Optimizing Signal Sampling Strategies for Magnetic Resonance Imaging
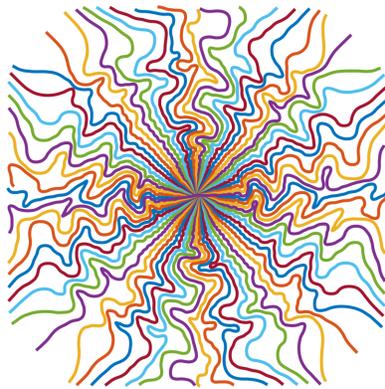
by

Guanhua Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biomedical Engineering)
in the University of Michigan
2023

Doctoral Committee:

        Professor Jeffrey A. Fessler, Co-Chair
        Professor Douglas C. Noll, Co-Chair
        Associate Professor Zhongming Liu
        Assistant Professor Qing Qu
        Associate Professor Kawin Setsompop, Leland Stanford Junior
        University

Guanhua Wang

guanhuaw@umich.edu

ORCID iD: 0000-0002-1622-5664

DEDICATION

To Yellow River.

ACKNOWLEDGMENTS

I am deeply thankful to the many individuals who have broadened my horizons and challenged my thinking. From mentors and colleagues to friends and family members, each person has contributed in unique ways to my intellectual growth. Their guidance and insights have shaped my worldview and provided a strong foundation as I navigate a world of constantly evolving narratives.

Words cannot express my gratitude to my parents, Bin Wang and Furong Hua, and other family members. Their unlimited love has been the eternal harbor during my journey. Years of the COVID-19 pandemic have kept us physically apart, but our hearts remain united. I could not have undertaken this journey without my girlfriend, Yuran Zhu. Her love and understanding always cheered me up in those low moments. We have traveled together far and to so many beautiful places in the past two years. Many of the ideas in this dissertation have emerged from our chats. I would be remiss in not mentioning Baobao the cat, for his constant disturbance.

When writing my thesis, I was reading the thesis of Renyu Huang, a Chinese alumnus of Michigan. Dr. Huang's unique analysis of the Ming dynasty significantly impacted the research community and popular culture in 80-90s. I dreamed of being a historian when I was a child, and I have not given up. Finding a new explanation framework that is less limited by epistemology and ideology is my greatest passion. Thankfully, my training and observations over the past four years have developed several related ideas. Now we are standing in front of another unfortunate diversion. I firmly believe that communication between cultures will help us escape repeated traps and figure out a new way forward.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI artificial intelligence

BOLD blood oxygenation level dependent

CG conjugate gradient

CNN convolutional neural network

CS compressed sensing

CT computed tomography

DFT discrete Fourier transform

EPI echo-planar imaging

FE frequency encoding

FFT fast Fourier transform

FLAIR fluid-attenuated inversion recovery

fMRI functional magnetic resonance imaging

FOV field-of-view

GRE gradient echo

GPU graphics processing unit

i.i.d. independent and identically distributed

IRB institutional review board

NMR nuclear magnetic resonance

NUFFT non-uniform fast Fourier transform

PE phase encoding

ppm parts per million

PSF point spread function

PSNR peak signal-to-noise ratio

RF radiofrequency

SNR signal-to-noise ratio

SAR specific absorption rate

SSIM structural similarity index measure

SWI susceptibility weighted imaging

ABSTRACT

Magnetic resonance imaging (MRI) is an important imaging modality in modern medicine that provides various biomarkers without harmful radiation. However, the low acquisition speed of MRI limits its spatiotemporal resolution, affordability, and patient experience. Efficient data acquisition schemes can enhance the quality and speed of MRI, thereby improving its scientific and clinical values.

Designing efficient and optimal sampling schemes is an important yet challenging task due to the scale and complexity of the problem. Current research broadly adopts oversimplified models and heuristics-driven methods, which have effectively sped up MRI, but hinder the design of even more efficient sampling schemes. Therefore, there is a need to explore new methods to automatically design or tailor sampling trajectories.

This dissertation presents new methods for optimizing MRI sampling trajectories using optimization-based and data-driven methods. We proposed gradient methods for optimizing non-Cartesian sampling trajectories, as well as Bayesian methods for optimizing Cartesian sampling trajectories. The proposed optimization methods can simultaneously improve image quality, hardware conformity, and patient comfort. Notably, these methods can automatically learn efficient sampling patterns from raw datasets, and the results are tailored to specific anatomical structures, scanning protocols, and hardware. To solve this large-scale, non-convex optimization problem, we introduced several computational methods, such as accurate Jacobian approximations for system matrices, parallel computing, and high-efficiency solvers.

We evaluated the proposed methods in multiple MRI applications, such as structural imaging, functional imaging, and dynamic imaging. In both simulation and prospective in-vivo studies, our methods improved image quality by 2-5 dB (in PSNR) and increased acquisition speed by 8-10$\times$. The optimization results also enhanced the patient experience by minimizing uncomfortable peripheral nerve stimulation. Notably, our learning-based approaches exhibited strong generalization ability and robustness to the shift between in-silico training and real-world prospective experiments, which can be explained by simple signal and system theories. These promising preliminary results highlight the potential of our methods to practically accelerate MRI and contribute to advancements in medical imaging research. To promote reproducible research, the accompanying codes, as well as an open-source reconstruction toolbox MIRTorch, are publicly available.

CHAPTER 1

Introduction

Magnetic resonance imaging (MRI) provides powerful tools to noninvasively visualize anatomy and physiology without ionizing radiation. In the modern continuum of care, MRI plays a pivotal role from early screening to diagnosis and monitoring. In addition to clinical values, MRI methods have also been driving scientific innovations across multiple areas, such as neuroscience and oncology. However, access to MRI is limited to only one-third of the world's population, and is heavily restricted by elevated cost, long scanning time, and patient discomfort during scans (such as nerve stimulation and acoustic noise) [47]. For scientific applications, the impossible triangle of image acquisition, the trade-offs between speed, quality, and precision, restrains the potential for further clinical and physiological discoveries. Therefore, accelerated MRI is of both high clinical and scientific values.

MRI is a primarily sequential imaging modality, where raw measurements are sampled one at a time rather than simultaneously (such as in digital cameras.) For example, a high-resolution brain scan may involve 10^7-10^8 samples, with each sample taking 10^{-6}-10^{-5}s, resulting in a total scan on the order of magnitude of 10^2-10^3s. Efficient sampling strategies increase acquisition speed by reducing the number of samples, while designing such efficient sampling is a challenging task due to the large scale. For instance, the degree of freedom of optimization is also 10^7-10^8 in the previous case. Many current methods rely on heuristics and strong simplifications in designing sampling strategies, and researchers may need to use trial-and-error approaches and conduct many experiments to tune relevant parameters. This approach limits both the efficiency of research and the improvement of MRI speed. Therefore, there is a growing demand for automatic and computer-based design.

This dissertation research addresses these challenges by combining optimization and machine learning to improve medical imaging acquisition and reconstruction. We build a 'digital twin' or simulation model of MR imaging physics, with sampling trajectories as tunable parameters. Then, we adopt common optimization concepts, such as gradient descent and greedy algorithm to optimize sampling schemes. The learning-based optimization can *learn* to optimize scan protocols, and the result is specific to anatomy and hardware, avoiding the limitations of fixed and

non-adaptive signal sampling strategies. The methods are carefully designed to generalize well between simulation studies and real-world systems. In many simulation studies and prospective in-vivo experiments, these methods significantly improved image quality, acquisition efficiency, and patient comfort, while remaining robust and interpretable.

The following chapters detail our proposed optimization methods:

- Ch. 3 discusses the theoretical and computational fundamentals for optimizing non-Cartesian sampling trajectories. We analytically derive the Jacobian operators involving non-uniform fast Fourier transform, which are more accurate and efficient than standard auto-differentiation methods by a factor of 10^2-10^3. This proposed method enables the application of gradient-based methods for optimizing sampling trajectories. The chapter demonstrates that gradient methods can learn anatomy-specific and reconstruction-specific information. The related results are published in [144].

- Ch. 4 delineates the joint optimization of 2D non-Cartesian sampling trajectories and learning-based reconstruction methods. We propose several optimization techniques to improve the results of this non-convex optimization problem, which successfully help avoid suboptimal local minima encountered in previous works [154]. The learned protocols were prospectively evaluated on MR scanners, demonstrating improved image quality and reduced artifacts than standard trajectories and heuristics-based trajectory optimization methods [85]. The learned sampling trajectory and reconstruction methods also generalized well between in-silico training and in-vivo experiments and did not require fine-tuning. This chapter is based on a journal publication [146].

- Ch. 5 extends the optimization to 3D sampling trajectories, which pose computational challenges due to their larger scale and potential side effects, such as peripheral nerve stimulation. This chapter proposes and evaluates several strategies for efficient computation, successfully reducing memory cost by 98%. The optimization can optimize parameters in existing protocols, making it more practical and generalized. We applied the optimized trajectory to multiple applications, including structural and functional imaging. In the prospective studies, by optimizing the rotation angles of a stack-of-stars (SOS) trajectory, SNOPY reduced the NRMSE of reconstructed images from 1.19 to 0.97 compared to the best empirical method (RSOS-GR). Optimizing the gradient waveform of a rotational EPI trajectory improved participants' rating of the PNS from 'strong' to 'mild.' A journal paper [147] presents related results.

- Ch. 6 focuses on Cartesian sampling, which has broader acceptance in current clinical settings. The chapter proposes patient-specific adaptive sampling patterns based on Bayesian

experimental design that can predict the next sampling locations on-the-fly according to existing observations. Compared to non-adaptive sampling, in several simulation experiment settings, the proposed method effectively improved the image quality by 1-3 dB in PSNR, and better restored subtle details. Preliminary results are presented in a preprint [148].

- Ch. 7 briefly presents several representative side projects, such as motion correction and longitudinal imaging, which are important practical topics in MRI. The related preliminary results are published in [145, 149].

- Ch. 8 discusses future research topics for sampling trajectory optimization.

CHAPTER 2

Background

MRI is a relatively complex sensing system, partially because it detects weak and subtle in-vivo signals, in the sense that only several out of a million nuclei are in the 'active' mode. Additionally, MRI signals typically have wavelengths in the radiofrequency (RF) range, making them susceptible to interference from a variety of sources. MRI can be broadly categorized into three steps: signal generation, data sampling, and image reconstruction (while new schemes such as [98] are being investigated.) Signal generation involves exciting certain atoms in the human body using carefully designed RF temporal waveforms, also known as pulse sequences. Data sampling typically encodes spatial information into frequency domains via gradient fields, and image reconstruction generates final images from raw signals. These steps are not strictly divided, and each one includes physical processes from other steps. The majority of MR research focuses on optimizing individual steps, whereas this dissertation research proposes a systemic optimization approach that leverages information from all three steps to achieve better results.

2.1 Excitation

MR physics typically uses the classical 'gyroscope' model for easier conceptual understanding. Nuclei have quantized spin values that can be half-integer or integer numbers, usually denoted as I. Only nuclei with non-zero spin can absorb and emit electromagnetic waves, and are hence detectable. In MRI, the most widely used nucleus is ^{1}H, while isotopes such as ^{13}C are useful in specific applications. Under external magnetic fields (B_0 in NMR terminology), charged nuclei precess around the field direction at the Lamour frequency $\omega_0 = \gamma B_0$, where γ is the gyromagnetic ratio. Due to the thermal flip, the alignment of spins follows a Boltzmann distribution between the lower energy state (parallel to the field) and higher energy state (anti-parallel to the field). A net proportion of spins runs parallel to the field, generating the average magnetic dipole density known as longitudinal equilibrium magnetization (M_0). To detect such magnetization, it needs to be tipped from the main field direction, or the low-energy state, to the transverse plane, which generates

detectable signals in receiving channels. The electromagnetic field used for flipping should match the Larmor frequency, which usually falls within the RF range. The corresponding field vector is often denoted as B_1 in NMR terminology. The flipped-down M_0 has a 3D distribution, represented as a vector $\boldsymbol{M} = [M_x\ M_y\ M_z]^T$.

The magnetization vector also experiences relaxation, the restoration to lower energy levels. There are two types of relaxation: spin-spin relaxation, or T_2 relaxation which can be described by $M_{xy}(t) = M_{xy}(t_0)e^{-\frac{t-t_0}{T_2}}$, and spin–lattice relaxation or T_1 relaxation following $M_z(t) = M_0 + (M_z(t_0) - M_0)e^{-\frac{t-t_0}{T_1}}$. More generally, the Bloch equation describes how magnetic fields interacts with nuclear magnetization

$$\frac{d}{dt} \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} -\frac{1}{T_2} & \gamma B_z & -\gamma B_y \\ -\gamma B_z & -\frac{1}{T_2} & \gamma B_x \\ \gamma B_y & -\gamma B_x & -\frac{1}{T_1} \end{pmatrix} \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{M_0}{T_1} \end{pmatrix}. \tag{2.1}$$

The $\boldsymbol{B} = [B_x\ B_y\ B_z]$ includes contributions from B_0, B_1, gradient field, and other sources like concomitant fields. By altering the timing of data sampling, images can have different 'contrasts' based on the tissue properties.

Many mechanisms can influence the observed magnetization of a voxel in MRI, such as diffusion, magnetization transfer, flow, and local B_0 variability. Together with T_1 and T_2 relaxations, these mechanisms enable various contrasts in MRI, such as diffusion-weighted imaging (DWI), susceptibility-weighted imaging (SWI), blood oxygenation level-dependent (BOLD) imaging, and chemical exchange saturation transfer (CEST) imaging. Additionally, objective quantities such as T_1, T_2 or even temperature can be measured. A key research paradigm in MRI for medicine involves coupling various contrasts with pathologies and indications.

2.2 Sampling

The second step performs signal sampling. Directly recording the flux brought by M_{xy} produces a temporal waveform, similar to NMR. MRI differentiates from NMR in encoding spatial information, which can be achieved through several strategies. One such strategy is the excitation of a thin slice in 2D/multi-slice imaging. Another strategy involves assigning different local fields to locations, typically via a linear magnetic field with a gradient $\boldsymbol{G} = [G_x\ G_y\ G_z]$. The gradient may vary across time, denoted as $\boldsymbol{G}(t)$. As a result, each spatial location at a distance of $\boldsymbol{r} = [r_x\ r_y\ r_z]$ to isocenter experiences an additional field of $\boldsymbol{G} \cdot \boldsymbol{r}$, leading to different rotation frequencies of the magnetization at different locations. By sampling the temporal waveform at a rate higher than the Nyquist rate, the spatial information can be encoded into the measured waveforms as differ-

ent frequency components. Mathematically, this encoding process resembles the structure of a Fourier transform when completely ignoring other physics processes such as relaxation and field inhomogeneity

$$s(t) = \int_{\mathcal{D}} \rho(\boldsymbol{x}) e^{-\imath\gamma \int_{\tau=0}^{t} \boldsymbol{G}(\tau)\tau\cdot\boldsymbol{x}\mathrm{d}\tau} \mathrm{d}\boldsymbol{x}, \tag{2.2}$$

where t denotes the sampling time, $\mathcal{D}$ denotes the FOV, and $\rho(x)$ denotes the magnetization density that is proportional to M_{xy}. To encoding 2D/3D spatial dimensional, $\boldsymbol{G}(t)$ is usually carefully designed using strategies such as frequency encoding (FE) and phase encoding (PE). To conceptualize the temporal trajectory of $\boldsymbol{G}(t)$, k-space location vectors are defined as $\boldsymbol{k}(t) = \int_{\tau=0}^{t} \frac{\gamma}{2\pi} \boldsymbol{G}(\tau)\tau\mathrm{d}\tau$. Obviously, signals in spatial domain $\rho(\boldsymbol{x})$ and signals in the frequency domain (k-space) $s(t)$ construct a Fourier pair. With the concept of k-space, $\boldsymbol{k}(t)$ can take various geometrical forms. The common designs include Cartesian rasters, and spiral/radial curves.

2.3   Reconstruction

The reconstruction of MRI can be viewed as an inverse problem of the forward process

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon}. \tag{2.3}$$

$\rho(\boldsymbol{x})$ is a continuous object intrinsically. However, due to limited sampling bandwidth, $\boldsymbol{x}$ is typically assumed to be band-limited, and parameterized by basis functions such as the Dirac pulse and spline functions. $\boldsymbol{y}$ is the sampled k-space data. $\boldsymbol{A}$ describes the excitation, relaxation, and sampling processes discussed above. If we consider the simplest case where the system is perfect without motion, $\boldsymbol{A}$ can be represented as a discrete Fourier transform (DFT). For non-Cartesian sampling, $\boldsymbol{A}$ is a more general non-equidistant Fourier transform with elements

$$a_{ij} = \mathrm{e}^{-\imath\vec{\omega}_i\cdot\vec{r}_j}, \tag{2.4}$$

where $\vec{\omega}_i$ denotes a k-space sampling location and $\vec{r}_j$ denotes the center of a voxel. Non-uniform fast Fourier transform (NUFFT) is often used to accelerate the non-equidistant Fourier transform. For parallel receiving coils, $\boldsymbol{A}$ should also consider sensitivity maps. Suppose there are N_c receiver coils

$$\boldsymbol{A} = (\boldsymbol{I}_{N_c} \otimes \boldsymbol{F}) \begin{bmatrix} \boldsymbol{S}_1 \\ \vdots \\ \boldsymbol{S}_{N_c} \end{bmatrix}. \tag{2.5}$$

$\boldsymbol{A}$ may further involve factors such as B_0 inhomogeneity and motion, which will be discussed in following sections. Generally, if $\boldsymbol{A}$ is sampled equidistantly and satisfies the Nyquist rate, the re-

construction simply involves inverse FFT (iFFT) (possibly with correction procedures such as for B_0 inhomogeneity [104].) For non-Cartesian sampling patterns, gridding operations can interpolate non-Cartesian samples to the Cartesian grid before applying iFFT. To accelerate MRI, $\boldsymbol{A}$ can be under-sampled, which poses a (potentially) under-determined inverse problem. Various reconstruction algorithms can solve the undersampled k-space data. Interpolation of the k-space data is a common approach that utilizes characteristics such as conjugate symmetry or linear predictability [99]. Representative works include GRAPPA [50], homodyne [105], LORAKS [57], and ESPIRiT [141]. The interpolation can either be 'learned' from the signal itself using the coil dependence, or from a historical databank.

Many other methods regard reconstruction as a parameter estimation problem. In particular, the maximum a posteriori (MAP) approach and constrained reconstruction have been extensively studied. These methods have a general form of finding an estimate of $\boldsymbol{x}$, denoted as $\hat{\boldsymbol{x}}$, that minimizes the following objective function:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathsf{R}(\boldsymbol{x}). \tag{2.6}$$

The first term $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is usually referred to as data fidelity term or data consistency term, which can be interpreted as the maximum likelihood estimation considering a Gaussian noise distribution. The term $\mathsf{R}(\boldsymbol{x})$ denotes image-domain regularizers. For example, in CG-SENSE, $\mathsf{R}(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_2^2$, which is a Tikhonov regularizer. In compressed sensing, $\mathsf{R}(\boldsymbol{x}) = \lambda\|\boldsymbol{T}\boldsymbol{x}\|_1$ is a regularizer that promote sparsity, where $\boldsymbol{T}$ is a total variation transform or wavelet transform. Recent years have also seen studies on learned regularizers, where $\mathsf{R}(\boldsymbol{x})$ is either learned from a training dataset or from a single image itself. Related works also proposed to jointly estimate elements in $\boldsymbol{A}$, such as sensitivity maps and field maps with $\boldsymbol{x}$ [44, 140]. Newer methods, including deep learning and Bayesian methods, are also receiving widespread attention.

2.4   Imperfections and Constraints

Similar to any sensing system, an MRI system also suffers from innate defects and imperfections. Given the widespread use of MRI for medical purposes, artifacts or distortions present in the images may result in serious consequences such as misdiagnosis. As a result, extensive research and development efforts are devoted to minimizing, compensating for, and correcting these imperfections to enhance image fidelity. Furthermore, since the imaging object is typically a human or animal, special attention must be paid to avoid tissue damage and discomfort.

In Ch. 2.1, it is assumed that the B_0 magnetic field is spatially uniform, which poses a challenging and costly engineering problem, especially under high magnetic field strengths. Even a

small deviation from the nominal field strength (e.g., 10 parts per million, ppm) can have a negative impact on both excitation and sampling in MRI. To address this issue, modern MRI scanners use passive and active shimming techniques to improve the B_0 uniformity. Additionally, various approaches, including pulse sequences [136], hardware design [135], sampling schemes [122], reconstruction algorithms [40], and post-processing methods [7, 106, 164], can be used to prospectively or retrospectively correct for B_0 non-uniformity. It is also important to note that the B_0 field can drift over time [36], and the gradient field can introduce concomitant fields [153] that may affect phase-contrast-based applications such as MRI thermometry.

B_1 also suffers from imperfect spatial profiles, due to hardware imperfections and conductivity/permittivity of subjects [129]. Specific hardware [161, 162], pulse designs [170], shimming methods [13, 96] and retrospective correction methods [8] can correct potential B_1 inhomogeneity. For the in-vivo study, RF pulse may introduce thermal energy following SAR $\propto \omega_0^2 B_1^2$. Therefore, for in-vivo applications, the design of pulse sequences should also consider SAR limits [150].

The gradient system is also affected by various factors such as heating and eddy currents, which may impact the accuracy of gradients. Passive and active shielding [139], pre-emphasis [4], measurements and retrospective correction [31, 35, 142] are particularly useful for non-Cartesian and EPI-like trajectories. Image post-processing methods can also correct certain artifact incarnations [174]. The linearity of the gradient varies spatially, deteriorating the image quality around edges and corners of the FOV [34]. The maximum value of $\boldsymbol{G}(t)$ (gradient strength) and its derivative $\boldsymbol{S}(t) = \frac{\partial \boldsymbol{G}(t)}{\partial t}$ (slew rate) is bound by hardware limits, which is an important factor in trajectory design. The gradient also introduces peripheral nerve stimulation (PNS). Suppression of PNS may require hardware design [138, 153], and optimization-based methods that are discussed in Ch. 5. Gradient systems may also introduce mechanical vibrations and cause motion artifacts.

2.5 Efficient Signal Sampling for Accelerated MRI

MRI is a slow imaging modality because its data sampling is primarily sequential. The total scanning time is proportional to the number of measurements taken. The prolonged scanning time increases the cost for payers and reduces MRI's availability. Additionally, patients unavoidably move their bodies if the scan lasts long, which further deteriorates image quality. Some subjects may experience claustrophobia inside the closed bore. For certain dynamic imaging applications, such as functional MRI (fMRI) or cardiac MRI, the long scanning time results in lower spatiotemporal resolution that hampers new scientific discoveries. Hence, accelerated MRI is of both scientific and clinical value, with research spanning fields including hardware design, pulse sequence design, inverse problems, and image processing.

Efficient pulse design improves the duty cycle, which refers to the ratio of data sampling com-

pared to the whole sequence, by effectively utilizing signal relaxation under an acceptable SAR and SNR. Reducing the number of measurements during data sampling can also accelerate the scan. However, simply skipping measurements violates the Nyquist rate and can introduce aliasing artifacts. To address this issue, parallel imaging introduces parallel arrays of coils that have spatially-variant sensitivity, providing additional information for image reconstruction. As mentioned in the previous section, many reconstruction methods have been proposed to reconstruct the multi-channel and undersampled images. Recently, deep learning has also emerged as a promising approach for accelerating MRI scans, and has been demonstrated to be effective in many settings.

Efficient sampling strategies in k-space are important for accelerated data acquisition. However, due to the complexity of the MRI system, the concept of 'efficiency' can be difficult to be translated to an optimization objective. There are several factors to consider, including signal-to-noise ratio, spatial resolution, eddy currents, field inhomogeneity, SAR, PNS, and most importantly, the trade-off between scan time and image quality. EPI and spiral trajectories are commonly used due to their high sampling efficiency and have been the standard for fMRI and diffusion imaging, but they also have limitations, such as susceptibility to image artifacts and blurring. CAIPIRINHA was proposed for parallel imaging to reduce the aliasing. For 3D sampling, trajectories such as rosettes and cones also receive wide attention. However, designing these trajectories is a complex optimization problem, with many degrees of freedom and a large search space. For example, for a 3D radial trajectory with 10000 spokes, the design involves 20000 degrees of freedom.

Moreover, a long-neglected question is how the reconstruction interacts with the data acquisition. For example, Fig. 2.1 shows the point spread functions (PSFs) of three sampling trajectories, including phase-encoding undersampling, Poisson-disk sampling, and CAIPI [19]. The PSF of the 1D sampling pattern causes aliasing in the undersampling direction, while the PSF of the Poission-disk is similar to a 2D blurring kernel. Therefore, they will benefit from different regularization terms of image reconstruction. Recently, data-driven methods have investigated the synergy between sampling patterns and reconstruction [1, 9, 154], which is also the subject of Ch. 3 and Ch. 4. Additionally, tailoring sampling trajectories to different anatomies and hardware is also of great value in accelerating acquisition and improve image contrast, and has been investigated in my dissertation research.

In general, MRI has two categories of sampling patterns: Cartesian and Non-Cartesian. Cartesian sampling patterns are based on rastering in k-space, while non-Cartesian patterns are continuous. Correspondingly, the Optimization of Cartesian sampling trajectories poses a discrete subset selection problem which is discussed in Ch. 6. On the other hand, non-Cartesian sampling patterns are continuous, and in Ch. 3, Ch. 4 and Ch. 5, we propose a series of gradient methods to optimize them.
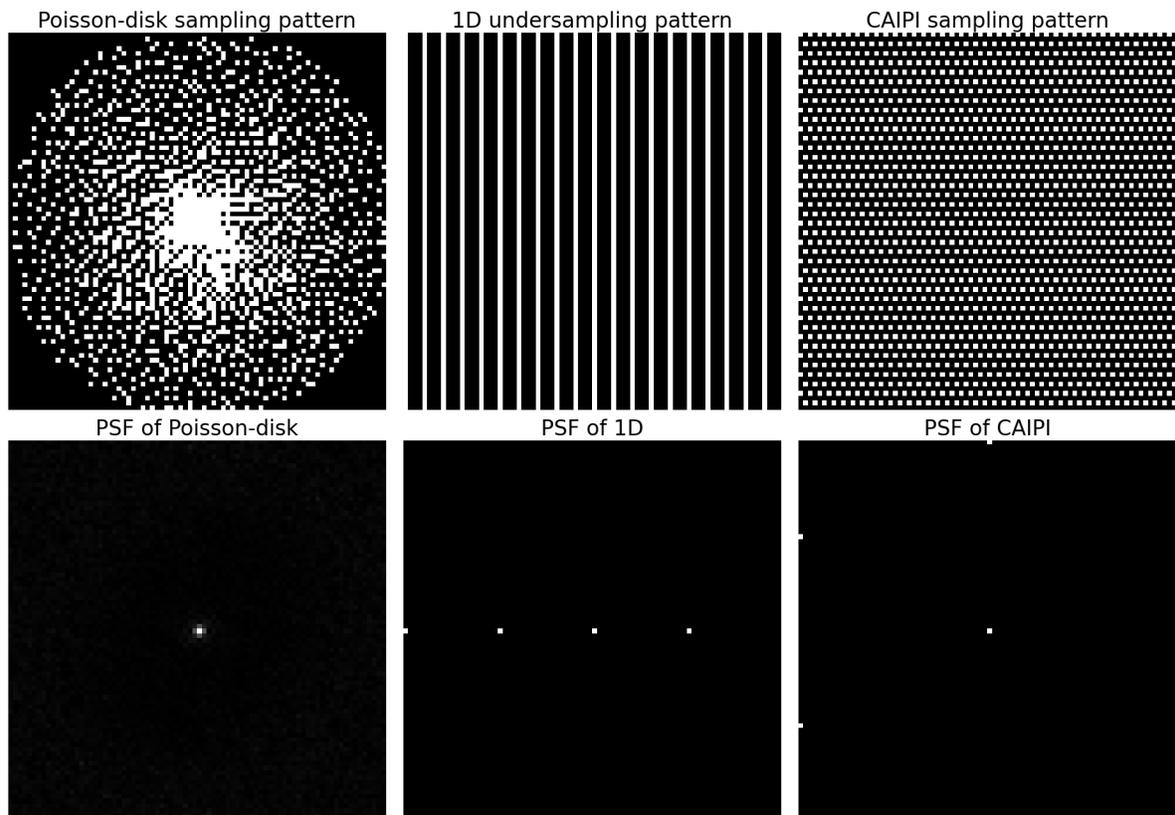
Figure 2.1: Examples of k-space sampling patterns (first row), as well as corresponding image domain PSFs (second row.)

Gradient-based Methods for Optimizing Non-Cartesian Sampling Trajectories

3.1 Introduction

There are several computational imaging modalities where the raw measurements can be modeled as samples of the imaged object's spectrum, where those samples need not lie on the Cartesian grid, including radar, diffraction ultrasound tomography, parallel-beam tomography, and MRI. Image reconstruction methods for such modalities may use non-uniform fast Fourier transform (NUFFT) operations to accelerate computation [41]. The quality of the reconstructed image depends both on the image reconstruction method and on the characteristics of the frequency domain sampling pattern.

MRI has particular flexibility in designing frequency domain sampling patterns. Many MR sampling patterns are discrete subsets of the Cartesian grid, and the corresponding optimization/learning strategies include greedy algorithms [56, 119, 175], reparameterization [9, 20, 70, 80, 127], Bayesian optimization [58, 125], and system matrix analysis [46, 89, 143]. The other type is non-Cartesian sampling, which uses a collection of continuous functions in k-space. Several studies applied gradient methods to optimize non-Cartesian sampling trajectories [1, 124, 146, 154], and it is also possible to use derivative-free optimization algorithms in certain applications. This paper develops efficient tools for applying gradient methods to non-Cartesian sampling pattern optimization[1].

Some data-driven optimization methods for non-Cartesian sampling solve an optimization problem involving both forward system models and image reconstruction methods [1, 146, 154]. The forward models and reconstruction methods both depend on NUFFT operations. In principle, the Fourier transform operation is a continuous function of the k-space sample locations and thus should be applicable to gradient-based optimization methods. In practice, the NUFFT is an ap-

---

[1]This chapter is based on our journal paper [144].

proximation to the non-uniform discrete Fourier transform (NUDFT) and that approximation often is implemented using non-differentiable lookup table operations or other interpolation techniques [11, 28]. Such approximations are sufficient for image reconstruction (forward mode), but have problematic efficiency and accuracy if one attempts to use standard auto-differentiation tools for gradient-based optimization. Standard auto-differentiation methods using subgradients can lead to incorrect NUFFT Jacobians. They also require prohibitively large amounts of memory for back-propagation through certain algorithm stages such as conjugate gradient (CG) steps that involve NUFFT operations.

This chapter proposes an efficient approach that replaces memory-intensive and inaccurate auto-differentiation steps with fast Jacobian approximations that are themselves based on NUFFT operations [144]. The proposed approach requires substantially less memory for iterative updates like CG steps.

We applied the proposed Jacobian to learn MRI sampling trajectories via stochastic gradient descent. By applying the forward system model and subsequent reconstruction, reconstructed images were simulated from reference images in the training set. The similarity between simulated and reference images was the metric for updating the sampling trajectory. We used model-based reconstruction methods, such as regularized least-squares and compressed sensing. In comparison with previous works using reconstruction neural networks (NN) [154], such model-based reconstruction methods can be more robust and require less training data.

In addition to simple NUFFT-based sensing matrices, we also considered several scenarios in MR sampling and reconstruction, including the multi-coil (sensitivity-encoded) imaging [110] system models that account for B_0 field inhomogeneity [134]. The derivation also includes fast Jacobian approximations for Gram and 'data consistency' operations commonly used in iterative reconstruction methods.

Jacobians with respect to the non-Cartesian sampling pattern are also relevant for tomographic image reconstruction problems with unknown view angles (like cryo-EM) where the view angles must be estimated [168].

3.2 Theory

3.2.1 Efficient Jacobian Approximation

We denote matrices, vectors and scalars by $\boldsymbol{A}$, $\boldsymbol{a}$ and a, respectively. $\boldsymbol{A}'$, $\boldsymbol{A}^T$ and $\boldsymbol{A}^*$ denote the Hermitian transpose, the transpose and the complex conjugate of $\boldsymbol{A}$, respectively.

Consider a scalar function $f(z)$, $z = x + y\imath \in \mathbb{C}$, $x, y \in \mathbb{R}$. Following the conventions in

Wirtinger calculus [114, p. 67], the differential operators are defined as

$$\frac{\partial}{\partial z} = \frac{1}{2}\frac{\partial}{\partial x} - \frac{\imath}{2}\frac{\partial}{\partial y}, \quad \frac{\partial}{\partial z^*} = \frac{1}{2}\frac{\partial}{\partial x} + \frac{\imath}{2}\frac{\partial}{\partial y}.$$

A function f is *complex differentiable* or *holomorphic* iff $\frac{\partial f}{\partial z^*} = 0$ (Cauchy–Riemann equation) [114, p. 66]. In the context of optimization, a cost function L (usually a real scalar) is not holomorphic w.r.t. complex variables. A common approach (as adopted by PyTorch and TensorFlow) regards the real and imaginary components of a complex variable as two real-valued variables, and updates them separately, similar to the real-valued calculus [82]. For example, the nth gradient descent step uses an update

$$\boldsymbol{z}_{n+1} = \boldsymbol{z}_n - \alpha\left(\frac{\partial L}{\partial \boldsymbol{x}} + \imath\frac{\partial L}{\partial \boldsymbol{y}}\right) = \boldsymbol{z}_n - 2\alpha\frac{\partial L}{\partial \boldsymbol{z}^*},$$

where $\alpha \in \mathbb{R}^+$ denotes the step size. The chain rule still applies to calculating $\frac{\partial L}{\partial \boldsymbol{z}^*}$ [65] [114, p. 68]; for $s = f(z)$:

$$\frac{\partial L}{\partial \boldsymbol{z}^*} = \left(\frac{\partial L}{\partial \boldsymbol{s}^*}\right)^* \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{z}^*} + \frac{\partial L}{\partial \boldsymbol{s}^*}\left(\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{z}}\right)^*. \tag{3.1}$$

To derive the Jacobians of system matrices, we initially consider the simplified (single-coil, homogeneous) MRI measurement model for non-Cartesian sampling based on the NUDFT [38]:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{y} \in \mathbb{C}^M$ denotes the measured k-space data, $\boldsymbol{x} \in \mathbb{C}^N$ denotes the unknown image to be reconstructed, and $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ denotes the system matrix or encoding matrix, where $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{\omega})$ has elements

$$a_{ij} = \mathrm{e}^{-\imath\vec{\omega}_i \cdot \vec{r}_j}, \quad i = 1, \ldots, M, \quad j = 1, \ldots, N \tag{3.2}$$

for $\vec{\omega}_i \in \mathbb{R}^D$ and $\vec{r}_j \in \mathbb{R}^D$ where $D \in \{1, 2, 3 \ldots\}$ denotes the image dimension, and where

$$\boldsymbol{\omega} = [\boldsymbol{\omega}^{[1]}\, \boldsymbol{\omega}^{[2]} \ldots \boldsymbol{\omega}^{[D]}]$$

is the $M \times d$ matrix consisting of all the k-space sampling locations and $\boldsymbol{\omega}^{[d]} \in \mathbb{R}^M$ denotes its dth column. (For simplicity here, we ignore other physical effects like field inhomogeneity and relaxation that are sometimes included in the forward model in MRI [38].) The center locations of voxels $\{\vec{r}_j\}$ usually lie on a Cartesian grid, but the k-space sample locations $\boldsymbol{\omega}$ in principle can be arbitrary subject to the Nyquist constraint.

Typically $\boldsymbol{A}$ is approximated by a NUFFT [41]. Usually, the NUFFT operator involves

frequency-domain interpolation operations that are often non-differentiable. One previous trajectory optimization approach that used auto-differentiation [154] replaced the non-differentiable lookup table with a bilinear interpolator. Bilinear interpolation is differentiable everywhere except at the sample locations. Auto-differentiation of bilinear interpolation involves differentiating some `floor` and `ceiling` operations and those derivatives are defined to be zero in popular deep learning frameworks such as PyTorch and TensorFlow, leading to suboptimal sub-gradient calculations. Nearest-neighbor interpolation has even worse properties for auto-differentiation because its derivative is zero almost everywhere, leading to a completely vanishing gradient.

Instead, we investigate a different approach where we analyze the Jacobians w.r.t. $\boldsymbol{\omega}$ and $\boldsymbol{x}$ using the NUDFT expression. Analytically, the Jacobian for the forward system matrix $\boldsymbol{A}$ is:

$$\frac{\partial \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{0}, \quad \frac{\partial \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} = -\imath \operatorname{diag}\big\{\boldsymbol{A}(\boldsymbol{x} \odot \boldsymbol{r}^{[d]})\big\}. \tag{3.3}$$

The Jacobian for the adjoint system matrix $\boldsymbol{A}$' is:

$$\frac{\partial \boldsymbol{A}'\boldsymbol{y}}{\partial \boldsymbol{y}} = \boldsymbol{0}, \quad \frac{\partial \boldsymbol{A}'\boldsymbol{y}}{\partial \boldsymbol{\omega}^{[d]}} = \imath \operatorname{diag}\big\{\boldsymbol{r}^{[d]}\big\} \boldsymbol{A}'\operatorname{diag}\{\boldsymbol{y}\}. \tag{3.4}$$

The Jacobian for the Gram matrix $\boldsymbol{A}'\boldsymbol{A}$ is:

$$\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \boldsymbol{x}}\boldsymbol{x} = \boldsymbol{0},$$

$$\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \boldsymbol{\omega}^{[d]}}\boldsymbol{x} = -\imath \boldsymbol{A}'\operatorname{diag}\big\{\boldsymbol{A}(\boldsymbol{x} \odot \boldsymbol{r}^{[d]})\big\} + \imath \operatorname{diag}\big\{\boldsymbol{r}^{[d]}\big\} \boldsymbol{A}'\operatorname{diag}\{\boldsymbol{A}\boldsymbol{x}\}. \tag{3.5}$$

The Jacobian for the 'inverse' operator $\boldsymbol{F} = \boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T}$ is:

$$\frac{\partial \boldsymbol{F}^{-1}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{0},$$

$$\frac{\partial \boldsymbol{F}^{-1}\boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} = -\boldsymbol{F}^{-1}\Big(-\imath \boldsymbol{A}'\operatorname{diag}\big\{\boldsymbol{A}(\boldsymbol{z} \odot \boldsymbol{r}^{[d]})\big\} + \imath \operatorname{diag}\big\{\boldsymbol{r}^{[d]}\big\} \boldsymbol{A}'\operatorname{diag}\{\boldsymbol{A}\boldsymbol{z}\}\Big). \tag{3.6}$$

The appendix provides the detailed derivation and the use of sensitivity and field inhomogeneity information within $\boldsymbol{A}$ does not affect the formulation.

For efficient implementation, one may replace the NUDFT operations ($\boldsymbol{A}$) within the Jacobians with NUFFT approximations. The next section numerically examines the accuracy of such approximations and the appendix discusses the error bounds. This approach enables faster computation and reduces memory requirements, which is crucial for stochastic optimization of large-scale sampling trajectories.

14

3.2.2 Applying Gradient Methods

For modern MRI systems, the sampling trajectory $\boldsymbol{\omega}$ is a programmable parameter. Traditionally $\boldsymbol{\omega}$ is a geometrical curve controlled by few parameters (such as radial spokes or spiral leaves), and its tuning relies on derivative-free optimizers such as grid-search. In this paper, we optimize $\boldsymbol{\omega}$ by minimizing a training loss function from image reconstruction, where the descent direction of $\boldsymbol{\omega}$ is the negative gradient of that loss [146, 154]. We adopt such a 'reconstruction loss' because the terminal goal of sampling pattern optimization is to improve image quality. To learn from large datasets, the optimization uses stochastic gradient descent (SGD)-like algorithms. Additionally, the loss function may include other terms, such as a penalty on the maximum gradient strength and slew rate [146, 154] or peripheral nerve stimulation effects [147].

