DEVELOPMENT OF ADVANCED GENERATIVE PRIORS FOR MRI RECONSTRUCTION

Dissertation
for the award of the degree
“Doctor rerum naturalium"
of the Georg-August-Universität Göttingen

within the doctoral program
Computer Science
of the Georg-August University School of Science (GAUSS)

submitted by

GUANXIONG LUO

from Pingjiang, China
in Göttingen 2023

THESIS COMMITTEE

FIRST REFEREE AND SUPERVISOR

Prof. Dr. Martin Uecker
Institute of Biomedical Imaging
Technische Universität Graz

Prof. Dr. Markus Haltmeier
Applied Mathematics
Universität Innsbruck

Prof. Dr. Philipp Wieder
Institut für Informatik
Georg-August-Universität Göttingen

MEMBERS OF THE EXAMINIATION BOARD

Prof. Dr. Martin Uecker

Prof. Dr. Markus Haltmeier

Prof. Dr. Philipp Wieder

Prof. Dr. Christoph Lehrenfeld
Institut für Numerische und Angewandte Mathematik
Georg-August-Universität Göttingen

Prof. Dr. Axel Munk
Institut für Mathematische Stochastik
Georg-August-Universität Göttingen

Prof. Dr. David Russell Luke
Institut für Numerische und Angewandte Mathematik
Georg-August-Universität Göttingen

Date of oral examiniation: xx. xx. 2023

To my family

ACKNOWLEDGEMENTS

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

1

GENERAL INTRODUCTION

1.1 MOTIVATION OF STUDY

Fast magnetic resonance imaging (MRI) has been a crucial area of research since the inception of the first magnetic resonance (MR) scanner. Compared to other imaging modalities such as X-ray or ultrasound, MRI does not require ionizing and can measure many physical parameters. But it is inherently slower due to its fundamental imaging principles. The acquisition of raw data involves many repetitions of excite-and-sample experiments. Over the years, these repetitive experiments have been gradually sped up through technical innovations in the fast sequence design, receiver coils, and gradient filed systems. One major development was parallel imaging. It exploits the local coil profiles, enable the acquisition of fewer samples than prescribed by the Nyquist theorem and leads to accelerated imaging. The technique compressed sensing (CS)[1, 2] further reduced the samples required in k-space by exploiting the sparsity and incoherent sampling. CS could fail as artifacts may arise from inaccurate sparse modeling assumptions. In recent years, CS has been extended to incorporate deep neural networks to harness the prior knowledge from historical data. These techniques have proven to be a compelling and practical approach to address the challenges of CS.

The conventional way to represent the prior information used in CS, such as an image is sparsely represented in a certain wavelet domain, is described in mathematical form with few parameters, and can be studied analytically and exploited to given algorithmic solutions. The fast growth of machine learning (ML), fueled by the explosion of computation power such as graphics processing unit (GPU), makes it feasible to harness prior knowledge from the vast amounts of data. The reinvention of convolutional neural networks (CNNs) enables the effective modelling of high-dimensional data, even in the case of image that ensembles megabytes of constants to capture the intricate structures and variations. Consequently, many deep learning (DL)-based methods[3, 4] for MRI image reconstruction have emerged.

Amid these advancements, certain challenges remain in terms of model interpretability, data availability, and generalizability. Ensuring robustness against variations in acquisition protocols, hardware, and imaging conditions is essential for reliable MR clinical use of these methods. A significant revelation, as highlighted in Refs. [5, 6], underscores a critical aspect: DL typically yields unstable methods for image reconstruction. The instabilities usually occur in several forms:

1) uncertainty, undetectable perturbations, both in the image and sampling domain, may result in severe artifacts in the reconstruction; 2) a small structural change, for example, a tumor, may not be captured in the reconstructed image; 3) learned prior knowledge might lead to false structures being hallucinated in the reconstructed image. Apart from these aspects, many of these methods are designed and evaluated within a singular dataset, potentially leading to biased models due to overfitting to specific dataset characteristics. Besides the theoretical considerations, the deployment of ML models in production systems can present a number of issues and concerns across stages of the ML deployment workflow. Therefore, despite the exciting advancements, the domain of CS-MRI using DL continues to grapple with these critical challenges.

1.2 STRUCTURE OF THESIS

This thesis delves into three aspects of development of generative models, a subset of ML models, for CS-MRI parallel imaging reconstruction techniques.

Chapter 3 attempts to address the uncertainty introduced by the missing k-space data points due to undersampling. This is accomplished through a Bayesian imaging framework employing diffusion models. The posterior distribution of images, given measured k-space data, is sampled using k-space using the Markov chain Monte Carlo (MCMC) techniques. In addition to the maximum a posterior (MAP) estimate for the image, the minimum mean square error (MMSE) estimate and uncertainty maps can also be computed from those drawn samples. This chapter investigates various aspects, including the interpretation of image uncertainty, the impact of noise scales in training diffusion models, the optimization of MCMC sampling with a burn-in phase, and the transferability of learned information. In essence, a reconstruction framework is proposed that integrates diffusion processes, advanced generative models, and Markov chains.

Chapter 4 presents a workflow to construct generic and robust generative image priors from magnitude-only images. The priors can then be used for regularization in reconstruction to improve image quality. This workflow begins with the preparation of training datasets from magnitude-only images. This dataset is augmented with phase information and used to train generative priors of complex images. Finally, trained priors are evaluated using both linear and nonlinear reconstruction with various undersampling schemes. The results of our experiments demonstrate that priors trained on complex images outperform priors trained only on magnitude images. Additionally, a prior trained on a larger dataset exhibits higher robustness. These findings stress the importance of incorporating phase information and leveraging large datasets to raise the performance and reliability of the generative priors for MRI reconstruction. Phase augmentation makes it possible to use existing image databases for training.

Chapter 5 delves into practical aspects relevant for the entire development life cycle of generative priors, from the experimental stage to deployment stage. A Python package called "spreco" is introduced. Noteworthy features of spreco include distributed training capabilities, interruptible training mechanisms, efficient data loading, customizable model structures, and seamless deployment.

2

BASICS OF MR IMAGING AND MACHINE LEARNING

2.1 BASIC PRINCIPLES OF MAGNETIC RESONANCE IMAGING

We are able to look at MRI through the lens of energy transfer in its most basic form. At the beginning of a measurement, a radiofrequency (RF) pulse of the correct frequency is applied to the object, and the carried energy will be absorbed by some proton spins inside the object. After the RF pulse, the absorbed energy is reemitted in the form of signals that carry information about those spins and will be detected and processed to create an MRI image for the object being examined. The formation of a high-quality image requires multiple measurements, and the general description of this process will be provided below.

2.1.1 *Protons, Net Magnetization and Relaxation Processes*

Magnetic resonance is the phenomena that describes the interaction between an applied magnetic field and a nucleus that possesses a spin. Spin is a fundamental quantity carried by elementary particles. Nuclear spins, as typically used in conventional MRI, show a magnetic dipole moment, just like a bar magnet in classical electrodynamics. In a strong external field the nuclei can interact with a weak oscillating magnetic field, e.g. RF pulse, and respond by producing an electromagnetic signal with a frequency that characterizes the magnetic field at the nucleus. In most medical applications, hydrogen nuclei, which consist solely of a proton, that are in tissues create a signal that is processed to form an image of the body in terms of the density of those nuclei in a specific region.

PROTONS Atoms consist of three fundamental particles: protons, which possess a positive charge; neutrons, which have no charge; and electrons, which have a negative charge. The protons and neutrons are located in the nucleus or core of an atom, whereas the electrons are located in shells or orbitals surrounding the nucleus.

A third property of the nucleus is spin or intrinsic spin angular momentum. Although not an accurate physical description, this can be pictured in analogy to a classical particle rotating constantly around an axis at a constant rate or velocity. A limited number of values for the spin are found in nature; this is, the spin, I, is quantized to certain discrete values. These values depend on the atomic number and atomic weight of the particular nucleus. There are three groups of

values for I: zero, half-integral values, an integral values. A nucleus has no spin (I=0) if it has an even number atomic weight and an even atomic number. Such a nucleus does not interact with an external magnetic field and cannot be studied using conventional MR. The ^{1}H nucleus, consisting of a single proton, is a natural choice for probing the body using MR techniques for several reasons. It has a spin of $\frac{1}{2}$ and is the most abundant isotope for hydrogen. The human body is composed of tissues that contain primarily water and fat, both of which contain hydrogen.

A rigorous mathematical description of a nucleus with spin and its interactions requires the use of quantum mechanical principles, but most of MR in medical practice can be described using the concepts of classical mechanics, particularly in describing the actions of a nucleus with spin. The subsequent discussions of MR phenomena use a classical approach. In addition, although the concepts of resonance absorption and relaxation apply to all nuclei with spin, the descriptions are focused on ^{1}H since most imaging experiments visualize the ^{1}H nucleus.

In general, MR measurements are made on collections of similar spins rather than on an individual spin. It is useful to consider such a collection both as individual spins acting independently (a "microscopic" picture) and as single entity (a "macroscopic" picture). For many concepts, the two pictures provide equivalent results, even though the microscopic picture is more complete. Conversion between the two pictures requires a quantum statistical description. Though necessary for a complete understanding of MR phenomena, the nature of this conversion is beyond the scope of this thesis. In most cases, the macroscopic picture is sufficient for an adequate description.

STATIC MAGNETIC FIELD Consider an arbitrary volume of tissue containing hydrogen atoms (protons). Each proton has a spin vector of equal magnitude. However, the spin vectors for the entire collection of protons within the tissue are randomly oriented in all directions. Performing a vector addition of these spin vectors produces a zero-sum; that is, no net magnetization is observed in the tissue. If the tissue is placed inside a magnetic field $\mathbf{B}_0$, the individual protons begin to rotate perpendicular to, or precess about, the magnetic field. The protons are tilted slightly away from the axis of the magnetic field, but the axis of rotation is parallel to $\mathbf{B}_0$. This precession is at a constant rate and occurs because of the interaction of the magnetic field with the spinning positive charge of the nucleus. By convention, $\mathbf{B}_0$ and the axis of the precession are defined to be oriented in the z direction of a Cartesian coordinate system. The motion of each proton can be described by a unique set of coordinates perpendicular (x and y) and parallel (z) to $\mathbf{B}_0$. The perpendicular, or transverse, coordinate are nonzero and vary with time as the proton precesses, but the z coordinate is constant with time. The rate

or frequency of precession is proportional to the strength of the magnetic field and is expressed by Equation 2.1, the Larmor equation:

$$\omega_0 = \gamma \mathbf{B}_0 / 2\pi \tag{2.1}$$

where ω_0 is the Larmor frequency in megahertz (MHz), $\mathbf{B}_0$ is the magnetic field strength in Tesla (T) that the proton experiences, and γ is a constant for each nucleus in $1/s \cdot T$, known as the gyromagnetic ratio. Values for γ and ω can vary from nuclei to nuclei.

NET MAGNETIZATION If a vector addition is performed, as before, for the spin vectors inside the magnetic field, the results will be slightly different from the sum without the field. In the direction perpendicular to $\mathbf{B}_0$, the spin orientations are still randomly distributed just as they were outside the magnetic field, in spite of the time-varying nature of each transverse component. There is still no net magnetization perpendicular to $\mathbf{B}_0$. However, in the direction parallel to the magnetic field, there is a different result. Because there is an orientation to the precessional axis of the proton that is constant with time, there is a constant, nonzero interaction or coupling between the proton and $\mathbf{B}_0$. This is known as the Zeeman interaction. This coupling causes a difference in energy between protons aligned parallel or along $\mathbf{B}_0$ and protons aligned antiparallel or against $\mathbf{B}_0$. This energy difference ΔE is proportional to $\mathbf{B}_0$.

The result of the Zeeman interaction is that spins in the two orientations, parallel (also known as spin up) and antiparallel (spin down), have different energies. The orientation that is parallel to $\mathbf{B}_0$ is of lower energy than the antiparallel orientation. For a collection of protons, more will be oriented to $\mathbf{B}_0$ than will be oriented antiparallel; that is, there is an induced polarization of the spin orientation by the magnetic field. The expected number of protons in each energy level is governed by a distribution known as the Boltzmann distribution:

$$N_{\text{upper}} / N_{\text{lower}} = \exp(-\Delta E / kT)\,, \tag{2.2}$$

where N_{upper} and N_{lower} are the number of protons in the upper and lower energy levels, respectively, and k is Boltzmann's constant. Since the separation between the energy levels ΔE depends on the field strength $\mathbf{B}_0$, the exact number of spins in each level also depends on the $\mathbf{B}_0$ and increases with the increasing $\mathbf{B}_0$. For a collection of protons at body temperature (310K) at 1.5T, there will typically be an excess of ~1:10^6 protons in the lower level of the approximately 10^{25} protons within the tissue. This unequal number of protons in each energy level means that vector sum of spin will be nonzero and will point parallel to the magnetic field. In other words, the tissue will become polarized or magnetized in the presence of $\mathbf{B}_0$ with a value $\mathbf{M}_0$, known as magnetization. The orientation of this

net magnetization will be in the same direction as $\mathbf{B}_0$ and will be constant with respect to time. For tissue in the body, the magnitude of $\mathbf{M}_0$ is proportional to $\mathbf{B}_0$:

$$\mathbf{M}_0 = \chi \mathbf{B}_0 \,, \tag{2.3}$$

where χ is known as the bulk magnetic susceptibility or simply the magnetic susceptibility. This arrangement with $\mathbf{M}_0$ aligned along the magnetic field with no transverse component is the normal, or equilibrium, configuration for the protons. This configuration of spins has the lowest energy and is the arrangement to which the protons will naturally try to return following any perturbations such as energy absorption. This induced magnetization, $\mathbf{M}_0$, is the source of signal for all the MR imaging experiments. Consequently, all other things being equal, the greater the field strength, the greater the value of $\mathbf{M}_0$ and the greater the potential MR signal.

RF PULSE AND SLICE SELECTION To manipulate the magnetization, $\mathbf{M}_0$, a short RF pulse, which generates a magnetic field $\mathbf{B}_1$ in the transverse direction and rotates at the resonant frequency of the magnetization, is applied. Classically, $\mathbf{B}_1$ induces a torque on the magnetization, causing $\mathbf{M}_0$ to rotate away from its equilibrium. A common excitation RF pulse used to tip the net magnetization $\mathbf{M}_0$ to transverse plane has a flip angle. A pulse with 90° rotate the magnetization to lie in the transverse plane completely with no $\mathbf{M}_z$ component.

FID SIGNAL Given a transverse component to the magnetization, detection of the magnetization becomes possible via Faraday's law of induction. This detection occurs because the precessing magnetization causes a change in flux Φ in the coil, thereby inducing a small electromotive force (emf). The resulting time signal is commonly called a free induction decay (FID) represents and creates the basic MR signal that is recorded.

Following the excitation of RF pulse, the transverse component $\mathbf{M}_{xy}$ of magnetization decays away while the longitudinal component $\mathbf{M}_z$ returns to its equilibrium state. The mechanics by which these relaxation events occur are diverse and complex, but can be appreciated from a classical description.

LONGITUDINAL RELAXATION The recovery of longitudinal component after the RF pulse excitation is called longitudinal relaxation, which behaves according to

$$\frac{d\mathbf{M}_z}{dt} = -\frac{\mathbf{M}_z - \mathbf{M}_0}{T_1}, \tag{2.4}$$

where T_1 characterizes the return to equilibrium along the longitudinal direction and is called the spin-lattice time. Physically, T_1 involves the exchange of energy

between nuclei and the surrounding lattice. The equilibrium is reestablished when $\mathbf{M}_z$ is fully recovered to $\mathbf{M}_0$; that is, when the lowest energy state is achieved.

TRANSVERSE RELAXATION The transverse component will gradually disappear after the RF pulse excitation. This process is called transverse relaxation and is described by

$$\frac{d\mathbf{M}_{xy}}{dt} = -\frac{\mathbf{M}_{xy}}{T_2}. \tag{2.5}$$

The solution for the case after a 90° excitation is

$$\mathbf{M}_{xy} = \mathbf{M}_0 \exp(-t/T_2), \tag{2.6}$$

where T_2 characterizes the decay of transverse magnetization and is called the spin-spin time constant. From a macroscopic view, the transverse decay ascribes to the loss of phase coherence of the transverse components.

BLOCH EQUATION The Bloch equation is to describe the longitudinal and transverse relaxation of the nuclear magnetization through a classical approach.

$$\frac{d\mathbf{M}}{dt} = \mathbf{M} \times \gamma\mathbf{B} - \frac{\mathbf{M}_x\mathbf{i} + \mathbf{M}_y\mathbf{j}}{T_2} - \frac{(\mathbf{M}_z - \mathbf{M}_0)\mathbf{k}}{T_1}, \tag{2.7}$$

where

γ	=	gyromagnetic ratio
T_1	=	longitudinal relaxation time constant
T_2	=	transverse relaxation time constant
$\mathbf{M}_0$	=	net magnetization due to $\mathbf{B}_0$ field
$\mathbf{i}, \mathbf{j}, \mathbf{k}$	=	unit vectors in x, y, z-directions respectively.

The cross-product relation describes a precessional behavior. It should be noted that this exponential behavior is an approximation for nuclear spins in the assumption that an ensemble of identical spins is measured.

2.1.2 *Spatial Encoding, K-space Acquisition and Sequence*

With Bloch equation, we are able to describe the magnetization $\mathbf{M}$ in the presence of a magnetic field $\mathbf{B}(t)$. In MR imaging, $\mathbf{B}(t)$ mainly consists of three types of fields: 1) $\mathbf{B}_0$, the static filed; 2) $\mathbf{B}_1(t)$, radiofrequency fields used to excite spins; 3) $\mathbf{G}(t)$, gradient fields used for spatial encoding. The signal in an MR experiment is emf induced by the sweeping of transverse component $\mathbf{M}_{xy}$ after the excitation of RF pulse. To encode the magnetization at different locations

$\mathbf{r} = (x, y, z)$, we manipulate the magnetization with time-varying and spatially dependent gradients in which

$$\mathbf{B}(\mathbf{r}, t) = (\mathbf{B}_0 + \mathbf{B}_G(\mathbf{r}, t))\mathbf{k} \,, \tag{2.8}$$

where $\mathbf{B}_G(\mathbf{r}, t)$ is the location field generated by gradients system and $\mathbf{k}$ is the unit vector in three-dimensional space.

GRADIENT FIELDS Gradient fields are the magnetic fields that vary in the amplitude at different positions but have the same z-direction as the static field $\mathbf{B}_0$. For example, the gradient $\mathbf{G}_x$ along the x-direction is

$$\frac{d\mathbf{B}_z}{dx} = \mathbf{G}_x. \tag{2.9}$$

When the x-gradient field is turned on, it will cause Larmor frequency change related to x-direction

$$\omega(x) = \omega_0 + \Delta\omega = \gamma(\mathbf{B}_0 + \mathbf{G}_x x). \tag{2.10}$$

In order to locate magnetization $\mathbf{M}$ in the three-dimensional space, the gradient system is constructed with

$$\mathbf{G} = \mathbf{G}_x\mathbf{i} + \mathbf{G}_y\mathbf{j} + \mathbf{G}_z\mathbf{k}. \tag{2.11}$$

This yields the total magnetic field

$$\mathbf{B}(\mathbf{r}, t) = (\mathbf{B}_0 + \mathbf{G}(t) \cdot \mathbf{r})\mathbf{k}, \tag{2.12}$$

where t denotes the duration when the gradient is applied. Then, the transverse magnetization is

$$\mathbf{M}(\mathbf{r}, t) = \mathbf{M}_0(\mathbf{r}) e^{-t/T_2(\mathbf{r})} e^{-i\omega_0 t} \exp\left(-i\gamma \int_0^t \mathbf{G}(\tau) \cdot \mathbf{r} d\tau\right) \tag{2.13}$$

SIGNAL EQUATION AND K-SPACE Thus far, we have been able to utilize the Bloch equation to describe the change of the magnetization under the manipulation of RF pulse and gradient system through a classical approach. To characterize the magnetization in the object being studied, the receiver coil is delicately designed to detect the change of magnetization in the transverse plane. With the Faraday's law of induction, the received signals are derived from the

change in the flux of magnetization, which is the contribution of all the transverse magnetization in the volume.

$$\begin{aligned} s_r(t) &= \int_{vol} \mathbf{M}(\mathbf{r}, t)\, dV \\ &= \int_x \int_y \int_z \mathbf{M}(x, y, z, t)\, dx\, dy\, dz. \end{aligned} \tag{2.14}$$

With the solution for Bloch equation, we obtained

$$s_r(t) = \iiint \mathbf{M}_0(x, y, z) e^{-t/T_2(\mathbf{r})} \exp\left(-i\gamma \int_0^t \mathbf{G}(\tau) \cdot \mathbf{r} d\tau\right) dx\, dy\, dz. \tag{2.15}$$

In 2D imaging after slice selection by z-direction gradient and RF pulse, the signal equation is simplified to

$$s_r(t) = \iint \mathbf{M}_0(x, y) e^{-iw_0 t} \exp(-i\gamma \int_0^t \mathbf{G}(\tau) \cdot \mathbf{r} d\tau)\, dx\, dy \tag{2.16}$$

The term $e^{-iw_0 t}$ is eliminated by demodulation using phase sensitive detection. Then, this yields

$$s(t) = s_r(t) e^{iw_0 t} = \iint \mathbf{M}_0(x, y) \exp\left(-i\gamma \int_0^t \mathbf{G}(\tau) \cdot \mathbf{r}\, d\tau\right) dx\, dy, \tag{2.17}$$

where $\mathbf{M}_0(x, y)$ accounts for the proton density $\rho(x, y)$, $T_1(x, y)$ and $T_2(x, y)$. It is intuitive to further expand the equation above into Fourier format

$$s(t) = \iint \mathbf{M}_0(x, y) \exp\left[-i\gamma \left(\int_0^t \mathbf{G}_x(\tau) d\tau\right) x\right] \exp\left[-i\gamma \left(\int_0^t \mathbf{G}_y(\tau) d\tau\right) y\right] dx\, dy, \tag{2.18}$$

then,

$$s(t) = \int_x \int_y \mathbf{M}_0(x, y) e^{-i2\pi[k_x(t)x + k_y(t)y]}\, dx\, dy$$

and

$$k_x(t) = \frac{\gamma}{2\pi} \int_0^t G_x(\tau)\, d\tau, \quad k_y(t) = \frac{\gamma}{2\pi} \int_0^t G_y(\tau)\, d\tau,$$

where k_x and k_y are in units of spatial-frequency, typically cycles/cm. That means the signal $s(t)$ at any time point t corresponds to a point in the domain of spatial frequency. This domain is called k-space in MR field. As the acquisition of image is performed in frequency domain shown in Figure 2.2, the minimum sampling rate, Δk, obeys the Nyquist sampling theorem to avoid aliasing in image domain, i.e., the representation of $\mathbf{M}_0(x, y)$ in spatial space.

SEQUENCE Put simply, the sequence diagram in Figure 2.1 is a specific experimental protocol that includes the precise timing for applying the RF pulse and the

gradient system to acquire sufficient data points in k-space. It also determines the contrast of image by varying the echo time (TE) and repetition time (TR). When a sequence is designed to perform a full acquisition at the minimum or above sampling rate, the reconstruction can be achieved with a fast Fourier transform (FFT).

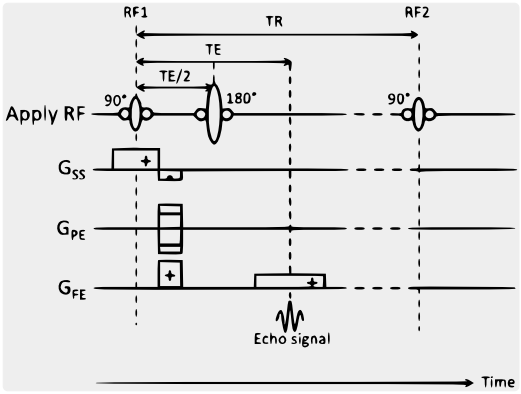


Figure 2.1: Sequence diagram for spin-echo sequence, which starts with 90 degree RF pulse to tip magnetization into the transverse plane. Then, the echo signal with spatial information encoded with gradient field system is detected by multiple coils after appling 180 degree pulse.

2.2 RECONSTRUCTION AS INVERSE PROBLEMS

Generally, the image reconstruction problem is to determine the image $\mathbf{x}$ that represents the magnetization $\mathbf{M}_0(x, y)$ generated with indirect measurements that are manipulated by the pulse sequence executed on an MR scanner. In inverse problems, this image has to be estimated with the forward model of k-space measurement and the measured k-space data. In the case of discrete data, which is how we actually practice in MR acquisition, $\mathbf{y} = [k^x_{y_1}, \cdots, k^x_{y_N}]$ is a vector of k-space elements $k^x_{y_n}$ in a Euclidean space (c.f. Figure 2.2). A linear inverse problem with discrete data assumes a set $\{F_n(\mathbf{x})\}_{n=1}^N$ of linear functions defined on $\mathcal{X}$, which maps $\mathbf{x} \in \mathcal{X}$ to $\mathbf{y} \in \mathcal{Y}$. Riesz theorem states that functions $\psi_1, \ldots, \psi_N$ exist such that

$$F_n(\mathbf{x}) = \langle \mathbf{x}, \psi_n \rangle, \tag{2.19}$$

when functions F_n are continuous on $\mathcal{X}$. The notation $\langle \cdot, \cdot \rangle$ denotes the scalar product used in space $\mathcal{X}$. The detected signal elements Equation 2.16 $s(t) = k^x_{y_n}(t)$ in coils is measured on a finite number of points with Fourier basis functions $\{\psi_n = e^{-i2\pi\omega_n(x,\, y,\, t)}\}_{n=1}^N$ in Equation 2.18, where $\omega_n(x,\ y,\ t) = k_x(t)x + k_y(t)y$. Therefore, the measurement is described with a forward Fourier operator F of $\mathcal{X}$ and $\mathcal{Y}$ by the Equation 2.19. As we practice in a discrete space with the presence of noise η, a matrix-multiplication notation is adopted $\mathbf{y} = \mathcal{A}\mathbf{x} + \eta$ to depict

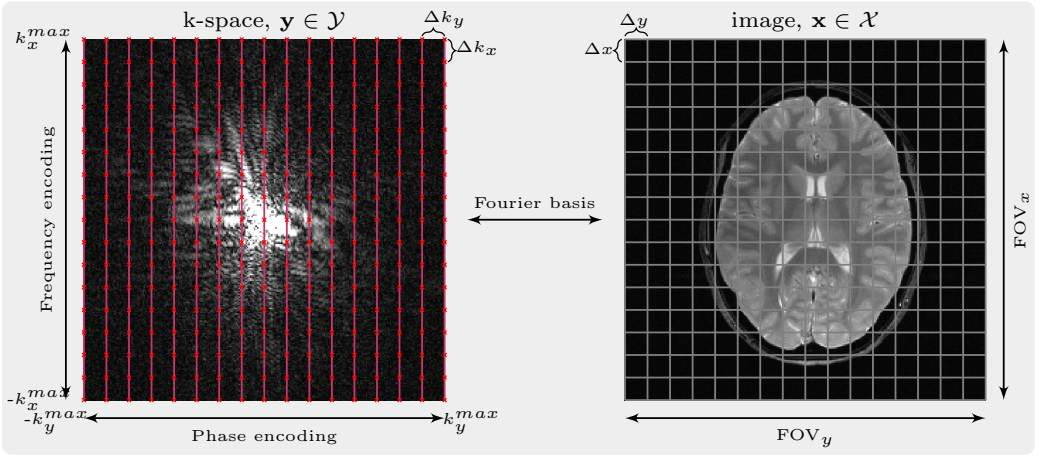


Figure 2.2: The relationship between k-space and image. The red line is a trajectory in k-space $s(t) = k_{y_n}^x(t)$ and the dots on a line are the data points that are acquired in the repetition experiment shown in Figure 2.1.

the measurement in Figure 2.2. When the forward operator $\mathcal{A}$ has the trivial solution $\mathbf{x} = 0$ to $\mathcal{A}\mathbf{x} = 0$ and $\mathcal{Y}$ is closed, the inverse problem is well-posed and has a unique solution, which means a sufficient measurement for $\mathbf{x}$ with $\mathcal{A}$. That is to say, we need to acquire $n \times n$ grid in the k-space for an $n \times n$ image as shown in Figure 2.2, i.e., fully acquistion, which leads to the slow imaging. There is a high level of undersampling in k-space commonly in accelerated MRI techniques, which leads to an ill-posed problem. In other words, the problem of reconstructing the image from highly undersampled k-space data may not have a unique or stable solution.

2.2.1 *Parallel Imaging*

Parallel imaging is the process of measuring the encoded signal with an array of coils simultaneously. Initially, the small array coil was intended to improve signal noise ratio (SNR) as the detected noise is weighted by the local coil's sensitivity and then its magnitude is reduced. The sensitivity of a coil is a spatial profile that describes the receiving $\mathbf{B}_1$ field that induces signals in the coil. The localized coils are able to cover the whole sample. The simultaneous data acquisition, with each coil's sensitivity corresponding to a different subregion, leads to a complete image without aliasing artifacts shown in Figure 2.3.

Since the late 1990s, many k-space acquisition strategies and reconstruction techniques for parallel imaging had emerged with abilities to exploit spatial information induced by the coils' sensitivities. Basically, the goal was to skip some phase encoding steps as the full acquisition of k-space always means a long scanning time, especially for a high-resolution image. The shortened time brings

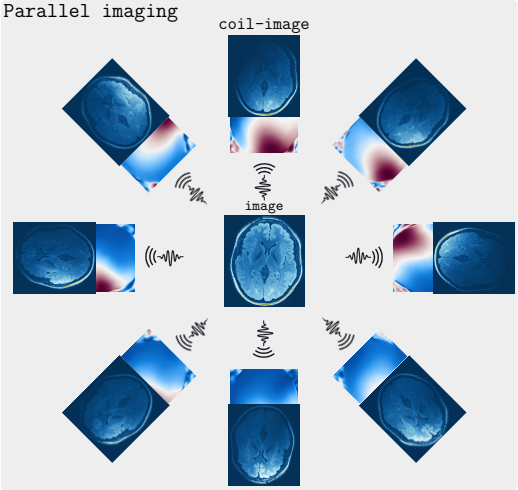


Figure 2.3: The signal detected by a coil is weighted by its local coil profile (the map below coil image), which is called sensitivities and imposes weights on the signal intensity. Consequently, it causes dark and bright regions in coil images.

many benefits in practice such as more patient comfort, less motion artifacts, increased coverage or increased resolution and the economy of more patient throughput. The most well-known techniques are simultaneous acquisition of spatial harmonics[7], sensitivity encoding (SENSE)[8] and generalized autocalibrating partially parallel acquisition (GRAPPA)[9]. These methods can be mainly categorized into two types: 1) in which the reconstruction is performed in image domain using an unfolding or inverse algorithm; and 2) in which the reconstruction is achieved through the completion of the missing k-space lines or points.

