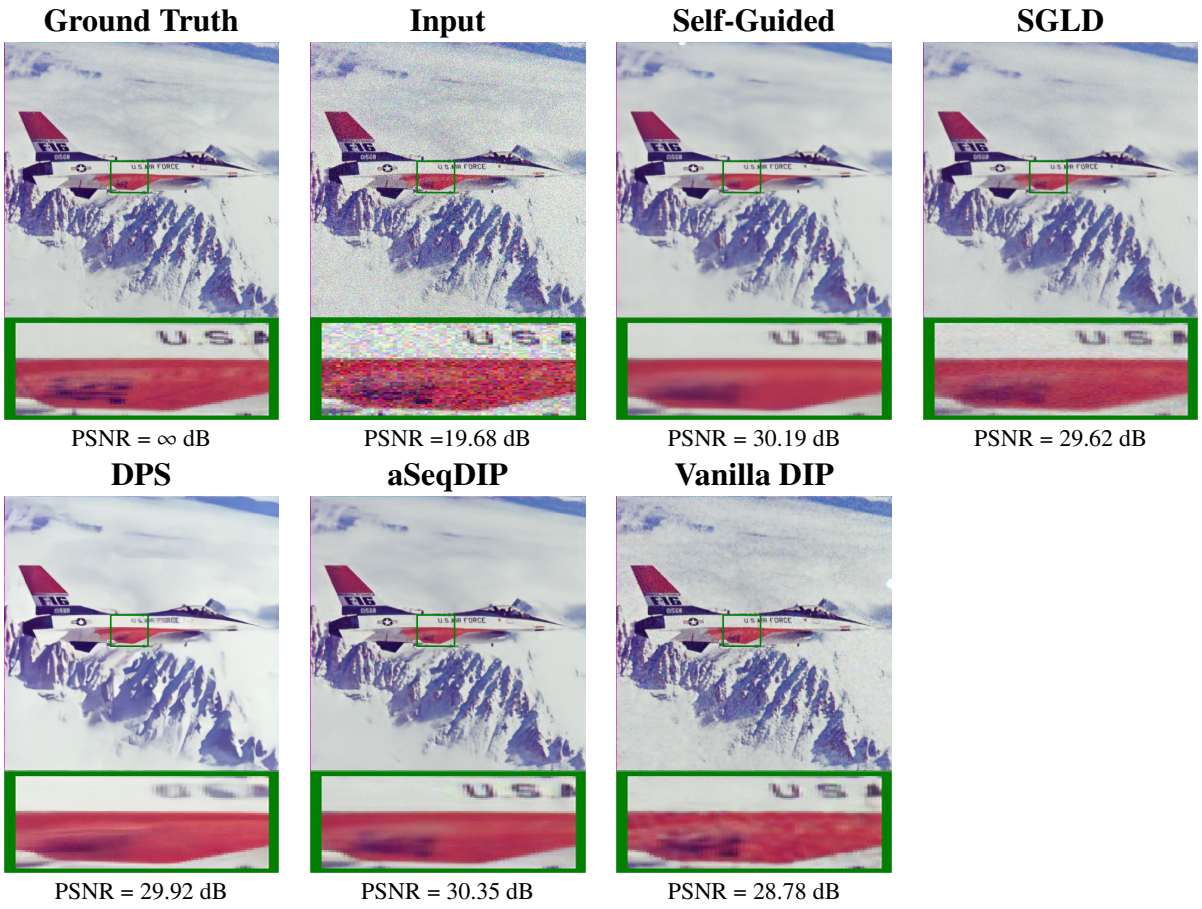


Figure B.11 Denoising example with $\sigma_{\mathrm{d}} = 25$ where image restorations of different methods are given using an example from the CBSD68 dataset. aSeqDIP provides the sharpest and clear reconstruction of image features.

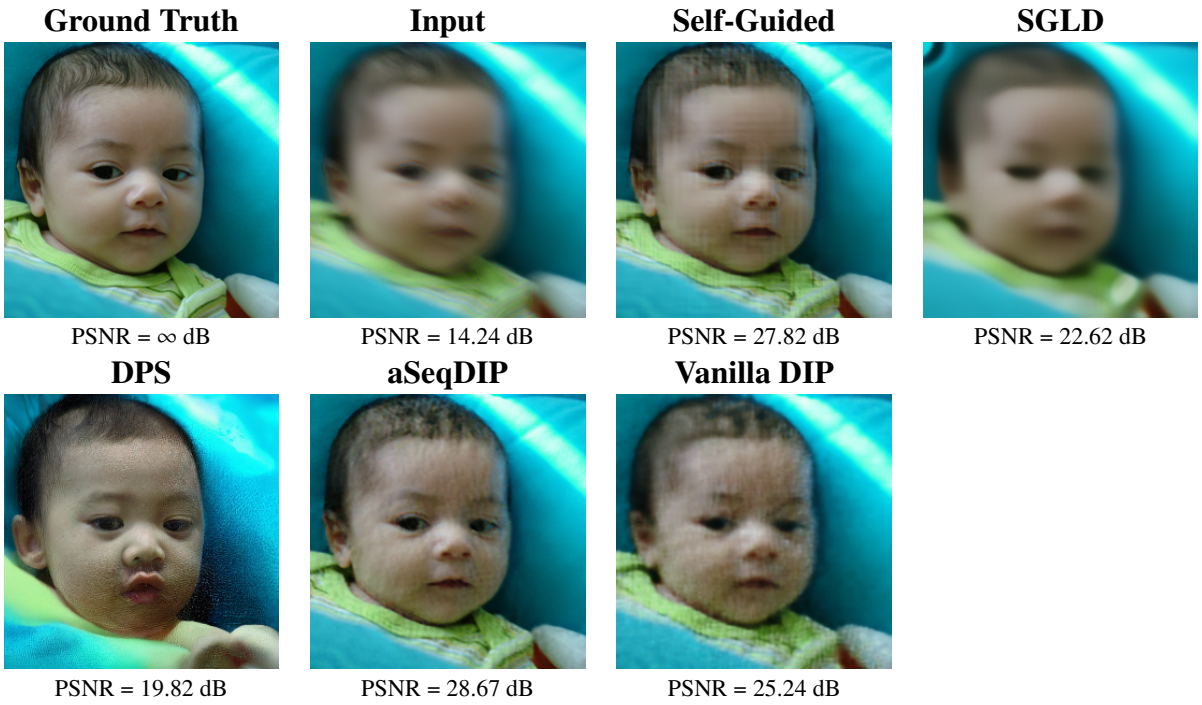


Figure B.12 Non-linear deblurring samples where image restorations of different methods are given using an example from the FFHQ dataset.

