



# Investigating Implicit Neural Functions for MRI reconstruction

Semester project

**Marina Crespo Aguirre**

Department of Information Technologies and Electrical Engineering

**Advisors:** PhD student - Georg Brunner  
**Supervisor:** Prof. Dr. Ender Konukoglu

March 15, 2025



# Abstract

This study investigates the use of Neural Implicit Fields (NIFs) for MRI reconstruction, with the goal of overcoming the limitations of conventional methods. By leveraging the advantages of NIFs, our work aims to reconstruct high-quality images from sparsely sampled k-space data.

We first examine a single-volume reconstruction scenario, revealing a tendency for the model to overfit to the high-magnitude central k-space values. To mitigate this, we explore model regularization strategy, that although conceptually promising, results in unstable and very computationally expensive training without improving the baselines. Hash encodings guide the model to better learning fully-sampled k-spaces, by capturing finer-grained spatial details. However, in under-sampled grids, their benefits are diminished.

Extending the approach to a multi-volume training setup, we observe a divergence between training and inference performance, even over volumes the model has seen during training, prediction over non-sampled regions is suboptimal.

These findings highlight both the potential, and the current limitations of NIFs for accelerated MRI reconstruction. Future research should focus on improving the generalization over unseen k-space regions, and explore some other more advanced architectures to capture meaningful volume-specific features in multi-volume scenarios.



# Acknowledgements

I wanted to thank Georg for the time, patience and valuable advice during these six months.



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b>  |
| 1.1      | Focus of this Work . . . . .   | 2         |
| 1.2      | Report Organization . . . . .  | 4         |
| <b>2</b> | <b>Related Work</b>  | <b>5</b>  |
| <b>3</b> | <b>Single-Volume</b>   | <b>7</b>  |
| 3.1      | Description of the problem . . . . .                                     | 7         |
| 3.2      | Method . . . . .   | 7         |
| 3.2.1    | Model . . . . .  | 7         |
| 3.2.2    | Optimization . . . . .   | 9         |
| 3.3      | Experiments & Results . . . . .  | 9         |
| 3.3.1    | Experimental Setup . . . . .   | 9         |
| 3.3.2    | Baseline . . . . .   | 10        |
| 3.3.3    | First Experiment: PISCO regularization . . . . .                         | 13        |
| 3.3.4    | Second Experiment: From fixed to learnable Position Embeddings . . . . . | 16        |
| <b>4</b> | <b>Multi-Volume</b>  | <b>21</b> |
| 4.1      | Description of the Problem . . . . .                                     | 21        |
| 4.2      | Method . . . . .   | 21        |
| 4.3      | Optimization . . . . .   | 21        |
| 4.4      | Experiments . . . . .  | 23        |
| 4.4.1    | Number of training samples . . . . .                                     | 23        |
| 4.4.2    | Embeddings Dimension . . . . .   | 24        |
| 4.4.3    | Stochasticity and Regularization . . . . .                               | 26        |
| <b>5</b> | <b>Conclusion</b>  | <b>31</b> |

## CONTENTS

# List of Figures

|      |  |    |
|------|--|----|
| 1.1  | GRAPPA reconstruction [7]. a. Undersampled Kspace data collected from each coil. b. Autocalibration signals collected from fully sampled regions. c. GRAPPA weights are estimated from a set of fully sampled kernels from the ACS region. d. Linear combination of acquired samples weighted with GRAPPA weights yields a reconstructed Kspace. e. Fourier Transform is used to retrieve single-coil images . . . . . | 3  |
| 3.1  | <b>a.</b> K-space modulus fully sampled (left) and fully sampled without central values (right). <b>b.</b> Corresponding inverse Fourier transformations. . . . .  | 8  |
| 3.2  | <b>a.</b> K-space masks with equispaced undersampling (left) and randomly-spaced undersampling (right), acceleration factor $\times 4$ , center fraction 0.15 . <b>b.</b> Resulting inverse Fourier transformations from undersampled k-spaces. . . . .  | 8  |
| 3.3  | Simplified diagram of single volume training pipeline. MLP with 8 hidden layers of dimension 512 and sinusoidal activation functions. . . . .  | 9  |
| 3.4  | Baseline model predictions with center k-space inserted. a. Fully sampled k-space training. b. Under-sampled k-space training. . . . .   | 11 |
| 3.5  | Baseline model predictions without the insertion of central k-space values. a. Fully sampled k-space training. b. Under-sampled k-space training. . . . .  | 11 |
| 3.6  | Baseline model modulus k-space predictions. a. Fully sampled k-space training. b. Under-sampled k-space training. . . . .  | 12 |
| 3.7  | Inverse transform of predicted k-space with inserted center values. . . . .  | 14 |
| 3.9  | Example diagram of the hash encoding of a coordinate $x$ at two resolution levels. . . . .   | 16 |
| 3.10 | Reconstructions from model predictions with hash encodings, with inserted center k-space values. a. Fully sampled k-space training. b. Under-sampled k-space training. . . . .   | 17 |
| 3.11 | Reconstructions from model predictions with hash encodings. a. Fully sampled k-space training. b. Under-sampled k-space training. . . . .  | 18 |
| 3.8  | Model predictions with PISCO regularization. a. Modulus k-space model prediction. b. Inverse fourier transformation of model prediction. . . . .   | 19 |
| 3.12 | k-space modulus predictions of model with hash encodings. a. Fully sampled k-space training. b. Undersampled k-space training. . . . .   | 20 |
| 4.1  | Simplified diagram of multivolume training pipeline. . . . .   | 22 |
| 4.2  | Reconstruction of volume from training distribution by models trained on various n# training samples a. Training phase. b. Inference phase . . . . .   | 24 |
| 4.3  | Reconstruction of volume from training distribution by models trained over various n# training samples, inserting the central k-space values. a. Training phase. b. Inference phase .  | 25 |
| 4.4  | Reconstruction of volume from training distribution, by models trained with different embedding sizes. a. Training phase. b. Inference phase. . . . .  | 27 |

## LIST OF FIGURES

---

|     |   |    |
|-----|---|----|
| 4.5 | Reconstruction of volume from training distribution by models trained with various embedding sizes, with center k-space values inserted. a. Training phase. b. Inference phase. . . . . | 28 |
| 4.6 | Results from third experiment, reconstruction of volume from training distribution. a. Model predictions b. Model predictions + k-space center . . . . .                                | 29 |

# List of Tables

|     |  |    |
|-----|--|----|
| 3.1 | Results from baseline model.   | 10 |
| 3.2 | Comparison of PSNR, SSIM, and NMSE between Center and Undersampled data. | 14 |
| 3.3 | Comparison of PSNR, SSIM, and NMSE across different sampling methods.    | 17 |
| 4.1 | Training and Inference scores from first experiment                      | 24 |
| 4.2 | Training scores from second experiment                                   | 26 |
| 4.3 | Inference results second experiment                                      | 26 |
| 4.4 | Inference results third experiment                                       | 29 |

**LIST OF TABLES**

---

# Chapter 1

## Introduction

### MRI Fundamentals

Neuroimaging encompasses a collection of techniques used to visualize both the structure and function of the brain. Among these, Magnetic Resonance Imaging (MRI) stands out as one of the most powerful and widely used methods due to its high-resolution imaging capabilities in soft tissues, and non-invasive nature. Leveraging Nuclear Magnetic Resonance (NMR), MRI is based on the absorption of radio-frequency (RF) energy by the magnetic moments of atomic nuclei in samples placed inside a strong magnetic field [13]. The acquisition of MRI scans is based on the energy transfer between these protons and the surrounding magnetic field [4]. More precisely, the combination of a series of magnetic coils acting on the patient, induce excitation of a selected soft tissue slice. The sample is exposed to pulses of RF energy at the correct frequency to be absorbed (excitation pulse); then, during the atoms' relaxation, this energy is re-emitted and captured by the gradient coils [6]. The received RF signal is processed to deduce position information, and is encoded in Fourier domain, composing the k-space grid.

The contrast in MRI images arises primarily from variations in two fundamental relaxation times: T1 (longitudinal relaxation time), or the time it takes for the protons to realign with the external magnetic field, and T2 (transverse relaxation time), which characterizes how long the proton spins remain coherent before dephasing due to interactions with neighboring spins. Different tissue types, depending on their composition and molecular environment, exhibit distinct T1 and T2 relaxation times, leading to intrinsic contrast in MRI images[6, 5]. By adjusting the parameters of the RF excitation pulse, different contrast modalities can be obtained. T1-weighted images distinguish with high contrast, structures whose atoms have short T1 relaxation times (e.g. fat tissue), appearing brighter than those that have longer T1 relaxation times (e.g. water). Some clinical applications where they become relevant are anatomical imaging and the detection of fat-containing structures (e.g. fatty liver disease). Contrarily, T2-weighted images enhance structures that present longer T2 relaxation times, or water rich structures (e.g. Cerebrospinal Fluid, edemas, etc) [2].

Although MRI is a highly powerful imaging technology, it requires significantly longer scan times than other modalities like X-ray and Computed Tomography (CT) due to the complex physics involved in data acquisition and image reconstruction [12]. Unlike X-ray or CT, which capture images almost instantaneously, MRI collects data in a stepwise manner. Each scan captures the k-space with the frequency and phase information from the RF signals detected by the receiver coils.

The speed at which the k-space is sampled is intrinsically related to the resulting image, denser k-spaces, take longer scan times, but result in images with no aliasing artifacts. Contrarily, sparser k-spaces resulting from faster sampling, break Nyquist theorem and result in images with more sampling artifacts. The extent of k-space sampled on the other hand is intrinsically related to the resolution of the resulting images. Central magnitudes encode general SNR and contrast information of the image, whereas boundary k-space values encode the image resolution [5, 3].

Longer acquisition times can be a significant drawback, not only regarding patient discomfort; but also leading to motion artifacts, and potential thermal effects on tissues from the extended absorption of the RF energy. It is for all these reasons that advanced techniques like parallel imaging [16] and compressed sensing have been developed, with the goal of accelerating MRI while preserving image quality.

## FastMRI

While it is possible to accelerate MRI data acquisition by collecting fewer encoding lines in the K-space, the resulting aliasing effect must be removed for the images to be useful for clinical purposes [7].

Parallel imaging is an advanced acceleration technique that allows faster scans without compromising image quality. This method leverages the use of multiple receiver coils, each positioned around the patient to detect different portions of the radiofrequency (RF) signal. By strategically undersampling k-space in the phase-encoding direction, MRI acquisition time is significantly reduced. However, since undersampling leads to aliasing artifacts, specialized reconstruction techniques are required to recover the full-resolution image. One such technique is Sensitivity Encoding (SENSE) [16]. It directly processes the aliased image obtained from reconstructing the undersampled k-space. SENSE uses coils sensitivity maps, which feature the variation in signal strength detected by each coil relative to the distance to the imaged sample. Since each coil receives a weighted version of the true image, with the weighting determined by its sensitivity profile, these maps provide the necessary spatial information to disentangle overlapping signals. To reconstruct the full image, SENSE employs linear algebra methods to unfold the aliased data, yielding a high-resolution image despite the reduced data acquisition [16].

Generalized Autocalibrating Partially Parallel Acquisition (GRAPPA) [8], is another technique of reconstruction. It operates directly in Kspace, focusing on recovering missing K-space data by exploiting the redundancy of information across multiple receiver coils [7]. The core principle of GRAPPA relies on the fact that coil sensitivity maps cause a spatially distributed smearing of information across neighboring k-space points [7]. To correct for undersampling, GRAPPA acquires a small fully sampled region of k-space, known as the Autocalibration Signal (ACS) region, from which the GRAPPA weights are estimated. Then, the undersampled k-space lines are estimated with a weighted linear combination of signals from the neighboring sampled points (Figure 1.1).

$$\mathcal{F}_{target} = \sum_i w_i \mathcal{F}_{neighbor,i}$$

Compressed sensing (CS) corresponds to a group of advanced MRI acceleration techniques that seek for high quality image reconstruction from highly under-sampled k-space data. It is based on the idea that images with sparse representations can be recovered from randomly under-sampled k-space data, provided that an appropriate non-linear recovery scheme is used [11].

While conventional techniques of reconstruction (PI and CS) are well established, they come with trade-offs. PI is limited by coil sensitivity maps (SENSE [16]), it requires greater hardware demands [10], and shows a decay in Signal to Noise Ratio (SNR) at large under-sampling ratios (GRAPPA [8]). On the other hand, CS is computationally more expensive, as it requires of iterative algorithms to solve the image reconstruction problem.

## 1.1 Focus of this Work

Traditional MRI reconstruction methods—such as Parallel Imaging (PI) and Compressed Sensing (CS)—face inherent limitations that constrain the benefits of sparse undersampling, particularly by restricting the potential reduction benefits in acquisition time, that are in theory achievable by the sparsity condition.