GENERALIZED INVERSE AND SENSE In the parallel imaging, the forward model is formulated as

$$F(\mathbf{x}, \mathbf{c}) := (\mathcal{F}_S(\mathbf{x} \odot c_1), \cdots, \mathcal{F}_S(\mathbf{x} \odot c_N)) = \mathbf{y}\,, \tag{2.20}$$

where $\mathcal{F}_S$ is an undersampled Fourier transform operator and the correspondingly obtained k-space data is $\mathbf{y} = [y_1, \cdots, y_{nc}]^T$; $\mathbf{y} \in \mathbb{C}^{d \times nc}$, $\mathbf{x} \in \mathbb{C}^{n \times n}$ denotes the image content and $\mathbf{c} = [c_1, \cdots, c_{nc}]^T$; $\mathbf{c} \in \mathbb{C}^{n \times n \times nc}$ denotes the coil sensitivities. Without any undersampling, i.e., the sampling rate is at the Nyquist rate or above, one optimal reconstruction is the root of the sum of square of the coil-weighted images. The undersampling of k-space would make Equation 2.20 ill-posed. The pseudosolution $\mathbf{x}$ satisfies

$$\hat{\mathbf{x}} = \arg\min \|\mathbf{y} - F_{\mathbf{c}}\mathbf{x}\|_2^2\,, \tag{2.21}$$

where $F_{\mathbf{c}}$ is the coil sensitivity encoded undersampled Fourier operator. In SENSE[8], the coil sensitivities $\mathbf{c}$ are estimated from a calibration scan. Then, we obtain

$$F_{\mathbf{c}}^* F_{\mathbf{c}} \mathbf{x} = F_{\mathbf{c}}^* \mathbf{y} \ , \tag{2.22}$$

which yields

$$\mathbf{x} = F_{\mathbf{c}}^\dagger \mathbf{y} \ . \tag{2.23}$$

The operator $F_{\mathbf{c}}^\dagger = (F_{\mathbf{c}}^* F_{\mathbf{c}})^{-1} F_{\mathbf{c}}^*$ is called generalized inverse, which minimizes Equation 2.21. In SENSE, the Landweber iterative method is used to solve Equation 2.22 with

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \alpha F_{\mathbf{c}}^* (\mathbf{y} - F_{\mathbf{c}} \mathbf{x}^n) \ , \tag{2.24}$$

where $0 < \alpha < 2/\|F_{\mathbf{c}}^* F_{\mathbf{c}}\|$.

REGULARIZATION When the original inverse problem is ill-posed in Equation 2.20, generalized inverse in a finite dimension guarantees that the corresponding problem in Equation 2.22 is well- posed and conditioned. However, it behaves in practice like an ill-posed problem and has to be treated with regularization techniques. The pseudosolution, i.e., generalized inversion, to Equation 2.21 minimizes the discrepancy between the model and the data. The high undersampling rate in k-space causes the forward operator $F(\mathbf{x}, \mathbf{c})$ has non-trivial solution and therefore leads to ill-posedness in Equation 2.20. In sequence, the exact solution to Equation 2.22 from insufficient data would not guarantee the real solution of object. We therefore introduce a prior, which enforce our prior knowledge on the solution. It is described as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} R(\mathbf{x}) \quad \text{s.t. } \|F(\mathbf{x}, \mathbf{c}) - \mathbf{y}\|_2^2 = \epsilon, \tag{2.25}$$

where $R(\mathbf{x})$ is the regularization function, and ϵ denotes the degree of relaxation catering for the noise during the measurement. In the application of compressed sensing[1, 10, 11], where the sampling pattern is random or pseudo random and the prior knowledge is that the image exhibits sparsity in a certain domain Φ (e.g., wavelet Φ). The exact reconstruction is guaranteed with $R(\mathbf{x}) = \|\Phi \mathbf{x}\|_1$, given a certain amount of incomplete frequency data, and achieved by minimizing,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|F(\mathbf{x}, \mathbf{c}) - \mathbf{y}\|_2^2 + \lambda \|\Phi \mathbf{x}\|_1 \ . \tag{2.26}$$

Therefore, the solution is faithful to the data up to a certain extent and sparsity in the wavelet domain about the reconstruction is enforced. The precomputing of coil sensitivities $\mathbf{c}$, which can be realized with the center k-space[12], causes Equation 2.26 a linear optimization problem. In nonlinear inversion (NLINV)[13, 14], the iteratively regularized Gauss-Newton method is used to solve Equation 2.20

and Equation 2.26 directly and estimate image and coil sensitivities jointly. In this thesis, NLINV is detailed and used in Chapter 4.

2.2.2 *Bayesian Approach*

As a statistical inference problem, the image $\mathbf{x}$ is treated as parameters in the parameters space $\mathcal{H}$. The estimation of true image is defined with propositions that: 1) the true value of parameters is in $\mathcal{H}$; 2) the observed k-space data $\mathbf{y}$ is in the sample space $\mathcal{S}$; 3) the parameters are linked with the data through the forward model F; 4) any prior information about the parameters. The inference is to draw the most possible conclusions from all the information at our disposal.

Probabilistic modeling is a natural way to describe our situation. The probability law of the observation $\mathbf{y}$ as a result of $\mathbf{x}$ is described by

$$p(\mathbf{y}|\mathbf{x}) \propto e^{-f(\mathbf{y},\,\mathbf{x})} \,, \tag{2.27}$$

where $f(\mathbf{y},\ \mathbf{x})$ presents the uncertainty on $\mathbf{x}$ with the observed information $\mathbf{y}$. Let $f(\mathbf{x},\mathbf{y}) = \|F(\mathbf{x},\mathbf{c}) - \mathbf{y}\|_2^2$. Then, we estimate the image by maximizing the likelihood $p(\mathbf{y}|\mathbf{x})$. When F has a trivial solution, the global maximum corresponds to a single point. Vice versa, in the case of a non-trivial solution, the global maximum encompasses a whole region, i.e., non-uniqueness. With Bayesian approach, we are able to take the advantage of prior information. The posterior of the image $\mathbf{x}$, given the observed data $\mathbf{y}$ is

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \,. \tag{2.28}$$

The prior distribution for any $\mathbf{x}$ is

$$p(\mathbf{x}|\lambda,\theta) = e^{-\lambda R_\theta(\mathbf{x})}/Z(\lambda,\theta) \,, \tag{2.29}$$

where $R_\theta(\mathbf{x})$ maps $\mathbf{x}$ to a vector of statistics and $Z(\lambda,\theta)$ is the normalizing constant given by

$$Z(\lambda,\theta) = \int_x e^{-\lambda R_\theta(\mathbf{x})}\, dx \,. \tag{2.30}$$

The L^1 prior has been applied to MRI, which is essentially a Laplace distribution that is used to promote sparse property of image in wavelet domain Φ,

$$g(\mathbf{x} \mid \mu, b) = \frac{|\Phi\mathbf{x} - \mu\mathbf{I}|}{b} \,, \tag{2.31}$$

where μ is the location and b is the scale, $\theta = (\mu, b)$. Once the $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x}|\lambda, \theta)$ are chosen, the posterior in Bayes' theorem is

$$p(\mathbf{x}|\mathbf{y}, \lambda, \theta) = \exp\left[f(\mathbf{x}, \mathbf{y}) - \lambda R_\theta(\mathbf{x})\right] / Z(\lambda, \theta) \ . \tag{2.32}$$

The posterior distribution comprises two parts, the observation $\mathbf{y}$ and the prior information. The MAP estimator is commonly used, given by

$$\hat{\mathbf{x}}_{\lambda,\theta,\mathrm{MAP}} = \arg\min_{\mathbf{x}} \left\{f(\mathbf{x}, \mathbf{y}) - \lambda R_\theta(\mathbf{x})\right\} \ . \tag{2.33}$$

Let $f(\mathbf{x}, \mathbf{y}) = \|F(\mathbf{x}, \mathbf{c}) - \mathbf{y}\|_2^2$. Then, this estimator meets with the regularized inverse problem and can be achieved in most case with modern optimization algorithms, such as proximal gradient, projected gradient and so on. The λ is a parameter that controls the balance between the observation and prior information. We may have prior information on the estimation we are looking for, e.g., the medical image database. The application of machine learning techniques allows us to take advantage of it. Within statistical framework, many tools are available to investigate the inverse problem, particularly when addressing the uncertainty and generalization capabilities of the learned methods even in the context of a black-box neural networks.

2.3 MACHINE LEARNING FOR IMAGE RECONSTRUCTION

The objective of ML is to understand and learn the patterns contained in data, and then use them for prediction of future data or decision-making. ML is a set of methods that achieves this by borrowing the ideas and methods from probability, statistics, numerical optimization, physics and so on. The last decades have witnessed the development in the construction of MRI for capturing medical images for diagnosis. With the fast growth of ML application, researchers have started to improve techniques and strategies used in MR imaging with it, from the design of k-space sampling pattern to image reconstruction and from data collection to data analysis.

2.3.1 *Unrolled Neural Networks*

One of the most common approaches in ML, where an algorithm learns to predict an output variable based on one or more input variables, by training on a labeled dataset. Specifically in the field of MR image reconstruction, many unrolling neural networks[3, 4, 15] utilize this approach to solve this problem

$$\arg\min_{\mathbf{x}} \|F(\mathbf{x}, \mathbf{c}) - \mathbf{y}\|_2^2 + R_\theta(\mathbf{x}) \ . \tag{2.34}$$

Probabilistically, the output of a network $F_\theta^\dagger$ parametrized by θ is trained to predict the reconstruction by maximizing the likelihood

$$\hat{\theta} = \arg\max_\theta \mathcal{L}(X = \mathbf{x}, \hat{X} = F_\theta^\dagger(\mathbf{y})) = p_D(\mathbf{x}, \mathbf{y}|F_\theta^\dagger) \,, \tag{2.35}$$

which is evaluated with

$$\hat{\theta} = \arg\min_\theta \mathbb{E}_D \left[\|F_\theta^\dagger(\mathbf{y}) - \mathbf{x}\|_2^2 \right] \,. \tag{2.36}$$

A dataset $D = \{X = \mathbf{x}_i, Y = \mathbf{y}_i\}_{i=1}^N$, consisting of paired the undersampled k-space data $\mathbf{y}$ and the reference images $\mathbf{x}$, and the pre-definition of the forward operator F are required for backpropagation training. Designed for Equation 2.34, these networks usually consist of two crucial components: 1) data consistency layer of F, which involves Fourier transform and coil sensitivities to ensure consistency of the output to measured k-space; 2) regularization layers $R_\theta(\mathbf{x})$ (θ, learnable weights), which are constructed with many image processing functionalities, such as learnable convolutions, upsampling, downsampling and so on, to learn the features that are able to enhance image quality. These networks are the result of the unrolling of iterative optimization algorithms used to solve Equation 2.34, which glues these components together.

Ref. [5] reported that tiny perturbations, small structural changes, and sampling pattern variations can fail most inverse networks $\mathbf{x} = F_\theta^\dagger(\mathbf{y})$. Ref. [16] proofed this phenomena showing that small deviations from the training data would lead to substantial decreases in image quality. This aligns with our assumption on ML that it learns the patterns contained in data and then use them for prediction. Once the testing data was out of the patterns Equation 2.35 because of deviations, the behavior of the learned model would be unpredictable. To circumvent this constraint, Ref. [17] introduced an approach that initially represents the training images within probabilistic subspace and subsequently the inverse network is trained in the subspace domain, which demonstrates improved robustness in the presence of perturbations. Apart from this, the prior knowledge could be imposed on learnable weights θ in an unrolling network, when training it by maximizing the posterior of θ

$$\hat{\theta} = \arg\max_\theta \{\log p_D(\theta|\mathbf{x}, \mathbf{y}) \propto \log p_D(\mathbf{x}, \mathbf{y}|F_\theta^\dagger) + \log p(\theta)\} \,. \tag{2.37}$$

In Ref. [18], it assumes θ follows a Gaussian distribution and learns this distribution using backpropagation to demonstrate the uncertainty in weights.

In short, unrolling networks perform well in the distribution of training data, but deviations from the learned distribution always happen, as we may change sampling patterns, coils, MR contrasts, sequence, and so on. One possible way

Table 2.1: A comparison of generative models

Model	Training	Likelihood	Sampling	Compression	Representation
autoregressive	stable	exact	slow	lossless	no
flow-based	stable	exact	fast/slow	lossless	yes
generative adversarial network (GAN)s	unstable	no	fast	no	no
variational autoencoder (VAE)s	stable	approximate	fast	lossy	yes
energy-based	stable	unnormalized	fast	no	yes
score-based	stable	estimated	fast/slow	lossless	yes

out is to decouple the learned distribution from the forward operator, which essentially pertains to many factors that lead to deviations.

2.3.2 *Generative Models*

From the perspective of Bayes in Equation 2.33, the prior knowledge $R_\theta(\mathbf{x})$ about our images is naturally decoupled from the forward operator. We could probabilistically learn the distribution of images $p_\theta(\mathbf{x})$ from a dataset $D = \{X = \mathbf{x}_i\}_{i=1}^N$ using generative models and then utilize it as regularization $R_\theta(\mathbf{x}) = \log p_\theta(\mathbf{x})$. Then, the estimation of reconstruction is realized by

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \ \log p_\theta(\mathbf{x}) \quad \text{s.t. } \|F(\mathbf{x}, \mathbf{c}) - \mathbf{y}\|_2^2 = \epsilon \ . \tag{2.38}$$

Within a Bayesian framework, the posterior of the image $\mathbf{x}$, given a measured k-space data $\mathbf{y}$, is

$$p_\theta(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}) \cdot p_\theta(\mathbf{x}) \ . \tag{2.39}$$

The MAP estimator from Equation 2.39 could be equivalent to Equation 2.38 using Equation 2.27 and has been investigated in Refs. [19, 20]. Furthermore, many stochastic sampling methods to explore the posterior $p_\theta(\mathbf{x}|\mathbf{y})$, such as Monte Carlo and simulated annealing algorithms, etc., become viable options for imaging inverse problems while harnessing the prior information in a dataset $D = \{X = \mathbf{x}_i\}_{i=1}^N$. A generative model is a statistical model $p_\theta(X = \mathbf{x})$ that is used to approximate distribution of a high-dimensional random variable $p_D(X = \mathbf{x})$ from an observed dataset D. Usually we can divide generative models into five main groups: 1) autoregressive generative models; 2) flow-based models; 3) latent variable models; 4) energy-based models; 5) score-based models. (c.f. Table 2.1) The autoregressive and score-based models are used in this thesis and discussed below.

AUTOREGRESSIVE GENERATIVE MODELS In statistics and signal processing, an autoregressive model is a representation of a type of random process; as

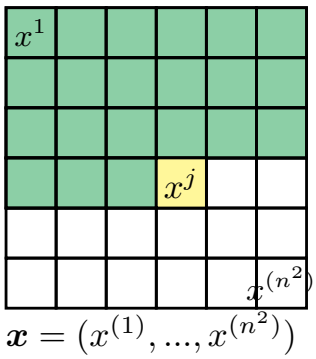


Figure 2.4: The pixel x^j is conditioned on the previous pixels, namely the pixels on the above and left.

such, it is used to describe certain time-varying processes in nature. This model specifies that the output variable depends on its own previous values.

$$p(\mathbf{x}) = p(x_0) \prod_{i=1}^{N} p(x_i|\mathbf{x}_{<i}) \ , \tag{2.40}$$

where $\mathbf{x}_{<i} = [x_1, \cdots, x_{i-1}]$. We can treat an image as such a vector shown in Figure 2.4. Each pixel value ν follows a logistic distribution parametrized by μ, s. For instance, when modeling discrete pixel value $\nu \in [0, 255]$, the probability for each value is the integration over the short interval $[\nu - 0.5, \nu + 0.5]$

$$P(x|\mu, s) = \begin{cases} \sigma(\frac{x-\mu+0.5}{s}) & \text{for } x = 0 \\ \sigma(\frac{x-\mu+0.5}{s}) - \sigma(\frac{x-\mu-0.5}{s}) & \text{for } 0 < x < 255 \\ 1 - \sigma(\frac{x-\mu-0.5}{s}) & \text{for } x = 255 \end{cases}$$

where σ is the sigmoid function, which is the cumulative density function for the logistic distribution. For the boundary value i.e., 0 or 255, the integration for them is to the positive or negative infinity. To avoid uni-modal, we use a mixture of logistics for ν:

$$p(\nu) = \sum_{i=1}^{K} \pi_i \ \text{logistic}(\mu_i, s_i) \ , \tag{2.41}$$

where π_i is categorical weights. To deal with images that have multiple channels such as RGB-valued images, we write

$$p(x) = p(\nu_r, \nu_g, \nu_b) = p(\nu_r) p(\nu_g|\nu_r) p(\nu_b|\nu_g, \nu_r) \ , \tag{2.42}$$

where

$$p(\nu_r) = \sum_{i=1}^{K} \pi_i \, \text{logistic}(\mu_i^r, s_i^r) \tag{2.43}$$

$$p(\nu_g|\nu_r) = \sum_{i=1}^{K} \pi_i \, \text{logistic}(\mu_i^g + \alpha_i^0 \nu_r, s_i^g) \tag{2.44}$$

$$p(\nu_b|\nu_g, \nu_r) = \sum_{i=1}^{K} \pi_i \, \text{logistic}(\mu_i^b + \alpha_i^1 \nu_g + \alpha_i^2 \nu_r, s_i^b) \; . \tag{2.45}$$

Then, x_i in an image vector $\mathbf{x}$ shown in Figure 2.4 is parametrized by $\Theta = \{\pi_i, \mu_i^{(r,g,b)}, s_i^{(r,g,b)}, \alpha_i^{(0,1,2)}\}_{i=1}^{K}$. Therefore, the autoregressive model in Equation 2.40

$$p(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{\alpha}) = p(x^{(1)}) \prod_{i=2}^{n^2} p(x^{(i)} \mid x^{(1)}, .., x^{(i-1)}) \; , \tag{2.46}$$

which describes the dependencies among sub-pixels. PixelCNN[21] is a deep neural network parametrized by θ and trained to predict the distribution parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{\alpha}) = \text{NET}_\theta(\mathbf{x})$ for the image $\mathbf{x}$ by maximizing

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_D[\log p(\mathbf{x}; \text{NET}_\theta(\mathbf{x}))] \; . \tag{2.47}$$

When applying this model to audio and image, the conditional distributions are computationally inefficient. The causal convolutions as presented in Ref. [22] for audio and in Ref. [21, 23] accelerate the computation and make it possible for this type of models to model the complicated and high dimensional data. Readers could find technical details about the causal convolution in those publications.

SCORE-BASED MODELS In Ref. [24], Hyvärinen proposed score mathcing where a learn score function of distribution $p_\theta(\mathbf{x})$ with respect to $\mathbf{x}$,

$$\psi(\mathbf{x}, \theta) = \frac{\partial \log p(\mathbf{x}, \theta)}{\partial \mathbf{x}} \; , \tag{2.48}$$

is enforced to match the corresponding score of the data distribution,

$$\frac{\partial \log p_D(\mathbf{x})}{\partial \mathbf{x}} \; , \tag{2.49}$$

by minimizing the expected squared error between these two terms

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_D \left[\frac{1}{2} \left\| \psi(\mathbf{x}, \theta) - \frac{\partial \log p_D(\mathbf{x})}{\partial \mathbf{x}} \right\|^2 \right] \; . \tag{2.50}$$

Given a dataset $D = \{X = \mathbf{x}_i\}_{i=1}^N$ and $p_D(\mathbf{x})$ is unknown, the equivalent objective is

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_D \left[\frac{1}{2} \left\| \psi(\mathbf{x}, \theta) \right\|^2 + \frac{\partial \log \psi(\mathbf{x}, \theta)}{\partial \mathbf{x}} \right] . \tag{2.51}$$

In a slightly different situation, a score network $\psi(\tilde{\mathbf{x}}, \theta)$ is trained with pairs of clean and corrupted samples $(\mathbf{x}, \tilde{\mathbf{x}})$ using the following objective

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_D \left[\frac{1}{2} \left\| \psi(\tilde{\mathbf{x}}, \theta) - \frac{\partial \log p_D(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} \right\|^2 \right] , \tag{2.52}$$

where we have a joint density $p_D(\tilde{\mathbf{x}}, \mathbf{x}) = p_D(\tilde{\mathbf{x}}|\mathbf{x}) p_D(\mathbf{x})$, $\tilde{\mathbf{x}} = \mathbf{x} + \sigma \mathbf{z}$. σ is the standard deviation and $\mathbf{z}$ is the uniform isotopic Gaussian noise. Note that

$$\nabla_{\tilde{\mathbf{x}}} \log p_D(\tilde{\mathbf{x}}_{\sigma_i}^k|\mathbf{x}) = \frac{\partial \log p_D(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} = \frac{1}{\sigma^2}(\mathbf{x} - \tilde{\mathbf{x}}) = \frac{\mathbf{z}}{\sigma} . \tag{2.53}$$

This means the prediction of $\mathbf{z}/\sigma$ from noisy $\tilde{\mathbf{x}}$ would point to $\mathbf{x}$, and we want the score network match that as good as it can. In Ref. [25] constructed a sequence distribution

$$p(\tilde{\mathbf{x}}_{\sigma_i}|\tilde{\mathbf{x}}_{\sigma_{i-1}}),\ \tilde{\mathbf{x}}_{\sigma_i} = \tilde{\mathbf{x}}_{\sigma_{i-1}} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\mathbf{z} , \tag{2.54}$$

where $0 \approx \sigma_0 < \cdots < \sigma_i = \sigma_{max}$. When σ is small, $\tilde{\mathbf{x}}$ is close to $\mathbf{x}$ and retains the most of structural information of it. Vice versa, $\tilde{\mathbf{x}}$ has less structural information. It is able to sample a distribution with its score function using Langevin dynamics. Then, an annealed Langevin dynamics is proposed with

$$\tilde{\mathbf{x}}_{\sigma_i}^{k+1} = \tilde{\mathbf{x}}_{\sigma_i}^k + \frac{\gamma}{2} \nabla_{\tilde{\mathbf{x}}} \log p_D(\tilde{\mathbf{x}}_{\sigma_i}^k|\mathbf{x}) + \sigma_i \sqrt{\gamma} \mathbf{z} , \tag{2.55}$$

where the score $\nabla_{\tilde{\mathbf{x}}} \log p_D(\tilde{\mathbf{x}}_{\sigma_i}^k|\mathbf{x})$ is estimated by $\psi(\tilde{\mathbf{x}}, \theta)$ and K iterations are performed at each σ_i step. As proofed in Ref. [26], the capability of a score network to predict $\mathbf{z}/\sigma$ from $\tilde{\mathbf{x}}$ decreases as σ increases in Equation 2.52. Therefore, when there is no other information and we want to generate samples for $p_D(\mathbf{x})$ with a trained score network, we should use Equation 2.55 reversely by starting at σ_{max} and ending at σ_0, initializing with Gaussian noise. In Ref. [27, 28], Equation 2.54 and Equation 2.55 are referred to as forward and reverse process respectively.The forward process gradually transforms the distribution of data $p_D(\mathbf{x})$ to a distribution $\mathcal{N}(\mathbf{x}, \sigma_{max}^2 \mathbf{I})$ that is close to a known distribution $\mathcal{N}(\mathbf{0}, \sigma_{max}^2 \mathbf{I})$. The reverse process does the opposite. Essentially, with a defined continuous $\sigma(t)$ and $\mathbf{x}_t := \tilde{\mathbf{x}}_{\sigma(t)}$, Equation 2.54 is equivalent to a diffusion process

$$d\mathbf{x}_t = \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\mathbf{w} \tag{2.56}$$

where $\mathbf{w}$ is a Weiner process, and whose corresponding reverse process is

$$d\mathbf{x}_t = -d[\sigma^2(t)]\nabla_{\mathbf{x}_t} \log p_D(\mathbf{x}_t) + \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\mathbf{w} \,. \tag{2.57}$$

Therefore, this type of generative models also called the diffusion model that is constructed with a score network.

LATENT VARIABLE MODELS Latent variable models assume there is a generator that maps a lower dimensional latent variable $\mathbf{z} \sim p(\mathbf{z})$ to a high dimensional variable $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$. The probabilistic principal component analysis (pPCA)[29] is an example of this type of models where $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$ are Gaussian distributions, and the dependency between $\mathbf{x}$ and $\mathbf{z}$ is linear. A non-linear extension of pPCA with arbitrary distribution is the VAE framework as shown in Figure 2.5. In VAE, variational inference is utilized to approximate the posterior $p(\mathbf{x}|\mathbf{z})$ and neural networks are used to parametrize the distributions. There is an adversarial approach[30] to creating such a generator where a discriminator determines a difference between real data and generated data yielded by the generator, which is referred to as GAN as shown in Figure 2.5. This group of models is called implicit models as they do not compute or approximate the log-likelihood. Different to flow-based models, VAE and GAN models do not need to satisfy the invertibility of neural networks, and therefore we have more freedom in design architecture for encoders and decoders.

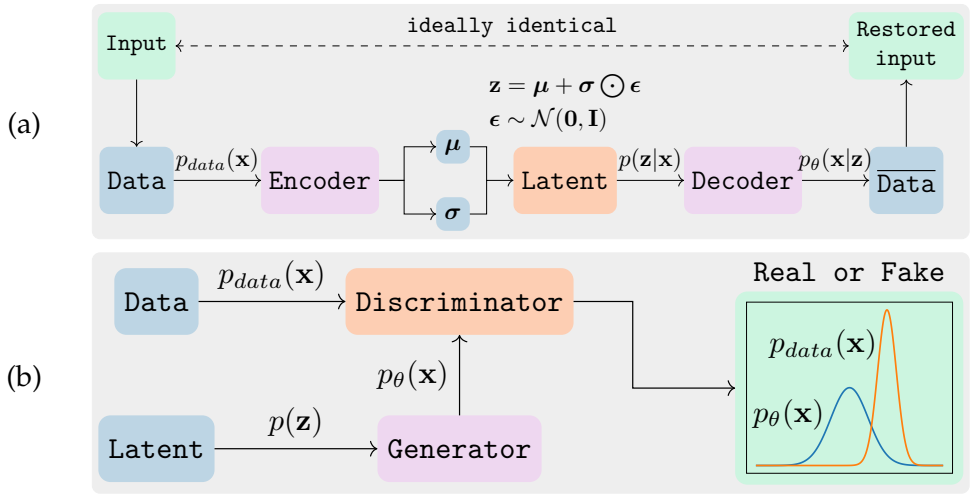


Figure 2.5: (a) A variational autoencoder encodes the data into an assumed multivariate Gaussian distribution and decodes the latent space into the reconstructed data. (b) A GAN has a discriminator to determine if the samples from the generator and training dataset are real or fake. Once it can not distinguish from the real and the fake, the distribution parametrized by the generator stays close the data distribution.

3

BAYESIAN MRI RECONSTRUCTION WITH JOINT UNCERTAINTY ESTIMATION USING DIFFUSION PRIORS

The chapter contains a reprint of the original article.

G. Luo, M. Blumenthal, M. Heide, M. Uecker. "Bayesian MRI reconstruction with joint uncertainty estimation using diffusion models". *Magn Reson Med.* March 2023;1-17. DOI: 10.1002/mrm.29624

GL conceptualized this work, implemented the algorithm, designed and performed the numerical experiments, and analyzed the results. *MB* implemented the algorithm in Berkeley advanced reconstruction toolbox (BART). *GL* discussed the related concepts with *MH*. *MU* supervised this work. *GL* and *MU* wrote the manuscript.

ABSTRACT

Purpose: We introduce a framework that enables efficient sampling from learned probability distributions for MRI reconstruction.

Method: Samples are drawn from the posterior distribution given the measured k-space using the Markov chain Monte Carlo (MCMC) method, different from conventional deep learning-based MRI reconstruction techniques. In addition to the maximum a posteriori (MAP) estimate for the image, which can be obtained by maximizing the log-likelihood indirectly or directly, the minimum mean square error (MMSE) estimate and uncertainty maps can also be computed from those drawn samples. The data-driven Markov chains are constructed with the score-based generative model learned from a given image database and are independent of the forward operator that is used to model the k-space measurement.

Results: We numerically investigate the framework from these perspectives: 1) the interpretation of the uncertainty of the image reconstructed from undersampled k-space; 2) the effect of the number of noise scales used to train the generative models; 3) using a burn-in phase in MCMC sampling to reduce computation; 4) the comparison to conventional ℓ_1-wavelet regularized reconstruction; 5) the transferability of learned information; and 6) the comparison to fastMRI challenge.

Conclusion: A framework is described that connects the diffusion process and advanced generative models with Markov chains. We demonstrate its flexibility in terms of contrasts and sampling patterns using advanced generative priors and the benefits of also quantifying the uncertainty for every pixel.

3.1 INTRODUCTION

Modern Magnetic Resonance Imaging (MRI) formulates reconstruction from raw data in Fourier space (k-space) as an inverse problem. Undersampling to reduce acquisition time then leads to an ill-posed reconstruction problem. To solve this problem, parallel imaging can exploit spatial information from multiple receive coils in an extended forward model [31]. Compressed sensing uses the sparsity of images in a transform domain (i.e. wavelet domain, finite differences) as prior knowledge. Combined with incoherent sampling this allows recovery of sparse images from highly undersampled data [1, 2]. Learning-based techniques for compressed sensing include methods using dictionary learning [32] or a patch-based nonlocal operator [33].