For image reconstruction, consider a convex and smooth regularizer $\mathsf{R}(\cdot)$ for simplicity. Since the noise statistics are Gaussian, a typical regularized cost function used for model-based image reconstruction is (2.6):

$$\Psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathsf{R}(\boldsymbol{x}).$$

During training, the observation $\boldsymbol{y}$ can be retrospectively simulated using $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x}^{\text{true}}$. For illustration, consider applying the kth step of gradient descent (GD) to that cost function:

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \alpha\nabla\Psi(\boldsymbol{x}_k) \\
&= \boldsymbol{x}_k - \alpha\,\boldsymbol{A}(\boldsymbol{\omega})'\,(\boldsymbol{A}(\boldsymbol{\omega})\,\boldsymbol{x}_k - \boldsymbol{y}) - \alpha\nabla\mathsf{R}(\boldsymbol{x}_k),
\end{aligned}$$

where $\boldsymbol{x}_k \in \mathbb{C}^N$, $\alpha \in \mathbb{R}^+$ is the step size. After K iterations, we have a reconstructed image (batch) $\boldsymbol{x}_K = \boldsymbol{x}_K(\boldsymbol{\omega}) = f(\boldsymbol{\omega}, \boldsymbol{y})$, where the reconstruction method $f(\boldsymbol{\omega}, \boldsymbol{y})$ is a function of both the data $\boldsymbol{y}$ and the sampling pattern $\boldsymbol{\omega}$. To learn/update the sampling pattern $\boldsymbol{\omega}$, consider a simple loss function for a single training example:

$$L(\boldsymbol{\omega}) = \left\|\boldsymbol{x}_K(\boldsymbol{\omega}) - \boldsymbol{x}^{\text{true}}\right\|_2^2 \tag{3.7}$$

where $\boldsymbol{x}^{\text{true}}$ is the reference fully-sampled image (batch). Learning $\boldsymbol{\omega}$ via backpropagation (or chain-rule) requires differentiating L w.r.t. the sampling pattern $\boldsymbol{\omega}$, which in turn involves Jacobians of quantities like $\boldsymbol{A}(\boldsymbol{\omega})$ that we derived above.

Here we use the forward operator as an example to illustrate one step in propagation. As needed in a backpropagation step (Jacobian-vector product, JVP), the Jacobian (3.3) is multiplied with a gradient vector $\boldsymbol{v} = \frac{\partial L}{\partial(\boldsymbol{A}\boldsymbol{x})^*} \in \mathbb{C}^M$ calculated in the prior step. Using (3.1), the corresponding JVP is

$$\frac{\partial L}{\partial\boldsymbol{\omega}} = \text{real}\big\{(-\imath\,\boldsymbol{A}(\boldsymbol{x} \odot \boldsymbol{r}^{[d]}))' \odot \boldsymbol{v}\big\}. \tag{3.8}$$

15

Efficiently computation can simply apply a NUFFT operation to $\boldsymbol{x} \odot \boldsymbol{r}^{[d]}$, followed by a point-wise multiplication with $\boldsymbol{v}$. The Gram (3.5) and PSD inverse (3.6) ('data consistency') terms follow a similar pattern during backpropagation.

Although we illustrate the GD algorithm with a simple smooth regularizer, more generally, the reconstruction method $f(\boldsymbol{\omega}, \boldsymbol{y})$ can involve more sophisticated regularizers such as neural networks [154, 146] or non-smooth sparsity models [91] used in compressed sensing. In such cases, back-propagation uses sub-gradients, instead of gradients, as is common in stochastic optimization. The loss JVPs are backpropagated through iterative reconstruction steps to compute a gradient w.r.t. $\boldsymbol{\omega}$.

The proposed approach is applicable only to non-Cartesian MRI, because Cartesian sampling pattern design is usually a discrete optimization problem, incompatible with gradient-based methods. However, one could optimize phase-encoding locations continuously (in 2D or 3D) with the frequency-encoding direction being fully sampled, leading to a hybrid Cartesian / non-Cartesian approach [1].

3.3 Validation

This section validates the accuracy and efficiency of the proposed methods. We performed numerical experiments to examine the following test cases:

$$\frac{\partial \left\| f(\boldsymbol{x}) \right\|_2^2}{\partial \boldsymbol{\omega}^{[d]}} \text{ and } \frac{\partial \left\| f(\boldsymbol{x}) \right\|_2^2}{\partial \boldsymbol{x}^*},$$

where $f(\cdot)$ denotes multiplication by $\boldsymbol{A}$, by the Gram matrix $\boldsymbol{A}'\boldsymbol{A}$, or by the 'inverse of PSD matrix' of sensitivity-informed NUFFTs. The Gram and inverse experiments implicitly test the adjoint operator's approximations. The $\boldsymbol{x}$ adopted is a 40×40 patch cropped from the center of a Shepp–Logan phantom with random additional phases uniformly distributed in $[-\pi, \pi]$. $\boldsymbol{S}$ is a simulated 8-channel sensitivity map, and $\boldsymbol{\omega}$ is one radial spoke crossing the k-space center. The Jacobian calculation methods are: (1) auto-differentiation of NUFFT; the lookup table operation [28] is replaced by bilinear interpolation to enable auto-differentiation, similar to [154], (2) our approximation described above, (3) auto-differentiation of exact non-uniform discrete Fourier transform (NUDFT), implemented with single precision. We regard method 3 (NUDFT) as the ground truth. Since NUDFT (in its simplest form) involves only one exponential function, multiplication, and addition for each element, its backpropagation introduces minimal numerical errors. For the PSD inverse, we applied 20 CG iterations for all three methods, which was sufficiently close to convergence based on the residual norm $\|\boldsymbol{r}\| / \|\boldsymbol{b}\|$ (the definition follows [128, (45)]).

Fig. 3.1 and Fig. 3.2 illustrate representative profiles of the gradients w.r.t. $\boldsymbol{x}$ and $\boldsymbol{\omega}$. For $\boldsymbol{\omega}$, the auto-differentiation (method 1) approach has larger deviations from method 3 (NUDFT) because
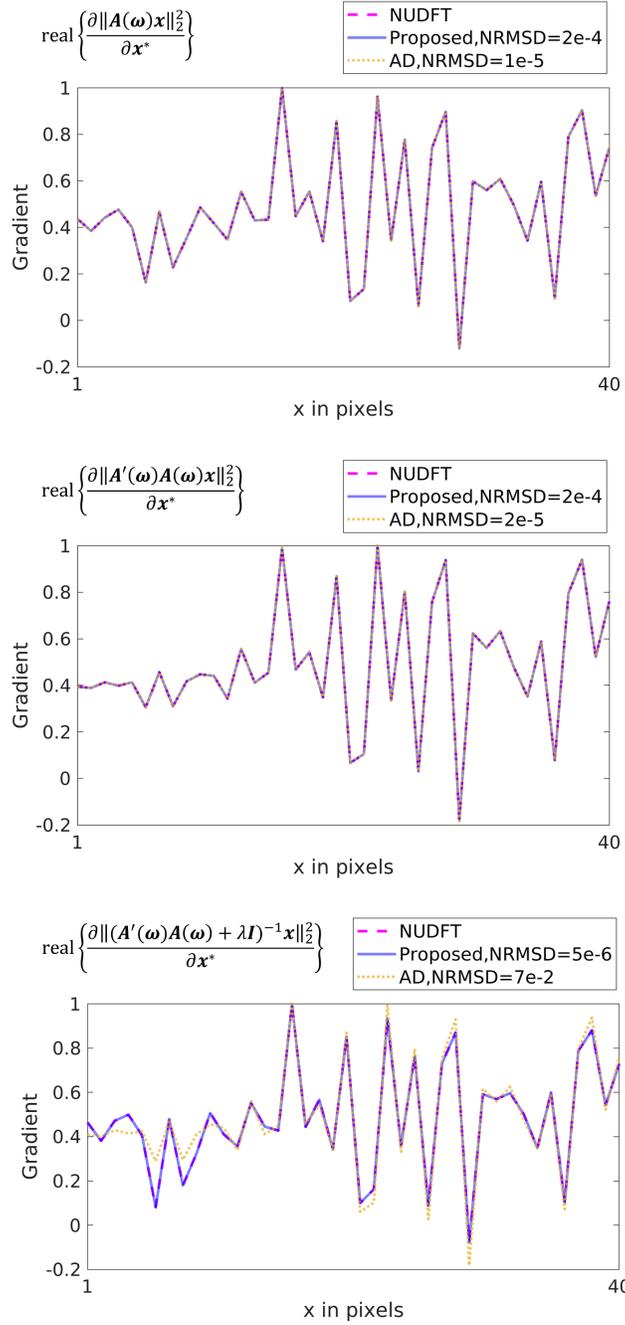
Figure 3.1: Examples of gradients w.r.t. $\boldsymbol{x}^*$ (the real part is plotted) via different calculation methods. Plots show one representative row of a 40×40 matrix (rescaled to [-1,1]). The rows are the forward, Gram, and PSD inverse operator cases. The horizontal axis is the pixel index. The legend reports the normalized root-mean-square difference (NRMSD) compared with the referential NUDFT calculation.

17
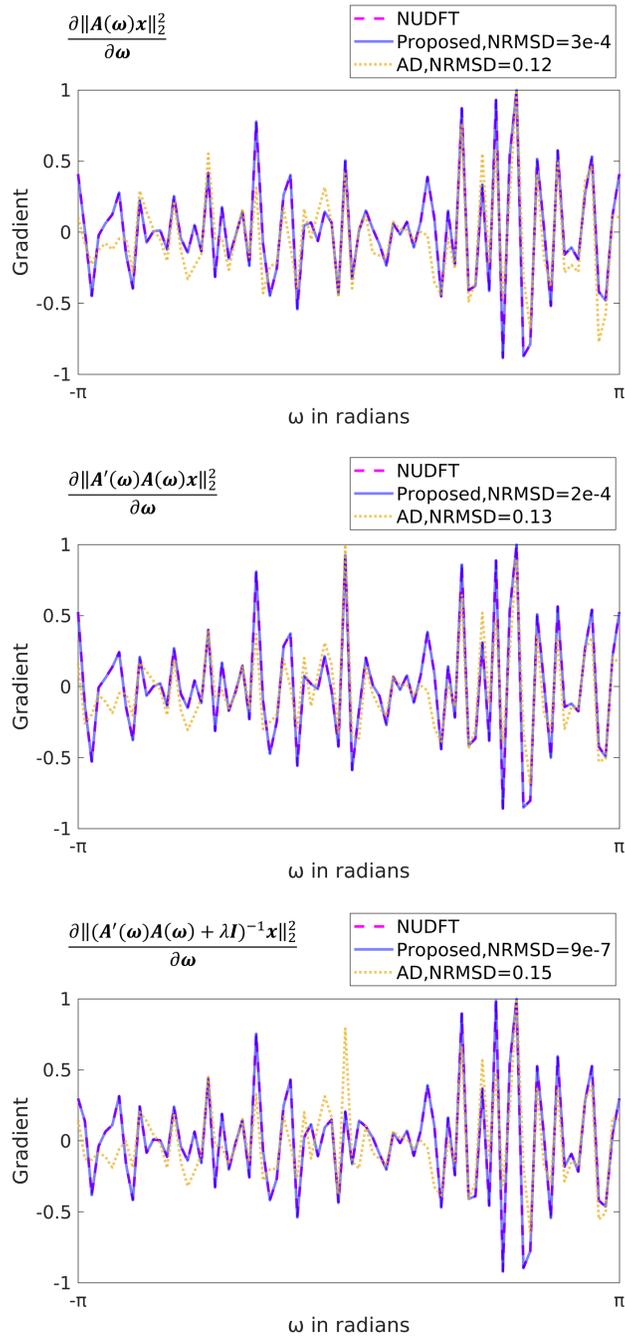
Figure 3.2: Examples of gradients w.r.t. $\boldsymbol{\omega}$ via different calculation methods. Plots show one spoke of 80 points (rescaled to [-1,1]). The rows are the forward, Gram, and PSD inverse operator cases. The proposed approximation better matches the gradient of the NUDFT. The legend reports the normalized root-mean-square difference (NRMSD) compared with the reference NUDFT calculation. The proposed approximation has $400\times$ smaller NRMSD for this nonlinear case.

18

of the non-differentiability of interpolation operations w.r.t. coordinates. For the gradient w.r.t. x, both method 1 and method 2 generate accurate results for forward and Gram operators. The reason is that in method 1 (auto-diff), the interpolation operation w.r.t x is linear, hence accurately differentiable. For the PSD inverse, method 1 led to a slightly inaccurate gradient, stemming from the accumulated errors of backpropagating CG iterations.

Table 3.1 and Table 3.2 compare the time and memory cost of methods 1 (auto-diff) and 2 (proposed). The CPU is Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz and the GPU is an Nvidia(R) RTX2080Ti. We used PyTorch 1.9.1 and torchkbnufft 1.1.0. The memory usage was tracked by `torch.cuda.max_memory_allocated` on Nvidia GPUs. We implemented the numerical experiments with torchkbnufft [101] and MIRTorch toolboxes.

Our method is much faster than auto-differentiation on both GPUs and CPUs, and uses less memory. Importantly, the PSD inverse Jacobian is impractical for the 3D case, whereas the proposed approach fit comfortably in GPU's onboard memory.

Table 3.1: Computation time of test cases via different Jacobian calculations.

	Gram		Inverse	
	auto-diff	proposed	auto-diff	proposed
Large image - GPU	0.3s	**0.2s**	4.3s	**2.5s**
Small image - GPU	0.1s	**0.1s**	1.3s	**0.9s**
Large image - CPU	5.2s	**1.7s**	276.2s	**48.5s**
Small image - CPU	0.8s	**0.5s**	27.4s	**6.9s**

Large size: 400×400; small size: 40×40
20 CG iterations were applied in the PSD inverse cases.

Table 3.2: Memory use of test cases via different Jacobian calculations.

	Gram		Inverse	
	standard	proposed	standard	proposed
Small	3.1MB	**2.8MB**	145.7MB	**2.9MB**
Large	375.9MB	**267.5MB**	5673.2MB	**272.0MB**
3D	N/A	**10.1GB**	N/A	**10.8GB**

Large size: 400×400; small size: 40×40;
3D size: $200 \times 200 \times 100$.
N/A: the memory usage was too large for a single GPU.
20 CG iterations were applied in the PSD inverse cases.

3.4 Trajectory Optimization

This initial experiment optimized the MRI sampling trajectory using the proposed Jacobian approximations and stochastic optimization. The reconstruction methods (2.6) here consider two types of algorithms, namely smooth (regularized) least-squares reconstruction and sparsity-based reconstruction.

The smooth reconstruction method uses the cost function

$$\Psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{E}(\boldsymbol{\omega})\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{T}\boldsymbol{x}\|_2^2,$$

where $\boldsymbol{T}$ is a finite-difference operator encouraging smoothness. Correspondingly, the reconstructed image is:

$$\boldsymbol{x}_K = (\boldsymbol{E}'\boldsymbol{E} + \lambda\boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{E}'\boldsymbol{y},$$

which we solved using CG [128]. The following descriptions refer to this method as quadratic penalized least-squares (QPLS). We also implemented a simpler case, where $\boldsymbol{T} = \boldsymbol{I}$, which is referred as CG-SENSE. In both scenarios, we set λ to 10^{-3} empirically and still applied 20 CG iterations. The initialization of CG used the density compensated reconstruction [67].

The sparsity-based compressed (CS) sensing algorithm adopts a wavelets-based sparsity penalty, and has the following objective function

$$\Psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{E}(\boldsymbol{\omega})\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{W}\boldsymbol{x}\|_1,$$

where $\boldsymbol{W}$ is an orthogonal DWT matrix and we set $\lambda = 10^{-5}$ empirically. We used 40 iterations of the proximal optimized gradient method (POGM) [39, 78] to solve this non-smooth optimization problem.

For the purpose of comparing trajectories and image quality, we also applied the proposed approximations to an unrolled neural network (UNN) reconstruction method that followed the definition of [2] (Ch. 4 extends it to non-Cartesian cases.) We used the same network configuration as in Ch. 4.

To optimize the k-space trajectory for each of these reconstruction methods, the training loss (3.7) is:

$$L(\boldsymbol{\omega}) = \left\|\boldsymbol{x}_K(\boldsymbol{\omega}) - \boldsymbol{x}^{\text{true}}\right\|_2^2 + \mu_1\phi_{\gamma\Delta t g_{\max}}(|\boldsymbol{D}_1\boldsymbol{\omega}|) + \mu_2\phi_{\gamma\Delta t^2 s_{\max}}(|\boldsymbol{D}_2\boldsymbol{\omega}|),$$

where $\boldsymbol{x}^{\text{true}}$ is the conjugate phase reconstruction of fully sampled Cartesian data [104]. The second and third terms applied a soft constraint on gradient strength and slew rate similar to Ch. 4 and Ch. 5, where $\phi_\lambda(|\boldsymbol{x}|) = \sum \max(|\boldsymbol{x}| - \lambda, 0)$. The maximum gradient strength ($g_{\max}$) was 5

Gauss/cm and the maximum slew rate ($s_{\max}$) was 15 Gauss/cm/ms. $\mu_1 = \mu_2 = 10$. We estimated sensitivity maps in $\boldsymbol{E}$ using ESPIRiT [141], and simulated noiseless raw signals $\boldsymbol{y} = \boldsymbol{E}(\boldsymbol{\omega})\boldsymbol{x}^{\text{true}}$ retrospectively w.r.t. candidate trajectories. The training used the fastMRI brain dataset [167] containing 15902 T_1w slices, 16020 T_2w slices, and 3311 FLAIR slices cropped to size 320×320. The number of coils ranges from 2 to 28. We used the Adam optimizer [79], with step size 10^{-4} and mini-batch size 12. We used 6 epochs for training model-based methods (CG-SENSE, QPLS and CS) and 60 epochs for the UNN training. The initialization of learned trajectories was an under-sampled radial trajectory in all experiments. The initialization had 16 'spokes' and each spoke was 5ms long with 1280 sampling points. We also adopted the k-space parameterization trick detailed in (4.2) to avoid sub-optimal local minima, and parameterized each shot with 40 quadratic spline kernels.
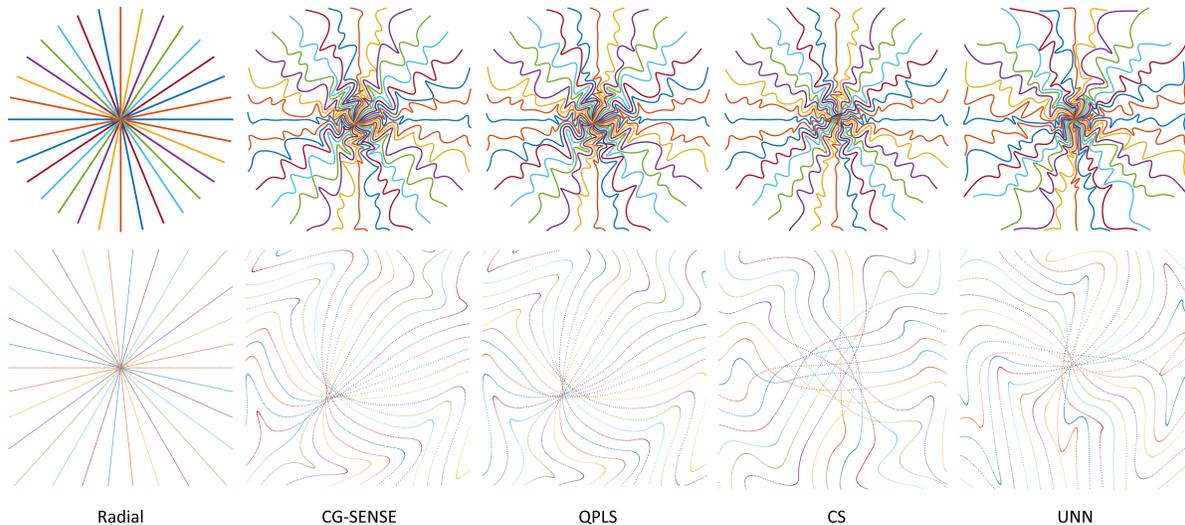
Figure 3.3: Optimized sampling trajectories for several iterative reconstruction methods. The left column shows the uniform radial initialization. The second row shows the $8\times$ zoomed-in central k-space.

Fig. 3.3 showcases the trajectories optimized for each of the reconstruction methods. The centers of trajectories optimized with quadratic regularizers (CG-SENSE and QPLS) are not aligned with the k-space origin. We hypothesize that regularizers (and corresponding iterative algorithms) handle image phases differently, resulting in distinct trajectory centers.

Table 3.3 reports the average image reconstruction quality (PSNR and SSIM [68], fully sampled image as the ground truth) on 500 test slices. It also showcases the image quality of these learned trajectories with reconstruction methods different from the training phase. All learned trajectories led to improved reconstruction quality compared to the initial radial trajectory (unopt.), even with different reconstruction methods. Importantly, the same reconstruction algorithm across training
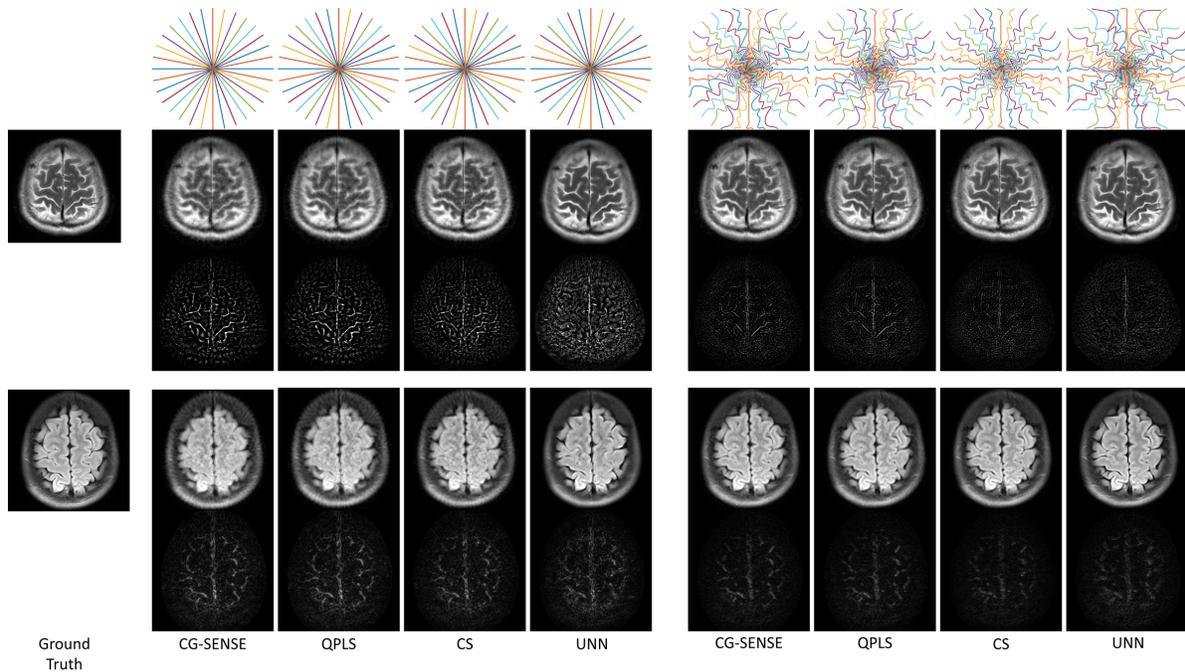
Figure 3.4: Examples of the reconstructed images with unoptimized (left) and optimized trajectories (right). Rows 3 and 5 show corresponding error maps.

and test led to the greatest improvement (the bold diagonal entries). Fig. 3.4 shows reconstruction examples.
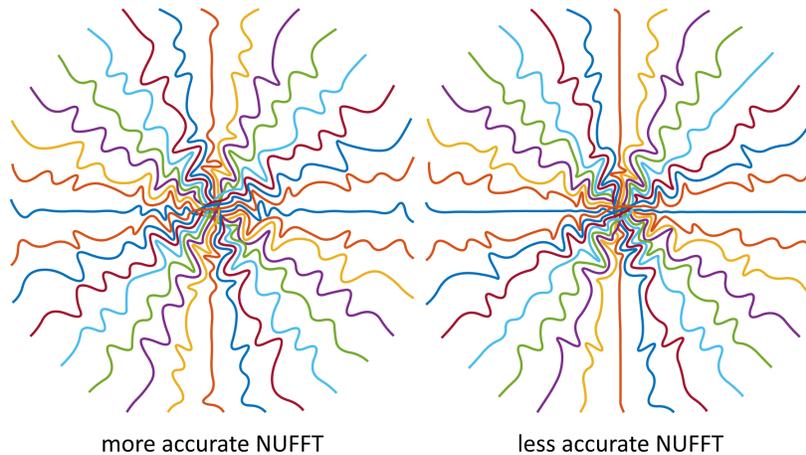
Figure 3.5: Learned trajectories with different NUFFT accuracies.

The major computation cost of trajectory learning is proportional to NUFFTs and their Jacobian calculations. An empirical acceleration method is to use faster NUFFT approximations (low over-sampling factors and/or small interpolation neighborhoods) in training. Later, when the learned

22

Table 3.3: Average reconstruction quality on test set with trajectories learned for different reconstruction methods.

SSIM

Test \ Learn	QPLS	SENSE	CS	UNN	unopt.
QPLS	**0.963**	0.963	0.962	0.961	0.947
SENSE	0.964	**0.964**	0.963	0.961	0.946
CS	0.962	0.963	**0.966**	0.964	0.946
UNN	0.960	0.960	0.958	**0.964**	0.950

PSNR (in dB)

Test \ Learn	QPLS	SENSE	CS	UNN	unopt.
QPLS	**35.1**	35.1	34.9	35.0	33.1
SENSE	35.2	**35.2**	34.9	35.1	33.1
CS	34.8	34.9	**35.4**	35.2	33.0
UNN	34.6	34.6	34.5	**35.0**	33.5

Learn: reconstruction method in joint optimization.
Test: reconstruction method being tested.
unopt.: the unoptimized initialization.

trajectory is deployed on test data or prospectively acquired data, one could use default NUFFT accuracy. We investigated learning trajectories with two different NUFFT accuracies: (1) gridding size = $1.25\times$ image size, interpolation kernel size = 5 and (2) gridding size = $2\times$ image size, interpolation kernel size = 6 which is a commonly used setting. On our GPUs, the lower-accuracy setting was $1.4\times$ faster than the higher-accuracy one. We used the CS-based reconstruction and corresponding training strategy described in the previous subsection. Fig. 3.5 shows the trajectory optimized for the two NUFFT accuracy levels. To compare the trajectory optimized by these two settings, we used the reconstruction image quality as the evaluation metric. We simulated and reconstructed images using the two trajectories on the test data (same as the previous experiment). The trajectories optimized with the 'low accuracy' and 'high accuracy' NUFFT had mean PSNR values of 35.4±4.6 dB and 35.4±4.7 dB.

3.5 Discussion

This chapter presents a model-based approximation of Jacobian matrices involving NUFFTs. Compared to direct auto-differentiation, the proposed method is faster, needs less memory, and better approximates the reference NUDFT results. As discussed, the error of auto-differentiation is

23

not a software limitation, but rather a problem that stems from the non-differentiability of interpolation or lookup table operations. NUFFT alternatives such as re-gridding or filtered back-projection also suffer from similar non-differentiability issues and are less effective than (NUFFT-based) iterative reconstruction.

Sampling patterns learned with different reconstruction methods showed distinct characteristics. This phenomenon was also observed in previous literature [56, 175]. The differences in sampling patterns may stem from different regularizers, as well as different iterative algorithms. Importantly, as shown in Table 3.3, synergistic sampling and reconstruction led to the best image quality. Several previous studies [1, 154] only used NN-based reconstruction methods, while the stability and generalizability of NN-based reconstruction are still being investigated. In comparison, using our method, one may optimize trajectories for model-based reconstruction methods that may be more robust. Our results show that with a suitably tailored sampling pattern, traditional model-based reconstruction can compete with NN-based reconstruction, reinforcing related observations in recent studies [51]. Future studies should conduct more comprehensive validation on this point. Additionally, sampling optimization for model-based reconstruction requires less training data than for NN-based reconstruction. This property is beneficial for medical imaging where the data availability is often limited.

The training used discrete-space image datasets, whereas the actual objects in practice are continuous. Ideally, using an accurate continuous image model could better approximate the actual situation. This implicit bias is common for learning-based methods, and may lead to suboptimal results, such as the backtracking in the edge/corner of k-space (Fig. 3.3). The training also ignored physical processes such as relaxation and magnetization transfer. Future studies may consider these processes in the forward system model. The mismatch or domain shift from training to prospective scans may influence the results. For example, there exist differences in protocols (RF pulses, FOVs, and resolutions), hardware (field strengths and Tx/Rx coils), system imperfections (eddy currents, gradient non-linearity, and inhomogeneity), demography, and pathology. Ch. 4 and Ch. 5 tested the optimized trajectory in prospective in-vivo experiments, and discussed practical issues, including eddy currents, and contrast/SNR mismatch between the training set and prospective protocols. Subsequent studies should evaluate the robustness of learned sampling trajectories in more scenarios.

CHAPTER 4

Joint Optimization of 2D Sampling Trajectories and Model-based Deep Learning-based Reconstruction

4.1   Introduction

In MRI, most scanning protocols sample data points sequentially according to a pre-determined sampling pattern. The most common sampling patterns are variants of Cartesian rasters and non-Cartesian trajectories such as radial spokes and spiral interleaves. The local smoothness of these patterns facilitates ensuring that they obey hardware limits, namely the maximum gradient and slew rate that constrain the speed and acceleration when traversing k-space. These patterns also easily ensure sufficient sampling densities. In recent years, hardware improvements, especially with the RF and gradient systems, enable more complex gradient waveform. For a given readout time, optimized designs can cover a broader and potentially more useful region in k-space, reducing the overall scanning time and/or improving image quality, particularly when combined with multiple receive coils.[1]

For fast imaging, many works focus on acceleration in the phase-encoding (PE) direction with fully sampled frequency-encoding (FE) lines [23, 80, 83, 125, 151]. Usually, there is enough time for Δk shifts in the PE direction, so gradient and slew rate constraints are readily satisfied. More general non-Cartesian trajectory designs in 2D and 3D can further exploit the flexibility in the FE direction. However, in addition to hardware physical constraints, MRI systems are affected by imperfections such as the eddy currents that cause the actual trajectory to deviate from the nominal one and introduce undesired phase fluctuations in the acquired data [116]. Some studies optimize properties of existing trajectories such as the density of spiral trajectories [87] or the rotation angle of radial trajectories [159]. More complex waveforms, e.g., wave-like patterns [14], can provide more uniform coverage of k-space and mitigate aliasing artifacts. To accommodate the incoherence requirements of compressed sensing-based methods, [15, 92] introduce slight perturbations

---

[1]This chapter is based on our journal paper [146].

to existing trajectories, like radial or spiral trajectories. Some works explore genetic algorithms to solve this non-convex, constrained problem [118].

The recent SPARKLING method [85, 86] considers two criteria for trajectory design: (1) the trajectory should match a pre-determined sampling density according to a certain measure, and (2) the sampling points should be locally uniform to avoid clusters or gaps. The density and uniformity criteria are transformed into 'attraction' and 'repulsion' forces among the sampling points. The work uses fast multipole methods (FMM) [42] to efficiently calculate the interactions between points. Projection-based optimization handles the gradient and slew rate constraints [24]. In-vivo and simulation experiments demonstrate that this approach reduces aliasing artifacts for 2D and 3D T_2^*-weighted imaging. However, in SPARKLING, the density is determined heuristically; determining the optimal sampling density for different protocols remains an open problem. The work also does not consider some k-space signal characteristics such as conjugate symmetry. Furthermore, the point spread function (PSF) of the calculated trajectory for high under-sampling rates may be suboptimal for reconstruction algorithms like those based on convolution neural networks, because the reconstruction algorithm is not part of the SPARKLING design process.

With rapid advances in deep learning and auto-differentiation software, learning-based signal sampling strategies are being investigated in multiple fields such as optics and ultrasound [37, 69]. In MRI, most learning-based works have focused on sampling patterns of phase encoding locations. Some studies formulate the on-grid sampling pattern as i.i.d. samples from multivariate Bernoulli distribution [9, 70]. Since random sampling operations are not differentiable, different surrogate gradients, such as Gumbel-Softmax, are developed in these works. Instead of gradient descent, [119] uses a greedy search method. [175] further reduces the complexity of greedy search by Pareto optimization, an evolutionary algorithm for sparse regression [111]. Some works have used reinforcement learning. For example, [74] and [169] adopted a double network setting: one for reconstruction and the other for generating a sampling pattern, where the first work used Monte-Carlo Tree Search (MCTS) and the second used Q-learning to optimize 1D sub-sampling. Instead of using an end-to-end CNN as the reconstruction algorithm in other works, [127] constructs a differentiable compressed sensing reconstruction framework. [1] used an unrolled neural network as the reconstruction algorithm. To our knowledge, PILOT [154] is the first work to optimize a 2D non-Cartesian trajectory and an image reconstruction method simultaneously. The training loss is the reconstruction error since the ultimate goal of trajectory optimization is high image quality. The trained parameters were the locations of sampling points and the weights of the reconstruction neural network. Large datasets and stochastic gradient descent were used to optimize the parameters. To meet the hardware limits, a penalty was applied on the gradient and slew rate. Since the reconstruction involves non-Cartesian data, PILOT uses a (bilinear, hence differentiable almost everywhere) gridding reconstruction algorithm to map the k-space data into the image domain,

followed by a U-Net [117] to refine the gridded image data. Simulation experiments report encouraging results compared to ordinary trajectories. Nevertheless, the algorithm often gets stuck in sub-optimal local minima where the initial trajectory is only slightly perturbed yet the slew rate rapidly oscillates. To reduce the effect of initialization, [154] uses a randomized initialization algorithm based on the traveling salesman problem (TSP) which works only with single-shot long TE sequences, limiting its utility in more applications. The implementation in [154] relies on auto-differentiation to calculate the Jacobian of the non-uniform Fourier transform; here we adopt a new NUFFT Jacobian approximation that is faster and more accurately approximates the non-Cartesian discrete Fourier transform (DFT), as described in Ch. 3.

To overcome the limitations of previous methods and further expand their possible applications, this chapter introduces an improved supervised learning workflow called **B**-spline parameterized **J**oint **O**ptimization of **R**econstruction and **K**-space trajectory (**BJORK**). Our main contributions include the following. (1) We parameterize the trajectories with quadratic B-spline kernels. The B-spline reparameterization reduces the number of parameters and facilitates multilevel optimization, enabling non-local improvements to the initial trajectory. Moreover, the local smoothness of B-spline kernels avoids rapid waveform oscillations. (2) We adopt an unrolled neural network reconstruction method for non-Cartesian sampling patterns [2]. Compared to the image-domain U-Net implemented in previous works, the proposed approach combines the strength of learning-based and model-based reconstruction, improving the effect of both reconstruction and trajectory learning. (3) We adopt accurate and efficient NUFFT-based approximations of the Jacobian matrices of the DFT operations used in the reconstruction algorithm. (See [144] for detailed derivations and validation.) (4) In addition to a simulation experiment, we also conducted phantom and in-vivo experiments with protocols that differ from the training dataset to evaluate the generalizability and applicability of the model. (5) We used a k-space mapping technique to correct potential eddy-current-related artifacts. (6) Compared with SPARKLING, the proposed learning-based approach does not need to assume signal characteristics such as spectrum energy density. Instead, BJORK learns the sampling trajectories from a large data set in a supervised manner.

The remaining materials are organized as follows. Ch. 4.2 details the proposed method. Ch. 4.3 describes experiment settings and control methods. Ch. 4.4 and Ch. 4.5 present and discuss the results.

4.2 Methods

This section describes the proposed approach for supervised learning of the sampling trajectory and image reconstruction method.
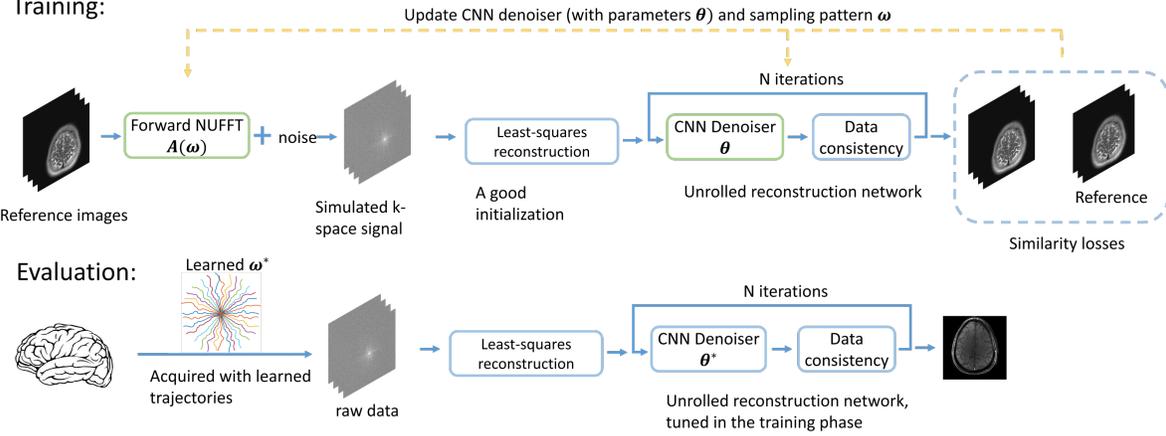
Figure 4.1: Diagram of the proposed BJORK method. To optimize the sampling trajectory and the reconstruction algorithm jointly using a stochastic gradient descent (SGD)-type method, we construct a differentiable forward MRI system matrix $\boldsymbol{A}(\boldsymbol{\omega})$ that simulates k-space data w.r.t. trajectory $\boldsymbol{\omega}$ from ground truth images, and an unrolled neural network for reconstruction. The reconstruction errors compared with the ground truth are used as the training loss to update learnable parameters (the trajectory $\boldsymbol{\omega}$ and the network's parameters $\boldsymbol{\theta}$).

4.2.1 Problem Formulation

Fig. 4.1 shows the overall workflow of the proposed approach. The goal is to optimize $\boldsymbol{\omega} \in \mathbb{R}^{N_\mathrm{s} \times N_\mathrm{d}}$, a trainable (possibly multi-shot) sampling pattern, and $\boldsymbol{\theta} \in \mathbb{R}^M$, the M parameters of the image reconstruction method, where N_s denotes the total number of k-space samples, and N_d denotes the image dimensionality. (The results in this chapter are for $N_\mathrm{d} = 2$, i.e., 2D images, but the method is general.)

The training loss for jointly optimizing the parameters is as follows:

$$\operatorname*{arg\,min}_{\boldsymbol{\omega} \in \mathbb{R}^{N_\mathrm{s} \times N_\mathrm{d}}, \boldsymbol{\theta} \in \mathbb{R}^M} \mathbb{E}_{\boldsymbol{x} \in \mathcal{X}}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{\omega}; \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{\varepsilon}), \boldsymbol{x})] \tag{4.1}$$

$$\text{s.t. } \|\boldsymbol{D}_1\boldsymbol{\omega}^{[d]}\|_\infty \leq \gamma \Delta t g_{\max},$$

$$\|\boldsymbol{D}_2\boldsymbol{\omega}^{[d]}\|_\infty \leq \gamma \Delta t^2 s_{\max}, \ d = 1, \ldots, N_\mathrm{d},$$

where each $\boldsymbol{x} \in \mathbb{C}^{N_\mathrm{v}}$ is a fully sampled reference image having N_v voxels drawn from the training data set $\mathcal{X}$ and $\boldsymbol{\varepsilon}$ is simulated additive complex Gaussian noise. (In practice the expectation is taken over mini-batches of training images.) The system matrix $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{\omega}) \in \mathbb{C}^{N_\mathrm{s}N_\mathrm{c} \times N_\mathrm{v}}$ describes the MR imaging physics (encoding), where N_c denotes the number of receiver coils. For multi-coil non-Cartesian acquisition, it is a non-Cartesian SENSE operator [109] that applies a pointwise multiplication of the sensitivity maps followed by a NUFFT operator (currently we do not consider

field inhomogeneity but it would be straightforward to extend because the Jacobian approximation in Ch. 3 can cover such cases.) The function $f_{\boldsymbol{\theta}}(\boldsymbol{\omega};\cdot)$ denotes an image reconstruction algorithm with parameters $\boldsymbol{\theta}$ that is applied to simulated under-sampled data $\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{\varepsilon}$. As detailed in section 4.2.3, we use an unrolled neural network. The reconstruction loss $\ell(\cdot,\cdot)$ quantifies the similarity between a reconstructed image and the ground truth, and can be a combination of different terms. Here we chose the loss ℓ to be a combined ℓ_1 and square of ℓ_2 norm. The matrices $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ denote the first-order and second-order finite difference operators. Δt is the raster time and γ denotes the gyromagnetic ratio. For multi-shot imaging, the finite difference operator applies to each shot individually. The optimization is constrained by maximum gradient field strength ($g_{\max}$), and slew rate ($s_{\max}$). To use the stochastic gradient descent (SGD) method, we convert the box constraint into a penalty function ϕ, where

$$\phi_\lambda(|\boldsymbol{x}|) = \mathbf{1}^T \max.(|\boldsymbol{x}| - \lambda, 0),$$

where $\max.(\cdot)$ operates point-wisely. Our final optimization problem has the following form:

$$\underset{\boldsymbol{\omega}\in\mathbb{C}^{N_\mathrm{s}\times N_\mathrm{d}},\boldsymbol{\theta}\in\mathbb{R}^M}{\arg\min} \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}\big[\ell(f_{\boldsymbol{\theta},\boldsymbol{\omega}}(\boldsymbol{\omega};\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{\varepsilon}), \boldsymbol{x})\big] + \mu_1\phi_{\gamma\Delta t g_{\max}}(|\boldsymbol{D}_1\boldsymbol{\omega}|) + \mu_2\phi_{\gamma\Delta t^2 s_{\max}}(|\boldsymbol{D}_2\boldsymbol{\omega}|).$$

We update $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ simultaneously for each mini-batch of training data.