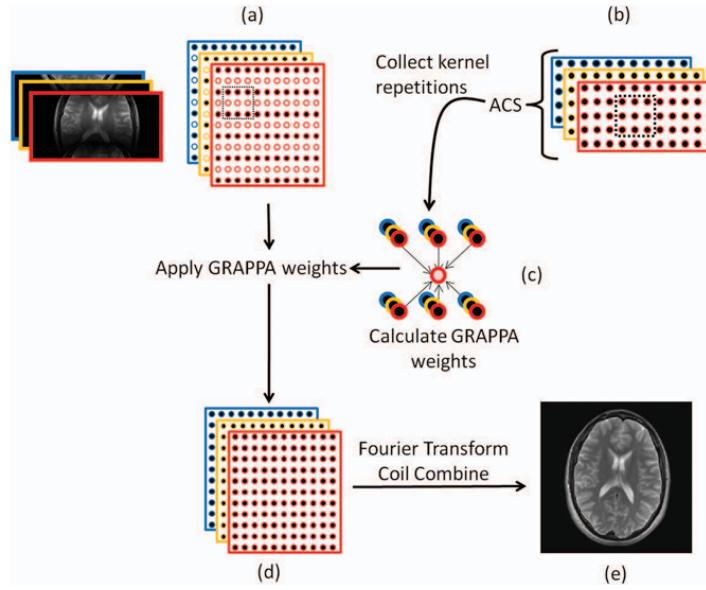


Figure 1.1: GRAPPA reconstruction [7]. a. Undersampled Kspace data collected from each coil. b. Autocalibration signals collected from fully sampled regions. c. GRAPPA weights are estimated from a set of fully sampled kernels from the ACS region. d. Linear combination of acquired samples weighted with GRAPPA weights yields a reconstructed Kspace. e. Fourier Transform is used to retrieve single-coil images

With the rapid advancement of Machine Learning (ML), this project focuses on Neural Implicit Functions (NIF), as a promising alternative for conventional physics-based MRI reconstruction approaches.

This work, is the continuation of previous research that assessed the feasibility of Neural Implicit Functions for k-space reconstruction in a volume-specific and multi-volume training setup. Past experiments were conducted relying on the work of *Huang et al.* on Cardiac MRI reconstruction using Neural Implicit Functions, and evaluated the model's capacity to predict non-sampled regions of the k-space when trained over sampled coordinate points. Different optimization losses, normalization techniques and training data configurations were used. Nonetheless, the work presented some limitations influenced by the high dynamic range of the k-space, that together with the small k-value magnitudes, contributed to the model overfitting to high magnitude regions, and presenting generalization deficiencies over unseen regions.

In this new project, we attempt to overcome these limitations by implementing several approaches on top of the baseline model. We depart from a single volume training perspective, where we explore the addition of a PISCO regularization, moving away from an unsupervised to a self-supervised training of the Neural Network. Furthermore, we explore the benefits of parametric position encodings, teaching the network in a multi-resolution fashion while speeding up the training. In the lines of multi-volume training, we explore the concept of the 'Auto-Decoder' proposed in *DeepSDF*, where we will extend the training of the model from being volume specific, to multiple volumes, guiding the model predictions.

We aim to investigate a series of approaches, related to different training setups and model hyperparameter regimes, to help solve the generalization issue and address the key limitation of previous methods [19, 9].

## 1.2 Report Organization

The report is organized as follows. In Chapter 2, we review related work on NIFs and their application to the ill-posed problem of k-space reconstruction, as well as other promising approaches that will be analyzed in this project. Chapter 3 focuses on volume-specific reconstruction using NIFs. The experiments in this chapter are divided into two parts. First, we implement PISCO regularization and analyze its impact over the model performance. In the second part, hash encodings are incorporated to guide the model to more accurate predictions.

Chapter 4 examines the multi-volume setup. Here, the experiments and results are organized into three sections, each assessing the effect of various hyperparameters on model performance. Finally, Chapter 5 provides an overall analysis of the results, discusses project's limitations, and proposes directions for future work.

## Chapter 2

# Related Work

Neural Implicit Fields (NIF) are continuous implicit neural representations (INR) of structured signals (*e.g.* 2D images, 3D shapes, volumes), parametrized by a function  $f_\theta(x)$  where  $\theta$  corresponds to the learnable weights. The foundations of INRs come from the work of Park et al. [15] which proposes a method of regressing continuous 3D shapes from point samples, using deep neural networks. Effectively, one inputs point coordinates in the corresponding domain (2D, 3D, etc), and the NIF predicts the corresponding value. Tancik et al. [20] proof how encoding the input coordinates of a NIF into fourier feature vectors increases the representational power of the model, which has a natural spectral bias that results in a convergence to lower frequencies being captured better than higher frequencies (details in the image). In this context, Müller et al. [14] present a novel technique to encode the input coordinates, moving away from fixed feature vectors, to a parametric approach. Some of the benefits of continuous representation networks compared to more conventional approaches include smoothness, resolution independence, improved memory efficiency, and differentiability. Additionally, continuous representations enhance robustness to noise and sampling variations, making them particularly useful for applications in medical imaging, and 3D reconstruction [18].

In the context of medical imaging, NIFs have been employed in multiple scenarios, ranging from shape modeling and anatomical structure representation [1], to MRI and CT reconstructions [9, 17]. *Marimont et al.* solve the task of unsupervised anomaly detection by a combination of NIFs and a simplified Auto-Decoder approach, that learns the distribution of healthy images from a training set of multiple volumes. More particularly, the model gets trained by optimizing both the NIF and the Auto-Decoder, learning the optimal latent representation of each volume. At inference time, a restored image is obtained by searching for the latent vector that best matches the healthy distribution while remaining maximally consistent to the input unhealthy image.

Neural Implicit Kspace representation functions (NIKs) are a specialized extension of NIFs that have gained increasing attention in recent years. These models are trained to map continuous k-space coordinates to their corresponding Fourier signal, allowing for the reconstruction of missing data points within the k-space grid. At inference time, any coordinate can in theory be sampled, enabling flexible and data-efficient completion of undersampled k-space.

An example of this strategy is observed in the work of Huang et al. [9], where NIKs are employed to reconstruct undersampled kspace of cardiac MRI k-space using only data from the volume of interest. This eliminates the need for an extensive dataset of training volumes, a common limitation in deep-learning-based MRI reconstruction. However, without proper regularization, NIKs are susceptible to overfitting and noise artifacts in the reconstructed images. To address this, Spieker et al. [19] introduce PISCO, a parallel-imaging-inspired regularization strategy designed to improve the robustness and accuracy of Neural-Implicit-function based k-space reconstruction.



# Chapter 3

## Single-Volume

### 3.1 Description of the problem

MRI reconstruction, in a Multi-coil setup and from an under-sampled K-space, is often framed as an inverse problem and it can be described by the following equation:

$$\hat{x} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda \mathcal{R}(\mathbf{x}) \quad (3.1)$$

Where  $\|\mathbf{y} - A\mathbf{x}\|_2^2$  ensures consistency between the measured data  $y$  (in k-space) and the image to be reconstructed  $x$ . This latter gets multiplied by  $A = U\mathcal{F}S_i$ , or the measurement operator consisting of a sampling matrix ( $U \in \mathbb{C}^N$ ), the Fourier transform operator ( $\mathcal{F}$ ), and a sensitivity map matrix  $S_i$  for the  $i$ -th coil. A regularization term  $\mathcal{R}(\mathbf{x})$  is frequently added to constrain the solution space.

In a NIK scenario, the equation becomes:

$$\theta^* = \arg \min_{\theta} \frac{1}{2} \|\mathbf{y} - G_\theta(\mathbf{v})\|_2^2 \quad (3.2)$$

Where  $\mathbf{y} \in \mathbb{C}$  corresponds to a flattened version of the MRI volume in k-space, that is approximated by the implicit function  $G_\theta$ . This latter maps k-space coordinates  $v \in \mathbb{R}^4$ , to their corresponding Fourier value.

$$G_\theta : \mathcal{K} \rightarrow \mathbb{C} \quad (3.3)$$

Our training objective considers random samples from a Cartesian k-space grid, ignoring the central values (Figure 3.1).

In the Single volume setup, masks of the kind observed in Figure 3.2 are employed to simulate k-space Cartesian undersampling, the model is trained on the acquired samples, and it is tested to predict the whole volume except the center.

### 3.2 Method

#### 3.2.1 Model

The model used in this study consists of a Multi-Layer Perceptron (MLP) with Sinusoidal activation functions, inspired by the work of *Sitzmann et al.* [18].

The input to the model are the coordinates  $k \in (k_x, k_y, k_z, c)$  normalized to the range  $[-1, 1]$ . The coordinates are concatenated with position encoding vectors described by the following expression [20]:

$$\gamma(k) = (\sin(2^0 \pi k), \cos(2^0 \pi k), \dots, \sin(2^{L-1} \pi k), \cos(2^{L-1} \pi k)) \quad (3.4)$$

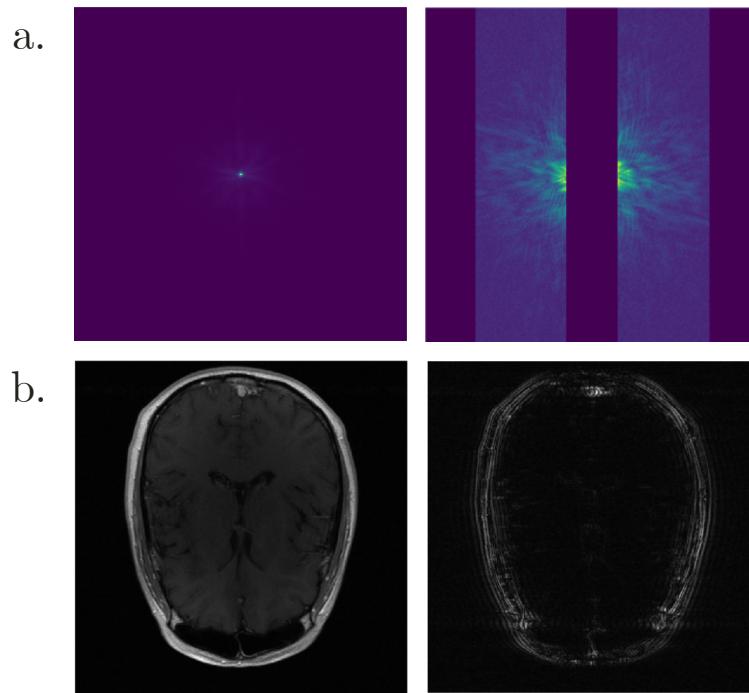


Figure 3.1: **a.** K-space modulus fully sampled (left) and fully sampled without central values (right). **b.** Corresponding inverse Fourier transformations.

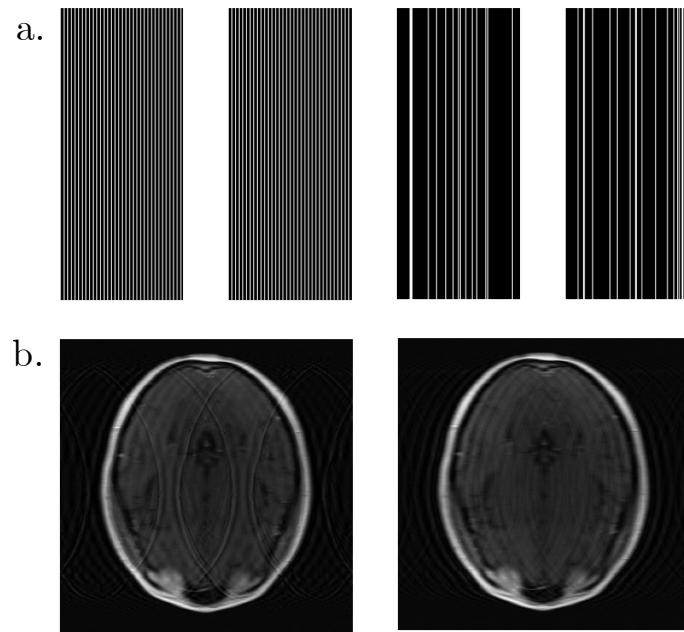


Figure 3.2: **a.** K-space masks with equispaced undersampling (left) and randomly-spaced undersampling (right), acceleration factor  $\times 4$ , center fraction 0.15 . **b.** Resulting inverse Fourier transformations from undersampled k-spaces.

The network outputs a two-dimensional vector representing the real and imaginary parts of the corresponding complex value. Figure 3.3, illustrates this model.