In recent years, the application of deep learning pushed these ideas forward by integrating learned prior knowledge [34]. Most of these methods can be classified into two categories: First, methods that unroll the existing iterative reconstruction algorithms into a neural network and train their parameters by maximizing the similarity to a ground truth. In Ref. [3], the authors replaced the handcrafted regularization term with convolution layers, and derived a neural network from the iterative procedure of the Alternating Direction Method of Multipliers (ADMM) algorithm. Ref. [15, 4] investigated similar approaches. The downside of this kind of method is the need for supervised training, which requires raw k-space data with fixed known sampling patterns and corresponding ground truth images. The second category consists of methods that learn a prior from high-quality images, then plug it into existing iterative algorithms as a regularization term. In Ref. [19, 20, 35], the image prior was constructed with a variational auto-encoder [36], a denoising auto-encoder [37] and an autoregressive generative model [23], respectively. These methods then compute a maximum a posterior (MAP) as the estimator of the image. These types of methods separate the learned information from the encoding matrix (sampling pattern in k-space and coil sensitivities), which permits more flexibility in practice because they allow the acquisition patterns and receive coils to change without retraining. Generative adversarial networks were also used for image reconstruction in Ref. [38]. There, the discriminator is used to confine the space of the output of a generator that is designed to generate images with conformity to k-space data.

Although deep learning based approaches provide promising results, worries about the uncertainty caused by undersampling strategies and algorithms have limited their usage in clinical practice until now. Therefore, the uncertainty assessment constitutes an important step for deep learning based approaches. The uncertainty is two-fold: 1) the uncertainty of weights inside the neural network [39, 40]; and 2) the uncertainty introduced by the missing k-space data points. The uncertainty from missing k-space data points can be addressed in a

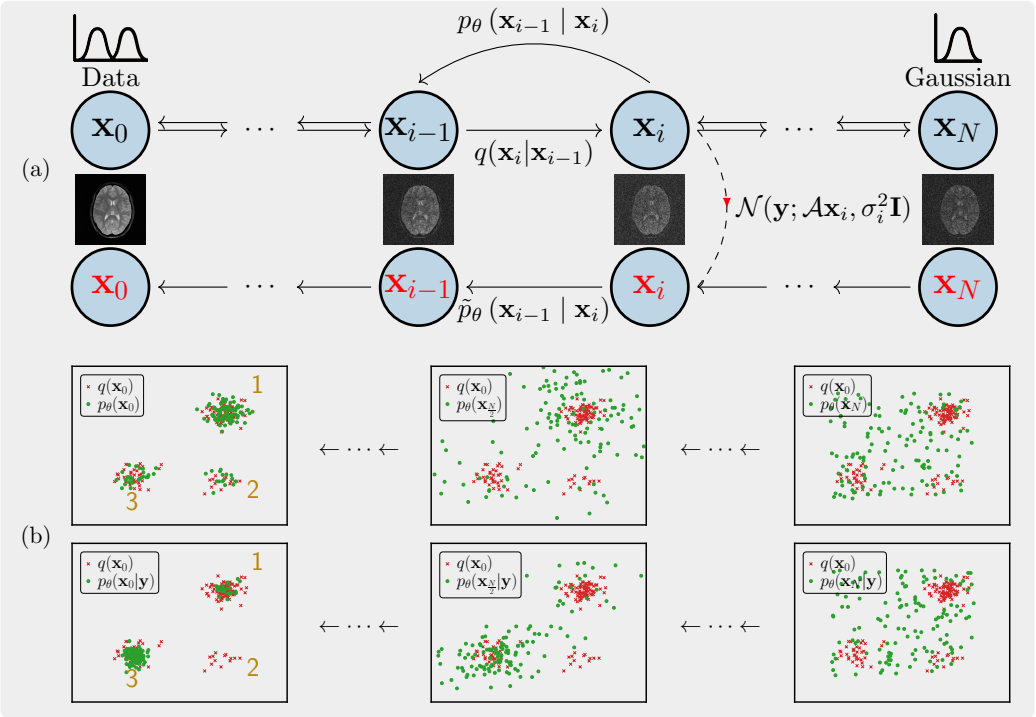


Figure 3.1: Overview of the proposed method. (a) The unknown data distribution $q(\mathbf{x}_0)$ of the training images goes through repeated Gaussian diffusion and finally reaches a known Gaussian distribution $q(\mathbf{x}_N)$, and this process is reversed by learned transition kernels $p_\theta(\mathbf{x}_{i-1} \mid \mathbf{x}_i)$. To compute the posterior of the image $p(\mathbf{x}|\mathbf{y})$, a new Markov chain $\tilde{p}_\theta(\mathbf{x}_{i-1} \mid \mathbf{x}_i)$ is constructed by incorporating the measurement model into the reverse process (red chain). (b) Training samples (red dots) from a mixture of bivariate Gaussian distribution are shown. The upper and bottom rows illustrate how samples (green dots) gradually gather around training samples in the reverse process, without and with the observation, respectively. In this example, the likelihood for the observation was a bivariate Gaussian mixture, so that cluster 2 has a lower and cluster 3 has a higher probability.

Bayesian imaging framework. We refer the readers to Ref. [41, 42]. In Ref. [20], the MAP estimator is used, but it provides only the mode of the posterior density $p(\mathbf{x}|\mathbf{y})$ and practical optimization may also even only provide a local maximum. In the setting of Bayesian inference, it is possible to investigate the full shape of posterior distribution $p(\mathbf{x}|\mathbf{y})$. In particular, it is possible to draw sample from the posterior distribution for priors based on diffusion models using the Markov chain Monte Carlo (MCMC) method as described previously by Jalal et al. [43] and others [44, 45, 46], which are closely related to the present work. Jalel et al. use Langevin sampling to sample the posterior using score-based generative model and this is extendend in Ref. [46] to also include a motion model. The method in Ref. [45] uses the predictor-and-corrector framework proposed in Ref. [28]. These publications point out the relationship to Bayesian reconstruction and show some results related to uncertainty estimation, but a complete Bayesian formulation of this framework applied to MRI multi-channel reconstruction is not provided. A general problem with this approch is the large number of iterations required during sampling, e.g. Ref. [45] reports the use of several thousands of iterations.

Following these ideas, a generic framework for MRI reconstruction emerges, which is based on a series of publications related to generative models [28, 24, 47, 27, 25], in which the essential idea is to: 1) systematically and slowly destroy the underlying prior knowledge in a data distribution through an iterative forward diffusion process; 2) learn a reverse diffusion process that restores the patterns by a so-called score-based neural network and 3) incorporate the forward model of the measurement into the learned reverse process. The general picture of the proposed method is illustrated in Figure 4.1.

In the present work, we recapitulate the framework of Bayesian reconstruction and score-based diffusion models and numerically investigate this framework from the following different perspectives: 1) the interpretation of the uncertainty of the image reconstructed from undersampled k-space; 2) the effect of the number of noise scales used the generative models on image quality on computation time; 3) using a burn-in phase in MCMC sampling to reduce computation; 4) the comparison to conventional ℓ_1-wavelet regularized reconstruction; 5) the transferability of learned information; and 6) the comparison to fastMRI challenge[48, 49].

3.2 THEORY

3.2.1 *Magnetic Resonance Image Reconstruction as Bayesian Inference*

We consider image reconstruction as a Bayesian problem where the posterior of image $p(\mathbf{x}|\mathbf{y})$ given with the measured data $\mathbf{y}$ and a prior $p(\mathbf{x})$ learned from a database of images [20, 41, 42]. Here, the image is denoted as $\mathbf{x} \in \mathbb{C}^{n\times n}$, where

$n \times n$ is the size of image, and $\mathbf{y} \in \mathbb{C}^{m \times m_C}$ is the vector of m complex-valued k-space samples from m_C receive coils. Assuming the noise η circularly-symmetric normal with zero mean and covariance matrix $\sigma_\eta^2 \mathbf{I}$, the likelihood $p(\mathbf{y}|\mathbf{x})$ for observing the $\mathbf{y}$ determined by $\mathbf{y} = \mathcal{A}\mathbf{x} + \eta$ and given the image $\mathbf{x}$ is given by a complex normal distributions

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \mathcal{CN}(\mathbf{y}; \mathcal{A}\mathbf{x}, \sigma_\eta^2 \mathbf{I}) \\ &= (\sigma_\eta^2 \pi)^{-N_p} e^{-\|\sigma_\eta^{-1} \cdot (\mathbf{y} - \mathcal{A}\mathbf{x})\|_2^2} , \end{aligned} \tag{3.1}$$

where $\mathbf{I}$ is the identity matrix, σ_η the standard deviation of the noise, $\mathcal{A}\mathbf{x}$ is the mean and N_p is the length of the k-space data vector. $\mathcal{A} : \mathbb{C}^{n \times n} \to \mathbb{C}^{m \times m_C}$ is the forward operator and given by $\mathcal{A} = \mathcal{P}\mathcal{F}\mathcal{S}$, where $\mathcal{S}$ are the coil sensitivity maps, $\mathcal{F}$ the two-dimensional Fourier transform, and $\mathcal{P}$ the k-space sampling operator. According to Bayes' theorem the posterior density function $p(\mathbf{x}|\mathbf{y})$ is then

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x})}{p(\mathbf{y})} . \tag{3.2}$$

In this work, the reconstruction is based on the sampling of this posterior distribution. We utilize an efficient technique based on the Markov Chain Monte Carlo method with the application of a diffusion probabilistic generative model. This consists of two processes: 1) a forward diffusion process which converts a complicated distribution used as prior for the image into a simple Gaussian distribution; and 2) a learned finite-time reversal of this diffusion process with which a Gaussian distribution is gradually transformed back to the posterior (cf. Figure 4.1).

3.2.2 *The Forward Diffusion Process*

In probabilistic diffusion models, the data distribution characterized by density $q(\mathbf{x}_0)$ is gradually converted into an analytically tractable distribution (Gaussian noise) [27]. The image $\mathbf{x}_0$ is perturbed with a sequence of noise scales $0 = \sigma_0 < \sigma_1 < \cdots < \sigma_N$. When the number of steps used for discretization $N \to \infty$, the diffusion process becomes a continuous process. Here, we consider the discrete Markov chain

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{z}_{i-1}, \quad i = 1, \cdots, N, \tag{3.3}$$

where $\mathbf{z}_{i-1} \sim \mathcal{CN}(\mathbf{0}, (\sigma_i^2 - \sigma_{i-1}^2)\mathbf{I})$, i.e. the i-th transition kernel is then given by

$$q(\mathbf{x}_i|\mathbf{x}_{i-1}) = \mathcal{CN}(\mathbf{x}_i; \mathbf{x}_{i-1}, (\sigma_i^2 - \sigma_{i-1}^2)\mathbf{I}) . \tag{3.4}$$

Instead of doing transitions step by step [28, 50] a single perturbation kernel

$$q(\mathbf{x}_i \mid \mathbf{x}_0) = \mathcal{CN}\left(\mathbf{x}_i; \mathbf{x}_0, \sigma_i^2 \mathbf{I}\right) \tag{3.5}$$

can be computed as a convolution of Gaussians. With Bayes' theorem we can write:

$$q\left(\mathbf{x}_{i-1} \mid \mathbf{x}_i, \mathbf{x}_0\right) = q\left(\mathbf{x}_i \mid \mathbf{x}_{i-1}\right) \frac{q\left(\mathbf{x}_{i-1} \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_i \mid \mathbf{x}_0\right)} \ . \tag{3.6}$$

Given the initial image $\mathbf{x}_0$, the posterior of a single step of the forward process is then given by (see Appendix Section 3.7.1)

$$q(\mathbf{x}_{i-1} \mid \mathbf{x}_i, \mathbf{x}_0) = \mathcal{CN}\big(\mathbf{x}_{i-1}; \frac{\sigma_{i-1}^2}{\sigma_i^2}\mathbf{x}_i + \big(1 - \frac{\sigma_{i-1}^2}{\sigma_i^2}\big)\mathbf{x}_0, \tau_i^2 \mathbf{I}\big) \tag{3.7}$$

with variance $\tau_i^2 := \left(\sigma_i^2 - \sigma_{i-1}^2\right)\left(\sigma_{i-1}^2 / \sigma_i^2\right)$.

3.2.3 *Learning the Reverse Process*

The joint distribution of the reversal diffusion process is characterized by the probability density

$$p(\mathbf{x}_N, \mathbf{x}_{N-1}, \cdots, \mathbf{x}_0) = p(\mathbf{x}_N) \prod_{i=1}^{N} p(\mathbf{x}_{i-1} | \mathbf{x}_i) \ , \tag{3.8}$$

where $p(\mathbf{x}_N)$ is the initial Gaussian distribution. The reverse is given by Kolmogorov's backward equation which has the same form as the forward process [27, 50]. Thus, the transitions $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$ of the reverse process can be parameterized with the Gaussian transition kernel

$$p\left(\mathbf{x}_{i-1} \mid \mathbf{x}_i\right) = \mathcal{CN}\left(\mathbf{x}_{i-1}; \boldsymbol{\mu}\left(\mathbf{x}_i, i\right), \tau_i^2 \mathbf{I}\right), \tag{3.9}$$

where $\boldsymbol{\mu}\left(\mathbf{x}_i, i\right)$ and $\tau_i^2 \mathbf{I}$ are the mean and variance of the reverse transitions, respectively. Here, we learn the mean μ_θ of the reverse transitions using a neural network parameterized by training parameters θ. Since the learned reverse transitions $p_\theta(\mathbf{x}_{i-1}|\mathbf{x}_i)$ lead to a new density $p_\theta(\mathbf{x}_0)$, which should match $q(\mathbf{x}_0)$, they can be learned by minimizing the cross entropy

$$H(p_\theta, q) = -\mathbb{E}_{q(\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0)\right] \ . \tag{3.10}$$

Following Ref. [27] a lower bound ℓ can be written in terms of KL divergence between the transition kernel Equation 3.9 and the posterior of forward process Equation 3.7

$$\begin{aligned}\ell &= \sum_{i=2}^{N} \mathbb{E}_{q(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_i|\mathbf{x}_0)} \left[D_{\mathrm{KL}}(q(\mathbf{x}_{i-1} \mid \mathbf{x}_i, \mathbf{x}_0) \| \, p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1} \mid \mathbf{x}_i)) \right] \\ &= \sum_{i=2}^{N} \mathbb{E}_{q(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_i|\mathbf{x}_0)} \\ &\quad \left[\frac{1}{\tau_i^2} \left\| \frac{\sigma_{i-1}^2}{\sigma_i^2} \mathbf{x}_i + \left(1 - \frac{\sigma_{i-1}^2}{\sigma_i^2}\right) \mathbf{x}_0 - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_i, i) \right\|_2^2 \right] + C, \end{aligned} \tag{3.11}$$

where C is a constant. The derivation of KL divergence between two Gaussian distributions is detailed in Appendix Section 3.7.2. Using Equation 3.5 we can express $\mathbf{x}_i = \mathbf{x}_0 + \mathbf{z}$ with $\mathbf{z} \sim \mathcal{CN}\left(\mathbf{0}, \sigma_i^2 \mathbf{I}\right)$, and obtain

$$\ell = \sum_{i=2}^{N} \mathbb{E}_{\mathbf{x}_0, \mathbf{z}} \left[\frac{1}{\tau_i^2} \left\| \frac{\sigma_{i-1}^2}{\sigma_i^2} \mathbf{z} + \mathbf{x}_0 - \boldsymbol{\mu}_{\boldsymbol{\theta}} \left(\mathbf{x}_i, i\right) \right\|_2^2 \right] + C. \tag{3.12}$$

Thus, we can learn the mean of the reverse transitions by learning to denoise the training data disturbed by noise. In Ref. [24, 47], the generative model is estimated by minimizing the expected squared distance between the gradient of the log-probability given by the score network and the gradient of the log-probability of the observed data. This technique was extended and generalized in Ref. [25, 28]. In the following, we quickly point out the connection to score matching networks. Let:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}} \left(\mathbf{x}_i, i\right) - \mathbf{x}_0 = \sigma_{i-1}^2 \mathbf{s}_{\boldsymbol{\theta}} \left(\mathbf{x}_i, i\right), \tag{3.13}$$

where $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_i, i)$ denotes the denoising score matching network that is conditional on the index of noise scales i. Then, we have

$$\ell = \sum_{i=2}^{N} \mathbb{E}_{\mathbf{x}_0, \mathbf{z}} \left[\frac{\sigma_{i-1}^2}{\tau_i^2} \left\| \frac{\mathbf{z}}{\sigma_i^2} - \boldsymbol{s}_{\boldsymbol{\theta}} \left(\mathbf{x}_i, i\right) \right\|_2^2 \right] + C \, . \tag{3.14}$$

Expressing the noise again as $\mathbf{z} = \mathbf{x}_i - \mathbf{x}_0$, we can rewrite

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}_0, \mathbf{z}} \left[\left\| \frac{\mathbf{x}_i - \mathbf{x}_0}{\sigma_i^2} - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_i, i) \right\|_2^2 \right] \\ &= \mathbb{E}_{q(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_i|\mathbf{x}_0)} \left[\| \nabla_{\mathbf{x}_i} \log q(\mathbf{x}_i \mid \mathbf{x}_0) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_i, i) \|_2^2 \right] \end{aligned} \tag{3.15}$$

which shows that Equation 3.14 is equivalent to score matching. For the later use of the transition kernel, Equation 3.13 is equivalent to

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\mathbf{x}_i, i\right) - \mathbf{x}_i = \left(\sigma_i^2 - \sigma_{i-1}^2\right) \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_i, i\right) \ . \tag{3.16}$$

In summary, the score network is trained via Equation 3.15 to output the gradient fields that are used to construct the Markov transitions (Equation 3.9) which nudges coarse samples $\mathbf{x}_i$ toward finer ones $\mathbf{x}_{i-1}$, namely the reverse process. In later sections, we will discuss how we construct and train the score networks.

3.2.4 *Computing the Posterior for MRI Reconstruction*

In order to compute the posterior probability $p(\mathbf{x}|\mathbf{y})$ for the image $\mathbf{x}$ given the data $\mathbf{y}$, we need to modify the learned reverse process. We achieve this by multiplying each of the intermediate distributions $p(\mathbf{x}_i)$ with the likelihood term $p(\mathbf{y}|\mathbf{x}_i)$ according to Bayes' theorem. We use $\tilde{p}\left(\mathbf{x}_i\right) = p(\mathbf{x}_i|\mathbf{y})$ to denote the resulting sequence of intermediate distributions

$$\tilde{p}\left(\mathbf{x}_i\right) \propto p\left(\mathbf{x}_i\right) p(\mathbf{y}|\mathbf{x}_i) \tag{3.17}$$

up to the unknown normalization constant. Following Ref. [27], the transition from $\mathbf{x}_{i+1}$ to $\mathbf{x}_i$ of the modified reverse process is

$$\tilde{p}\left(\mathbf{x}_i \mid \mathbf{x}_{i+1}\right) \propto p\left(\mathbf{x}_i \mid \mathbf{x}_{i+1}\right) p(\mathbf{y}|\mathbf{x}_i) \ . \tag{3.18}$$

The sampling at each intermediate distribution of Markov transitions Equation 3.18 is performed with the unadjusted Langevin algorithm [51]

$$\mathbf{x}_i^{k+1} \leftarrow \mathbf{x}_i^k + \frac{\gamma}{2} \nabla_{\mathbf{x}_i} \log \tilde{p}(\mathbf{x}_i^k \mid \mathbf{x}_{i+1}) + \sqrt{\gamma}\mathbf{z}, \tag{3.19}$$

where $\mathbf{z}$ is standard complex Gaussian noise $\mathcal{CN}(0, \mathbf{I})$. We now go over to the modified learned process $\tilde{p}_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{x}_{i+1})$ parameterized by θ and obtain the log-derivative with respect to $\mathbf{x}_i$ using the learned reverse transitions $p_{\boldsymbol{\theta}}\left(\mathbf{x}_i \mid \mathbf{x}_{i+1}\right)$ as

$$\nabla_{\mathbf{x}_i} \log \tilde{p}_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{x}_{i+1}) = \nabla_{\mathbf{x}_i} \log p_{\boldsymbol{\theta}}\left(\mathbf{x}_i \mid \mathbf{x}_{i+1}\right) + \nabla_{\mathbf{x}_i} \log p(\mathbf{y}|\mathbf{x}_i). \tag{3.20}$$

From Equation 3.9 and Equation 3.16, we have

$$\nabla_{\mathbf{x}_i} \log p_{\boldsymbol{\theta}}\left(\mathbf{x}_i \mid \mathbf{x}_{i+1}\right) = \frac{1}{\tau_{i+1}^2}\left(\sigma_{i+1}^2 - \sigma_i^2\right) \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_{i+1}, i\right), \tag{3.21}$$

and from Equation 3.1 we have

$$\nabla_{\mathbf{x}_i} \log p(\mathbf{y}|\mathbf{x}_i) = -\frac{1}{\sigma_\eta^2}(\mathcal{A}^H\mathcal{A}\mathbf{x}_i - \mathcal{A}^H\mathbf{y}) \ . \tag{3.22}$$

After inserting these expressions into Section 4.3.2 we obtain

$$\begin{aligned}\mathbf{x}_i^{k+1} \leftarrow \mathbf{x}_i^k + \frac{\gamma}{2\tau_{i+1}^2}(\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_i^k, i) \\ - \frac{\gamma}{2\sigma_\eta^2}(\mathcal{A}^H\mathcal{A}\mathbf{x}_i^k - \mathcal{A}^H\mathbf{y}) + \sqrt{\gamma}\mathbf{z} \ .\end{aligned} \tag{3.23}$$

The starting point for each chain $\mathbf{x}_i^0 = \mathbf{x}_{i+1}^{\mathrm{K}}$ is the last sample from the previous distribution $\tilde{p}(\mathbf{x}_{i+1} \mid \mathbf{x}_{i+2})$ after K Langevin steps. We found it advantageous to modify the likelihood term in each step according $\sigma_\eta^2 = \tau_{i+1}/\lambda$, which should approach the variance of the data noise in the last step. Since the noise variance was unknown for the data set we used, we empirically selected a λ that determines how strong the k-space data consistency is relative to the prior. We set γ to $2\tau_{i+1}^2$. At last, the algorithm used to sampling the posterior is presented in 2.

Algorithm 1 Sampling the posterior with a Markov chain Monte Carlo method

1: Give the acquired k-space $\mathbf{y}$.
2: Construct the forward operator $\mathcal{A}$ with sampling pattern $\mathcal{P}$ and coil sensitivities $\mathcal{S}$.
3: Set the Langevin steps K, the factor λ, the start noise level index N, and γ.
4: Generate $\mathbf{x}_{\mathrm{N}}^0$ from a suitable Gaussian distribution (e.g., $\mathcal{CN} \sim (0, \mathbf{I})$).
5: **for** i in $\{\mathrm{N}-1, \cdots, 1\}$ **do**
6: Draw samples from $\tilde{p}(\mathbf{x}_i|\mathbf{x}_{i+1})$ by running K Langevin steps with Equation 3.23.
7: **end for**

To characterize the shape of a posterior, we run multiple chains to draw samples in parallel. To reduce the amount of computation, the burn-in phase is introduced as shown in Figure 3.2. That means only one chain proceeds through the several beginning noise levels, and after that we split it up into multiple Markov chains using the sample from the burn-in phase as initial point indicated by the blue dot. To further reduce computation, we introduce the continuously decreasing noise scales, which reduces the number of iterations when performing Langevin dynamics at each intermediate distribution.

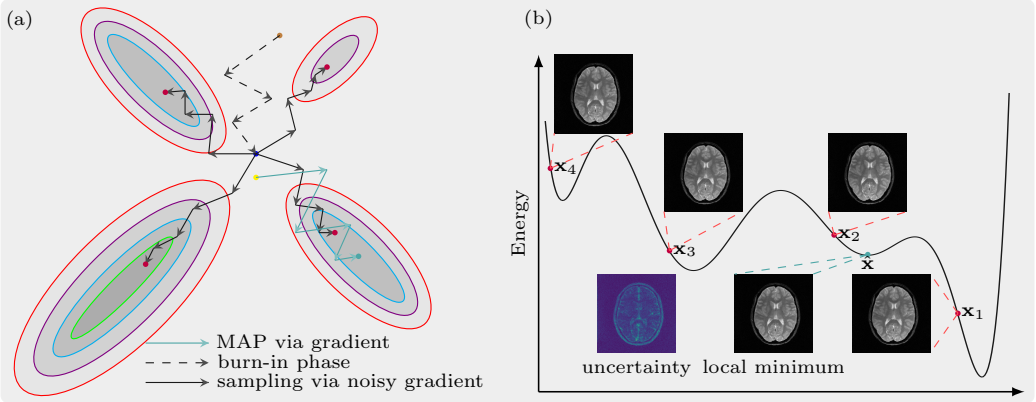


Figure 3.2: Illustration for the sampling of the posterior $p(\mathbf{x}|\mathbf{y})$. (a) The four possible sampling trajectories are indicated the solid lines, sharing the same burn-in phase (dashed line). The MAP approach via gradient descent reaches a locally optimal solution. (b) Possible reconstructions are showed over the energy curve and the uncertainty map is the pixelwise variance over samples.

3.2.5 *The Analysis of Samples*

Given a posterior probability distribution $p(\mathbf{x}|\mathbf{y})$ the minimum mean square error (MMSE) estimator minimizes the mean square error:

$$\mathbf{x}_{\text{MMSE}} = \arg\min_{\tilde{\mathbf{x}}} \int \|\tilde{\mathbf{x}} - \mathbf{x}\|^2 p(\mathbf{x}|\mathbf{y}) d\mathbf{x} = \mathbb{E}[x|y] \ . \tag{3.24}$$

The MMSE estimator cannot be computed in a closed form, and numerical approximations are typically required. Since we demonstrated how to generate samples from the posterior in previous sections, let us consider the samples $\mathbf{x}_0^K$ at the last stage, and a consistent estimate of $\mathbf{x}_{\text{MMSE}}$ can be computed by averaging those samples, i.e. the empirical mean of samples converges in probability to $\mathbf{x}_{\text{MMSE}}$ due to weak law of large numbers. The variance of those samples is a solution to the error assessment for the reconstruction if we trust the model parameterized by Equation 3.9 that is learned from a image database. The 95% confidence interval is computed for each pixel with its mean and variance. Since a wider confidence interval (CI) means a larger margin of error, the mean is overlaid with it to indicate the variability of each pixel, and up to a certain point, the variability can cause a visual change on the image (cf. Section 3.3.3).

3.3 METHODS

3.3.1 *Score Networks' Architecture*

The denoising score network is designed to predict the noise given an image degraded by Gaussian noise of a particular scale σ_i. To improve the quality of the predictions for different noise scales, we consider networks conditional on discrete and pseudo-continuous noise scales. The discrete one has a much larger gap between σ_i and σ_{i-1} than the pseudo-continuous one and usually has a smaller number of noise scales N, while the pseudo-continuous network is adaptive to a certain trained range of noise scales. The sequence of noise scales $\{\sigma_i\}_{i=1}^{\mathrm{N}}$ is geometrically generated following the scheme in Ref. [28], i.e. $\sigma_i = \sigma(\frac{i}{N}) = \sigma_{\min}(\frac{\sigma_{\max}}{\sigma_{\min}})^{\frac{i-1}{N-1}}$.

For a discrete model, we add modified instance normalization layers that are conditional on the index of the noise scales following each convolution layer. The conditional instance normalization [52] is

$$\hat{f}_k = \Phi[i,k]\frac{f_k - \mu_k}{s_k} + \Omega[i,k], \tag{3.25}$$

where $\Phi \in \mathbb{R}^{N\times C}$ and $\Omega \in \mathbb{R}^{N\times C}$ are learnable parameters, k denotes the index of a feature map f_k, μ_k and s_k are the means and standard deviation over its spatial locations of the k-th feature map computed in each pass through the network, and i denotes the index of σ in $\{\sigma_i\}_{i=1}^{\mathrm{N}}$.

For a continuous model, we let networks be conditional on the index of noise scales by inserting random Fourier features [53]. Three steps used to encode a noise index into random features are as follows:

- Draw a random vector which has i.i.d. Gaussian m entries with the specified standard deviation,
- Scale the random vector with the index i, then multiply it with 2π,
- Apply sines and cosines to the scaled random vector, then concatenate them into $m \times 2$ matrix,

where m is embedding size. The encoded index is added to all the blocks listed in Supporting Table S1.

With either one of the two modifications above, a network $\mathbf{s}_\theta(\mathbf{x}, i)$ has two inputs, i.e. noise corrupted image $\mathbf{x}$ and noise index i. Real and imaginary parts of the images are interpreted as seperate channels when input into the neural network. RefineNet [54] is the backbone of all the score networks used in this work (cf. Supporting Figure S2). Three variants from that are trained for different reconstruction experiments. The architectures of three networks are presented

in detail in Supporting Table S1. We refer the readers to the codes available online for more information about them. We labeled the three networks with $\texttt{NET}_1, \texttt{NET}_2$, and $\texttt{NET}_3$, respectively, for ease of reference in the following. $\texttt{NET}_1$ is conditional on discrete noise scales, $\texttt{NET}_2$ and $\texttt{NET}_3$ are conditional on continuous noise scales. We introduce self-attention modules into $\texttt{NET}_3$ to capture long-range dependencies by adding non-local blocks as described previously[55] so that the network has the capability to model the dataset of high-resolution images.