4.2.2   Parameterization and Multi-level Optimization

We parameterize the sampling pattern with 2nd-order quadratic B-spline kernels:

$$\boldsymbol{\omega}^{[d]} = \boldsymbol{B}\boldsymbol{c}^{[d]}, \; d = 1,\ldots,N_\mathrm{d}, \tag{4.2}$$

where $\boldsymbol{B} \in \mathbb{R}^{N_\mathrm{s}\times L}$ denotes the interpolation matrix, and $\boldsymbol{c}^{[d]}$ denotes the dth column of the coefficient matrix $\boldsymbol{c} \in \mathbb{R}^{L\times N_\mathrm{d}}$. L denotes the length of $\boldsymbol{c}^{[d]}$, or the number of interpolation kernels in each dimension. The decimation rate in Fig. 4.8 is defined as Decim. $= N_\mathrm{s}/L$. B-spline kernels reduce the number of individual inequality constraints (on maximum gradient strength and slew rate) from $4N_\mathrm{d}N_\mathrm{s}$ to $4N_\mathrm{d}L$ where typically $L \ll N_\mathrm{s}$. See [61] for more details.

Early versions of previous work [154] and our preliminary experiments found optimized trajectories that were often local minima near the initialization, only slightly perturbing the initial trajectory. We use a multilevel training strategy to improve the optimization process [17].

We initialized the decimation rate N_s/L with a large value (like 64). Thus, many neighboring sample points are controlled by the same coefficient, which introduces more 'non-local' improvements. After both $\boldsymbol{c}$ and $\boldsymbol{\theta}$ converge, we reduce the decimation rate, typically by a factor of 2, and

begin a new round of training initialized with $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ of the previous round. Fig. 4.8 depicts the evolution along with decimation rates.

4.2.3 Reconstruction

In the joint learning model, we adopted a model-based unrolled neural network (UNN) approach to image reconstruction [2, 60, 121, 163]. Compared to the previous joint learning model (PILOT) that used a single image domain network [154], an unrolled network can lead to a more accurate reconstruction [2], at the price of longer reconstruction time.

A typical cost function for regularized MR image reconstruction has the form:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathsf{R}(\boldsymbol{x}). \tag{4.3}$$

The first term is usually called the data-consistency term that ensures the reconstructed image is consistent with the acquired k-space data $\boldsymbol{y}$. (In the training phase, $\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{\varepsilon}$ is the simulated $\boldsymbol{y}$.) The regularization term $\mathcal{R}(\cdot)$ is designed to control aliasing and noise when the data is under-sampled. By introducing an auxiliary variable $\boldsymbol{z}$, one often replaces (4.3) with the following alternative:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \min_{\boldsymbol{z}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathsf{R}(\boldsymbol{z}) + \mu\|\boldsymbol{x} - \boldsymbol{z}\|_2^2, \tag{4.4}$$

where $\mu > 0$ is a penalty parameter. Using an alternating minimization approach, the optimization updates become:

$$\boldsymbol{x}_{i+1} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mu\|\boldsymbol{x} - \boldsymbol{z}_i\|_2^2, \tag{4.5}$$

$$\boldsymbol{z}_{i+1} = \arg\min_{\boldsymbol{z}} \mathsf{R}(\boldsymbol{z}) + \mu\|\boldsymbol{x}_{i+1} - \boldsymbol{z}\|_2^2. \tag{4.6}$$

The analytical solution for the $\boldsymbol{x}$ update is

$$\boldsymbol{x}_{i+1} = (\boldsymbol{A}'\boldsymbol{A} + \mu\boldsymbol{I})^{-1}(\boldsymbol{A}'\boldsymbol{y} + \mu\boldsymbol{z}_i),$$

which involves a matrix inverse that would be computationally prohibitive to compute directly. Following [2], we use a few iterations of the conjugate gradient (CG) method for the $\boldsymbol{x}$ update. The implementation uses the Toeplitz embedding trick to accelerate the computation of $\boldsymbol{A}'\boldsymbol{A}$ [40, 101].

For a mathematically defined regularizer, the $\boldsymbol{z}$ update would be a proximal operator. Here we follow previous work [2, 49] and use a CNN-based denoiser $\boldsymbol{z}_{i+1} = \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_{i+1})$. To minimize memory usage and avoid over-fitting, we used the same $\boldsymbol{\theta}$ across iterations, though iteration-specific networks may improve the reconstruction result [121].

For the CNN-based denoiser, we used the Deep Iterative Down-Up CNN (DIDN) [121, 166].

As a state-of-art model for image denoising, the DIDN model uses less memory than popular models like U-Net [117] with improved reconstruction results. In our experiments, it also led to faster training convergence than previous denoising networks.

Since neural networks are sensitive to the scale of the input, a good and consistent initial estimate of x is important. We used the following quadratic roughness penalty approach to compute an initial image estimate:

$$x_0 = \arg\min_x \|Ax - y\|_2^2 + \lambda\|Tx\|_2^2 \tag{4.7}$$
$$= (A'A + \lambda T'T)^{-1} A'y,$$

where T denotes the N_d-dimensional first-order finite difference (roughness) operator. We also used the CG method to (approximately) solve this quadratic minimization problem.

4.2.4 Correction of Eddy-current Effects

Rapidly changing gradient waveforms may suffer from eddy-current effects, even with shielded coils. This hardware imperfection requires additional measurements and corrections so that the actual sampling trajectory is used for reconstructing real MRI data. Some previous works used a field probe and corresponding gradient impulse-response (GIRF) model [142]. In this chapter, we adopted the 'k-space mapping' method [35, 116] that does not require additional hardware. Instead of mapping the k_x and k_y components separately as in previous papers, we excited a pencil-beam region using one $90°$ flip and a subsequent $180°$ spin-echo pulse [103]. We averaged multiple repetitions to estimate the actual acquisition trajectory. We also subtracted a zero-order eddy current phase term from the acquired data [116].

The following pseudo-code summarizes the BJORK training process.

4.3 Experiments

Table 4.1: Protocols for data acquisition

Protocols for the prospective experiment:

Name	FOV(cm)	dz(mm)	Gap(mm)	TR(ms)	TE(ms)	FA	Acqs	dt(us)	Time
Radial-like	22*22*4	2	0.5	318.4	3.56	90°	32*1280	4	0:11
Radial-full	22*22*4	2	0.5	318.4	3.56	90°	320*1280	4	1:40

dz: slice thickness; Gap: gap between slices; Acqs: number of shots * readout points; FA: flip angle

---

Algorithm 1 Training algorithm for BJORK

Require: Training set $\mathcal{X}$; denoiser $\mathcal{D}_{\boldsymbol{\theta}}$ for initial CNN weights $\boldsymbol{\theta}_0$; initial trajectory $\boldsymbol{\omega}_0$; levels of optimization N_{level}; number of epoch N_{epoch}; step size of denoiser update $\eta_{\mathcal{D}}$; step size of trajectory update $\eta_{\boldsymbol{\omega}}$; penalty parameter for gradient/slew rate constraint μ_1 and μ_2.

Ensure: $\boldsymbol{\omega} = \boldsymbol{B}\boldsymbol{c}$

1: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
2: $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega}_0$
3: Pre-train $\mathcal{D}_{\boldsymbol{\theta}}$ with fixed $\boldsymbol{\omega}_0$.
4: **for** $l = 1$ to N_{level} **do**
5:     Initialize new coefficient matrix $\boldsymbol{B}_l$.
6:     Initialize new coefficient $\boldsymbol{c}_l^0$ with $\boldsymbol{\omega}_{l-1} \approx \boldsymbol{B}_l \boldsymbol{c}_l^0$.
7:     **for** $j = 1$ to N_{epoch} **do**
8:         **for** training batches $\boldsymbol{x}^K$ in $\mathcal{X}$ **do**
9:             **Simulate the k-space w.r.t. $\boldsymbol{\omega}_l$:**
10:             $\boldsymbol{y}^K = \boldsymbol{A}(\boldsymbol{\omega}_l^K)\boldsymbol{x}^K + \boldsymbol{\varepsilon}$
11:             **Reconstruction with UNN:**
12:             Reconstruct initial images using (4.7) with CG
13:             **for** $i = 1$ to N_{iter} **do**
14:                 $\boldsymbol{x}_{i+1}$: UNN reconstruction update of $\boldsymbol{z}_i$ using (4.5)
15:                 Apply CNN: $\boldsymbol{z}_{i+1} = \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_{i+1})$
16:             **end for**
17:             **Calculate loss function:**
18:             $L = \ell(\hat{\boldsymbol{x}}^K, \boldsymbol{x}^K) + \mu_1 \phi_{\gamma \Delta t g_{\max}}(|\boldsymbol{D}_1 \boldsymbol{\omega}_i^K|) + \mu_2 \phi_{\gamma \Delta t^2 s_{\max}}(|\boldsymbol{D}_2 \boldsymbol{\omega}_i^K|)$
19:             **Update denoiser and trajectory:**
20:             $\boldsymbol{\theta}^K = \boldsymbol{\theta}^{K-1} - \eta_{\mathcal{D}} \nabla_{\boldsymbol{\theta}^{K-1}} L$
21:             $\boldsymbol{\omega}_l^K = \boldsymbol{\omega}_l^{K-1} - \eta_{\boldsymbol{\omega}} \nabla_{\boldsymbol{\omega}_l^{K-1}} L$
22:         **end for**
23:     **end for**
24: **end for**

We compared the proposed BJORK approach with the SPARKLING method for trajectory design in all experiments, and have set the readout length and physical constraints to be the same for both methods.

Both BJORK and PILOT [154] are methods for joint sampling design and reconstruction optimization. We compared three key differences between the two methods individually. (1) The NUFFT Jacobian matrices, as discussed in [144]. (2) The reconstruction method involved. Our BJORK approach uses an unrolled neural network, while PILOT uses a single reconstruction neural network in the image domain (U-Net). We also presented the effect of trajectory parameterization (BJORK uses quadratic B-splines following [61], whereas versions 1-3 of PILOT used no parameterization and more recent versions of PILOT use cubic splines [154]).

To evaluate the reconstruction quality provided by different trajectories, we used two types of reconstruction methods in the test phase: unrolled neural network (UNN) (with learned θ) and a compressed sensing approach (sparsity regularization for a discrete wavelet transform). For SPARKLING-optimized trajectories and standard undersampled trajectories (radial/spiral), we used the same unrolled neural networks as in BJORK for reconstruction. Only the network parameters θ were trained, with the trajectory ω fixed.

We also used compressed sensing-based reconstruction to test the generalizability of BJORK-optimized trajectories. The penalty function is the ℓ_1 norm of a discrete wavelet transform with a Daubechies 4 wavelet. The ratio between the penalty term and the data-fidelity term is 10^{-7}. We used the SigPy package[2] and its default primal-dual hybrid gradient (PDHG) algorithm (using 50 iterations). This study includes two evaluation metrics: the structural similarity metric (SSIM) and peak signal-to-noise ratio (PSNR) [68].

For both simulation and real acquisition, the acquisition sampling time and gradient raster time are both 4 μs, with a target matrix size of 320×320. The maximum gradient strength is 50 mT/m, and the maximum slew rate is 149 T/m/s, which were set to limit peripheral nerve stimulation.

To demonstrate the proposed model's adaptability, we investigated two types of initialization of waveforms: an undersampled in-out radial trajectory with a shorter readout time ($\sim$5 ms) and an undersampled center-out spiral trajectory with a longer readout time ($\sim$16 ms). For the in-out radial initialization, the number of spokes is 16/24/32, and each spoke has 1280 points of acquisition (4 μs samples). The rotation angle is equidistant between $-\pi/2$ and $\pi/2$. For the center-out spiral initialization, the number of spokes is 8, and each leaf has $\sim$4000 points of acquisition. We used the variable-density spiral design package[3] from [87]. For SPARKLING, $\tau = 0.6$ and $d = 2.5$ for 32-spoke radial and $\tau = 0.5$ and $d = 2$ for 8-shot spiral ([85, Eqn. 8]) which are selected by grid search using CS-based reconstruction.

---

[2] https://github.com/mikgroup/sigpy
[3] https://mrsrl.stanford.edu/~brian/vdspiral/

The simulation experiments used the NYU fastMRI brain dataset to train the trajectories and neural networks [167]. The dataset consists of multiple contrasts, including T_1w (23220 slices), T_2w (42250 slices), and FLAIR (5787 slices). FastMRI's knee subset was also used in a separate training run to investigate the influence of training data on learned sampling patterns. The central 320×320 region was cropped (or zero-filled). Sensitivity maps were estimated using the ESPIRiT method [141] with the central 24 phase-encoding lines, and the corresponding conjugate phase reconstruction was regarded as the ground truth during training.

The batchsize was 4. The number of blocks or the number of outer iterations for the unrolled neural network was 6. The weight μ in (4.4) could also be learned, but this operation would double the computation load with minor improvement. We set $\mu = 2$. The number of training epochs was set to 3 for each level of B-spline kernel length, which is empirically enough for the training to converge. We used $N_{\text{level}} = 4$ optimization levels, and $N_{\text{epoch}} = 3$ so the total number of epochs was 12. We set $N_{\text{iter}} = 6$ of the unrolled neural network. For training the reconstruction network with existing trajectories (radial, spiral, and SPARKLING-optimized), we also used 12 training epochs. We used the Adam optimizer [79], with parameter $\boldsymbol{\beta} = [0.5, 0.999]$, for both trajectories $\boldsymbol{\omega}$ and network parameters $\boldsymbol{\theta}$. The learning rate linearly decayed from 10^{-3} to 0 for the trajectory update, and from 10^{-5} to 0 for the network update. We did not observe obvious over-fitting phenomena on the validation set. The training on an Intel Xeon Gold 6138 CPU and an Nvidia RTX2080Ti GPU took around 120-150 hours.

4.3.1 Prospective Studies

Table 4.1 details the scanning protocols of the RF-spoiled, gradient echo (GRE) sequences used. For in-vivo acquisitions, a fat-saturation pulse was applied before the tip-down RF pulse. We chose the TR and FA combination for desired T_1w contrast. For radial-like sequences, we tested a GRE sequence with 3 different readout trajectories: standard undersampled radial, BJORK initialized with undersampled radial, and SPARKLING initialized with undersampled radial. Radial-full denotes the fully sampled radial trajectory. Both simulation experiments (training) and real experiments used the same readout trajectory.

We also acquired an additional dual-echo Cartesian GRE image for generating the sensitive map and (potentially) B_0 map. The sensitivity maps were generated by ESPIRiT [141] methods. The sequences were programmed with TOPPE [103], and implemented on a GE MR750 3.0T scanner with a Nova Medical 32-channel head coil. Subjects gave informed consent under local IRB approval. For phantom experiments, we used a water phantom with 3 internal cylinders.

The k-space mapping was implemented on a water phantom. The thickness of the pencil beam was 2mm $\times$ 2mm. The trajectory estimates were based on an average of 30 repetitions.

34

Table 4.2: Quantitative results for simulation experiments

SSIM:

		Standard	SPARKLING	BJORK
radial-like	UNN	0.957	0.963	**0.968**
	CS	0.932	0.946	**0.956**
spiral-like	UNN	0.986	0.989	**0.990**
	CS	0.976	0.978	**0.981**

PSNR (in dB):

		Standard	SPARKLING	BJORK
radial-like	UNN	35.0	36.0	**36.9**
	CS	33.9	35.7	**36.3**
spiral-like	UNN	40.9	41.7	**41.9**
	CS	39.9	40.4	**40.7**

16 spokes   24 spokes   32 spokes
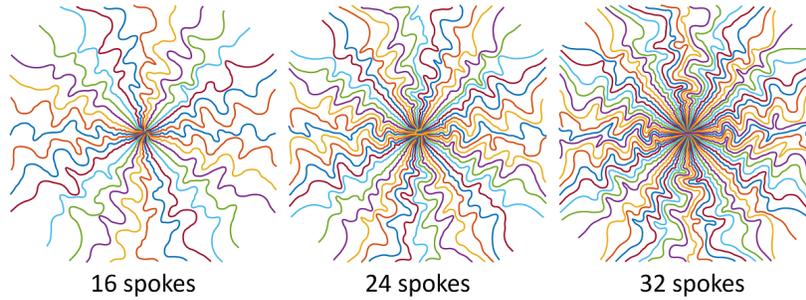
Figure 4.2: Learned radial-like trajectories with different acceleration ratios.
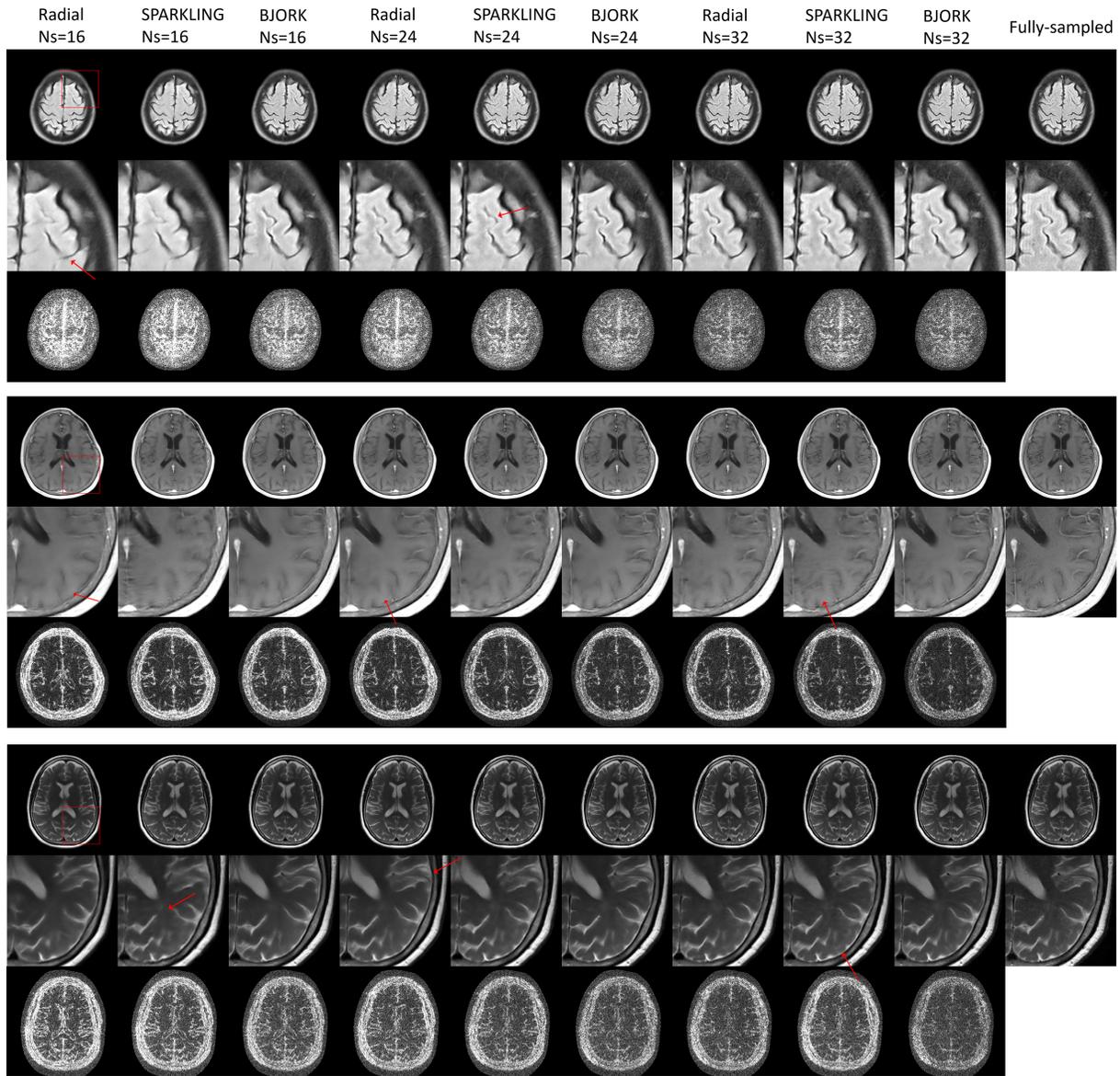
Figure 4.3: Examples from the simulation experiment using the UNN-based reconstruction algorithm, with three different acceleration ratios. N_s stands for the number of shots or spokes. The first slice is FLAIR contrast. The second slice is T_1w contrast. The third slice is T_2w contrast. Red boxes indicate zoom-in regions and red arrows point to reconstruction artifacts/blur. Below the zoomed-in regions are the corresponding error maps, compared with fully sampled images.
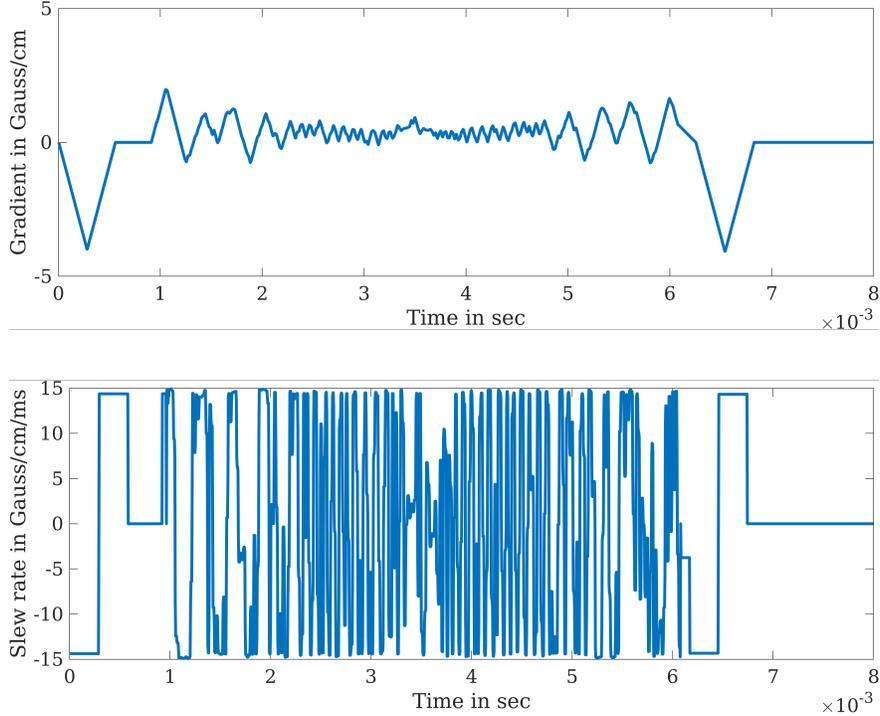
Figure 4.4: The gradient strength and slew rate of one spoke from BJORK-optimized radial trajectory.

4.4 Results

4.4.1 Simulation Study

The test set includes 1520 slices, and the validation set includes 500 slices. Table 4.2 shows the quantitative results (SSIM and PSNR). The radial-like sequences have 32 spokes. Fig. 4.2 displays optimization results under different acceleration ratios. Fig. 4.3, Fig. 4.4, and Fig. 4.5 display the reconstruction results, gradient waveforms, and training loss curve of the optimization results. The proposed method has significant improvement compared with un-optimized trajectories ($P < 0.005$). It also has improved reconstruction quality compared with SPARKLING considering unrolled neural network-based reconstruction. Compared to the undersampled radial trajectory or SPARKLING trajectory, the proposed method has a better restoration of details and lower levels of artifacts. In the experiment, different random seeds in the training led to nearly identical learned sampling trajectories.

Fig. 4.6 displays point spread functions of 32-spoke radial-like trajectories. The BJORK's PSF has a narrower central lobe than SPARKLING and fewer streak artifacts than standard radial. Fig. 4.7 shows the approximate conjugate symmetry relationship implicitly learned in the BJORK trajectory. However, it is important to note that quadratic reconstruction, as discussed in Ch. 3,
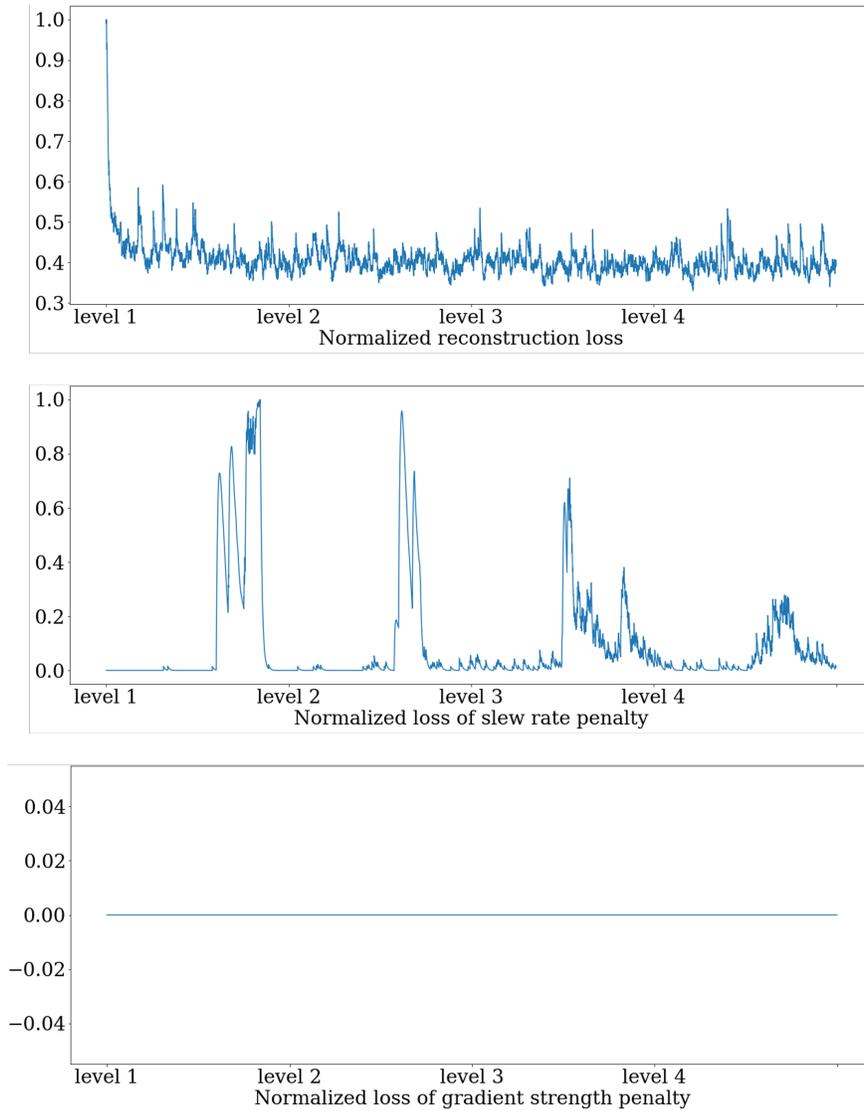
Figure 4.5: Smoothed training losses of a 16-spoke radial-initialized sequence. We use 4 levels and each level contains 3 epochs. The three columns are the reconstruction loss, penalty on the maximum slew rate, and penalty on the maximum gradient strength.
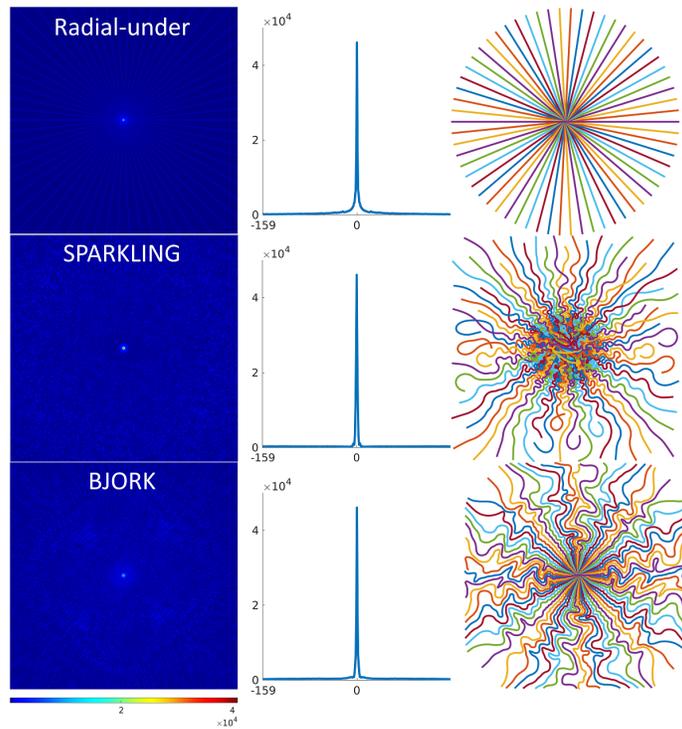
Figure 4.6: PSFs of different sampling patterns. Each middle plot is the averaged profile of different views (angles) through the origin. The FWHM for undersampled radial, BJORK, and SPARKLING are 1.5, 1.6, and 2.1 pixels, respectively.
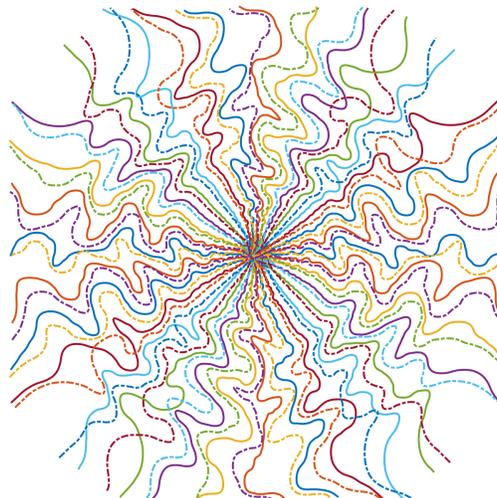
Figure 4.7: The dash-dot line shows the $180°$ rotated BJORK trajectory. The original and rotated trajectory have little overlap, suggesting that the BJORK automatically discovered a sampling pattern that exploits the (approximate) k-space Hermitian symmetry.

does not display obvious conjugate symmetry, as indicated by the Jacobian expression in (3.6).

Fig. 4.8 shows the evolution of sampling patterns using our proposed multi-level optimization. Different widths of the B-spline kernels introduce different levels of improvement as the trajectory is optimized. Fig. 4.8 also shows a nonparametric trajectory that is similar to early versions of the PILOT paper [154, versions 1-3]. Directly optimizing sampling points seems only to introduce a small perturbation than the initialization.

Fig. 4.9 shows radial-initialized trajectories trained by BJORK with brain and knee datasets. Different trajectories are learned from different datasets. We hypothesize that the difference is related to frequency distribution of energy, as well as the noise level, which requires further study. This phenomenon was also observed in [9].

| Init. | Decim=32 | Decim=16 | Decim=8 | Decim=4 | Nonparametric |
| SSIM=0.957 | SSIM=0.963 | SSIM=0.965 | SSIM=0.967 | SSIM=0.968 | SSIM=0.959 |
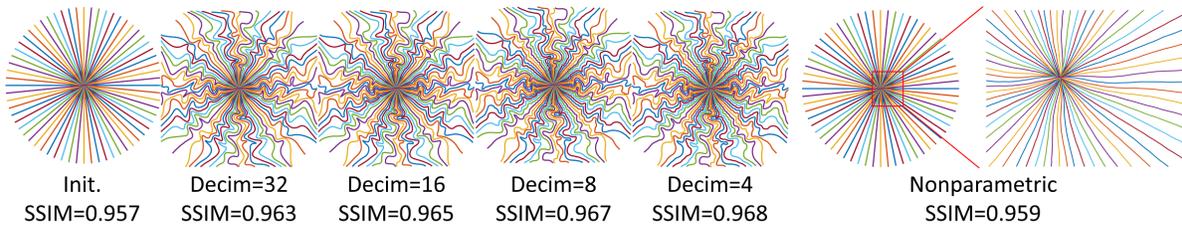
Figure 4.8: The evolution of the learned trajectories. Decim denotes $N_{\rm s}/L$ in (4.2). Nonparametric means the locations of each sampling points are independent trainable variables, rather than being parameterized by quadratic B-spline kernels. SSIM denotes the average reconstruction quality on the evaluation set of each level. The rightmost zoomed-in set shows the very small perturbations produced by the nonparametric approach (stuck into local-minima).

Table 4.3: Effects of different contrasts on learned models.

test \ training	T_1w	T_2w	FLAIR
T_1w +noise	0.981	0.980	0.981
T_2w +noise	0.951	0.953	0.953
FLAIR+noise	0.974	0.974	0.975

To explore the influence of image contrast in the training set, we trained the model with one image contrast from the fastMRI brain dataset (without simulated additive noise), and tested the learned trajectory with all contrasts (with simulated additive Gaussian noise whose variance is 10^{-3} of the mean magnitude of the signal). Each contrast has 4500 training slices and 500 test slices. We fine-tuned the reconstruction unrolled neural network for different test contrasts. The initialization is a 16-spoke radial trajectory. Table 4.3 reports the average reconstruction quality. The learned trajectories are insensitive to different contrasts within the fastMRI dataset.
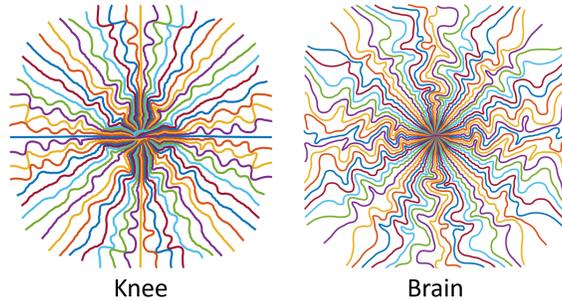
Figure 4.9: Trajectories learned from different datasets.

To test the influence of reconstruction methods on trajectory optimization, we tried a single image-domain refinement network as the reconstruction method in the joint learning model, similar to PILOT's approach. Quadratic roughness penalty reconstruction in (4.7) still is the network's input. The initialization of the sampling pattern is an undersampled radial trajectory. Table 4.4 shows that the proposed BJORK reconstruction method (unrolled neural network, UNN) improves reconstruction quality compared to a single end-to-end model. Such improvements are consistent with other comparisons between UNN methods and image-domain CNN methods using fixed sampling patterns (reconstruction only) [2, 121, 163].

To show the effect of warm initialization, we compared two inputs of the unrolled neural network: the adjoint of undersampling signal ($\boldsymbol{A}'\boldsymbol{y}$) and quadratic roughness penalized reconstruction $(\boldsymbol{A}'\boldsymbol{A} + \lambda\boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{A}'\boldsymbol{y}$. The experiment optimized a 16-spoke radial trajectory and used 1520 test slices. The average reconstruction quality (SSIM values) of the two settings are 0.944 and 0.950, respectively.

Table 4.4: Effects of different reconstruction networks involved in the joint learning model

	SSIM	PSNR(dB)
UNN	**0.968**	**36.9**
Single U-Net	0.934	32.8

4.4.2 Prospective Experiments

Fig. 4.10 shows the results of a water phantom for different reconstruction algorithms. The rightmost column is the fully-sampled ground truth (Radial-full). Note that the unrolled neural network (UNN) here was trained with fastMRI brain dataset, and did not receive fine-tuning in all prospective experiments. The BJORK-optimized trajectory leads to fewer artifacts and improved contrast for the UNN-based reconstruction.

CS Recon

UNN Recon

Radial-under   SPARKLING   BJORK   Radial-full
                                                         (ground truth)
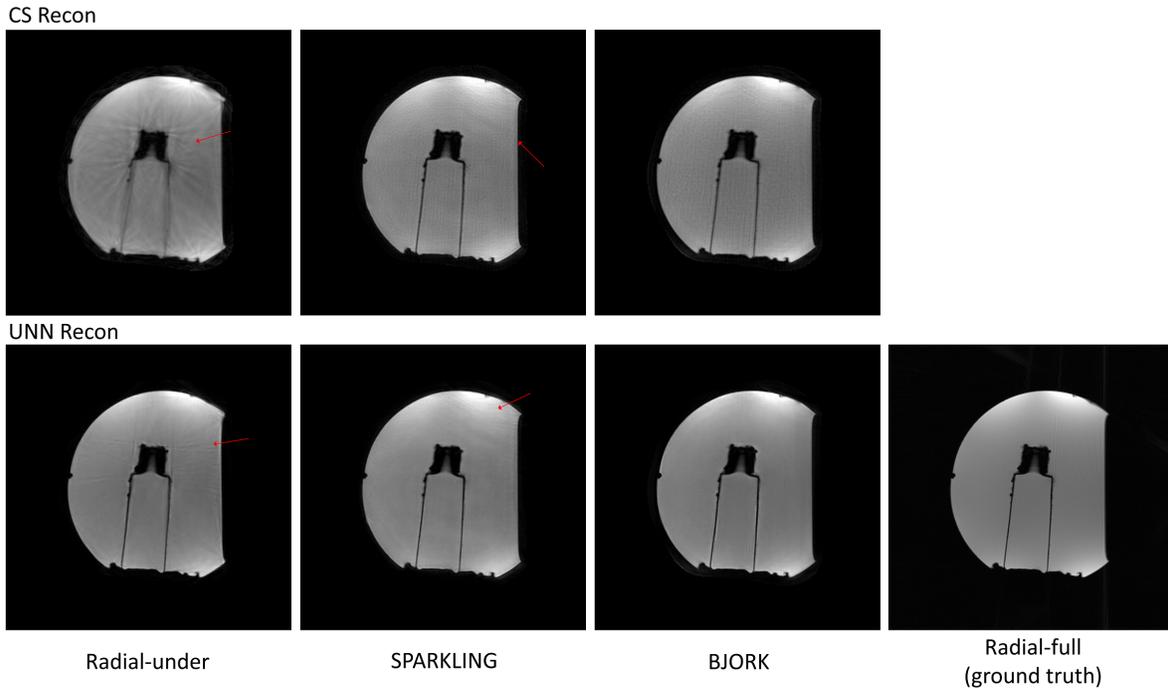
Figure 4.10: Representative results of the prospective phantom experiment using CS-based and UNN-based reconstruction algorithms. The sequences involved were radial-like GRE (detailed in Table 4.1) with T_1w contrast. The parameters of UNNs are trained with fastMRI dataset without fine-tuning. The readout length was 5.12 ms, and we used 32/320 spokes for undersampled (Radial-Under, SPARKLING, BJORK) trajectories and the fully-sampled radial trajectory, respectively.
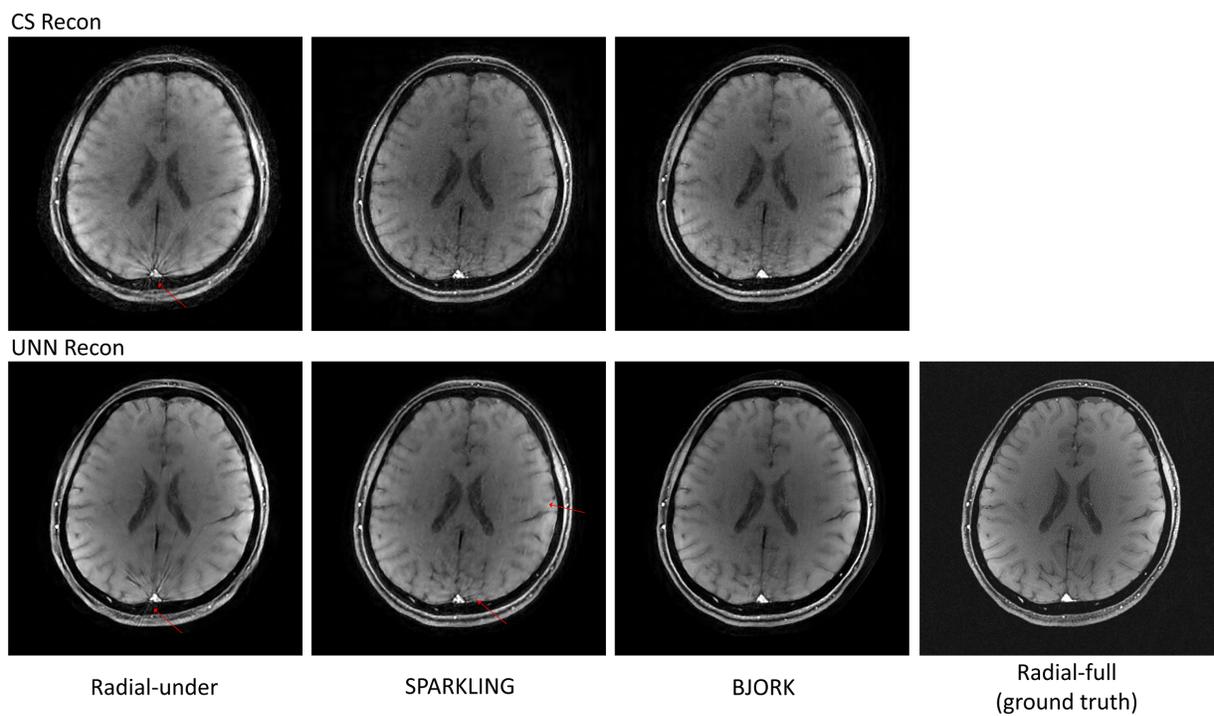
Figure 4.11: Results of the T_1w prospective in-vivo experiment. The trajectories were also radial-like (detailed in Table 4.1). The readout time was 5.12 ms. The parameters of UNNs are trained with the fastMRI dataset without fine-tuning. The number of shots for undersampled trajectories was 32, and for the fully-sampled radial trajectories was 320 ($10\times$ acceleration). The FOV was 22 cm. Red arrows point out reconstruction artifacts.