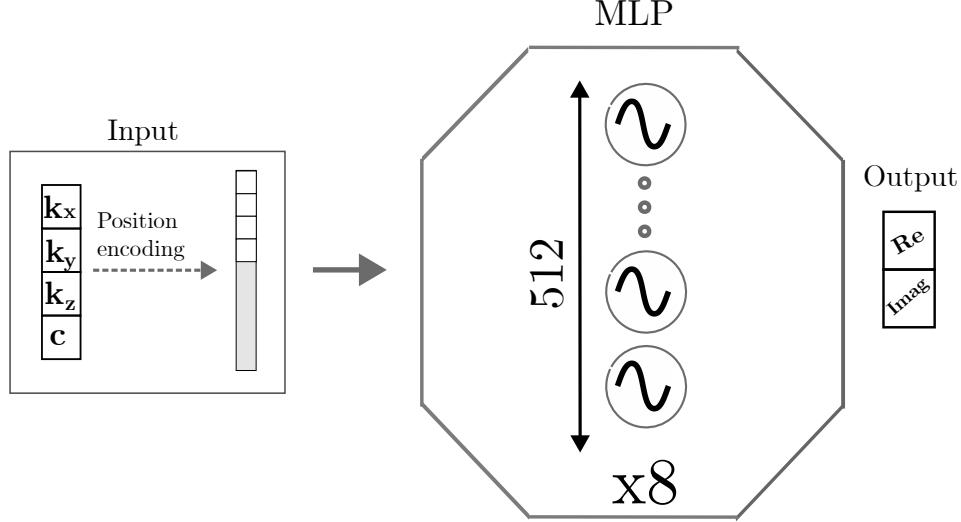


Figure 3.3: Simplified diagram of single volume training pipeline. MLP with 8 hidden layers of dimension 512 and sinusoidal activation functions.

### 3.2.2 Optimization

The optimization objective is the Mean Squared Error (MSE) loss, more explicitly defined by the following expression:

$$\min_{\theta} \frac{1}{|\mathcal{D}|} \sum_{(k_x, k_y, k_z, c, v) \in \mathcal{D}} |G(k_x, k_y, k_z, c; \theta) - \tilde{y}|^2 \quad (3.5)$$

where  $\theta$  represents the parameters of the network,  $|.|$  represents the modulus, and  $\tilde{y} = y/q_{0.999}$  (i.e. the target values are normalized by the 0.999 quantile of the modulus).

## 3.3 Experiments & Results

### 3.3.1 Experimental Setup

#### Dataset:

Three axial post-contrast T1-weighted brain MRI scans were selected from the fastMRI challenge dataset [21]. To speed up training, we only consider 4 slices from each volume. We worked with multi-coil scans; more particularly, the volumes selected come from 4 and 16 coils. Undersampling was simulated using Randomly-spaced masks with a 4x acceleration factor and a 0.15 center fraction.

**Model:**

The MLP has 8 Fully Connected (FC) layers which present a hidden dimension of 512, one skip connection from the input to the 4<sup>th</sup> layer, and a sine activation hyperparameter  $w_0 = 30$ . Fixed position encoding was applied with hyperparameter  $L = 10$ .

**Optimization:**

The optimization parameters are as follows:

- Optimizer: Adam
  - Betas (0.9, 0.999)
  - Epsilon:  $10^{-8}$
  - Learning Rate :  $5 \times 10^{-6}$
- Batch size: 120,000
- Number of epochs: 1,000

**Evaluation Metrics:**

Model performance is evaluated with the metrics of: Peak Signal-to-Noise ratio (PSNR), Structural Similarity Index Measure (SSIM), and the Normalized Mean Squared Error (NMSE). More particularly, we will compare the inverse transformation of the predicted k-space, (with the inserted reference central values), to the volume groundtruth.

### 3.3.2 Baseline

To test the representation capabilities of the reference model (from previous research), we train it on a fully sampled k-space grid, and compare the performance metrics to the ones obtained with an under-sampled k-space training. Table 3.1 displays these values.

|          | Center |       |       | Fully Sampled |       |        | Undersampled |       |       |
|----------|--------|-------|-------|---------------|-------|--------|--------------|-------|-------|
|          | PSNR   | SSIM  | NMSE  | PSNR          | SSIM  | NMSE   | PSNR         | SSIM  | NMSE  |
| Volume 1 | 30.031 | 0.874 | 0.045 | 33.719        | 0.903 | 0.0197 | 30.699       | 0.895 | 0.039 |
| Volume 2 | 30.489 | 0.883 | 0.033 | 33.978        | 0.900 | 0.016  | 31.195       | 0.897 | 0.029 |
| Volume 3 | 31.417 | 0.884 | 0.032 | 35.698        | 0.940 | 0.012  | 32.083       | 0.899 | 0.028 |

Table 3.1: Results from baseline model.

Figure 3.6 illustrates the k-space modulus prediction differences when trained on fully sampled datasets and undersampled datasets. Note that the latter is sparser, the model predicts more non-zero values near the image center, but on the boundaries it outputs zeroes.

These results can also be observed in image-domain. Figure 3.4, displays the results of inserting the reference center values to the the model predictions. Figure3.5 illustrates the outcome of performing the inverse transformation of the model predictions without any insertion. Note that when training on fully sampled datasets, the model is able to predict the edges quite accurately. Nonetheless, when an undersampled dataset is used, almost no added gain is provided by the model predictions (Figure 3.5.).

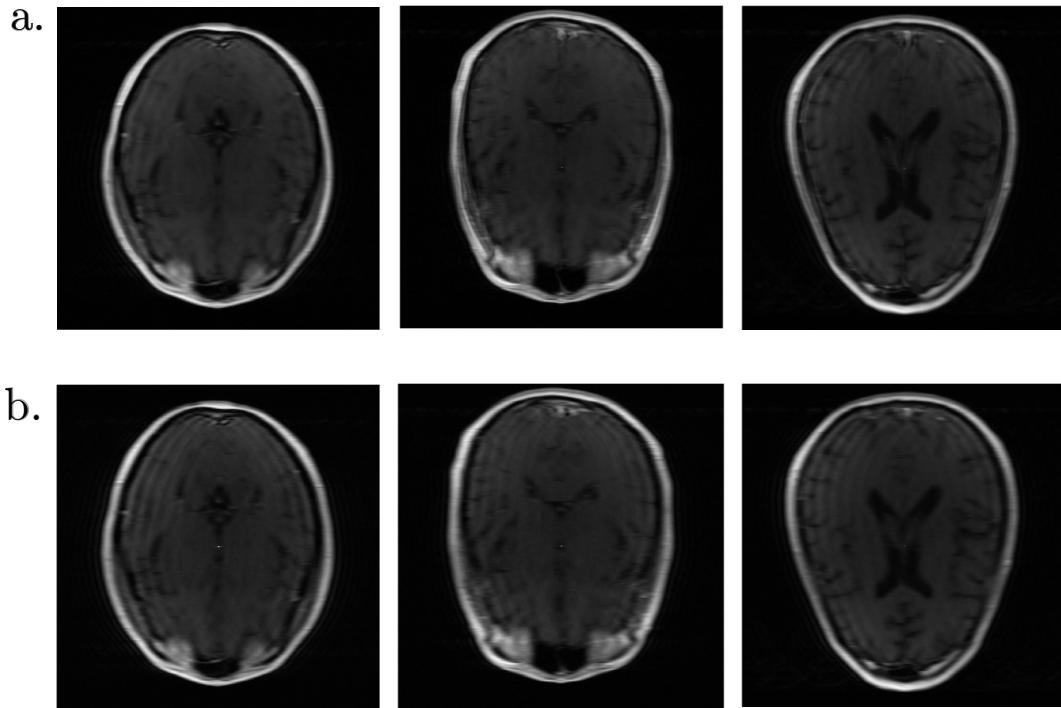


Figure 3.4: Baseline model predictions with center k-space inserted. a. Fully sampled k-space training. b. Under-sampled k-space training.

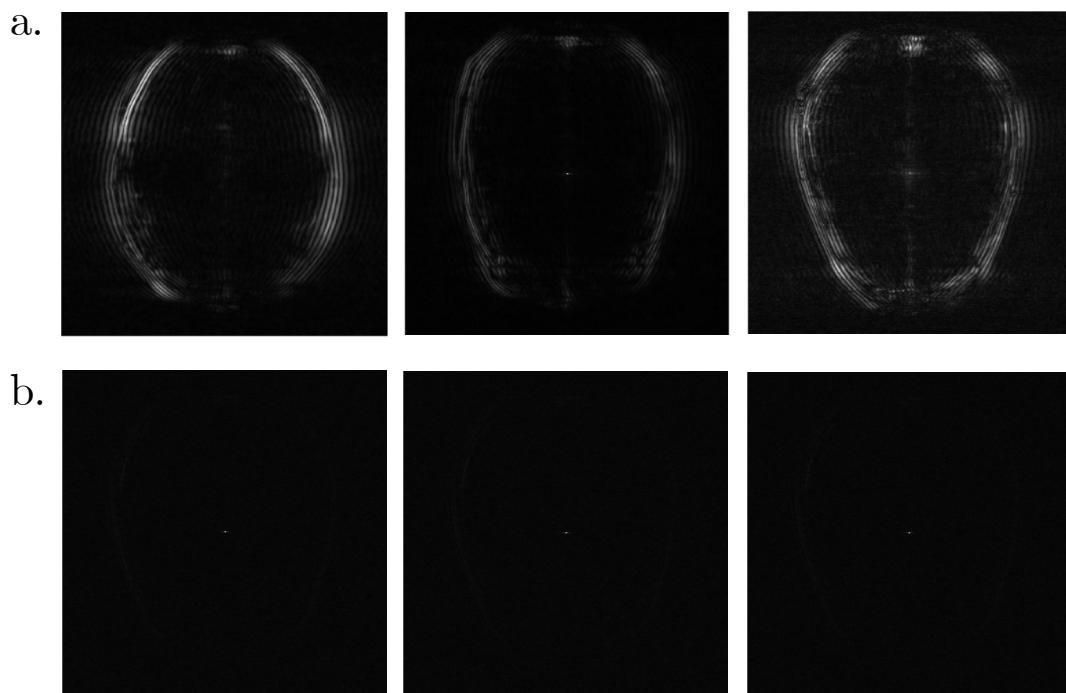


Figure 3.5: Baseline model predictions without the insertion of central k-space values. a. Fully sampled k-space training. b. Under-sampled k-space training.

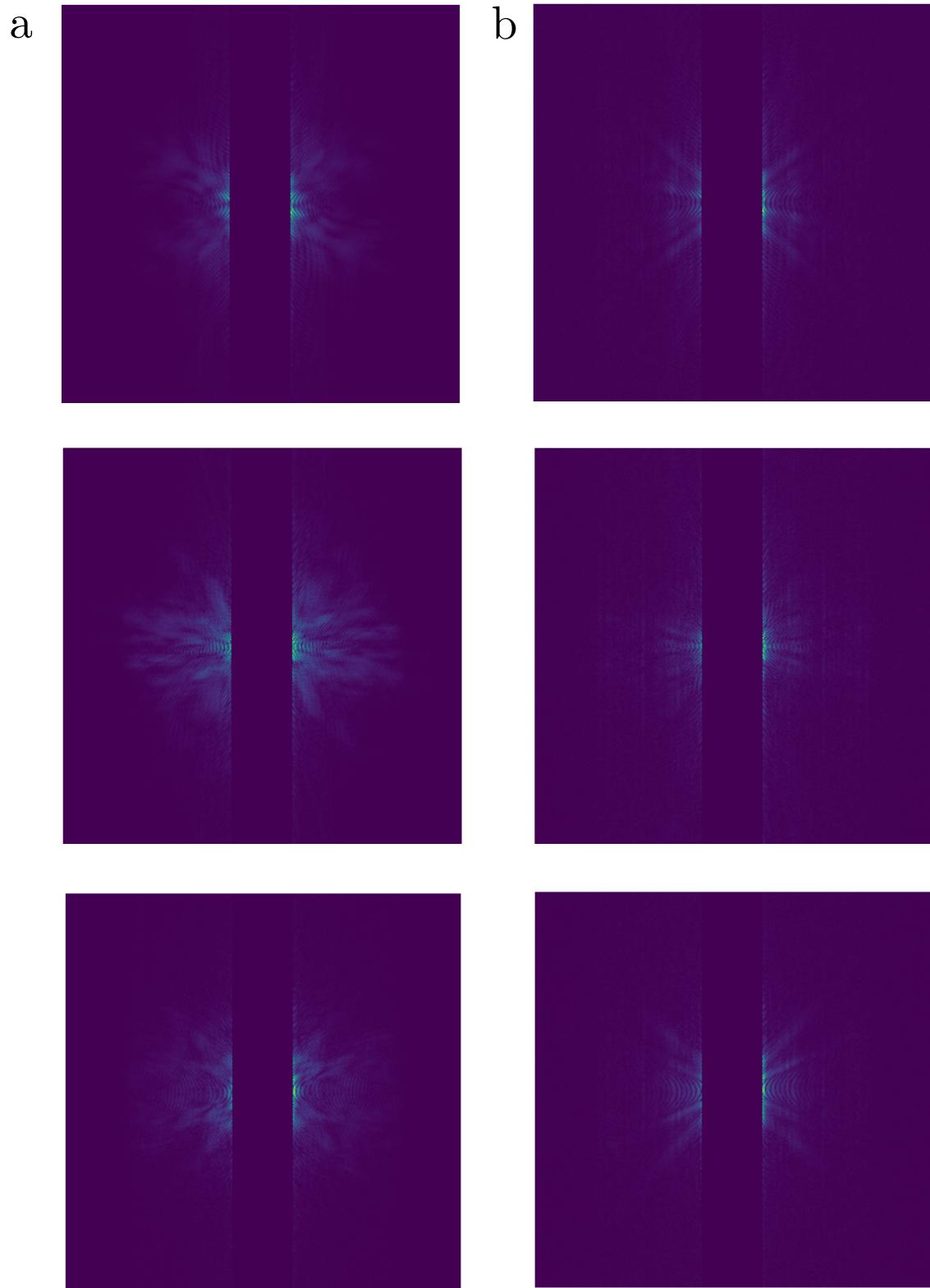


Figure 3.6: Baseline model modulus k-space predictions. a. Fully sampled k-space training. b. Undersampled k-space training.