3.3.2 *Dataset, Training and Inference*

We trained $\texttt{NET}_1$ and $\texttt{NET}_2$ on a dataset acquired by us already used and described in Ref. [20]. $\texttt{NET}_3$ was trained on a subset of the fastMRI dataset [48]. Our dataset has 1300 images containing T1-weighted, T2-weighted, T2-weighted fluid-attenuated inversion recovery (FLAIR), and $T2^*$-weighted brain images from 13 healthy volunteers examined with clinical standard-of-care protocols. The brain images from fastMRI dataset [48] were used for benchmark that contains T1-weighted (some with post contrast), T2-weighted and FLAIR images. For the detailed information of both dataset, we refer readers to corresponding publication. Regarding the data partitioning, we first separated all multi-slice volumes into training and testing groups. Then we split the volume into two-dimensional slices (i.e., images). Reference images - denoted $\mathbf{x}_0$ in the theory - were reconstructed from fully-sampled multi-channel k-space. Then, these complex image datasets after coil combination were normalized to a maximum magnitude of 1. The coil sensitivity maps were computed with BART toolbox using ESPIRiT [12, 56]. 1300 images of size 256×256 from the dataset used in Ref. [20] were used to train $\texttt{NET}_1$ and $\texttt{NET}_2$. 1000 images were used for training, and 300 images were used for testing. All networks are trained for 1000 epochs, i.e. iterations over all training images. For the training of $\texttt{NET}_3$, we used the T2-weighted FLAIR contrast images of size 320×320 that are reconstructed from fastMRI raw k-space data. 2937 images are for training, 326 images are for testing.

Three score networks are implemented with Tensorflow [57]. The hyperparameters used to train the three score networks are listed in Supporting Table S2. With the trained networks, we implemented MCMC sampling Algorithm 2 with Tensorflow and Numpy [58], and then explored the posterior $p(\mathbf{x}|\mathbf{y})$ in different experimental settings. We trained three score networks once separately for all the experiments we did in this work. These three models can support all experiments performed in this study with variable undersampling patterns, coil sensitivity maps, channel numbers. It took around 43 and 67 seconds, respectively, to train $\texttt{NET}_1$ and $\texttt{NET}_2$ for one epoch on one NVIDIA A100 GPU with 80GB. For $\texttt{NET}_3$, it took around 500 seconds per epoch on two NVIDIA A100 GPUs using the

multi-GPU support from Tensorflow. In the spirit of reproducible research, codes and data to reproduce all experiments are made available[1].

3.3.3 *Experiments*

Single Coil Unfolding: To investigate how the Markov chain explores the solution space of the inverse problem $\mathbf{y} = \mathcal{A}\mathbf{x} + \eta$, we designed the single coil unfolding experiment. The single channel k-space is simulated out of multi-channel k-space data. The odd lines in k-space are retained. 10 samples were drawn from the posterior $p(\mathbf{x}|\mathbf{y})$. $\texttt{NET}_1$ was used to construct transition kernels and the parameters in Algorithm 2 are $\mathrm{K} = 50, \mathrm{N} = 10, \lambda = 6$. We redo the experiment with the object shifted to bottom. This experiment has an inherent ambiguity which can not be resolved using the data alone and where the reconstruction is strongly determined by the prior. Thus, it mimics in a synthetic setting a situation with high undersampling where hallucinations were observed in the reconstruction of some deep-learning methods [59].

Multi-Coil Reconstruction Multi-channel data points from Cartesian k-space are randomly picked with variable-density poisson-disc sampling and the central 20×20 region is fully acquired. The acquisition mask covers 11.8% k-space and the corresponding zero-filled reconstruction is shown Figure 3.4b. We initialized 10 chains and the $\mathbf{x}_{\mathrm{MMSE}}$ was computed using different numbers of samples. $\texttt{NET}_1$ was used to construct transition kernels and the parameters in Algorithm 2 are $\mathrm{K} = 30, \mathrm{N} = 15, \lambda = 13$. To visualize the process of sampling, we use peak-signal-noise-ratio (PSNR in dB) and similarity index (SSIM) as metrics to track intermediate samples. The comparisons are made between the magnitude of $\mathbf{x}_{\mathrm{MMSE}}$ and the ground truth $\tilde{\mathbf{x}}$ after normalized with ℓ_2-norm.

More Noise Scales To investigate how the number of noise scales influences the proposed method, we reconstructed the image from the undersampled k-space that was used in the multi-coil experiment. $\texttt{NET}_2$ was used to construct transition kernels and the parameters in Algorithm 2 are $\mathrm{K} = 5, \mathrm{N} = 70, \lambda = 25$.

Investigation of the Burn-in Phase To investigate the burn-in phase illustrated in Figure 3.2, we split up into multiple chains at a certain noise scale when drawing samples from the posterior $p(\mathbf{x}|\mathbf{y})$. For instance, we denote by $(\mathbf{x}_{\mathrm{MMSE}}, 60)$ the $\mathbf{x}_{\mathrm{MMSE}}$ that is computed with 10 samples drawn from $p(\mathbf{x}|\mathbf{y})$ by splitting up into 10 chains at the 60^{th} noise scale. By changing the splitting point, we got different sets of samples that are from chains of different length and computed the final $\mathbf{x}_{\mathrm{MMSE}}$ respectively. We have two sets of $\mathbf{x}_{\mathrm{MMSE}}$ that are reconstructed from the undersampled k-space using two sampling patterns separately. The central 20x20 region is obtained and the k-space, outside the center, is randomly picked

1 https://github.com/mrirecon/spreco

up retrospectively (10%, 20%). $\texttt{NET}_2$ was used to construct Markov transition kernels and the parameters in Algorithm 2 are $\mathrm{K} = 5, \mathrm{N} = 70, \lambda = 25$.

Investigation into MAP To verify the samples are located around the local modality of the posterior, we disabled the disturbance with noise after stochastic inference with the last distribution $\tilde{p}(\mathbf{x}_0 \mid \mathbf{x}_1)$ and ran 200 iterations more to get extended samples. What's more, we repeated this procedure with determinate inference, in which the disturbance was disabled during sampling iterations to get one deterministic sample, i.e., MAP estimation. A Poisson-disc sampling pattern is generated without variable density and with 2-fold undersampling along phase and frequency encoding directions. $\texttt{NET}_2$ was used to construct transition kernels and the parameters in Algorithm 2 are $\mathrm{K} = 5, \mathrm{N} = 70, \lambda = 25$.

Comparison to ℓ_1-regularized Reconstruction A comparison using the fastMRI dataset was used to evaluate the performance of the proposed method. We noticed that the raw k-space data is padded with zeros to make them have the same dimension. The effect caused by zero paddings is investigated in Ref. [60]. Since we only used the images that were reconstructed from the zero padded k-space for training, the issue caused by the synthesized k-space does not exist in our work. The undersampling pattern for each slice is randomly generated in all retrospective experiments. $\texttt{NET}_3$ was used to construct transition kernels. The parameters in Algorithm 2 are $\mathrm{K} = 3, \mathrm{N} = 90, \lambda = 20$ and 10 samples were drawn to compute $\mathbf{x}_{\mathrm{MMSE}}$. The data range for computing PSNR and SSIM is determined by the maximum over each slice.

Transferability To investigate the transferability of learned prior information from T2 FLAIR images to other contrasts, we acquired T1-weighted (TR=2000ms, TI=900ms, TE=9ms) and T2-weighted (TR=9000ms, TI=2500ms, TE=81ms) FLAIR k-space data using a 2D multi-slice turbo spin-echo sequence with a 16-channel head coil at 3T (Siemens, 3T Skyra). $\texttt{NET}_3$ (trained with T2 FLAIR images) was used to construct transition kernels. The parameters in Algorithm 2 are $\mathrm{K} = 5, \mathrm{N} = 70, \lambda = 20$.

Comparison to fastMRI challenge As a comparison to the unrolled neural network, the XPDNet [49] is selected as the reference which ranked 2nd in the fastMRI challenge. Two networks were trained for acceleration factors 4 and 8, using retrospectively undersampled data from the fastMRI dataset [10] using equidistant Cartesian masks and the trained models that are publicly available[2]. For the proposed method, $\texttt{NET}_3$ was used to construct transition kernels. The parameters in Algorithm 2 are $\mathrm{K} = 4, \mathrm{N} = 90, \lambda = 20$. The confidence interval after thresholding is used as the color map to indicate that a region has high uncertainty. Be consistent with the evaluation the XPDNet provided, 30 FLAIR volumes are used for validation to compute metrics.

2 https://huggingface.co/zaccharieramzi

3.4 RESULTS

3.4.1 *Single Coil Unfolding*

As expected, the lack of spatial information from coil sensitivities without parallel imaging leads to huge errors and folding artifacts still exist in $\mathbf{x}_{\text{MMSE}}$ as shown in Figure 3.3. Since only odd lines are acquired, all images in which the superposition of points P_l and $\mathrm{P}_{l+2/n}$ equals to the points P_r in ground truth are solutions to $\mathbf{y} = \mathcal{A}\mathbf{x} + \epsilon$ with the same error (the residual norm $\|\mathbf{y} - \mathcal{A}\mathbf{x}\|^2$). Selected solutions are presented in Figure 3.3c. The variance map indicates the uncertainty of the solutions, which in this experiment is similar to the hallucinations observed in for some deep-learning methods for high undersampling[59]. The errors of the estimation $\mathbf{x}_{\text{MMSE}}$ are largely reduced compared to the zero-filled reconstruction because of prior knowledge from the learned reverse process (cf. Figure 3.3a). The shift of the object increases the symmetry and then leads to even bigger errors as learned reverse process know less about images that were shifted (cf. Figure 3.3b).

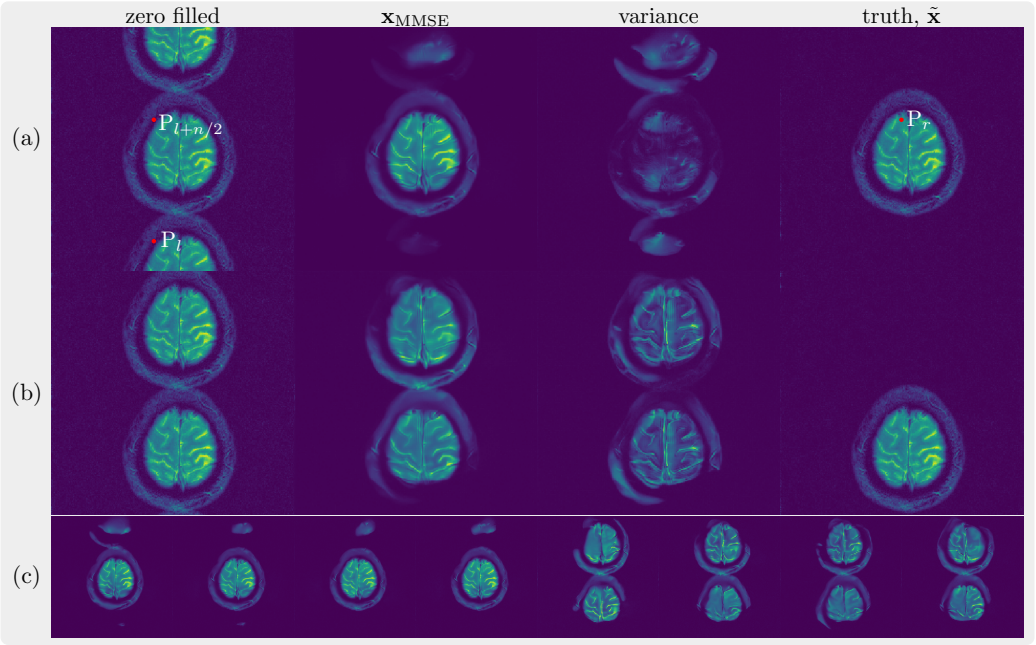


Figure 3.3: Single-coil unfolding with $\texttt{NET}_1$. The k-space is undersampled by skipping every second line. Aliased images, $\mathbf{x}_{\text{MMSE}}$, variance maps and ground truth are shown. (a) The object is centered. (b) The object is shifted. (c) Selected solutions are presented. The left four are centered and the right four are shifted.

3.4.2 *Multi-Coil Reconstruction*

Figure 3.4 shows the results for the multi-coil experiment. Figure 3.4a shows the evolution of the samples' PSNR and SSIM over the transitions of the data-driven Markov chain. Intermediate samples are presented in Supporting Figure S1. The convergence of samples at each noise level was reached as indicated by the PSNR and SSIM curves. When there are more samples, the $\mathbf{x}_{\text{MMSE}}$ converges to higher PSNR and SSIM. In Figure 3.4b, 10 converged samples were used to compute $\mathbf{x}_{\text{MMSE}}$ and the variance map. Comparing with the ground truth, the variance map mainly reflects the edge information, which can be interpreted by the uncertainty that is introduced by the undersampling pattern used in k-space where many high frequency data points are missing but the low frequency data points are fully acquired. In contrast to the single coil unfolding, the local spatial information from coil sensitivities reduces the uncertainties of missing k-space data. Moreover, error maps qualitatively correspond to the variance map, with larger errors in higher variance regions as shown in Figure 3.4c. Lastly, the average over more samples leads to smaller error.

3.4.3 *More Noise Scales*

We also plotted the curve of PSNRs and SSIMs over iterations in Figure 3.5a for $\texttt{NET}_2$ which uses continuous noise scales. The PSNR and SSIM of $\mathbf{x}_{\text{MMSE}}$, which is computed with 10 samples, are 37.21dB and 0.9360, respectively. Two $\mathbf{x}_{\text{MMSE}}$ reconstructed separately with the application of $\texttt{NET}_1$ and $\texttt{NET}_2$ are presented in Figure 3.5b and variance maps are presented as well. The variance of the samples that are drawn with $\texttt{NET}_2$ is less than those drawn with $\texttt{NET}_1$, which means that we are more confident about the reconstuction using $\texttt{NET}_2$. When we zoom into the region that has more complicated structures, the boundaries between white matter and gray matter are more distinct in the image recovered with $\texttt{NET}_2$ and the details are more obvious, as shown in Figure 3.5c. Hence, increasing the number of noise scales in $\texttt{NET}_2$ relative to $\texttt{NET}_1$ reduces the number of iterations and improves the quality of reconstruction using score networks of comparable size. More noise scales make chains constructed with $\texttt{NET}_2$ exploit the prior knowledge from training image dataset more effectively than chains constructed with $\texttt{NET}_1$ which has fewer noise scales.

3.4.4 *Investigation of the Burn-in Phase*

The two sets of $\mathbf{x}_{\text{MMSE}}$ are presented in Figure 3.6. In Figure 3.6a, the earlier we split chains, the closer the $\mathbf{x}_{\text{MMSE}}$ gets to the truth. Especially, when we zoom into the region that has complicated structures (indicated by the red rectangle),

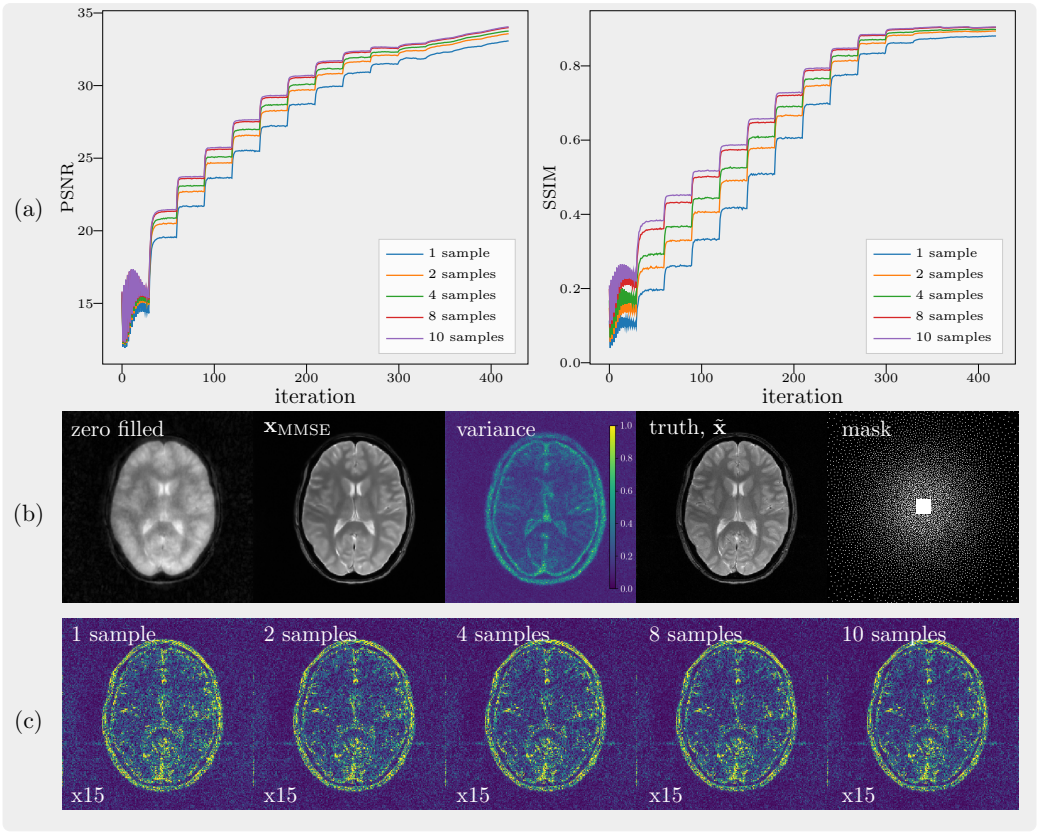


Figure 3.4: Multi-coil reconstruction with $\mathtt{NET}_1$. Results : (a) The curves of PSNR and SSIM over iterations for $\mathbf{x}_{\mathrm{MMSE}}$s estimated by averaging a different number of samples (b) Zero-filled, $\mathbf{x}_{\mathrm{MMSE}}$, variance maps, truth and mask are presented. The final PSNR and the SSIM of $\mathbf{x}_{\mathrm{MMSE}}$ are 34.05dB and 0.9050, respectively (c) The error maps between different $\mathbf{x}_{\mathrm{MMSE}}$s and the ground truth are presented.

the longer chains make fewer mistakes. The slightly distorted structure is seen in $(\mathbf{x}_{\mathrm{MMSE}}, 60)$ highlighted with blue circles. The distortion has disappeared in $(\mathbf{x}_{\mathrm{MMSE}}, 0)$ but some details are still missing. However, given more k-space data points, the longer chains do not cause a huge visual difference in the $\mathbf{x}_{\mathrm{MMSE}}$ as shown in Figure 3.6b, even though there is a slight increase in PSNR and SSIM. Although fewer data points mean more uncertainties, longer chains permit better exploration of the solution space, as shown by this experiment. Here, the image $(\mathbf{x}_{\mathrm{MMSE}}, 60)$ took about one fourth of the time (4 minutes and 30 seconds) to compute than the image $(\mathbf{x}_{\mathrm{MMSE}}, 0)$. For moderate undersampling rates, a burn-in phase is recommended for reducing computation time.

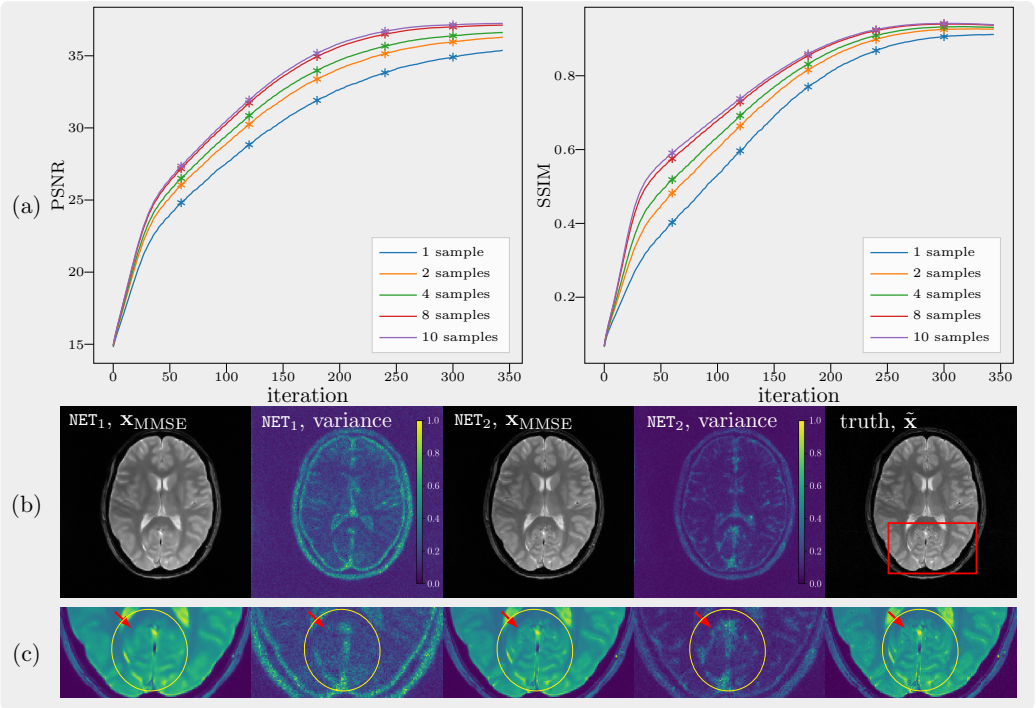


Figure 3.5: Effect of using continuous noise scales in $\texttt{NET}_2$. (a) The convergence curves of PSNR and SSIM over iterations for $\texttt{NET}_2$. (b) Reconstructed MMSE and variance maps for $\texttt{NET}_2$ and $\texttt{NET}_2$. (c) Zoomed view of selected structures (yellow circle, red arrow).

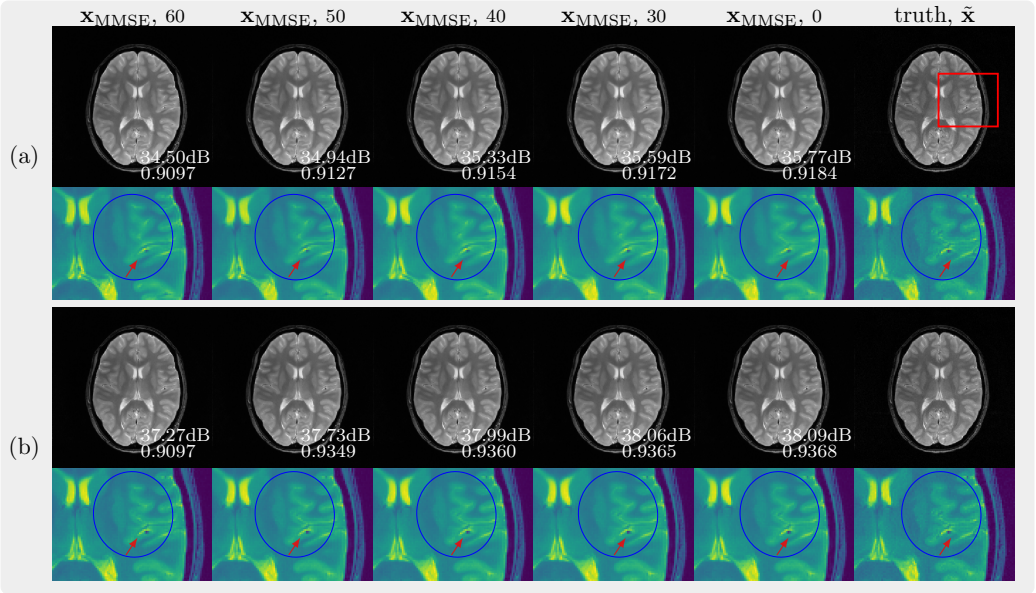


Figure 3.6: To investigate the burn-in phase the effect of splitting chains at different time points is shown for $\texttt{NET}_2$ for reconstruction with (a) 10% k-space data points and (b) 20% k-space data points.

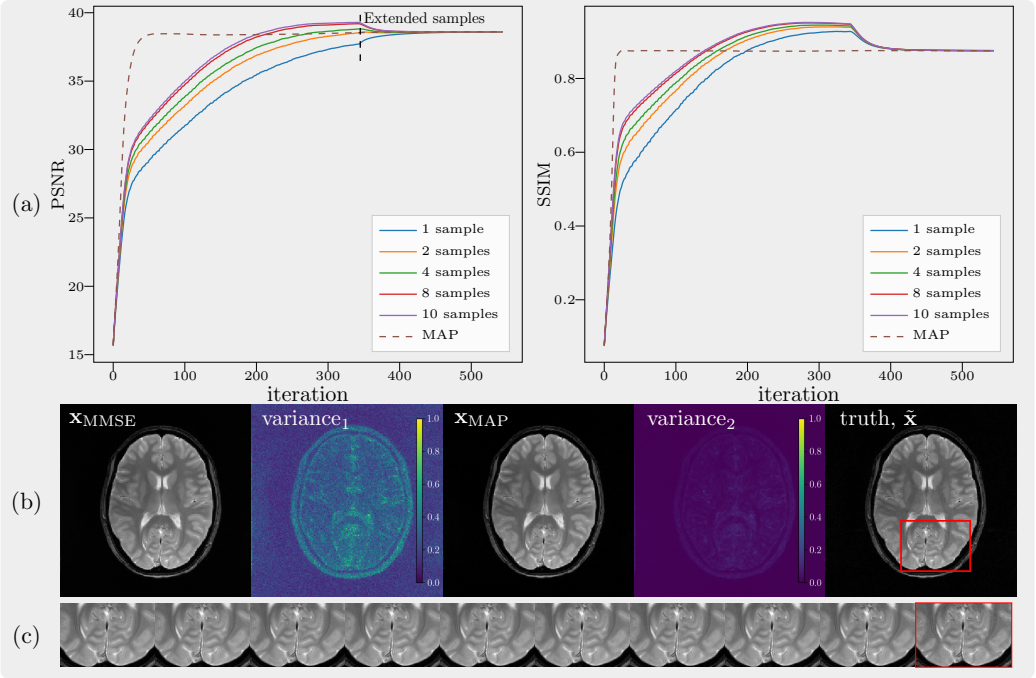


Figure 3.7: Investigation of the MAP reconstruced with $\texttt{NET}_2$. 200 extended iterations after random exploration versa a deterministic estimate of MAP that are indicated by solid and dashed lines respectively. (a) The curves of PSNR and SSIM over iterations. (b) The sub-figure variance$_1$ and variance$_2$ were computed from unextended samples and extended samples respectively. $\mathbf{x}_{\text{MAP}}$ is an extended sample. (c) The zoom-in region of 9 extended samples and the ground truth.

3.4.5 *Investigation of the MAP*

In Figure 3.7, we plotted the curves of PSNR and SSIM over extended iterations for $\texttt{NET}_2$ and presented reconstructions that are from the MMSE and MAP estimator. As indicated by zoom-in images and curves in Figure 7a and 7c, the extended samples converge to a consistent estimate of the MAP. Measured by PSNR and SSIM, the MAP has better quality than individual samples. As expected, the MMSE obtained from averaging ten (non-extended) samples has better PSNR and SSIM than the MAP.

3.4.6 *Comparison to L^1-regularized Reconstruction*

The reconstructions with different methods are presented in Figure 3.8. ℓ_1-ESPIRiT denotes the reconstruction with the `pics` command of BART toolbox using ℓ_1-wavelet regularization (0.01), which mostly recovers general structures while smoothing out some details. In $\mathbf{x}_{\text{MMSE}}$, the majority of details are recovered, and the texture is almost identical to the ground truth, although some microscopic

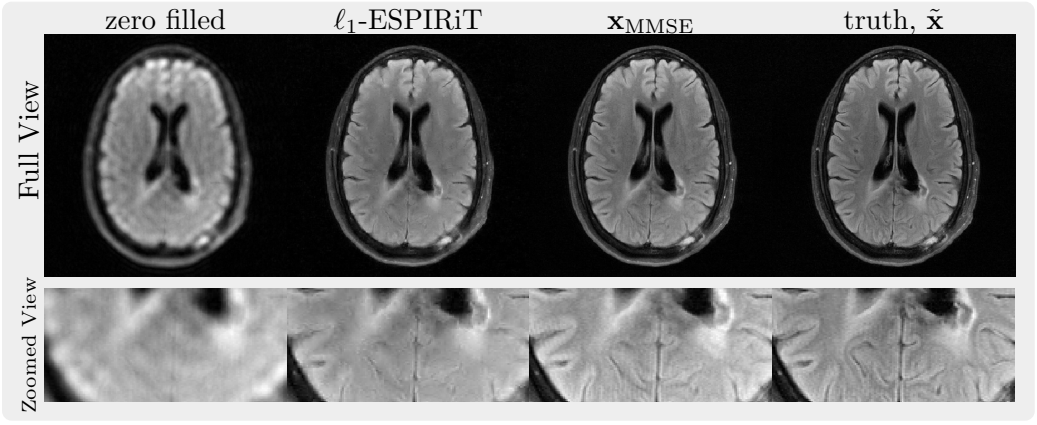


Figure 3.8: Comparison of the MMSE computed with NET_3 to the ℓ_1-wavelet regularized and zero-filled reconstruction. The high resolution image (320×320) was reconstructed from k-space data using 10-fold undersampling. The regularization parameter was set to 0.01.

structures are still missing. Each subject has 16 slices and the metrics of 3 subjects presented in Supporting Table S3 are the average over slices of each subject. It's worth mentioning that PSNR and SSIM are influenced by the value-range of a slice in the evaluation of MR images.

3.4.7 *Transferability*

Figure 3.9 shows a NET_3 trained with T2 FLAIR contrast used to reconstruct a T1 FLAIR image (red box) in comparison to a T2 FLAIR image. No loss of quality can be observed.

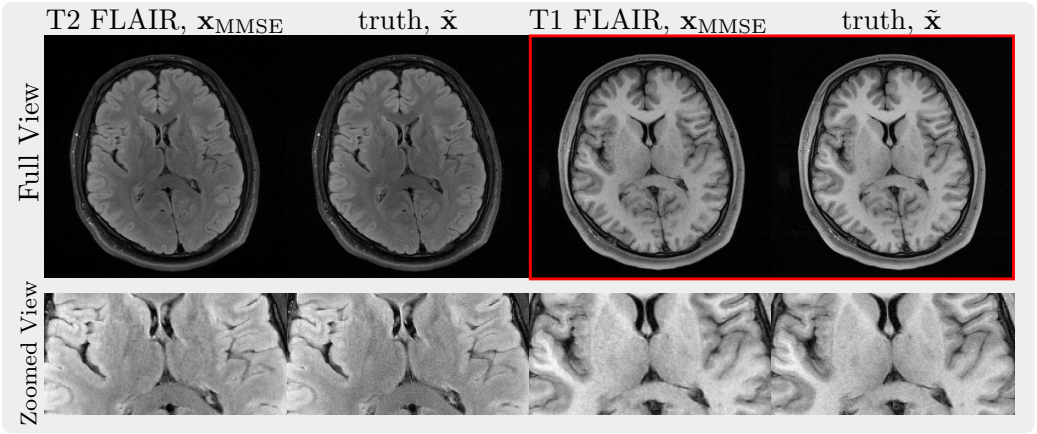


Figure 3.9: Transferability: Reconstruction of T2 and T1 FLAIR images (red box) using a Poisson-disc pattern with 8x undersampling in k-space using NET_3 trained on T2 FLAIR images.