Fig. 4.11 showcases one slice from the in-vivo experiment. For CS-based reconstruction, the undersampled radial trajectory exhibits stronger streak artifacts than SPARKLING- and BJORK-optimized trajectories. For UNN-based reconstruction, all trajectories' results show reductions of artifacts compared to CS-based reconstruction. The proposed method restores most of the structures and fine details, with minimal artifacts.

4.4.3 Correction of Eddy Currents

Fig. 4.12 displays the CS-based reconstruction of real acquisitions reconstructed using both the nominally designed trajectories and the measured trajectories.

Fig. 4.13 shows the results of the trajectory measurements. Using the measurement of the actual trajectory seems to mitigate the influence of eddy current effects in the reconstruction results.
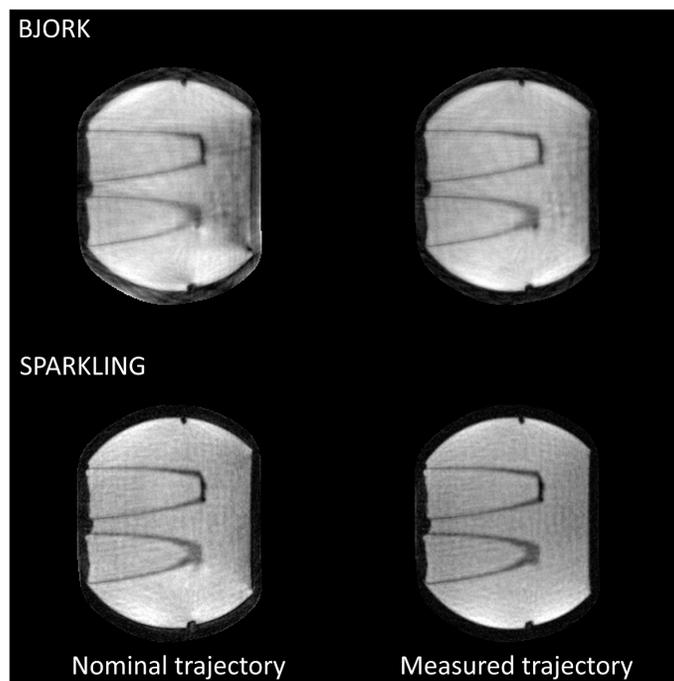
Figure 4.12: Compressed sensing-based reconstruction of a water phantom. The left column is the reconstruction with the nominal trajectory, and right is with the measured trajectory. Reconstruction with the mapped trajectory introduced fewer artifacts.

4.5 Discussion

This chapter proposes an efficient learning-based framework for the joint design of MRI sampling trajectories and NN-based reconstruction parameters. Defining an appropriate objective function
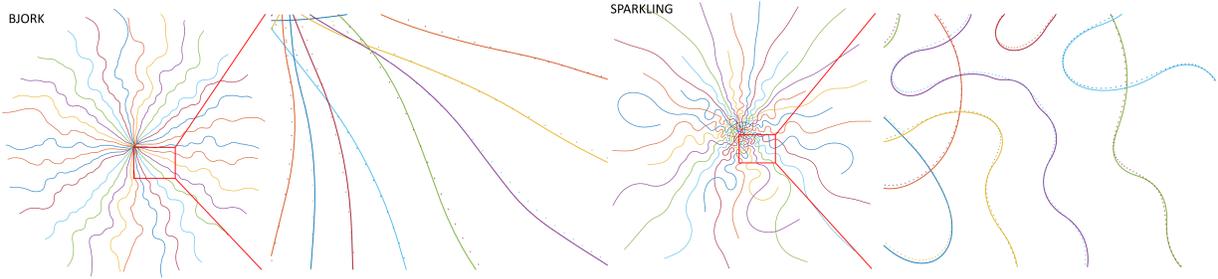
Figure 4.13: The influence of eddy currents on readout waveform. The solid line is the nominal trajectory, and the dotted line is the measurement.

for trajectory optimization is an open question. We circumvented this long-lasting problem by directly using the reconstruction quality as the training loss function in a supervised learning paradigm. The workflow includes a differentiable reconstruction algorithm for which the learning process obtains an intermediate gradient w.r.t. the reconstruction loss. However, solely depending on backpropagation and stochastic gradient descent cannot guarantee optimal results for this non-convex problem. To improve the training effect, we adopted several techniques, including trajectory parameterization, multi-level training, warm initialization of the reconstruction network, and an accurate approximation of NUFFT's Jacobian (Ch. 3). Results show that these approaches can stabilize the training and provide better local minimizers than previous methods.

We trained an unrolled neural network-based reconstruction method for non-Cartesian MRI data. The single image-domain network used in previous work does not efficiently remove aliasing artifacts. Additionally, the k-space 'hard' data-consistency trick for data fidelity [97, 120] is inapplicable for non-Cartesian sampling. An unrolled algorithm can reach a balance between data fidelity and the de-aliasing effect across multiple iterations. For 3D trajectory design using the proposed approach, the unrolled method's memory consumption can be huge. More memory-efficient reconstruction models, such as the memory-efficient network [88] should be explored in further study. We would also investigate recent calibration-less unrolled neural networks, which do not require external sensitivity maps, and shows improved performance relative to MoDL [100].

For learning-based medical imaging algorithms, one main obstacle towards clinical application is the gap between simulation and the physical world. Some factors include the following.

First, inconsistency exists between training datasets and real-world acquisitions, such as different vendors and protocols, posing a challenge to reconstruction algorithms' robustness and generalizability. Our training dataset consisted of T_1w/T_2w/FLAIR Fast Spin-Echo (FSE or TSE) sequences, acquired on Siemens 1.5T/3.0T scanners. The number of receiver channels includes 4, 8, and 16, etc. We conducted the in-vivo/phantom experiment on a 3.0T GE scanner equipped with a 32-channel coil. The sequence is a GRE sequence that has lower SNR compared to FSE

sequences in the training set. Despite the very large differences with the training set, our work still demonstrated improved and robust results in the in-vivo and phantom experiment, without any fine-tuning.

We hypothesize that several factors could contribute to the generalizability: (1) the reconstruction network uses the quadratic roughness penalized reconstruction as the initialization, normalized by the median value. Previous works typically use the adjoint reconstruction as the input of the network. In comparison, our regularized initialization helps provide consistency between different protocols, without too much compromise of the computation time/complexity, (2) the PSF of the learned trajectory has a compact central lobe, without significant streak artifacts. Therefore, the reconstruction is reduced to a de-blurring/denoising task that is a local low-level problem and thus may require less training data than de-aliasing problems. For de-blurring of natural images, networks are usually adaptive to different noise levels and color spaces, and require small cohorts of data [90, 102]. For trajectories like radial and SPARKLING, in contrast, a reconstruction CNN needs to remove global aliasing artifacts, such as streak artifacts and ringing artifacts. The dynamics behind the neural network's ability to resolve such artifacts is still an unsolved question, and the training requires a large amount of diverse data. Other theories, such as restricted isometry property (RIP), may also be exploited to probe the properties of optimized sampling trajectories.

Secondly, it is not easy to simulate system imperfections like eddy currents and off-resonance in the training phase. These imperfections can greatly affect image quality in practice. We used a trajectory measurement method to correct the eddy-current effect. Future works should consider field inhomogeneity in the workflow.

Furthermore, even though the BJORK sampling was optimized for a UNN reconstruction method, the results in Fig. 4.10 and Fig. 4.11 suggest that the learned trajectory is also useful with a CS-based reconstruction method or other model-based reconstruction algorithms. This approach can still noticeably improve the image quality by simply replacing the readout waveform in the existing workflow, promoting the applicability of the proposed approach, similar to [9]. We plan to apply the general framework to optimize a trajectory for (convex) CS-based reconstruction and compare it to the (non-convex) open-loop UNN approach in future work.

Though the proposed trajectory is learned via a data-driven approach, it can also reflect the ideas behind SPARKLING and Poisson disk sampling: sampling patterns having large gaps or tight clusters of points are inefficient, and the sampling points should be somewhat evenly distributed (but not too uniform). Furthermore, BJORK appears to have learned some latent characteristics, like the conjugate symmetry for these spin-echo training datasets. To combine both methods' strengths, a promising future direction is to use SPARKLING as a primed initialization of BJORK.

The learning used here exploited a big public data set. As is shown in the results, knee imaging and brain imaging led to different learned trajectories. This demonstrates that the data set can

greatly influence the optimization results, as was observed in [9]. We also implemented a complementary experiment on a smaller training set (results not shown). We found that a small subset (3000 slices) also led to similar learned trajectories. Therefore, for some organs where a sizeable dataset is not publicly available, this approach may still work with small-scale private datasets. To examine the influence of scanner models, field strength, and sequences, follow-up studies should investigate more diverse datasets.

The eddy-current effect poses a long-term problem for non-Cartesian trajectories and impedes their widespread clinical use. This work used a simple k-space mapping technique as the correction method. The downside of this method is its long calibration time, although it can be performed in a scanner's idle time. This method is waveform-specific, which means that correction should be done for different trajectories. Other methods relying on field probes can get a more accurate correction with less time, albeit with dedicated hardware. In a future study, the eddy current-related artifacts could be simulated according to the GIRF model in the training phase, so that the trajectory is learned to be robust against eddy-current effects.

Aside from practical challenges with GPU memory, the general approach described here is readily extended from 2D to 3D sampling trajectories, which will be discussed in Ch. 5 A more challenging future direction is to extend the work to dynamic imaging applications like fMRI and cardiac imaging, where both the sampling pattern and the reconstruction method should exploit redundancies in the time dimension, e.g., using low-rank models [72]. To optimize sampling in higher dimensions, the next chapter will introduce additional regularizations on the PNS effect.

CHAPTER 5

Optimization of High-dimensional Sampling Trajectories

5.1 Introduction

Most magnetic resonance imaging systems sample data in the frequency domain (k-space) following prescribed sampling trajectories. Efficient sampling strategies can accelerate acquisition and improve image quality. Many well-designed sampling strategies and their variants, such as spiral, radial, CAIPIRINHA, and PROPELLER [3, 19, 84, 108], have enabled MRI's application to many areas [43, 48, 75, 165]. Sampling patterns in k-space are either located on the Cartesian raster or arbitrary locations (non-Cartesian sampling). This paper focuses on optimizing 3D non-Cartesian trajectories and introduces a generalized gradient-based optimization method for automatic trajectory design or tailoring.[1]

The design of sampling patterns usually considers certain properties of k-space signals. For instance, the variable density (VD) spiral trajectory [87] samples more densely in the central k-space where more energy is located. For higher spatial frequency regions, the VD spiral trajectory uses larger gradient strengths and slew rates to cover k-space as quickly as possible. Compared to 2D sampling, designing 3D sampling analytically is more challenging for several reasons. The number of parameters increases in 3D, and the parameter selection is more difficult due to the larger search space. For example, a 3D radial trajectory with 10000 spokes has 20000 degrees of freedom, while its 2D multi-slice counterpart with 200 spokes per slice has only 200 degrees of freedom. Additionally, analytical designs usually are based on the Shannon-Nyquist relationship [54, 77, 173] that might not fully consider properties of sensitivity maps and non-linear reconstruction methods. For 3D sampling patterns with high undersampling (acceleration) ratios, there are limited analytical tools for designing sampling patterns with an anisotropic FOV and resolution. The peripheral nerve stimulation (PNS) effect [59] is also more severe in 3D imaging because of

---

[1]This chapter is based on our paper [147].

the additional spatial encoding gradient, further complicating manual designs. For these reasons, automatic designs of 3D sampling trajectories are crucial for efficient acquisition.

Many 3D sampling approaches exist. The 'stack-of-2D' strategy stacks 2D sampling patterns in the slice direction [75, 173]. This approach is easier to implement and enables slice-by-slice 2D reconstruction. Another design applies Cartesian sampling in the frequency-encoding direction and non-Cartesian sampling in the phase-encoding direction [1, 14]. However, these approaches do not fully explore the design space in three dimensions and may not perform as well as true 3D sampling trajectories [21].

Recently, 3D SPARKLING [21] proposes to optimize 3D sampling trajectories based on the goal of conforming to a given density while distributing samples as uniformly as possible [85]. That method demonstrated improved image quality compared to the 'stack-of-2D-SPARKLING' approach. In both 2D and 3D, the SPARKLING approach uses a pre-specified sampling density in k-space that is typically an isotropic radial function. This density function cannot readily capture distinct energy distributions of different imaging protocols. In SPARKLING, the PNS effects are not controlled explicitly, and the user may need to lower the slew rate to reduce PNS. SPARKLING optimizes the location of every sampling point, or the gradient waveform (freeform optimization), and cannot optimize parameters of existing sampling patterns.

In addition to analytical methods, learning-based methods have been investigated for designing trajectories. Since different anatomies have distinct energy distributions in the frequency domain, an algorithm may learn to optimize sampling trajectories from training datasets. Several studies have shown that different anatomies produce distinct optimized sampling patterns, and these optimized sampling trajectories can improve image quality [9, 55, 70, 74, 119, 127, 146, 154, 169]. Some methods can optimize sampling trajectories with respect to specific reconstruction algorithms to further enhance reconstruction image quality [1, 144]. Several recent studies also applied learning-based approaches to 3D non-Cartesian trajectory design. J-MoDL [1] proposes to learn sampling patterns and model-based deep learning reconstruction algorithms jointly. J-MoDL optimizes the sampling locations in the phase-encoding direction, to avoid the computation cost of non-uniform Fourier transform. PILOT/3D-FLAT [6, 154] jointly optimizes freeform 3D non-Cartesian trajectories and a reconstruction neural network with gradient-based methods. These studies use the standard auto-differentiation approach to calculate the gradient used in optimization, which can be inaccurate and lead to sub-optimal optimization results [144].

This chapter extends Ch. 3 and Ch. 4, and introduces a generalized **S**tochastic optimization framework for 3D **NO**n-Cartesian sam**P**ling trajector**Y** (**SNOPY**). The proposed method can automatically tailor given trajectories and learn k-space features from training datasets. We present several optimization objectives, including image quality, hardware constraints, PNS effect suppression and image contrast. Users can simultaneously optimize one or multiple characteristics

49
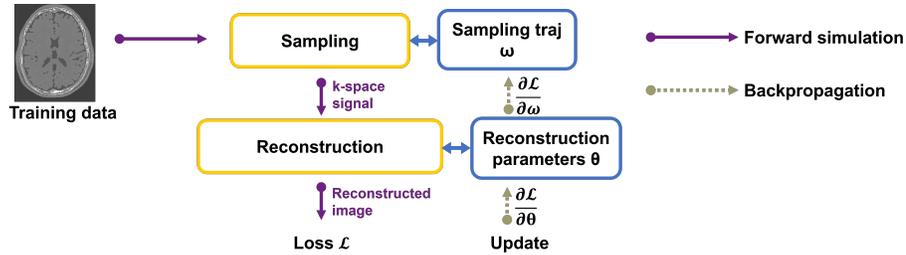
Figure 5.1: Diagram of SNOPY. The sampling trajectory (and possibly reconstruction parameters) are updated using gradient methods. The training/optimization process uses differentiable programming to obtain the gradient necessary for the update.

of a given sampling trajectory. Similar to previous learning-based methods [1, 9, 146, 154], the sampling trajectory can be jointly optimized with trainable reconstruction algorithms to improve image quality. The joint optimization approach can thus exploit the synergy between acquisition and reconstruction, and learn optimized trajectories specific for different anatomies and reconstruction methods [1, 56, 144, 146, 175]. The algorithm can optimize various properties of a sampling trajectory, such as readout waveforms, or rotation angles of readout shots, making it more practical and applicable. We also introduced several techniques to improve efficiency, enabling large-scale 3D trajectory optimization. We tested the proposed methods with multiple imaging applications, including structural and functional imaging. These applications benefited from the SNOPY-optimized sampling trajectories in both simulation and prospective studies.

5.2 Methods

This section describes the proposed gradient-based methods for trajectory optimization. We use the concept of differentiable programming to compute the descent gradient with respect to sampling trajectories required in the gradient-based methods. The sampling trajectory and reconstruction parameters are differentiable parameters, whose gradients can be computed by auto-differentiation. To learn or update these parameters, one may apply (stochastic) gradient descent algorithms. Fig. 5.1 illustrates the basic idea. The sampling trajectories can be optimized in conjunction with the parameters of learnable reconstruction algorithms so that the learned sampling trajectories and reconstruction methods are in synergy and produce high-quality images. The SNOPY algorithm combines several optimization objectives to ensure that the optimized sampling trajectories have desired properties. Ch. 5.2.1 delineates these objective functions. Ch. 5.2.3 shows that the proposed method is applicable to multiple scenarios with different parameterization strategies. For non-Cartesian sampling, the system model usually involves non-uniform fast Fourier transforms (NUFFT). Ch. 5.2.4 suggests several engineering approaches to make this large-scale optimization

problem solvable and efficient.

5.2.1 Optimization Objectives

This section outlines the optimization objectives in SNOPY. As SNOPY is a stochastic gradient descent-like algorithm, the objective function, or loss function, is by default defined on a mini-batch of data. The final loss function can be a linear combination of following loss terms to ensure the optimized trajectory possesses multiple required properties.

5.2.1.1 Image quality

For many MRI applications, efficient acquisition and reconstruction aim to produce high-quality images. Consequently, the learning objective should encourage images reconstructed from sampled k-space signals to match the reference images. We formulate this similarity objective as the following image quality training loss:

$$\mathcal{L}_{\text{recon}} = \ell(f_{\boldsymbol{\theta}, \boldsymbol{c}}(\boldsymbol{A}(\boldsymbol{\omega}(\boldsymbol{c}))\boldsymbol{x} + \boldsymbol{\varepsilon}) - \boldsymbol{x}). \tag{5.1}$$

Here, $\boldsymbol{\omega}(\boldsymbol{c}) \in \mathbb{R}^{N_{\text{fe}} \times N_{\text{s}} \times N_{\text{d}}}$ denotes the trajectory to be optimized, with N_{s} shots, N_{fe} sampling points in each shot, and N_{d} image dimensions. For 3D MRI, $N_{\text{d}} = 3$. $\boldsymbol{\varepsilon}$ is simulated complex Gaussian noise. $\boldsymbol{A}(\boldsymbol{\omega})$ is the forward system matrix for sampling trajectory $\boldsymbol{\omega}(\boldsymbol{c})$ [41]. $\boldsymbol{c}$ denotes the parameterization coefficients of sampling trajectories $\boldsymbol{\omega}$, which is introduced in Ch. 5.2.3. In this study, $\boldsymbol{A}$ also incorporated multi-coil sensitivity information [109]. $\boldsymbol{x}$ denotes the reference image from the training set $\mathcal{X}$, which is typically reconstructed from fully-sampled signals. In addition to contrast-weighted imaging, if the training dataset $\mathcal{X}$ includes *quantitative parameter maps*, one may also simulate $\boldsymbol{x}$ using the Bloch equation, and $\boldsymbol{A}$ can subsequently consider imaging physics such as relaxation. $f_{\boldsymbol{\theta}, \boldsymbol{\omega}}(\cdot)$ is the reconstruction algorithm to be delineated in Ch. 5.2.2. $\boldsymbol{\theta}$ denotes the reconstruction algorithm's parameters. It can be kernel weights in a convolutional neural network (CNN), or the regularizer coefficient in a model-based reconstruction method. The similarity term $\ell(\cdot)$ can be ℓ_1 norm, ℓ_2 norm, or a combination of both. There are also other ways to measure the distance between $\boldsymbol{x}$ and $f_{\boldsymbol{\theta}, \boldsymbol{\omega}}(\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{\varepsilon})$, such as the structural similarity index (SSIM [152]) and perceptual loss [76]. For simplicity, this work used a linear combination of ℓ_1 norm and square-of-ℓ_2 norm, which is a common practice in deep learning-based image reconstruction[172].

5.2.1.2 Hardware limits

The gradient system of MR scanners has physical constraints, namely maximum gradient strength and slew rate. Ideally, we would like to enforce a set of constraints of the form

$$\|\boldsymbol{g}_i[j,:]\|_2 \leq g_{\text{max}}, \quad \boldsymbol{g}_i = \boldsymbol{D}_1\boldsymbol{\omega}[:,i,:]/(\gamma\Delta t) \in \mathbb{R}^{(N_{\text{fe}}-1)\times N_{\text{d}}},$$

for every shot $i = 1, \ldots, N_{\text{s}}$ and time sample $j = 1, \ldots, N_{\text{fe}}$, where $\boldsymbol{g}_i$ denotes the gradient strength of the i shot and g_{max} denotes the desired gradient upper bound. One may use a Euclidean norm along the spatial axis so that any rotation of the sampling trajectory still obeys the constraint. Applying the penalty to each individual gradient axis is also feasible. A similar constraint is enforced on the Euclidean norm of the slew rate $\boldsymbol{s}_i = \boldsymbol{D}_2\boldsymbol{\omega}[:,i,:]/(\gamma\Delta t^2)$, where $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ denote first-order and second-order finite difference operators applied along the readout dimension. Δt denotes the raster time interval and γ denotes the gyromagnetic ratio.

To make the optimization more practical, we follow previous studies [146, 154], and formulate the hardware constraint as a soft penalty term:

$$\mathcal{L}_{\text{g}} = \sum_{i=1}^{N_{\text{s}}} \sum_{j=1}^{N_{\text{fe}}-1} \phi_{g_{\text{max}}}(\|\boldsymbol{g}_i[j,:]\|_2) \tag{5.2}$$

$$\mathcal{L}_{\text{s}} = \sum_{i=1}^{N_{\text{s}}} \sum_{j=1}^{N_{\text{fe}}-2} \phi_{s_{\text{max}}}(\|\boldsymbol{s}_i[j,:]\|_2). \tag{5.3}$$

Here ϕ is a penalty function, and we use a simple soft-thresholding function $\phi_\lambda(x) = \max(|x| - \lambda, 0)$, because it is sub-differentiable and easy to implement. It is possible to use more sophisticated functions. Since ϕ here is a soft penalty and the optimization results may exceed the threshold, s_{max} and g_{max} can be slightly lower than the scanner's actual physical limits to ensure that the optimization results are feasible on the scanner. Applying a sanity check before sequence programming is also useful. In addition to the soft-penalty approach, recent studies [45] also used projection-based methods.

5.2.1.3 Suppression of PNS effect

The additional gradient axis in 3D imaging can result in stronger peripheral nerve stimulation (PNS) effects compared to 2D imaging. To quantify possible PNS effects of candidate gradient waveforms, SNOPY uses a convolution model described in [123]:

$$R_{id}(t) = \frac{1}{s_{\text{min}}} \int_0^t \frac{\boldsymbol{s}_{id}(\theta)c}{(c+t-\theta)^2}d\theta, \tag{5.4}$$

where R_{id} denotes the PNS effect of the gradient waveform from the ith shot and the dth dimension. s_{id} is the slew rate of he ith shot in the dth dimension. c (chronaxie) and $s_{\min}$ (minimum stimulation slew rate) are scanner-specific parameters.

Likewise, we discretize the convolution model and use a soft penalty term as the following loss function:

$$\boldsymbol{p}_{id}[j] = \sum_{k=1}^{j} \frac{\boldsymbol{s}_{id}[k]c\Delta t}{s_{\min}(c + j\Delta t - k\Delta t)^2},$$

$$\mathcal{L}_{\mathrm{pns}} = \sum_{i=1}^{N_{\mathrm{s}}} \sum_{j=1}^{N_{\mathrm{fe}}} \phi_{p_{\max}}((\sum_{d=1}^{N_{\mathrm{d}}} \boldsymbol{p}_{id}[j]^2)^{\frac{1}{2}}). \tag{5.5}$$

Again, ϕ denotes the soft-thresholding function, with PNS threshold $p_{\max}$ (usually $\leq 80\%$[123]). This model combines the 3 spatial axes via the sum-of-squares manner and does not consider anisotropic characteristics of PNS [29]. The implementation may use an FFT (with zero padding) for efficient convolution.

5.2.1.4 Image contrast

In many applications, the optimized sampling trajectory should maintain certain parameter-weighted contrasts. For example, ideally the (gradient) echo time (TE) should be identical for each shot. Again, we replace this hard constraint with an echo time penalty. Other parameters, like repetition time (TR) and inversion time (TI), can be predetermined in the RF pulse design. Specifically, the corresponding loss function encourages the sampling trajectory to cross the k-space center at certain time points:

$$\mathcal{L}_c = \sum_{\{i,j,d\}\in C} \phi_0(|\boldsymbol{\omega}[i,j,d]|), \tag{5.6}$$

where C is a collection of gradient time points that are constrained to cross the k-space zero point. ϕ is still a soft-thresholding function, with threshold 0.

The total loss function is a linear combination of the above terms

$$\mathcal{L} = \lambda_{recon}\mathcal{L}_{\mathrm{recon}} + \lambda_g\mathcal{L}_{\mathrm{g}} + \lambda_s\mathcal{L}_{\mathrm{s}} + \lambda_{pns}\mathcal{L}_{\mathrm{pns}} + \lambda_c\mathcal{L}_{\mathrm{c}}.$$

Note that not every term is required. For example, experiment 5.3.2.2 only used the $\mathcal{L}_{\mathrm{recon}}$. Ch. 5.5 further discusses how to choose linear weights λs.

5.2.2 Reconstruction

In (5.1), the reconstruction algorithm $f_{\theta,\omega}(\cdot)$ can be various algorithms. Consider a typical cost function for regularized MR image reconstruction

$$\hat{x} = \arg\min_{x} \|A(\omega)x - y\|_2^2 + \mathsf{R}(x). \tag{5.7}$$

$\mathsf{R}(x)$ here can be a Tikhonov regularization $\nu\|x\|_2^2$ (CG-SENSE [95]), a sparsity penalty $\nu\|Tx\|_1$ (compressed sensing [91], T is a finite-difference operator), a roughness penalty $\nu\|Tx\|_2^2$ (penalized least squares, PLS), or a neural network (model-based deep learning, MoDL [2]). Ch. 5.4 shows that different reconstruction algorithms lead to distinct optimized sampling trajectories. In training, y is retrospectively simulated as $\bar{y} = A(\omega)x + \varepsilon$ (following (5.1)). In prospective studies, y is the acquired k-space data.

To get a reconstruction estimation $\hat{x}$, one may use iterative reconstruction algorithms. Specifically, the algorithm should be step-wise differentiable (or sub-differentiable) to enable differentiable programming. The backpropagation uses the chain rule to traverse every step of the iterative algorithm to calculate gradients with respect to variables such as ω.

5.2.3 Parameterization

As is shown in [146], directly optimizing every k-space sampling location (or equivalently every gradient waveform time point) may lead to sub-optimal results. Additionally, in many applications, one may need to optimize certain properties of existing sampling patterns, such as the rotation angles of a multi-shot spiral trajectory, so that the optimized trajectory can be easily integrated into existing workflows. For these needs, we propose two parameterization strategies.

The first approach, spline-based freeform optimization, represents the sampling pattern using a linear basis, i.e., $\omega = Bc$, where B is a matrix of samples of a basis such as quadratic B-spline kernels and c denotes the coefficients to be optimized [146, 154]. This approach fully exploits the generality of a gradient system. Using a linear parameterization like B-splines reduces degrees of freedom and facilitates applying hardware constraints [61, 146]. Additionally, the parameterization can be combined with multi-scale optimization to avoid sub-optimal local minima and further improve optimization results [85, 146, 154]. However, freeformly optimized trajectories could introduce implementation challenges. For example, some MRI systems can not store hundreds of different gradient waveforms.

The second approach is to optimize attributes c of existing trajectories, where $\omega(c)$ is a differentiable function of the attributes c. For example, many applications use radial trajectories, where the rotation angles can be optimized. Suppose $s \in \mathbb{R}^{3 \times N}$ is one radial sampling spoke, and

Table 5.1: The memory/time use reduction brought by proposed techniques. Here we used a 2D 400×400 test case, and CG-SENSE reconstruction (20 iterations). '+' means adding the technique to previous columns.

Plain	+Efficient Jacobian	+In-place ops	+Toeplitz embed.	+Low-res NUFFT
5.7GB / 10.4s	272MB / 1.9s	253MB / 1.6s	268MB / 0.4s	136MB / 0.2s

consider an M-shot 3D radial trajectory,

$$\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{R}_1 & \cdots & \boldsymbol{R}_M \end{bmatrix} \boldsymbol{I}_M \otimes \boldsymbol{s}, \tag{5.8}$$

where $\boldsymbol{R}_i \in \mathbb{R}^{3 \times 3}$ denotes a rotation matrix, $\boldsymbol{I}_M$ denotes an identity matrix of size M, and $\otimes$ denotes the Kronecker product. In this case, the list of $\boldsymbol{R}_i$ is the coefficient to be optimized. This approach is easier to implement on scanners, and can work with existing workflows.

5.2.4 Efficient Optimization

5.2.4.1 Optimizer

Generally, to optimize the sampling trajectory $\boldsymbol{\omega}$ and other parameters (such as reconstruction parameters $\boldsymbol{\theta}$) via stochastic gradient descent-like methods, each update takes a step (in the simplest form)

$$\boldsymbol{\theta}^K = \boldsymbol{\theta}^{K-1} - \eta_t \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}(\boldsymbol{\omega}^{K-1}, \boldsymbol{\theta}^{K-1})$$
$$\boldsymbol{\omega}^K = \boldsymbol{\omega}^{K-1} - \eta_\omega \frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}}(\boldsymbol{\omega}^{K-1}, \boldsymbol{\theta}^{K-1}),$$

where $\mathcal{L}$ is the loss function described in Section 5.2.1 and where η_t and η_ω denote step-size parameters.

The optimization is highly non-convex and may suffer from sub-optimal local minima. We used stochastic gradient Langevin dynamics (SGLD) [156] as the optimizer to improve results and accelerate training. Each update of SGLD injects Gaussian noise into the gradient to introduce randomness

$$\boldsymbol{\theta}^K = \boldsymbol{\theta}^{K-1} - \eta_t \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}^{K-1}} + \sqrt{2\eta_\theta}\mathcal{N}(0, 1)$$
$$\boldsymbol{\omega}^K = \boldsymbol{\omega}^{K-1} - \eta_\omega \frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}^{K-1}} + \sqrt{2\eta_\omega}\mathcal{N}(0, 1). \tag{5.9}$$

Across most experiments, we observed that SGLD led to improved results and faster conver-

gence compared with SGD or Adam [79]. Fig. 5.2 shows a loss curve of SGLD and Adam of experiment 5.3.2.3.
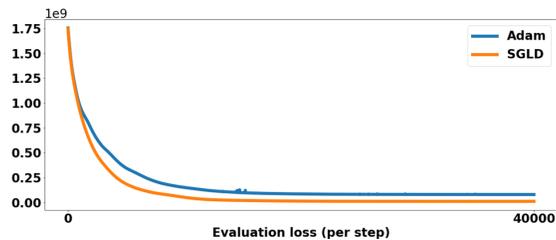
Figure 5.2: The evaluation loss curve for SGLD and Adam. The training process costs ∼1 hr.
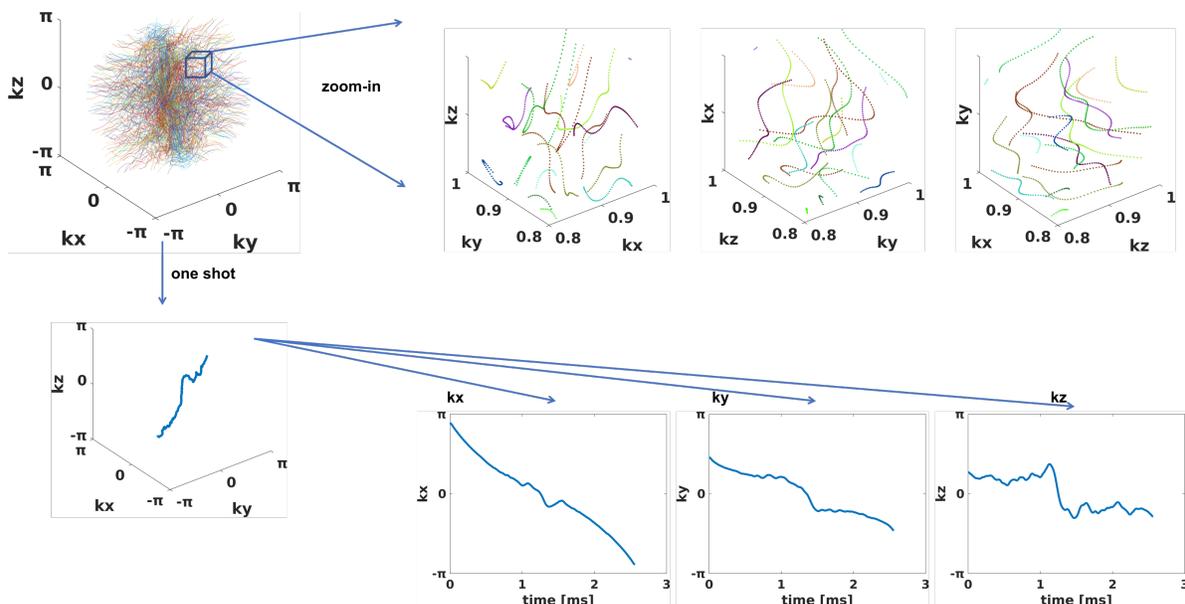
Figure 5.3: The optimized sampling trajectory of experiment 5.3.2.1. The training process involves the SKM-TEA dataset and CG-SENSE reconstruction. The upper row shows a zoomed-in region from different viewing perspectives. The lower row displays one shot from different perspectives.

5.2.4.2   Memory saving techniques

Due to the large dimension, the memory cost for naive 3D trajectory optimization would be prohibitively intensive. We developed several techniques to reduce memory use and accelerate training.

In the similarity loss (5.1), the sampling trajectory is embedded in the forward system matrix A. The system matrix for non-Cartesian sampling usually includes NUFFT operators [41]. Updating the sampling trajectory in each optimization step requires the Jacobian, or the gradient with respect
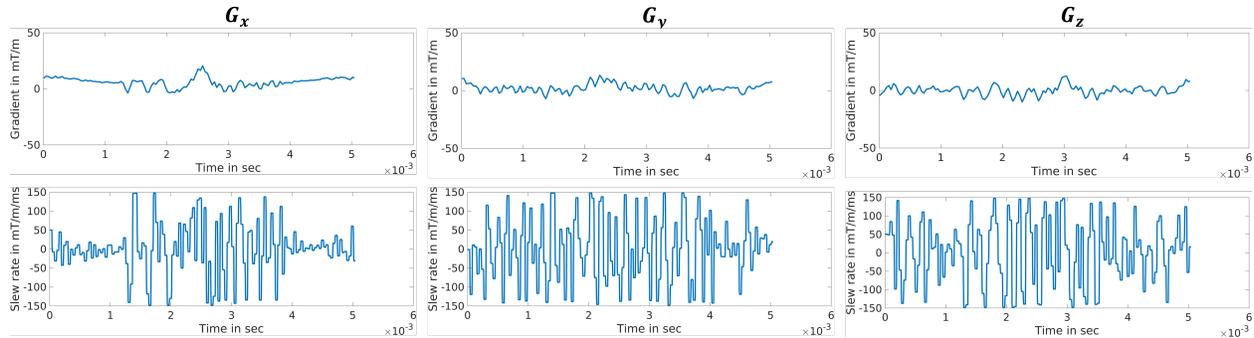
Figure 5.4: Profile of gradient strength and slew rate for one shot in experiment 5.3.2.1.

to the sampling trajectory. The NUFFT operator typically involves interpolation in the frequency domain, which is non-differentiable due to rounding operations. Several previous works used auto-differentiation (with sub-gradients) to calculate an approximate numerical gradient [154, 6], but that approach is inaccurate and slow [144]. As discussed in Ch. 3, we derived an efficient and accurate Jacobian approximation method. Calculating this Jacobian simply uses another NUFFT, which is more efficient than the auto-differentiation approach. We also used in-place operations in certain reconstruction steps, such as the conjugate gradient (CG) method, because with careful design it will not interrupt auto-differentiation. (See our open-source code[2] for details.) The primary memory bottleneck relates to 3D NUFFT operators. One can pre-calculate the Toeplitz embedding kernel to save memory and accelerate computation [40, 101]. In the training phase, we used NUFFTs with lower accuracy, for instance, with a smaller oversampling ratio for gridding. Table 5.1 shows the incrementally improved efficiency achieved with these techniques. Without the proposed techniques, optimizing 3D trajectories would require hundreds of gigabytes of memory, which would be impractical for a single node. SNOPY enables solving this otherwise prohibitively large problem on a single GPU.

5.3 Experiments

5.3.1 Datasets

We used two publicly available datasets; both of them contain 3D multi-coil raw k-space data. SKM-TEA [30] is a 3D quantitative double-echo steady-state (qDESS [157]) knee dataset. It was acquired by 3T GE MR750 scanners and 15/16-channel receiver coils. SKM-TEA includes 155 subjects. We used 132 for training, 10 for validation, and 13 for the test. Calgary brain dataset [133] is a 3D brain T1w MP-RAGE [18] k-space dataset. It includes 67 available subjects, acquired by

---

[2]https://github.com/guanhuaw/Bjork
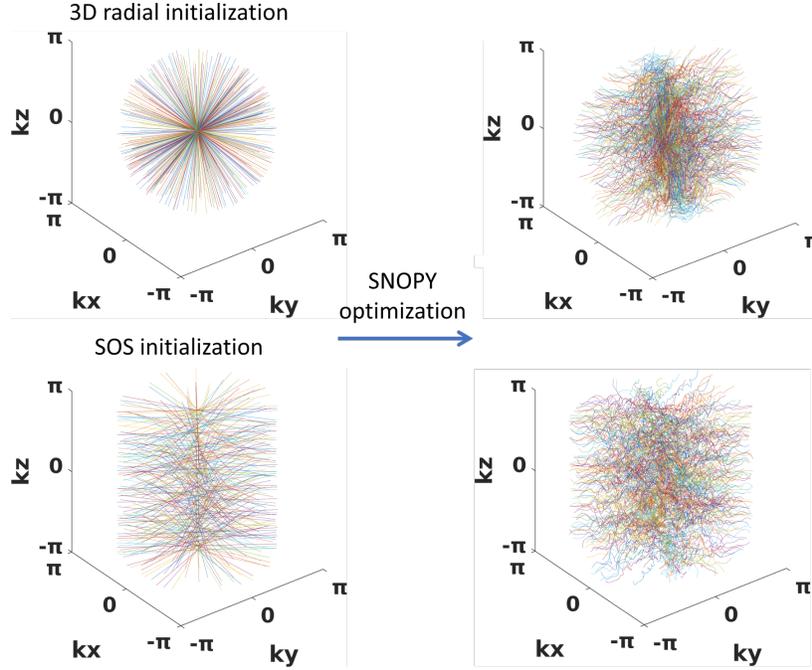
Figure 5.5: Experiment 5.3.2.1 with two types of initialization.