## Discussion

The drop in performance when training on under-sampled datasets compared to the fully-sampled one, is likely due to the model overfitting to lines near the center of the k-space. This happens because the dataset has a wide range of magnitudes—values at the boundaries are orders of magnitude smaller than those at the center. Even with randomized training, this imbalance causes the model to perform poorly.

### 3.3.3 First Experiment: PISCO regularization

#### Preliminaries

Due to the limitations of the previous approach (i.e. the high dynamic range of k-space values, and the small values issue), we hypothesize that introducing a physics-inspired regularization strategy will improve these issues.

In this first experiment, we aim to train the model with the parallel-imaging inspired regularization strategy proposed in *Spieker et al.*

#### Method

The idea of PISCO is based on the concepts presented in the conventional accelerated MRI strategy of GRAPPA [8]. They suggest that there is a linear relationship between k-space points and their neighbors in the grid, modeled by some weight set  $W_s$ . Additionally, they claim that this linear relationship should be constant for the whole k-space. Based on these ideas, they propose the following expression to derive weight sets:

$$W_s = \arg \min_W \|P_s W - T_s\|_2^2 + \alpha \|W\|_2^2 \quad \text{s.t.} \quad T_s, P_s \subseteq y_s \quad (3.6)$$

Where  $T_s \in \mathbb{C}^{N_m}$  is a vector of k-space values,  $P_s \in \mathbb{C}^{N_n N_m}$  corresponds to a flattened patch of  $n$  neighbors to those targets, and  $W \in \mathbb{C}^{N_n N_m \times N_m}$  is a particular weight matrix relating the two terms. PISCO optimization strategy consists in minimizing the distance between different sets of matrix weights:

$$\mathcal{L}_{PISCO} = \frac{1}{N_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \mathcal{L}_{dist}(W_i, W_j) \quad (3.7)$$

where  $\mathcal{L}_{dist}(W_i, W_j)$  measures the pair-wise  $\mathcal{L}_1$  norm between real and imaginary parts of weight matrixes  $W_i$  and  $W_j$ . Our optimization equation is the one proposed in 3.5, which after a fixed number of epochs, is modified to include this PISCO regularization term (see Algorithm 1). More particularly, the new loss to be optimized becomes:

$$\min_{\theta} = \mathcal{L}_{consistency} + \lambda \cdot \mathcal{L}_{PISCO} \quad (3.8)$$

Algorithm 1 describes the precise training pipeline used, where we set the new hyperparameters for PISCO as :  $\alpha = 1 \times 10^{-6}$ ,  $\lambda = 0.01$ ,  $n_s = 30$ ,  $E_{pisco} = 800$ . These concrete values were selected based on some preliminary experiments.

Note that, due to the high computational demand of this training pipeline, we only trained the models with under-sampled volumes and the batch size had to be reduced to half of the size employed in the baseline experiment (i.e. 60,000). The rest of the optimization parameters remain constant to the baseline experiment.

**Algorithm 1 Neural Implicit k-Space Representation with PISCO [19]**

**Require:** Acquired k-space coordinates  $k^{meas}$  and signal values  $y^{meas}$ , total epochs  $E$ , pretraining epochs  $E_{pre}$ , NIK architecture  $G_\theta$ , initial model parameters  $\theta_0$

```

1: for  $e = 0$  to  $E$  do
2:   for batch  $(k_B, y_B)$  in  $(k^{meas}, y^{meas})$  do
3:      $y_B^{pred} \leftarrow G_{\theta_e}(k_B)$                                  $\triangleright$  Sample batch of coordinates
4:      $\theta_{e+1} \leftarrow L_{DC}(y_B^{pred}, y_B)$                        $\triangleright$  Predict k-space with NIK
5:     if  $e > E_{pre}$  then                                          $\triangleright$  Update model with data consistency loss
6:        $k_B^T, k_B^P \leftarrow k_B$ , patches around  $k_B$                    $\triangleright$  Apply PISCO regularization
7:        $y_B^T, y_B^P \leftarrow G_{\theta_e}(k_B^T), G_{\theta_e}(k_B^P)$            $\triangleright$  Sample coordinate patches
8:        $[P_s]_{s=1}^{N_s}, [T_s]_{s=1}^{N_s} \leftarrow$  divide  $y_B^T, y_B^P$  into multiple subsets
9:        $[W_s]_{s=1}^{N_s} \leftarrow$  solve each subset for weights with regularization  $\alpha$ 
10:       $\theta_{e+1} \leftarrow \lambda \cdot LPISCO([W_s]_{s=1}^{N_s})$ 
11:    end if
12:  end for
13: end for
14: return Learned NIK model  $G_\theta$ 

```

**Results & Discussion**

The baseline model performance when trained with this regularization strategy can be analyzed from the following results. Table 3.2 shows the metric scores. Figure 3.7 illustrates the resulting inverse Fourier transform of the predicted k-space with the inserted central values.

|                 | Center |       |       | Undersampled |       |       |
|-----------------|--------|-------|-------|--------------|-------|-------|
|                 | PSNR   | SSIM  | NMSE  | PSNR         | SSIM  | NMSE  |
| <b>Volume 1</b> | 30.031 | 0.874 | 0.045 | 29.854       | 0.841 | 0.047 |
| <b>Volume 2</b> | 30.489 | 0.883 | 0.033 | 30.149       | 0.839 | 0.036 |
| <b>Volume 3</b> | 31.417 | 0.884 | 0.032 | 31.156       | 0.852 | 0.033 |

Table 3.2: Comparison of PSNR, SSIM, and NMSE between Center and Undersampled data.

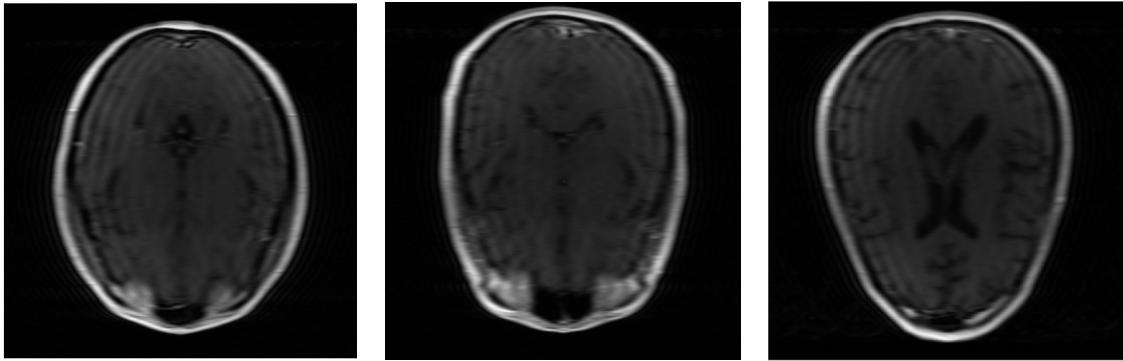


Figure 3.7: Inverse transform of predicted k-space with inserted center values.

The results of the predicted k-space can be appreciated in Panel A from Figure 3.8; the corresponding inverse Fourier transforms without introducing any central k-space values are observed in Panel B of that

same figure.

Note that although the predicted slices look equally as bad as the ones obtained with the baseline model, the metric scores are slightly worse with this regularization. We also noted that the training loss with this pipeline becomes rather unstable. It is also very time consuming to work with this pipeline; extending the training from roughly 2 hours to almost 1.5 days.

### 3.3.4 Second Experiment: From fixed to learnable Position Embeddings

#### Preliminaries

Another strategy we tested to smoothen the weight of the central values over the learning process, which leads to the model being incapable of predicting details, is inspired by the work of Müller *et al.* [14]. We aim to explore the potential of learnable position embeddings or 'Hash tables', and their effect on the learning process.

Our goal is to introduce trainable feature vectors for the input coordinate grids, in a multi-resolution setup, from coarser to finer grids. By back-propagating the error loss over these multi-resolution grids, we aim to teach the model to focus on those areas where the details are encoded.

#### Method

The methodology for encoding the spatial coordinates into the hash table is illustrated in figure 3.9. Given a coordinate  $x$  ( $k_x, k_y, k_z, c$ ) from the k-space grid, we encode the spatial coordinates  $k_x, k_y$  into a feature vector obtained from bilinear interpolation with respect to the closest 4 points at each resolution level (see Figure 3.9). Since coarser resolution grids are sparser than finer ones, lower-resolution features are always updated, while higher-resolution grids update those features that are more relevant to decrease the training loss. Finally, the multi-resolution vectors are concatenated together with the  $k_z$  and  $c$ , forming a vector  $y$  that becomes the new input to the MLP.

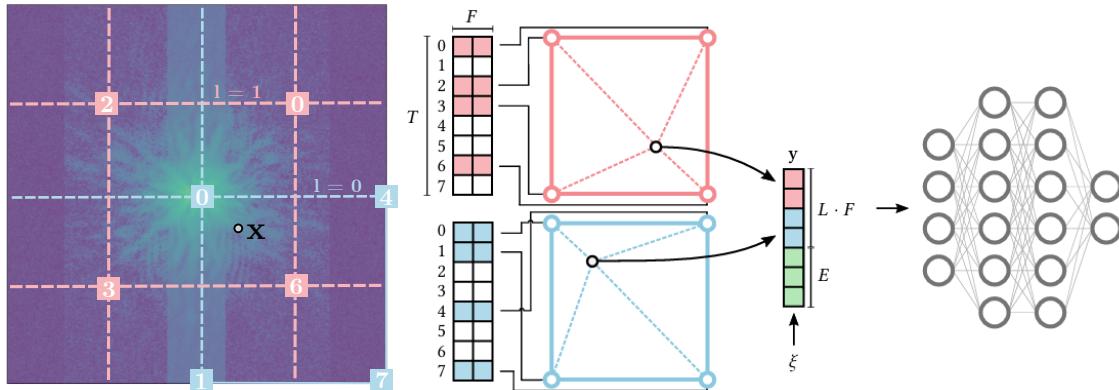


Figure 3.9: Example diagram of the hash encoding of a coordinate  $x$  at two resolution levels.

By introducing this set of new trainable parameters, we can reduce the model depth from 8 to 6 hidden layers, speeding up the training. We will be using hash tables with the following hyperparameters:

| Parameter                                | Symbol     | Value    |
|--|------------|----------|
| Number of levels                         | $L$        | 6        |
| Max. entries per level (hash table size) | $T$        | $2^{12}$ |
| Number of feature dimensions per entry   | $F$        | 3        |
| Coarsest resolution                      | $N_{\min}$ | 45       |
| Finest resolution                        | $N_{\max}$ | 320      |

The optimization parameters are the same as in the baseline experiment. Training is done for 1000 epochs (for fully-sampled k-space grids), and 2000 epochs (for under-sampled k-space grids).

## Results & Discussion

The influence of the hash tables on the model performance can be observed from the results displayed on Table 3.3 and Figures 3.11, 3.12, 3.10.

Similarly to previous experiments, we compare the performance of the model trained on fully-sampled datasets and under-sampled datasets over the same 3 volumes. Figure 3.12 illustrates the k-space modulus prediction differences from the training on both type of datasets. Like in previous experiments, the k-space prediction in the under-sampled dataset looks sparser than in the fully-sampled case.

One can also observe these results in the image domain. As in previous experiments, Figure 3.10 displays the results of inserting the reference center values to the k-space predictions. Similarly, Figure 3.11 illustrates the outcome of performing the inverse Fourier transformation of the model predictions without any k-space center insertion.

|                 | Center |       |       | Fully-sampled |       |       | Undersampled |       |       |
|-----------------|--------|-------|-------|---------------|-------|-------|--------------|-------|-------|
|                 | PSNR   | SSIM  | NMSE  | PSNR          | SSIM  | NMSE  | PSNR         | SSIM  | NMSE  |
| <b>Volume 1</b> | 30.031 | 0.874 | 0.045 | 39.098        | 0.965 | 0.006 | 30.681       | 0.897 | 0.039 |
| <b>Volume 2</b> | 30.489 | 0.883 | 0.033 | 38.009        | 0.963 | 0.009 | 30.970       | 0.887 | 0.031 |
| <b>Volume 3</b> | 31.417 | 0.884 | 0.032 | 37.918        | 0.963 | 0.007 | 32.084       | 0.899 | 0.027 |

Table 3.3: Comparison of PSNR, SSIM, and NMSE across different sampling methods.