3.4.8 *Comparison to fastMRI challenge*

As discussed in Ref. [61], the ground truth matters when computing comparison metrics. We plotted the metrics of 30 volumes against a root sum of squares (RSS) and a coil combined image (CoilComb) in Supporting Figure S3, which shows XPDNet favors RSS that was used as labels for training it while $\mathbf{x}_{\text{MMSE}}$ favors the other. Besides, the data range can be determined slice by slice or volume by volume, and the influences of that are not ignorable.

Both methods provide nearly aliasing-free reconstruction at 4 or 8-fold acceleration. However, the hallucinations appear when using 8-fold acceleration, highlighted with the green color (cf. Figure 3.10).

All in all, a deep learning-based method has enough capability to generate a realistic-looking image even when the problem is highly underdetermined as a result of undersampling, but the uncertainties inside it cannot be ignored.

3.5 DISCUSSION

Generally, the Bayesian statistical approach provides a foundation for sampling the posterior $p(\mathbf{x}|\mathbf{y})$ and a natural mechanism for incorporating the prior knowledge that is learned from images. The generative model is used to construct Markov chains to sample the posterior. The utilization of probabilistic generative models allows: 1) flexibility for changing the forward model of measurement; 2) exact sampling from the posterior term $p(\mathbf{x}|\mathbf{y})$; and 3) the estimation of uncertainty due to limited k-space data points.

Uncertainties of Reconstruction One advantage of the proposed approach over classical deterministic regularization methods is that it allows the quantification of uncertainties of the reconstruction with the variance map. That requires MCMC sampling technique. The loss of spatial information of coils leads to the failure of unfolding, as demonstrated in Section Section 3.4.1. High undersampling implies a high uncertainty about the solution, which may lead to hallucinations as observed in Ref. [59] and Figure 3.10. The regions with aliasing correspond to the high variance areas of the uncertainty map. With multiple coils, the reduction of high frequency data points in k-space leads to the loss of fine details, as demonstrated in Section Section 3.4.2. The $\mathbf{x}_{\text{MMSE}}$ represents the reconstruction with minimum mean square error and the variance map evaluates the confidence interval of $\mathbf{x}_{\text{MMSE}}$. Furthermore, it is possible to derive error bounds from the variance of the posterior as reported Ref. [62].

Overfitting and Distortion The proposed algorithm is an iterative refining procedure that starts from generating coarse samples with rich variations under large noise, before converging to fine samples with less variations under small noise. For early iterations of the algorithm, each parameter update mimics stochastic

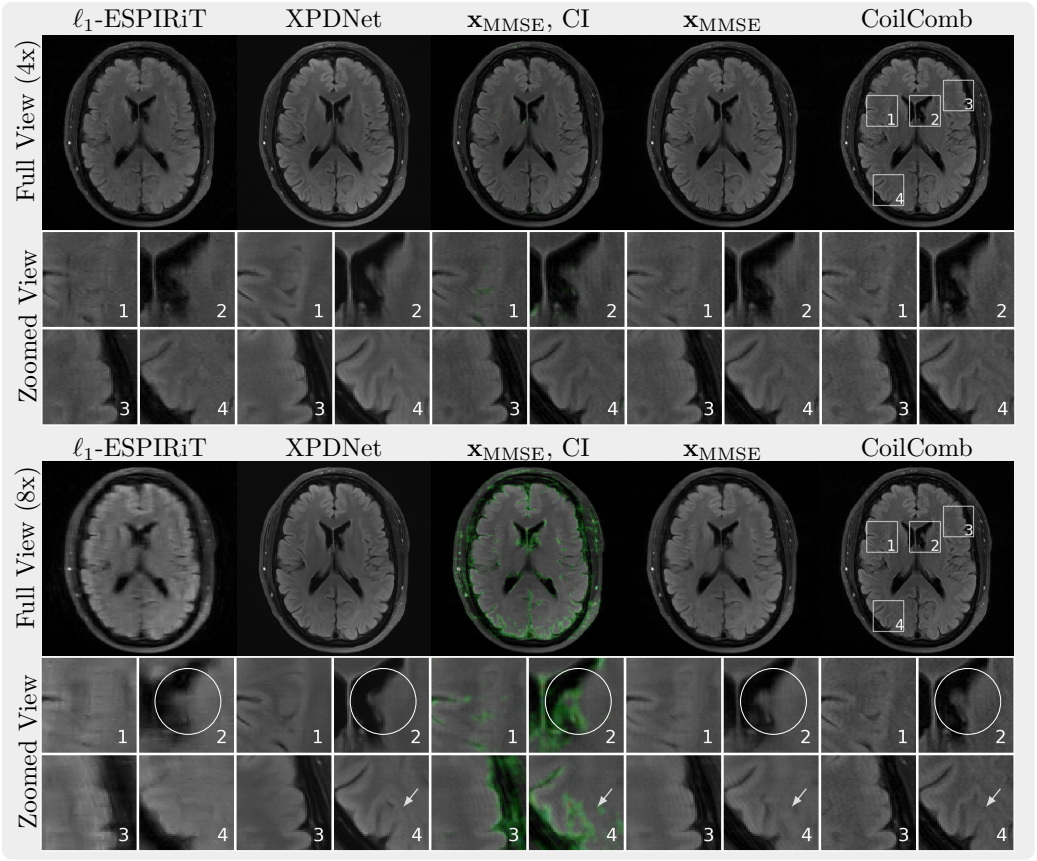


Figure 3.10: Comparison to fastMRI challenge. From the leftmost to rightmost column, reconstructions are ℓ_1-ESPIRiT, XPDNet, $\mathbf{x}_{\text{MMSE}}$ highlighted with confidence interval, $\mathbf{x}_{\text{MMSE}}$ and a fully-sampled coil-combined image (CoilComb). Hallucinations appear when using 8-fold acceleration along the phase-encoding direction (horizontal) and are highlighted with the confidence interval after thresholding. Selected regions of interests are presented in a zoomed view.

gradient descent; however, as the algorithm approaches a local minimum, the gradient shrinks and the chain produces the samples from the posterior. Lastly, we noticed that the balance between the learned transition and the data consistency plays an important role generally in the generation of realistic samples; here we refer readers to Supporting Figure S4. The larger λ, the stronger the consistency of data. Besides, we found that a large value of K is required for using the discrete noise conditional score network in 2 while a smaller value is sufficient for the continuous noise scales. While the N in 2 is larger for the continuous case, the total number of iterations in both cases is comparable.

Computational Burden The promising performance of this method comes at the price of demanding computation. It takes around 10 minutes to reproduce the results in Figure 3.5 while ℓ_1-ESPIRiT takes about 5 seconds with BART for a single slice. The possible solutions to the computation burden are to: 1) accelerate the inferencing of neural networks; 2) parallelize the sampling process when multiple chains are used; and 3) reduce the number of iterations using more efficient MCMC sampling techniques. Furthermore, reducing the scale of networks is also viable. The introduction of burn-in experiment in Section 3.4.4 is a direct way to overcome this shortcoming when the undersampling factor is moderate.

Relationship to Generative Models To our knowledge, the construction of image models to exploit prior knowledge was first introduced in Ref. [63] in which the handcrafted model which extracts edge information was used for image restoration. Following that framework, the learned generic image priors from generative perspective are investigated in Ref. [28, 64, 65], which permits more expressive modeling. In the medical imaging field, image priors learned with variational autoencoder [19, 36] and PixelCNN [20, 23] were applied to MRI image reconstruction. As a comparison to the method in Ref. [20], the result is presented in Supporting Figure S5. Compared with some unrolled network based deep learning image reconstruction methods, the application of image priors is independent of k-space data and coil sensitivities, which permits a more versatile use of the method using different k-space acquisition strategies.

Limitations PSNR and SSIM only give a partial and distorted view of image quality. The influence of the ground truth and noise properties of the background have a severe influence, as does the selected data range used for computing the metrics. Thus, rating of image quality by human readers would be an important next step in the evaluation of the technique. Also the clinical usefulness of the uncertainty maps requires further investigations. To facilitate the use in clinical studies, we implemented the sampling in the BART toolbox.[66, 67]

3.6 CONCLUSION

The proposed reconstruction method combines concepts from machine learning, Bayesian inference and image reconstruction. In the setting of Bayesian inference, the image reconstruction is realized by drawing samples from the posterior term $p(\mathbf{x}|\mathbf{y})$ using data-driven Markov chains, providing a minimum mean square reconstruction and uncertainty estimation. The prior information can be learned from an existing image database,where the generic generative priors based on the diffusion process allow for flexibility regarding contrast, coil sensitivities, and sampling pattern.

ACKNOWLEDGEMENT

We acknowledge funding by the "Niedersächsisches Vorab" funding line of the Volkswagen Foundation. We would like to thank Xiaoqing Wang for his help in preparing this manuscript as well as Christian Holme for help with our computer systems.

3.7 APPENDIX

3.7.1 *A.1 Rewrite in terms of posterior*

Because the forward diffusion is a Markov process and start at $\mathbf{x}_0$, with Bayes' rule we have

$$q\left(\mathbf{x}_i \mid \mathbf{x}_{i-1}, \mathbf{x}_0\right) = q\left(\mathbf{x}_{i-1} \mid \mathbf{x}_i\right) \frac{q\left(\mathbf{x}_i \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_{i-1} \mid \mathbf{x}_0\right)}. \tag{3.26}$$

Substituting density function into Equation 3.26 yields

$$q(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0) = q(\mathbf{x}_i|\mathbf{x}_{i-1}) \cdot \frac{q(\mathbf{x}_{i-1}|\mathbf{x}_0)}{q(\mathbf{x}_i|\mathbf{x}_0)} \tag{3.27}$$

$$= \frac{1}{\sqrt{(2\pi\beta_i^2)^{N_p}}} \cdot \frac{b_i^{2N_p}}{b_{i-1}^{2N_p}} \exp\Big[-\Big(\frac{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|^2}{\beta_i^2} + \frac{\|\mathbf{x}_{i-1} - \mathbf{x}_0\|^2}{b_{i-1}^2} - \frac{\|\mathbf{x}_i - \mathbf{x}_0\|^2}{b_i^2}\Big)\Big]. \tag{3.28}$$

Let $\frac{b_{i-1}^2 + \beta_i^2}{b_i^2} = 1$, which is satisfied with Equation 3.5, we have

$$q(\mathbf{x}_{i-1}|\mathbf{x}_i, \mathbf{x}_0) = \frac{1}{\sqrt{(2\pi\beta_i^2)^{N_p}}} \cdot \frac{b_i^{2N_p}}{b_{i-1}^{2N_p}} \exp\Big[-\Big(\frac{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|^2}{\beta_i^2} + \frac{\|\mathbf{x}_{i-1} - \mathbf{x}_0\|^2}{b_{i-1}^2} - \frac{\|\mathbf{x}_i - \mathbf{x}_0\|^2}{b_i^2}\Big)\Big] \tag{3.29}$$

$$= \frac{1}{\sqrt{(2\pi\beta_i^2)^{N_p}}} \cdot \frac{b_i^{2N_p}}{b_{i-1}^{2N_p}} \exp\Big[-\Big(\frac{\|\mathbf{x}_{i-1} - \boldsymbol{\mu}\|^2}{\beta_i^2 \cdot \frac{b_{i-1}^2}{b_i^2}}\Big)\Big], \tag{3.30}$$

where

$$\boldsymbol{\mu} = \frac{b_{i-1}^2}{b_i^2} \cdot \mathbf{x}_i + \frac{\beta_i^2}{b_i^2} \cdot \mathbf{x}_0. \tag{3.31}$$

3.7.2 *A.2 KL divergence of two Gaussian distributions*

Let $p(\mathbf{x}) = \mathcal{CN}(\boldsymbol{\mu}_1, \sigma_1^2\mathbf{I})$ and $q(\mathbf{x}) = \mathcal{CN}(\boldsymbol{\mu}_2, \sigma_2^2\mathbf{I})$ and the KL divergence is defined by

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(\mathbf{x}) \log\Big(\frac{p(\mathbf{x})}{q(\mathbf{x})}\Big)\, d\mathbf{x}.$$

Therefore,

$$\begin{aligned} D_{\mathrm{KL}}(P \parallel Q) &= \int [\log p(\mathbf{x}) - \log q(\mathbf{x})] p(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}_{p(\mathbf{x})} \left[N_p \log(\frac{\sigma_2}{\sigma_1}) + \frac{1}{\sigma_2^2} \|\mathbf{x} - \boldsymbol{\mu}_2\|^2 - \frac{1}{\sigma_1^2} \|\mathbf{x} - \boldsymbol{\mu}_1\|^2 \right] \\ &= N_p \cdot \log(\frac{\sigma_2}{\sigma_1}) + \frac{1}{\sigma_2^2} \mathbb{E}_{p(\mathbf{x})} \left[\|\mathbf{x} - \boldsymbol{\mu}_2\|^2 \right] - 1. \end{aligned}$$

where N_p is the dimensionality $n \times n \times 2$. Noting that

$$\|\mathbf{x} - \boldsymbol{\mu}_2\|^2 = \|\mathbf{x} - \boldsymbol{\mu}_1\|^2 + 2\mathbb{R}(\mathbf{x} - \boldsymbol{\mu}_1)^H(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2$$

we arrive at

$$\begin{aligned} D_{\mathrm{KL}}(P \parallel Q) =& N_p \cdot \log(\frac{\sigma_2}{\sigma_1}) + \frac{1}{\sigma_2^2} \big(\mathbb{E}_{p(\mathbf{x})} \left[\|\mathbf{x} - \boldsymbol{\mu}_1\|^2 \right] \\ &+ 2\mathbb{R}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^H \mathbb{E}_{p(\mathbf{x})} \left[\mathbf{x} - \boldsymbol{\mu}_1 \right] \\ &+ \mathbb{E}_{p(\mathbf{x})} \left[|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2 \right] \big) - 1 \\ =& N_p \cdot \log(\frac{\sigma_2}{\sigma_1}) + \frac{\sigma_1^2 + \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2}{\sigma_2^2} - 1. \end{aligned}$$

3.8 SUPPORTING INFORMATION

3.8.1 *Grid of Samples*

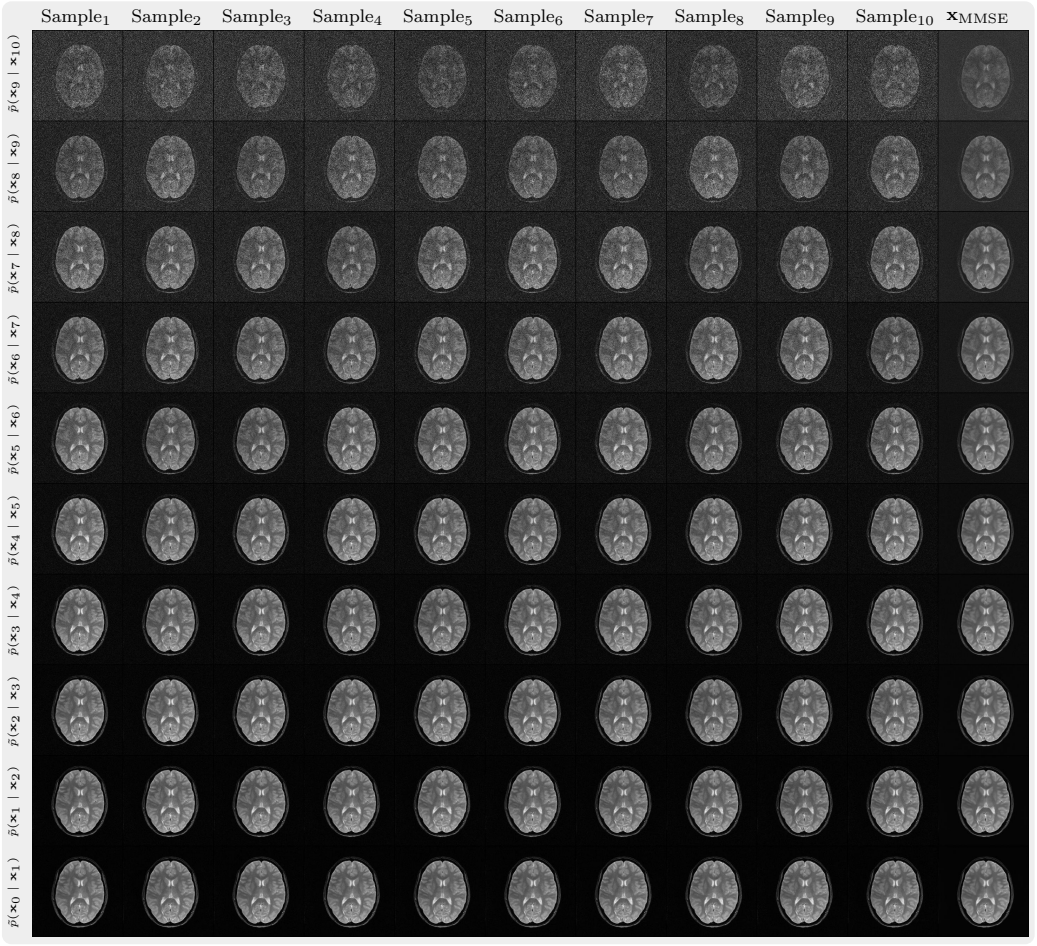


Figure 3.11: Samples and $\mathbf{x}_{\text{MMSE}}$ from intermediate distributions are presented here. Each $\mathbf{x}_{\text{MMSE}}$ is the average over 10 samples.

3.8.2 *Implementation Details of Networks*

The architecture of RefineNet[36] is U-Net based as shown below. The *ref*, *res*, *rcu*, *msf* and *crp* are abbreviations of refine block, residual block, residual convolution unit, multi-resolution fusion and chained residual pooling. All the blocks used in the neural networks are conditional on noise scales. For a discrete score network, the conditioning is achieved by instance normalization layers, while in a continuous score network, the conditioning is achieved by adding sinusoidal position embeddings. Details are listed in below Table 3.1.

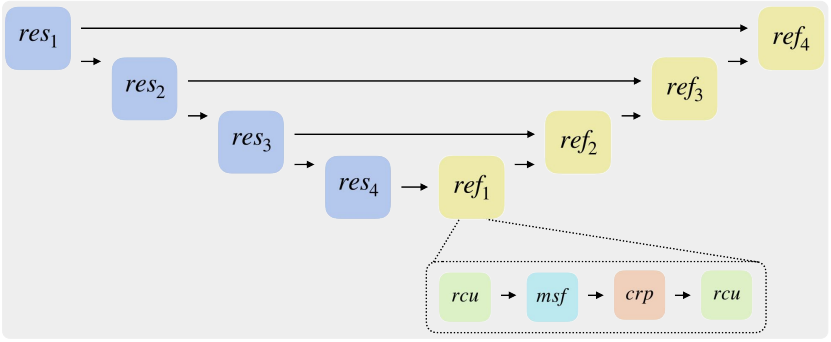


Figure 3.12: Unet-based RefineNet with refine blocks.

3.8.3 *PSNR and SSIM Metrics for Different Ground Truths and Data Ranges*

PSNR is defined via the mean squared error (MSE) and the maximum possible pixel value of the image (MAX)

$$\text{PSNR} = 20 \cdot \log_{10}(\text{MAX}) - 10 \cdot \log_{10}(\text{MSE}).$$

As shown in the following figure, the PSNR metric is influenced by the ground truth, i.e., RSS or CoilComb, and whether the maximum is computed over the slice or volume. The RSS of the volumes are the labels used to train XPDNet and the samples used to compute $\mathbf{x}_{\text{MMSE}}$ are generated slice by slice.

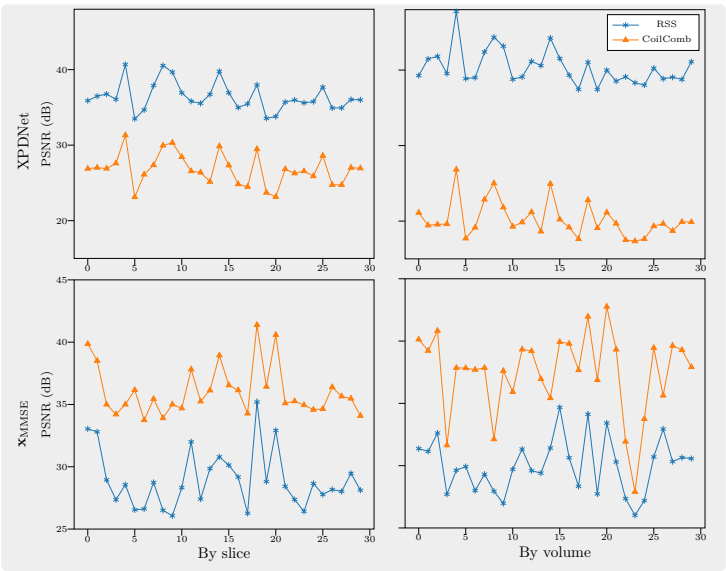


Figure 3.13: PSNR and SSIM metrics for different ground truths and data ranges.

3.8.4 *Distortion*

To investigate how λ affects the sampling process, we repeated the *more noise scales* reconstruction experiment five times with different $\lambda = \{1, 2, 3, 5, 25\}$. $\texttt{NET}_2$ was used to construct transition kernels and the other parameters in Algorithm 1 are $K = 2, N = 80$.

Figure 3.14 shows the samples that explore the distribution space with different strength of the data consistency by changing λ. Comparing samples that are reconstructed with small λ, we observed large variations between two selected samples in Figure 3.14a and Figure 3.14b and the distorted samples are far away from the truth. As λ increases, the variations decreases shown in the variance map Figure 3.14c.

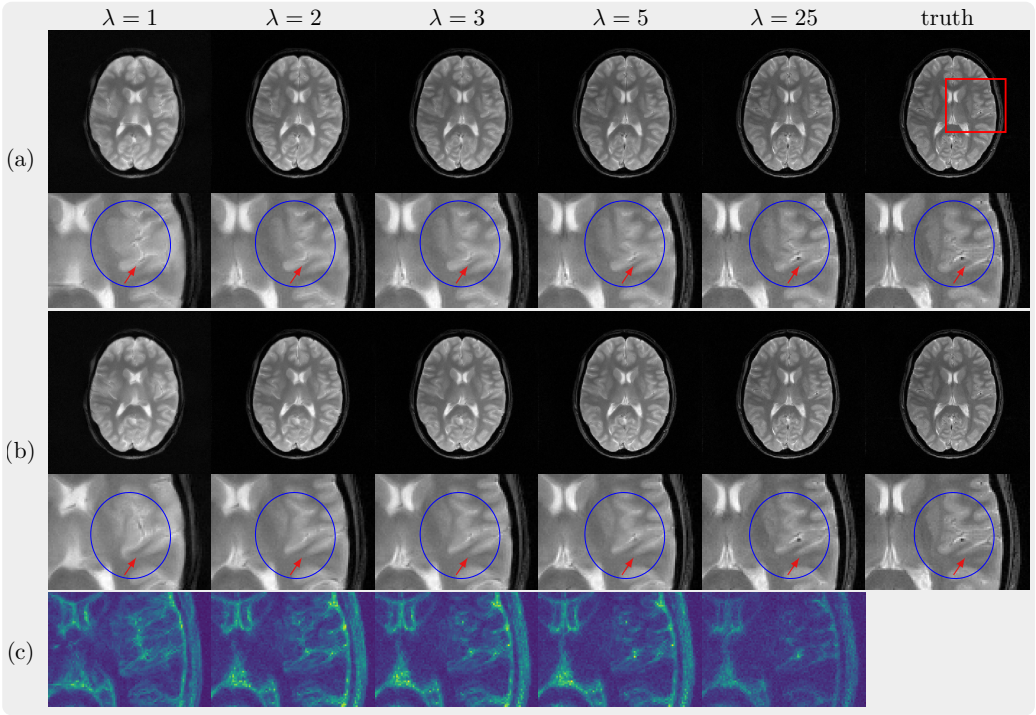


Figure 3.14: Samples reconstructed with different λ. Two selected samples with a particular λ are presented in (a) and (b). The variance maps over 10 samples reconstructed with each λ are shown in (c).

3.8.5 *Comparison of Prior-based Methods*

Figure 3.15 shows that prior-based methods for reconstruction have comparable performance even for 10x undersampling pattern as quantified with the `nrmse` metric. The image reconstructed with the combined diffusion prior and autoregressive (AR) prior preserves more of the small details than the other methods.

The `nrmse` metrics are AR: 0.0582, $\mathbf{x}_{\text{MMSE}}$: 0.0577, $\mathbf{x}_{\text{MMSE}}$+AR: 0.0559. The PSNRs are AR: 36.21, $\mathbf{x}_{\text{MMSE}}$: 36.27, $\mathbf{x}_{\text{MMSE}}$+AR: 36.55 in dB. The SSIMs are AR: 0.8731, $\mathbf{x}_{\text{MMSE}}$: 0.8516, $\mathbf{x}_{\text{MMSE}}$+AR: 0.8354.

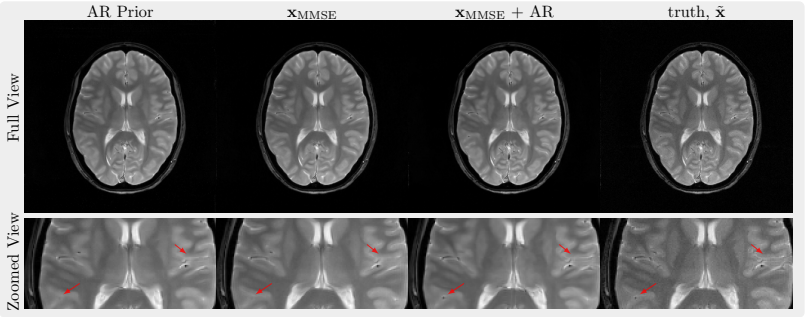


Figure 3.15: Reconstruction using different prior-based methods with Poisson-disc sampling with 10x undersampling in k-space. The arrows indicate small details lost in some reconstructions.

Table 3.1: Architectures of the score networks

$\texttt{NET}_{1/2}$
3×3 Conv2D, filters=64
CondResBlock_1, filters=64
CondResBlock_1, filters=64
CondResBlock_2, downsampling filters=128
CondResBlock_2, filters=128
CondResBlock_3, downsampling, filters=128, dilation=2
CondResBlock_3, filters=128 dilation=2
CondResBlock_4, downsampling, filters=128, dilation=4
CondResBlock_4, filters=128 dilation=4
CondRefineBlock_1, filters=128
CondRefineBlock_2, filters=128
CondRefineBlock_3, filters=64
CondRefineBlock_4, filters=64
3x3 Conv2D, filters=2
(~7M trainable parameters)

$\texttt{NET}_3$
3×3 Conv2D, filters=100
CondResBlock_1, filters=100
CondResBlock_1, filters=100
CondResBlock_2, downsampling filters=200
CondResBlock_2, filters=200
CondResBlock_3, downsampling, filters=200, dilation=2
CondResBlock_3, filters=200
SelfAttention, filters=200
CondResBlock_4, downsampling, filters=200, dilation=4
CondResBlock_4, filters=200
SelfAttention, filters=200
CondResBlock_5, downsampling, filters=200, dilation=4
CondResBlock_5, filters=200
SelfAttention, filters=200
CondRefineBlock_1, filters=200
CondRefineBlock_2, filters=200
CondRefineBlock_3, filters=200
CondRefineBlock_4, filters=100
CondRefineBlock_5, filters=100
3x3 Conv2D, filters=2
(~27M trainable parameters)

3.8.6 *Hyperparameters for Training*

The hyperparameters used for training networks are listed in Table 3.2. The indexed noise scale is $\sigma_i = \sigma(\frac{i}{N}) = \sigma_{\text{min}}(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}})^{\frac{i-1}{N-1}}$.

Table 3.2: Hyperparameters for training

	$\texttt{NET}_1$	$\texttt{NET}_2$	$\texttt{NET}_3$
$\sigma_{\text{max}}/\sqrt{2}$	0.3	0.5	0.5
$\sigma_{\text{min}}/\sqrt{2}$	0.01	0.01	0.01
N	15	100	100
batch size	10	5	5
embedding size	-	128	200
Fourier scale	-	16	16
learning rate	0.0001	0.0001	0.0001
activation	elu	elu	elu
optimizer	adam	adam	adam

Performance on Open Dataset

Table 3.3: Average PSNR (dB) and SSIM(%) for test subjects

Sampling	ℓ_1-ESPIRiT			$\mathbf{x}_{\text{MMSE}}$		
PSNR	sub_1	sub_2	sub_3	sub_1	sub_2	sub_3
10%	32.27 ± 1.03	34.67 ± 1.77	34.03 ± 2.98	34.07 ± 0.91	36.12 ± 1.85	35.65 ± 3.37
20%	33.69 ± 0.86	35.60 ± 2.18	35.17 ± 3.41	35.04 ± 0.89	37.15 ± 2.08	36.58 ± 3.84
SSIM						
10%	79.69 ± 3.28	82.74 ± 8.57	79.98 ± 12.40	84.21 ± 3.24	86.06 ± 6.99	83.24±12.07
20%	83.68 ± 3.38	84.60 ± 9.20	82.70 ± 12.93	86.38 ± 3.04	88.26 ± 7.07	84.85±12.99

4

GENERATIVE IMAGE PRIORS FOR MRI RECONSTRUCTION TRAINED FROM MAGNITUDE-ONLY IMAGES

The chapter is a reprint of the original manuscript.

G. Luo, X. Wang, M. Blumenthal, M. Schilling, EHU Rauf, R. Kotikalapudi, N. K. Focke, M. Uecker. "Generative Image Priors for MRI Reconstruction Trained from Magnitude-Only Images"

which was under revision at *Magnetic Resonance in Medicine* at the time of thesis submission and was shared on arXiv:2308.02340 under a CC-BY 4.0 license.