Table 5.2: The quantitative reconstruction quality (NRMSE) of the test set.

	CG-SENSE	PLS	MoDL
3D kooshball	28.1 dB	28.1 dB	30.0 dB
SNOPY	32.3 dB	32.4 dB	33.6 dB

an MR750 scanner and 12-channel head coils. We used 50 volumes for training, 6 for validation, and 7 for testing. All sensitivity maps were calculated by ESPIRiT [141].

5.3.2 Simulation Experiments

We experimented with multiple scenarios to show the broad applicability of the proposed method. All the experiments used a node equipped with an Nvidia Tesla A40 GPU for training.

5.3.2.1 Optimizing 3D gradient waveform

We optimized the sampling trajectory with a 3D radial ('kooshball') initialization [10, 63]. As is described in 5.2.3, the experiment optimized the readout waveform of each shot with B-spline parameterization, to reduce the number of degrees of freedom and enable multi-scale optimization. The initial 3D radial trajectory had a 5.12 ms long readout (raster time = 4 μs) and 1024 shots (8$\times$
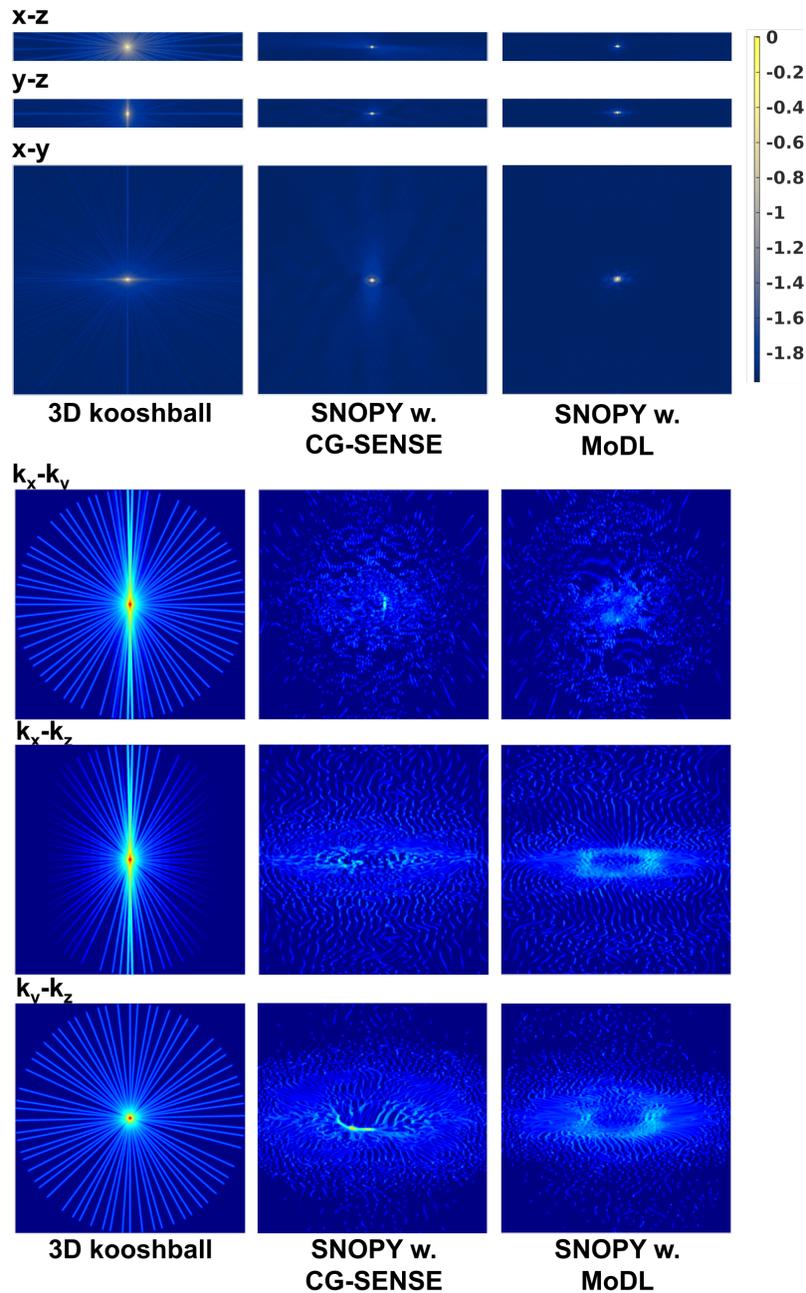
Figure 5.6: Visualization of the optimized trajectory in experiment 5.3.2.1. The upper subfigure displays PSFs (log-scaled, single-coil) of trajectories optimized with different reconstruction methods. The lower subfigure shows the density of sampling trajectories, obtained by convolving the sampling points with a Gaussian kernel. Three rows are central profiles from three perspectives.

PSNR: 25.3dB	PSNR: 26.8dB	PSNR: 28.8dB	PSNR: 30.1dB	
PSNR: 26.9dB	PSNR: 28.8dB	PSNR: 30.2dB	PSNR: 31.6dB	
3D kooshball + CG-SENSE	3D kooshball + MoDL	SNOPY w. CG-SENSE	SNOPY w. MoDL	Reference
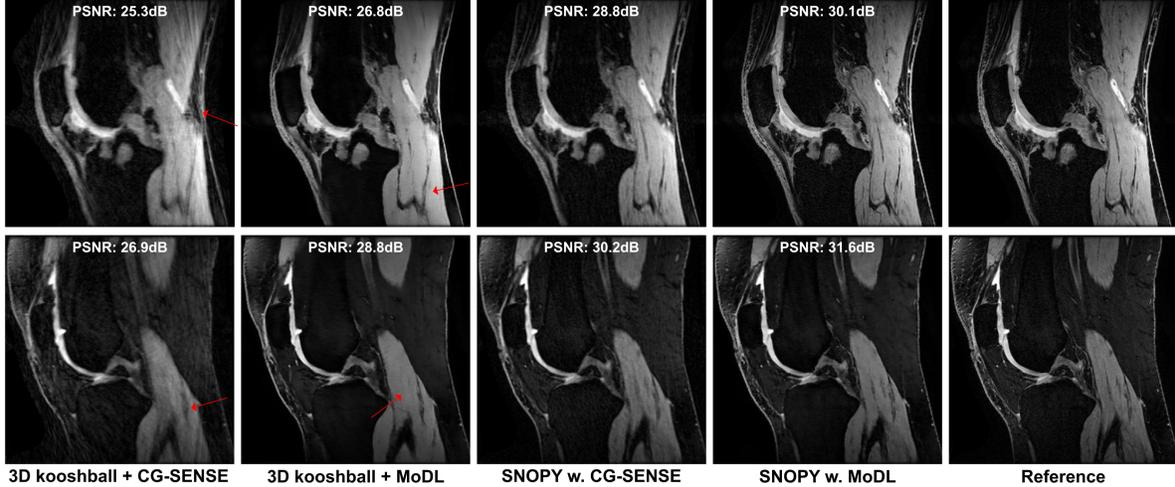
Figure 5.7: Examples of the reconstructed images for two knee slices in experiment 5.3.2.1.

acceleration), using the rotation angle described in [21]. The training used the SKM-TEA dataset. The retrospectively cropped FOV was $158 \times 158 \times 51$ mm^3 with $0.76 \times 0.62 \times 1.6$ mm^3 simulated resolution. The receiver bandwidth was ± 125 kHz (dwell time = 4 μs). The training loss was

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + 10^3 \mathcal{L}_{\text{g}} + 10^3 \mathcal{L}_{\text{s}} + \mathcal{L}_{\text{pns}}.$$

The gradient strength ($g_{\max}$) and slew rate ($s_{\max}$) were 50 mT/m and 150 mT/m/ms (for individual axis). The PNS threshold ($p_{\max}$) was 80%. The simulated noise ε was 0. The batch size was 3. The learning rate η_{ω} decayed from 10^{-4} to 0 linearly. For multi-level optimization, we used 3 levels (with B-spline kernel widths = 32, 16, and 8), and each level used 200 epochs. The total training time was $\sim$240 hrs. The trajectory was optimized with respect to several image reconstruction algorithms. We used a regularizer weight $\nu = 10^{-3}$ and 30 CG iterations for CG-SENSE and PLS. For learning-based reconstruction, we used the MoDL [2] network that alternates between a neural network-based denoiser and data consistency updates. We used a 3D version of the denoising network [166], 20 CG iterations for the data consistency update, and 6 outer iterations. Similar to previous investigations [1, 146], SNOPY jointly optimized the neural network's parameters and the sampling trajectory using (5.9).

5.3.2.2 Optimizing rotation angles of stack-of-stars trajectory

This experiment optimized the rotation angles of a stack-of-stars trajectory, which is a widely used volumetric imaging sequence. The training used the Calgary brain dataset. We used PLS as the reconstruction method for simplicity, with $\nu = 10^{-3}$ and 30 iterations. The simulated noise ε was 0 and the batch size was 12. We used 200 epochs and a learning rate linearly decaying from 10^{-4}
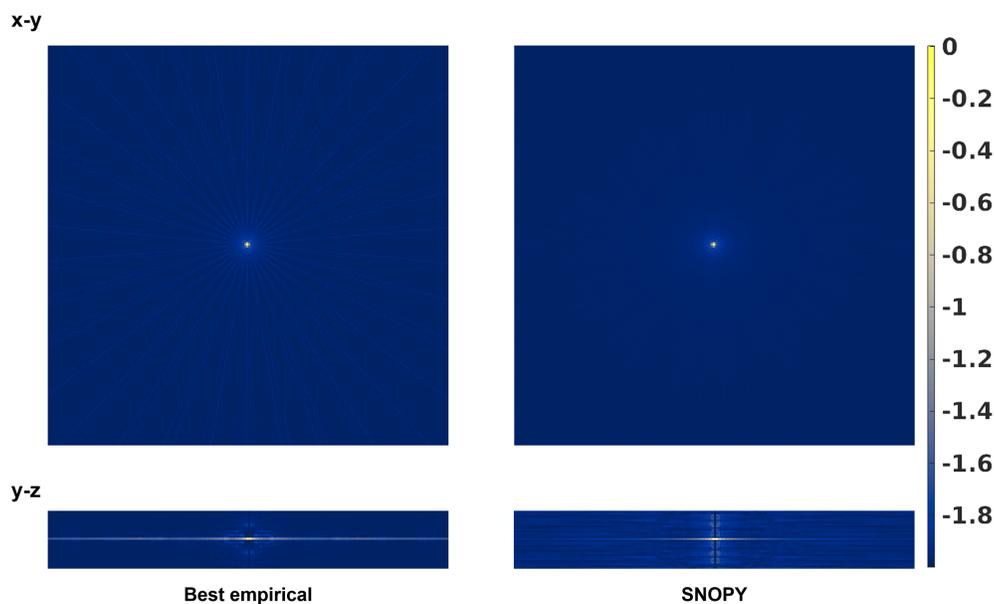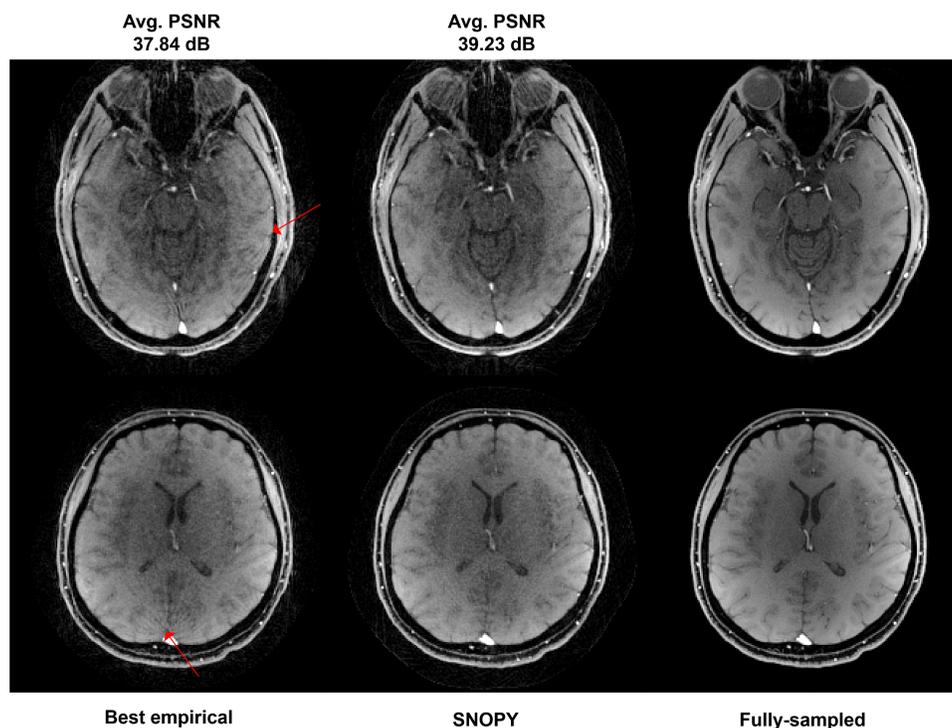
Figure 5.8: Prospective results of 5.3.2.2, optimizing the rotation angles of the stack-of-stars ($6\times$ acceleration). 'Best empirical' uses the design from a previous study [173]. The upper subfigure shows two slices from prospective in-vivo experiments. The reconstruction algorithm was PLS. Avg. PSNR denotes the average PSNR of the 4 subjects compared to the fully sampled reference. The lower subfigure shows the log-scaled PSF (single-coil) of two trajectories.

GSOS-GR

SNOPY-optimized
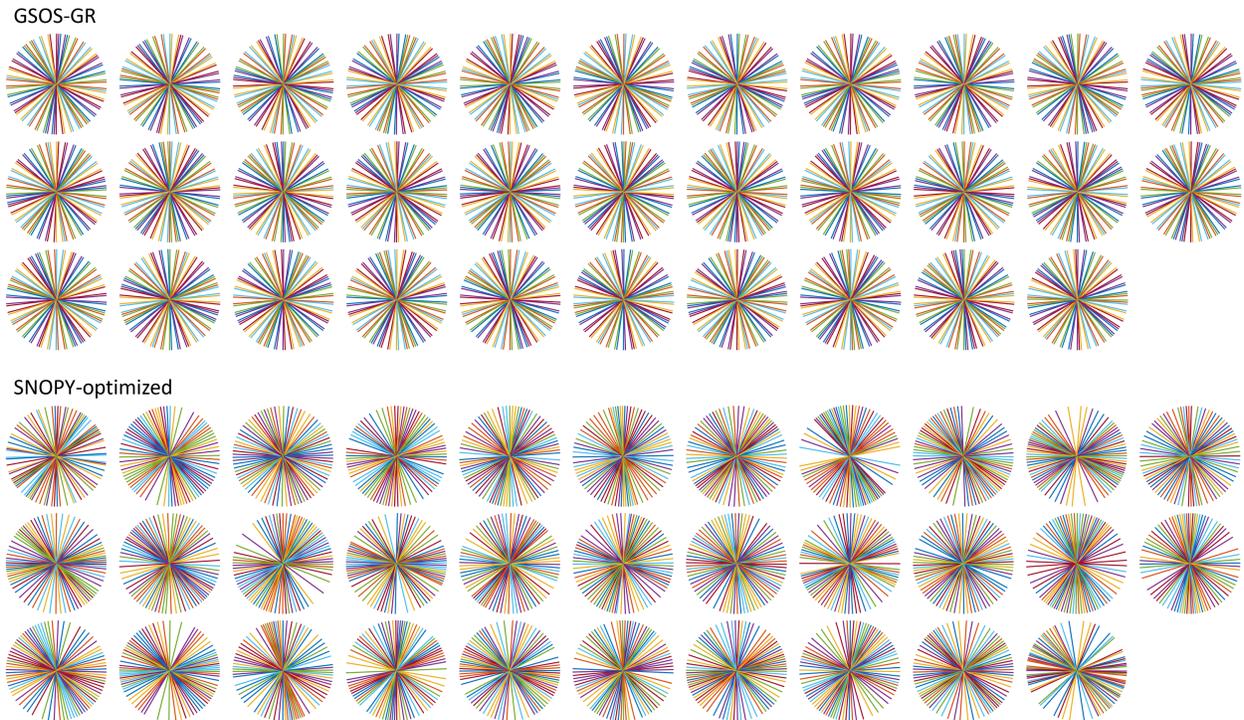
Figure 5.9: Sampling trajectories in experiment 5.3.2.2. Each figure shows an inplane $(kx - ky)$ sampling trajectory for a kz location.

to 0. The FOV was retrospectively cropped to $256{\times}218{\times}32$ mm^3 with 1 mm^3 resolution. We used 40 spokes per kz location (6$\times$ acceleration), and 1280 spokes in total. The readout length was 3.5 ms. The receiver bandwidth was $\pm$125kHz (dwell time = 4 μs). The trajectory was a stack of 32 kz planes, hence SNOPY optimized 1280 rotation angles in this case.

Since optimizing rotation angles does not impact the gradient strength, slew rate, PNS, and image contrast, we only used the reconstruction loss $\mathcal{L} = \mathcal{L}_{\text{recon}}$. We regarded the method (RSOS-GR) proposed in previous works [173] as the best empirical scheme. We applied 200 epochs with a linearly decaying learning rate from 10^{-3} to 0. The training cost $\sim$20 hrs.

5.3.2.3 PNS suppression of 3D rotational EPI trajectory for functional imaging

The third application optimizes the rotation EPI (REPI) trajectory [115], which provides an efficient sampling strategy for fMRI. For high resolution (i.e., $\leq$1 mm), we found that subjects may experience strong PNS effects introduced by REPI. This experiment aimed to reduce the PNS effect of REPI while preserving the original image contrast. We optimized one shot of REPI, being parameterized by B-spline kernels (width=16). The optimized readout shot was rotated using the angle scheme similar to [115] for multi-shot acquisition.

We designed the REPI readout for an oscillating stead steady imaging (OSSI) sequence, a novel
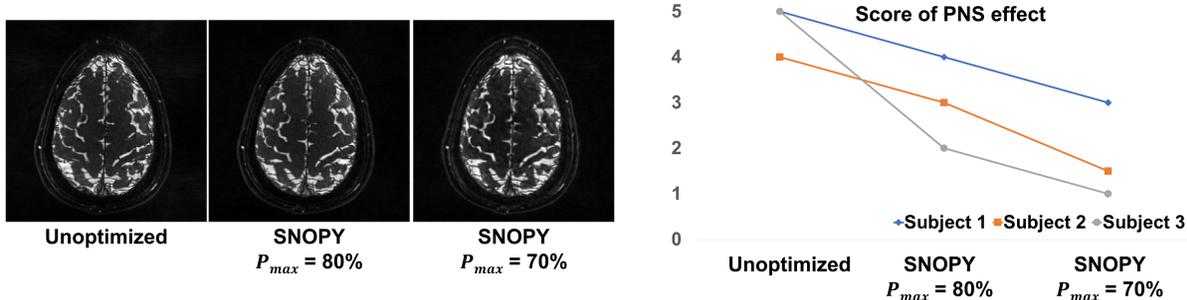
Figure 5.10: Prospective results of experiment 5.3.2.3. We showed three different trajectories: the unoptimized REPI, as well as SNOPY-optimized with PNS thresholds of 80% and 70%. The left subfigure shows one slice of reconstructed images. The reconstruction used PLS and 120 shots (volume TR = 2s). The right subfigure shows subjective scores of the PNS effect.

fMRI signal model that can improve the SNR [53, 52]. The FOV was $200 \times 200 \times 12$ mm^3, with 1 mm^3 isotropic resolution, TR = 16 ms, and TE = 7.4 ms. The readout length was 10.6 ms. The receiver bandwidth was $\pm$ 250kHz (dwell time = 2 μs). The gradient strength ($g_{\max}$), and slew rate ($s_{\max}$) constraints were 58 mT/m and 200 mT/m/ms (3 axes combined).

To accelerate training, the loss term here excluded the reconstruction loss $\mathcal{L}_{\mathrm{recon}}$

$$\mathcal{L} = 10^{-2}\mathcal{L}_g + 10^{-2}\mathcal{L}_s + \mathcal{L}_{pns} + 10^2\mathcal{L}_c.$$

The training used 40,000 steps, with the learning rate decaying linearly from 10^{-4} to 0. The training cost $\sim$1 hrs.

5.3.3   In-vivo Experiments

We implemented the optimized trajectory prospectively on a GE UHP 3.0T scanner equipped with a Nova Medical 32-channel head coil. Participants gave informed consent under local IRB approval. Since the cache space in this MR system cannot load hundreds of distinct gradient waveforms, the experiment 5.3.2.1 was not implemented prospectively. Readers may refer to the corresponding 2D prospective studies [146] for image quality improvement and correction of eddy current effects. For experiment 5.3.2.2, we programmed the sampling trajectory with a 3D T1w fat-saturated GRE sequence [103], with TR/TE = 14/3.2 ms and FA = 20°. The experiment included 4 healthy subjects. For experiment 5.3.2.3, to rate the PNS effect, we asked 3 participants to score the nerve stimulation with a 5-point Likert scale from 'mild tingling' to 'strong muscular twitch.'

5.3.4 Reproducible Research

The code is publicly available[3]. As an accompanying project, MIRTorch[4] facilitates applying differentiable programming to MRI sampling and reconstruction.

5.4 Results

For the spline-based freeform optimization experiment delineated in 5.3.2.1, Fig. 5.3 presents an example of the optimized trajectory, along with zoomed-in regions and plots of a single shot. Similar to the 2D case [146] and SPARKLING [21, 85], the multi-level B-spline optimization generates a swirling trajectory that can cover more k-space in the fixed readout time, to reduce large gaps between sampling locations and, consequently, aliasing artifacts. Notably, the zoomed-in region highlights that different shots were automatically learned not to overlap with each other, which implicitly improved the sampling efficiency [85]. Fig. 5.6 displays point spread functions (PSFs) of trajectories jointly optimized with different reconstruction algorithms. To visualize the sampling density in different regions of k-space, we convolved the trajectory with a Gaussian kernel, and Fig. 5.6 shows the density of central profiles from different views. Compared with 3D kooshball, the SNOPY optimization led to fewer radial patterns in PSFs, corresponding to fewer streak artifacts in Fig. 5.7. Trajectories optimized with different reconstruction algorithms generated different PSFs and densities, which agrees with previous studies [56, 144, 175]. Table 5.2 lists the quantitative reconstruction quality of different trajectories. The image quality metric is the average peak signal-to-noise ratio (PSNR) of the test set. SNOPY led to ∼4 dB higher PSNR than the kooshball initialization. Fig. 5.7 includes examples of reconstructed images. Compared to kooshball, SNOPY's reconstructed images have reduced artifacts and blurring. Though MoDL (and its variants) are well-performing NN-based reconstruction algorithms according to the open fastMRI reconstruction challenge [100], many important structures are distorted using the kooshball trajectory. Using the SNOPY-optimized trajectory, a simple model-based reconstruction (CG-SENSE) can reconstruct such structures. The gradient strength and the slew rate of optimized sampling trajectories are exhibited in Fig. 5.4. SNOPY solves a non-convex problem; therefore, its results depend on the initialization. We compared different initialization trajectories, including stack-of-stars (SOS) and 3D radial, as illustrated in Fig. 5.5. The number of readout points, as well as training configurations, were kept constant for both initialization methods. The average PSNR on the test set was 32.4 dB for the optimized trajectory using 3D radial initialization, and 34.8 dB for the optimized trajectory using SOS initialization.

---

[3]https://github.com/guanhuaw/SNOPY
[4]https://github.com/guanhuaw/MIRTorch
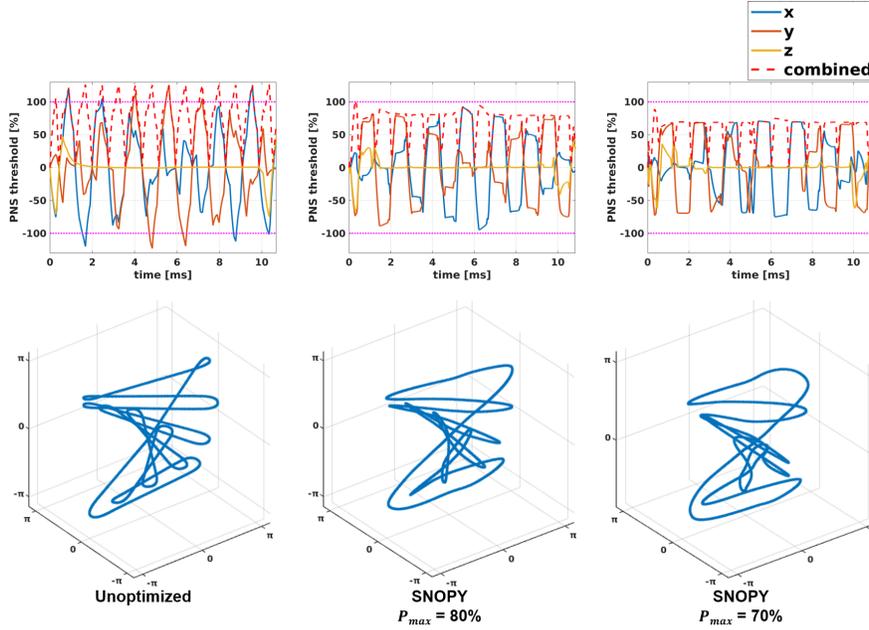
Figure 5.11: The first row of plots displays the PNS effect calculated by the convolution model (5.5) used in experiment 5.3.2.3. The second row shows the corresponding readout trajectories before and after SNOPY optimization.

For experiment 5.3.2.2, Fig. 5.8 shows the PSF of the optimized and RSOS-GR schemes [173]. For the in-plane (x-y) PSF, the SNOPY rotation shows noticeably reduced streak-like patterns. In the y-z direction, SNOPY optimization leads to a narrower central lobe and suppressed aliasing artifacts. The prospective in-vivo experiments also support this theoretical finding. In Fig. 5.8, the example slices (reconstructed by PLS) from prospective studies show that SNOPY reduced streaking artifacts. The average PSNR of SNOPY and RSOS-GR for the 4 participants were 39.23 dB and 37.84 dB, respectively. Fig. 5.9 show the rotation angles before and after SNOPY optimization.

In experiment 5.3.2.3, we tested three settings: unoptimized REPI, optimized with PNS threshold ($p_{\max}$ in (5.5)) = 80%, and optimized with $p_{\max}$ = 70%. Fig. 5.10 shows one slice of reconstructed images by the CS-SENSE algorithm, as well as the subjective ratings of PNS. Though SNOPY suppressed the PNS effect, the image contrast was well preserved by the image contrast regularizer (5.6). Fig. 5.11 presents one shot before and after the optimization, and one plot of simulated PNS effects. The SNOPY optimization effectively reduced subjective PNS effects of given REPI readout gradients in both simulation and in-vivo experiments. Intuitively, SNOPY smoothed the trajectory to avoid a constantly high slew rate, preventing a high PNS effect. To show the effect of penalty weights, we also tested 4 different settings of experiment Ch. 5.3.2.3 to showcase the

impact of different weight combinations, including

$$\mathcal{L} = 10^{-2}\mathcal{L}_g + 10^{-2}\mathcal{L}_s + 10^{-3}\mathcal{L}_{pns} + 10^2\mathcal{L}_c,$$
$$\mathcal{L} = 10^{-2}\mathcal{L}_g + 10^{-2}\mathcal{L}_s + 10^{-2}\mathcal{L}_{pns} + 10^2\mathcal{L}_c,$$
$$\mathcal{L} = 10^{-2}\mathcal{L}_g + 10^{-2}\mathcal{L}_s + 10^{-1}\mathcal{L}_{pns} + 10^2\mathcal{L}_c,$$
$$\mathcal{L} = 10^{-2}\mathcal{L}_g + 10^{-2}\mathcal{L}_s + \mathcal{L}_{pns} + 10^2\mathcal{L}_c.$$

Fig. 5.12 displays the optimized sampling trajectory and the corresponding PNS calculation. A higher weight of $\mathcal{L}_{pns}$ led to better adherence to the PNS constraint.

5.5 Discussion

SNOPY presents a novel and intuitive approach to optimizing non-Cartesian sampling trajectories. Via differentiable programming, SNOPY enables the application of gradient-based and data-driven methods to trajectory design. Various applications and in-vivo experiments demonstrated the applicability and robustness of SNOPY and its 2D predecessor [146].
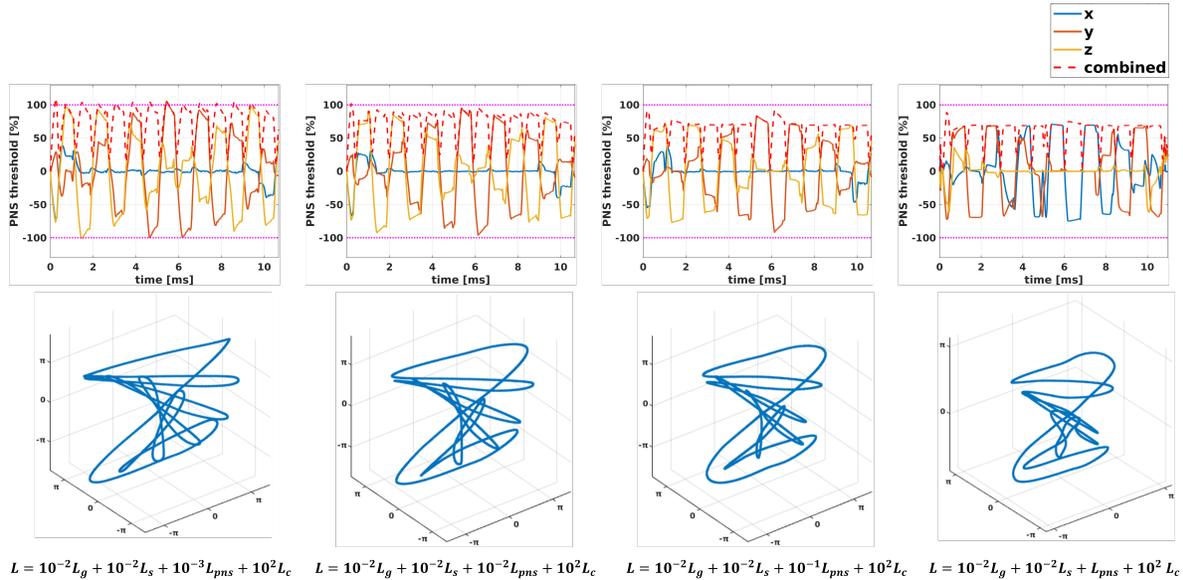
Figure 5.12: Examples of experiment 5.3.2.3 optimized by training losses with different weight combinations. The first row of figures shows the PNS effect calculated by the convolution model. The second row depicts the optimized trajectory.

Experiments 5.3.2.1 and 5.3.2.2 used SNOPY to tailor sampling trajectories according to specific training datasets and reconstruction algorithms, by formulating reconstruction image quality as a training loss. One concern was whether the learned trajectories would overfit the training

dataset. In experiment 5.3.2.2, the training set used an MP-RAGE sequence, while the prospective sequence was an RF-spoiled GRE. Similarly, 2D prospective and retrospective experiments [146] showed that trajectories learned with particular pulse sequences and hardware still improved the image quality of other sequences and hardware, and the NN-based reconstruction did not require fine-tuning with respect to prospective experiments. These empirical studies suggest that trajectory optimization is robust to moderate distribution shifts between training and inference. An intuitive explanation is that SNOPY can improve the PSF by reducing aliasing, and such improvements are universally beneficial. Future investigations will explore the robustness of SNOPY in more diverse settings, such as optimizing trajectories with healthy controls and prospectively testing them with pathological participants to examine image quality for pathologies. It will also be desirable to test SNOPY with different FOVs, resolutions, and B_0 strengths.

Our experiments demonstrated that iterative reconstruction with simple analytical regularizers, such as CG-SENSE, can benefit from the SNOPY-optimized sampling trajectories. As depicted in Fig. 5.3, CG-SENSE with SNOPY optimization can successfully reconstruct many anatomical structures that were blurred in the MoDL reconstruction without SNOPY trajectory. This result is consistent with Ch. 3, where compressed sensing algorithms with trajectory optimization also outperformed NN-based reconstruction. These findings indicate untapped potentials of model-based reconstruction by optimizing sampling trajectories.

A model mismatch may happen at the digitization level: the training set typically consists of concrete discrete-space images, whereas real objects are continuous. This inverse crime is common in learning-based methods and may lead to suboptimal results. Future research should investigate strategies for mitigating this issue.

SNOPY uses a relatively simplified model of PNS. More precise models, such as [29], may lead to improved PNS suppression results.

The training process incorporates several loss terms, including image quality, PNS suppression, hardware limits, and image contrast. By combining these terms, the optimization can lead to trajectories that have multiple desired characteristics. One may alter the optimization results by controlling the coefficients. For example, with a larger coefficient of the hardware constraint loss, the trajectory will better conform to $s_{\max}$ and $g_{\max}$. Setting the weights of several terms can be complicated. Empirically, the weight of soft constraints, including hardware ($\mathcal{L}_{\mathrm{g}}$ and $\mathcal{L}_{\mathrm{s}}$), PNS suppression ($\mathcal{L}_{\mathrm{pns}}$), and contrast ($\mathcal{L}_{\mathrm{c}}$) can be tuned to a higher value if the optimized trajectory significantly violates these constraints. Additionally, the training losses may sometimes contradict each other, and the optimization process would get stuck in a local minimum. To address this, several empirical tricks have been employed. Similar to SPARKLING [85], the constraint on maximum gradient strength can be relaxed using a higher receiver bandwidth. Bayesian optimization is another option for finding optimal loss weights, but may increase training time. Using SGLD

can introduce randomness that helps escape local minima. In spline-based optimization, one can use a larger B-spline kernel width in the early stages of a coarse-to-fine search.

Trajectory optimization is a non-convex problem. SNOPY uses several methods, including effective Jacobian approximation, parameterization, multi-level optimization, and SGLD, to alleviate the non-convexity and achieve better optimization results. These methods were also found to be effective in Ch. 4. Initialization is also important for non-convex problems, as demonstrated in Fig. 5.5. SNOPY can leverage existing knowledge of MR sampling as a benign initialization. For instance, our experiments used the widely accepted golden-angle stack-of-stars as optimization bases. The SNOPY algorithm can sequentially improve these skillfully designed trajectories to combine the best of both stochastic optimization and researchers' insights.

SNOPY has a wide range of potential applications, including dynamic and quantitative imaging, particularly if large-scale quantitative datasets are available. These new applications may require task-specific optimization objectives in addition to the ones described in Ch. 5.2.1. In particular, if the reconstruction method is not easily differentiable, such as the MR fingerprinting reconstruction based on dictionary matching [93], one needs to design a surrogate objective for image quality.

CHAPTER 6

Subject-specific Adaptive Sampling using Langevin Dynamics

6.1 Introduction

Many imaging systems acquire measurements sequentially. Reducing the number of measurements can accelerate the signal acquisition process and benefit modalities that require lower radiation, such as computed tomography (CT) and scanning electron microscopy (SEM). Nevertheless, this can result in an under-determined image reconstruction problem. To address this challenge, various reconstruction methods have been proposed, such as compressed sensing [33], to enable the recovery of an object from undersampled measurements.

Sampling strategy also plays a critical role in achieving high-quality images. For instance, many sub-Nyquist sampling patterns have been investigated in MRI, including analytical and data-driven designs [175]. However, predetermined strategies may not always be optimal for various imaging scenarios. To address this challenge, adaptive sampling or dynamic sampling techniques can select the next batch of 'important' data points based on existing observations. This approach enables better use of prior information from both signal statistics and observed signals, leading to improved image quality and acquisition speed. Relevant methods include Bayesian experimental design (BED) [58], neural network-based regression [171], and reinforcement learning [107]. These methods improved image quality in various applications. However, many neural network-based methods may lack generalization ability and explainability to out-of-distribution test sets and real-world applications.

This chapter presents a model-based dynamic sampling approach that predicts new sampling locations by greedily minimizing the variance of posterior samples drawn from the posterior distribution [32].[1] The sampler uses stochastic gradient Langevin dynamics (SGLD) [156] and supports various image priors. We applied the proposed dynamic sampling to accelerate Cartesian MRI ac-

---

[1]This chapter is based on our preprint [148].

quisition. Across many experiment settings, the proposed approach significantly improved image quality.

6.2 Theory
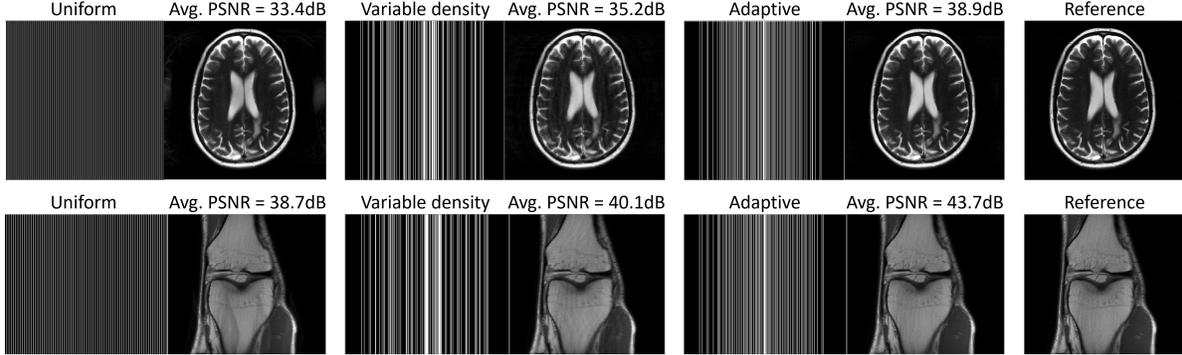
Figure 6.1: Comparison of different 1D sampling strategies with the analytical (roughness) prior. The undersampling ratio is $5\times$ for all sampling patterns. The test set has $n = 20$ slices. Dynamic sampling leads to reduced aliasing artifacts.

Consider a linear sensing system

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ denotes a sensing matrix, $\boldsymbol{x} \in \mathbb{C}^N$ denotes the object, and $\boldsymbol{y} \in \mathbb{C}^M$ denotes raw measurements. To accelerate the acquisition, we consider the 'undersampled' case where $\boldsymbol{y}$ has $L < N$ non-zero entries. Typically, the locations of non-zero entries in $\boldsymbol{y}$ follows pre-determined patterns. The proposed method, instead, dynamically chooses additional sample locations in a sequence of K sampling iterations where the samples for iteration $k+1$ are based on the measurements $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ recorded in previous iterations.

Specifically, we apply a Bayesian approach [156]. At the k iteration of additive sampling, based on the measurements acquired up until this point $\boldsymbol{y}^{(k)} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$, the first step draws samples from the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}^{(k)})$, yielding a collection of reconstructed images denoted $\{\hat{\boldsymbol{x}}_i^{(k)}\}_{i=1}^{N_{\mathrm{sample}}p}$. We use an SGLD sampler detailed below. The second step projects each estimate $\hat{\boldsymbol{x}}_i^{(k)}$ (typically in the image domain) back to the *measurement domain* using the sensing equation $\hat{\boldsymbol{y}}_i^{(k)} = \boldsymbol{A}\hat{\boldsymbol{x}}_i^{(k)}$. The third step selects the next sampling locations by greedily minimizing the variance of samples $\left\{\hat{\boldsymbol{y}}_i^{(k)}\right\}$ in the measurement domain. In detail, we select the next measurement location(s) l for the $k+1$ iteration using the k-space locations having the maximum variance:

$$l = \underset{n \in 1,2,\ldots N}{\arg\max} \mathrm{Var}\{[\hat{\boldsymbol{y}}_1^{(k)}]_n, \ldots, [\hat{\boldsymbol{y}}_{N_{\mathrm{sample}}p}^{(k)}]_n\}.$$

70

To compute a collection of reconstructions or estimates $\{\hat{x}_i\}$, we sample from the posterior

$$\hat{x} \sim p(x|y^{(k)}) = p(x)p(y^{(k)}|x)/p(y^{(k)}),$$

where $p(x)$ denotes the prior and $p(y^{(k)}|x)$ denotes the likelihood. In contrast, a typical iterative image reconstruction algorithm gives a point estimate, such as the MAP estimator. SGLD [156] samples from the posterior distribution using the update

$$\Delta x_t = \eta_t(\nabla \log p(y^{(k)}|x_t) + \nabla \log p(x_t)) + \sqrt{2\eta_t} \mathcal{N}(0, 1),$$

where η_t denotes the time-dependent step size [71, 130]. Intuitively, SGLD explores the solution space by injecting Gaussian noise similar to the Langevin Monte Carlo sampler.