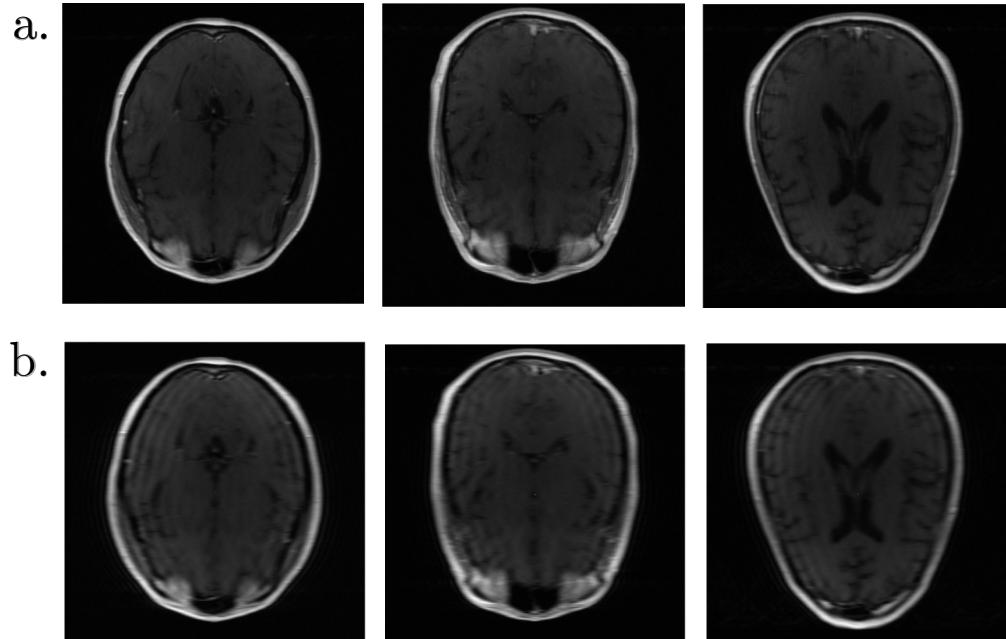


Figure 3.10: Reconstructions from model predictions with hash encodings, with inserted center k-space values. a. Fully sampled k-space training. b. Under-sampled k-space training.

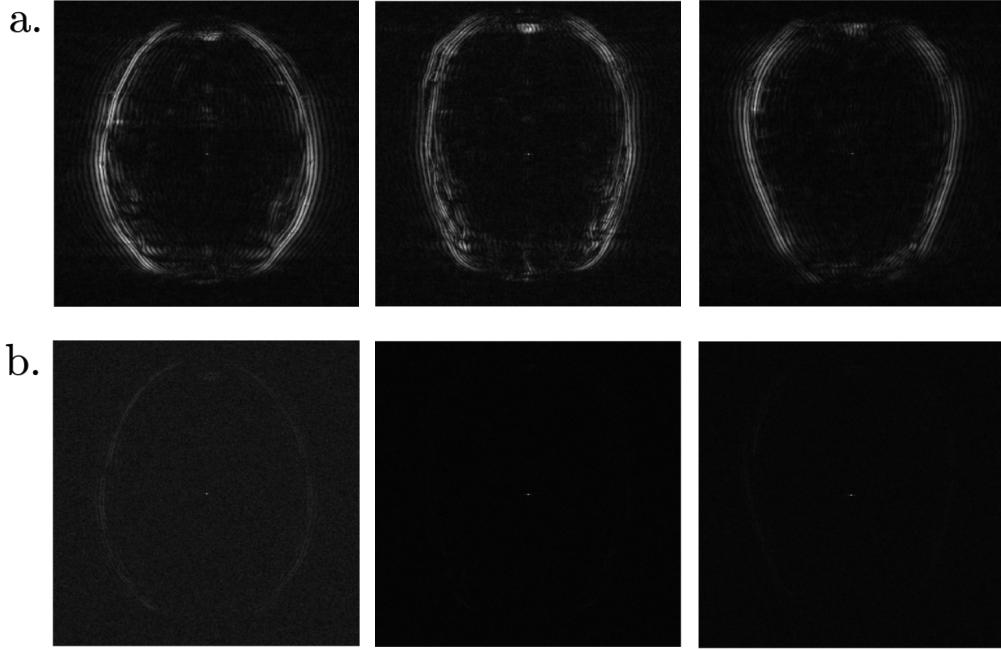


Figure 3.11: Reconstructions from model predictions with hash encodings. a. Fully sampled k-space training. b. Under-sampled k-space training.

Compared to previous experiments, hash encodings lead to an improvement of the model performance on fully sampled data, in fact, k-space predictions contain less zeroes on the boundaries, which result in volumes with higher image quality and almost no visible artifacts. Metrics scores from Table 3.3 (fully-sampled) quantify this improvement.

However, when trained with undersampling masks, despite the scores being better than those obtained with PISCO regularization, they are not improving the baseline.

Note that as illustrated in Figure 3.11.b, the output is roughly just a black image, this indicates that the images observed in Figure 3.10.b, are just the result of inverting the injected central kspace values, as the model predictions in the rest of kspace are barely contributing to the resulting images.

Note that this happens despite the model being trained for 2000 epochs over under-sampled k-space grids (double the number of steps involved in training the baseline and PISCO regularized models).

The significant drop in performance between fully-sampled and under-sampled grids, can be due to the fact that over non-sampled k-space positions, feature vectors corresponding to the nearest hashed points will not be updated as often. Consequently, the model will not follow the same criterion as in the fully-sampled k-space, and it will be incapable of focusing in predicting the details as accurately as in the previous case.

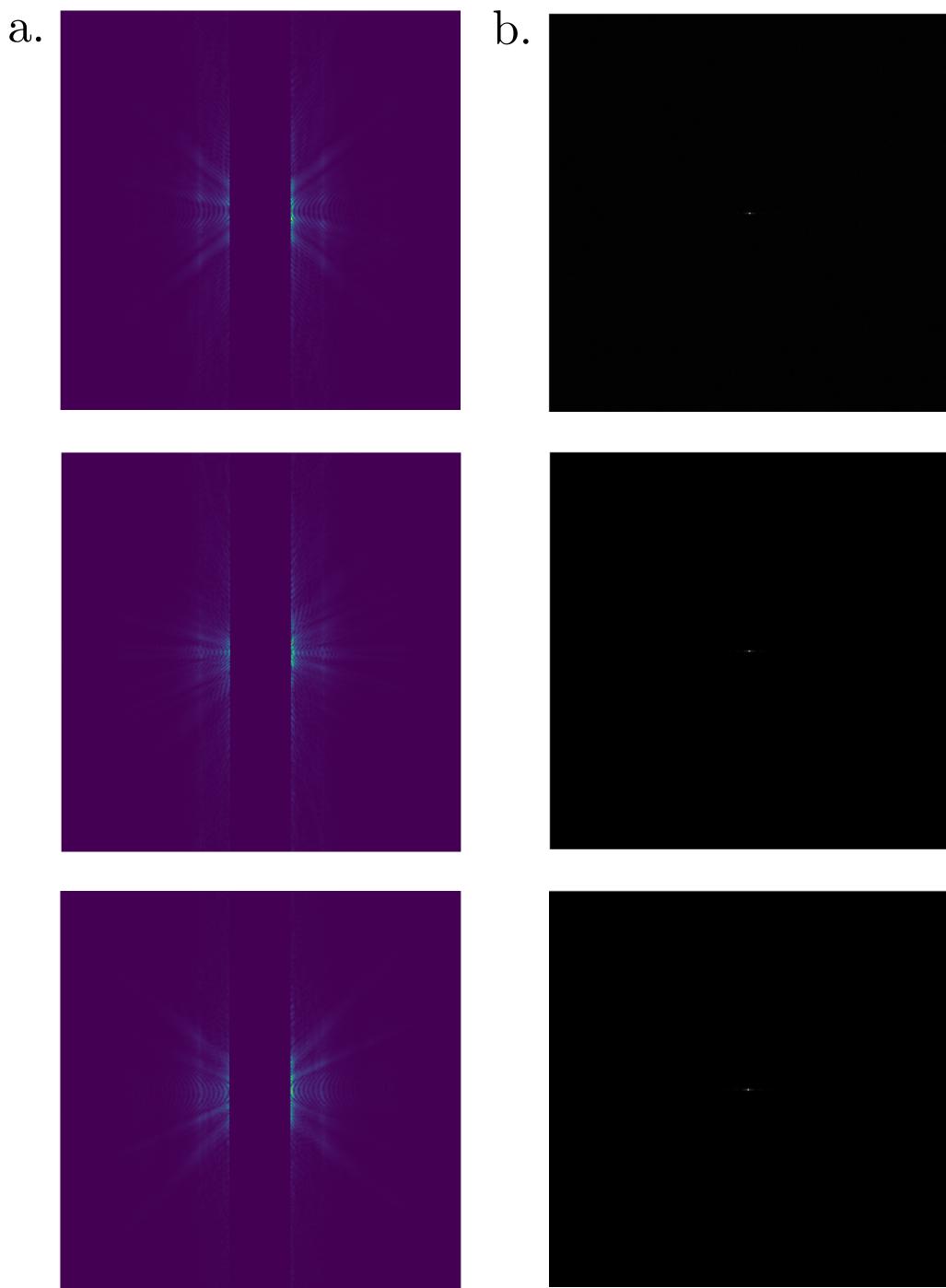


Figure 3.8: Model predictions with PISCO regularization. a. Modulus k-space model prediction. b. Inverse fourier transformation of model prediction.

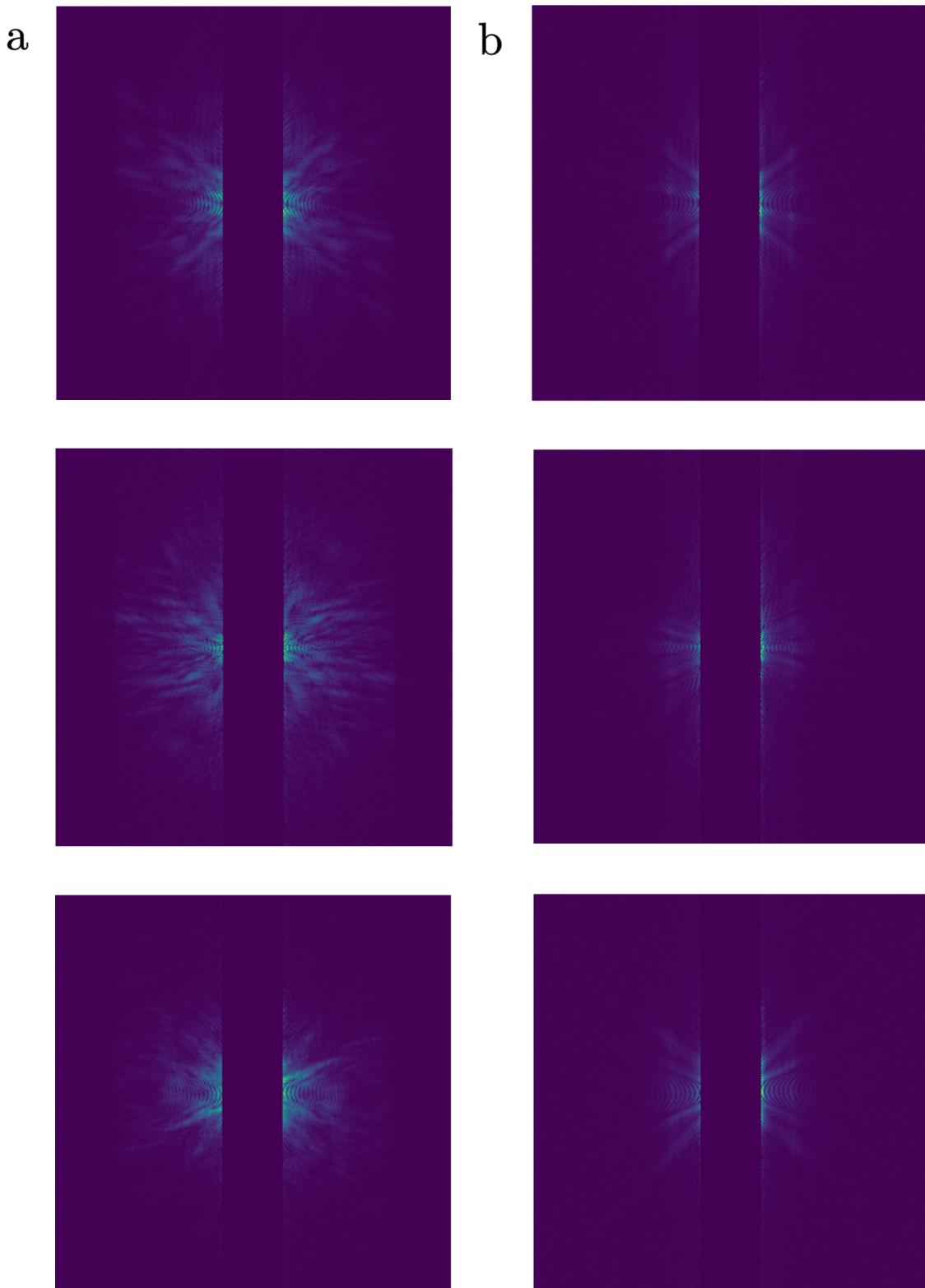


Figure 3.12: k-space modulus predictions of model with hash encodings. a. Fully sampled k-space training.  
b. Undersampled k-space training.