GL conceptualized this work, implemented the algorithm, designed and performed the numerical experiments, and analyzed the results. *XW* provided initial codes for nonlinear reconstruction using fast iterative shrinkage-thresholding method (FISTA). *MB* provided functionalites for loading TensorFlow computation graph. *MS* provided functionalites for creating NII files. *ER*, *RK* and *NF* evaluated the image quality. *MU* supervised this work. *GL* and *MU* wrote the manuscript.

ABSTRACT

Purpose: In this work, we present a workflow to train generic and robust generative image priors from magnitude-only images. The priors can then be used for regularization in reconstruction to improve image quality.
Methods: The workflow begins with the preparation of training datasets from magnitude-only MR images. This dataset is then augmented with phase information and used to train generative priors of complex images. Finally, trained priors are evaluated using both linear and nonlinear reconstruction for compressed sensing parallel imaging with various undersampling schemes.
Results: The results of our experiments demonstrate that priors trained on complex images outperform priors trained only on magnitude images. Additionally, a prior trained on a larger dataset exhibits higher robustness. Finally, we show that the generative priors are superior to L^1-wavelet regularization for compressed sensing parallel imaging with high undersampling.
Conclusion: These findings stress the importance of incorporating phase information and leveraging large datasets to raise the performance and reliability of the generative priors for MRI reconstruction. Phase augmentation makes it possible to use existing image databases for training.

4.1 INTRODUCTION

Regularizing the inverse problem for parallel MRI reconstruction is an effective and flexible approach for improving image quality, especially when the obtained k-space is highly undersampled in order to shorten the scan time. The prior knowledge that images are sparse in a transform domain as used in compressed sensing is known as ℓ^1-norm regularization[1, 2]. Combined with incoherent sampling, this allows recovery of images from moderately undersampled k-space data with clinically acceptable quality[68, 69].

The application of deep learning makes it possible to further increase undersampling without compromising image quality by leveraging the learned prior information from a training dataset. Popular methods can be classified into three main categories: supervised methods[3, 4], where the neural network is a result of unrolling an iterative algorithm trained with labels and used to predict the reconstruction, self-supervised methods[70, 71] that involve splitting the acquired k-space data of a scan into two disjoint sets where only the first set is used for reconstruction and the second set provides supervision, and decoupled methods[19, 35, 20], where a generative model or a denoiser is trained to learn the empirical distribution of data which is then used in a conventional iterative reconstruction method. In the following, we will refer to a generative model also as a prior.

Training in supervised methods based on unrolled iterative algorithms requires not only fully sampled k-space data, but also pre-defined sampling patterns and precomputed coil sensitivities. The prior knowledge learned in this way then pertains to these pre-defined settings. However, protocol settings for clinical and research are changed often, and the preparation of reference data, which is used as labels for training, is costly. Decoupled methods are able to avoid these constraints, and the learned prior can even be transferred to new scenarios such as different contrasts[72].

As a crucial part of decoupled methods, the use of generative models, such as variational autoencoders and autoregressive models, was investigated previously by formulating the linear reconstruction problem for accelerated MRI from the Bayesian perspective and solving it via maximum a posterior (MAP) estimation [19, 20]. More recently, diffusion models emerged as effective priors for MRI reconstruction and were combined with Monte Carlo methods that sampling the posterior[43, 45, 72]. However, their performance is heavily dependent on the size and quality of the training dataset and the computational resources available.

For this reason, it is desirable to use existing databases of MR images for training. But as shown here, training from magnitude-only images leads to inferior priors. This work therefore proposes a new approach to construct priors using magnitude-only training images as illustrated in Figure 4.1. The workflow begins

with the preparation of training datasets from magnitude-only MR images. This dataset is then augmented with phase information and used to train generative priors on complex images. Finally, trained priors can be used with both linear and nonlinear reconstruction for compressed sensing parallel imaging. The contributions of our work are:

Complex vs. magnitude-only priors: We demonstrate that priors trained on complex images are superior to priors trained only on magnitude images.

Phase augmentation: We leverage a diffusion model trained on a small dataset (~1000 images) of complex images to augment a much larger dataset (~80k images) for which the phase information of the image is not available.

Robustness: We show that we can train more robust generative priors by incorporating knowledge from a larger training dataset, which contains a diverse range of images. Such a database can be obtained by phase augmentation of magnitude images which are readily available.

Flexibility: By integrating the priors as regularization terms into existing reconstruction techniques, we maintain the flexibility of existing reconstruction algorithms that can be used with various undersampling schemes and receive coils.

Parts of this work have been presented in Refs. 73, 74.

4.2 THEORY

4.2.1 *Linear and nonlinear reconstruction*

The reconstruction in parallel imaging can be formulated as an inverse problem

$$F(\mathbf{x}, \mathbf{c}) := (\mathcal{F}_S(\mathbf{x} \odot c_1), \cdots, \mathcal{F}_S(\mathbf{x} \odot c_N)) = \mathbf{y}\,, \tag{4.1}$$

where $\mathcal{F}_S$ is an undersampled multi-channel Fourier transform operator and the correspondingly obtained k-space data is $\mathbf{y} = (y_1, \cdots, y_{nc})^T$; $\mathbf{y} \in \mathbb{C}^{d \times nc}$, $\mathbf{x} \in \mathbb{C}^{n \times n}$ denotes the image content and $\mathbf{c} = (c_1, \cdots, c_{nc})^T$; $\mathbf{c} \in \mathbb{C}^{n \times n \times nc}$ denotes the coil sensitivities. Equation 4.1 can be solved in the following two ways.

One common way for MR image reconstruction is to predetermine the coil sensitivities $\mathbf{c}$ from a reference scan or from a fully sampled k-space center. Following coil estimation, we can solve the linear inverse problem using by optimization regularized least-squares functional

$$\min_{\mathbf{x}} \frac{1}{2} \|F_{\mathbf{c}}(\mathbf{x}) - \mathbf{y}\|^2 + \alpha R(\mathbf{x}), \tag{4.2}$$

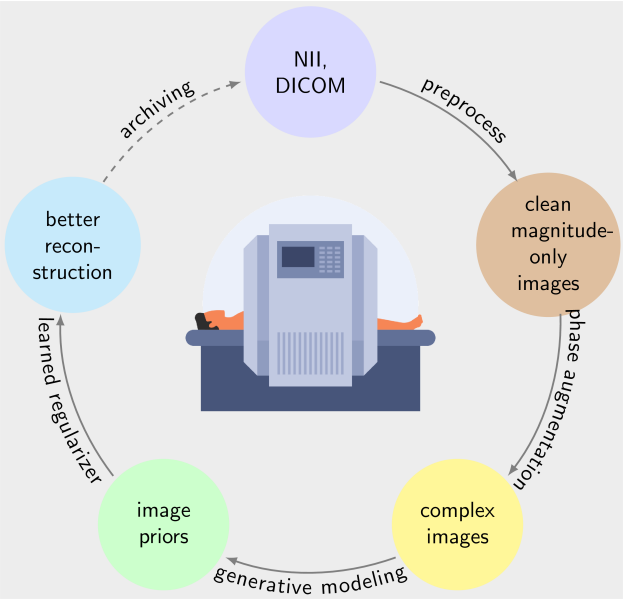


Figure 4.1: The proposed workflow for extracting prior knowledge and using it for regularization in image reconstruction. It comprises data preparation, phase augmentation, generative modeling, and concludes with the use as learned regularizers in reconstruction.

where $F_{\mathbf{c}}(\mathbf{x}) := F(\mathbf{x}, \mathbf{c})$ is a linear operator and $R(\mathbf{x})$ is the regularization term employing prior knowledge about the image, such as ℓ^2 regularization[75], total variation[2], ℓ^1-sparsity[1], or a learned log-likelihood function[20].

Alternatively, both image and coil sensitivities can be jointly estimated from the same acquired data[76, 13]. Ref. 13 formulates MR image reconstruction as a nonlinear inverse problem and proposes to solve Equation Equation 4.1 using the Iteratively Regularized Gauss Newton Method (IRGNM). This method linearizes the nonlinear inverse problem at each Gauss Newton step k, and estimates the update $\delta\mathbf{m} := (\delta\mathbf{x}, \delta\mathbf{c})$ for the pair $\mathbf{m}^k := (\mathbf{x}^k, \mathbf{c}^k)$ by minimizing a regularized least-squares functional for the linearized sub-problem

$$\begin{aligned}\min_{\delta\mathbf{m}} \frac{1}{2}\|F'(\mathbf{m}^k)\delta\mathbf{m} \;+\; F(\mathbf{m}^k) \;-\; \mathbf{y}\|^2 \\ +\; \beta^k \mathcal{W}(\mathbf{c}^k \;+\; \delta\mathbf{c}) \;+\; \alpha^k R(\mathbf{x}^k \;+\; \delta\mathbf{x})\,.\end{aligned} \tag{4.3}$$

Here, $\mathcal{W}(\mathbf{c}) = \|w \odot \mathcal{F}\mathbf{c}\|^2$ is a penalty on the high Fourier coefficients of the coil sensitivities and $R(\mathbf{x})$ is a regularization term on the image $\mathbf{x}$, e.g., ℓ_2-norm[13], ℓ_1-sparsity in the wavelet domain[14], or total variation[77]. The α_k and β_k are decreasing in each iteration step.

4.2.2 *Learned priors as regularization*

Learned prior knowledge can be used for regularization in image reconstruction. Generative priors, such as variational autoencoder, autoregressive models (e.g., PixelCNN), and diffusion models, are used to incorporate empirical knowledge about images into iterative optimization algorithms. We want to use generative priors directly as a drop-in replacement for conventional priors in existing image reconstruction algorithms, which are often based on proximal methods.

The proximal operator for the log-prior $\log p(\mathbf{x})$ is defined as

$$\text{prox}_t(\mathbf{z}) \;=\; \arg\min_{\mathbf{x}} \frac{1}{2t}\|\mathbf{x}-\mathbf{z}\|^2 + \log p(\mathbf{x}) \;. \tag{4.4}$$

When the mapping above is not analytically computable, the proximal operator could be approximated by minimizing Equation 4.4 using gradient descent. Note that the minimization problem for the proximal operator is the same as for a denoising problem for complex Gaussian noise[78]. Assuming some regularity of the prior and noise-like properties of the error during reconstruction, the gradient can be expected to always point in the same directions towards denoised images. Therefore, the optimality condition at the solution is approximately

$$0 \approx \frac{1}{t}(prox_t(\mathbf{z}) - \mathbf{z}) + \nabla_{\mathbf{x}} \log p(\mathbf{x}) \;.$$

The solution is then equivalent to a single gradient-descent step with an arbitrary initial guess and unit step size

$$\text{prox}_t(\mathbf{z}) \approx \mathbf{z} - t\nabla_{\mathbf{x}} \log p(\mathbf{x}) \;,$$

which simply yields a gradient-descent step for the log-prior in the overall algorithm. In this work, two types of log-priors are used which are described below.

PixelCNN prior: This prior is formulated using a joint distribution over the elements of an image vector

$$\log p(\mathbf{x}; \text{NET}(\hat{\Theta}, \mathbf{x})) = \log p(x^{(1)}) \prod_{i=2}^{n^2} p(x^{(i)} \mid x^{(1)}, .., x^{(i-1)}) \;, \tag{4.5}$$

where the neural network $\text{NET}(\hat{\Theta}, \mathbf{x})$ predicts the distribution parameters of a mixture of logistic distributions that is used to describe every pixel and where the dependencies between the channels for the real and imaginary parts are described with nonlinear dependencies[20]. All these parameters used to probabilistically model the image are predicted by a causal network[23] that encodes the relation-

ship between pixels as formulated in item 4.5. The gradient of $\log p(\mathbf{x}; \mathrm{NET}(\hat{\Theta}, \mathbf{x}))$ with respect to $\mathbf{x}$ can be computed by back-propagation through the neural network.

Probabilistic diffusion prior: The diffusion probabilistic model proposed in Ref. 27 is constructed with a forward Markovian process and a learned reverse process. The forward process is to gradually transfer a data distribution $q(\mathbf{x}_0)$ to a smoother known distribution $q(\mathbf{x}_N)$, e.g., a Gaussian distribution, by adding noise to data points. The reverse process is to undo this forward process with learned reverse transitions, which are described as

$$p_\theta(\mathbf{x}_{i-1} \mid \mathbf{x}_i) = \mathcal{CN}(\mathbf{x}_{i-1} \mid \mu_\theta(\mathbf{x}_i, i), \tau_i^2 \mathbf{I}) \,, \tag{4.6}$$

where τ_i can be computed from the noise scales σ_i at each step and $\mu_\theta(\mathbf{x}_i, i)$ can be understood as a denoised image based on the smoothed prior at each noise level. Instead of learning this distribution directly, the gradient of the log-prior is learned for all noise scales

$$\nabla_{\mathbf{x}_i} \log p_{\boldsymbol{\theta}} \left(\mathbf{x}_{i-1} \mid \mathbf{x}_i\right) = \frac{1}{\tau_i^2} \left(\sigma_i^2 - \sigma_{i-1}^2\right) \mathbf{s}_{\boldsymbol{\theta}} \left(\mathbf{x}_{i-1}, i\right) \,,$$

where $\mathbf{s}_\theta(\mathbf{x}_i, i)$ is a trained score network[28], which is computationally efficient because it avoids backpropagation. We refer the reader to Ref. 72 for details about this method. The reverse transitions start with σ_{max} and end at a $\sigma_{min} \approx 0$. In this work we use $\sigma_{max} = 0.3$ and $\sigma_{min} = 0.01$, and

$$\sigma_i = \sigma_{min} + \sigma_{max} \cdot \log(1 + (1 - i/N) \cdot (\mathsf{e} - 1)) \,.$$

Here, e is Euler constant and N is the number of noise scales which in this work also corresponds to the total number of iterations of the reconstruction algorithm.

For linear reconstruction, the two regularization terms can be directly plugged into proximal optimization algorithms available in image reconstruction frameworks, such as the Fast Iterative Soft-Thresholding Method (FISTA)[79], or the Alternating Direction Methods of Multipliers (ADMM)[80].

Similarly, for nonlinear inverse problems, we can apply the regularization terms to the linearized sub-problem in Equation 4.3. In nonlinear reconstruction, the image content $\mathbf{x}$ is usually smooth at early Gauss Newton steps and the distribution of $\mathbf{x}$ is far from the learned empirical distribution. Correspondingly, in this work, the Gauss Newton optimization is split into two stages. In the first stage, an ℓ_2-norm regularization is applied, and the method of conjugate gradients (CG) is used to minimize Equation 4.3. In the second stage, i.e., the later Gauss-Newton steps, FISTA is utilized with the proximal operators. The entire algorithm is outlined in 2.

Algorithm 2 Two-stage IRGNM for NLINV reconstruction

Inputs:
  y - kspace data, n - MaxIter, r - RegIter
Initialization:
  $\mathbf{x}^0 = \mathbf{1},\ \mathbf{c}^0 = \mathbf{0},\ \alpha^0 = 1,\ \beta^0 = 1$
for k in $\{0, \ldots, n-1\}$ **do**
  if $k < n - r$ **then**
    $R(\mathbf{x}) = \|\mathbf{x}\|^2$
    Estimate $\delta\mathbf{x},\ \delta\mathbf{c}$ in Equation 4.3 with CG
  else
    $R(\mathbf{x}) = \log p(\mathbf{x})$ or $\|\Psi\mathbf{x}\|^1$
    Estimate $\delta\mathbf{x},\ \delta\mathbf{c}$ in Equation 4.3 with FISTA
  end if
  $\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x},\ \mathbf{c}^{k+1} = \mathbf{c}^k + \delta\mathbf{c}$
  $\alpha^{k+1} = \max(\alpha_{\min},\ \alpha^k/2)$
  $\beta^{k+1} = \beta^k/2$
end for

4.3 METHODS

In this section, we first describe how we implemented the proposed workflow, as shown in Figure 4.1, for extracting prior knowledge from an image dataset and then how we use the learned prior for regularization in image reconstruction. We detail how we evaluated the performance of the priors used for image regularization in different settings.

4.3.1 *Preprocessing of the training dataset*

As for training data, we use human brain images from the Autism Brain Imaging Data Exchange (ABIDE)[81, 82]. After we downloaded the dataset, which comes in 3D volumes in NII[1] format, we performed the following steps to preprocess it.

1. Load each 3D volume and resample it with the conform function of NiBabel[83] to make its axial plane have a size of 256×256.
2. Split the volume into 2D image slices that are oriented in axial plane.
3. Add background Gaussian noise ($\mu = 0.003, \sigma = 5$) to all slices, and then normalize every slice by dividing by its maximum pixel value.
4. Crop a 30×30 patch from the XY-corner of each normalized slice and compute the mean μ and standard deviation σ over all pixels of the patch.

1 The Neuroimaging Informatics Technology Initiative (NIfTI) is an open file format commonly used to store brain imaging data obtained using Magnetic Resonance Imaging methods.

5. Exclude slices from phase augmentation when mean $\mu < 0.04$ and standard deviation $\sigma < 0.0061$.

4.3.2 *Phase augmentation*

ABIDE images are provided solely as magnitude images without phase information. The magnitude of an MR image is determined by the proton density, relaxation effects, and receive fields, while the phase is affected by the phase of the receive field, inhomogeneities of the static field, eddy currents, and chemical shift. Phase augmentation can be used to add a phase to obtain more realistic complex-valued images. Here, we describe a procedure to obtain new samples with phase information from magnitude images using a prior previously trained prior for complex-valued images. The method is based on previous research[27] for the sampling of a posterior. Given the likelihood term of the magnitude $p(\mathbf{m}|\mathbf{x})$ and a prior for complex-valued images $p(\mathbf{x})$, the posterior of the complex image is proportional to $p(\mathbf{x}|\mathbf{m}) \propto p(\mathbf{x}) \cdot p(\mathbf{m}|\mathbf{x})$ where

$$\bar{\mathbf{m}} = \sqrt{\mathbf{x}_r^2 + \mathbf{x}_i^2} \,,$$

and $\mathbf{x}_r$ and $\mathbf{x}_i$ are the real part and imaginary part, respectively. The likelihood term for a given magnitude image $\mathbf{m}$ is $p(\mathbf{m}|\mathbf{x}) = \delta(\mathbf{m} - \sqrt{\mathbf{x}_r^2 + \mathbf{x}_i^2})$. To be able to apply gradient-based methods, we approximate this with a narrow Gaussian distribution

$$p(\mathbf{m}|\mathbf{x}) \propto \exp\left(-\epsilon \left\| \mathbf{m} - \sqrt{\mathbf{x}_r^2 + \mathbf{x}_i^2} \right\|_2^2\right).$$

Specifically, we initialize samples with random complex Gaussian noise and then transfer them gradually to the distribution of complex images with learned transition kernels $p_\theta(\mathbf{x}_n|\mathbf{x}_{n+1})$. We run unadjusted Langevin iterations sequentially at each intermediate distribution

$$\mathbf{x}_n^{k+1} \leftarrow \mathbf{x}_n^k + \frac{\gamma}{2} \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_n^k \mid \mathbf{x}_{n+1}^{\mathrm{K}}) + \frac{\gamma}{2} \nabla_{\mathbf{x}} \log p(\mathbf{m}|\mathbf{x}_n^k) + \sqrt{\gamma}\mathbf{z} \,.$$

Here, $\mathbf{z}$ is complex Gaussian noise, which introduces random fluctuations and γ controls the step size of the Langevin algorithm. The sampling algorithm was implemented with TensorFlow and used with the pre-trained generative model NET_1 from Ref. 72, which was trained on the small dataset from Ref. 20. For each magnitude image five complex images were generated.

Table 4.1: Datasets and computational resources used to train the six different priors used in this work.

Prior		Model	Phase	Nr. of Images	MR Contrasts	GPUs	Parameters	Time × epochs
P_{SC}	(small, complex)	PixelCNN	preserved	1000	T_1, T_2, T_2-FLAIR, T_2^*	4×A100, 80G	~22M	~40s × 500
P_{SM}	(small, magnitude)	PixelCNN	not available	1000	T_1, T_2, T_2-FLAIR, T_2^*	4×V100, 32G	~22M	~144s × 500
P_{LM}	(large, magnitude)	PixelCNN	not available	23078	MPRAGE	4×A100, 80G	~22M	~748s × 100
P_{LC}	(large, complex)	PixelCNN	generated	23078	MPRAGE	3×A100, 80G	~22M	~1058s × 100
D_{SC}	(SMLD, complex)	Diffusion	generated	79058	MPRAGE	4×A100, 80G	~8M	~2330s × 50
D_{PC}	(DDPM, complex)	Diffusion	generated	79058	MPRAGE	8×V100, 32G	~8M	~1430s × 200

4.3.3 *Training of priors*

In total, we trained six priors in this work. The PixelCNN priors, P_{SC} and P_{SM}, were trained on the small brain image dataset used in Ref. 20 using complex and magnitude images, respectively. P_{LM} and P_{LC} were trained on a subset of the preprocessed ABIDE dataset corresponding of 500 volumes and the corresponding phase-augmented complex images, respectively. We also trained two diffusion priors, SMLD (D_{SC}) and DDPM (D_{PC}) with phase-augmented images using the full ABIDE dataset with 1206 volumes.

During the training, images were normalized to have a maximum magnitude of one and then subjected to random mirroring, flipping, and rotation prior to being fed into the neural network. Complex images were fed as two-channel maps (i.e., real and imaginary), and magnitude images were fed as single-channel maps. The networks used for PixelCNN and the diffusion models were implemented with TensorFlow (TF) and the optimizer ADAM was used for all training tasks, which was performed using multi-GPU systems using different GPUs (Nvidia Corporation, Santa Clara, CA, USA).

PixelCNN prior: We trained this generative model by maximizing the probability of the joint distribution over all the pixels in the image using the discretized logistic mixture distribution loss proposed in Ref. 23.

Diffusion prior: There exist two types of diffusion models that are based on the denoising score matching method[47], namely, denoising Score Matching with Langevin Dynamics (SMLD)[25] and Denoising Diffusion Probabilistic Models (DDPM)[84]. Both are unified in a common framework described in Ref. 28. We train diffusion priors using both SMLB and DDPM with the same Refine-Net[85] architecture also used in Ref. 72. The loss function used to train the score network $\mathbf{s}_\theta(\mathbf{x}_i, i)$ is given by

$$\theta^* = \arg\min_\theta \sum_i \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_{\mathbf{x}_i|\mathbf{x}_0} \left[\lambda_i \left\| \mathbf{s}_\theta(\mathbf{x}_i, i) - \nabla_{\mathbf{x}_i} \log p\left(\mathbf{x}_i \mid \mathbf{x}_0\right) \right\|_2^2 \right]$$

where λ_i is the weighting function described in Ref. 28.

Information about priors and training is detailed in Table 4.1.

4.3.4 *Experimental evaluation*

In this section, we use Berkeley Advanced Reconstruction Toolbox (BART)[86] to evaluate the trained priors using Parallel Imaging Compressed Sensing (PICS) and Nonlinear Inversion (NLINV). The corresponding commands have an option for loading an exported TensorFlow computation graph and using it for regularization. The exported graph was wrapped into a nonlinear operator and then used in the proximal mapping step[67]. For the linear reconstruction using PICS, the coil sensitivities are estimated with ESPIRiT[12]. In the nonlinear reconstruction using NLINV, 2 was implemented with the nonlinear operator framework[67].

We then performed different experiments using PICS and NLINV for the six priors using different sampling patterns in comparison to zero-filled, ℓ_2, ℓ_1-wavelet reconstructions and coil-combined images form fully sampled data. Here, we used a T_1-weighted k-space from the test dataset used in Ref. 20. We additionally performed a study using quantitative image quality metrics using 3D MPRAGE data to evaluate the impact of the size of the training dataset, and performed an evaluation study with human readers for six fully sampled 3D TurboFLASH datasets as described below.

The influence of phase maps: We performed retrospective reconstruction using all six priors. Three types of undersampling pattern were used in this retrospective experiment, including five-fold acceleration along phase direction, two-times and three times acceleration along frequency and phase direction, respectively, and 8.2-times undersampling using Poisson-disc sampling. While the acceleration along frequency-encoding direction is not realistic, we use it to explore how the priors handle different 2D different patterns. In a later experiment below, we then use 3D k-space acquisitions where these sampling pattern are feasible.

The influence of the size of dataset: We performed the reconstruction using P_{SC} and P_{LC}. The k-space data were acquired from the brain of a healthy volunteer using MPRAGE sequence on 3T Siemens Skyra scanner (Siemens Healthineers, Erlangen, Germany) with 16-channel head coils. The protocol parameters were: TE = 2.45 ms, flip angle $\alpha = 8°$, TI = 900 ms, and TR = 2000 ms, 4/5 partial parallel Fourier imaging, and 2-fold acceleration along one phase encoding direction. This acquired 3D volume has dimensions $256 \times 256 \times 224$ and isotropic voxel size of 1 mm. We further undersampled the acquired 3D k-space data two and three times along two phase-encoding directions with the central region of size 30×25 reserved. The reconstruction was performed slice-by-slice in a 2D plane. To quantitatively assess the robustness of the priors, we computed PSNR and SSIM for the P_{LC}-regularized reconstruction from 4 and 6-times undersampled k-space against the P_{LC}-regularized reconstruction from 2-times undersampled k-space

and compared it to PSNR and SSIM for the ℓ_2-wavelet regularization and the prior P_{SC}.

3D reconstruction quality: For reconstruction of 3D data sets, we chose the two diffusion models (D_{SC} and D_{PC}) that are less computationally expensive than PixelCNN. The k-space data was acquired from 6 volunteers using a 3D TurboFLASH sequence (TE = 3.3 ms, TR = 2250 ms, TI = 900 ms, flip angle $\alpha = 9°$) using a 3T Siemens Prisma scanner and a 64-channel head-coil. These acquired 3D volumes have dimensions $256 \times 256 \times 176$ and isotropic size of 1 mm. We undersampled the acquired 3D k-space data using a prospectively feasible Poisson-disc pattern with 8.2-time undersampling. When applying the prior during reconstruction, all slices in the axial plane are padded to a size of 256×256 Following this, the prior is applied on all slices in parallel and the computed gradient will be resized to the original image size. The reconstructed volumes were blindly evaluated by three clinicians with variable experience in neuroimaging (~20 years, ~10 years, ~5 years). ℓ_1-wavelet reconstructions and a reference reconstructed from fully sampled k-space data by coil-combination were included in this evaluation study. The grading scale used in this study ranged from 5 to 1, where a score of 5 represents "excellent" image quality and a 1 denotes "bad" image quality.

4.4 RESULTS

Figure 4.2(a) shows three magnitude images of different quality from the ABIDE dataset and the corresponding magnitude and phase maps of complex-valued images generated by phase augmentation. The magnitude part of the generated images stays very close to the original image but exhibits a bit less noise. The phase of the generated images maps is smooth and looks realistic with some random variations as expected.

In Figure 4.2(b), we present the magnitude and phase from images reconstructed using PICS with ESPIRiT sensitivities and NLINV using the same P_{SC} prior for 8.2-times undersampled Poisson-disc k-space data in comparison to a zero-filled image and fully-sampled reference. For both methods, similar reconstructions with excellent quality can be obtained for this prior for linear and non-linear reconstruction. This prior trained from a small dataset of complex-valued images will serve as a baseline for the other learned priors.

4.4.1 *The influence of phase maps*

Figure 4.3 presents the magnitude and phase of images that are reconstructed using PICS with priors trained from magnitude image, complex images with pre-

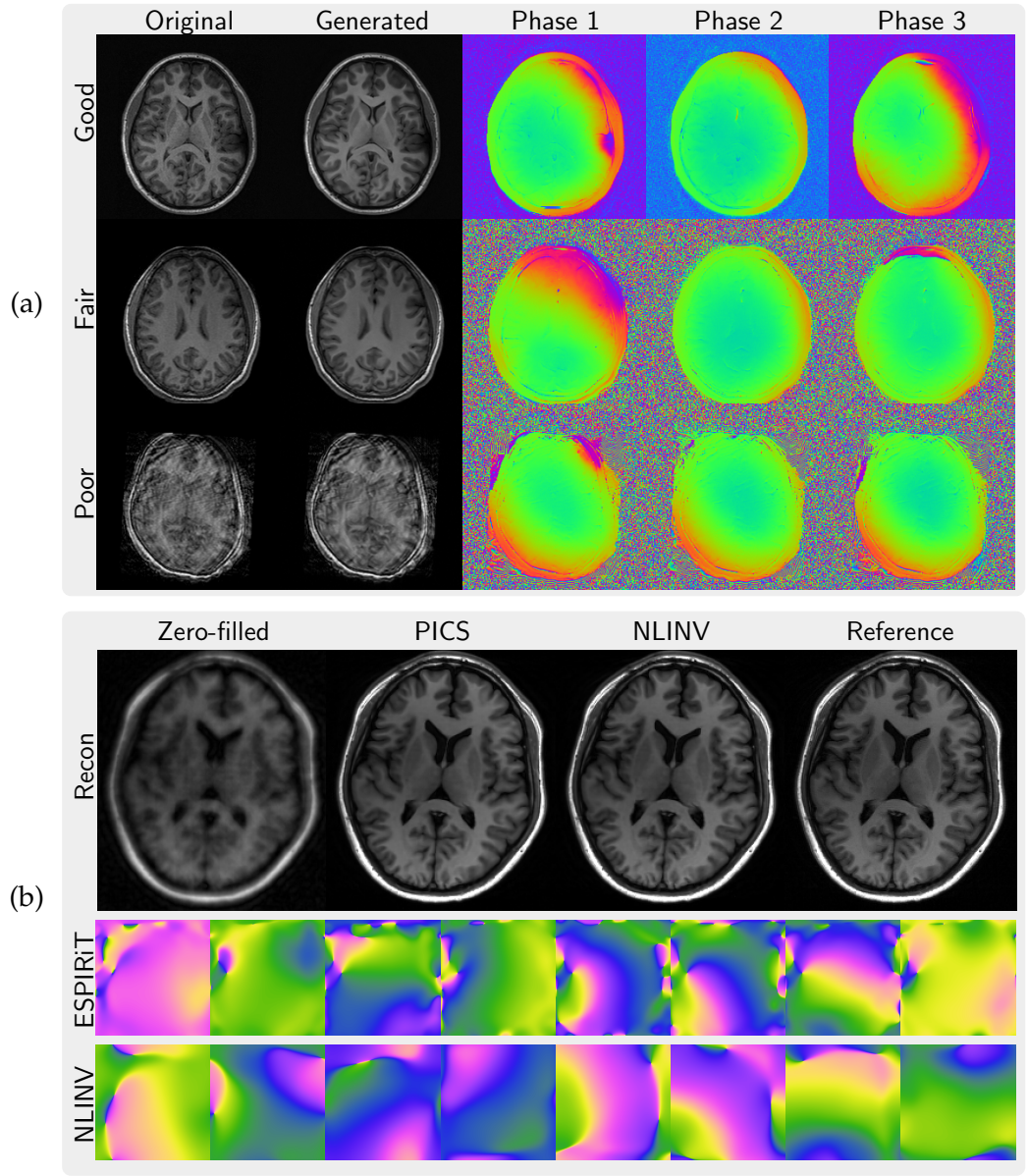


Figure 4.2: (a) Human brain images of different quality are shown (rows). On the left, the original magnitude-only images are compared to the magnitude of a corresponding image generated using phase augmentation. On the right, the phase maps of three different generated images are shown. (b) Linear (PICS) and nonlinear (NLINV) reconstruction using a prior for complex-valued images trained from a dataset of images with preserved phase P_{SC} in comparison to zero-filled image and a fully-sampled reference. The magnitude and coil sensitivity maps estimated with ESPIRiT and NLINV are shown below.