Algorithm 2 Adaptive sampling algorithm

Require: Score function $f_{\theta}(x) \approx \nabla \log p(x)$; number of additive dynamic sampling iterations N_{add}; number of SGLD steps N_{step}; number of samples drawn from a posterior distribution N_{sample} p; step size in SGLD η; penalty parameter for image prior μ;
 1: Acquire initial measurements y^0
 2: (optional) Pre-train $f_{\theta}(x)$ on dataset $\mathcal{X}$ via score matching.
 3: **for** $k = 1$ to N_{add} **do**
 4:     **for** $i = 1$ to N_{sample} p **do**
 5:         **for** $t = 1$ to N_{step} **do**
 6:             Initialize $\tilde{x}_0$
 7:             $\tilde{x}_t = \tilde{x}_{t-1} + \mu_t f_{\theta}(\tilde{x}_{t-1}) - \mu_t \eta_t A'(A\tilde{x}_{t-1} - y^{(k)}) + \sqrt{2\mu_t} \mathcal{N}(0, 1)$
 8:         **end for**
 9:         $\hat{x}_i^{(k)} = \tilde{x}_{N_{\text{add}}}$
10:         $\hat{y}_i^{(k)} = A\hat{x}_i^{(k)} + \varepsilon$
11:     **end for**
12:     $l = \arg\max_{n \in 1,2,...N} \text{Var}\{[\hat{y}_1^{(k)}]_n, \ldots, [\hat{y}_{N_{\text{sample}}p}^{(k)}]_n\}.$
13:     Acquire additive measurements with index l and concatenate it with previous measurements $y^{(k)} = [y^{(k-1)}, y_l]$.
14: **end for**

In applications where the noise ε is Gaussian, the gradient of likelihood has the closed-form solution $\nabla \log p(y|x) = -A'(Ax - y)$. The prior term $p(x)$, or the score function $\nabla \log p(x)$ can take various forms. For example, a simple prior that penalizes first-order roughness has the form $p(x) = e^{-\lambda \|Tx\|_2^2/2}$, where T is the first-order finite difference transform; its corresponding score function is $\nabla \log p(x) = -\lambda T'Tx$. Analytical priors may not be informative and many studies propose to learn score functions from datasets. Score matching approximates the score function

with a learnable function $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ and learns from a training set $\mathcal{X}$:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}} \| \log p(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x}) \|_2^2$$

Recent improvements in score matching, such as sliced score matching and denoising score matching [71, 130], have extended the method's effectiveness and made it more applicable to large datasets [66, 131] To demonstrate the adaptability of our algorithm, we tested both analytical priors and score functions based on neural networks. Alg. 2 details the proposed approach.
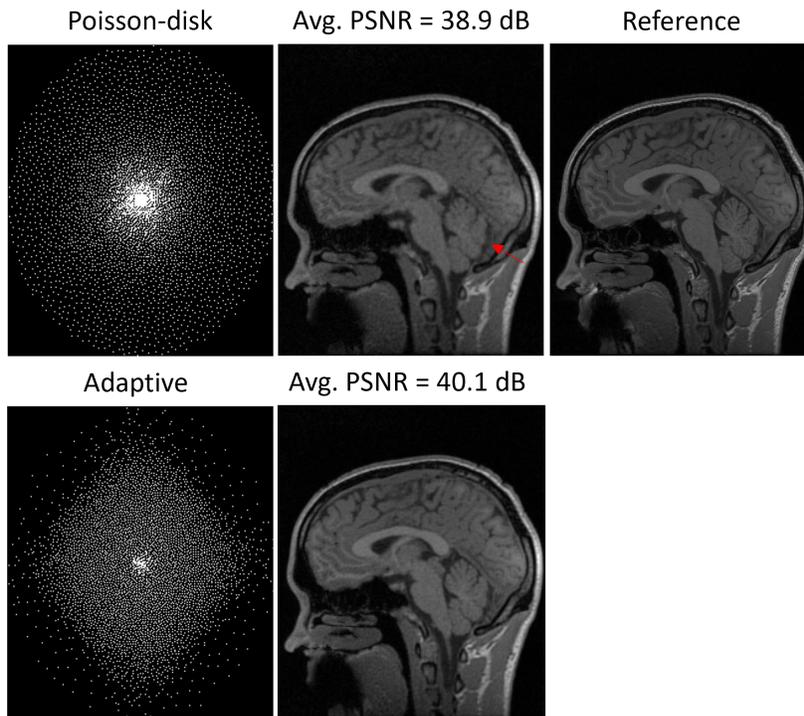
Figure 6.2: Comparison of 2D sampling strategies with reconstruction based on an analytical (roughness) prior. The undersampling ratio was $12\times$ for both sampling patterns. The test set had $n = 10$ volumes. Dynamic sampling reduced blurring and artifacts.

6.3 Experiments

We applied the proposed dynamic sampling method to MRI data that reside in the Fourier domain (k-space). For our experiment with Cartesian sampling, the sensing matrix $\boldsymbol{A}$ contained both FFT and coil sensitivity (calculated by methods described [141]). The score functions included both a simple analytical one $f(\boldsymbol{x}) = -\lambda \boldsymbol{T}'\boldsymbol{T}\boldsymbol{x}$ and a learned U-Net-based model. We evaluated the analytical priors on multiple MRI datasets [30, 133, 167], using both 1D and 2D sampling patterns.
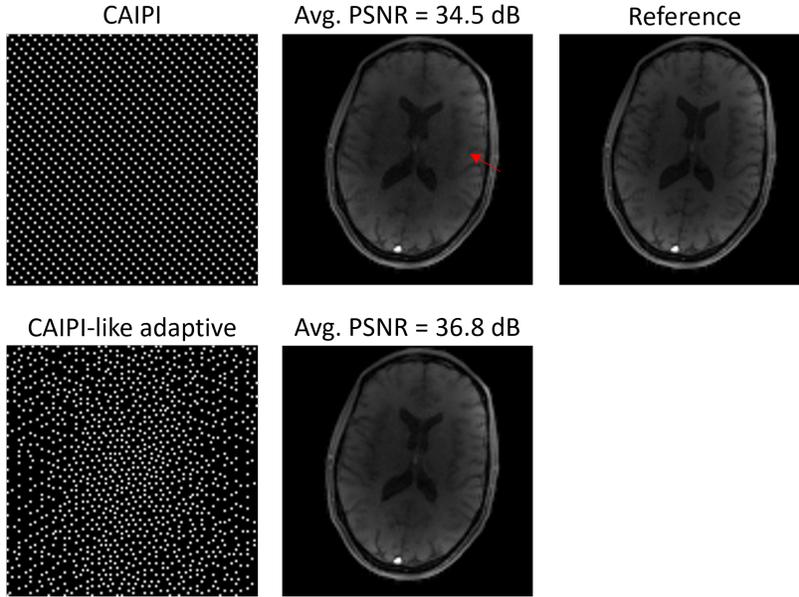
Figure 6.3: Comparison of 2D sampling strategies with the learned (NCSN++) prior. The undersampling ratio was $10\times$ for both sampling patterns. The test set had $n = 16$ slices. Adaptive sampling improved tissue contrast and reduced blurring.

We compared the dynamic sampling patterns with well-received fixed sampling patterns, such as Poisson-disk, for $N_{\mathrm{add}} = 50$ and $N_{\mathrm{step}} = 200$.

We used the same U-Net-based architecture (NCSN++) and configurations as in [132] to train the learned prior on the fastMRI brain dataset. The complex-valued image was formulated as two input channels. To demonstrate the generalization ability, we tested it on test sets that contained different anatomies and sequences than the fastMRI database, including an MP-RAGE sequence of human brains [133] and a GRE sequence of mouse brains, without any fine-tuning. For the mouse brain dynamic contrast-enhanced (DCE) data, we learned the sampling pattern from a 'pilot' frame and then applied it to subsequent frames. We used $N_{\mathrm{add}} = 30$ and $N_{\mathrm{step}} = 100$ and the accelerated sampler described in [25]. The sequence $\boldsymbol{\eta}$ used the same configuration as described in [132].

6.4   Discussion

The posterior sampling processes can be computationally expensive, determined by both the system matrix $\boldsymbol{A}$ and the score function $\nabla \log p(\boldsymbol{x})$. Simpler analytical priors may accelerate the sampling process. The sampling process is embarrassingly parallel and can leverage parallel computing and hardware improvements. In its current form, the proposed dynamic sampling is particularly useful for dynamic imaging applications such as fMRI and DCE-MRI where a 'pilot' scan is available to design tailored sampling patterns for subsequent frames and avoid the long computa-

tion time that may compromise the benefits of dynamic sampling. The sampling from the posterior
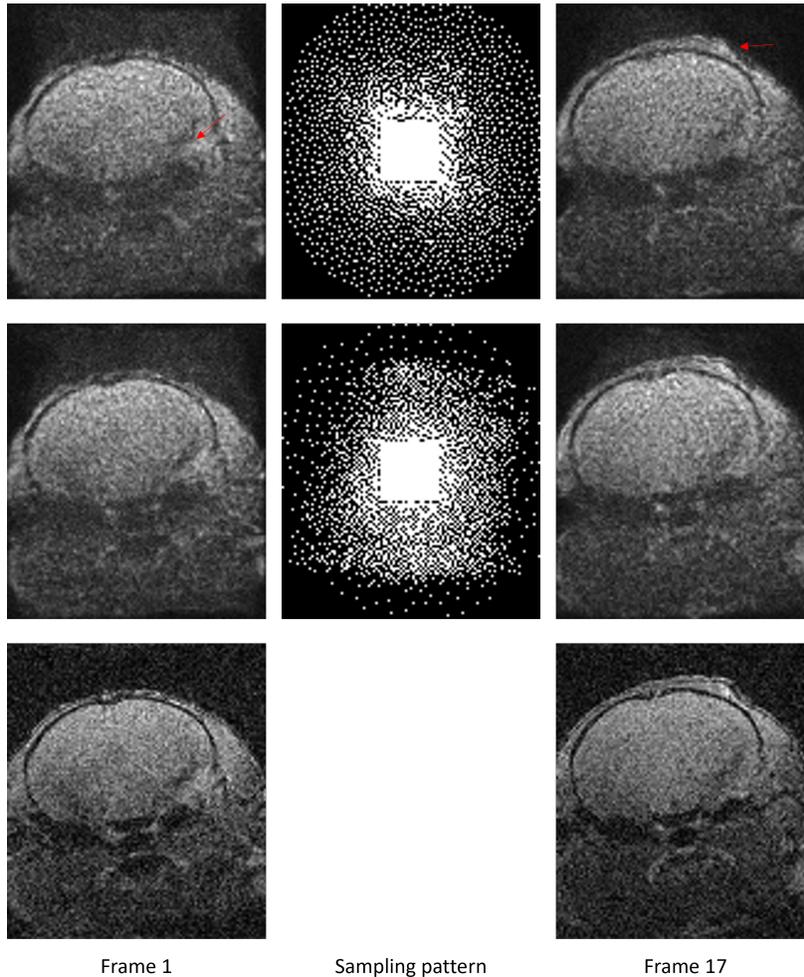
Figure 6.4: Comparison of 2D sampling strategies with the learned (NCSN++) prior. The first row shows the Poisson-disk sampling pattern. The second row displays the adaptive sampling pattern optimized with the 1st frame and applied to the 17th frame. The third row shows the reference images. The undersampling ratio was $4\times$ for both sampling patterns. Adaptive sampling led to reduced artifacts and higher SNR across different time frames.

distribution may benefit from faster samplers [160]. Some 'single-shot' samplers based on neural network methods can sample faster than SGLD [137] however, they are trained on a certain dataset and may lack the ability to generalize to out-of-distribution applications.

The proposed dynamic sampling method has demonstrated decent robustness in simulated experiments and analytical priors worked well for different test cases. The learned priors were trained on a fastMRI brain dataset but generalized well to different anatomies, vendors, sequences, and field strengths. Future work will include a systematic comparison with prior arts such as [9, 175] and prospective in-vivo experiments.

CHAPTER 7

Other Contributions

7.1   Model-based Motion Correction for fMRI

Motion correction is critical for fMRI due to the long acquisition time. Currently, registration-based retrospective correction is standard in post-processing pipelines [73]. Typically, this approach estimates the motion parameters for the whole volume at a certain time frame. For high-resolution multi-shot fMRI, the intra-frame motion between shots may impede image-based approaches and cause blurring. We propose an iterative approach for fine-grained motion correction and reconstruction. Our method iterates between motion estimation and model-based reconstruction to achieve improved motion correction compared to single-pass registration. The implementation utilizes multiple GPUs to accelerate computation.

Our proposed method alternates between motion estimation, using registration, and motion-informed model-based reconstruction to synergistically improve both. The first step reconstructs a 'crude' image for each shot. We initialize this step with a 'data-sharing' (multi-shot) reconstruction to avoid strong aliasing artifacts. Next, we estimate rigid motion using gradient correlation as the metric. Then the method updates the k-space data and corresponding system matrices of each shot, including the sampling trajectory and coil sensitivity and B_0 inhomogeneity maps, according to the motion parameters.

Our model-based reconstruction involves minimizing the cost function $\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathsf{R}(\boldsymbol{x})$, where $\boldsymbol{A}$ denotes the system matrix. Here the system matrix is a stack of system matrices of different shots

$$A = \begin{bmatrix} \boldsymbol{A}_{\text{shot1}} \\ \boldsymbol{A}_{\text{shot2}} \\ \vdots \\ \boldsymbol{A}_{\text{shotn}} \end{bmatrix},$$

to accommodate different trajectories and shifted sensitivity/B_0 maps. $\mathsf{R}(\boldsymbol{x})$ denotes the regularization terms, such as a Tikhonov regularizer.

Compared to standard reconstruction, the proposed reconstruction method requires more NUFFTs. The implementation uses the GPU-based MIRTorch and torchkbnufft [101] for efficient computation.
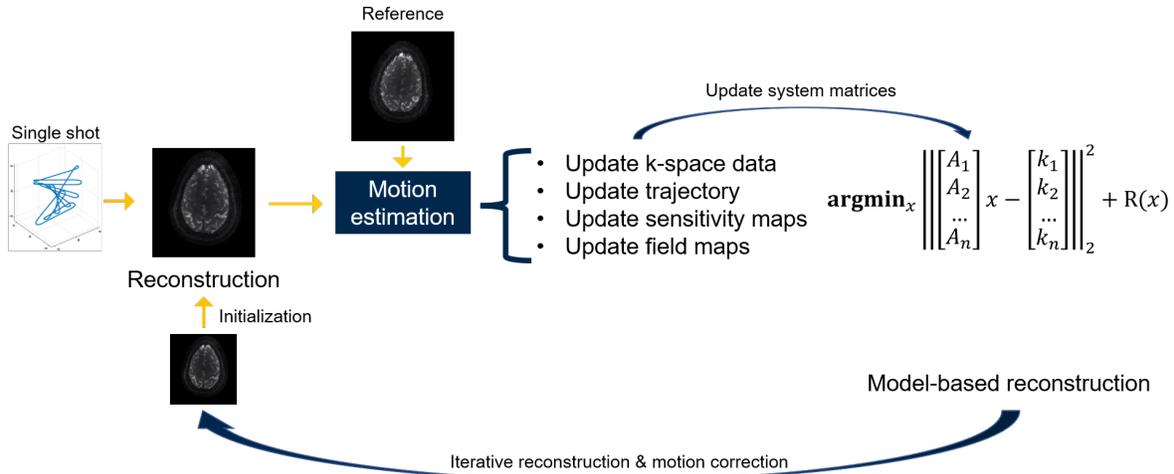
Figure 7.1: Workflow of the proposed motion correction method.
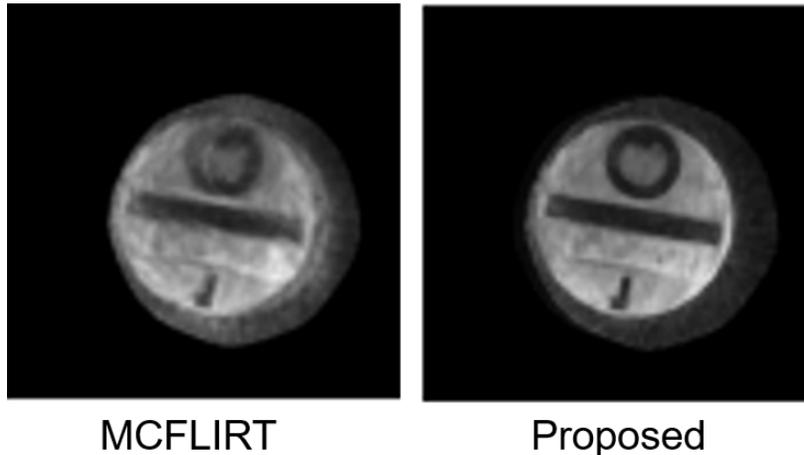
Figure 7.2: Results of the phantom experiment. In the presence of strong motion, the registration-based method (MCFLIRT) cannot resolve the blurring caused by the inter-shot movement.

We evaluated the proposed motion correction using oscillating steady state (OSS) fMRI [53], a novel high-SNR BOLD fMRI method. The sampling trajectory used the rotation EPI (REPI) [115], as shown in Fig. 7.1. The FOV was 20×20×3.2 cm, with a 2 mm isotropic resolution. The TR was 16 ms, and TE was 7.4 ms. We used 10 shots and 10 OSS 'fast-time' acquisitions to reconstruct a single time frame. In the phantom experiment, we cyclically rotated a cylinder water phantom. In the in-vivo task-based study, a volunteer performed a finger-tapping task and rotated the head

MCFLIRT
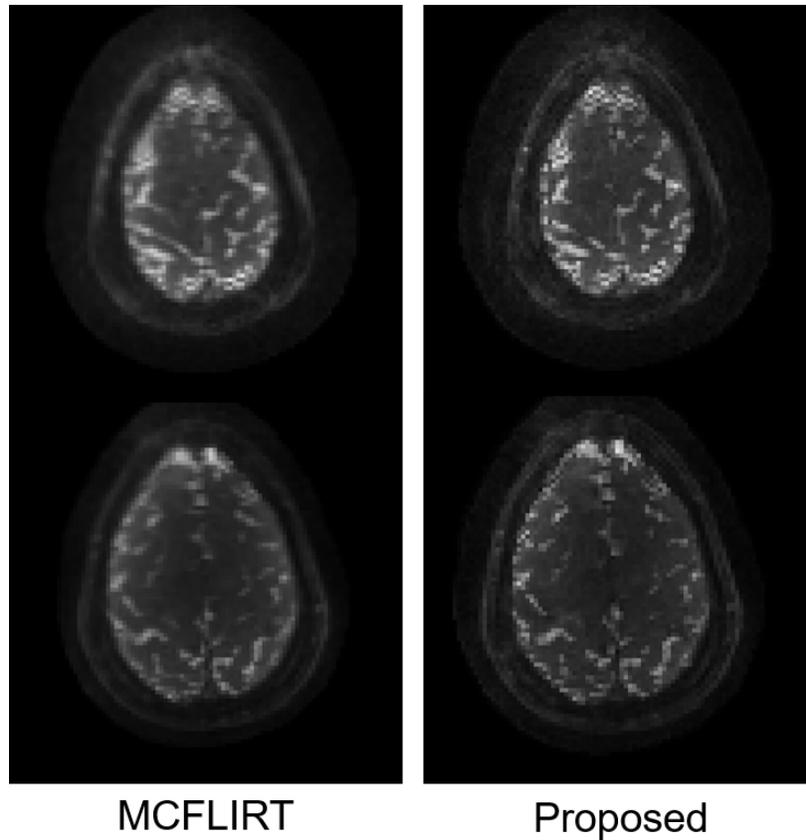
Proposed

Figure 7.3: Results of the in-vivo experiment. Two slices from the same time frame are displayed, and corrected by different methods. Similar to the phantom experiment, the proposed method effectively reduced blurring artifacts.
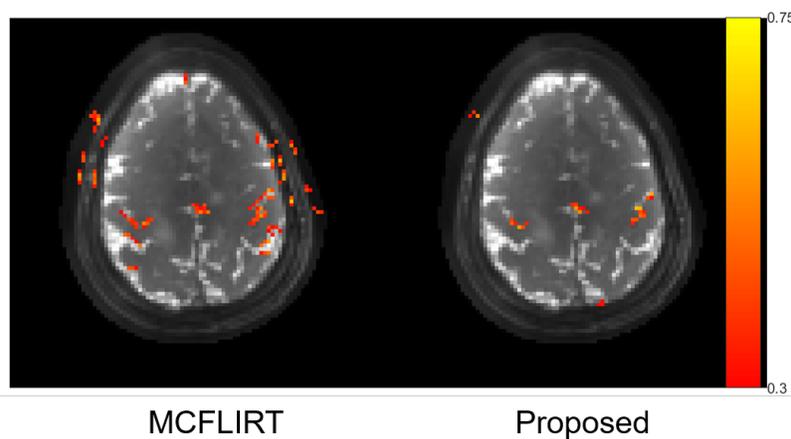
MCFLIRT

Proposed

Figure 7.4: Activation maps for the two motion correction methods. MCFLIRT led to scattered false positives due to motion, while the proposed iterative method reflected accurate correlations in the motor cortex.

to simulate motion. The baseline registration used the standard MCFLIRT function in the FSL toolbox [73]. The reconstruction algorithm was CG-SENSE reconstruction (roughness penalty, 20 iterations) and used Nvidia RTX8000 GPUs.

Fig. 7.2 depicts the results of the phantom experiment. Compared with MCFLIRT, our method effectively reduced the blurring artifacts caused by the intra-frame motion. Fig. 7.3 presents the in-vivo reconstruction results, where the proposed method reduced blurring. Fig. 7.4 shows activation maps calculated from MCFLIRT and the proposed method. MCFLIRT erroneously showed activation in the skull due to motion, whereas the proposed method accurately reflected voxels in the motor cortex. Our approach with the multi-GPU optimization significantly improves computation efficiency: the reconstruction time of using 3 GPUs was less than 1 hr, whereas an 18-core CPU calculation took more than 20 hr (for 60 temporal frames).

This section presented an application of our method to fMRI, which can also benefit other applications such as diffusion imaging and MR fingerprinting.

7.2 Model-based Acceleration for Longitudinal Imaging

Longitudinal MRI holds great potential for tracking in-vivo dynamics, such as aging, disease progression, and cognitive change [16, 81, 158]. Based on the underlying similarity between longitudinal scans, we propose Delta Scan, a model-based and explainable method that uses the previous MRI exams to accelerate future scans.[1]

The basic assumptions of modeling longitudinal imaging include: (1) exams from the same participant have similar energy distribution (in k-space), and (2) structural changes between exams should be sparse (under a certain transform). For (1), the sampling pattern is optimized based on the spectrum of the first exam. For (2), we design a model-based reconstruction algorithm to exploit prior information. The sampling pattern optimization workflow uses a Pareto optimization paradigm [175]. We propose to use stochastic optimization when extending to the non-cartesian case. The underlying assumption is that the energy distribution for the same subject should be similar across time. For image reconstruction, we model images as a combination of two components: (1) historical static information (i.e., the majority of the tissue structure remains the same), which we formulate as a block-matching process; (2) dynamic information (i.e., accounting for possible structural changes in tissue level and/or positional changes due to subject motion), which was modeled as a dictionary learning problem. The two sources of information are incorporated

---

[1]This section is based on a conference abstract [145], accomplished during the internship at Q Bio, Inc.

into the following two-stage optimization problem. The cost function of the first stage is:

$$\boldsymbol{x}_1 = \arg\min_{\boldsymbol{x}} \min_{i_j \in \{1,2...N\}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \nu \sum_{j=1}^{N_1} \left\|\mathcal{P}_j \boldsymbol{x} - \boldsymbol{D}_1^{i_j}\right\|_2^2, \tag{7.1}$$

where $\boldsymbol{A}$ denotes MRI forward model and $\mathcal{P}$ is a patch operator that decomposes the image into local patches, with N_1 patches in total. The first term is usually termed as 'data consistency.' The second term involves block-matching operations inspired by the BM3D method [27], with $\boldsymbol{D}_1$ extracted from previous scans using the same $\mathcal{P}$. To make the block-matching process more robust, the algorithm subtracts the phase and normalizes the atoms. The optimization alternates between the minimization of the data-consistency term (solved by conjugate gradient) and the block-matching operation. With the solution ($\boldsymbol{x}_1$) of the first step, the cost function of the second stage is

$$\boldsymbol{x}_2 = \left\{\arg\min_{\boldsymbol{x}} \min_{\boldsymbol{D}_2, \boldsymbol{Z}_2} \|\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{x}_1) - \boldsymbol{y}\|_2^2 + \lambda_1 \|\mathcal{P}\boldsymbol{x} - \boldsymbol{D}_2 \boldsymbol{Z}_2\|_2^2 + \lambda_2^2 \|\boldsymbol{Z}_2\|_0\right\} + \boldsymbol{x}_1 \tag{7.2}$$

The second stage assumes that the changes between exams should be sparse w.r.t. a certain transform. Instead of using fixed operations, such as wavelet transforms or discrete cosine transforms (DCT), we learn a sparsifying dictionary $\boldsymbol{D}_2$ and its sparse code $\boldsymbol{Z}_2$ adaptively [113].

Table 7.1: The average image quality of different reconstruction methods.

Methods	CS	Delta	LACS	DL
SSIM	0.964	**0.986**	0.974	0.981
PSNR	30.3dB	**32.1dB**	28.0dB	31.3dB

In simulation experiment, we compared the Delta reconstruction with previous reference-based MRI methods, namely LACS [155], which uses a weighted point-wise similarity constraint between MRI exams. We also compared the proposed method with referenceless methods, including ℓ_1-wavelet CS [91] and blind dictionary learning (DL) [113]. The cost function of the dictionary learning method is similar to (7.2) without $\boldsymbol{x}_1$.

The cohort consisted of 8 subjects. Each received two consecutive exams with at least a six months gap on a 1.5T GE scanner. Local IRB approved. Each exam acquired a 3D MP-RAGE sequence with FOVs between 24-26 cm and a matrix size of 192*192*170. The first scan used a 1D 2.5x undersampling pattern, while the second scan was fully sampled.

We compared the following experiment settings to simulate reference scans and corresponding aggressively undersampled scans:

(1) The second exam was further retrospectively undersampled, with an undersampling ratio
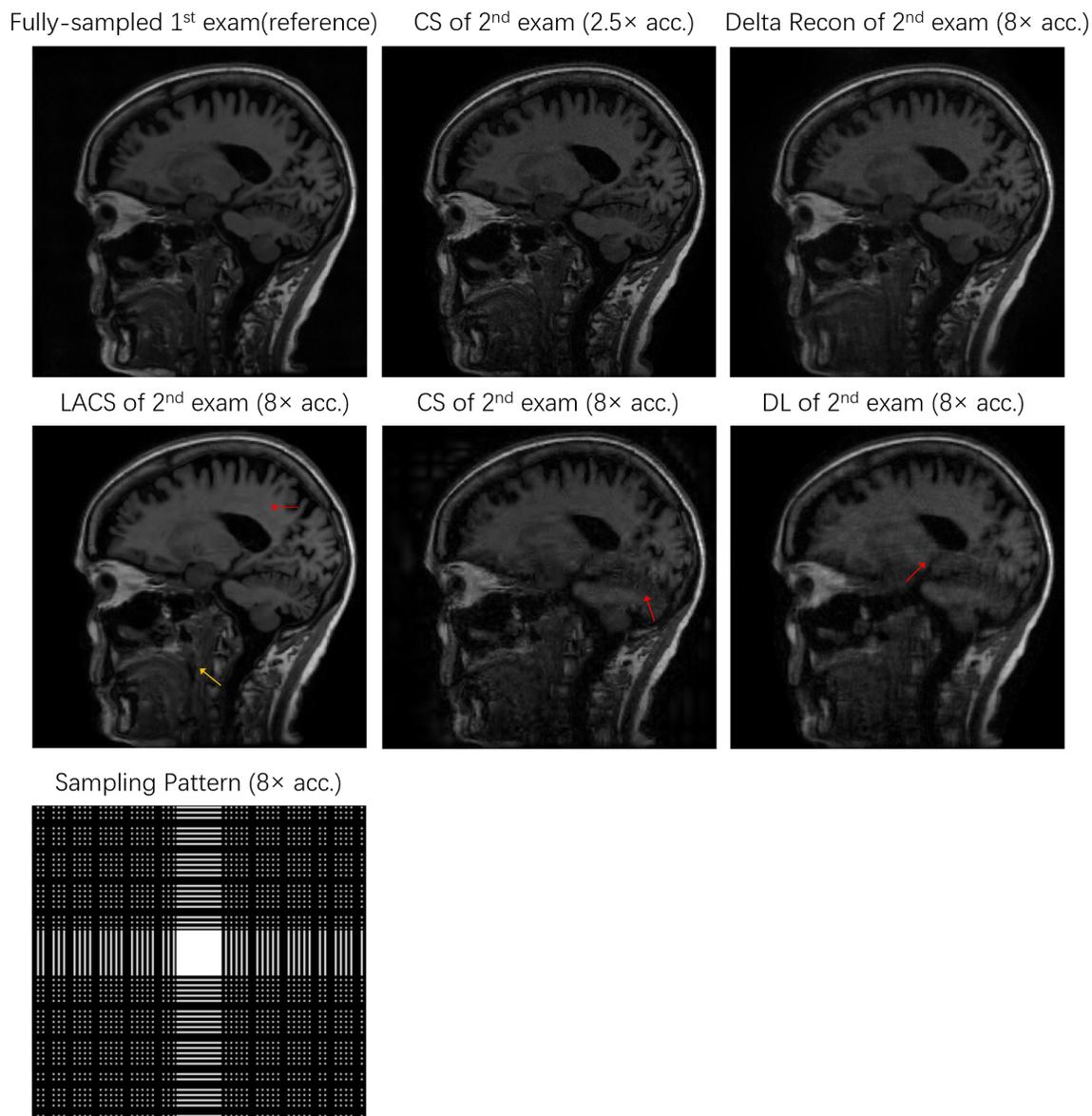
Figure 7.5: Reconstructed examples for the experiment setting (1), where the reference scan is the first fully-sampled scan. We performed a further under-sampling of the second scan to $8\times$ acceleration. Here we regard the CS-based reconstruction of $2.5\times$ accelerated second scan as the high-quality reference reconstruction. Red arrows indicate artifacts, and yellow arrows indicate the possible over-fitting on the reference scan. Compared with CS and DL, Delta Recon produces fewer artifacts. Compared with LACS, Delta Scan tracks the changes more faithfully.
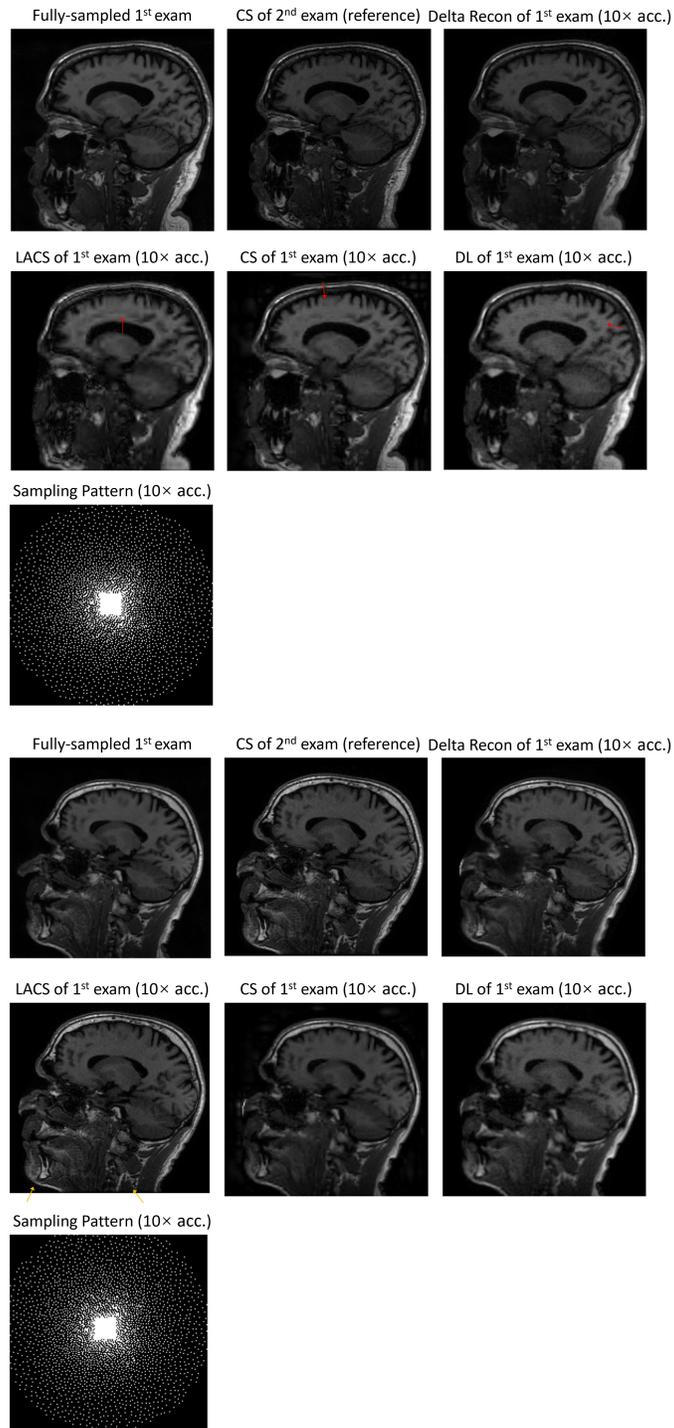
Figure 7.6: Reconstructed examples for the experiment setting (2), where the reference scan is the CS reconstruction of $2.5\times$ accelerated second exam. We retrospectively undersampled the first example with a $10\times$ accelerated Poisson disk mask. Red arrows indicate artifacts, and yellow arrows point out over-fitting with the reference scan. Delta reconstruction shows fewer artifacts and improved stability than other reconstruction methods.

Fully-sampled   Variable density Poisson-disk   Optimized sampling pattern
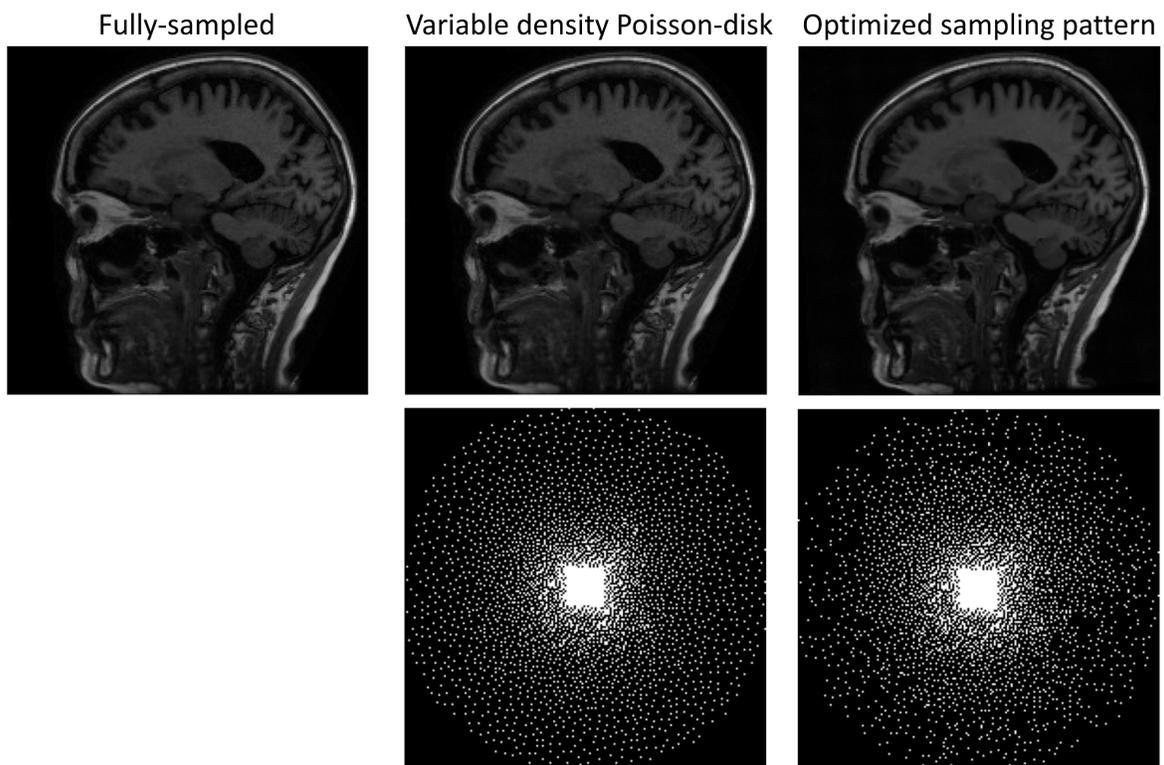
Figure 7.7: Examples of the joint optimization of sampling pattern and reconstruction. The optimized sampling pattern leads to a higher SNR.

$\sim 8\times$. For reference-based methods, we used the conjugate phase reconstruction of the first fully-sampled exam as the prior. Fig. 7.5 shows the results.

(2) The first exam was retrospectively undersampled following a 10x Poisson-disk sampling pattern. For reference-based methods, we used a CS-based reconstruction of the $2.5\times$ undersampled second exam as the prior. Fig. 7.6 shows the results.

(3) Joint sampling pattern optimization and delta reconstruction. Fig. 7.7 shows related results.

Fig. 7.5, Fig. 7.6, and Fig. 7.7 exhibit examples of reconstructed images in different scenarios. Compared with referenceless (CS or DL) reconstruction, Delta reconstruction improved the image quality, with the fully-sampled reference scans as priors. More importantly, it can still represent subtle anatomical changes (yellow arrows) not existing in the reference exam. Fig. 7.6 demonstrates that even if the reference exam is undersampled, the Delta reconstruction can still enhance the following exams. Compared to LACS, the method is more robust and avoids overfitting on the first exam. Though (7.1) is not convex, the convergence rate w.r.t. ground truth is still pretty fast, usually within 5-6 iterations when ν is small.

In conclusion, Delta Scan provides a fast, explainable, and robust method for longitudinal MRI, where the longitudinal MR exams are substantially accelerated with the structural information from the past scans as an image. In the future, we will conduct a more comprehensive reader study and extend Delta Scan to non-Cartesian and multi-contrast MRI.

7.3 MIRTorch: An Open-source Differentiable Image Reconstruction Toolbox

Image reconstruction converts raw signals into digitized images and is an essential part of modern medical imaging. High-quality image reconstruction results provide powerful tools for radiologists in accurate diagnosis. Consequently, fast and accurate image reconstruction has become an active area of research in optimization, signal processing, and computational imaging. Image reconstruction toolboxes provide algorithmic infrastructures and baselines to support these research efforts.

For many imaging modalities, such as magnetic resonance imaging (MRI), image reconstruction is an inverse problem that is often underdetermined and large-scale. Reconstruction toolboxes should model the imaging physics, implement regularization, and provide corresponding solvers. Recent years have also seen a surge in deep learning-based reconstruction that learns to invert [97]. One may integrate physics modeling into deep learning frameworks to combine the best of both worlds [2]. These model-based or physics-informed deep learning methods received wide attention because of their robustness and explainability. The main goal of MIRTorch is to assist research

on image reconstruction algorithm development using data-driven approaches. Natively built with PyTorch, MIRTorch fully supports auto-differentiation (AD), having an affinity to deep learning modules such as CNN or Transformers.[2]

Following the earlier MIRT (Michigan Image Reconstruction Toolbox)[3] and SigPy[4], MIRTorch has a clear and modular structure, facilitating fast prototyping of novel algorithms. The main features and components include: (1) Generalization of linear operators as matrices. In many image modalities, the forward system model is linear; thus, one may regard these models as matrices (though they may have otherwise efficient implementations like FFTs). By overloading operators such as +. -, *, MIRTorch aids researchers in defining system models and avoiding erroneous hard-coding. (2) Efficient iterative solvers. MIRTorch includes various solvers, including CG [128], FISTA [12], and primal-dual [26]. Users can easily combine learnable modules (such as CNNs) with numerical solvers to investigate model-based deep learning methods. Several common proximal operators are also provided.

With native PyTorch support, MIRTorch satisfies the following specific needs. (1) Fast prototyping of model-based deep learning. Many related projects hard-coded the physics-informed components, such as system operators and iterative solvers. This non-modular approach hampers reproducibility and comparability. MIRTorch provides a standardized and modular implementation. It also facilitates transferring algorithms across different imaging modalities. Additionally, the code using MIRTorch better matches the mathematical expressions, facilitating understanding. (2) Optimization of the imaging system. Since the toolbox is fully differentiable, it enables gradient methods for tuning imaging system parameters. For example, Ch. 4 and Ch. 5 use MIRTorch to optimize MRI sampling trajectories via stochastic gradient descent. (3) User-friendly GPU-based fast reconstruction. Many applications, such as fMRI, are vectorized and large-dimensional. CPU-based computation can be very time-consuming for iterative algorithms. Benefitting from PyTorch's intrinsic multi-GPU support, user-friendly installation, and cross-platform capability, MIRTorch provides researchers with fast reconstruction at a minimal switching cost.

---

[2]This section is based on a conference abstract [149].
[3]https://github.com/JeffFessler/mirt
[4]https://github.com/mikgroup/sigpy

CHAPTER 8

Future Work

The preceding chapters present a series of methods to optimize MRI sampling strategies. This chapter discusses several unsolved problems of current studies, and potential future research topics.

As discussed in Ch. 2.4, MRI systems exhibit various innate imperfections. Ch. 4 demonstrated the feasibility of implementing eddy currents correction for a 2D freeform optimized trajectory. In future studies, incorporating system imperfections into the forward learning/optimization phase, such as off-resonance maps and gradient delay [142], may enhance the intrinsic robustness of the optimized trajectory. However, this approach requires the distribution of system imperfections, which is typically scanner-specific. To address this limitation, we plan to investigate prospective simulation approaches in future studies. SNOPY-like algorithms can also incorporate other optimization objectives to encourage properties such as robustness to field inhomogeneity and reduction of acoustic noise. The pre-compensation to field inhomogeneity can be reformulated as a penalty of the point spread function considering a spatially constant B_0 shift [5, 106]. The reduction of acoustic noise can use noise response models as a loss term [62].

One question of particular interest is how the receiver coil layout affects the training. Other factors to consider include the variability of image phases (considering TE and off-resonance) and long-readout trajectories. Future studies should include related simulations and prospective studies to investigate these effects.

In Ch. 4 and Ch. 5, the optimized sampling trajectories showed good generalization ability between simulation and in-vivo prospective experiments. The protocols are also robust to the shift of image contrasts and hardware. We use the PSF as an intuitive explanation of the superior robustness. In the future, we plan to implement several relevant studies. For example, the sampling trajectories can be optimized with healthy cohorts and tested with pathological cases.

Ongoing and future research can extend the design by incorporating both upstream (signal generation) and downstream (post-processing and analysis) aspects of medical imaging scans. The physics-informed learning should include the (general) Bloch equation to consider physical processes such as relaxation, which will enable the learning of pulse sequences for both qualitative and

quantitative imaging. Notably, this can lead to the discovery of new MR sequences that enhance contrast for specific pathologies by jointly optimizing with AI-based image analysis.

Lastly, the proposed methods can contribute to clinical and physiological studies. They are mostly compatible with existing hardware and require minimal modification of workflows. The high image quality and spatiotemporal resolution may facilitate clinical and scientific innovations. Potential topics include:

- Neuroimaging. The proposed methods can optimize existing protocols such as simultaneous multi-slice (SMS) [126] to improve the spatiotemporal resolution and improve the patient experience (by suppressing the PNS effect and acoustic noise). The learned sampling trajectories can better utilize modern neuroimaging hardware, especially powerful gradient systems.

- Quantitative imaging. The dissertation research is also promising to improve the speed and quality of quantitative imaging by systematic optimization of pulse sequences, signal sampling, image reconstruction, and parameter estimation. As suggested, future research may automatically discover pulse sequences that maximize the contrast for certain pathologies with quantitative datasets and paired pathological annotations.

- Dynamic imaging and MR-guided intervention. Future studies should also consider the motion in the optimization. The extended SNOPY may also learn to compensate for the system imperfections and produce intrinsically robust protocols suitable for MR-guided intervention and low-field scanners.

APPENDIX A

Jacobians for Linear Operations Involving NUFFTs

A.1   Notations and Basics

For Jacobian matrices, we follow the 'numerator-layout' notation. For example, the derivative of an m-element column vector $\boldsymbol{y}$ w.r.t. an n-element vector $\boldsymbol{x}$ is an $m \times n$ matrix:

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \triangleq \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}. \tag{A.1}$$

However, this convention does not handle scenarios such as the derivatives of the elements of one matrix w.r.t. the elements of another matrix. Thus, we adopt a natural extension by using the vec (vectorization) operation. Specifically, for a $M \times N$ matrix $\boldsymbol{A}$ that is a function of a $P \times Q$ matrix $\boldsymbol{B}$, we write the derivative as a $MN \times PQ$ matrix by applying (A.1) to the vec of each matrix:

$$\mathcal{D}_{\boldsymbol{B}}\, \boldsymbol{A} = \mathcal{D}_{\boldsymbol{B}}\, \boldsymbol{A}(\boldsymbol{B}) \triangleq \frac{\partial\, \mathrm{vec}(\boldsymbol{A})}{\partial\, \mathrm{vec}(\boldsymbol{B})}. \tag{A.2}$$

The following equalities are useful in our derivations. (Equalities involving products all assume the sizes are compatible.) For $\boldsymbol{A} \in \mathbb{C}^{K \times L}$, $\boldsymbol{B} \in \mathbb{C}^{L \times M}$, $\boldsymbol{C} \in \mathbb{C}^{M \times N}$:

$$\begin{aligned} \mathrm{vec}(\boldsymbol{ABC}) &= (\boldsymbol{I}_N \otimes \boldsymbol{AB})\, \mathrm{vec}(\boldsymbol{C}) \\ &= (\boldsymbol{C}^T \boldsymbol{B}^T \otimes \boldsymbol{I}_K)\, \mathrm{vec}(\boldsymbol{A}). \end{aligned} \tag{A.P1}$$

In general:

$$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC}) \otimes (\boldsymbol{BD}). \tag{A.P2}$$

For $\boldsymbol{A} \in \mathbb{C}^{K \times L}$, $\boldsymbol{B} \in \mathbb{C}^{M \times N}$:

$$\boldsymbol{A} \otimes \boldsymbol{B} = (\boldsymbol{I}_K \otimes \boldsymbol{B})(\boldsymbol{A} \otimes \boldsymbol{I}_N) = (\boldsymbol{A} \otimes \boldsymbol{I}_M)(\boldsymbol{I}_L \otimes \boldsymbol{B}). \tag{A.P3}$$

For $\boldsymbol{A} \in \mathbb{C}^{M \times N}$, $\boldsymbol{x} \in \mathbb{C}^N$:

$$\mathcal{D}_{\boldsymbol{A}}(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{x}^T \otimes \boldsymbol{I}_M, \ \mathcal{D}_{\boldsymbol{A}^*}(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{0}. \tag{A.P4}$$

For an invertible matrix $\boldsymbol{A}$:

$$\boldsymbol{A} \in \mathbb{C}^{N \times N} \implies \mathcal{D}_{\boldsymbol{A}} \boldsymbol{A}^{-1} = -(\boldsymbol{A}^T)^{-1} \otimes \boldsymbol{A}^{-1},$$
$$\mathcal{D}_{\boldsymbol{A}^*} \boldsymbol{A}^{-1} = \boldsymbol{0}. \tag{A.P5}$$

The chain rule still holds for the extended Jacobian formulation. Suppose $F : \mathbb{C}^{K \times L} \to \mathbb{C}^{M \times N}$ and $G : \mathbb{C}^{M \times N} \to \mathbb{C}^{P \times Q}$ are both holomorphic. For $\boldsymbol{X} \in \mathbb{C}^{K \times L}$, the Jacobian of the composite function is:

$$\underbrace{\mathcal{D}_{\boldsymbol{X}} G(F(\boldsymbol{X}))}_{PQ \times KL} = \underbrace{\mathcal{D}_{\boldsymbol{Y}} G(\boldsymbol{Y})|_{\boldsymbol{Y}=F(\boldsymbol{X})}}_{PQ \times MN} \underbrace{\mathcal{D}_{\boldsymbol{X}} F(\boldsymbol{X})}_{MN \times KL},$$
$$\mathcal{D}_{\boldsymbol{X}^*} G(F(\boldsymbol{X})) = \boldsymbol{0}. \tag{A.P6}$$

Equalities (A.P1)-(A.P3) are common matrix vectorization properties. See [94, Ch. 9] for (A.P4), [65] for (A.P5) and (A.P6).

A.2 Approximation of Jacobians involving NUFFT

Consider the (single-coil, initially) MRI measurement model for non-Cartesian sampling based on the NUDFT [38]:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{y} \in \mathbb{C}^M$ denotes the measured k-space data, $\boldsymbol{x} \in \mathbb{C}^N$ denotes the unknown image to be reconstructed, and $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ denotes the system matrix or encoding matrix, where $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{\omega})$ has elements

$$a_{ij} = \mathrm{e}^{-\imath \vec{\omega}_i \cdot \vec{r}_j}, \quad i = 1, \ldots, M, \quad j = 1, \ldots, N$$

for $\vec{\omega}_i \in \mathbb{R}^D$ and $\vec{r}_j \in \mathbb{R}^D$ where $D \in \{1, 2, 3 \ldots\}$ denotes the image dimension, and where

$$\boldsymbol{\omega} = [\boldsymbol{\omega}^{[1]} \, \boldsymbol{\omega}^{[2]} \ldots \boldsymbol{\omega}^{[D]}]$$

is the $M \times d$ matrix consisting of all the k-space sampling locations and $\boldsymbol{\omega}^{[d]} \in \mathbb{R}^M$ denotes its dth column.

We first focus on the forward operation $\boldsymbol{A}(\boldsymbol{\omega}) \, \boldsymbol{x}$ and determine Jacobian matrices with respect to $\boldsymbol{x}$ and $\boldsymbol{\omega}$. The $M \times N$ Jacobian matrix of the forward linear operation with respect to $\boldsymbol{x}$ is

$$\frac{\partial \boldsymbol{A} \boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{A}, \qquad \frac{\partial \boldsymbol{A} \boldsymbol{x}}{\partial \boldsymbol{x}^*} = \boldsymbol{0}.$$

For the dth column of the spectrum sampling pattern $\boldsymbol{\omega}$, the Jacobian has elements

$$\left[\frac{\partial \boldsymbol{A} \boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} \right]_{il} = \frac{\partial [\boldsymbol{A} \boldsymbol{x}]_i}{\partial \omega_l^{[d]}} = \frac{\partial}{\partial \omega_l^{[d]}} \sum_{j=1}^{N} \mathrm{e}^{-\imath \vec{\omega}_i \cdot \vec{r}_j} x_j$$

$$= \begin{cases} -\imath \sum_{j=1}^{N} \mathrm{e}^{-\imath \vec{\omega}_i \cdot \vec{r}_j} x_j r_j^{[d]}, & i = l \\ 0, & \text{otherwise,} \end{cases}$$

for $i, l = 1, \dots, M$. The above summation is the product of the ith row of $-\imath \boldsymbol{A}$ with $\boldsymbol{x} \odot \boldsymbol{r}^{[d]}$. Thus the $M \times M$ Jacobian matrix for the partial derivatives of $\boldsymbol{A} \boldsymbol{x}$ w.r.t. $\boldsymbol{\omega}^{[d]}$ is (3.3):

$$\frac{\partial \boldsymbol{A} \boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} = -\imath \operatorname{diag} \left\{ \boldsymbol{A} (\boldsymbol{x} \odot \boldsymbol{r}^{[d]}) \right\}.$$

Consequently, the Jacobian calculation should apply $\boldsymbol{A}$ to vector $\boldsymbol{x} \odot \boldsymbol{r}^{[d]}$ once. In the above derivation, $\boldsymbol{A}$ is a NUDFT operator. In the practical implementation, we use a NUFFT to approximate $\boldsymbol{A}$, both for the forward model and for the Jacobian calculation.

Derivations of the Jacobians for the adjoint operation $\boldsymbol{A}'(\boldsymbol{\omega}) \, \boldsymbol{y}$ follow a similar approach. For $\boldsymbol{y}$:

$$\frac{\partial \boldsymbol{A}' \boldsymbol{y}}{\partial \boldsymbol{y}} = \boldsymbol{A}', \qquad \frac{\partial \boldsymbol{A}' \boldsymbol{y}}{\partial \boldsymbol{y}^*} = \boldsymbol{0}.$$

For the dth column of $\boldsymbol{\omega}$, the $N \times M$ Jacobian matrix has elements:

$$\left[\frac{\partial \boldsymbol{A}' \boldsymbol{y}}{\partial \boldsymbol{\omega}^{[d]}} \right]_{jl} = \frac{\partial [\boldsymbol{A}' \boldsymbol{y}]_j}{\partial \omega_l^{[d]}} = \frac{\partial \sum_{i=1}^{M} \mathrm{e}^{\imath \vec{\omega}_i \cdot \vec{r}_j} y_i}{\partial \omega_l^{[d]}}$$

$$= \imath \mathrm{e}^{\imath \vec{\omega}_i \cdot \vec{r}_j} y_i r_j^{[d]}.$$

Thus the Jacobian matrix is (3.4):

$$\frac{\partial \boldsymbol{A}' \boldsymbol{y}}{\partial \boldsymbol{\omega}^{[d]}} = \imath \operatorname{diag} \left\{ \boldsymbol{r}^{[d]} \right\} \boldsymbol{A}' \operatorname{diag} \{ \boldsymbol{y} \}.$$

The product $\boldsymbol{A}'(\boldsymbol{\omega}) \, \boldsymbol{A}(\boldsymbol{\omega}) \, \boldsymbol{x}$ of the Gram matrix of the NUDFT with a vector also arises in

optimization steps and requires appropriate Jacobian matrices. For $\boldsymbol{x}$:

$$\frac{\partial \boldsymbol{A}'\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{A}'\boldsymbol{A}, \qquad \frac{\partial \boldsymbol{A}'\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}^*} = \boldsymbol{0}.$$

The (k, j)th element of the $N \times N$ matrix containing the partial derivatives of the Gram matrix w.r.t. $\omega_l^{[d]}$ is

$$
\begin{aligned}
\left[\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_l^{[d]}}\right]_{k,j} &= \frac{\partial}{\partial \omega_l^{[d]}} \sum_{i=1}^{M} \mathrm{e}^{-\imath \vec{\omega}_i \cdot (\vec{r}_j - \vec{r}_k)} \\
&= -\imath \left(r_j^{[d]} - r_k^{[d]}\right) \mathrm{e}^{-\imath \vec{\omega}_l \cdot (\vec{r}_j - \vec{r}_k)} \\
&= -\imath \left(r_j^{[d]} - r_k^{[d]}\right) a_{lk}^* a_{lj}.
\end{aligned}
\tag{A.3}
$$

In matrix form:

$$\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_l^{[d]}} = \imath \operatorname{diag}\{\boldsymbol{r}^{[d]}\} \boldsymbol{A}'\boldsymbol{e}_l \boldsymbol{e}_l' \boldsymbol{A} - \imath \boldsymbol{A}'\boldsymbol{e}_l \boldsymbol{e}_l' \boldsymbol{A} \operatorname{diag}\{\boldsymbol{r}^{[d]}\}. \tag{A.4}$$

When multiplying the Jacobian with a vector $\boldsymbol{x}$:

$$
\begin{aligned}
\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_l^{[d]}} \boldsymbol{x} &= \imath \operatorname{diag}\{\boldsymbol{r}^{[d]}\} \boldsymbol{a}_l (\boldsymbol{a}_l' \boldsymbol{x}) - \imath \boldsymbol{a}_l \boldsymbol{a}_l' \operatorname{diag}\{\boldsymbol{r}^{[d]}\} \boldsymbol{x} \\
&= \imath (\boldsymbol{a}_l' \boldsymbol{x})(\boldsymbol{r}^{[d]} \odot \boldsymbol{a}_l) - \imath (\boldsymbol{a}_l' (\boldsymbol{x} \odot \boldsymbol{r}^{[d]})) \boldsymbol{a}_l,
\end{aligned}
\tag{A.5}
$$

where $\boldsymbol{a}_l = \boldsymbol{A}'\boldsymbol{e}_l$ denotes the lth column of $\boldsymbol{A}'$.

Consider the extended Jacobian expression:

$$\mathcal{D}_{\omega_l^{[d]}} \boldsymbol{A}'\boldsymbol{A} = \operatorname{vec}\left(\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_l^{[d]}}\right).$$

Multiplying by $\boldsymbol{x}$ yields:

$$\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_l^{[d]}}\boldsymbol{x} = \mathrm{vec}\left(\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_l^{[d]}}\boldsymbol{x}\right)$$

$$= (\boldsymbol{x}^T \otimes \boldsymbol{I}_N)\,\mathrm{vec}\left(\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_l^{[d]}}\right) \qquad \text{(use A.P1)}$$

$$= (\boldsymbol{x}^T \otimes \boldsymbol{I}_N)\left(\mathcal{D}_{\omega_l^{[d]}}\,\boldsymbol{A}'\boldsymbol{A}\right)$$

$$= (\mathcal{D}_{\boldsymbol{A}'\boldsymbol{A}}\,\boldsymbol{A}'\boldsymbol{A}\boldsymbol{x})\left(\mathcal{D}_{\omega_l^{[d]}}\,\boldsymbol{A}'\boldsymbol{A}\right) \qquad \text{(use A.P4)}$$

$$= \mathcal{D}_{\omega_l^{[d]}}\,\boldsymbol{A}'\boldsymbol{A}\boldsymbol{x}. \qquad \text{(use A.P6)}$$

Concatenating (A.5) by columns leads to the matrix (3.5):

$$\left[\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_1^{[d]}} \quad \cdots \quad \frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_M^{[d]}}\right]\boldsymbol{x} = -\imath\,\boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}(\boldsymbol{x}\odot\boldsymbol{r}^{[d]})\} + \imath\,\mathrm{diag}\{\boldsymbol{r}^{[d]}\}\,\boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}\boldsymbol{x}\}.$$

Alternatively, we can express the extended Jacobian as

$$\left[\frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_1^{[d]}} \quad \cdots \quad \frac{\partial \boldsymbol{A}'\boldsymbol{A}}{\partial \omega_M^{[d]}}\right]\boldsymbol{x} = (\boldsymbol{x}^T \otimes \boldsymbol{I}_n)\left(\mathcal{D}_{\boldsymbol{\omega}^{[d]}}\,\boldsymbol{A}'\boldsymbol{A}\right)$$

$$= (\mathcal{D}_{\boldsymbol{A}'\boldsymbol{A}}\,\boldsymbol{A}'\boldsymbol{A}\boldsymbol{x})\left(\mathcal{D}_{\boldsymbol{\omega}^{[d]}}\,\boldsymbol{A}'\boldsymbol{A}\right)$$

$$= \mathcal{D}_{\boldsymbol{\omega}^{[d]}}\,\boldsymbol{A}'\boldsymbol{A}\boldsymbol{x}. \qquad (\text{A.6})$$

Again we use NUFFT operations for efficient approximation.

Image reconstruction methods based on algorithms like the augmented Lagrangian approach [64] use "data consistency" steps [2, 22, 112] that often involve least-squares problems with solutions in the following form:

$$(\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{I})^{-1}\boldsymbol{x},$$

for some vector $\boldsymbol{x} \in \boldsymbol{C}^N$, or

$$(\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{x}, \qquad (\text{A.7})$$

where $\boldsymbol{T}$ denotes a linear regularization operator that is independent of $\boldsymbol{\omega}$. In both cases, $\lambda > 0$ and the null spaces of $\boldsymbol{T}$ and $\boldsymbol{A}$ are usually disjoint, so the Hessian matrix is invertible. A few iterations of a CG method usually suffices to efficiently compute the approximate product of such a matrix inverse with a vector. The direct inverse is impractical for large-scale problems like MRI. Following [2], we treat CG as solving the above equations accurately, so that we can derive efficient approximations as follows. Otherwise, attempting to auto-differentiate through a finite number of

CG iterations would require large amounts of memory. Here we derive the corresponding Jacobian matrices for the exact inverse to (A.7) and then apply fast approximations. For $\boldsymbol{x}$, the $N \times N$ Jacobian is

$$\frac{\partial (\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{x}}{\partial \boldsymbol{x}} = (\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T})^{-1},$$

$$\frac{\partial (\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{x}}{\partial \boldsymbol{x}^*} = 0.$$

We can still use CG (with NUFFT) to efficiently multiply this Jacobian by a vector, albeit approximately.

To consider the Jacobian w.r.t. the sampling pattern $\boldsymbol{\omega}^{[d]}$, define $\boldsymbol{z} = (\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{x}$ and $\boldsymbol{F} = \boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T}$. We assume that $\boldsymbol{A}$ and $\boldsymbol{T}$ have disjoint null spaces, so that $\boldsymbol{F}$ is positive definite and hence invertible. Applying equalities derived above leads to the following expression for the $M \times N$ Jacobian:

$$
\begin{aligned}
\mathcal{D}_{\boldsymbol{\omega}^{[d]}} \, \boldsymbol{F}^{-1}\boldsymbol{x} &= \left(\mathcal{D}_{\boldsymbol{F}} \, \boldsymbol{F}^{-1}\boldsymbol{x}\right)\left(\mathcal{D}_{\boldsymbol{\omega}^{[d]}} \, \boldsymbol{F}\right) && \text{use A.P6} \\
&= -(\boldsymbol{x}^T \otimes \boldsymbol{I})((\boldsymbol{F}^T)^{-1} \otimes \boldsymbol{F}^{-1})\left(\mathcal{D}_{\boldsymbol{\omega}^{[d]}} \, \boldsymbol{F}\right) && \text{use A.P5} \\
&= -\left((\boldsymbol{x}^T(\boldsymbol{F}^T)^{-1}) \otimes \boldsymbol{F}^{-1}\right)\left(\mathcal{D}_{\boldsymbol{\omega}^{[d]}} \, \boldsymbol{F}\right) && \text{use A.P2} \\
&= -\boldsymbol{F}^{-1}(\boldsymbol{x}^T(\boldsymbol{F}^T)^{-1} \otimes \boldsymbol{I})\left(\mathcal{D}_{\boldsymbol{\omega}^{[d]}} \, \boldsymbol{F}\right) && \text{use A.P3} \\
&= -\boldsymbol{F}^{-1}\left(\mathcal{D}_{\boldsymbol{\omega}^{[d]}} \, \boldsymbol{F}\boldsymbol{z}\right) && \text{use A.P4} \\
&= -(\boldsymbol{A}'\boldsymbol{A} + \lambda \boldsymbol{T}'\boldsymbol{T})^{-1}\Big(-\imath \, \boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}(\boldsymbol{z} \odot \boldsymbol{r}^{[d]})\} \\
&\qquad\qquad + \imath \, \mathrm{diag}\{\boldsymbol{r}^{[d]}\} \, \boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}\boldsymbol{z}\} \Big) && \text{use (3.5).}
\end{aligned}
$$

We apply this Jacobian to a vector by using four NUFFT operations followed by running CG to approximate the product of $\boldsymbol{F}^{-1}$ times a vector. Notably, the memory cost of (3.6) is constant w.r.t the number of iterations, whereas the standard auto-differentiation approach has linear memory cost. Using the proposed method, one may apply enough iterations to ensure convergence to a desired tolerance. This new fast and low-memory Jacobian approximation is particularly important for the MRI applications shown in the following sections. Without this approximation, memory cost can be prohibitively large.

In multi-coil (parallel) acquisition, the MRI system model contains another linear operator

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_1 \\ \vdots \\ \boldsymbol{S}_{N_{\mathrm{c}}} \end{bmatrix},$$

where $\boldsymbol{S}_i = \mathrm{diag}\{\boldsymbol{s}_i\}$ denotes a diagonal matrix containing the receiver coil sensitivity map [110]. The total number of receiver channels is N_c. The system matrix ($\boldsymbol{E}$) for MRI in this case becomes $(\boldsymbol{I}_{N_c} \otimes \boldsymbol{A})\boldsymbol{S}$. Because of the special block-diagonal structure of $\boldsymbol{S}$, all the Jacobian matrices in previous sections still hold by simply replacing $\boldsymbol{A}$ with $\boldsymbol{E}$.

The Jacobian derivations are as follows. For the forward operator, one can show

$$
\frac{\partial \boldsymbol{E}\boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} = \frac{\partial \begin{bmatrix} \boldsymbol{A}\boldsymbol{S}_1\boldsymbol{x} \\ \vdots \\ \boldsymbol{A}\boldsymbol{S}_{N_c}\boldsymbol{x} \end{bmatrix}}{\partial \boldsymbol{\omega}^{[d]}} = \begin{bmatrix} -\imath\,\mathrm{diag}\{\boldsymbol{A}(\boldsymbol{s}_1 \odot \boldsymbol{x} \odot \boldsymbol{r}^{[d]})\} \\ \vdots \\ -\imath\,\mathrm{diag}\{\boldsymbol{A}(\boldsymbol{s}_{N_c} \odot \boldsymbol{x} \odot \boldsymbol{r}^{[d]})\} \end{bmatrix}
$$

$$
= \imath\,\mathrm{diag}\{(\boldsymbol{I}_{N_c} \otimes \boldsymbol{A})\boldsymbol{S}(\boldsymbol{x} \odot \boldsymbol{r}^{[d]})\}
$$

$$
= \imath\,\mathrm{diag}\{\boldsymbol{E}(\boldsymbol{x} \odot \boldsymbol{r}^{[d]})\} \, .
$$

The adjoint operator follows the same proof and produces

$$
\frac{\partial \boldsymbol{E}'\boldsymbol{y}}{\partial \boldsymbol{\omega}^{[d]}} = \imath\,\mathrm{diag}\{\boldsymbol{r}^{[d]}\}\,\boldsymbol{E}'\mathrm{diag}\{\boldsymbol{y}\} \, .
$$

For the gram operator we have

$$
\frac{\partial \boldsymbol{E}'\boldsymbol{E}\boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} = \sum_i \frac{\partial \boldsymbol{S}_i'\boldsymbol{A}'\boldsymbol{A}\boldsymbol{S}_i\boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} = \sum_i \boldsymbol{S}_i'\frac{\partial \boldsymbol{A}'\boldsymbol{A}\boldsymbol{S}_i\boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}}
$$

$$
= \sum_i -\imath\,\boldsymbol{S}_i'\boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}(\boldsymbol{S}\boldsymbol{x} \odot \boldsymbol{r}^{[d]})\} + \imath\,\boldsymbol{S}_i'\mathrm{diag}\{\boldsymbol{r}^{[d]}\}\,\boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}\boldsymbol{S}_i\boldsymbol{x}\}
$$

$$
= \sum_i -\imath\,\boldsymbol{S}_i'\boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}(\boldsymbol{S}\boldsymbol{x} \odot \boldsymbol{r}^{[d]})\} + \imath\,\mathrm{diag}\{\boldsymbol{r}^{[d]}\}\,\boldsymbol{S}_i'\boldsymbol{A}'\mathrm{diag}\{\boldsymbol{A}\boldsymbol{S}_i\boldsymbol{x}\}
$$

$$
= -\imath\,\boldsymbol{E}'\mathrm{diag}\{\boldsymbol{E}(\boldsymbol{x} \odot \boldsymbol{r}^{[d]})\} + \imath\,\mathrm{diag}\{\boldsymbol{r}^{[d]}\}\,\boldsymbol{E}'\mathrm{diag}\{\boldsymbol{E}\boldsymbol{x}\} \, . \tag{A.8}
$$

For the inverse of the PSD matrix let $\boldsymbol{G} = \boldsymbol{E}'\boldsymbol{E} + \lambda\boldsymbol{T}'\boldsymbol{T}$ and $\boldsymbol{z} = \boldsymbol{G}^{-1}\boldsymbol{x}$ (in the usual case where the regularizer matrix $\boldsymbol{T}$ is designed such that $\boldsymbol{G}$ is invertible). Combining (3.6) and (A.8) produces:

$$
\frac{\partial\,(\boldsymbol{E}'\boldsymbol{E} + \lambda\boldsymbol{T}'\boldsymbol{T})^{-1}\,\boldsymbol{x}}{\partial \boldsymbol{\omega}^{[d]}} = -\boldsymbol{G}^{-1}\,(\boldsymbol{x}^T\boldsymbol{G}(t)^{-1} \otimes \boldsymbol{I})\,\mathcal{D}_{\boldsymbol{\omega}^{[d]}}\,\boldsymbol{G}
$$

$$
= -(\boldsymbol{E}'\boldsymbol{E} + \lambda\boldsymbol{T}'\boldsymbol{T})^{-1}\Big(-\imath\,\boldsymbol{E}'\mathrm{diag}\{\boldsymbol{E}(\boldsymbol{z} \odot \boldsymbol{r}^{[d]})\}
$$

$$
+ \imath\,\mathrm{diag}\{\boldsymbol{r}^{[d]}\}\,\boldsymbol{E}'\mathrm{diag}\{\boldsymbol{E}\boldsymbol{z}\}\Big).
$$

Again, we apply this Jacobian matrix to a vector by combining NUFFTs and CG.

For MRI scans with long readouts, one should also consider the effects of off-resonance (e.g., B_0 field inhomogeneity), in which case the system matrix elements are given by [38]

$$a_{ij} = \mathrm{e}^{-\imath \vec{\omega}_i \cdot \vec{r}_j} \, \mathrm{e}^{-\imath w_j t_i},$$

where w_j denotes the field map value at the jth voxel and t_i is the time of the ith readout sample.

This form is no longer a Fourier transform operation, but there are fast and accurate approximations [40] that enable the use of NUFFT steps and avoid the very slow matrix-vector multiplication. Such approximations of system matrix $\boldsymbol{E}$ usually have the form:

$$\boldsymbol{E}_{\mathrm{f}} \approx \sum_{l=1}^{L} \mathrm{diag}\{b_{il}\} \, \boldsymbol{A}(\boldsymbol{\omega}) \, \mathrm{diag}\{c_{lj}\} \,,$$

where $\boldsymbol{A}$ denotes the usual (possibly non-uniform) DFT that is usually approximated by a NUFFT, $b_{il} \in \mathbb{C}^M$, and $b_{il} \in \mathbb{C}^N$. It is relatively straightforward to generalize the Jacobian expressions in this paper to handle the case of field inhomogeneity, by simply replacing $\boldsymbol{A}$ with $\boldsymbol{E}_{\mathrm{f}}$, similar to the sensitivity map case.

A.3   Error Bounds

The maximum error is

$$\varepsilon_p \triangleq \|\mathrm{vec}(\boldsymbol{E})\|_\infty = \max_{m,n} |e_{mn}|$$
$$\leq \frac{\max_{n,\omega_m} \left| \sum_{l\neq 0} c_{n+lK} \, \mathrm{e}^{-\imath \omega_m \cdot (n+lK)} \right|}{\min_n |c_n|}.$$

The maximum absolute value of e_{mn} depends on the frequency behavior of the interpolator, and is tabulated in figures in [41] for various NUFFT parameters p.

The Jacobian of the forward operator (3.3) is

$$\boldsymbol{J} = \frac{\partial \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{\omega}} = -\imath \, \mathrm{diag}\{\boldsymbol{A}(\boldsymbol{x} \odot \boldsymbol{r})\} \,.$$

Let $\tilde{\boldsymbol{J}}$ denote the case where an NUFFT is applied. Since the backpropagation uses Jacobians in the JVP calculation, here we analyze the error of JVPs using $\boldsymbol{J}$ and $\tilde{\boldsymbol{J}}$. We define the worst-case

relative error for a JVP with a (gradient) vector $\boldsymbol{v}$ as follows:

$$
\begin{aligned}
E_1(\boldsymbol{\omega}, \boldsymbol{x}, p) &\triangleq \max_{\|\boldsymbol{v}\|_\infty = 1} \|\tilde{\boldsymbol{J}}\boldsymbol{v} - \boldsymbol{J}\boldsymbol{v}\|_\infty / \|\boldsymbol{x}\|_2 \\
&= \max_{\|\boldsymbol{v}\|_\infty = 1} \|(\boldsymbol{E}\,(\boldsymbol{x} \odot \boldsymbol{r})) \odot \boldsymbol{v}\|_\infty / \|\boldsymbol{x}\|_2 \\
&\leq \|\boldsymbol{E}\,(\boldsymbol{x} \odot \boldsymbol{r})\|_\infty / \|\boldsymbol{x}\|_2 \\
&\leq \varepsilon_p \|\boldsymbol{x} \odot \boldsymbol{r}\|_2 / \|\boldsymbol{x}\|_2 \leq \varepsilon_p \|\boldsymbol{r}\|_\infty \,.
\end{aligned}
\tag{A.9}
$$

Similarly, the worst-case relative error of a JVP with (3.4) is bounded by

$$
\begin{aligned}
E_2(\boldsymbol{\omega}, \boldsymbol{x}, p) &\triangleq \max_{\|\boldsymbol{v}\|_\infty = 1} \|\operatorname{diag}\{\boldsymbol{r}\}\,\boldsymbol{E}'\operatorname{diag}\{\boldsymbol{y}\}\,\boldsymbol{v}\|_\infty / \|\boldsymbol{y}\|_2 \\
&\leq \max_{\|\boldsymbol{v}\|_\infty = 1} \|\boldsymbol{r}\|_\infty \|\boldsymbol{E}'(\boldsymbol{y} \odot \boldsymbol{v})\|_\infty / \|\boldsymbol{y}\|_2 \\
&\leq \varepsilon_p \|\boldsymbol{r}\|_\infty \max_{\|\boldsymbol{v}\|_\infty = 1} \|\boldsymbol{y} \odot \boldsymbol{v}\|_2 / \|\boldsymbol{y}\|_2 \\
&\leq \varepsilon_p \|\boldsymbol{r}\|_\infty \|\boldsymbol{y}\|_2 / \|\boldsymbol{y}\|_2 \leq \varepsilon_p \|\boldsymbol{r}\|_\infty \,.
\end{aligned}
\tag{A.10}
$$

In both cases, the worst-case error of the NUFFT approximation for a JVP is bounded by the usual NUFFT error multiplied by a constant $\|\boldsymbol{r}\|_\infty$ that is usually half of the field of view (FOV) in imaging applications. This constant is expected from unit analysis. If the sampling grid r_j has a unit in cm, then the sample locations $\boldsymbol{\omega}$ have units in radians/cm. Corresponding, the Jacobian matrices in (3.3) and (3.4) have units in cm, because $\boldsymbol{A}$ is unitless. The NUFFT error ε_p is unitless, so there is an $\boldsymbol{r}$-related factor in the JVP error E. In other words, the error bounds above depend on the choice of units. One could express the FOV in voxels to get the unitless error bound $\varepsilon_p N/2$. However, the accuracy of JVPs does not necessarily deteriorate with a larger N. Above we normalized the error by $\|\boldsymbol{x}\|_2$ or $\|\boldsymbol{y}\|_2$, whereas the Jacobians are ' scaled' with $\|\boldsymbol{x} \odot \boldsymbol{r}\|_2$ or $\|\boldsymbol{y}\|_2 \|\boldsymbol{r}\|_2$. A relative error could better describe the effect on optimization.

An alternate definition uses the worst-case in the numerator relative to an average case in the denominator, considering the stochastic gradient descent-like optimizers. For example, this relative error for the JVP of Jacobian (3.4) is

$$
\begin{aligned}
\epsilon &\triangleq \frac{\max_{\|\boldsymbol{x}\|_2 = 1} \|\tilde{\boldsymbol{J}} - \boldsymbol{J}\|_{\mathrm{F}}}{\sqrt{\mathsf{E}_{p(\boldsymbol{x})}[\|\boldsymbol{J}\|_{\mathrm{F}}^2]}} = \frac{\max_{\|\boldsymbol{x}\|_2 = 1} \|\boldsymbol{E}\,(\boldsymbol{x} \odot \boldsymbol{r})\|_2}{\sqrt{\mathsf{E}_{p(\boldsymbol{x})}[\|\boldsymbol{A}\,(\boldsymbol{x} \odot \boldsymbol{r})\|_2^2]}} \\
&\leq \frac{\max_{\|\boldsymbol{x}\|_2 = 1} \sqrt{M}\,\|\boldsymbol{E}\,(\boldsymbol{x} \odot \boldsymbol{r})\|_\infty}{\sqrt{\mathsf{E}_{p(\boldsymbol{x})}[\|\boldsymbol{A}\,(\boldsymbol{x} \odot \boldsymbol{r})\|_2^2]}} \leq \frac{\sqrt{M}\varepsilon_p \|\boldsymbol{r}\|_\infty}{\sqrt{\mathsf{E}_{p(\boldsymbol{x})}[\|\boldsymbol{A}\,(\boldsymbol{x} \odot \boldsymbol{r})\|_2^2]}},
\end{aligned}
$$

where $\mathsf{E}_{p(\boldsymbol{x})}[\cdot]$ denotes expectation w.r.t. a certain distribution $p(\boldsymbol{x})$. For parity with the unit sphere constraint in the numerator, we consider the case where $p(\cdot)$ is the random distribution on the unit N-sphere. Use the cyclic property of the trace:

$$\|\boldsymbol{A}\,(\boldsymbol{x}\odot\boldsymbol{r})\|_2^2 = \boldsymbol{x}'\mathrm{diag}\{\boldsymbol{r}\}\,\boldsymbol{A}'\boldsymbol{A}\mathrm{diag}\{\boldsymbol{r}\}\,\boldsymbol{x}$$
$$= \mathrm{Tr}\{\mathrm{diag}\{\boldsymbol{r}\}\,\boldsymbol{A}'\boldsymbol{A}\mathrm{diag}\{\boldsymbol{r}\}\,\boldsymbol{x}\boldsymbol{x}'\}\,.$$

Since the covariance of random points on the N-sphere is $(1/N)\boldsymbol{I}$, the denominator's expectation is

$$\mathsf{E}_{p(\boldsymbol{x})}[\|\boldsymbol{A}\,(\boldsymbol{x}\odot\boldsymbol{r})\|_2^2] = \mathrm{Tr}\{\mathrm{diag}\{\boldsymbol{r}\}\,\boldsymbol{A}'\boldsymbol{A}\mathrm{diag}\{\boldsymbol{r}\}\,\mathsf{E}_{p(\boldsymbol{x})}[\boldsymbol{x}\boldsymbol{x}']\}$$
$$= \frac{1}{N}\mathrm{Tr}\{\mathrm{diag}\{\boldsymbol{r}\}\,\boldsymbol{A}'\boldsymbol{A}\mathrm{diag}\{\boldsymbol{r}\}\}$$
$$= \frac{1}{N}\sum_j r_j^2[\boldsymbol{A}'\boldsymbol{A}]_{jj} = \frac{M}{N}\sum_j r_j^2 = \frac{M}{N}\,\|\boldsymbol{r}\|_2^2\,.$$

Thus we have the following bound for the relative error:

$$\epsilon \le \frac{\sqrt{M}\varepsilon_p\,\|\boldsymbol{r}\|_\infty}{\sqrt{M/N}\,\|\boldsymbol{r}\|_2} = \varepsilon_p\sqrt{N}\frac{\|\boldsymbol{r}\|_\infty}{\|\boldsymbol{r}\|_2} \le \varepsilon_p\sqrt{N}. \tag{A.11}$$

Note that the bound can be tighter when considering specific formulations of $\boldsymbol{r}$. Similarly, for the Jacobian operator (3.4), the alternate error of the JVP is

$$\epsilon \le \varepsilon_p\sqrt{M}. \tag{A.12}$$

BIBLIOGRAPHY

[1] Hemant K. Aggarwal and Mathews Jacob. Joint Optimization of Sampling Patterns and Deep Priors for Improved Parallel MRI. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8901–8905, May 2020.

[2] Hemant K. Aggarwal, Merry P. Mani, and Mathews Jacob. MoDL: Model-Based Deep Learning Architecture for Inverse Problems. *IEEE Transactions on Medical Imaging*, 38(2):394–405, February 2019.

[3] C. B. Ahn, J. H. Kim, and Z. H. Cho. High-Speed Spiral-Scan Echo Planar NMR Imaging-I. *IEEE Transactions on Medical Imaging*, 5(1):2–7, March 1986.