APPENDIX C

APPENDIX FOR ROBUST MRI RECONSTRUCTION BY SMOOTHED UNROLLING

C.1 Preliminary of Theorem 6.3.1

Lemma 1. Let $f : \mathbb{R}^d \to \mathbb{R}^m$ be any bounded function. Let $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. We define $g : \mathbb{R}^d \to \mathbb{R}^m$ as

$$g(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\eta}}[f(\boldsymbol{x} + \boldsymbol{\eta})].$$

Then, g is an $\frac{M}{\sqrt{2\pi}\sigma}$-Lipschitz map, where $M = 2\max_{\boldsymbol{x}\in\boldsymbol{R}^d}(\|f(\boldsymbol{x})\|_2)$. In particular, for any $\boldsymbol{x}, \boldsymbol{\delta} \in \mathbb{R}^d$:

$$\|g(\boldsymbol{x}) - g(\boldsymbol{x} + \boldsymbol{\delta})\|_2 \leq \frac{M}{\sqrt{2\pi}\sigma}\|\boldsymbol{\delta}\|_2.$$

Proof. The proof of this bound follows recent work (Wolf, 2019), with a modification on M. Let μ be the probability distribution function of random variable $\boldsymbol{\eta}$. By the change of variables $\boldsymbol{w} = \boldsymbol{x} + \boldsymbol{\eta}$ and $\boldsymbol{w} = \boldsymbol{x} + \boldsymbol{\eta} + \boldsymbol{\delta}$ for the integrals constituting $g(\boldsymbol{x})$ and $g(\boldsymbol{x} + \boldsymbol{\delta})$, we have $\|g(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{\delta})\|_2 = \|\int_{\mathbb{R}^d} f(\boldsymbol{w})[\mu(\boldsymbol{w}-\boldsymbol{x}) - \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})]\, d\boldsymbol{w}\|_2$. Then, we have $\|g(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{\delta})\|_2$

$$\leq \int_{\mathbb{R}^d} \|f(\boldsymbol{w})[\mu(\boldsymbol{w} - \boldsymbol{x}) - \mu(\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta})]\|_2\, d\boldsymbol{w},$$

which is a standard result for the norm of an integral. We further apply Holder's inequality to upper bound $\|g(\boldsymbol{x}) - g(\boldsymbol{x} + \boldsymbol{\delta})\|_2$ with

$$\max_{\boldsymbol{x}\in\boldsymbol{R}^d}(\|f(\boldsymbol{x})\|_2) \int_{\boldsymbol{R}^d} |\mu(\boldsymbol{w} - \boldsymbol{x}) - \mu(\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta})\,| d\boldsymbol{w}. \tag{C.1}$$

Observe that $\mu(\boldsymbol{w} - \boldsymbol{x}) \geq \mu(\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta})$ if $\|\boldsymbol{w} - \boldsymbol{x}\|_2 \leq \|\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta}\|_2$. Let $D = \{\boldsymbol{w} : \|\boldsymbol{w} - \boldsymbol{x}\|_2 \leq \|\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta}\|_2\}$. Then, we can rewrite the above bound as

$$= \max_{\boldsymbol{x}\in\boldsymbol{R}^d}(\|f(\boldsymbol{x})\|_2) \cdot 2\int_D [\mu(\boldsymbol{w} - \boldsymbol{x}) - \mu(\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta})]\, d\boldsymbol{w} \tag{C.2}$$

$$= \frac{M}{2}\left(2\int_D \mu(\boldsymbol{w} - \boldsymbol{x})\, d\boldsymbol{w} - 2\int_D \mu(\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta})\, d\boldsymbol{w}.\right) \tag{C.3}$$

Following Lemma 3 in (Lakshmanan et al., 2008), we obtain the bound

$$2\int_D \mu(\boldsymbol{w} - \boldsymbol{x})\, d\boldsymbol{w} - 2\int_D \mu(\boldsymbol{w} - \boldsymbol{x} - \boldsymbol{\delta})\, d\boldsymbol{w} \leq \frac{2}{\sqrt{2\pi}\sigma}\|\boldsymbol{\delta}\|_2\,, \tag{C.4}$$

which implies that $\|g(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{\delta})\|_2 \leq \frac{2\max_{\boldsymbol{x}\in\boldsymbol{R}^d}(\|f(\boldsymbol{x})\|_2)}{\sqrt{2\pi\sigma}}\|\boldsymbol{\delta}\|_2 = \frac{M}{\sqrt{2\pi\sigma}}\|\boldsymbol{\delta}\|_2$. This completes the proof. □

The proof of this bound closely follows (Wolf, 2019), with a correction on M. We have that

$$
\begin{aligned}
&||g(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{\delta})|| \\
&= || \int_{\mathbb{R}^d} f(\boldsymbol{w})[\mu(\boldsymbol{w}-\boldsymbol{x}) - \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta}]dw|| \\
&\leq || \int_{D^+} f(w)[\mu(\boldsymbol{w}-\boldsymbol{x}) - \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})]dw|| \\
&+ || \int_{D^-} f(\boldsymbol{w})[\mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta}) + \mu(\boldsymbol{w}-\boldsymbol{x})]dw||
\end{aligned}
$$

where $D^+ = \{\boldsymbol{w} : \mu(\boldsymbol{w}-\boldsymbol{x}) > \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})\} = \{\boldsymbol{w} : \|\boldsymbol{w}-\boldsymbol{x}\|^2 < \|\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta}\|^2\}$ and $D^- = \{\boldsymbol{w} : \mu(\boldsymbol{w}-\boldsymbol{x}) < \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})\} = \{\boldsymbol{w} : \|\boldsymbol{w}-\boldsymbol{x}\|^2 > \|\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta}\|^2\}$. We notice that

$$
\int_{D^+}[\mu(\boldsymbol{w}-\boldsymbol{x}) - \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})]dw = \int_{D^-}[\mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta}) - \mu(\boldsymbol{w}-\boldsymbol{x})]dw.
$$

Now we use Jensen's inequality on each norm on the right hand side to get

$$
\begin{aligned}
&||g(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{\delta})|| \\
&\leq \int_{D^+} ||f(\boldsymbol{w})[\mu(\boldsymbol{w}-\boldsymbol{x}) - \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})]||dw \\
&+ \int_{D^-} ||f(\boldsymbol{w})[\mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta}) + \mu(\boldsymbol{w}-\boldsymbol{x})]||dw
\end{aligned}
$$

Now we apply Holder's inequality to get

$$
\begin{aligned}
&\leq \max(f)\cdot(\int_{D^+} ||\mu(w-x) - \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})||dw \\
&+ \int_{D^-} ||\mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta}) - \mu(\boldsymbol{w}-\boldsymbol{x})||dw) \\
&= 2\max(f)\cdot(\int_{D^+} \mu(\boldsymbol{w}-\boldsymbol{x}) - \mu(\boldsymbol{w}-\boldsymbol{x}-\boldsymbol{\delta})dw)
\end{aligned}
$$

C.2 Proof of Theorem 6.3.1

Proof. Assume that the data consistency step in MoDL at iteration n is denoted by $\mathbf{x}_{\mathrm{M}}^{n}(\mathbf{A}^{H}\mathbf{y})$. We will sometimes drop the input and $\mathbf{y}$ dependence for notational simplicity. Then

$$\mathbf{x}_{\mathrm{M}}^{1} = (\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}(\mathbf{A}^{H}\mathbf{y} + \mathcal{D}_{\boldsymbol{\theta}}(\mathbf{A}^{H}\mathbf{y}))\,, \tag{C.5}$$

$$\mathbf{x}_{\mathrm{M}}^{n} = (\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}(\mathbf{A}^{H}\mathbf{y} + \mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_{\mathrm{M}}^{n-1})), \tag{C.6}$$

where $\mathcal{D}_{\boldsymbol{\theta}}$ is the denoiser function. For the sake of simplicity and consistency with the experiments, we use the weighting parameter $\lambda = 1$ (in the data consistency step). We note that the proof works for arbitrary λ. SMUG introduces an iteration-wise smoothing step into MoDL as follows:

$$\mathbf{x}_{\mathrm{S}}^{1} = ((\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}(\mathbf{A}^{H}\mathbf{y} + \mathbb{E}_{\boldsymbol{\eta}_1}[\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{A}^{H}\mathbf{y} + \boldsymbol{\eta}_1)]) \tag{C.7}$$

$$\mathbf{x}_{\mathrm{S}}^{n} = ((\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}(\mathbf{A}^{H}\mathbf{y} + \mathbb{E}_{\boldsymbol{\eta}_n}[\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_{\mathrm{S}}^{n-1} + \boldsymbol{\eta}_n)]) \tag{C.8}$$

$$= (\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}(\mathbf{A}^{H}\mathbf{y}) + \tag{C.9}$$

$$(\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}\mathbb{E}_{\boldsymbol{\eta}_n}[\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{x}_{\mathrm{S}}^{n-1} + \boldsymbol{\eta}_n)],$$

where we apply the expectation to the denoiser $\mathcal{D}_{\boldsymbol{\theta}}$ at each iteration. We use $\boldsymbol{\eta}_n$ to denote the noise during smoothing at iteration n. The robustness error of SMUG after n iterations is $\|\mathbf{x}_{\mathrm{S}}^{n}(\mathbf{A}^{H}\mathbf{y}) - \mathbf{x}_{\mathrm{S}}^{n}(\mathbf{A}^{H}(\mathbf{y} + \boldsymbol{\delta}))\|$. We apply Lemma 1 and properties of the norm (e.g., triangle inequality) to bound $\|\mathbf{x}_{\mathrm{S}}^{n}(\mathbf{A}^{H}\mathbf{y}) - \mathbf{x}_{\mathrm{S}}^{n}(\mathbf{A}^{H}(\mathbf{y} + \boldsymbol{\delta}))\|$ as

$$\leq \|(\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}\mathbf{A}^{H}\boldsymbol{\delta}\| \tag{C.10}$$

$$+ \|(\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1} \cdot \big(\boldsymbol{E}_{\boldsymbol{\eta}_n}[\mathcal{D}_{\boldsymbol{\theta}}\big(\mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^{H}\mathbf{y}) + \boldsymbol{\eta}_n\big)] -$$

$$\boldsymbol{E}_{\boldsymbol{\eta}_n}[\mathcal{D}_{\boldsymbol{\theta}}\big(\mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^{H}(\mathbf{y} + \boldsymbol{\delta})) + \boldsymbol{\eta}_n\big)]\big)\|$$

$$\leq \|(\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}\|_2\|\mathbf{A}^{H}\boldsymbol{\delta}\|_2$$

$$+ \|(\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}\|_2\|\boldsymbol{E}_{\boldsymbol{\eta}_n}[\mathcal{D}_{\boldsymbol{\theta}}\big(\mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^{H}\mathbf{y}) + \boldsymbol{\eta}_n\big)] -$$

$$\boldsymbol{E}_{\boldsymbol{\eta}_n}[\mathcal{D}_{\boldsymbol{\theta}}\big(\mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^{H}(\mathbf{y} + \boldsymbol{\delta})) + \boldsymbol{\eta}_n\big)]\|$$

$$\leq \|(\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}\|_2\|\mathbf{A}^{H}\boldsymbol{\delta}\|_2 + \|(\mathbf{A}^{H}\mathbf{A} + \mathbf{I})^{-1}\|_2 \times \tag{C.11}$$

$$\left(\tfrac{M}{\sqrt{2\pi}\sigma}\right)\|\mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^{H}\mathbf{y}) - \mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^{H}(\mathbf{y} + \boldsymbol{\delta}))\|.$$

Here, $M = 2\max_{\boldsymbol{x}}(\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x})\|)$. Then we plug in the expressions for $\mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^H\mathbf{y})$ and $\mathbf{x}_{\mathrm{S}}^{n-1}(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}))$ (from (C.8)) and bound their normed difference with $\|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\mathbf{A}^H\boldsymbol{\delta}\| + \|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1} \cdot \left(\boldsymbol{E}_{\boldsymbol{\eta}_{n-1}}[\mathcal{D}_{\boldsymbol{\theta}}\left(\mathbf{x}_{\mathrm{S}}^{n-2}(\mathbf{A}^H\mathbf{y})+\boldsymbol{\eta}_{n-1}\right)] - \boldsymbol{E}_{\boldsymbol{\eta}_{n-1}}[\mathcal{D}_{\boldsymbol{\theta}}\left(\mathbf{x}_{\mathrm{S}}^{n-2}(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}))+\boldsymbol{\eta}_{n-1}\right)]\right)\|$. This is bounded above similarly as for (C.10). We repeat this process until we reach the initial $\mathbf{x}_{\mathrm{S}}^0$ on the right hand side. This yields the following bound involving a geometric series.

$$\|\mathbf{x}_{\mathrm{S}}^n(\mathbf{A}^H\mathbf{y}) - \mathbf{x}_{\mathrm{S}}^n(\mathbf{A}^H(\mathbf{y}+\boldsymbol{\delta}))\| \tag{C.12}$$

$$\leq \|\mathbf{A}^H\boldsymbol{\delta}\|_2 \left(\sum_{j=1}^n \|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2^j \cdot \left(\frac{M}{\sqrt{2\pi}\sigma}\right)^{j-1}\right) + \|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2^n \left(\frac{M}{\sqrt{2\pi}\sigma}\right)^n \|\mathbf{A}^H\boldsymbol{\delta}\|_2 \tag{C.13}$$

$$\leq \|\mathbf{A}\|_2\|\boldsymbol{\delta}\|_2\|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2 \left(\frac{1-\left(\frac{M}{\sqrt{2\pi}\sigma}\right)^n \|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2^n}{1-\frac{M}{\sqrt{2\pi}\sigma}\|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2}\right) + \|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2^n \left(\frac{M}{\sqrt{2\pi}\sigma}\right)^n \|\mathbf{A}\|_2\|\boldsymbol{\delta}\|_2 \leq C_n\|\boldsymbol{\delta}\|_2, \tag{C.14}$$

where we used the geometric series formula, and $C_n = \alpha\|\mathbf{A}\|_2\left(\frac{1-\left(\frac{M\alpha}{\sqrt{2\pi}\sigma}\right)^n}{1-\frac{M\alpha}{\sqrt{2\pi}\sigma}}\right) + \|\mathbf{A}\|_2\left(\frac{M\alpha}{\sqrt{2\pi}\sigma}\right)^n$, with $\alpha = \|(\mathbf{A}^H\mathbf{A}+\mathbf{I})^{-1}\|_2$. □

APPENDIX D

APPENDIX FOR STEP-WISE TRIPLE-CONSISTENT DIFFUSION SAMPLING FOR INVERSE PROBLEMS

In the Appendix, we start by showing the equivalence between the second formula in (2.10) and (8.5) (Appendix D.1). Then, we discuss the known limitations and future extensions of SITCOM (Appendix D.2). Subsequently, we present experiments to highlight the impact of the proposed backward consistency (Appendix D.3). This is followed by a discussion on phase retrieval (Appendix D.4). In Appendix D.5, we provide further comparison results, and in Appendix D.6, we perform ablation studies to examine the effects of the stopping criterion and other components/hyper-parameters in SITCOM. Appendix D.7 covers the implementation details of tasks and baselines, followed by examples of restored images (Appendix D.8).

D.1 Derivation of (8.5)

From (Luo, 2022), we have

$$\mathbf{s}_\theta(\mathbf{x}_t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\ . \tag{D.1}$$

Rearranging the Tweedie's formula in (2.11) to solve for $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ yields

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t)}{\sqrt{1-\bar{\alpha}_t}}\ . \tag{D.2}$$

Now, we substitute into the recursive equation for $\mathbf{x}_{t-1}$:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left[\mathbf{x}_t + \beta_t \mathbf{s}_\theta(\mathbf{x}_t, t)\right] + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.3}$$

$$= \frac{1}{\sqrt{1-\beta_t}}\left[\mathbf{x}_t + \beta_t\left(-\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)\right] + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.4}$$

$$= \frac{1}{\sqrt{1-\beta_t}}\left[\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right] + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.5}$$

$$= \frac{1}{\sqrt{1-\beta_t}}\left[\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\left(\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t)}{\sqrt{1-\bar{\alpha}_t}}\right)\right] + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.6}$$

$$= \frac{1}{\sqrt{1-\beta_t}}\left[\mathbf{x}_t - \frac{\beta_t}{1-\bar{\alpha}_t}\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t)\right)\right] + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.7}$$

$$= \frac{1}{\sqrt{1-\beta_t}}\left[\left(1 - \frac{\beta_t}{1-\bar{\alpha}_t}\right)\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_t}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t)\right] + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.8}$$

$$= \frac{(1-\bar{\alpha}_t-\beta_t)}{\sqrt{1-\beta_t}\,(1-\bar{\alpha}_t)}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_t}\beta_t}{\sqrt{1-\beta_t}\,(1-\bar{\alpha}_t)}\hat{\mathbf{x}}_0(\mathbf{x}_t) + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.9}$$

$$= \frac{(\alpha_t-\bar{\alpha}_t)}{\sqrt{\alpha_t}\,(1-\bar{\alpha}_t)}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_t}\beta_t}{\sqrt{\alpha_t}\,(1-\bar{\alpha}_t)}\hat{\mathbf{x}}_0(\mathbf{x}_t) + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.10}$$

$$= \frac{\left(\sqrt{\alpha_t}-\sqrt{\alpha_t}\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t) + \sqrt{\beta_t}\boldsymbol{\eta}_t \tag{D.11}$$

$$= \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t) + \sqrt{\beta_t}\boldsymbol{\eta}_t\,, \tag{D.12}$$

which is equivalent to the second formula in (2.10).

D.2 Limitations & Future Work

In SITCOM, the stopping criterion parameter is set slightly higher than the level of measurement noise, determined by $\sigma_{\mathbf{y}}$. As a result, our method requires access to (or estimation of) the measurement noise prior to the restoration process. Knowledge of noise level is also assumed in other works such as DAPS (Zhang et al., 2024a). In practice, classical approaches, such as (Liu et al., 2006; Chen et al., 2015), can be used to estimate the noise.

Additionally, the stated conditions and proposed sampler are limited to the non-blind setting, as SITCOM assumes full access to the forward model, unlike works such as (Chung et al., 2023a), which perform both image restoration and forward model estimation.

For future work, in addition to addressing the aforementioned limitations, we aim to extend

SITCOM to the latent space and explore its applicability in medical image reconstruction.

D.3 Impact of the proposed Backward Consistency

Here, we demonstrate the impact of the proposed backward diffusion consistency in SITCOM using two experiments.

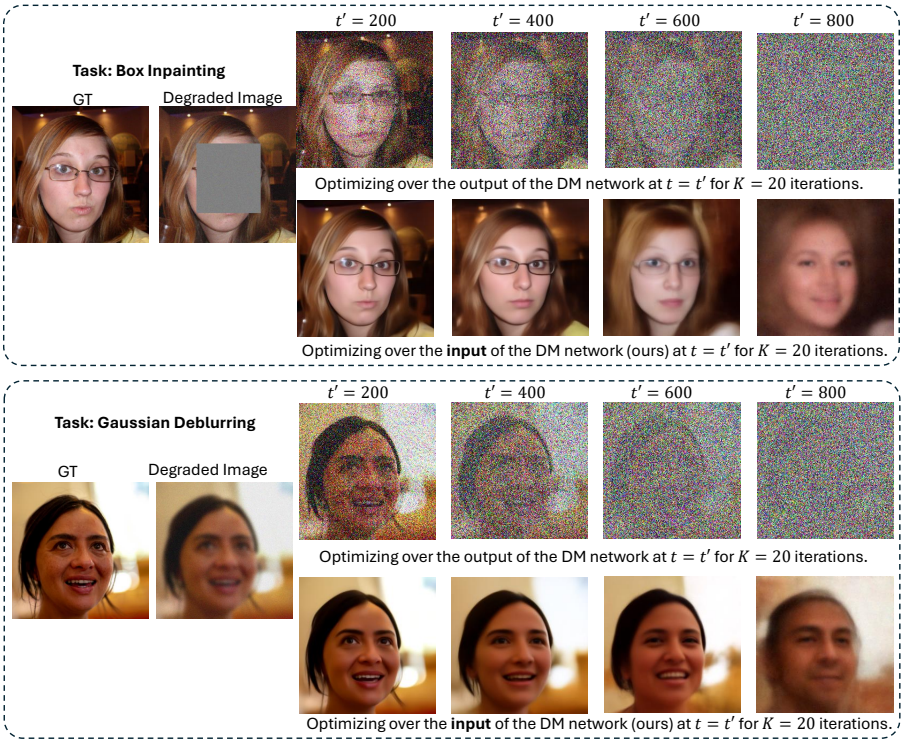


Figure D.1 Results of applying optimization-based measurement consistency, for which the optimization variable is the DM output (resp. input), are shown in the first (resp. second) row for each task: Box Inpainting (*top*) and Gaussian Deblurring (*bottom*).

First, for the box-painting task, we compare optimizing over the input to the DM (as in SITCOM) with optimizing over the output of the DM network (as is done in DCDP (Li et al., 2024) and DAPS (Zhang et al., 2024a)) at time steps $t' \in \{200, 400, 600\}$. For each case (selection of t'), we start from $t = T$ and run SITCOM with a step size of $\lfloor \frac{T}{N} \rfloor$. At $t = t'$, given $\mathbf{x}_{t'}$, we perform two separate optimizations with intializing the optimization variable as $\mathbf{x}_{t'}$: one iteratively over the DM network input (ours) and another iteratively over the DM network output (i.e., (8.4) but without the regularization), both running until convergence (i.e., when the loss stops decreasing). For our approach, the result of the optimization from ($\mathtt{S}_1$) is used as input to Tweedie's formula in ($\mathtt{S}_2$) to compute the posterior mean $\hat{\mathbf{x}}_0' = \hat{\mathbf{x}}_0(\mathbf{v}_t)$. For the case of optimizing over the DM output, we use

(8.4) without regularization. Figure 8.2 shows the results at different time steps. The consistency between the ground truth and the unmasked regions of the estimated images suggest the convergence of the measurement consistency. As observed, SITCOM produces significantly less artifacts in the masked region when compared to optimizing over the output. This is evident both at earlier time steps ($t' = 600$) and later steps ($t' = 400$ and $t' = 200$).

For the second experiment, the goal is to show that SITCOM requires much smaller number of optimization steps to remove the noise as compared to the case where the optimization variable is the output of the DM network. The results are given in Figure D.1, where we repeat the above experiment with two tasks: Box-inpainting (*top*) and Gaussian Deblurring (*bottom*), this time using a fixed number of optimization steps for both SITCOM, and optimizing over the DM output. Specifically, we run SITCOM from $t = T$ to $t = t' + 1$. Then, we apply $K = 20$ iterations (the setting in SITCOM) in ($\mathtt{S}_1$), and $K = 20$ when optimizing (8.4) (without regularization) where measurement noise is $\sigma_{\mathbf{y}} = 0.05$. As shown, compared to optimizing over the DM output, SITCOM significantly reduces noise across all considered t', underscoring the effect of the proposed backward diffusion consistency when optimizing over the DM input.

D.4 Discussion on Phase Retrieval

As discussed in our experimental results section, for the phase retrieval task, we report the best results from 4 independent runs, following the convention in (Chung et al., 2023b; Zhang et al., 2024a). For the phase retrieval results of Table 8.1 and Table D.1 (given in Appendix D.5), we use this approach across all baselines where the run-time is reported for one run.

The forward model for phase retrieval is adopted from DPS where the inverse problem is generally more challenging compared to other image restoration tasks. This increased difficulty arises from the presence of multiple modes that can yield the same measurements (Zhang et al., 2024a).

In Figure D.2, we present two examples comparing SITCOM, DPS, and DAPS. For each ground truth image, we show four results from which the best one was selected. In the first column, SITCOM avoids significant artifacts, while DAPS produces one image rotated by 180 degrees. In the second

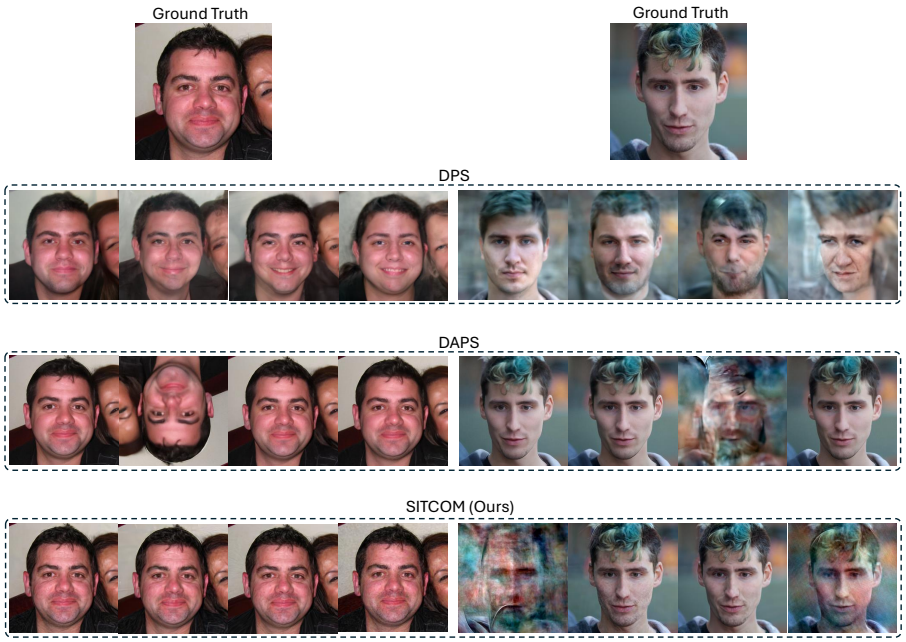


Figure D.2 Results of Phase Retrieval on two images (top row) from the FFHQ dataset. Rows 2, 3, and 4 correspond to the results of DPS, DAPS, and SITCOM (ours), respectively.

column, both SITCOM and DAPS exhibit one run with severe artifacts. However, the last image from SITCOM does exhibit more artifacts compared to the second worst-case result from DAPS. Additionally, the DPS results show severe perceptual differences in both cases, with artifacts being particularly noticeable in the second column.

D.5 Additional Comparison Results

In Table D.1, we present the average PSNR, SSIM, LPIPS, and run-time (minutes) of DPS, DAPS, DDNM, and SITCOM using the FFHQ and ImageNet datasets for which the measurement noise level is set to $\sigma_{\mathbf{y}} = 0.01$ (different from Table 8.1). The goal of these results is to evaluate our method and baselines under less noisy settings.

Overall, we observe similar trends to those discussed in Section 5.3 for Table 8.1. On the FFHQ dataset, SITCOM achieves higher average PSNR values compared to the baselines across all tasks, with improvements exceeding 1 dB in 5 out of 8 tasks. For the ImageNet dataset, we observe more than 1 dB improvement on the non-linear deblurring task, while for the remaining tasks, the improvement is less than 1 dB, except for Gaussian deblurring (where SITCOM underperforms by 0.22 dB) and phase retrieval (underperforming by 0.36 dB).

Task	Method	FFHQ				ImageNet			
		PSNR (↑)	SSIM (↑)	LPIPS (↓)	Run-time (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	Run-time (↓)
Super Resolution 4×	DPS	25.20±1.22	0.806±0.044	0.242±0.102	1.31±0.44	24.45±0.89	0.792±0.052	0.331±0.089	2.33±0.40
	DAPS	29.6±0.67	0.871±0.034	**0.132**±0.088	1.24±0.43	25.98±0.74	0.794±0.09	0.234±0.089	2.10±1.02
	DDNM	28.82±0.67	0.851±0.043	0.188±0.13	1.07±0.35	24.67±0.78	0.771±0.06	0.432±0.34	1.38±0.55
	Ours	**30.95**±0.89	**0.872**±0.045	0.137±0.046	**0.50**±0.34	**26.89**±0.86	**0.802**±0.057	**0.224**±0.056	**1.34**±0.45
Box In-Painting	DPS	23.56±0.78	0.762±0.034	0.191±0.087	1.52±0.43	20.22±0.67	0.69±0.034	0.297±0.077	1.55±0.44
	DAPS	24.41±0.67	0.791±0.034	0.129±0.067	1.33±0.42	21.79±0.34	0.734±0.045	0.214±0.034	2.44±0.34
	DDNM	24.67±0.067	0.788±0.024	0.229±0.055	1.02±0.42	21.99±0.54	0.737±0.034	0.315±0.022	1.42±0.45
	Ours	**24.97**±0.55	**0.804**±0.045	**0.118**±0.022	**0.37**±0.34	**22.23**±0.44	**0.745**±0.034	**0.208**±0.023	**1.23**±0.44
Random In-Painting	DPS	28.77±0.56	0.847±0.034	0.191±0.023	1.55±0.34	24.57±0.45	0.775±0.023	0.318±0.26	2.12±0.30
	DAPS	31.56±0.45	0.905±0.013	0.094±0.012	1.42±0.45	28.86±0.67	0.877±0.021	0.131±0.044	2.01±0.34
	DDNM	30.56±0.56	0.902±0.013	0.116±0.023	1.25±0.42	30.12±0.45	0.917±0.012	0.124±0.032	1.89±0.23
	Ours	**33.02**±0.44	**0.919**±0.012	**0.0912**±0.013	**0.47**±0.34	**30.67**±0.45	**0.918**±0.013	**0.118**±0.012	**1.40**±0.34
Gaussian Deblurring	DPS	25.78±0.68	0.831±0.034	0.202±0.014	1.33±0.44	22.45±0.42	0.778±0.067	0.344±0.041	2.12±0.44
	DAPS	29.67±0.45	0.889±0.045	0.163±0.033	2.15±0.37	26.34±0.55	0.836±0.034	0.244±0.023	2.22±0.43
	DDNM	28.56±0.45	0.872±0.024	0.211±0.034	1.24±0.34	**28.44**±0.021	0.882±0.021	0.267±0.00	1.76±0.33
	Ours	**32.12**±0.34	**0.913**±0.024	**0.139**±0.045	**0.45**±0.25	28.22±0.45	**0.891**±0.014	**0.216**±0.021	**1.34**±0.25
Motion Deblurring	DPS	23.78±0.78	0.742±0.042	0.265±0.024	1.65±0.34	22.33±0.727	0.726±0.034	0.352±0.00	2.21±0.40
	DAPS	30.78±0.56	0.892±0.034	0.146±0.023	1.44±0.34	28.24±0.62	0.867±0.023	0.191±0.017	2.12±0.44
	Ours	**32.34**±0.44	**0.908**±0.028	**0.135**±0.028	**0.52**±0.34	**29.12**±0.38	**0.882**±0.025	**0.182**±0.025	**1.45**±0.31
Phase Retrieval	DPS	17.56±2.15	0.681±0.056	0.392±0.021	1.52±0.42	16.77±1.78	0.651±0.076	0.442±0.037	2.18±0.38
	DAPS	31.45±2.78	0.909±0.035	0.109±0.044	1,85±0.32	**26.12**±2.12	0.802±0.023	0.247±0.034	2.32±0.35
	Ours	**31.88**±2.89	**0.921**±0.067	**0.102**±0.078	**0.54**±0.45	25.76±1.78	**0.813**±0.032	**0.238**±0.067	**1.31**±0.45
Non-Uniform Deblurring	DPS	23.78±2.23	0.761±0.051	0.269±0.064	1.56±0.45	22.97±1.57	0.781±0.023	0.302±0.089	2.34±0.44
	DAPS	28.89±1.67	0.845±0.057	0.150±0.056	1.41±0.37	28.02±1.15	0.831±0.082	0.162±0.034	2.23±0.56
	Ours	**31.09**±0.89	**0.911**±0.056	**0.132**±0.45	**0.56**±0.37	**29.56**±0.78	**0.844**±0.045	**0.147**±0.042	**1.34**±0.44
High Dynamic Range	DPS	23.33±1.34	0.734±0.049	0.251±0.078	1.34±0.42	19.67±0.056	0.693±0.034	0.498±0.112	2.34±0.41
	DAPS	27.58±0.829	0.828±0.00	0.161±0.067	1.26±0.44	26.71±0.088	0.802±0.032	0.172±0.066	2.12±0.32
	Ours	**28.52**±0.89	**0.844**±0.045	**0.148**±0.035	**0.51**±0.42	**27.56**±0.78	**0.825**±0.037	**0.162**±0.046	**1.45**±0.41

Table D.1 Average PSNR, SSIM, LPIPS, and run-time (minutes) of SITCOM and baselines using 100 test images from FFHQ and 100 test images from ImageNet with a **measurement noise level** of $\sigma_{\mathbf{y}} = 0.01$. The first five tasks are linear, while the last three tasks are non-linear (underlined). For each task and dataset combination, the best results are bolded, and the second-best results are underlined. Values after ± represent the standard deviation. All results were obtained using a **single RTX5000 GPU** machine. For phase retrieval, the run-time is reported for the best result out of four independent runs. This is applied for SITCOM and baselines.

In terms of run-time, generally, SITCOM significantly outperforms DDNM, DPS, and DAPS, with all methods evaluated on a single RTX5000 GPU. For the FFHQ dataset, SITCOM is at least twice as fast when compared to baselines. On ImageNet, SITCOM consistently requires much less run-time compared to DPS and DAPS. When compared to DDNM, SITCOM's run-time is similar or slightly lower. For example, on the super-resolution task, both SITCOM and DDNM average 1.34 minutes, but SITCOM achieves over a 2 dB improvement.

In Table D.2, we report the average PSNR and LPIPS results using three more baselines: Denoising Diffusion Restoration Models (DDRM) (Kawar et al., 2022), Plug-and-Play (PnP) ADMM (Chan et al., 2016) (a non diffusion-based solver), and Regularization by Denoising with

Task	Method	FFHQ		ImageNet	
		PSNR (↑)	LPIPS (↓)	PSNR (↑)	LPIPS (↓)
Super Resolution 4×	DDRM (Kawar et al., 2022)	27.65	0.210	25.21	0.284
	PnP-ADMM (Chan et al., 2016)	23.48	0.725	22.18	0.724
	SITCOM (ours)	**30.68**	**0.142**	**26.35**	**0.232**
Box In-Painting	DDRM (Kawar et al., 2022)	22.37	0.159	19.45	0.229
	PnP-ADMM (Chan et al., 2016)	13.39	0.775	12.61	0.702
	SITCOM (ours)	**24.68**	**0.121**	**21.88**	**0.214**
Random In-Painting	DDRM (Kawar et al., 2022)	25.75	0.218	23.23	0.325
	PnP-ADMM (Chan et al., 2016)	20.94	0.724	20.03	0.680
	SITCOM (ours)	**32.05**	**0.095**	**29.60**	**0.127**
Gaussian Deblurring	DDRM (Kawar et al., 2022)	23.36	0.236	23.86	0.341
	PnP-ADMM (Chan et al., 2016)	21.31	0.751	20.47	0.729
	SITCOM (ours)	**30.25**	**0.235**	**27.40**	**0.236**
Motion Deblurring	PnP-ADMM (Chan et al., 2016)	23.40	0.703	24.23	0.684
	SITCOM (ours)	**30.34**	**0.148**	**28.65**	**0.189**
Phase Retrieval	RED-Diff (Mardani et al., 2023)	15.60	0.596	14.98	0.536
	SITCOM (ours)	**30.97**	**0.112**	**25.45**	**0.246**
Non-Uniform Deblurring	RED-Diff (Mardani et al., 2023)	**30.86**	0.160	**30.07**	0.211
	SITCOM (ours)	30.12	**0.145**	28.78	**0.160**
High Dynamic Range	RED-Diff (Mardani et al., 2023)	22.16	0.258	22.03	0.274
	SITCOM (ours)	**27.98**	**0.158**	**26.97**	**0.167**

Table D.2 Average PSNR and LPIPS results of our method and other baselines over 100 FFHQ and 100 ImageNet test images. The measurement noise setting is $\sigma_{\mathbf{y}} = 0.05$. The results of DDRM and PnP-ADMM (resp. RED-Diff) are sourced from Tables 1 and 3 (resp. 2 and 4) in (Zhang et al., 2024a). The remaining results are as given in Table 8.1 of Section 5.3.

Diffusion (RED-Diff) (Mardani et al., 2023). The results of DDRM, PnP-ADMM, and RED-Diff are sourced from (Zhang et al., 2024a). DDRM and PnP-ADMM present results for linear tasks whereas RED-Diff is used for the non-linear tasks. The results of SITCOM are as reported in Table 8.1.

When compared to DDRM and PnP-ADMM, SITCOM demonstrates notable improvements in both PSNR and LPIPS across all tasks and datasets. For instance, SITCOM achieves over a 5 dB improvement in random in-painting on both datasets. Compared to RED-Diff, SITCOM outperforms by 5 dB on FFHQ and more than 10 dB on ImageNet for phase retrieval. A similar trend is observed in the High Dynamic Range task. For non-linear non-uniform deblurring, although SITCOM performs better in terms of LPIPS, it reports approximately 1 dB (FFHQ) and 2 dB (ImageNet) less PSNR than RED-Diff, all without requiring external denoisers.

D.6 Ablation Studies

D.6.1 Effect of the number of Optimization steps K, & the number of Sampling steps N

In this subsection, we perform an ablation study on the number of optimization steps, K, and the number of sampling steps, N. Specifically, for the tasks of Super Resolution, Motion Deblurring, and Random In-painting, we run SITCOM using combinations from $N \in \{10, 20, 30\}$ and $K \in \{20, 30, 40\}$. The average PSNR results over 20 test images from the FFHQ dataset are presented in Table D.3. As shown, for the first three tasks, SITCOM consistently achieves strong PSNR scores across all (N, K) pairs, demonstrating that its performance is not very sensitive to variations in (N, K) within these ranges as the results vary by nearly 1 dB. For High Dynamic Range tasks, we observe that the best results are obtained with $(N, K) = (20, 40)$. The selected (N, K) values for our main results are listed in Table D.5 of Appendix D.6.4.

(N, K)	$(10, 20)$	$(10, 30)$	$(10, 40)$	$(20, 20)$	$(20, 30)$	$(20, 40)$	$(30, 20)$	$(30, 30)$	$(30, 40)$
Super Resolution 4×	29.654	29.771	29.815	29.913	29.952	29.961	30.009	30.027	30.033
Motion Deblurring	29.976	30.820	31.264	31.259	31.380	30.452	31.282	30.624	30.438
Random Inpainting	33.428	34.444	34.699	34.546	34.558	34.574	34.619	34.634	34.639
High Dynamic Range	25.902	26.290	27.873	26.957	27.104	27.874	27.171	27.127	26.806

Table D.3 Effect of the number of sampling steps (N) and optimization steps per sampling iteration (K) on the tasks listed in the first column for SITCOM. The reported PSNR values are averaged over 20 FFHQ test images.

D.6.2 Effect of the Regularization Parameter λ

In this subsection, we perform an ablation study to assess the impact of the regularization parameter, λ, in SITCOM. Table D.4 shows the results across four tasks using various λ values. Aside from phase retrieval, the effect of λ is minimal. We hypothesize that initializing the optimization variable in $(\mathtt{S}_1)$ with $\mathbf{x}_t$ is sufficient to enforce forward diffusion consistency in **C3**. Therefore, we set $\lambda = 1$ for phase retrieval and $\lambda = 0$ for the other tasks.

Additionally, for all tasks other than phase retrieval, we observed that when $\lambda = 0$, the restored images exhibit enhanced high-frequency details. For visual examples, see the results of $\lambda = 0$ versus $\lambda = 1$ in Figure D.3.

λ	0	0.05	0.5	1	1.5
Super Resolution 4×	29.952	29.968	29.464	29.550	29.288
Motion Deblurring	31.380	31.393	31.429	31.382	31.150
Random Inpainting	34.559	34.537	34.523	34.500	34.301
Phase Retrieval	31.678	31.892	32.221	32.342	32.124

Table D.4 Ablation Study on the impact of the regularization parameter λ.

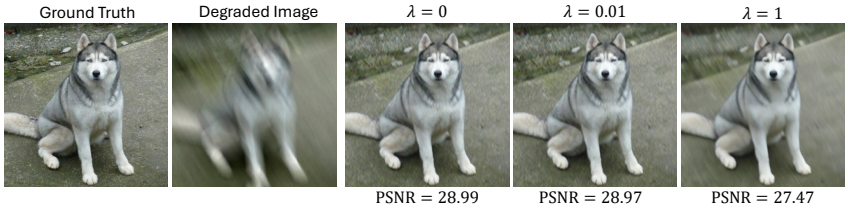


Figure D.3 Results of running SITCOM using different regularization parameters in ($\mathtt{S}_1$) for the task of Motion deblurring.

D.6.3 Impact of the Stopping criterion For Noisy Measurements

In this subsection, we demonstrate the impact of applying the stopping criterion in SITCOM when handling measurement noise. For the tasks of super resolution and motion deblurring, we run SITCOM with and without the stopping criterion for the case of $\sigma_{\mathbf{y}} = 0.05$. The results are presented in Figure D.4. As shown, for both tasks, using the stopping criterion (i.e., $\delta > 0$) not only improves PSNR values compared to the case of $\delta = 0$, but also visually reduces additive noise in the restored images. This is because, without the stopping criterion , the measurement consistency enforced by the optimization in ($\mathtt{S}_1$) tends to fit the noise in the measurements.

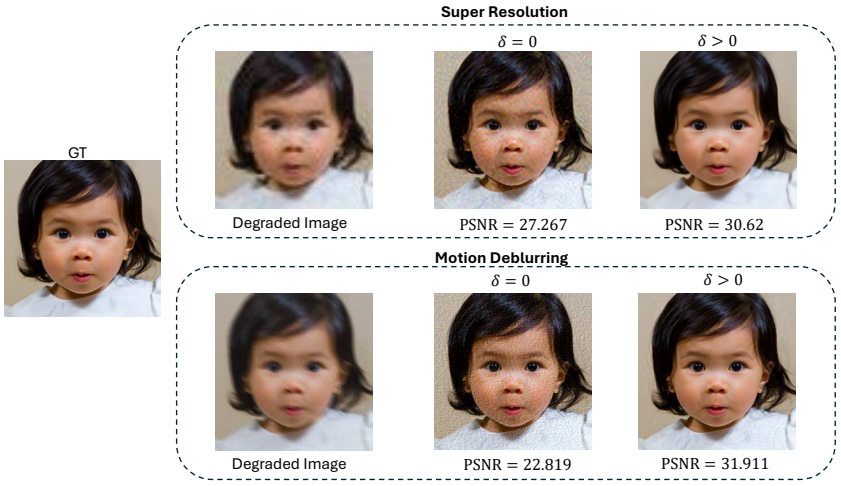


Figure D.4 Impact of the stopping criterion in preventing noise overfitting. For the most right column, δ is set as in Table D.5.

D.6.4 Complete List of hyper-parameters in SITCOM

Table D.5 summarizes the hyper-parameters used for each task in our experiments, as determined by the ablation studies in the previous subsections. Notably, the same set of hyper-parameters is applied to both the FFHQ and ImageNet datasets.

Task	Sampling Steps N	Optimization Steps K	Regularization λ	Stopping criterion δ for $\sigma_{\mathrm{y}} \in \{0.05, 0.01\}$
Super Resolution 4×	20	20	0	$\{0.051\sqrt{m_{\mathrm{SR}}}, 0.011\sqrt{m_{\mathrm{SR}}}\}$
Box In-Painting	20	20	0	$\{0.051\sqrt{m}, 0.011\sqrt{m}\}$
Random In-Painting	20	30	0	$\{0.051\sqrt{m}, 0.011\sqrt{m}\}$
Gaussian Deblurring	20	30	0	$\{0.051\sqrt{m}, 0.011\sqrt{m}\}$
Motion Deblurring	20	30	0	$\{0.051\sqrt{m}, 0.011\sqrt{m}\}$
Phase Retrieval	20	30	1	$\{0.051\sqrt{m_{\mathrm{PR}}}, 0.011\sqrt{m_{\mathrm{PR}}}\}$
Non-Uniform Deblurring	20	30	0	$\{0.051\sqrt{m}, 0.011\sqrt{m}\}$
High Dynamic Range	20	40	0	$\{0.051\sqrt{m}, 0.011\sqrt{m}\}$

Table D.5 Hyper-parameters of SITCOM for every task considered in this paper. The same set of hyper-parameters is used for FFHQ and ImageNet. The learning rate in Algorithm 8.1 is set to $\gamma = 0.01$ for all tasks, datasets, and measurement noise levels. For the stopping criterion column, $m_{\mathrm{SR}} = 64 \times 64 \times 3$, $m = 256 \times 256 \times 3$, and $m_{\mathrm{PR}} = 384 \times 384 \times 3$

.

D.7 Detailed Implementation of tasks and Baselines

The forward models of all tasks are adopted from DPS. We refer the reader to Appendix B of (Chung et al., 2023b) for details. For baselines, we used the codes provided by the authors of each paper: DPS, DDNM, DAPS, and DCDP. Default configurations are used for each task.

D.8 Qualitative results

Figure D.5 presents results with SITCOM, DPS, and DAPS using ImageNet. See also Figure D.6, Figure D.7, Figure D.8, and Figure D.9 for more images.

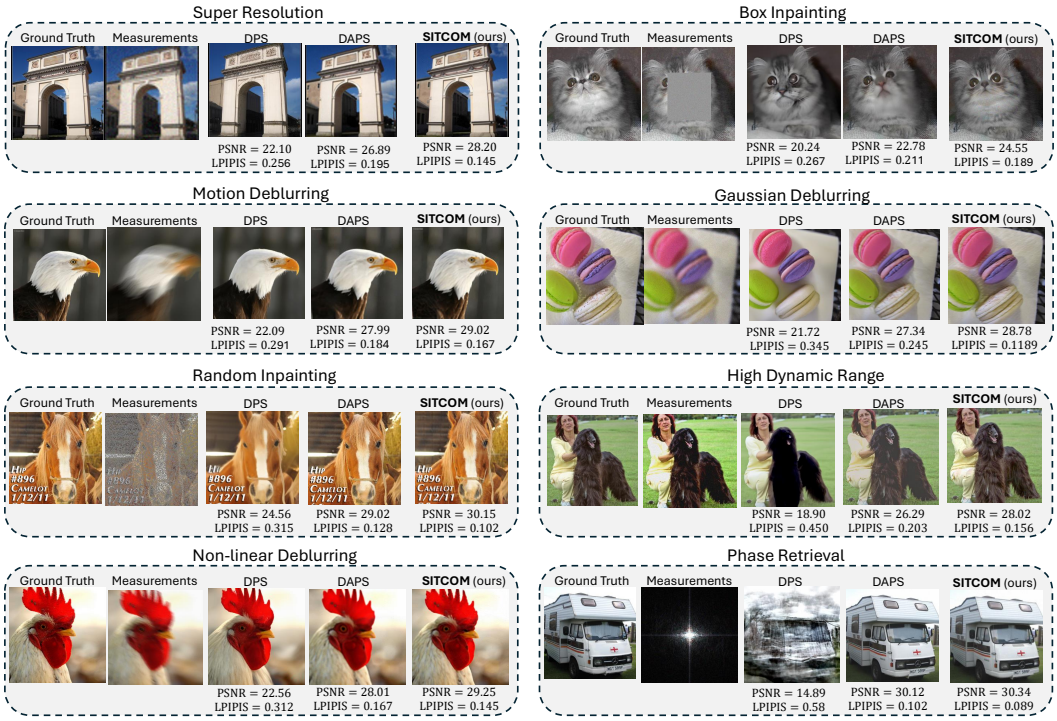


Figure D.5 **Qualitative results on the ImageNet dataset** for five linear tasks and three non-linear tasks under measurement noise of $\sigma_{\mathbf{y}} = 0.05$. The PSNR and LPIPS values are given below each restored image.

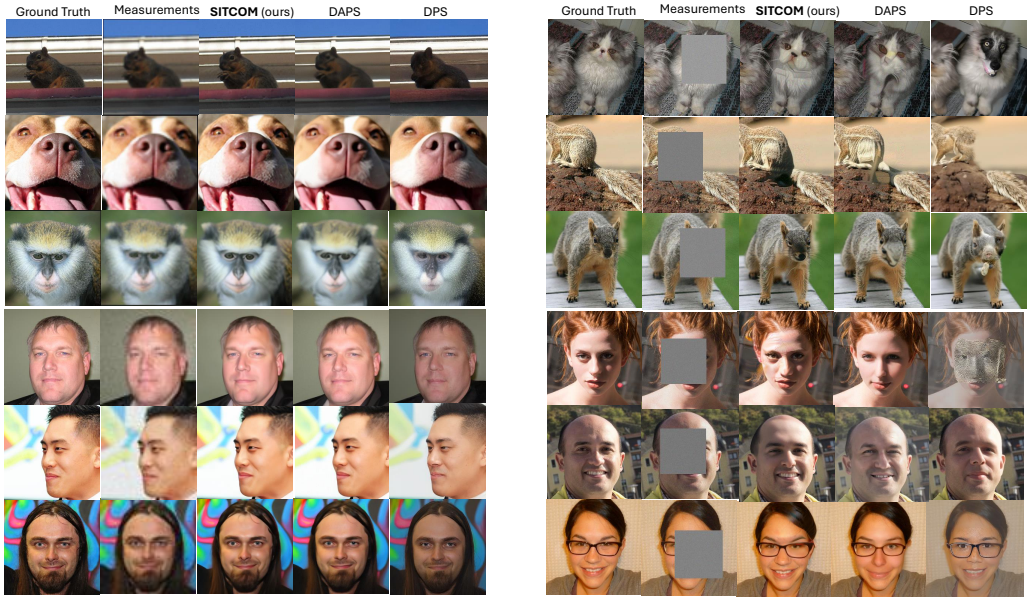


Figure D.6 **Super resolution** (*left*) and **box inpainting** (*right*) results. First (resp. last) three rows are for the FFHQ (resp. ImageNet) dataset.

Figure D.7 **Motion deblurring** (*left*) and **Gaussian deblurring** (*right*) results. First (resp. last) three rows are for the FFHQ (resp. ImageNet) dataset.

Figure D.8 **Random inpainting** (*left*) and **non-linear (non-uniform) deblurring** (*right*) results. First (resp. last) three rows are for the FFHQ (resp. ImageNet) dataset.

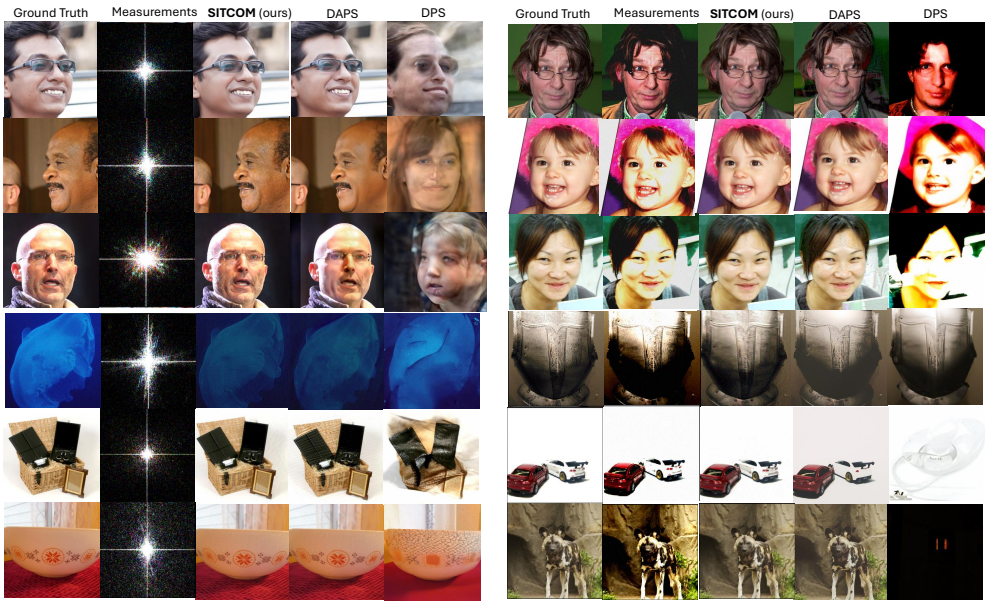


Figure D.9 **Phase retrieval** (*left*) and **high dynamic range** (*right*) results. First (resp. last) three rows are for the FFHQ (resp. ImageNet) dataset.