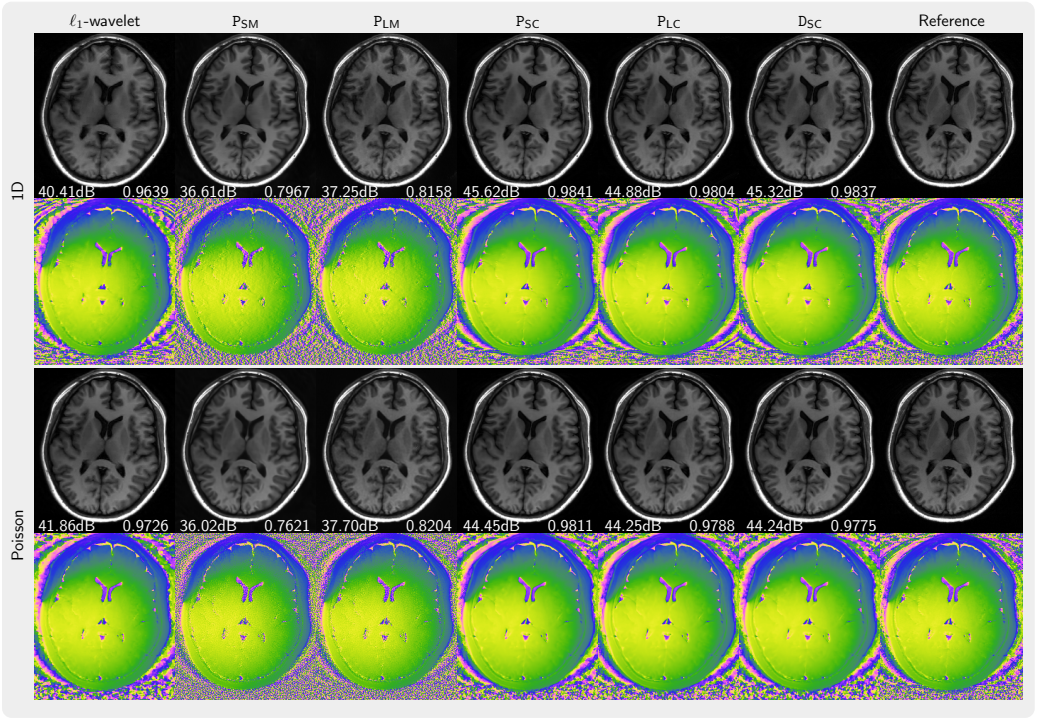


Figure 4.3: Comparison of images reconstructed using PICS using the priors P_{SM}, P_{LM}, P_{SC}, P_{LC}, D_{SC} in comparison to an ℓ_1-wavelet reconstruction and a reference. The top two rows (1D) present the results for 5-fold acceleration along phase-encoding direction with 30 calibration lines. The bottom two rows (Poisson) show the results using a Poisson-disc acquisition of 8.2x-undersampling. PSNR and SSIM values are shown in white text.

served phase, and complex images with generated using our phase augmentation method While the priors P_{SM} and P_{LM} trained from magnitude images can remove folding artifacts introduced by undersampling, they exhibit over-smoothing of the magnitude as indicated by its lower PSNR and SSIM values and also demonstrates poor capabilities in denoising the phase. In contrast, the prior P_{SC} trained on complex-valued images performs much better. Furthermore, the priors P_{LC} and D_{SC} trained on phase-augmented images and perform almost as well. Very similar results were obtained for NLINV as shown in Figure 4.4. In Figure 4.5, the k-space is sampled using 2×3 pattern. We observed artifacts (red arrow) introduced by the priors trained from magnitude-only images reconstructed with PICS method, but not with NLINV method. Under all investigated conditions, the priors trained on complex-valued images outperform the reconstruction with ℓ_1-wavelet regularization.

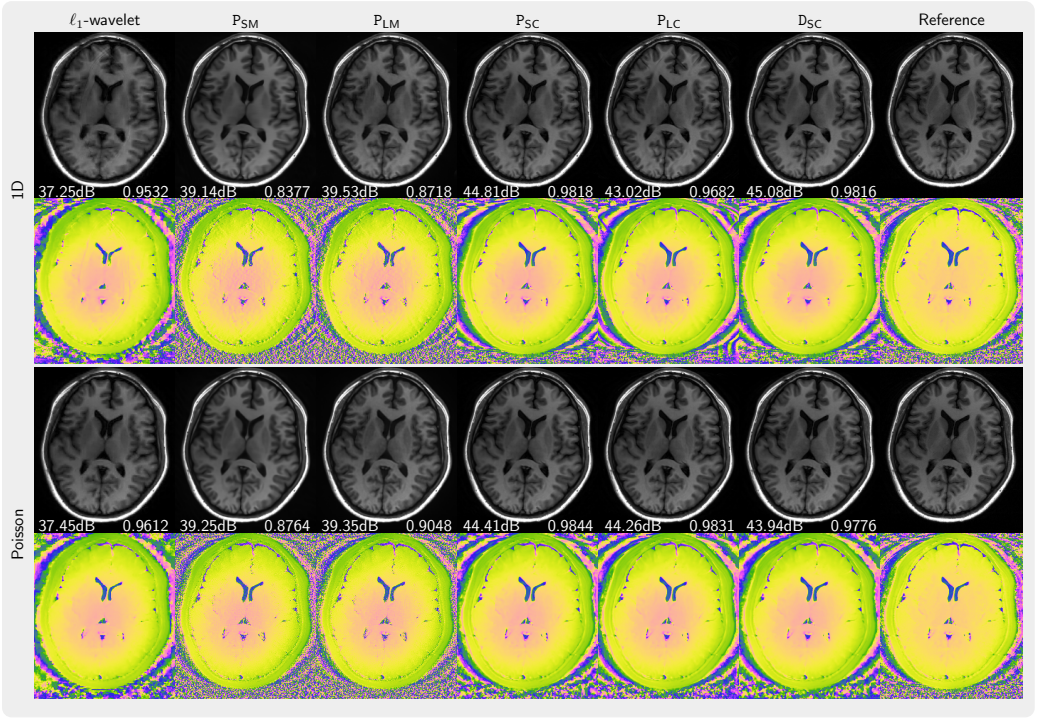


Figure 4.4: Comparison of images reconstructed using NLINV using the priors P_{SM}, P_{LM}, P_{SC}, P_{LC}, D_{SC} in comparison to an ℓ_1-wavelet reconstruction and a reference. The top two rows (1D) present the results for 5-fold acceleration along phase-encoding direction with 30 calibration lines. The bottom two rows (Poisson) show the results using a Poisson-disc acquisition of 8.2x-undersampling. PSNR and SSIM values are shown in white text.

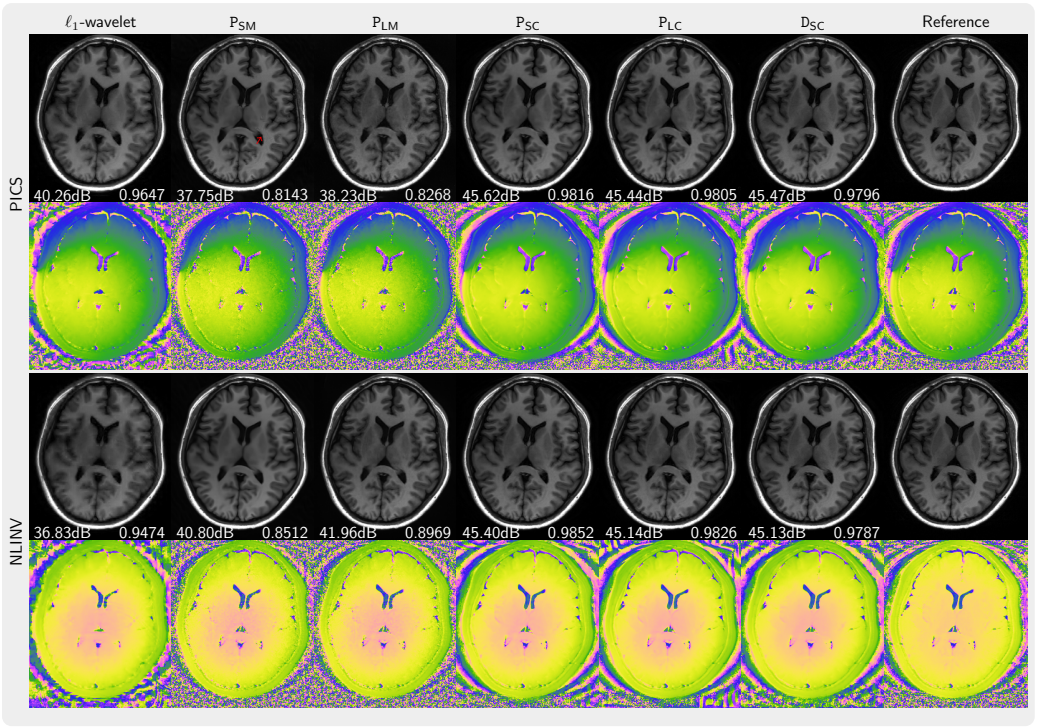


Figure 4.5: Comparison of images reconstructed using NLINV and PICS using the priors P_{SM}, P_{LM}, P_{SC}, P_{LC}, D_{SC} for a 2×3 sampling pattern in comparison to an ℓ_1-wavelet reconstruction and a reference. PSNR and SSIM values are shown in white text. Artifacts (red arrow) are introduced by the priors trained on magnitude images reconstructed with PICS.

4.4.2 *The influence of the size of dataset*

Figure 4.6 presents the images regularized by the priors (P_{SC} and P_{LC}) trained on small and large datasets, respectively. When using PICS with the prior P_{SC} artifacts become apparent in the background and in the brain, whereas no such artifacts are observed when applying the prior P_{LC}. Furthermore, image details appear to be better preserved with high undersampling for the prior P_{LC}.

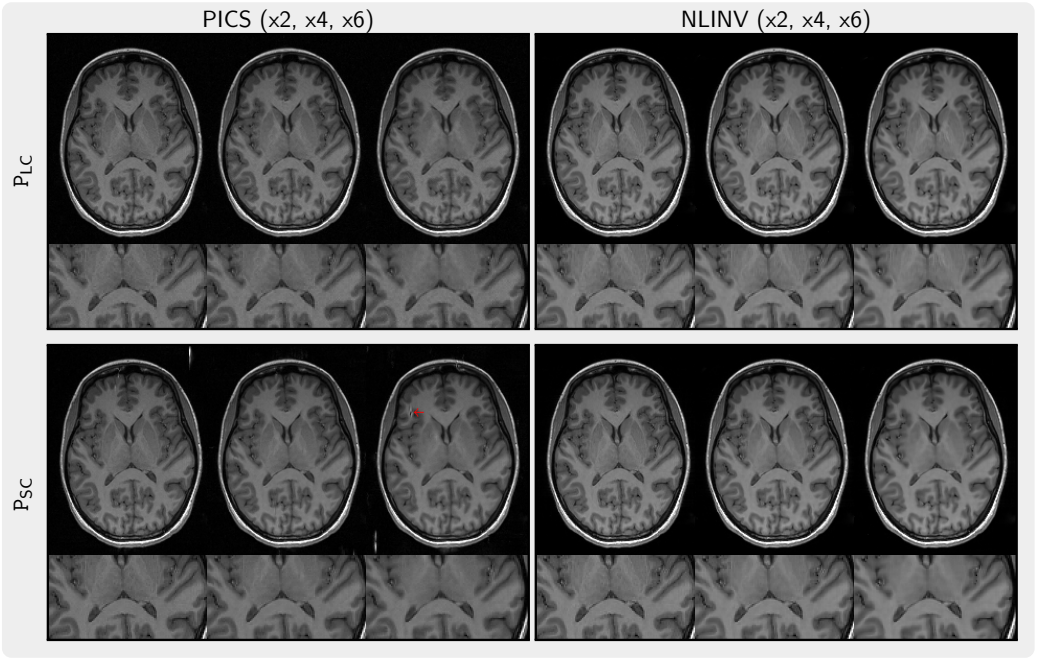


Figure 4.6: Comparison of images reconstructed with PICS (left) and NLINV (right) using priors P_{SC} (top) and P_{LC} (bottom) trained on small and large datasets. We observed artifacts (red arrow) when using PICS with the prior P_{SC} trained on the small dataset. The images in each column are reconstructed from k-space undersampled with factors ranging from 2 to 6 (left to right).

These observations can be confirmed quantitatively. We display the three sets of PSNR and SSIM metrics for ℓ_2, P_{SC}, and P_{DC} with boxplots in Figure 4.7 for 4x and 6x undersampling relative to a reconstruction from 2x undersampled k-space and using P_{LC}. Here, the ℓ_2-regularization serves as a baseline reconstruction which is not influenced by the properties of a learned prior. For the P_{LC} prior learned from a large dataset there are only a few outliers above the average when using PICS and NLINV. However, when using the P_{SC} prior learned from a small dataset there are many outliers below the average, especially when the undersampling factor is high in the case of PICS.

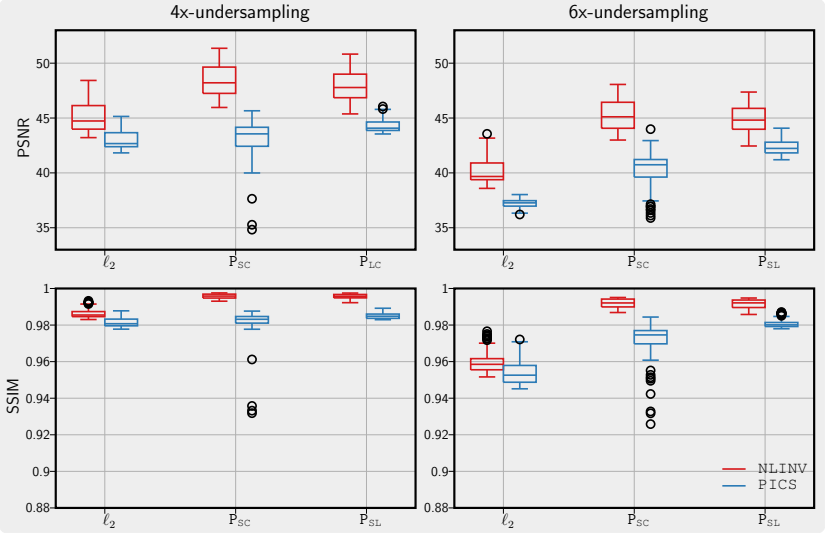


Figure 4.7: PSNR (top) and SSIM (bottom) metrics for images reconstructed with PICS (blue) and NLINV (red) when using ℓ_2-regularization and when using the priors P_{SC} and P_{LC} for 4x (left) and 6x (right) undersampling relative to a reconstruction from 2x undersampled k-space and using P_{LC}.

4.4.3 *3D reconstruction using diffusion priors*

As an example, Figure 4.8 presents three slices in the sagittal, axial, and coronal planes for a 3D volume reconstructed using the diffusion prior D_{SC} using PICS and NLINV in comparison to ℓ_1-wavelet regularization and a reconstruction by coil combination of Fourier-transformed fully-sampled k-space data. By visual inspection, the ℓ_1-regularized images appear to have reduced sharpness compared to the images regularized by the diffusion prior D_{SC} while also having more noise.

Figure 4.9 shows the results from the evaluation by clinical readers. The diffusion prior D_{SC} outperforms ℓ_1-wavelet regularization leveraging the learned knowledge. Here, D_{SC} demonstrates better performance when using PICS method compared to NLINV method. With the relatively high acceleration factor of 8.2 used, none of the reconstructions matches the quality of the reference. The images reconstructed using D_{PC} were very close to those using D_{SC} and are not shown.

4.5 DISCUSSION

A practical workflow was presented for extracting prior information from a set of magnitude-only images. It starts with the preparation of the training dataset, then followed by the generative modelling of complex-valued images and ends with the application of generative priors for regularization in image reconstruction.

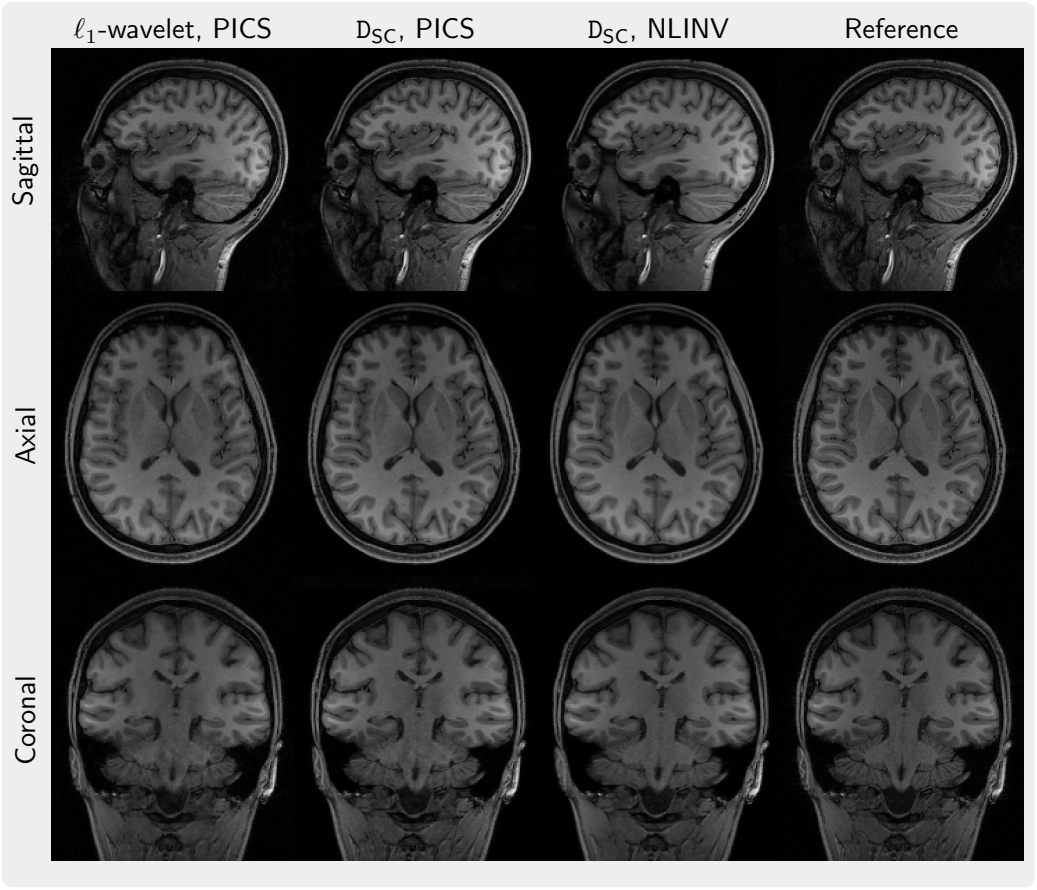


Figure 4.8: Slices in three orientations (Sagittal, Axial, Coronal, from top to bottom) from a 3D volume reconstructed using PICS from 8.2x-undersampled k-space with Poisson-disc sampling with ℓ_1-wavelet regularization and the diffusion prior D_{SC} and using NLINV with diffusion prior D_{SC} (from left to right).

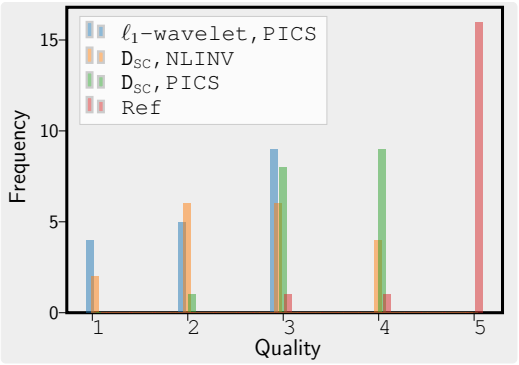


Figure 4.9: Blind evaluation by three clinicians of the volumes that are reconstructed with PICS using ℓ_1-wavelet regularization, PICS with diffusion prior D_{SC}, NLINV with the diffusion prior, and using coil combination of Fourier-transformed fully-sampled k-space data. The grading scale ranges from 5 (excellent) to 1 (bad).

The effectiveness of the prior in boosting image quality was assessed by clinicians. Different aspects of this work are discussed below.
To exploit the information in magnitude-only images, the prediction of phase maps using an U-net was reported in Ref. 73. Different to that, we prepared a training dataset of phase-augmented images through conditional generation using a complex diffusion prior that is first trained on a small dataset of complex-valued images. The prior trained on a large dataset of phase-augmented images exhibits high robustness, as shown in Figure 4.7. The proposed approach will allow us to leverage the information in the large number of Digital Imaging and Communications in Medicine (DICOM) images already available in the archives of radiology departments.

In Ref. 87, the authors applied a conditional generative adversarial network (GAN) to produce phase maps based on magnitude images and then used them to synthesize k-space data. They reported the comparable performance to raw k-space data when the synthetic k-space data were utilized to train a variational network[4] for reconstruction. However, this still required running ESPIRiT on prior ground truth data from fastMRI[48] to obtain sensitivity maps for simulations. Ref. 88 expanded this idea by generating coil sensitivity phase maps based on magnitude images. This was achieved through a three-stage approach, involving the generation of low-resolution coil sensitivity phase maps based on magnitude images with a GAN, upsampling of low-resolution maps to high-resolution ones, and transformation of the coil images to k-space data. Our work proposes a simple and less computationally intensive approach based on phase augmentation using a generic diffusion prior trained on complex-valued images. The advantage of this framework is that the learned prior is independent of k-space sampling patterns and coil sensitivities, and that it can be used as a regularization term in conventional reconstruction algorithms.
Training a prior is computationally expensive. For example, it took around 18 minutes to train P_{LC} per epoch through data parallelism using three A100 80G GPUs (Nvidia Corporation, Santa Clara, CA, USA). In contrast, use of the prior in conventional reconstruction algorithms is computationally efficient. While previous reports indicate that up to 2000 evaluations of a diffusion prior are needed for reconstructing a single image[45], the number of evaluations required for a conventional linear reconstruction algorithm as used in this work is only about 100.

4.6 CONCLUSION

This work focuses on how to extract prior knowledge from existing magnitude-only image datasets using phase augmentation with generative models. The extracted prior knowledge is then applied as regularization in image reconstruction.

The effectiveness of this approach in improving image quality is systematically evaluated across different settings. Our findings stress the importance of incorporating phase information and leveraging large datasets to raise performance and reliability of generative priors for MRI reconstruction.

4.7 DATA AVAILABILITY

In the spirit of reproducible research, the code used to preprocess the dataset and to generate complex-valued images by phase augmentation using the diffusion prior, and the shell scripts for the reconstruction are made available in our repository[1]. The Python library `spreco` used to train priors is available in this repository[2]. Pre-trained models are made available at Zenodo[3]. We refer readers to the webpage of our repository for additional information on the released materials.

4.8 ACKNOWLEDGEMENT

We acknowledge funding by the "Niedersächsisches Vorab" funding line of the Volkswagen Foundation.

1 https://github.com/mrirecon/image-priors
2 https://github.com/mrirecon/spreco
3 https://doi.org/10.5281/zenodo.8083750

5

A PYTHON TOOL FOR TRAINING GENERATIVE PRIORS FOR MRI RECONSTRUCTION

The following is an introduction to the Python package spreco, which is implemented for training generative image priors.

G. Luo, P. Cao, M. Uecker. "Speed up MR scanner with generative priors for image reconstruction (SPRECO)", GitHub repository, `https://github.com/mrirecon/spreco`.

GL and *PC* conceptualized and implemented this package. *GL* further expanded it during his Ph.D. study. *MU* provided supervision and guidance.

5.1 INTRODUCTION

Machine learning has indeed become a powerful tool to tackle real-world problems arising in many fields. However, the transition of machine learning models from the experimental stage to deployment stage can present a number of concerns and issues[89]. In our work, we focus on the domain of MRI reconstruction and aim to demonstrate our practices throughout the entire development life cycle, address practical considerations at each stage. The implementation of spreco in Figure 5.1 has several features that enhance its functionality and usability:

1) **Distributed training:** spreco supports distributed training, leveraging multiple GPUs to accelerate the model training process. This allows for efficient utilization of computational resources and significant reduction of training time for large-scale machine learning models;

2) **Interuptible training:** spreco is designed to handle interruptible training, ensuring that the training process can be paused and resumed seamlessly. This feature is crucial for handling interruptions such as user-initiated pauses without losing progress in the training;

3) **Efficient dataloader:** spreco incorporates an efficient dataloader specifically optimized for handling medical images. This dataloader ensures fast and effective data processing, enabling smooth integration of medical image datasets into the training pipeline;

4) **Customizable models:** spreco provides flexibility in model customization through the use of a configuration file. Users can easily customize model parameters, architecture, and hyperparameters using the configuration file, making it straightforward to experiment with different model setups;

5) **Painless deployment:** the models trained and tested with spreco can be seamlessly integrated into BART, an MRI reconstruction toolbox, using the TensorFlow C API. This enables the execution of models on various platforms and systems without the need for the full TensorFlow (TF) Python environment.

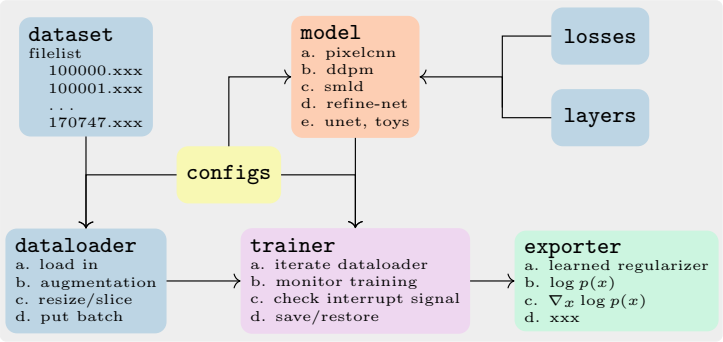


Figure 5.1: The loss and layer functions are used to create models. The trainer is fed by the dataloader and train the model on multiple GPUs. The exporter is used to constomize trained models for deployment.

5.2 METHODS

5.2.1 *Dataloader*

In practices, ensuring that the data pipeline operates asynchronously with the training process is crucial for efficient and seamless data feeding. A well-designed dataflow, such as the one presented in Ref. [90], allows for the production of adaptable data that can seamlessly integrate into the training process implemented in spreco. This eliminates the need for a fixed data format and provides flexibility in handling various data sources.

The provided code example in Listing 5.1 demonstrates the usage of the dataloader in spreco. It defines a function to load files specified in a file list. The data loader then handles the loading process, enabling the retrieval of normalized images with shape (batch_size, x, y, 2). The data is preprocessed and converted from complex to float format, ensuring consistency and compatibility with the training process. Additionally, the dataloader supports multi-threading to further enhance the efficiency of data loading. By using such a data pipeline, spreco can seamlessly and efficiently process data for training, making it a powerful tool for MRI reconstruction and other machine learning applications.

```
from spreco.common.parallel_map import dataloader
from spreco.common import utils
import numpy as np

## define the func to load the file in the list
train_files = utils.read_filelist(path_to_filelist)

def load_file(x):
    """
    x     ---> file path
    imgs  ---> normalized images with shape (b, x, y, 2)
    """
    path, ext = os.path.splitext(x)
    imgs = np.squeeze(utils.readcfl(path))
    imgs = imgs / np.max(np.abs(imgs), axis=(1,2), keepdims=True)
    imgs = utils.cplx2float(imgs)
    return imgs

d = dataloader(filelist, num_thread, load_file, batchsize, buffer_factor=5,
    shuffle=True)
```

Listing 5.1: An example for the usage of the dataloader in spreco.

5.2.2 *Model*

Each model implemented in spreco comprises three crucial components: configuration, loss, neural network. Upon initialization, the model constructs the computation graph for the neural network and sets up the loss function (c.f. Listing 5.2). Subsequently, a train operator is created, which is responsible for updating the shared trainable parameters of the network using the gradients of the loss function. The computation of gradients is distributed across the available GPUs to take advantage of parallel processing, enhancing the efficiency and speed of the training process. The flexibility of the configuration and neural network components allows the model to be adaptable to various tasks and datasets.

```
class model():
  def __init__(self, config):
    # init configs
    self.config = config
    self.net    = get_net(configs)

  def loss(self, x):
    # x ---> input of the model, x should include labels if needed
    ... implementation of loss_func ...
    return loss_func(x)

  def init(self, x, mode):
    # mode: training, inference, export
    if mode == training:

      loss      = []
      grads     = []
      optimizer = AdamOptimizer(self.learning_rate, ...)

      # distribute the computation over \ac{GPU}, the network was
          templatized, therefore all the trainable parameters are shared
          across \ac{GPU}s
      for i in range(self.config['nr_\ac{GPU}']):
        with tf.device('/\ac{GPU}:%d'%i):
          loss.append(self.loss(self.x[i]))
          grads.append(optimizer.compute_gradients(loss[-1], all_params))

      grads_avg = optimizer.average_gradients(grads)
      self.train_op = optimizer.apply_gradients(grads_avg)
```

Listing 5.2: The outline of a model in spreco.