[4] C.B. Ahn and Z.H. Cho. Analysis of the eddy-current induced artifacts and the temporal compensation in nuclear magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, 10(1):47–52, March 1991.

[5] Ergun Ahunbay and James G. Pipe. Rapid method for deblurring spiral MR images. *Magnetic Resonance in Medicine*, 44(3):491–494, 2000.

[6] Jonathan Alush-Aben, Linor Ackerman-Schraier, Tomer Weiss, Sanketh Vedula, Ortal Senouf, and Alex Bronstein. 3D FLAT: Feasible Learned Acquisition Trajectories for Accelerated MRI. In Farah Deeba, Patricia Johnson, Tobias Würfl, and Jong Chul Ye, editors, *Machine Learning for Medical Image Reconstruction*, Lecture Notes in Computer Science, pages 3–16, Cham, 2020. Springer International Publishing.

[7] Jesper L. R. Andersson, Stefan Skare, and John Ashburner. How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging. *NeuroImage*, 20(2):870–888, October 2003.

[8] John Ashburner and Karl J. Friston. Unified segmentation. *NeuroImage*, 26(3):839–851, July 2005.

[9] Cagla D. Bahadir, Alan Q. Wang, Adrian V. Dalca, and Mert R. Sabuncu. Deep-Learning-Based Optimization of the Under-Sampling Pattern in MRI. *IEEE Transactions on Computational Imaging*, 6:1139–1152, 2020.

[10] Andrew V. Barger, Walter F. Block, Yuriy Toropov, Thomas M. Grist, and Charles A. Mistretta. Time-resolved contrast-enhanced imaging with isotropic resolution and broad coverage using an undersampled 3D projection trajectory. *Magnetic Resonance in Medicine*, 48(2):297–305, 2002.

[11] Philip J. Beatty, Dwight G. Nishimura, and John M. Pauly. Rapid gridding reconstruction with a minimal oversampling ratio. *IEEE Transactions on Medical Imaging*, 24(6):799–808, June 2005.

[12] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009.

[13] Bob van den Bergen, Cornelis A. T. Van den Berg, Lambertus W. Bartels, and Jan J. W. Lagendijk. 7 T body MRI: B1 shimming with simultaneous SAR reduction. *Physics in Medicine & Biology*, 52(17):5429, August 2007.

[14] Berkin Bilgic, Borjan A. Gagoski, Stephen F. Cauley, Audrey P. Fan, Jonathan R. Polimeni, P. Ellen Grant, Lawrence L. Wald, and Kawin Setsompop. Wave-CAIPI for highly accelerated 3D imaging. *Magnetic Resonance in Medicine*, 73(6):2152–2162, 2015.

[15] A Bilgin, TP Troouard, AF Gmitro, and MI Altbach. Randomly Perturbed Radial Trajectories for Compressed Sensing MRI. In *Proc. Intl. Soc. Magn. Reson. Med. (ISMRM)*, volume 16, page 3152, 2008.

[16] Samuel Birns, Bohyun Kim, Stephanie Ku, Kevin Stangl, and Deanna Needell. A Practical Study of Longitudinal Reference Based Compressed Sensing for MRI, August 2016. arXiv:1608.04728 [cs, math].

[17] Claire Boyer, Nicolas Chauffert, Philippe Ciuciu, Jonas Kahn, and Pierre Weiss. On the Generation of Sampling Schemes for Magnetic Resonance Imaging. *SIAM Journal on Imaging Sciences*, 9(4):2039–2072, January 2016.

[18] M Brant-Zawadzki, G D Gillan, and W R Nitz. MP RAGE: a three-dimensional, T1-weighted, gradient-echo sequence–initial experience in the brain. *Radiology*, 182(3):769–775, March 1992.

[19] Felix A. Breuer, Martin Blaimer, Matthias F. Mueller, Nicole Seiberlich, Robin M. Heidemann, Mark A. Griswold, and Peter M. Jakob. Controlled aliasing in volumetric parallel imaging (2D CAIPIRINHA). *Magnetic Resonance in Medicine*, 55(3):549–556, 2006.

[20] Yue Cao and David N. Levin. Feature-recognizing MRI. *Magnetic Resonance in Medicine*, 30(3):305–317, 1993.

[21] G. R. Chaithya, Pierre Weiss, Guillaume Daval-Frérot, Aurélien Massire, Alexandre Vignaud, and Philippe Ciuciu. Optimizing Full 3D SPARKLING Trajectories for High-Resolution Magnetic Resonance Imaging. *IEEE Transactions on Medical Imaging*, 41(8):2105–2117, August 2022.

[22] Stanley H. Chan, Xiran Wang, and Omar A. Elgendy. Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, March 2017.

[23] Nicolas Chauffert, Philippe Ciuciu, and Pierre Weiss. Variable density compressed sensing in MRI. Theoretical vs heuristic sampling strategies. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 298–301, April 2013.

[24] Nicolas Chauffert, Pierre Weiss, Jonas Kahn, and Philippe Ciuciu. A Projection Algorithm for Gradient Waveforms Design in Magnetic Resonance Imaging. *IEEE Transactions on Medical Imaging*, 35(9):2026–2039, September 2016.

[25] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12403–12412, June 2022.

[26] Laurent Condat. A Primal—Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, August 2013.

[27] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, August 2007.

[28] B. Dale, M. Wendt, and J.L. Duerk. A rapid look-up table method for reconstructing MR images from arbitrary K-space trajectories. *IEEE Transactions on Medical Imaging*, 20(3):207–217, March 2001.

[29] Mathias Davids, Bastien Guérin, Axel vom Endt, Lothar R. Schad, and Lawrence L. Wald. Prediction of Peripheral Nerve Stimulation Thresholds of MRI Gradient Coils using Coupled Electromagnetic and Neurodynamic Simulations. *Magnetic resonance in medicine*, 81(1):686–701, January 2019.

[30] Arjun D. Desai, Andrew M. Schmidt, Elka B. Rubin, Christopher M. Sandino, Marianne S. Black, Valentina Mazzoli, Kathryn J. Stevens, Robert Boutin, Christopher Ré, Garry E. Gold, Brian A. Hargreaves, and Akshay S. Chaudhari. SKM-TEA: A Dataset for Accelerated MRI Reconstruction with Dense Image Labels for Quantitative Clinical Evaluation, March 2022. arXiv:2203.06823 [cs, eess].

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

[32] G. M. Dilshan Godaliyadda, Gregery T. Buzzard, and Charles A. Bouman. A model-based framework for fast dynamic image sampling. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1822–1826, May 2014.

[33] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[34] Simon J. Doran, Liz Charles-Edwards, Stefan A. Reinsberg, and Martin O. Leach. A complete distortion correction for MR images: I. Gradient warp correction. *Physics in Medicine & Biology*, 50(7):1343, March 2005.

[35] Jeff H. Duyn, Yihong Yang, Joseph A. Frank, and Jan W. van der Veen. Simple Correction Method for K-space Trajectory Deviations in MRI. *Journal of Magnetic Resonance*, 132(1):150–153, May 1998.

[36] AbdEl Monem El-Sharkawy, Michael Schär, Paul A. Bottomley, and Ergin Atalar. Monitoring and correcting spatio-temporal variations of the MR scanner's static magnetic field. *Magma (New York, N.Y.)*, 19(5):223–236, November 2006.

[37] Shay Elmalem, Raja Giryes, and Emanuel Marom. Learned phase coded aperture for the benefit of depth of field extension. *Optics Express*, 26(12):15316–15331, June 2018.

[38] Jeffrey A. Fessler. Model-Based Image Reconstruction for MRI. *IEEE Signal Processing Magazine*, 27(4):81–89, July 2010.

[39] Jeffrey A. Fessler. Optimization Methods for Magnetic Resonance Image Reconstruction: Key Models and Optimization Algorithms. *IEEE Signal Processing Magazine*, 37(1):33–40, January 2020.

[40] Jeffrey A. Fessler, Sangwoo Lee, Valur T. Olafsson, Hugo R. Shi, and Douglas C. Noll. Toeplitz-based Iterative Image Reconstruction for MRI with Correction for Magnetic Field Inhomogeneity. *IEEE Transactions on Signal Processing*, 53(9):3393–3402, September 2005.

[41] Jeffrey A. Fessler and Bradley P. Sutton. Nonuniform fast Fourier transforms using min-max interpolation. *IEEE Transactions on Signal Processing*, 51(2):560–574, February 2003.

[42] William Fong and Eric Darve. The black-box fast multipole method. *Journal of Computational Physics*, 228(23):8712–8725, December 2009.

[43] Kirsten P.N. Forbes, James G. Pipe, C. Roger Bird, and Joseph E. Heiserman. PROPELLER MRI: Clinical testing of a novel technique for quantification and compensation of head motion. *Journal of Magnetic Resonance Imaging*, 14(3):215–222, 2001.

[44] Amanda K. Funai, Jeffrey A. Fessler, Desmond T. B. Yeo, Valur T. Olafsson, and Douglas C. Noll. Regularized Field Map Estimation in MRI. *IEEE Transactions on Medical Imaging*, 27(10):1484–1494, October 2008.

[45] Chaithya G R, Zaccharie Ramzi, and Philippe Ciuciu. Hybrid Learning of Non-Cartesian K-Space Trajectory and Mr Image Reconstruction Networks. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, March 2022.

[46] Yun Gao and Stanley J. Reeves. Optimal k-space sampling in MRSI for images with a limited region of support. *IEEE Transactions on Medical Imaging*, 19(12):1168–1178, December 2000.

[47] Sairam Geethanath and John Thomas Vaughan Jr. Accessible magnetic resonance imaging: A review. *Journal of Magnetic Resonance Imaging*, 49(7):e65–e77, 2019.

[48] Gary H. Glover and Christine S. Law. Spiral-in/out BOLD fMRI for increased SNR and reduced susceptibility artifacts. *Magnetic Resonance in Medicine*, 46(3):515–522, 2001.

[49] Karol Gregor and Yann LeCun. Learning Fast Approximations of Sparse Coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 399–406, Madison, WI, USA, June 2010. Omnipress.

[50] Mark A. Griswold, Peter M. Jakob, Robin M. Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magnetic Resonance in Medicine*, 47(6):1202–1210, 2002.

[51] Hongyi Gu, Burhaneddin Yaman, Steen Moeller, Jutta Ellermann, Kamil Ugurbil, and Mehmet Akçakaya. Revisiting l1-wavelet compressed-sensing MRI in the era of deep learning. *Proceedings of the National Academy of Sciences*, 119(33):e2201062119, 2022.

[52] Shouchang Guo, Jeffrey A. Fessler, and Douglas C. Noll. High-Resolution Oscillating Steady-State fMRI Using Patch-Tensor Low-Rank Reconstruction. *IEEE Transactions on Medical Imaging*, 39(12):4357–4368, December 2020.

[53] Shouchang Guo and Douglas C. Noll. Oscillating Steady-State Imaging (OSSI): A Novel Method for Functional MRI. *Magnetic Resonance in Medicine*, 84(2):698–712, 2020.

[54] Paul T. Gurney, Brian A. Hargreaves, and Dwight G. Nishimura. Design and analysis of a practical 3D cones trajectory. *Magnetic Resonance in Medicine*, 55(3):575–582, 2006.

[55] Baran Gözcü, Rabeeh Karimi Mahabadi, Yen-Huan Li, Efe Ilıcak, Tolga Çukur, Jonathan Scarlett, and Volkan Cevher. Learning-Based Compressive MRI. *IEEE Transactions on Medical Imaging*, 37(6):1394–1406, June 2018.

[56] Baran Gözcü, Thomas Sanchez, and Volkan Cevher. Rethinking Sampling in Parallel MRI: A Data-Driven Approach. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, September 2019.

[57] Justin P. Haldar. Low-Rank Modeling of Local k-Space Neighborhoods (LORAKS) for Constrained MRI. *IEEE Transactions on Medical Imaging*, 33(3):668–681, March 2014.

[58] Justin P. Haldar and Daeun Kim. OEDIPUS: An Experiment Design Framework for Sparsity-Constrained MRI. *IEEE Transactions on Medical Imaging*, 38(7):1545–1558, July 2019.

[59] C. L. G. Ham, J. M. L. Engels, G. T. van de Wiel, and A. Machielsen. Peripheral nerve stimulation during MRI: Effects of high gradient amplitudes and switching rates. *Journal of Magnetic Resonance Imaging*, 7(5):933–937, 1997.

[60] Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a Variational Network for Reconstruction of Accelerated MRI data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018.

[61] Sun Hao, Jeffrey A. Fessler, Douglas C. Noll, and Jon-Fredrik Nielsen. Joint Design of Excitation k-Space Trajectory and RF Pulse for Small-Tip 3D Tailored Excitation in MRI. *IEEE Transactions on Medical Imaging*, 35(2):468–479, February 2016.

[62] Robert A. Hedeen and William A Edelstein. Characterization and prediction of gradient acoustic noise in MR imagers. *Magnetic Resonance in Medicine*, 37(1):7–10, 1997.

[63] Karl-Heinz Herrmann, Martin Krämer, and Jürgen R. Reichenbach. Time Efficient 3D Radial UTE Sampling with Fully Automatic Delay Compensation on a Clinical 3T MR Scanner. *PLOS ONE*, 11(3):e0150371, March 2016.

[64] Magnus R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, November 1969.

[65] Are Hjorungnes and David Gesbert. Complex-Valued Matrix Differentiation: Techniques and Key Results. *IEEE Transactions on Signal Processing*, 55(6):2740–2746, June 2007.

[66] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. arXiv:2006.11239 [cs, stat].

[67] Richard D. Hoge, Remi K. S. Kwan, and G. Bruce Pike. Density compensation functions for spiral MRI. *Magnetic Resonance in Medicine*, 38(1):117–128, 1997.

[68] Alain Horé and Djemel Ziou. Image Quality Metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition (ICPR)*, pages 2366–2369, August 2010.

[69] Iris A. M. Huijben, Bastiaan S. Veeling, Kees Janse, Massimo Mischi, and Ruud J. G. van Sloun. Learning Sub-Sampling and Signal Recovery With Applications in Ultrasound Imaging. *IEEE Transactions on Medical Imaging*, 39(12):3955–3966, December 2020.

[70] Iris A.M. Huijben, Bastiaan S. Veeling, and Ruud J.G. van Sloun. Learning Sampling and Model-Based Signal Recovery for Compressed Sensing MRI. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8906–8910, May 2020.

[71] Aapo Hyvärinen. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[72] Mathews Jacob, Merry P. Mani, and Jong Chul Ye. Structured Low-Rank Algorithms: Theory, Magnetic Resonance Applications, and Links to Machine Learning. *IEEE Signal Processing Magazine*, 37(1):54–68, January 2020.

[73] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images. *NeuroImage*, 17(2):825–841, October 2002.

[74] Kyong Hwan Jin, Michael Unser, and Kwang Moo Yi. Self-Supervised Deep Active Accelerated MRI, January 2019. arXiv:1901.04547 [cs].

[75] Adam Johansson, James Balter, and Yue Cao. Rigid-body motion correction of the liver in image reconstruction for golden-angle stack-of-stars DCE MRI. *Magnetic Resonance in Medicine*, 79(3):1345–1353, 2018.

[76] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[77] Kevin M. Johnson. Hybrid radial-cones trajectory for accelerated MRI. *Magnetic Resonance in Medicine*, 77(3):1068–1081, 2017.

[78] Donghwan Kim and Jeffrey A. Fessler. Adaptive Restart of the Optimized Gradient Method for Convex Optimization. *Journal of Optimization Theory and Applications*, 178(1):240–263, July 2018.

[79] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.

[80] Florian Knoll, Christian Clason, Clemens Diwoky, and Rudolf Stollberger. Adapted random sampling patterns for accelerated MRI. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 24(1):43–50, February 2011.

[81] Joel H. Kramer, Dan Mungas, Bruce R. Reed, Margaret E. Wetzel, Molly M. Burnett, Bruce L. Miller, Michael W. Weiner, and Helena C. Chui. Longitudinal MRI and cognitive change in healthy elderly. *Neuropsychology*, 21:412–418, 2007.

[82] Ken Kreutz-Delgado. The Complex Gradient Operator and the CR-Calculus. June 2009.

[83] David J. Larkman and Rita G. Nunes. Parallel Magnetic Resonance Imaging. *Physics in Medicine and Biology*, 52(7):R15–R55, March 2007.

[84] Paul C Lauterbur. Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance. *Nature*, 242(5394):190–191, 1973.

[85] Carole Lazarus, Pierre Weiss, Nicolas Chauffert, Franck Mauconduit, Loubna El Gueddari, Christophe Destrieux, Ilyess Zemmoura, Alexandre Vignaud, and Philippe Ciuciu. SPARKLING: Variable-Density k-space Filling Curves for Accelerated T2*-weighted MRI. *Magnetic Resonance in Medicine*, 81(6):3643–3661, 2019.

[86] Carole Lazarus, Pierre Weiss, Loubna El Gueddari, Franck Mauconduit, Aurélien Massire, Mathilde Ripart, Alexandre Vignaud, and Philippe Ciuciu. 3D variable-density SPARKLING trajectories for high-resolution T2*-weighted magnetic resonance imaging. *NMR in Biomedicine*, 33(9):e4349, 2020.

[87] Jin Hyung Lee, Brian A. Hargreaves, Bob S. Hu, and Dwight G. Nishimura. Fast 3D Imaging Using Variable-density Spiral Trajectories with Applications to Limb Perfusion. *Magnetic Resonance in Medicine*, 50(6):1276–1285, 2003.

[88] Sil C. van de Leemput, Jonas Teuwen, Bram van Ginneken, and Rashindra Manniesing. MemCNN: A Python/PyTorch package for creating memory-efficient invertible neural networks. *Journal of Open Source Software*, 4(39):1576, July 2019.

[89] Evan Levine and Brian Hargreaves. On-the-Fly Adaptive k-Space Sampling for Linear MRI Reconstruction Using Moment-Based Spectral Analysis. *IEEE Transactions on Medical Imaging*, 37(2):557–567, February 2018.

[90] Andreas Lugmayr, Martin Danelljan, Radu Timofte, Namhyuk Ahn, Dongwoon Bai, Jie Cai, Yun Cao, Junyang Chen, Kaihua Cheng, SeYoung Chun, Wei Deng, Mostafa El-Khamy, Chiu Man Ho, Xiaozhong Ji, Amin Kheradmand, Gwantae Kim, Hanseok Ko, Kanghyu Lee, Jungwon Lee, Hao Li, Ziluan Liu, Zhi-Song Liu, Shuai Liu, Yunhua Lu, Zibo Meng, Pablo Navarrete Michelini, Christian Micheloni, Kalpesh Prajapati, Haoyu Ren, Yong Hyeok Seo, Wan-Chi Siu, Kyung-Ah Sohn, Ying Tai, Rao Muhammad Umer, Shuangquan Wang, Huibing Wang, Timothy Haoning Wu, Haoning Wu, Biao Yang, Fuzhi Yang, Jaejun Yoo, Tongtong Zhao, Yuanbo Zhou, Haijie Zhuo, Ziyao Zong, and Xueyi Zou. NTIRE 2020 Challenge on Real-World Image Super-Resolution: Methods and Results. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2058–2076, June 2020.

[91] Michael Lustig, David L. Donoho, Juan M. Santos, and John M. Pauly. Compressed Sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72–82, March 2008.

[92] Michael Lustig, Seung-Jean Kim, and John M. Pauly. A Fast Method for Designing Time-Optimal Gradient Waveforms for Arbitrary k-Space Trajectories. *IEEE transactions on medical imaging*, 27(6):866–873, June 2008.

[93] Dan Ma, Vikas Gulani, Nicole Seiberlich, Kecheng Liu, Jeffrey L Sunshine, Jeffrey L Duerk, and Mark A Griswold. Magnetic resonance fingerprinting. *Nature*, 495(7440):187, 2013.

[94] Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, March 2019.

[95] Oliver Maier, Steven Hubert Baete, Alexander Fyrdahl, Kerstin Hammernik, Seb Harrevelt, Lars Kasper, Agah Karakuzu, Michael Loecher, Franz Patzig, Ye Tian, Ke Wang, Daniel Gallichan, Martin Uecker, and Florian Knoll. CG-SENSE revisited: Results from the first ISMRM reproducibility challenge. *Magnetic Resonance in Medicine*, 85(4):1821–1839, 2021.

[96] Weihua Mao, Michael B. Smith, and Christopher M. Collins. Exploring the limits of RF shimming for high-field MRI of the human head. *Magnetic Resonance in Medicine*, 56(4):918–922, 2006.

[97] Morteza Mardani, Enhao Gong, Joseph Y. Cheng, Shreyas S. Vasanawala, Greg Zaharchuk, Lei Xing, and John M. Pauly. Deep Generative Adversarial Neural Networks for Compressive Sensing MRI. *IEEE Transactions on Medical Imaging*, 38(1):167–179, January 2019.

[98] Jonathan B. Martin, Sai Abitha Srinivas, Christopher E. Vaughn, Heng Sun, Mark A. Griswold, and William A. Grissom. Selective excitation localized by the Bloch–Siegert shift and a B1+ gradient. *Magnetic Resonance in Medicine*, 88(3):1081–1097, 2022.

[99] Alex McManus, Stephen Becker, and Nicholas Dwork. The Dependence of Parallel Imaging with Linear Predictability on the Undersampling Direction, January 2023. arXiv:2301.07256 [eess].

[100] Matthew J. Muckley, Bruno Riemenschneider, Alireza Radmanesh, Sunwoo Kim, Geunu Jeong, Jingyu Ko, Yohan Jun, Hyungseob Shin, Dosik Hwang, Mahmoud Mostapha, Simon Arberet, Dominik Nickel, Zaccharie Ramzi, Philippe Ciuciu, Jean-Luc Starck, Jonas Teuwen, Dimitrios Karkalousos, Chaoping Zhang, Anuroop Sriram, Zhengnan Huang, Nafissa Yakubova, Yvonne Lui, and Florian Knoll. Results of the 2020 fastMRI Challenge for Machine Learning MR Image Reconstruction. *IEEE Transactions on Medical Imaging*, 40(9):2306–2317, September 2021.

[101] Matthew J. Muckley, Ruben Stern, Tullie Murrell, and Florian Knoll. TorchKbNufft: A high-level, hardware-agnostic non-uniform fast Fourier transform. In *ISMRM workshop on data sampling & image reconstruction*, 2020.

[102] Seungjun Nah, Sanghyun Son, Radu Timofte, Kyoung Mu Lee, Yu Tseng, Yu-Syuan Xu, Cheng-Ming Chiang, Yi-Min Tsai, Stephan Brehm, Sebastian Scherer, Dejia Xu, Yihao Chu, Qingyan Sun, Jiaqin Jiang, Lunhao Duan, Jian Yao, Kuldeep Purpohit, Maitreya Suin, A.N. Rajagopalan, Yuichi Ito, P.S. Hrishikesh, Densen Puthussery, K.A. Akhil, C.V. Jiji, Guisik Kim, P.L. Deepa, Zhiwei Xiong, Jie Huang, Dong Liu, Sangmin Kim, Hyungjoon Nam, Jisu Kim, Jechang Jeong, Shihua Huang, Yuchen Fan, Jiahui Yu, Haichao Yu, Thomas S. Huang, Ya Zhou, Xin Li, Sen Liu, Zhibo Chen, Saikat Dutta, Sourya Dipta Das, Shivam Garg, Daniel Sprague, Bhrij Patel, and Thomas Huck. NTIRE 2020 Challenge on Image and Video Deblurring. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1662–1675, June 2020.

[103] Jon-Fredrik Nielsen and Douglas C. Noll. TOPPE: A Framework for Rapid Prototyping of MR Pulse Sequences. *Magnetic Resonance in Medicine*, 79(6):3128–3134, 2018.

[104] Douglas C. Noll, Jeffrey A. Fessler, and Bradley P. Sutton. Conjugate phase MRI reconstruction with spatially variant sample density correction. *IEEE Transactions on Medical Imaging*, 24(3):325–336, March 2005.

[105] Douglas C. Noll, Dwight G. Nishimura, and Albert Macovski. Homodyne detection in magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, 10(2):154–163, June 1991.

[106] Douglas C. Noll, John M. Pauly, Craig H. Meyer, Dwight G. Nishimura, and Albert Macovskj. Deblurring for non-2D fourier transform magnetic resonance imaging. *Magnetic Resonance in Medicine*, 25(2):319–333, 1992.

[107] Luis Pineda, Sumana Basu, Adriana Romero, Roberto Calandra, and Michal Drozdzal. Active MR k-space Sampling with Reinforcement Learning. In *Medical Image Computing*

and Computer Assisted Intervention – MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II, pages 23–33, Berlin, Heidelberg, October 2020. Springer-Verlag.

[108] James G. Pipe. Motion correction with PROPELLER MRI: Application to head motion and free-breathing cardiac imaging. *Magnetic Resonance in Medicine*, 42(5):963–969, 1999.

[109] Klaas P. Pruessmann, Markus Weiger, Peter Börnert, and Peter Boesiger. Advances in sensitivity encoding with arbitrary k-space trajectories. *Magnetic Resonance in Medicine*, 46(4):638–651, 2001.

[110] Klaas P. Pruessmann, Markus Weiger, Markus B. Scheidegger, and Peter Boesiger. SENSE: Sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine*, 42(5):952–962, 1999.

[111] Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset Selection by Pareto Optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1774–1782, Cambridge, MA, USA, December 2015.

[112] Sathish Ramani and Jeffrey A. Fessler. Parallel MR Image Reconstruction Using Augmented Lagrangian Methods. *IEEE Transactions on Medical Imaging*, 30(3):694–706, March 2011.

[113] Saiprasad Ravishankar, Raj Rao Nadakuditi, and Jeffrey A. Fessler. Efficient Sum of Outer Products Dictionary Learning (SOUP-DIL) and Its Application to Inverse Problems. *IEEE Transactions on Computational Imaging*, 3(4):694–709, December 2017.

[114] Reinhold Remmert. *Theory of Complex Functions*. Springer Science & Business Media, 1991.

[115] Christoph A. Rettenmeier, Danilo Maziero, and V. Andrew Stenger. Three dimensional radial echo planar imaging for functional MRI. *Magnetic Resonance in Medicine*, 87(1):193–206, 2022.

[116] Ryan K. Robison, Zhiqiang Li, Dinghui Wang, Melvyn B. Ooi, and James G. Pipe. Correction of B0 eddy current effects in spiral MRI. *Magnetic Resonance in Medicine*, 81(4):2501–2513, 2019.

[117] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015.

[118] Sebastian Sabat, Roberto Mir, Marcelo Guarini, Andres Guesalaga, and Pablo Irarrazaval. Three Dimensional K-space Trajectory Design Using Genetic Algorithms. *Magnetic Resonance Imaging*, 21(7):755–764, September 2003.

[119] Thomas Sanchez, Baran Gözcü, Ruud B. van Heeswijk, Armin Eftekhari, Efe Ilıcak, Tolga Çukur, and Volkan Cevher. Scalable Learning-Based Sampling Optimization for Compressive Dynamic MRI. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8584–8588, May 2020.

[120] Jo Schlemper, Jose Caballero, Joseph V. Hajnal, Anthony N. Price, and Daniel Rueckert. A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction. *IEEE Transactions on Medical Imaging*, 37(2):491–503, February 2018.

[121] Jo Schlemper, Chen Qin, Jinming Duan, Ronald M. Summers, and Kerstin Hammernik. Sigma-Net: Ensembled Iterative Deep Neural Networks for Accelerated Parallel MR Image Reconstruction. *arXiv:1912.05480 [cs, eess, stat]*, December 2019.

[122] Rainer Schneider, Dieter Ritter, Jens Haueisen, and Josef Pfeuffer. B0-informed variable density trajectory design for enhanced correction of off-resonance effects in parallel transmission. *Magnetic Resonance in Medicine*, 71(4):1381–1393, 2014.

[123] Rolf F. Schulte and Ralph Noeske. Peripheral nerve stimulation-optimal gradient waveform design. *Magnetic Resonance in Medicine*, 74(2):518–522, 2015.

[124] Evan Scope Crafts, Hengfa Lu, Huihui Ye, Lawrence L. Wald, and Bo Zhao. An efficient approach to optimal experimental design for magnetic resonance fingerprinting with B-splines. *Magnetic Resonance in Medicine*, 88(1):239–253, 2022.

[125] Matthias Seeger, Hannes Nickisch, Rolf Pohmann, and Bernhard Schölkopf. Optimization of K-space Trajectories for Compressed Sensing by Bayesian Experimental Design. *Magnetic Resonance in Medicine*, 63(1):116–126, 2010.

[126] Kawin Setsompop, Borjan A. Gagoski, Jonathan R. Polimeni, Thomas Witzel, Van J. Wedeen, and Lawrence L. Wald. Blipped-controlled Aliasing in Parallel Imaging for Simultaneous Multislice Rcho Planar Imaging with Reduced G-factor Penalty. *Magnetic Resonance in Medicine*, 67(5):1210–1224, May 2012.

[127] Ferdia Sherry, Martin Benning, Juan Carlos De los Reyes, Martin J. Graves, Georg Maierhofer, Guy Williams, Carola-Bibiane Schönlieb, and Matthias J. Ehrhardt. Learning the Sampling Pattern for MRI. *IEEE Transactions on Medical Imaging*, 39(12):4310–4321, December 2020.

[128] Jonathan Richard Shewchuk and others. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[129] John G. Sled and G.Bruce Pike. Standing-wave and RF penetration artifacts caused by elliptic geometry: an electrodynamic analysis of MRI. *IEEE Transactions on Medical Imaging*, 17(4):653–662, August 1998.

[130] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution, October 2020. arXiv:1907.05600 [cs, stat].

[131] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021. arXiv:2011.13456 [cs, stat].

[132] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International conference on learning representations*, 2021.

[133] Roberto Souza, Oeslle Lucena, Julia Garrafa, David Gobbi, Marina Saluzzi, Simone Appenzeller, Letícia Rittner, Richard Frayne, and Roberto Lotufo. An open, multi-vendor, multi-field-strength brain MR dataset and analysis of publicly available skull stripping methods agreement. *NeuroImage*, 170:482–494, April 2018.

[134] Bradley P. Sutton, Douglas C. Noll, and Jeffrey A. Fessler. Fast, Iterative Image Reconstruction for MRI in the Presence of Field Inhomogeneities. *IEEE Transactions on Medical Imaging*, 22(2):178–188, February 2003.

[135] Ek T. Tan, Seung-Kyun Lee, Paul T. Weavers, Dominic Graziani, Joseph E. Piel, Yunhong Shu, John Huston III, Matt A. Bernstein, and Thomas K.F. Foo. High slew-rate head-only gradient for improving distortion in echo planar imaging: Preliminary experience. *Journal of Magnetic Resonance Imaging*, 44(3):653–664, 2016.

[136] Alberto Tannús and Michael Garwood. Adiabatic pulses. *NMR in Biomedicine*, 10(8):423–434, 1997.

[137] Kerem C. Tezcan, Neerav Karani, Christian F. Baumgartner, and Ender Konukoglu. Sampling Possible Reconstructions of Undersampled Acquisitions in MR Imaging With a Deep Learned Prior. *IEEE Transactions on Medical Imaging*, 41(7):1885–1896, July 2022.

[138] D. Tomasi, R.f. Xavier, B. Foerster, H. Panepucci, A. Tannús, and E.l. Vidoto. Asymmetrical gradient coil for head imaging. *Magnetic Resonance in Medicine*, 48(4):707–714, 2002.

[139] Robert Turner. Gradient coil design: A review of methods. *Magnetic Resonance Imaging*, 11(7):903–920, January 1993.

[140] Martin Uecker, Thorsten Hohage, Kai Tobias Block, and Jens Frahm. Image reconstruction by regularized nonlinear inversion—Joint estimation of coil sensitivities and image content. *Magnetic Resonance in Medicine*, 60(3):674–682, 2008.

[141] Martin Uecker, Peng Lai, Mark J Murphy, Patrick Virtue, Michael Elad, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. *Magnetic resonance in medicine*, 71(3):990–1001, 2014.

[142] Signe J. Vannesjo, Maximilan Haeberlin, Lars Kasper, Matteo Pavan, Bertram J. Wilm, Christoph Barmet, and Klaas P. Pruessmann. Gradient System Characterization by Impulse Response Measurements with a Dynamic Field Camera. *Magnetic Resonance in Medicine*, 69(2):583–593, 2013.

[143] Markus von Kienlin and Raymond Mejia. Spectral localization with optimal pointspread function. *Journal of Magnetic Resonance (1969)*, 94(2):268–287, September 1991.

[144] Guanhua Wang and Jeffrey A. Fessler. Efficient Approximation of Jacobian Matrices Involving a Non-Uniform Fast Fourier Transform (NUFFT). *IEEE Transactions on Computational Imaging*, 9:43–54, 2023.

[145] Guanhua Wang, Matteo Alessandro Francavilla, Ma Sen, Jeffrey Kaditz, and Thomas Witzel. Delta Scan: Reconstruction and Protocol Design for Ultra-fast Longitudinal MRI. In *Proc. Intl. Soc. Magn. Reson. Med. (ISMRM)*, page 3506, 2022.

[146] Guanhua Wang, Tianrui Luo, Jon-Fredrik Nielsen, Douglas C. Noll, and Jeffrey A. Fessler. B-Spline Parameterized Joint Optimization of Reconstruction and K-Space Trajectories (BJORK) for Accelerated 2D MRI. *IEEE Transactions on Medical Imaging*, 41(9):2318–2330, September 2022.

[147] Guanhua Wang, Jon-Fredrik Nielsen, Jeffrey A. Fessler, and Douglas C. Noll. Stochastic optimization of three-dimensional non-Cartesian sampling trajectory. *Magnetic Resonance in Medicine*, 90(2):417–431, 2023.

[148] Guanhua Wang, Douglas C. Noll, and Jeffrey A. Fessler. Adaptive Sampling for Linear Sensing Systems via Langevin Dynamics, February 2023. arXiv:2302.13468 [eess].

[149] Guanhua Wang, Neel Shah, Keyue Zhu, Douglas C. Noll, and Jeffrey A. Fessler. MIR-Torch: A PyTorch-powered Differentiable Toolbox for Fast Image Reconstruction and Scan Protocol Optimization. In *Proc. Intl. Soc. Magn. Reson. Med. (ISMRM)*, page 4982, 2022.

[150] Zhangwei Wang, James C. Lin, Weihua Mao, Wanzhan Liu, Michael B. Smith, and Christopher M. Collins. SAR and temperature: Simulations and comparison to regulatory limits for MRI. *Journal of Magnetic Resonance Imaging*, 26(2):437–441, 2007.

[151] Zhongmin Wang and Gonzalo R. Arce. Variable Density Compressed Image Sampling. *IEEE Transactions on Image Processing*, 19(1):264–270, January 2010.

[152] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[153] Paul T. Weavers, Shengzhen Tao, Joshua D. Trzasko, Louis M. Frigo, Yunhong Shu, Matthew A. Frick, Seung-Kyun Lee, Thomas K-F Foo, and Matt A. Bernstein. B0 concomitant field compensation for MRI systems employing asymmetric transverse gradient coils. *Magnetic Resonance in Medicine*, 79(3):1538–1544, 2018.

[154] Tomer Weiss, Ortal Senouf, Sanketh Vedula, Oleg Michailovich, Michael Zibulevsky, and Alex Bronstein. PILOT: Physics-Informed Learned Optimized Trajectories for Accelerated MRI. *arXiv:1909.05773 [physics]*, April 2021. arXiv: 1909.05773.

[155] Lior Weizman, Yonina C. Eldar, and Dafna Ben Bashat. Compressed sensing for longitudinal MRI: An adaptive-weighted approach. *Medical Physics*, 42(9):5195–5208, 2015.

[156] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 681–688, Madison, WI, USA, June 2011. Omnipress.

[157] Goetz H. Welsch, Klaus Scheffler, Tallal C. Mamisch, Timothy Hughes, Steven Millington, Michael Deimling, and Siegfried Trattnig. Rapid estimation of cartilage T2 based on double echo at steady state (DESS) with 3 Tesla. *Magnetic Resonance in Medicine*, 62(2):544–549, 2009.

[158] Jennifer L. Whitwell. Longitudinal imaging: change and causality. *Current Opinion in Neurology*, 21(4):410, August 2008.

[159] Stefanie Winkelmann, Tobias Schaeffter, Thomas Koehler, Holger Eggers, and Olaf Doessel. An Optimal Radial Profile Order Based on the Golden Ratio for Time-Resolved MRI. *IEEE Transactions on Medical Imaging*, 26(1):68–76, January 2007.

[160] Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson Flow Generative Models, October 2022. arXiv:2209.11178 [cs].

[161] Qing X. Yang, Weihua Mao, Jinghua Wang, Michael B. Smith, Hao Lei, Xiaoliang Zhang, Kamil Ugurbil, and Wei Chen. Manipulation of image intensity distribution at 7.0 T: Passive RF shimming and focusing with dielectric materials. *Journal of Magnetic Resonance Imaging*, 24(1):197–202, 2006.

[162] Qing X. Yang, Jianli Wang, Jinghua Wang, Christopher M. Collins, Chunsheng Wang, and Michael B. Smith. Reducing SAR and enhancing cerebral signal-to-noise ratio with high permittivity padding at 3 T. *Magnetic Resonance in Medicine*, 65(2):358–362, 2011.

[163] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for Compressive Sensing MRI. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 10–18, Red Hook, NY, USA, December 2016. Curran Associates Inc.

[164] Desmond T. B. Yeo, Jeffrey A. Fessler, and Boklye Kim. Concurrent correction of geometric distortion and motion using the map-slice-to-volume method in echo-planar imaging. *Magnetic Resonance Imaging*, 26(5):703–714, June 2008.

[165] Mi Hye Yu, Jeong Min Lee, Jeong-Hee Yoon, Berthold Kiefer, Joon Koo Han, and Byung-Ihn Choi. Clinical application of controlled aliasing in parallel imaging results in a higher acceleration (CAIPIRINHA)-volumetric interpolated breathhold (VIBE) sequence for gadoxetic acid-enhanced liver MR imaging. *Journal of Magnetic Resonance Imaging*, 38(5):1020–1026, 2013.

[166] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep Iterative Down-Up CNN for Image Denoising. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2095–2103, June 2019.

[167] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: An Open Dataset and Benchmarks for Accelerated MRI, December 2019. arXiv:1811.08839 [physics, stat].

[168] Mona Zehni, Laurène Donati, Emmanuel Soubies, Zhizhen Zhao, and Michael Unser. Joint Angular Refinement and Reconstruction for Single-Particle Cryo-EM. *IEEE Transactions on Image Processing*, 29:6151–6163, 2020.

[169] David Y. Zeng, Christopher Sandino, Dwight G. Nishimura, Shreyas S. Vasanawala, and Joseph Y. Cheng. Reinforcement learning for online undersampling pattern optimization. In *Proc. Intl. Soc. Magn. Reson. Med. (ISMRM)*, page 1092, 2019.

[170] Zhenghui Zhang, Chun-Yu Yip, William Grissom, Douglas C. Noll, Fernando E. Boada, and V. Andrew Stenger. Reduction of transmitter B1 inhomogeneity with transmit SENSE slice-select pulses. *Magnetic Resonance in Medicine*, 57(5):842–847, 2007.

[171] Zizhao Zhang, Adriana Romero, Matthew J. Muckley, Pascal Vincent, Lin Yang, and Michal Drozdzal. Reducing Uncertainty in Undersampled MRI Reconstruction With Active Acquisition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2049–2053. IEEE Computer Society, June 2019.

[172] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss Functions for Image Restoration With Neural Networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, March 2017.

[173] Ziwu Zhou, Fei Han, Lirong Yan, Danny J.J. Wang, and Peng Hu. Golden-ratio rotated stack-of-stars acquisition for improved volumetric MRI. *Magnetic Resonance in Medicine*, 78(6):2290–2298, 2017.

[174] Jiancheng Zhuang, Jan Hrabe, Alayar Kangarlu, Dongrong Xu, Ravi Bansal, Craig A. Branch, and Bradley S. Peterson. Correction of Eddy-Current Distortions in Diffusion Tensor Images Using the Known Directions and Strengths of Diffusion Gradients. *Journal of Magnetic Resonance Imaging*, 24(5):1188–1193, November 2006.

[175] Marcelo V. W. Zibetti, Gabor T. Herman, and Ravinder R. Regatte. Fast data-driven learning of parallel MRI sampling patterns for large scale problems. *Scientific Reports*, 11(1):19312, September 2021.