# Chapter 4

## Multi-Volume

### 4.1 Description of the Problem

Single-scan k-space reconstruction faces limitations, primarily related to time and generalization constraints, as it requires retraining the model for each new volume. In this second phase of the project, we are set to train on multiple fully-sampled K-space volumes, optimizing both the model weights and additional embeddings specific to each volume. This way, we attempt to divide the tasks of spatial frequency encoding (learned by the MLP weights) and general shape/ contrast information (summarized in the volume embeddings).

### 4.2 Method

Preliminary studies encoded only the volume information by means of an embedding vector. Since we are dealing with a multi-channel MRI problem, where we have additional coil sensitivity information, we will extend the embedding space to encode both volume information and coil sensitivity maps.

The methodology followed in this second approach shares several commonalities with the single-volume reconstruction. We normalize the input  $k_x, k_y, k_z$  coordinates to the range  $[-1, 1]$  and target values are also normalized by the 0.999 quantile of the modulus. The model used is illustrated by the diagram in Figure 4.1. Note that besides the addition of the embedding vectors, the model architecture remains the same as one used in the Single-Volume case; MLP with 8 hidden layers of dimension 512 and one skip connection from the input to the 4<sup>th</sup> layer.

Volume and Coil embeddings are initialized as follows:

$$\mathbf{z}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \quad (4.1)$$

$$\mathbf{c}_j \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \quad (4.2)$$

where  $\mathbf{z}_i$  and  $\mathbf{c}_j$  represent the  $i$ -th volume and  $j$ -th coil embedding vectors,  $\mathcal{N}$  denotes the gaussian distribution,  $\sigma^2$  is the variance and takes a constant value of 0.01 in all experiments, and  $\mathbf{I}$  is the identity matrix. We experiment with both fixed and learnable position encodings. In the latter case, we implement a grid of hash tables per volume.

### 4.3 Optimization

Two optimization stages are considered.

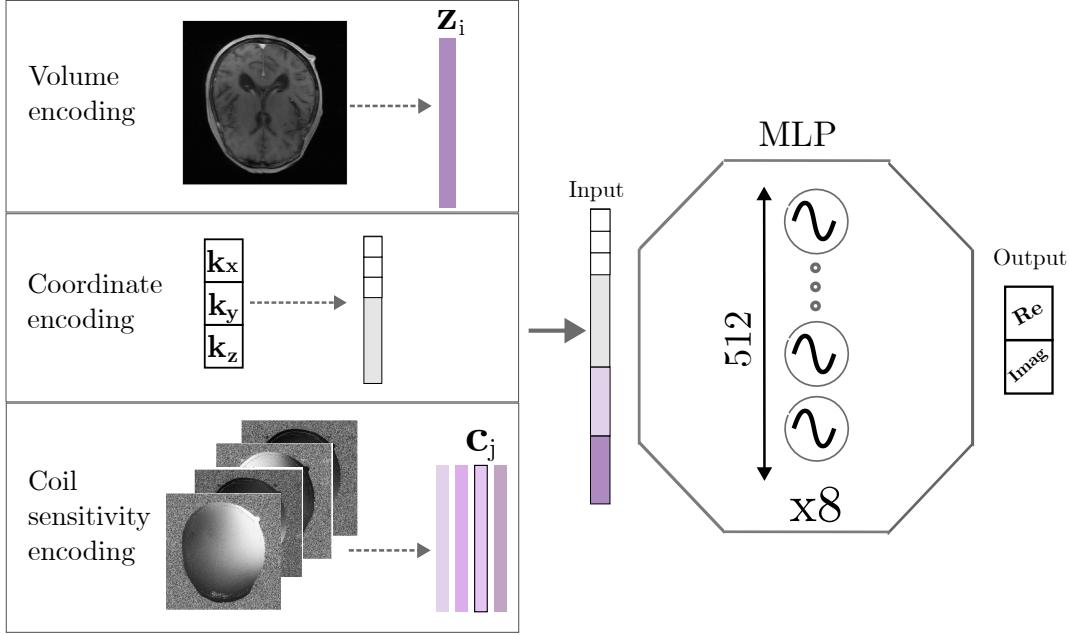


Figure 4.1: Simplified diagram of multivolume training pipeline.

## Training

During the training phase, fully sampled volumes are employed while excluding the central k-space values. In all experiments, two slices are used. The optimization problem is formulated as follows:

$$\min_{\theta, \mathbf{z}_i, \mathbf{c}_j} \frac{1}{|\mathcal{D}_1| + \dots + |\mathcal{D}_N|} \sum_{i=1}^N \sum_{j=1}^{C_i} \sum_{(k_x, k_y, k_z) \in \mathcal{D}_i} \left| \hat{f}(k_x, k_y, k_z; \theta, z_i, c_j) - \hat{y} \right|^2 \quad (4.3)$$

where  $C_i$  is the total number of coils of a particular volume  $n_i$  in  $N$ , and  $\mathcal{D}_i$  represents the number of data-points for the  $i$ -th volume. Thus the training objective jointly optimizes the model parameters  $\theta$ , along with the latent embeddings for volumes ( $z_i$ ) and coils ( $c_j$ ). The optimization parameters used are the following:

- Optimizer: Adam
  - Betas (0.9, 0.999)
  - Epsilon:  $10^{-8}$
  - Learning Rate :  $5 \times 10^{-6}$
- Batch size: 240'000
- Training epochs: 400

## Inference

At inference time, the model is evaluated on two slices from under-sampled volumes. Equispaced masks with an acceleration factor of  $\times 4$  and center fraction of 0.15 are used. Unlike during training, the MLP weights remain frozen, and only latent vectors  $\mathbf{z}_i$  and  $\mathbf{c}_j$  are optimized. The loss function in this stage is

given by:

$$\min_{\mathbf{z}_i, \mathbf{c}_j} \frac{1}{|\mathcal{D}_1| + \dots + |\mathcal{D}_N|} \sum_{i=1}^N \sum_{j=1}^{C_i} \sum_{(k_x, k_y, k_z) \in \mathcal{D}_i} \left| \hat{f}(k_x, k_y, k_z; \theta, z_i, c_j) - \hat{y} \right|^2 \quad (4.4)$$

In the cases where trainable coordinate encodings are used, optimization extends over  $\Phi$ , which represents the tensor of weights across all resolution levels and all the volume-specific hash tables.

Embeddings are initialized at inference from a mean vector. For volume embeddings, this vector represents an average volume estimate; for coil embeddings, it represents an average coil representation. In the experiments where Hash embeddings are also optimized, we initialize the Hash tables for all volumes from a level-based average feature representation.

Adam optimizer parameters remain constant with respect to the training phase. Optimization is done for 1000 and 2000 epochs in some experiments, with batches of sizes : 60'000 and 240'000 data points.

## 4.4 Experiments

In the multi-volume setup, and with the introduction of the volume and coil embeddings, new degrees of freedom are added to the model. We aim to analyze the effect of a series of hyperparameters (*i.e.* number of training samples, size of embedding vectors, training dynamics and regularization strategies), comparing the performance of the model at training vs inference.

### 4.4.1 Number of training samples

#### Preliminaries

We attempt to evaluate the influence of the number of training samples over the model performance. We hypothesize that larger training sample sizes enhance the generalization capabilities of the model, as it learns a 'mean' expected mapping of coordinates to k-space values.

We experimented with training sample sizes of 5, 15, 20 and 30 volumes, keeping the volume and coil embedding vectors fixed to sizes of 256 and 128 respectively.

Training is done for a total of 400 epochs. Inference takes place by optimizing the embeddings for 1000 epochs with 5 sample volumes from the training set, and 4 samples from a validation set.

#### Results & Discussion

Results from this first experiment are displayed on Table 4.4.1 and Figures 4.2 and 4.3. More particularly, Figure 4.2 shows the inverse transform of the model predictions over one sample from the training distribution. Similarly, Figure 4.3 shows the inverse transform of the kspace composed by model predictions and inserted central values for the same sample of the training set.

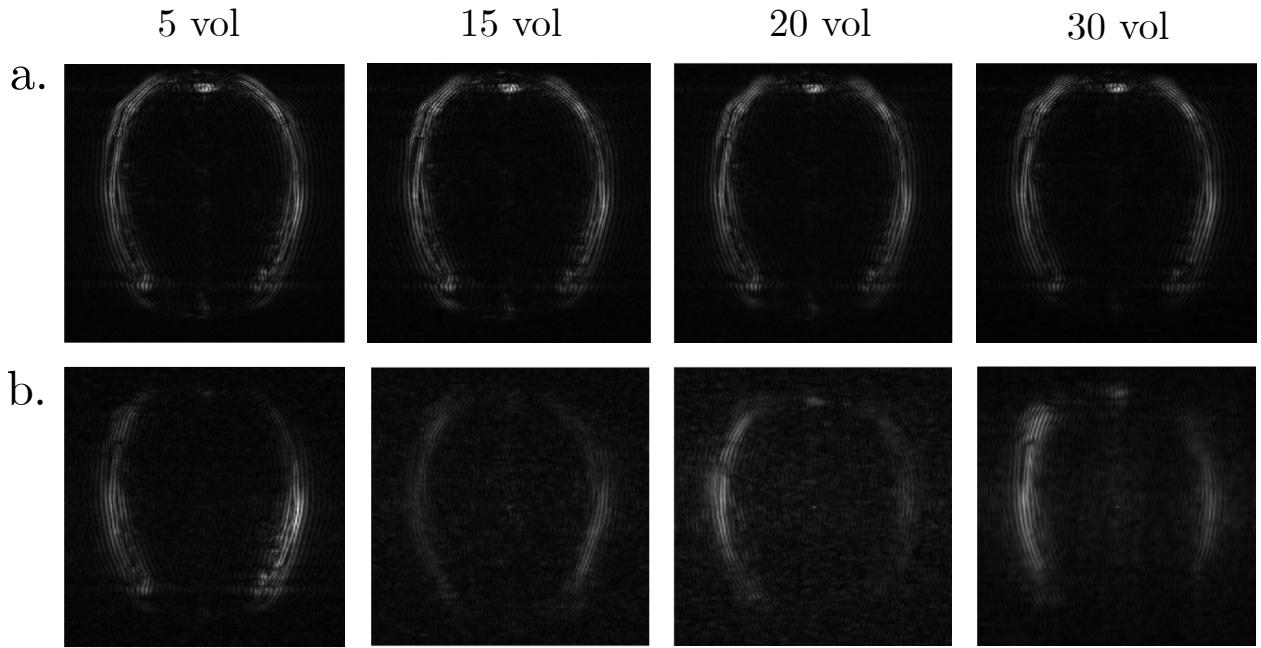
During training, the models converge to sufficiently accurate solutions—as illustrated in Figures 4.2.A and 4.3.A—but their performance degrades sharply at inference time, similarly to the Single-Volume case. This decline is evident in Table 4.4.1, which reports average metrics across multiple volumes. Note that the *Baseline* values represent the metric scores obtained by comparing the inverse transform of the k-space center values (without any model predictions) with respect to the ground-truth volumes.

Increasing the number of training samples appears to regularize the model, as indicated by a proportional decrease in training scores with a higher number of samples. One might therefore expect less over-fitted models to perform better at inference time. However, this is not the case.

Inference performance remains similarly poor in all models both over training and validation samples, slightly improving the baseline scores.

|                      | <i>Training</i> |        |       | <i>Inference</i>      |               |              |                         |               |              |
|----------------------|-----------------|--------|-------|-----------------------|---------------|--------------|-------------------------|---------------|--------------|
|                      |                 |        |       | Training Samples (x5) |               |              | Validation Samples (x5) |               |              |
| <i>n# samples</i>    | NMSE            | PSNR   | SSIM  | NMSE                  | PSNR          | SSIM         | NMSE                    | PSNR          | SSIM         |
| 5                    | 0.008           | 36.377 | 0.944 | 0.035                 | 30.568        | 0.858        | 0.035                   | 30.538        | 0.851        |
| 15                   | 0.011           | 35.333 | 0.936 | 0.036                 | 30.479        | 0.857        | 0.034                   | 30.570        | 0.851        |
| 20                   | 0.011           | 35.266 | 0.936 | 0.038                 | 30.205        | 0.853        | 0.035                   | 30.538        | 0.847        |
| 30                   | 0.022           | 32.203 | 0.884 | 0.038                 | 30.349        | 0.853        | 0.035                   | 30.548        | 0.846        |
| <i>Center scores</i> |                 |        |       | <b>0.044</b>          | <b>29.434</b> | <b>0.832</b> | <b>0.035</b>            | <b>30.466</b> | <b>0.841</b> |

Table 4.1: Training and Inference scores from first experiment

Figure 4.2: Reconstruction of volume from training distribution by models trained on various *n#* training samples a. Training phase. b. Inference phase

#### 4.4.2 Embeddings Dimension

##### Preliminaries

Given that *n#* of samples is actually not a determinant factor in model performance, in this second experiment we aim to assess the influence of the embeddings size. We fix the training set to 15 samples and vary the embedding sizes. Specifically, we test coil embeddings with sizes 128, 256, 512 and volume embeddings with sizes 256, 512, 1024. Additionally we evaluate the effect of Hash encodings in this new setup, defining one Hash table per training volume. We work with tables of 15 resolution levels (from coarsest resolution of 45 pixels to the finest resolution being 320 pixels). We also study the effect of different hash feature dimensions, by training models with 10 and 5 features per coordinate.

We expect to see better generalization capabilities at inference when increasing the embedding space.

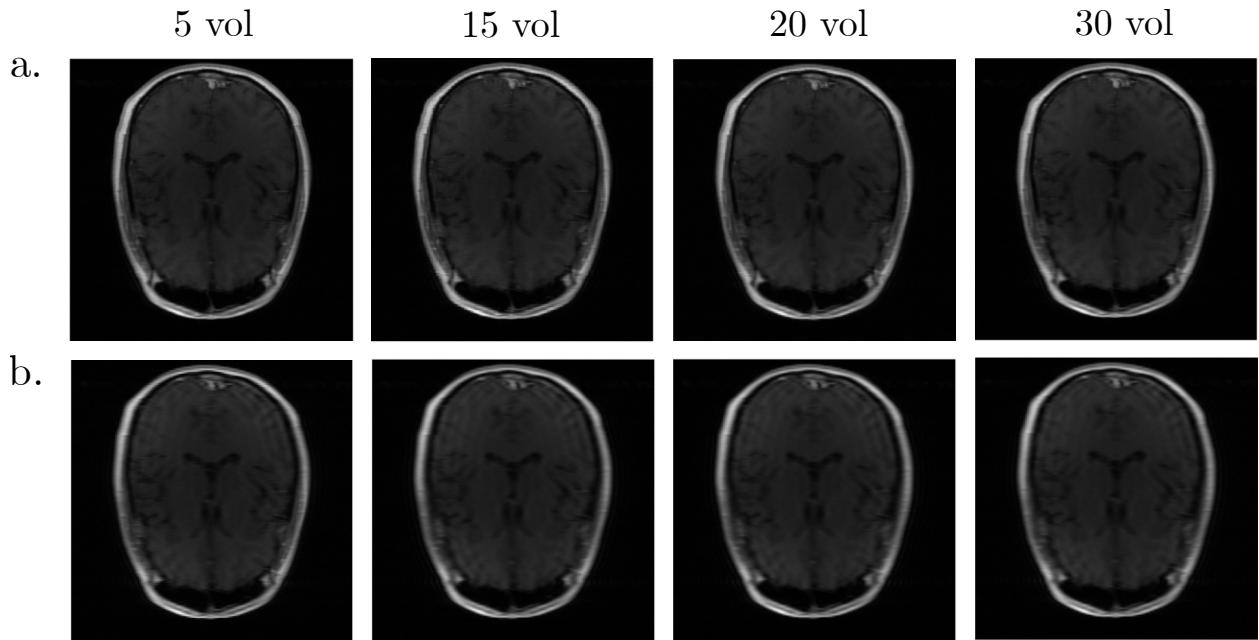


Figure 4.3: Reconstruction of volume from training distribution by models trained over various  $n\#$  training samples, inserting the central  $k$ -space values. a. Training phase. b. Inference phase

## Results & Discussion

Tables 4.4.2 and 4.3 display the results for training and inference of the models with different sets of embedding dimensions as well as the Hash-based model results. Figures 4.4 and 4.5 equally illustrate these results. The same criterion and slice from the first experiment is used for plotting.

Consistent with the previous experiment, the models converge to accurate solutions during training, achieving comparable scores across all embedding dimensions (see Table 4.4.2 and Figures 4.4.A and 4.5.A). However, in contrast to the Single-scan fully-sampled training—where hash-based models outperformed those based on fixed embeddings—the multi-volume setup does not exhibit this advantage. In fact, scores deteriorate when optimizing hash tables alongside volume and coil embeddings during training. A likely explanation is that the additional trainable parameters introduce an implicit regularization effect, with too many parameters to optimize, the MLP weights don't overfit to the training volumes, most of the effort is placed on optimizing the embeddings, thereby yielding lower training scores.

This reasoning suggests that larger embedding sizes should yield better inference scores. Indeed, when optimizing only coil and volume embeddings over samples coming from the training distribution, the model with the largest embeddings—512 for coils and 1024 for volume—performs best. Conversely, cases using smaller embeddings (i.e., volume embeddings of size 512, and no coil embeddings) result in the worst inference performance, with these examples visibly exhibiting the greatest number of artifacts compared to the optimal configuration.

The hypothesis that increasing the embedding space enhances inference performance, does not hold for hash encodings, where almost no model prediction is visible (see Figure 4.4.B.Hash). In fact, the reconstructed volumes for hash encodings display patterns similar to those observed with the smallest volume embeddings (see Plots 4.5.B.Hash). This issue may be related to the suboptimal updating of multi-resolution features over undersampling patterns, already observed in the Single-Volume training.

Furthermore, although larger embedding sizes yield a slight improvement in scores, the reconstructed slices still exhibit noticeable artifacts, even for the cases the model has seen during training. This implies that

the converged loss gets stuck in a suboptimal solution; one possible explanation is that an excessively large embedding space causes the model to converge to a local minimum rather than the true optimal solution.

| Config          | Size vol Embedding | Size coil Embedding | NMSE  | PSNR   | SSIM  |
|-----------------|--------------------|---------------------|-------|--------|-------|
| Vol embd        | 512                | -                   | 0.010 | 35.394 | 0.932 |
| Vol + coil embd | 256                | 128                 | 0.011 | 35.33  | 0.936 |
| Vol + coil embd | 512                | 256                 | 0.010 | 35.493 | 0.938 |
| Vol + coil embd | 1024               | 512                 | 0.011 | 35.33  | 0.934 |
|                 | n # levels         | Feature dim         |       |        |       |
| Hash embds      | 15                 | 10                  | 0.019 | 33.014 | 0.911 |
|                 | 15                 | 5                   | 0.014 | 34.158 | 0.922 |

Table 4.2: Training scores from second experiment

| Config               | Volume embedding | Coil embedding | Inference             |               |              |                         |               |              |
|----------------------|------------------|----------------|-----------------------|---------------|--------------|-------------------------|---------------|--------------|
|                      |                  |                | Training samples (x5) |               |              | Validation samples (x4) |               |              |
|                      |                  |                | NMSE                  | PSNR          | SSIM         | NMSE                    | PSNR          | SSIM         |
| Vol embd             | 512              | -              | 0.047                 | 29.215        | 0.841        | 0.031                   | 30.994        | 0.85         |
| Vol + coil embd      | 256              | 128            | 0.036                 | 30.479        | 0.857        | 0.032                   | 30.852        | 0.852        |
| Vol + coil embd      | 512              | 256            | 0.035                 | 30.645        | 0.859        | 0.032                   | 30.913        | 0.853        |
| Vol + coil embd      | 1024             | 512            | 0.033                 | 30.956        | 0.858        | 0.032                   | 30.935        | 0.851        |
| N resolution levels  |                  |                | N features/ point     |               |              |                         |               |              |
| Hash embds           | 15               | 10             | 0.045                 | 29.496        | 0.833        | 0.033                   | 30.721        | 0.842        |
|                      | 15               | 5              | 0.045                 | 29.496        | 0.832        | 0.033                   | 30.721        | 0.842        |
| <i>Center scores</i> |                  |                | <b>0.044</b>          | <b>29.434</b> | <b>0.832</b> | <b>0.033</b>            | <b>30.764</b> | <b>0.842</b> |

Table 4.3: Inference results second experiment

#### 4.4.3 Stochasticity and Regularization

##### Preliminaries

Excessively large embeddings, lead to non-smooth loss functions, which are hard to optimize, as a result one may fall into local minima instead of the actual optimum. We hypothesize that guiding the model through optimization strategies (i.e. regularization, stochasticity), may lead to better optima.

Our experiments utilize models trained on 15 fully sampled volumes, using volume embeddings of size 1024 and coil embeddings of size 512. Training is done for 400 epochs, with batches of 240'000 data points. We explore the effect of regularizing volume and coil embeddings introduced to the optimization function:

$$\min_{\theta, \mathbf{z}_i, \mathbf{c}_j} \frac{1}{|\mathcal{D}_1| + \dots + |\mathcal{D}_N|} \sum_{i=1}^N \sum_{j=1}^{C_i} \sum_{(k_x, k_y, k_z) \in \mathcal{D}_i} \left| \hat{f}(k_x, k_y, k_z; \theta, z_i, c_j) - \hat{y} \right|^2 + \gamma \frac{\|\mathbf{z}\|_2^2 + \|\mathbf{c}\|_2^2}{2} \quad (4.5)$$

Where  $\gamma = 0.01$ .

Another strategy we test to converge to more optimal solutions, makes use of *PAC-Bayesian* theory, that introduces stochasticity into the embeddings. We hypothesize that by injecting noise, sampled equally from a gaussian  $\mathcal{N}(0, \gamma)$ , where  $\gamma = \sigma_i^2$  (the standard deviation of the embedding vectors), we would be able to avoid local minima. In this line of reasoning, we also explore the effect of smaller batch sizes, given that Adam optimizer tends to perform best with medium-sized mini-batches [22].

Inference is performed over 5 under-sampled volumes from the training distribution, for 1000 epochs over batches of 240'000 and 60'000 data-points.

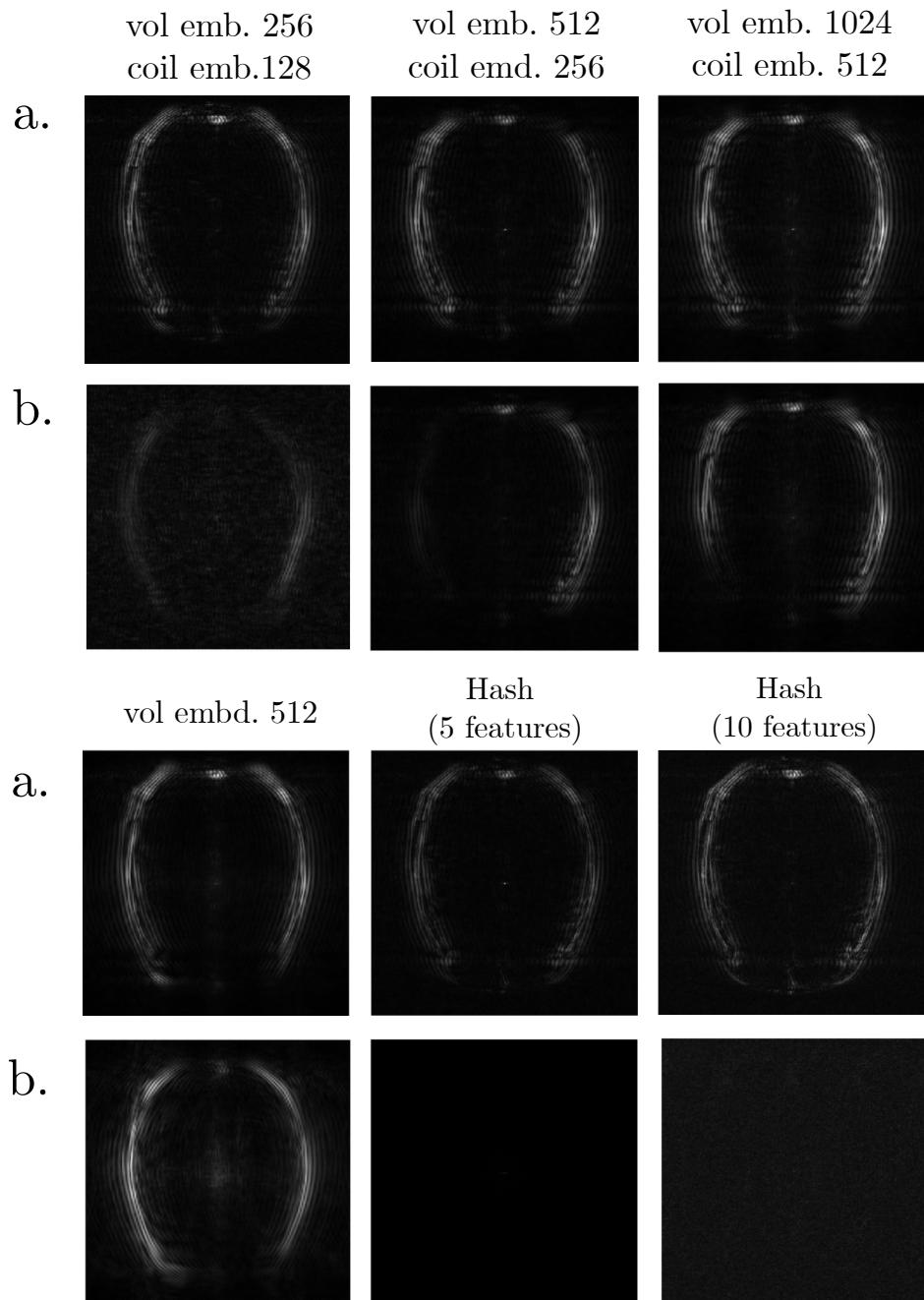


Figure 4.4: Reconstruction of volume from training distribution, by models trained with different embedding sizes. a. Training phase. b. Inference phase.

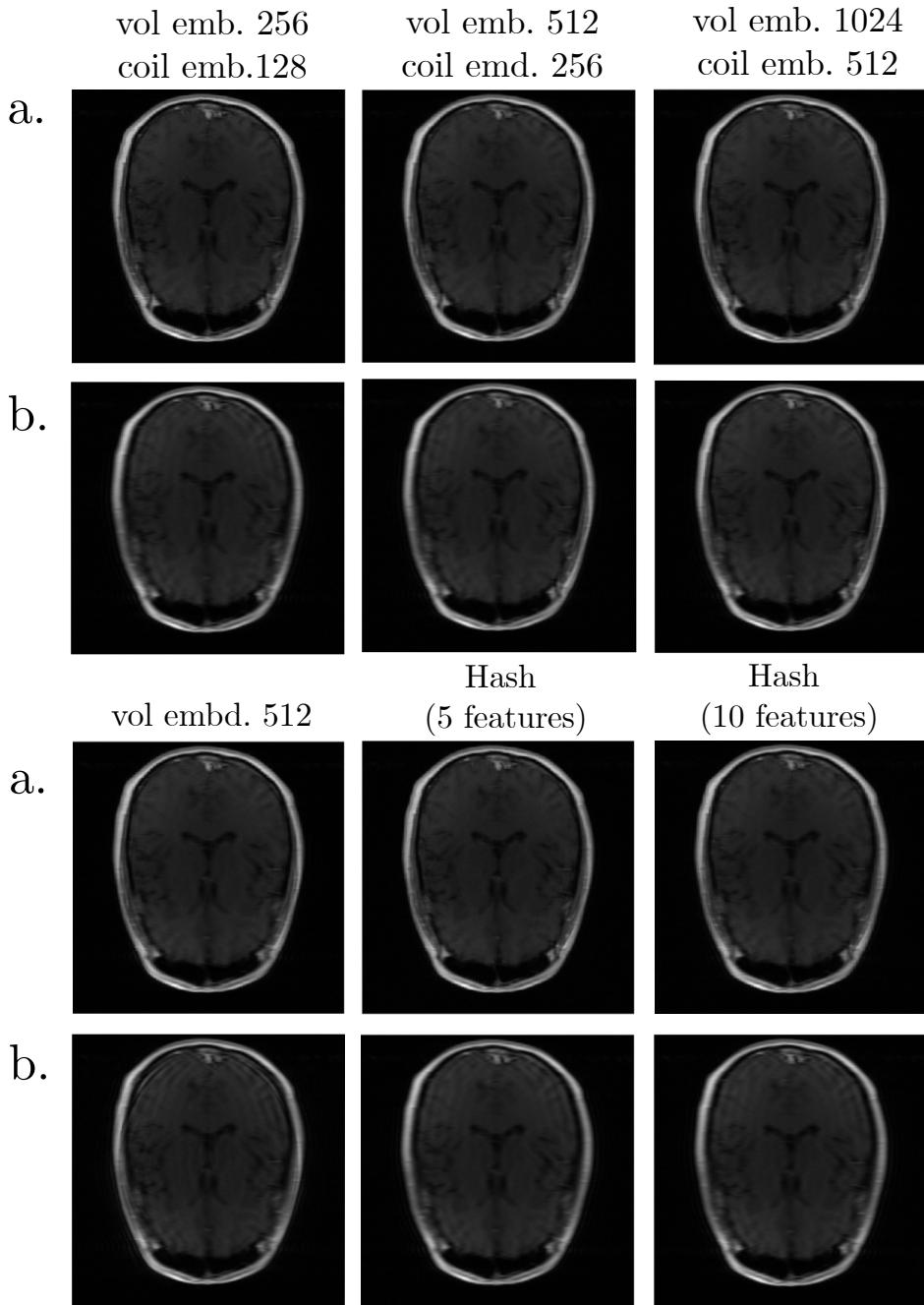


Figure 4.5: Reconstruction of volume from training distribution by models trained with various embedding sizes, with center k-space values inserted. a. Training phase. b. Inference phase.

## Results & Discussion

Results from this last experiment are observed in Table 4.4. In Figure 4.6, panel A displays the inverse transform of the model's prediction, panel B shows the inverse transform of the k-space data containing the central values as well.

| <b>Config</b>        | <b>NMSE</b>  | <b>PSNR</b>   | <b>SSIM</b>  |
|----------------------|--------------|---------------|--------------|
| Batch size 240'000   | 0.033        | 30.956        | 0.858        |
| Batch size 60'000    | 0.033        | 31.022        | 0.859        |
| Noise to embeddings  | 0.044        | 29.563        | 0.850        |
| Noise to model       | 0.044        | 29.574        | 0.851        |
| Reg $\gamma = 0.01$  | 0.450        | 19.533        | 0.248        |
| <b>Center scores</b> | <b>0.044</b> | <b>29.435</b> | <b>0.832</b> |

Table 4.4: Inference results third experiment

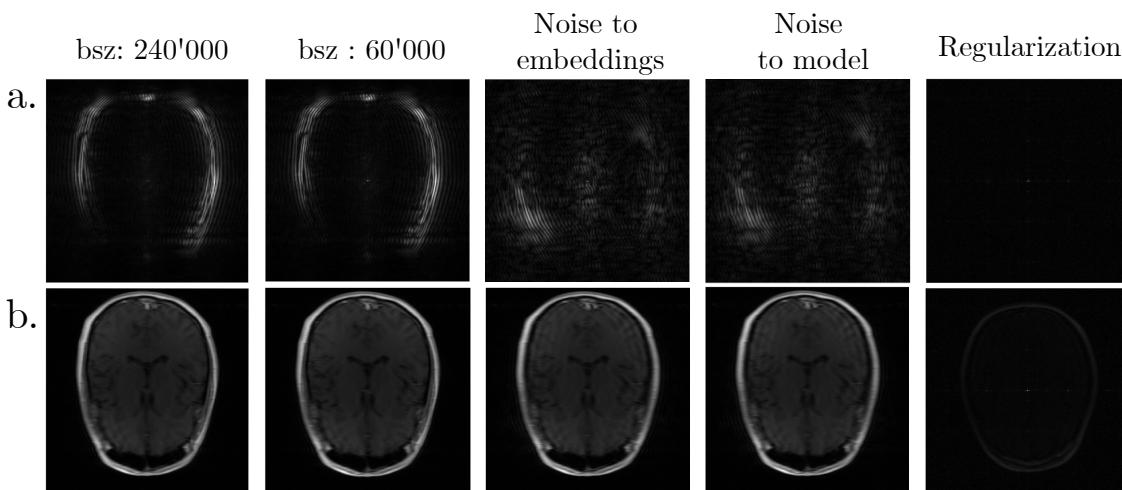


Figure 4.6: Results from third experiment, reconstruction of volume from training distribution. a. Model predictions b. Model predictions + k-space center

L2 regularization applied to the embeddings proved to be the least effective strategy. One possible explanation is that constraining the solution space simplifies the embeddings excessively, thereby preventing them from accurately capturing the volume and coil information.

In contrast, the stochastic approach yielded slightly better performance, although its predictions were somewhat blurred and the inverse transform over the k-space still exhibited artifacts. Notably, the experiment using the smallest batch size produced the most promising results, with the predicted volume showing almost no artifacts.

Overall, while increasing the embedding dimension appears to improve performance scores, these findings suggest that the current method for encoding volume and coil information is not ideally suited to the data. As a result, higher embedding dimensions may lead to a non-convex function space with too many local minima to be optimized effectively.



# Chapter 5

## Conclusion

This project explores the use of Implicit Neural Functions for MRI reconstruction and introduces several innovative methods to tackle the challenges of working in under-sampled k-space data regimes.

In the single-volume setting, initial experiments revealed a tendency for the model to overfit to k-space lines near the center. This issue arises from the high dynamic range of k-space, where central magnitudes are orders of magnitude larger than those at the periphery, biasing the model’s learning process.

To counteract this problem, we first evaluated the integration of the physics-inspired regularization method PISCO, which is based on the fact that neighboring k-space points share information coming from the multiple-coil sensitivity maps [8]. Although promising in concept, this approach led to significant instability and a considerable increase in training time, yielding worse results than those obtained with the baseline model.

An alternative strategy is also explored, where fixed positional encodings are replaced by learnable hash tables. This approach uses multi-resolution feature grids to better capture boundary regions of the predicted k-space. In fully sampled settings, hash encodings show marked improvements, presenting fewer artifacts and better metrics. Yet, when applied to under-sampled data, the benefit diminishes. The unbalanced update of feature vectors in un-sampled regions appears to undermine the model’s capacity to accurately learn boundary k-space information. This results in similar outputs to those obtained with baseline models. Future work should focus on exploring methods to update hash tables independently of under-sampling patterns to improve generalization to unseen regions.

The second part of the project focuses on the multi-volume training setup, where the MLP is jointly optimized together with latent embeddings that capture volume-specific information and coil sensitivity maps. A divergence between training and inference is observed.

In spite of the fact that training across different sample volumes and embedding dimensions converge to solutions with minimal artifacts and generally favorable scores; the models struggle to reconstruct high-quality images in under-sampled scenarios, even for volumes included in the training set. Despite testing various strategies (such as noise injection and regularization) to improve the optimization and guide the models to more optimal solutions, no substantial improvements are achieved.

These observations suggest that the model’s architecture is too simple to implicitly capture the inherent complexity of multi-volume data. Future efforts should explore more advanced architectures to encode volume and coil information (i.e. Variational Autoencoders, Transformers) adapting them to the complex nature of k-space.



# Bibliography

- [1] Tamaz Amiranashvili, David Lüdke, Hongwei Bran Li, Bjoern Menze, and Stefan Zachow. Learning shape reconstruction from sparse measurements with neural implicit functions. In *International Conference on Medical Imaging with Deep Learning*, pages 22–34. PMLR, 2022.
- [2] Robert S Balaban and TL Ceckler. Magnetization transfer contrast in magnetic resonance imaging. *Magnetic resonance quarterly*, 8(2):116–137, 1992.
- [3] Matt A Bernstein, Kevin F King, and Xiaohong Joe Zhou. *Handbook of MRI pulse sequences*. Elsevier, 2004.
- [4] Mark A Brown and Richard C Semelka. *MRI: basic principles and applications*. John Wiley & Sons, 2011.
- [5] Robert W Brown, Y-C Norman Cheng, E Mark Haacke, Michael R Thompson, and Ramesh Venkatesan. *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons, 2014.
- [6] Luis M De León-Rodríguez, André F Martins, Marco C Pinho, Neil M Rofsky, and A Dean Sherry. Basic mr relaxation mechanisms and contrast agent design. *Journal of Magnetic Resonance Imaging*, 42(3):545–565, 2015.
- [7] Anagha Deshmane, Vikas Gulani, Mark A Griswold, and Nicole Seiberlich. Parallel mr imaging. *Journal of Magnetic Resonance Imaging*, 36(1):55–72, 2012.
- [8] Mark A Griswold, Peter M Jakob, Robin M Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase. Generalized autocalibrating partially parallel acquisitions (grappa). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 47(6):1202–1210, 2002.
- [9] Wenqi Huang, Hongwei Bran Li, Jiazen Pan, Gastao Cruz, Daniel Rueckert, and Kerstin Hammernik. Neural implicit k-space for binning-free non-cartesian cardiac mr imaging. In *International Conference on Information Processing in Medical Imaging*, pages 548–560. Springer, 2023.
- [10] David J Larkman and Rita G Nunes. Parallel magnetic resonance imaging. *Physics in Medicine & Biology*, 52(7):R15, 2007.
- [11] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.
- [12] Donald W McRobbie, Elizabeth A Moore, Martin J Graves, and Martin R Prince. *MRI from Picture to Proton*. Cambridge university press, 2017.

- [13] Chrit TW Moonen, Peter CM Van Zijl, Joseph A Frank, Denis Le Bihan, and Edwin D Becker. Functional magnetic resonance imaging in medicine and physiology. *Science*, 250(4977):53–61, 1990.
- [14] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.
- [15] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.
- [16] Klaas P Pruessmann, Markus Weiger, Markus B Scheidegger, and Peter Boesiger. Sense: sensitivity encoding for fast mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 42(5):952–962, 1999.
- [17] Liyue Shen, John Pauly, and Lei Xing. Nerp: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):770–782, 2022.
- [18] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.
- [19] Veronika Spieker, Hannah Eichhorn, Wenqi Huang, Jonathan K Stelter, Tabita Catalan, Rickmer F Braren, Daniel Rueckert, Francisco Sahli Costabal, Kerstin Hammernik, Dimitrios C Karmpinos, et al. Pisco: Self-supervised k-space regularization for improved neural implicit k-space representations of dynamic mri. *arXiv preprint arXiv:2501.09403*, 2025.
- [20] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.
- [21] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018.
- [22] Zijun Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pages 1–2. Ieee, 2018.