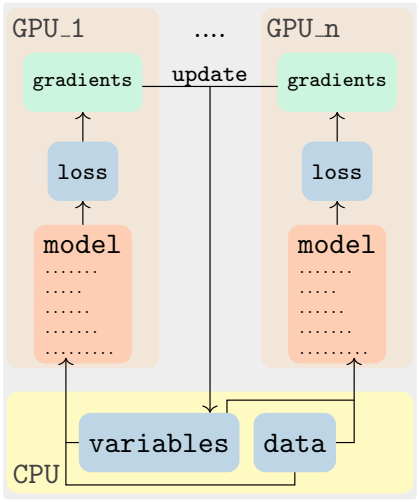


Figure 5.2: Data parallelism training. The trainable variables are shared across GPUs and dataloader feed data for every GPU worker.

5.2.3 *Trainer*

The "trainer" class acts as the heart of neural network training process. It facilitates the seamless integration of the model with the data loading pipeline and provides the necessary functionalities for efficient training and evaluation. The trainer class has two data loaders, "train_loader" and "test_loader". The class allows users to select and initialize the neural network model based on the configuration provided. The "train_loop" function iterates through the training data, running the training operation and loss computation for each batch. The model is updated with the gradients of the loss. In the training loop, the model is also evaluated on the test data using the loss computation. The trainer class ensures that training progress is monitored through logging, with TensorBoard used for visualizing key metrics. Additionally, the class supports checkpoint saving, allowing users to save the model's state at specific intervals and resume training from a saved checkpoint if necessary.

```
class trainer():
  def __init__(self, train_loader, test_loader, config):
    self.train_loader = train_loader
    self.test_loader = test_loader
    self.config = config
    utils.save_config(config['log_path'])

  def init_model(self, config):
    # select and initialize the model specified in config
    self.model = select(config)
    self.model.init()
```

```
12
13  def run_op(self, train_op, loss_op, feed_dict, is_training):
14    if is_training:
15      # run train_op and loss_op
16    else:
17      # run loss_op
18
19  def train_loop(self):
20    # restore the model if there is one
21    # loop over the dataloader
22    # check interrupt signal
23    # log training information with tensorboard
24    if self.config['restore']:
25      saver.restore(sess, self.config['model_folder'])
26
27    for epoch in self.config['epochs']:
28      for train_batch in self.train_loader:
29        self.run_op(self.model.train_op, self.model.loss_op, train_batch,
              is_training=True)
30
31      for test_batch in self.test_loader:
32        self.run_op(None, self.model.loss_op, test_batch, is_training=False)
33
34      if check_interrupt:
35        if self.epoch % self.config["save_interval"] == 0:
36          saver.save(self.config['log_path'])
37
38      self.epoch = self.epoch + 1
39      if self.epoch % self.config["save_interval"] == 0:
40        saver.save(self.config['log_path'])
41
42      self.log(self.config['log_path'])
43
44  def train(self):
45    self.init_model()
46    self.train()
```

Listing 5.3: The outline of the trainer.

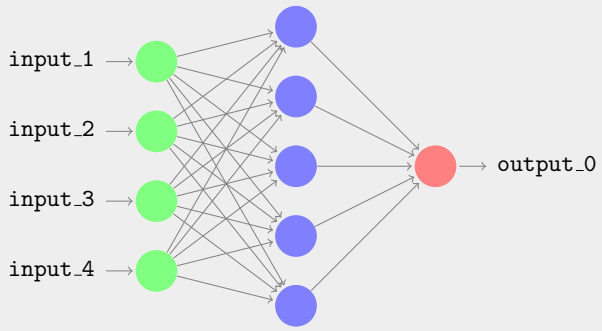
5.2.4 *Deployment*

TensorFlow provides a C API that can be used to build bindings for other languages. The API is defined in c_api.h and designed for simplicity and uniformity. With the implemented TF wrapper in BART using C API, the integration of a

trained prior into the BART reconstruction's implemented with the following these steps:

- Export the trained model with customized labels for inputs $\mathbf{x}$ and outputs $\mathbf{y} = \texttt{Net}_\theta(\mathbf{x})$;
- Initialize an exported graph, the restoration of a saved model using C API;
- Wrap the exported computation graph into BART's non-linear operator (nlop)[67]. The forward pass is called via nlop's forward function;
- Call the attached gradient $\partial\mathbf{y}/\partial\mathbf{x}$ via nlop's adjoint function if necessary.

(a) Export the trained model as computation graph



(b) Use the graph as regularization in BART

```
$ bart pics -R TF:<graph_path>:λ <kspace> <coils> <reco>
```

Figure 5.3: TF computation graph as regularization with BART

Section 4.2.2 details how a generative prior is used as regularization for image reconstruction. The following example demonstrates how to utilize the "exporter" to create such a regularizer with the pre-trained prior using TF computation graph that is compatible with BART. Upon the export of graph, specify the graph with "`-R TF`" option.

```
from spreco.exporter import exporter
import tensorflow.compat.v1 as tf

e = exporter(log,  # the folder where the trained models are
             meta, # a specific meta file
             path, # the path to save the exported graph
             name, # the save the exported graph with name
             default_out=False)

x = tf.placeholder(tf.float32, shape=[1, 256, 256, 2], name="input_0")
```

```
11
12 logits = e.model.eval(x)
13 loss   = e.model.loss_func(x, logits) / np.log(2.0) / np.prod(e.model.config
       ['input_shape'])
```

Listing 5.4: An example for the usage of exporter in spreco.

5.3 CONCLUSION

As a result, it is demonstrated that spreco has versatility in the development stage, making it a valuable tool for researchers and practitioners working with medical imaging data and machine learning models. The deployment of models trained with TF into BART requires minimal environmental prerequisites, making it a practical and user-friendly solution for medical image reconstruction tasks.

6

SUMMARY

In this thesis, three aspects of the application of generative models to CS-MRI parallel imaging reconstruction are presented in separate chapters.

In Chapter 3, the presented approach holds a pivotal advantage over traditional deterministic regularization methods. Bayesian methods considers both the statistical character of the observed k-space and any prior information about the image, which is learned from an existing dataset using diffusion generative models. MCMC seeks to generate samples proportional to the posterior in order to arrive at an optimal estimate of our expectation value. Consequently, it has capacity to provide a means for quantifying the uncertainties associated with the reconstruction, as evident from the variance map. What's more, the results reveals that high undersampling implies a high uncertainty for the reconstructed image and the prior has ability to dominate the image content in an extreme undersampling case. Overall, the utilization of a DL-based approach demonstrates its capability to generate visually realistic images even in cases of substantial underdetermination due to undersampling. However, it's crucial to be aware of the presence of uncertainties within the reconstructed images.

In Chapter 4, it is demonstrated that generative image priors can be constructed with magnitude-only images and then used as regularization for reconstruction. The priors trained on different datasets are evaluated in different k-space sampling settings. Notably, the large dataset with phase information is crucial to produce robust priors capable of handling variations in MR k-space acquisition. The images reconstructed are eventually evaluated by clinicians. With a relatively high acceleration factor of 8.2, none of reconstruction matches the quality of the reference, but they outperformed ℓ_1-wavelet regularization. What's more, the phase augmentation technique allows us to leverage the information in the large number of DICOM images readily accessible in radiology departments' archives.

In Chapter 5, the efficient tool "spreco" designed for training generative priors is introduced, which provides basic functionalities for the whole development stage. Moreover, the model trained with it can be seamlessly deployed into the versatile toolbox BART, serving as a regularization term within iterative algorithms. Notably, this tool has been tested on a local GPU workstation and high performance computing cluster.

When looking at MR imaging from a broader perspective, it becomes apparent that the scanning process is to acquire information for diagnostic purposes. This means that with an increased availability of prior information, the necessity to

acquire new information decreases, which leads to the acceleration of scanning. The learned generative prior was preliminarily demonstrated to be more superior to ℓ_1-prior in terms of the performance evaluated in specific settings. However, the performance bound of a prior and how it relates to the uncertainties of the expected image, which can indicate the least information required in the k-space, is worth further investigation. Furthermore, MR imaging is a sophisticated process, providing many options to capture information for clinical and research purposes. These options consist of contrast weighting, pulse sequences, diffusion weighting techniques and quantitative mapping. As a result, the spectrum of data available through scanning is considerably intricate compared to the generative prior learned from a dataset in this thesis. Therefore, the investigation of how effectively the prior information transfers to a wider array of scenarios stands as a critical direction for future work.

In conclusion, the integration of machine learning techniques, especially generative models, into MRI reconstruction has potential to lead to transformative advancements in medical imaging. These approaches show great promise in improving image quality, reducing scan times, and expanding the capabilities of MRI in clinical diagnosis and research. As research in this domain continues, the synergy between machine learning and MRI reconstruction is expected to drive further innovation and benefit patients and healthcare practitioners alike.

BIBLIOGRAPHY

[1] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[2] K. T. Block, M. Uecker, and J. Frahm, "Undersampled radial MRI with multiple coils. Iterative image reconstruction using a total variation constraint," *Magn. Reson. Med.*, vol. 57, no. 6, p. 1086–1098, 2007.

[3] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for Compressive Sensing MRI," in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016.

[4] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, "Learning a variational network for reconstruction of accelerated MRI data," *Magn. Reson. Med.*, vol. 79, no. 6, pp. 3055–3071, 2017.

[5] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30088–30095, 2020.

[6] S. Bhadra, V. A. Kelkar, F. J. Brooks, and M. A. Anastasio, "On hallucinations in tomographic image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 40, no. 11, pp. 3249–3260, 2021.

[7] D. K. Sodickson and W. J. Manning, "Simultaneous acquisition of spatial harmonics (SMASH): fast imaging with radiofrequency coil arrays," *Magn. Reson. Med.*, vol. 38, no. 4, pp. 591–603, 1997.

[8] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: sensitivity encoding for fast MRI," *Magn. Reson. Med.*, vol. 42, no. 5, pp. 952–962, 1999.

[9] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase, "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magn. Reson. Med.*, vol. 47, no. 6, pp. 1202–1210, 2002.

[10] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[11] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Lett.*, vol. 14, no. 10, pp. 707–710, 2007.

[12] M. Uecker, P. Lai, M. J. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S. Vasanawala, and M. Lustig, "ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA," *Magn. Reson. Med.*, vol. 71, no. 3, pp. 990–1001, 2014.

[13] M. Uecker, T. Hohage, K. T. Block, and J. Frahm, "Image reconstruction by regularized nonlinear inversion-joint estimation of coil sensitivities and image content," *Magn. Reson. Med.*, vol. 60, no. 3, pp. 674–682, 2008.

[14] M. Uecker, K. T. Block, and J. Frahm, "Nonlinear Inversion with L1-Wavelet Regularization - Application to Autocalibrated Parallel Imaging," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 16, (Toronto), p. 1479, 2008.

[15] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-Based Deep Learning Architecture for Inverse Problems," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 394–405, 2019.

[16] F. Knoll, K. Hammernik, E. Kobler, T. Pock, M. P. Recht, and D. K. Sodickson, "Assessment of the generalization of learned image reconstruction and the potential for transfer learning," *Magnetic Resonance in Medicine*, vol. 81, no. 1, pp. 116–128, 2019.

[17] Y. Guan, Y. Li, R. Liu, Z. Meng, Y. Li, L. Ying, Y. P. Du, and Z.-P. Liang, "Subspace model-assisted deep learning for improved image reconstruction," *IEEE Transactions on Medical Imaging*, pp. 1–1, 2023.

[18] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," *arXiv*, 2015.

[19] K. C. Tezcan, C. F. Baumgartner, R. Luechinger, K. P. Pruessmann, and E. Konukoglu, "MR image reconstruction using deep density priors," *IEEE transactions on medical imaging*, vol. 38, no. 7, pp. 1633–1642, 2019.

[20] G. Luo, N. Zhao, W. Jiang, E. S. Hui, and P. Cao, "MRI reconstruction using deep bayesian estimation," *Magn. Reson. Med.*, vol. 84, pp. 2246–2261, apr 2020.

[21] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1747–1756, PMLR, 20–22 Jun 2016.

[22] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[23] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," in *5th International Conference on Learning Representations, ICLR Toulon, France*, OpenReview.net, 24-26 April 2017.

[24] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, no. 24, pp. 695–709, 2005.

[25] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada* (H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds.), pp. 11895–11907, 2019.

[26] A. Hyvärinen, "Sparse Code Shrinkage: Denoising of Nongaussian Data by Maximum Likelihood Estimation," *Neural Computation*, vol. 11, pp. 1739–1768, 10 1999.

[27] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 2256–2265, PMLR, 07–09 Jul 2015.

[28] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[29] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[31] K. P. Pruessmann, M. Weiger, P. Boernert, and P. Boesiger, "Advances in sensitivity encoding with arbitrary k-space trajectories," *Magn. Reson. Med.*, vol. 46, no. 4, pp. 638–651, 2001.

[32] S. Ravishankar and Y. Bresler, "Mr image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1028–1041, 2011.

[33] X. Qu, Y. Hou, F. Lam, D. Guo, J. Zhong, and Z. Chen, "Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator," *Medical Image Analysis*, vol. 18, no. 6, pp. 843–856, 2014. Sparse Methods for Signal Reconstruction and Medical Image Analysis.

[34] S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang, "Accelerating magnetic resonance imaging via deep learning," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pp. 514–517, 2016.

[35] Q. Liu, Q. Yang, H. Cheng, S. Wang, M. Zhang, and D. Liang, "Highly undersampled magnetic resonance imaging reconstruction using autoencoding priors," *Magnetic Resonance in Medicine*, vol. 83, no. 1, pp. 322–336, 2020.

[36] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[37] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3563–3593, 2014.

[38] M. Mardani, E. Gong, J. Y. Cheng, S. S. Vasanawala, G. Zaharchuk, L. Xing, and J. M. Pauly, "Deep generative adversarial neural networks for compressive sensing mRI," *IEEE transactions on medical imaging*, vol. 38, no. 1, pp. 167–179, 2018.

[39] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*, pp. 1613–1622, PMLR, 2015.

[40] D. Narnhofer, A. Effland, E. Kobler, K. Hammernik, F. Knoll, and T. Pock, "Bayesian uncertainty estimation of learned variational mri reconstruction," *IEEE Transactions on Medical Imaging*, vol. 41, no. 2, pp. 279–291, 2022.

[41] D. Calvetti and E. Somersalo, "Hypermodels in the bayesian imaging framework," *Inverse Problems*, vol. 24, p. 034013, may 2008.

[42] A. M. Stuart, "Inverse problems: A bayesian perspective," *Acta Numerica*, vol. 19, p. 451–559, 2010.

[43] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust compressed sensing mri with deep generative priors," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin,

P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 14938–14954, Curran Associates, Inc., 2021.

[44] G. Luo, M. Heide, and M. Uecker, "Using data-driven markov chains for mri reconstruction with joint uncertainty estimation," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 30, (London, UK), p. 0298, 2022.

[45] H. Chung and J. C. Ye, "Score-based diffusion models for accelerated mri," *Medical Image Analysis*, vol. 80, p. 102479, 2022.

[46] B. Levac, A. Jalal, and J. I. Tamir, "Accelerated motion correction for mri using score-based generative models," 2022.

[47] V. Pascal, "A connection between score matching and denoising autoencoders," *Neural Computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[48] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdzal, A. Romero, M. Rabbat, P. Vincent, N. Yakubova, J. Pinkerton, D. Wang, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson, and Y. W. Lui, "fastmri: An open dataset and benchmarks for accelerated mri," *arXiv*, 2019.

[49] Z. Ramzi, P. Ciuciu, and J.-L. Starck, "Xpdnet for mri reconstruction: an application to the 2020 fastmri challenge," *arXiv*, 2020.

[50] S. Särkkä and A. Solin, *Applied stochastic differential equations*, vol. 10. Cambridge University Press, 2019.

[51] R. Douc, E. Moulines, P. Priouret, and P. Soulier, *Markov chains*, ch. 2, pp. 38–41. Springer, 2018.

[52] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 1510–1519, IEEE Computer Society, 2017.

[53] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems* (J. Platt, D. Koller, Y. Singer, and S. Roweis, eds.), vol. 20 of *NIPS'07*, (Red Hook, NY, USA), pp. 1177–1184, Curran Associates, Inc., 2008.

[54] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5168–5177, IEEE Computer Society, 2017.

[55] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7794–7803, Computer Vision Foundation / IEEE Computer Society, 2018.

[56] M. Uecker, S. Rosenzweig, H. C. M. Holme, M. Blumenthal, Z. Tan, X. Wang, J. I. Tamir, and M. Lustig, "mrirecon/bart: version 0.6.00," July 2020.

[57] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, (USA), p. 265–283, USENIX Association, 2016.

[58] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[59] M. J. Muckley, B. Riemenschneider, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha, S. Arberet, D. Nickel, Z. Ramzi, P. Ciuciu, J.-L. Starck, J. Teuwen, D. Karkalousos, C. Zhang, A. Sriram, Z. Huang, N. Yakubova, Y. W. Lui, and F. Knoll, "Results of the 2020 fastMRI challenge for machine learning MR image reconstruction," *IEEE Trans. Med. Imag.*, pp. 1–1, 2021.

[60] E. Shimron, J. I. Tamir, K. Wang, and M. Lustig, "Implicit data crimes: Machine learning bias arising from misuse of public data," *Proceedings of the National Academy of Sciences*, vol. 119, no. 13, p. e2117203119, 2022.

[61] M. Arvinte and J. Tamir, "The truth matters: A brief discussion on MVUE vs. RSS in mri reconstruction," in *European Society for Magnetic Resonance in Medicine and Biology*, vol. 38, (Virtual Conference), Oct 2021.

[62] D. Narnhofer, A. Habring, M. Holler, and T. Pock, "Posterior-variance-based error quantification for inverse problems in imaging," 2022.

[63] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984.

[64] S. Roth and M. J. Black, "Fields of experts: a framework for learning image priors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 860–867, 2005.

[65] U. Schmidt, Q. Gao, and S. Roth, "A generative perspective on MRFs in low-level vision," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1751–1758, 2010.

[66] G. Luo, M. Blumenthal, and M. Uecker, "Using data-driven image priors for image reconstruction with bart," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 29, (Virtual Conference), p. 3768, 2021.

[67] M. Blumenthal, G. Luo, M. Schilling, H. C. M. Holme, and M. Uecker, "Deep, deep learning with bart," *Magn. Reson. Med.*, vol. 89, no. 2, pp. 678–693, 2023.

[68] S. S. Vasanawala, M. T. Alley, B. A. Hargreaves, R. A. Barth, J. M. Pauly, and M. Lustig, "Improved pediatric mr imaging with compressed sensing," *Radiology*, vol. 256, no. 2, pp. 607–616, 2010.

[69] L. Feng, T. Benkert, K. T. Block, D. K. Sodickson, R. Otazo, and H. Chandarana, "Compressed sensing for body mri," *Journal of Magnetic Resonance Imaging*, vol. 45, no. 4, pp. 966–987, 2017.

[70] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, "Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data," *Magnetic Resonance in Medicine*, vol. 84, no. 6, pp. 3172–3191, 2020.

[71] M. Blumenthal, G. Luo, M. Schilling, M. Haltmeier, and M. Uecker, "NLINV-Net: Self-supervised end-2-end learning for reconstructing undersampled radial cardiac real-time data," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 30, (London, UK), p. 0499, 2022.

[72] G. Luo, M. Blumenthal, M. Heide, and M. Uecker, "Bayesian mri reconstruction with joint uncertainty estimation using diffusion models," *Magn. Reson. Med.*, vol. 90, no. 1, pp. 295–311, 2023.

[73] G. Luo, M. Blumenthal, X. Wang, and M. Uecker, "All you need are dicom images," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 30, (London, UK), p. 1510, 2022.

[74] G. Luo, X. Wang, V. Roeloffs, Z. Tan, and M. Uecker, "Joint estimation of coil sensitivities and image content using a deep image prior," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 29, (Virtual Conference), p. 280, 2021.

[75] L. Ying, D. Xu, and Z.-P. Liang, "On tikhonov regularization for image reconstruction in parallel mri," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, pp. 1056–1059, 2004.

[76] L. Ying and J. Sheng, "Joint image reconstruction and sensitivity estimation in SENSE (JSENSE)," *Magn. Reson. Med.*, vol. 57, no. 6, pp. 1196–1202, 2007.

[77] F. Knoll, C. Clason, K. Bredies, M. Uecker, and R. Stollberger, "Parallel imaging with nonlinear reconstruction using variational penalties," *Magn. Reson. Med.*, vol. 67, no. 1, pp. 34–41, 2012.

[78] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Img. Sci.*, vol. 10, pp. 1804–1844, jan 2017.

[79] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Img. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[80] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*, IEEE, dec 2013.

[81] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. Castellanos, K. Alaerts, J. Anderson, M. Assaf, S. Bookheimer, M. Dapretto, B. Deen, S. Delmonte, I. Dinstein, B. Ertl-Wagner, D. Fair, L. Gallagher, D. Kennedy, C. Keown, C. Keysers, J. Lainhart, C. Lord, B. Luna, V. Menon, N. Minshew, C. Monk, S. Mueller, R. Mueller, M. Nebel, J. Nigg, K. O'Hearn, K. Pelphrey, S. Peltier, J. Rudie, S. Sunaert, M. Thioux, J. Tyszka, L. Uddin, J. Verhoeven, N. Wenderoth, J. Wiggins, S. Mostofsky, and M. Milham, "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," *Molecular Psychiatry*, vol. 19, pp. 659–667, June 2014.

[82] A. Di Martino, D. O'connor, B. Chen, K. Alaerts, J. S. Anderson, M. Assaf, J. H. Balsters, L. Baxter, A. Beggiato, S. Bernaerts, L. M. E. Blanken, S. y. Bookheimer, B. B. Braden, L. Byrge, F. X. Castellanos, M. Dapretto, R. Delorme, D. A. Fair, I. Fishman, J. Fitzgerald, L. Gallagher, R. J. J. Keehn, D. P. Kennedy, J. E. Lainhart, B. Luna, S. H. Mostofsky, R.-A. Müller, M. B. Nebel, J. T. Nigg, K. O'hearn, M. Solomon, R. Toro, C. J. Vaidya, N. Wenderoth, T. White, R. C. Craddock, C. Lord, B. L. Leventhal, and M. Milham, "Enhancing studies of the connectome in autism using the autism brain imaging data exchange II," *Scientific Data*, vol. 4, p. 170010, Mar. 2017.

[83] M. Brett, C. J. Markiewicz, M. Hanke, M.-A. Côté, B. Cipollini, P. McCarthy, D. Jarecka, C. P. Cheng, Y. O. Halchenko, M. Cottaar, E. Larson, S. Ghosh, D. Wassermann, S. Gerhard, G. R. Lee, Z. Baratz, H.-T. Wang, E. Kastman, J. Kaczmarzyk, R. Guidotti, J. Daniel, O. Duek, A. Rokem, C. Madison, D. Papadopoulos Orfanos, A. Sólon, B. Moloney, F. C. Morency, M. Goncalves, R. Markello, C. Riddell, C. Burns, J. Millman, A. Gramfort, J. Leppäkangas, J. J. van den Bosch, R. D. Vincent, H. Braun, K. Subramaniam, A. Van, K. J. Gorgolewski, P. R. Raamana, J. Klug, B. N. Nichols, E. M. Baker, S. Hayashi, B. Pinsard, C. Haselgrove, M. Hymers, O. Esteban, S. Koudoro, F. Pérez-García, J. Dockès, N. N. Oosterhof, B. Amirbekian, H. Christian, I. Nimmo-Smith, L. Nguyen, S. Reddigari, S. St-Jean, E. Panfilov, E. Garyfallidis, G. Varoquaux, J. H. Legarreta, K. S. Hahn, L. Waller, O. P. Hinds, B. Fauber, F. Perez, J. Roberts, J.-B. Poline, J. Stutters, K. Jordan, M. Cieslak, M. E. Moreno, T. Hrnčiar, V. Haenel, Y. Schwartz, B. C. Darwin, B. Thirion, C. Gauthier, I. Solovey, I. Gonzalez, J. Palasubramaniam, J. Lecher, K. Leinweber, K. Raktivan, M. Calábková, P. Fischer, P. Gervais, S. Gadde, T. Ballinger, T. Roos, V. R. Reddam, and freec84, "nipy/nibabel: 5.1.0," Apr. 2023.

[84] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 6840–6851, Curran Associates, Inc., 2020.

[85] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," *arXiv*, 2016.

[86] M. Uecker, F. Ong, J. I. Tamir, D. Bahri, P. Virtue, J. Y. Cheng, T. Zhang, and M. Lustig, "Berkeley advanced reconstruction toolbox," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 23, (Toronto), p. 2486, 2015.

[87] N. Deveshwar, A. Rajagopal, S. Sahin, E. Shimron, and P. E. Z. Larson, "Synthesizing complex-valued multicoil mri data from magnitude-only images," *Bioengineering*, vol. 10, no. 3, 2023.

[88] F. Zijlstra and P. T. While, "Deep-learning-based transformation of magnitude images to synthetic raw data for deep-learning-based image reconstruction," in *Proc. Intl. Soc. Mag. Reson. Med.*, vol. 31, (Toronto, Canada), p. 0825, 2023.

[89] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: A survey of case studies," *ACM Comput. Surv.*, vol. 55, dec 2022.

[90] Y. Wu *et al.*, "Tensorpack." `https://github.com/tensorpack/`, 2016.

ACRONYMS

CS	compressed sensing
DL	deep learning
ML	machine learning
MR	magnetic resonance
RF	radiofrequency
TE	echo time
TF	TensorFlow
TR	repetition time
emf	electromotive force
FID	free induction decay
FFT	fast Fourier transform
GAN	generative adversarial network
GPU	graphics processing unit
MAP	maximum a posterior
MRI	magnetic resonance imaging
SNR	signal noise ratio
VAE	variational autoencoder
pPCA	probabilistic principal component analysis
CNNs	convolutional neural networks
BART	Berkeley advanced reconstruction toolbox
MMSE	minimum mean square error
MCMC	Markov chain Monte Carlo
SENSE	sensitivity encoding
DICOM	Digital Imaging and Communications in Medicine
FISTA	fast iterative shrinkage-thresholding method
NLINV	nonlinear inversion
GRAPPA	generalized autocalibrating partially parallel acquisition

CURRICULUM VITAE

GUANXIONG LUO

14.04.1995, Lunar Calendar, born in Pingjiang, China

SCIENTIFIC EDUCATION:

01/2020–09/2023	PhD Student with Prof. Dr. Martin Uecker (main supervisor) in the program of Computer Science at University of Göttingen, Germany
09/2017–10/2019	Master of Philosophy with Prof. Dr. Peng Cao (main supervisor) at The University of Hong Kong, China completed thesis titled: *"The application of generative networks in MR image reconstruction"* in the Department of Radiology
09/2013–07/2017	B. Eng in Biomedical Engineering Awarded with Outstanding Graduate in the class of 2017 Xi'an Jiaotong University, China Final year project *"Compressed sensing MRI"* with Prof. Dr. Duan Juanbo

PUBLICATIONS:

2023	Guanxiong Luo, Moritz Blumenthal, Martin Heide, Martin Uecker. *Bayesian MRI Reconstruction with Joint Uncertainty Estimation Using Diffusion Priors*, Magn Reson Med. 2023;
2023	Moritz Blumenthal, Guanxiong Luo, Martin Schilling, H. Christian M. Holme, Martin Uecker *Deep, deep learning with BART*, Magn Reson Med. 2023; 89: 678- 693.
2020	Guanxiong Luo, Na Zhao, Wenhao Jiang, Edward S. Hui, Peng Cao. *MRI reconstruction using deep Bayesian estimation*, Magn Reson Med. 2020; 84: 2246-2261.

CONFERENCE AND CONTRIBUTIONS:

2022 Guanxiong Luo, Martin Heide, Martin Uecker. *Using data-driven Markov chains for MRI reconstruction with Joint Uncertainty Estimation*, Power Pitch Session, ISMRM 2022.

2022 Moritz Blumenthal, Guanxiong Luo, Martin Schilling, Markus Haltmeier, Martin Uecker. *NLINV-Net: Self-Supervised End-2-End Learning for Reconstructing Undersampled Radial Cardiac Real-Time Data*, Oral Scientific Session, ISMRM 2022.

2022 Guanxiong Luo, Moritz Blumenthal, Xiaoqing Wang, Martin Uecker. *All you need are DICOM images*, Poster Session, ISMRM 2022.

2021 Guanxiong Luo, Xiaoqing Wang, Volkert Roeloffs, Zhengguo Tan, Martin Uecker. *Joint estimation of coil sensitivities and image content using a deep image prior*, Oral Scientific Session, ISMRM 2021.

2021 Guanxiong, Moritz Blumenthal, Martin Uecker. *Using data-driven image priors for image reconstruction with BART*, Poster Session and Software Session, ISMRM 2021.

2020 Guanxiong Luo, Cao Peng. *MRI Reconstruction Using Deep Bayesian Inference*, Oral Scientific Session, ISMRM 2020.

COURSES AND WORKSHOPS:

2021 SS	Training in Good Scientific Practice
2021 SS	Mental Strategies for Doctoral Students
2021 WS	Introduction to Public Relations for Young Scientists
2021 WS	Machine Learning for Inverse Problems
2023 SS	Voice and Body Coaching: Communicating with Confidence and Accuracy

TEACHING EXPERIENCE:

2021 WS	Teaching assistant for a course on the application of data science to smart city.
2021 WS	Tutorials for undergraduates and graduates, teaching assistant for a course on deep learning.

LANGUAGES:

Mandarin/Xiang Chinese (native)
English (working language)
Cantonese (daily use)
German (A2 level)

This thesis was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić.