

MSc ARTIFICIAL INTELLIGENCE  
MASTER THESIS

---

**Recurrent Inference Machines for  
accelerated 3D TOF-MRA  
reconstruction**

---

by  
JERKE JONAS VAN DEN BERG  
11034106

July 20, 2022

48 EC  
November 2021 - July 2022

*Supervisor:*  
Dr Matthan CAAN  
*Examiner:*  
Dr Xiantong ZHEN

*Second readers:*  
MSc Luka LIEBRAND  
MSc Dimitris KARKALOUSOS



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background and related work</b>	<b>3</b>
2.1	MRI physics . . . . .	3
2.2	The inverse problem . . . . .	4
2.3	ML advances . . . . .	5
<b>3</b>	<b>Methods</b>	<b>7</b>
3.1	Recurrent Inference Machine . . . . .	7
3.2	3D and 2.5D setup . . . . .	8
<b>4</b>	<b>Experiments</b>	<b>10</b>
4.1	Data . . . . .	10
4.1.1	3D T1 MRI . . . . .	10
4.1.2	Synthetic TOF-MRA . . . . .	10
4.1.3	Fully sampled TOF-MRA . . . . .	12
4.2	Undersampling . . . . .	12
4.2.1	2D Gaussian masking . . . . .	12
4.2.2	Fast Variable Density 2D Poisson . . . . .	12
4.3	Masking techniques comparison . . . . .	14
4.4	2.5D and 3D . . . . .	14
4.5	Synthetic 3D TOF-MRA reconstruction . . . . .	14
4.6	Fully-sampled 3D TOF-MRA reconstruction . . . . .	14
<b>5</b>	<b>Results</b>	<b>15</b>
5.1	Masking techniques comparison . . . . .	15
5.2	2.5D and 2D model comparison . . . . .	16
5.3	Synthetic 3D TOF-MRA reconstruction . . . . .	16
5.4	Out-of-training distribution 3D TOF-MRA reconstruction . . . . .	17
5.5	Synthetic-trained model on 3D T1 data . . . . .	17
<b>6</b>	<b>Discussion</b>	<b>22</b>
6.1	Conclusion . . . . .	23

## Abstract

Fully sampling Magnetic Resonance Imaging (MRI) scans is time consuming, which makes it inviable in the acute clinical setting. Techniques for reconstructing MRI with undersampled data have become clinically accepted, such as Compressed Sensing. However, they still often suffer from noise and aliasing. Recently, Machine Learning approaches in MRI reconstruction such as Recurrent Inference Machines (RIM) have significantly improved on this [10].

Time-Of-Flight Magnetic Resonance Angiography (TOF-MRA) is a specific way of acquiring an Angiogram using an MRI scanner, by relying on the flow of blood within vessels for a higher signal intensity. This thesis expands on applying the RIM to accelerated 3D TOF-MRA reconstruction. To this end, we use both synthetic 3D TOF-MRA data and non-synthetic 3D T1 MRI data for training the RIM model. These trained RIM models are tested on a fully sampled 3D TOF-MRA scan.

Reconstruction is trained and tested on retrospectively undersampling data by a 10x factor. A way of undersampling data that is not often used during training, called variable density Poisson sampling, is explored. This technique improves performance on reconstruction in all setups of the model tested, as well as for Compressed Sensing.

In an attempt to exploit the data consisting of slabs of consecutive slices, the model is expanded to handle 3D data and reconstruct in 3D with multiple consecutive slices at the time, however this did not increase model performance. Using the synthetic TOF-MRA dataset for training resulted in a model capable of reconstructing out-of-training distribution real TOF-MRA data, comparable to the 3D T1 trained model. The overall performance on the RIM and interchangability of MRI datatypes displays generalizability of the model.

# Chapter 1

## Introduction

Magnetic Resonance Imaging (MRI) is a type of medical imaging focusing on visualizing tissue within the patient's body. MRI data are acquired in  $k$ -space, the spatial frequency domain. When visualizing  $k$ -space on a grid, each point denotes phase and frequency of samples acquired during scanning.

Magnetic Resonance Angiography (MRA) is a specific technique of acquiring MRI data. The goal of MRA is to visualize the arteries and veins within a patient. For example, Phase Contrast (PC) MRA relies on injecting the patient's blood with a contrast enhancer to receive a higher intensity signal where bloodflow occurs. Time-Of-Flight MRA (TOF-MRA) does not require such enhancements and is viable in the acute setting as it requires the least screening and preparation. TOF-MRA relies on the inherent presence of continuous blood flow within the vessels for visualization. While 2D MRI techniques acquire slices sequentially, 3D MRI techniques acquire volumes at a time. In the next chapter, TOF-MRA, 3D acquisition and  $k$ -space will be explained in greater detail.

Fully-sampling MRI data, meaning acquiring all phase- and frequency samples needed for a clear image, is very time-consuming. For scanning a patient's brain this can easily take more than 30 minutes and will require the patient to lay still for the whole duration. This is not always feasible depending on the patient's condition and movement can cause artifacts in the acquired image. Being able to shorten the scan time reduces the risk of such movement-induced artifacts. Furthermore, if a patient suffers from a stroke in the acute setting, being able to perform and reconstruct a scan as fast as possible is essential. We can accelerate the scan by undersampling, saving time. However, this introduces aliasing and noise in the reconstruction. In this thesis, we aim at accelerating the acquisition and reconstruction of 3D TOF-MRA scans while retaining image quality. Over the past years, multiple techniques have been introduced to speed up the scanning process.

Parallel Imaging (PI) is one of the first advances in MRI to decrease scanning time. By using multiple receiver coils during scanning at the same time, we can combine images over multiple coils [14]. Incorporating sensitivity maps per coil allows for spatial localization, reducing phase-encoding steps. Coil sensitivity maps are calculated before imaging begins in a short, roughly 20 second acquisition. This has since been a staple technique in the clinical setting, and today most MRI data are acquired in the multi-coil setting. Compressed Sensing (CS) is a technique that has been widely researched for MRI and MRA, and can also be used in combination with PI. By relying on the sparsity of MRI in the image domain,  $k$ -space allows for significant undersampling in a sub-Nyquist ratio without aliasing [11]. In this setting, each coil scans under reduced field-of-view (FOV) which introduces folding, a reconstruction where the image overlaps itself. Using additional reconstruction steps in CS, this is reverted allowing for a coil-combined image reconstruction. The theory behind CS proves that images with a sparse representation can be fully recovered from randomly undersampled  $k$ -space data given a nonlinear recovery

scheme. However, CS in MRI introduces additional iterative steps in the reconstruction phase, slowing reconstruction down. This is a drawback for use in acute cases where acquiring an image as fast as possible is of utmost importance. However, the major problem with CS is that it cannot fully remove aliasing at higher acceleration factors. Ideally, we would like to speed up both the acquisition of a scan and the reconstruction of an image while retaining quality. By using Machine Learning (ML), we can further accelerate both MRI acquisition and reconstruction while reconstructing a high-quality image.

Accelerated MRI reconstruction has recently progressed with the help of ML algorithms [18]. With supervised learning in the context of MRI reconstruction, we use a fully sampled high-quality reconstruction, denoted as the Ground Truth, as our target for reconstruction. With unsupervised learning we don't explicitly match undersampled data with a target. The U-net is a supervised learning technique that has achieved great progress in MRI reconstruction [16] and some of the current state-of-the-art (SOTA) models rely on this technique [18]. With the introduction of large public datasets from the fastMRI consortium, results and effectiveness of different models have become easier to compare. However, research in applying these techniques to 3D TOF-MRA data remains to be published, or relies on unsupervised learning [3], [5].

Another ML technique catered to MRI reconstruction is the Recurrent Inference Machine (RIM), which in current published work has been applied in a 2D manner [10], [8]. This model allows for a recurrent approach to reconstruction without the explicit formulation of a prior, allowing the model to generalize better which is often a problem in RNNs. This is also one of the main pros for RIMs as they can be trained and tested on separate MRI datasets and perform well. That is why this paper focuses on applying the RIM to 3D TOF-MRA reconstruction.

We extend the RIM to a 3D model, as we expect this to be a benefit in reconstruction quality. By introducing 3D, we allow for multiple slices as in- and output at a time. An additional way of undersampling is experimented with, called variable density Poisson, to resemble undersampling of the data more closely to how MRI scanners sample [7]. To evaluate the performance of the RIM on 3D TOF-MRA data, we use two different datasets to train the RIM on. The first is a fully sampled 3D T1 dataset of the brain. T1-weighted MRI data are explained in the next chapter. The second dataset is a synthetically altered 3D TOF-MRA dataset, to simulate a 3D TOF-MRA dataset as the raw data was not able to be recovered. Both datasets are trained and validated on the 2D RIM as described and successfully tested in other work [8]. For testing, we use one fully sampled 3D TOF-MRA scan of the brain. By retrospectively undersampling data 10-fold, we simulate a 10x speedup factor.

As mentioned, the next chapter will explain some of the technical and physics-based details about MRI scanning, acquisition and reconstruction. Next, the method of research conducted will be described. The experiments chapter describes the specific setup for each experiment along with the setup of the various trained RIM models. The results contains both scores of reconstruction for comparison between models and figures of slice examples reconstructed during testing. Lastly, we will go over the results achieved and a conclusion on the research in the discussion.

# Chapter 2

## Background and related work

Before diving into recent developments in accelerated MRI and MRA reconstruction, we first need to understand the domain of the problem we are trying to solve. First we explain some of the basic physics concepts behind MRI and techniques used for scanning, before describing how (3D) TOF-MRA works and the clinical relevance of it to solidify the need for acceleration in this scanning process particularly. We then describe the forward model of accelerated MRI reconstruction. The inverse problem going from raw  $k$ -space data acquired on the scanner to the image domain (figure 1) is expanded upon for ML. Lastly we dive into the advances made by ML in accelerated MRI reconstruction before expanding on the RIM model in the next chapter.

### 2.1 MRI physics

MRI relies on the magnetic properties of tissue present within the body, and the properties of atomic nuclei. By creating a static magnetic field around the scanned patient, the protons present within align along the axis of this magnetic field. By sending an external Radio Frequency (RF) pulse through this field, the alignment of protons gets disrupted. Through a process called relaxation, the nuclei return to their aligned state in the static magnetic field. Through this process, the nuclei release RF energy of their own, which can be measured by the receiver coils of the MRI scanner. By varying the Time of Repetition (TR), the time between RF pulses, and Time of Echo (TE), the time in between sending the RF pulse and recording the RF energy emitted by nuclei, different relaxation processes are captured.

T1-weighted imaging captures the so-called T1 relaxation of nuclei. This visualizes predominantly fat tissue present within the FOV of the scan. This is done with a relative short TE and TR. Wherever T1 relaxation is present, T2 relaxation also happens (however not vice-versa). As a result, T2-weighted scans also highlight fat tissue, but additionally also return a higher intensity where water is present. A T2 scan is done by increasing both TR and TE times roughly 8-fold compared to a T1 scan.

To make an image sensitive to blood, we need a rapid TR, faster than a T1-weighted scan. The signal intensity of the (static) tissue will decrease, as the relaxation process will not have time to complete in between RF pulses. Meanwhile, because of the continuous blood flow in the scanned area, the signal intensity of the blood will remain high as the nuclei present in the blood will be fully relaxed, resulting in bright areas in the image where blood flow occurs. This concept is called Time-of-Flight (TOF) and is the main concept on which TOF-MRA relies on for creating an angiogram. After acquisition and reconstruction of the images from TOF-MRA acquisition is completed, Maximum Intensity Projection (MIP) is used to combine slices and get a 2D image of the blood vessels throughout the slab [17]. This MIP visualization is used by experts to diagnose a patient, e.g. by locating the blocked area within the vessel

structure in case of a stroke. The main advantage of doing a TOF-MRA over for example a Computer Tomography Angiogram (CTA) is that TOF-MRA does not require a contrast agent to be injected. Additionally, CTA has safety concerns that include allergic reactions, additional radiation dose and risk of contrast-induced nephropathy. [4].

The MRI scanner is equipped with gradient coils that can be used to alter the magnetic field in three axes. As explained earlier, the coils can be adjusted to produce a static field, constant over its space covered. Similarly, they can be tuned to produce a gradient, essentially producing a magnetic field that varies in strength depending on your position. Varying frequency with a gradient along an axis means that the signal's frequency components can be related to a location along this axis. This technique is known as frequency encoding. We can't apply frequency encoding in multiple directions, so we repeatedly apply and measure multiple gradients along each axis and measure phase differences. After acquisition, this can be resolved. This concept is called phase encoding. Together, phase and frequency encoding is what the scanner relies on for decoding location and intensity of samples. For 2D MRI acquisition, applying a gradient in the x and y axes is enough for recovering voxels. For 3D, additional phase encoding in the third dimension is used, to be able to acquire volumes of slices at a time, and still successfully decode into individual slices. TOF-MRA is a scan method that often uses this technique to acquire volumes of slices at a time. However, the FOV of each volume cannot be too big, or bloodflow will not occur fast enough in and out of view to make use of the TOF concept. This is why 3D TOF-MRA often gets scanned in smaller volumes, called slabs, that are combined during construction into one volume.

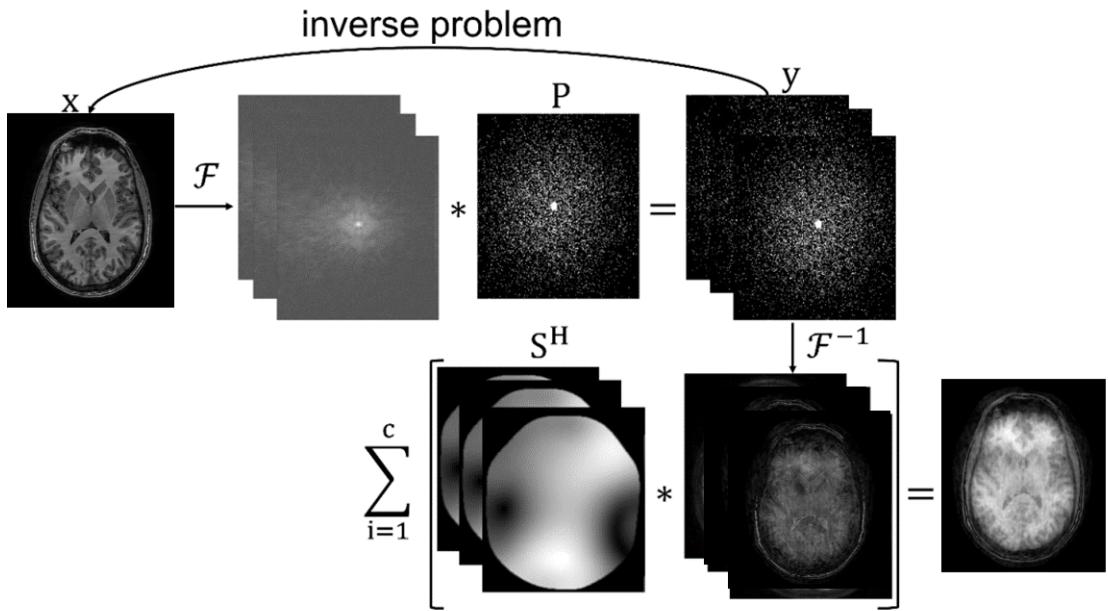


Figure 1: An overview of the inverse problem for accelerated MRI reconstruction using fully sampled data. Figure taken from [8].

## 2.2 The inverse problem

Let  $x$  be the true image we want to recover and  $y$  the undersampled  $k$ -space data we acquire from scanning, then the following equation defines the forward model:

$$y = \mathcal{F}Sx + n, \quad (1)$$

Where  $\mathcal{F}$  represents the Fourier transform from image space to  $k$ -space, and  $S$  denotes the set of coil sensitivity maps. Finally,  $n$  stands for additional noise picked up during scanning. By introducing an undersampling mask  $P$ , we can simulate the acceleration in scan acquisition by discarding a fraction of the samples in  $k$ -space:

$$y = P\mathcal{F}Sx + n, \quad (2)$$

This formula holds when we receive our data from a single receiver coil. Expanding for multi-coil data, the forward model is given by:

$$y = P\mathcal{F}S_\iota x_\iota + n_\iota, \quad \iota = 1, \dots, c. \quad (3)$$

Where  $S_\iota$  is the sensitivity map of coil  $\iota$  out for a total of  $c$  coils.

To reconstruct our desired image starting from  $k$ -space, we need to solve find the inverse transformation of the forward model of equation (3). Solving the inverse Fourier transform, going from undersampled  $k$ -space to image space will result in a noisy and low-quality reconstruction. We can use this as input for our model, by taking the fully-sampled reconstruction as our target and solving for a Maximum A Posteriori (MAP) estimation, defined as

$$x_{\text{MAP}} = \text{argmax}\{\log p(y|x) + \log p(x)\}. \quad (4)$$

Essentially, the MAP maximizes the sum of the log-likelihood of  $x$  and the log-likelihood of  $y$  given  $x$  as our prior. By introducing both the likelihood of  $x$  and  $y$  in our estimation, we solve for the most probable  $x$  while constraining its space to a probable image after reconstruction. For our multi-coil data setting, the MAP can be written as the optimization problem

$$x_{\text{MAP}} = \text{argmin}_x \left\{ \sum_{\iota=1}^c d(y_\iota, P\mathcal{F}S_\iota x_\iota) + \lambda R(x) \right\}, \quad (5)$$

introducing a data consistency function  $d$  between the measurement  $y$  and the reconstruction, and a regularizer  $R$  restricting the image domain in the form of a prior over  $x$ .

Finally, formulating a negative log-likelihood as our optimization problem, assuming gaussian-distributed data and leaving out the prior used in (5) we get

$$\log p(y|x) = \frac{1}{\sigma^2} \sum_{\iota=1}^c \|P\mathcal{F}S_\iota x_\iota - y_\iota\|. \quad (6)$$

representing our log-likelihood for optimization.

## 2.3 ML advances

Most recently, ML techniques have made great progress in accelerating MRI reconstruction. In this section, we will cover one of the most prominent approaches to solving reconstruction through ML, and current work for accelerated reconstruction in 3D TOF-MRA specifically.

The U-Net derives its name from the "U" shape it resembles when looking at the architecture [16]. The architecture consists of a series of convolutional layers, making the architecture similar to a Convolution Neural Network (CNN). However, by using Max pooling to decrease and Bilinear upsampling to increase dimensions, the convolutional layers can be applied to different resolutions of the input. By adding the output at each layer (from top to bottom) back in after each iteration of upscaling, detail in the reconstructed image can be recovered from the input even after max pooling. The idea behind this approach is that it is easier for the model to learn

and memorize the general structure of the desired (non-aliased) image in lower dimensionality, before iteratively adding more detail in higher dimensions during upscaling.

As mentioned, TOF-MRA data are often acquired in 3D, making this variant of MRI data a candidate for 3D reconstruction models. 3D models can however be taxing on memory when training models, as they require significantly more parameters compared to 2D approaches. In [3], a variant of the U-Net was proposed to reconstruct 3D TOF-MRA data. The authors of this paper solve this issue by linking two 2D models together handling the data in a separate view. The two models each consist of a cycleGAN, in which the 'generator' part of the model is essentially a U-net on retrospectively undersampled data (using masking).

As this model is implemented as a GAN, it uses unsupervised learning for training the model. This is a nice way of being able to use data when a fully-sampled target is not available to match undersampled data. However a concern with this is that false pathologies and hallucinations can occur in reconstructions when such target is missing. The model improves over CS, and the proposed model by the authors includes reconstruction scores with test data that includes a Ground Truth. However as not all visual comparisons include a Ground Truth, this remains to be a concern in this recent research.

One result by this paper compares reconstruction per number of consecutive slices fed to the model at once. The authors report the best scores when using five to seven slices at once. This is a relevant baseline for our expansion of the RIM to a 3D model.

# Chapter 3

## Methods

### 3.1 Recurrent Inference Machine

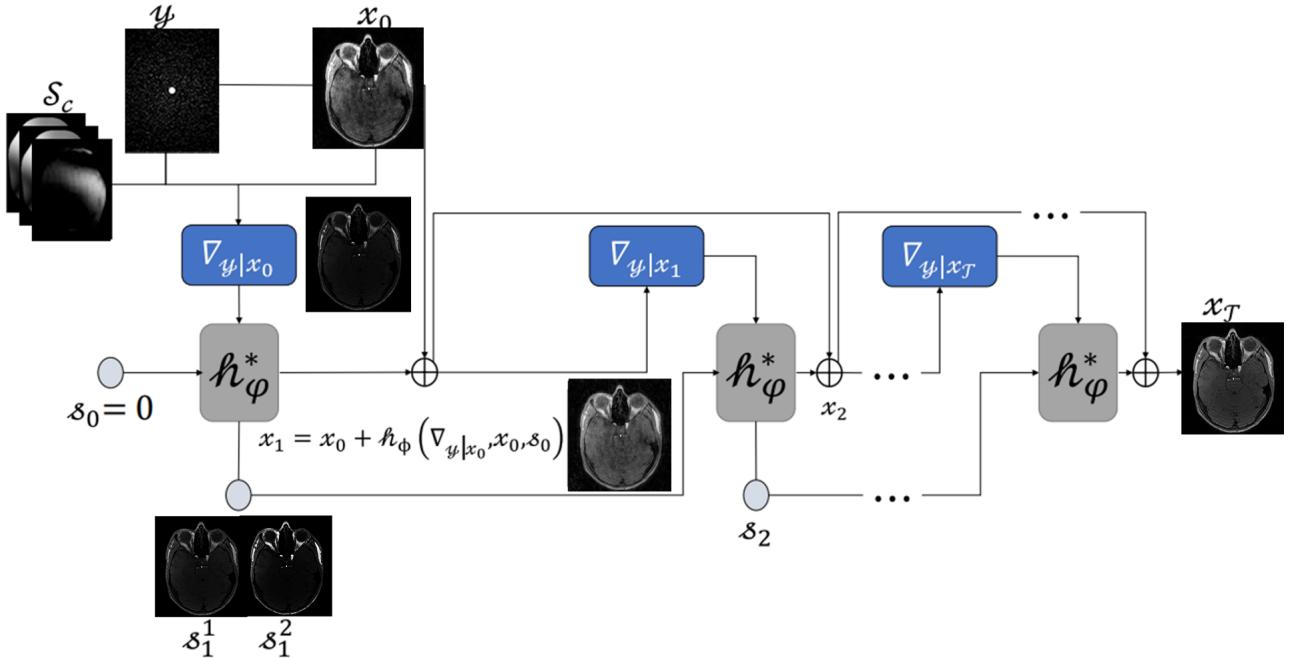


Figure 2: Unrolled time-steps of the RIM model by [8].

RIM models encapsulate, as the name implies, a Recurrent Neural Network (RNN) architecture. RIMs have been specifically designed to solve inverse problems. The RIM, as opposed to RNNs, does not need a prior and instead evaluates the gradient derived from equation 6:

$$\nabla_{y|x} = \sigma^{-2} \mathcal{F}^{-1} (P \mathcal{F} S x - y) \quad (7)$$

which implicitly defines the prior within the model parameters in the form of designated convolutional layers. The architecture of  $h_\varphi$  consists of a repeated sequence of layers, each sequence consisting of a convolutional layer and a recurrent layer. In this thesis, the IndRNN is used as the recurrent layer. A non-linearity is applied in between each convolutional and recurrent layer. The specific setup used here for  $h_\varphi$  can be found in the next section, along with its parameters used for experimentation. An overview of the model is visualized in Figure 2. Taking the current estimation  $x_t$  and the hidden state  $s_t$  as input at each time-step  $\tau \in T$ ,  $h_\varphi$  updates itself and the gradient of the log-likelihood  $\nabla_{y|x_t}$ . Through a sequence of alternating

convolutional and recurrent layers where  $T$  denotes the total number of time-steps, the state variable  $s$  is updated between each convolutional layer, maintaining a notion of memory over the time-steps. The update equations are given by:

$$s_0 = 0 , \quad x_0 = \sum_{t=1}^C S_t^H \mathcal{F}^{-1} P^T y_t , \\ s_{\tau+1} = h_\varphi^*(\nabla_{y|x_\tau}, x_\tau, s_\tau) , \quad x_{\tau+1} = x_\tau + h_\varphi(\nabla_{y|x_\tau}, x_\tau, s_{\tau+1}) . \quad (8)$$

in which  $C$  represents total coils,  $H$  the Hermitian complex conjugation and  $h_\varphi^*$  updates the intermediate hidden state  $s$  before  $x$  is updated by adding  $h_\varphi$ .

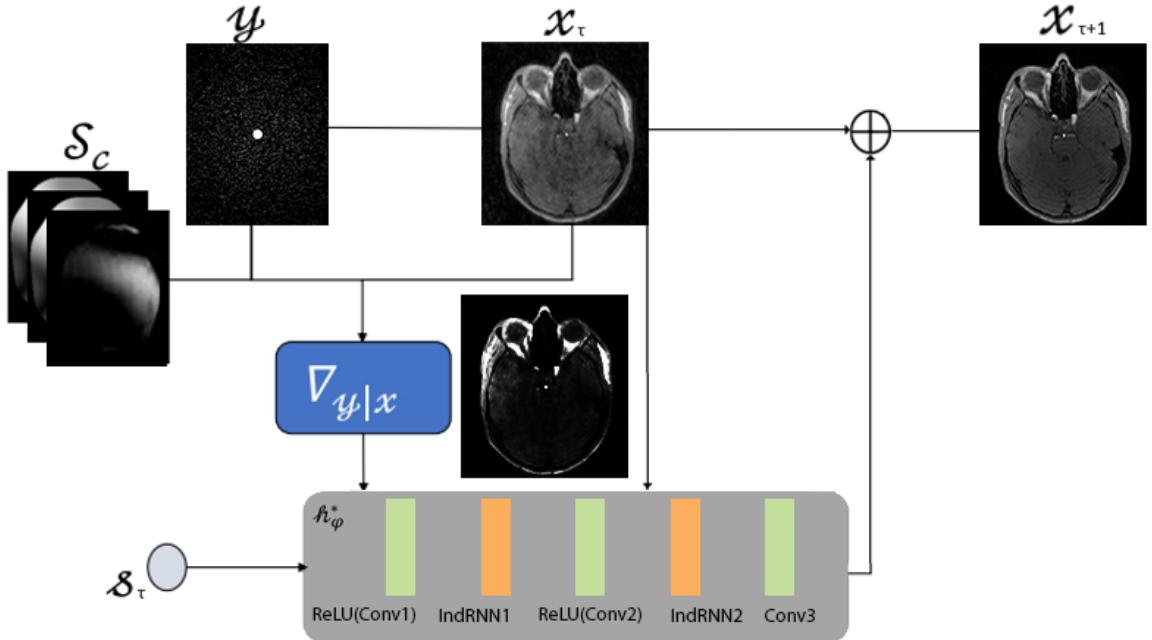


Figure 3: Overview of the RIM setup used for experiments and training.

For the two RNN layers in the model, an Independent RNN (IndRNN) is used [9]. The IndRNN layers rely on convolutions as well, for which the kernel size is set to 1. The benefit of this form of RNN is that it can be easily chained with other layers as the RIM does and still train robustly. Additionally, this implementation has shown to handle the common exploding/vanishing gradients problem in RNNs better. An overview of the RIM architecture used can be found in figure 3. The loss function used is the  $l1$ -loss, defined as the absolute difference between target and prediction:

$$\mathcal{L}^{l_1}(\hat{x}) = |\hat{x} - x| \quad (9)$$

### 3.2 3D and 2.5D setup

The RIM model has been extended to accept multiple slices at a time (and reconstruct them simultaneously). As opposed to a batch approach, where reconstructions and loss are calculated separately before updating the model averaging in gradients, this approach takes as input multiple slices at a time and handles it as one input, returning the same size as output. The loss is then calculated over this volume of slices before updating the model during training phase. The rate of consecutive slices fed to the model at a time has been kept at 5 to make training the model in terms of memory feasible.

Two setups are used for handling 3D data. In the 3D approach, we extend convolutional kernels in the depth direction, meaning 5x5 in our 2D approach becomes a cube of 5x5x5. The dilation parameter is used in the depth dimension as well. In 2.5D, we use the same convolutional kernel sizes as in 2D, but use two in succession, transposing the width to the depth dimension for the second convolution. In essence, this allows us to reduce the total number of parameters of the model, while retaining learning structures in two separate planes simultaneously.

# Chapter 4

## Experiments

### 4.1 Data

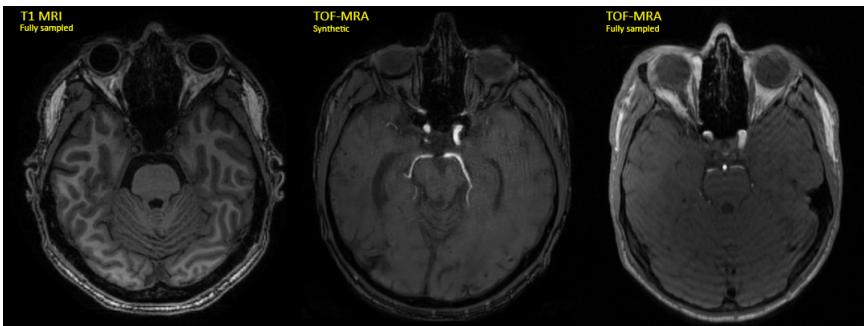


Figure 4: A random representative slice for each used dataset. The left and middle are used for training models, and the rightmost is an example slice of the out-of-training distribution test set.

In this thesis, two separate datasets were used for training, and one dataset was used for testing. Scanning was performed in the Amsterdam UMC. The first training dataset consists of twelve 3D T1 MRI brain scans, the second of six synthetic 3D TOF-MRA brain scans. The testing dataset consists of one fully-sampled 3D TOF-MRA brain scans. A representative slice of each dataset is shown in figure 4. Each dataset is explained in more detail in the next sections.

#### 4.1.1 3D T1 MRI

This dataset contains twelve 3D T1-weighted multi-coil brain scans on a 3T Philips Ingenia scanner in Amsterdam UMC. Since this dataset is the only MRI data in T1-weighted acquisition, we will denote this dataset as the 3D T1 dataset. As the data are acquired in 3D, it can be transposed to attain different viewpoints of the scan in 2D. Each scan is used for training in three views separately, namely coronal, sagittal and axial. Across the three different views, the average dimension of slices is 256x256, with a total of 280 slices per scan (70 in axial and 105 in coronal and sagittal view). The dataset is split in ten scans for training, one for validation and one for testing, for a total of 3360 slices.

#### 4.1.2 Synthetic TOF-MRA

This data were acquired through a 3T Philips Ingenia Elition scanner. The original DICOM reconstructions and coil sensitivity scans were sourced to Amsterdam UMC by the Rothschild

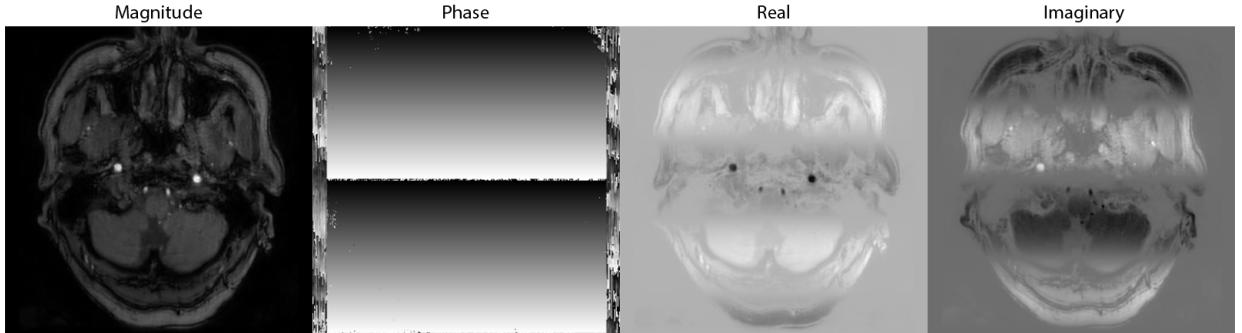


Figure 5: A slice of a zero-filled reconstruction performed on the synthetic TOF-MRA, visualized in magnitude and phase as well as the real and imaginary part of the complex data separately.

Hospital in Paris. The data was acquired through patients with acute neurological deficit, to diagnose a potential stroke within the brain. The data were IRB approved by the hospital for research.

As opposed to the 3D T1 dataset, for the 3D TOF-MRA dataset only have the DICOM files available, lacking the complex data format. These reconstructions are coil-combined and do not contain a phase. This data are synthetically altered into multi-coil data by multiplying the reconstructions with the scanners’ coil sensitivity maps, resulting in the data expanding from 1 to 32 in the coil dimension as turning into complex data by taking over the phase of said sensitivity maps.

To ensure that the magnitude image of the reconstruction is properly normalized through the sensitivity maps, we take the conjugate of the sensitivity map for constructing our Ground Truth. However, since our data have the same phase as the sensitivity map, taking the conjugate of it will negate the phase of the data and result in a reconstruction only containing the magnitude, meaning the imaginary part of our complex data contains nothing. This is problematic for a dataset used for training, as we need complex data for training. This is why for this dataset an extra synthetic phase gets added per volume. This synthetic phase is the same for every voxel in the width direction, and ranges from  $-\pi$  to  $\pi$  twice in the height direction of each volume.

Again, since this data are originally acquired in 3D, we can simply permute it to attain coronal, axial and sagittal views of each volume. For each plane, the synthetic phase gets added, along with additive noise. The noise is generated in the form of Gaussian noise  $\mathcal{N}(\mu = 0, \sigma = 0.01)$  as it was found by similar synthetic MRI ML research to be beneficial for training [12]. New noise is generated for both real and imaginary parts for each entry in the coil dimension and for each view of the scans separately. This noise then gets repeated along the slice dimension.

A total of six scans were used, split into five for training and one for validation. No data was split into training as the dataset is sparse (compared to the twelve scans present in 3D T1). Additionally, testing separately on a test set does not make a significant difference in terms of distribution for this set. A volume viewed axially consists of 140 slices of 384x384 voxels, each with 32 coils. Originally, this data are acquired in four slabs before combined by the scanner into the mentioned DICOM file reconstructions. Permuted into coronal and sagittal views, the first and last 60 slices are dropped as they contain noise and little to no structure of the anatomy of the subject. This results in a total of 4008 slices used, 3340 of which are used for training.

### 4.1.3 Fully sampled TOF-MRA

The out-of-training distribution fully-sampled TOF-MRA dataset was composed of a single healthy test subject, using a 3T Philips Ingenia scanner. The scans consist of four slabs of 66 slices along the axial dimension, each containing 432x396 voxels. To realize the effect of 3D reconstruction along different views, the models are tested by reconstructing only the axial plane.

## 4.2 Undersampling

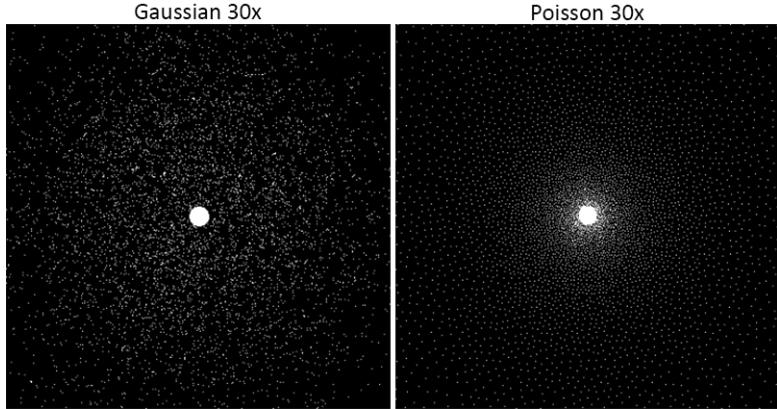


Figure 6: Visualization of the two ways of (2D) masking used for retrospectively undersampling, the Gaussian approach on the left and Variable Density Poisson on the right. Both examples result in an acceleration factor of 30x with a fully sampled center (included in the rate of undersampling).

As discussed in Chapter 2, we can undersample our data retrospectively through masking to simulate a speedup in the scanning process. In this thesis we mask in 2D, meaning we acquire 2D slices in  $k$ -space through a 2D inverse Fourier-transform along the slice dimension before masking. 2D masks are generated by the size of the height and width of the volume. We then expand along the slice and coil dimension. By setting the value of certain coordinates to zero within the mask, we discard data of this respective voxel, and by setting the value to one, we keep it. By generating a mask retaining one tenth of the samples, we simulate a speed up factor of ten. A new mask is generated for each new volume to combat overfitting to specific samples within the data.

### 4.2.1 2D Gaussian masking

2D Gaussian mask generation relies on a probability distribution for  $x$  and  $y$  separately. Probabilities are independent of surrounding or previously generated samples, and as such this algorithm does not rely on an iterative process making it faster than other implementations, like sampling on a Poisson distribution making it fast and practical for masking. Because of this and the easy handling and scaling of a higher distribution in the center, it is a popular choice for MRI reconstruction research.

### 4.2.2 Fast Variable Density 2D Poisson

A drawback to 2D Gaussian masking is that samples can be cluttered or be far in between, with no restrictions on minimum or maximum distance between samples. This results in a weaker

encapsulation of the spread phase and frequency encoding present in  $k$ -space. Additionally, undersampling in Philips scanners and Compressed SENSE reconstruction relies on variable density Poisson distribution [7]. To stay closer to the clinical way of undersampling a variable density Poisson distribution is also researched for masking.

Sampling with a variable density Poisson distribution relies on an iterative process for sample-selection, as the minimum and maximum distance is regulated between every subsequent sample generated. The variable density comes into play at this parameter, reducing the minimum and maximum distance between samples towards the center for a higher density in the center. In a naive implementation, new samples have to be checked against every existing sample to enforce this distance, exponentially increasing time of mask generation when lowering undersampling factors. When generating new masks during training of the models, this can drastically slow down training. However, using a smarter algorithm, mask generation can be sped up drastically [6]. This algorithm has been implemented in Python in the research shown here to accommodate the RIM. Unlike other implementations of this algorithm in Python like in the SigPy library [13], this one does not rely on C conversion to python, which caused errors with the RIM implementation. For details on the implementation refer to the GitHub directory of this thesis [2].

	ReLU(Conv1), ReLU(Conv2), Conv3		IndRNN1, IndRNN2	Total parameters		
Type	Filter Size	Kernel Size	Filter Size	2D	2.5D	3D
64F	64, 64, 2	5, 3, 3	64, 64	53000	127000	154000
96F	96, 96, 2	5, 5, 3	96, 64	257000	-	-

Table 4.1: Different parameter setups for the used RIM models. Column 1 contains filter sizes and kernel sizes for Conv1, Conv2 and Conv3 in that order. The second column only includes filter sizes for the IndRNNs as kernel size is kept to one for all setups. The last column includes total parameter count per setup. For 2D, Kernel size is repeated for both dimensions. 2.5D duplicates the 2D kernel and permutes the width-axis into the slice dimension for the last kernel. For 3D, kernel dimensions are repeated 3 times.

The RIM is trained under varying combinations of both filter and kernel sizes for both 2D, 2.5D and 3D approaches of reconstruction. The setup for each model used is gathered in table 4.1. Experiments are ran chronologically for a hyperparameter search. By doing so we limit the amount of experiments needed by carrying over optimal parameters from intermediate results. Finally, the best setups are compared on the fully-sampled 3D TOF-MRA data that is available, to limit this comparison to the best setups found. Performance is evaluated by measuring the Structural Similarity Index Measurement (SSIM) and the Peak Signal-to-Noise Ratio (PSNR).

Besides ML, CS is also used for reconstructing undersampled data as a base case. For this PICS is used for iterative reconstruction [1], for a total of 60 iterations. Additionally, undersampled data are Fourier-transformed to image space as-is and included in comparison, to show how data used before using any models. This is also known as a zero-filled reconstruction, which simultaneously represents the initial estimation of the RIM. Undersampling is done 10-fold, which has been feasible for previous research with applying the RIM on MRI data [8].

Models are trained for at least 20 epochs for each dataset separately, which is increased when the observed validation loss has not yet reached convergence. Furthermore, parameters that are kept constant from the original setup [8] are the dilation of one for the first Convolutional layer, two for the second and one for the last. The first two Convolutional and IndRNN layers additionally contain bias weights which are left out for the last Convolutional layer. For specific setup of normalization and handling data refer to the GitHub directory which includes the experimental setups for training [2]. For testing the effectiveness of the RIM on both different datasets and different masking techniques, the following experiments are carried out.

## 4.3 Masking techniques comparison

As a comparison to decide on the effectiveness of both undersampling techniques, Poisson and Gaussian, two identical 2D RIM models are trained with the same setup except for their masking algorithm used during training. The setup of the layers for these can be found in table 4.1 under 64F in 2D approach, resulting in 53000 parameters. To isolate the masking parameter here, training, validating and testing is done using the 3D T1 dataset. As the 2D model will have considerably less parameters, an additional more complex 2D model is trained on 3D T1 data to test the influence of more complex models in 2D (see 96F in table 4.1).

After this direct comparison between masking techniques, we decide on which mask performs better and use it for the rest of the experiments.

## 4.4 2.5D and 3D

For 2.5D and 3D convolutional approaches for the RIM, the following experiments are conducted. Because of the bigger input and output per batch of the 2.5D and 3D models compared to the 2D implementation, these models take both longer per epoch and need more epochs to train before stabilizing. Therefore, both a 2.5D and 3D model are trained for 20 epochs on the 3D T1 dataset, training on 5 consecutive slices per one input. The parameters of both models are in table 4.1 under 64F. After training both the 3D and 2.5D models for 20 epochs on the 3D T1 data, it became apparent that the 2.5D model converged better, optimizing loss faster. Compared to the 2D model, convergence of the 2.5D model is slower, so while the 2D models converged before 20 epochs, an additional 30 epochs were needed to get a stable loss for the 2.5D model. Because of this, the 2.5D was used for comparison to 2D and the model including full 3D convolutions is discarded from the results.

The 2.5D model is first trained and tested against the 2D model on the 3D T1 dataset. Results are documented for two setups: the same as training, where 5 consecutive slices get fed at the same time, and using the 2.5D model one slice at a time. This essentially renders the depth-wise convolution part of the 2.5D convolution useless. This is done to test the influence of feeding fractions of slabs directly and how it affects performance.

## 4.5 Synthetic 3D TOF-MRA reconstruction

After deciding on using the 2.5D RIM as opposed to 3D, we train both 2D and 2.5D models on the synthetic 3D TOF-MRA dataset. As this is TOF data, we also visualize MIP reconstructions in all 3 planes, as visual evaluation shows little of overall vessel structure when displayed per slice alone. This is also the way this data are used clinically.

## 4.6 Fully-sampled 3D TOF-MRA reconstruction

Narrowing down the models using 3D T1 and synthetic data with intermediate evaluation from results, we test the two best models, one trained on the 3D T1 dataset and one trained on synthetic 3D TOF-MRA dataset, we perform evaluation on out-of-training distribution data to test if synthetic TOF-MRA data are beneficial for training with the purpose of 3D TOF-MRA reconstruction. Again, MIPs are included, to compare overall vessel structure.

# Chapter 5

## Results

### 5.1 Masking techniques comparison

2D 64F RIM (Poisson)		2D 64F RIM (Gaussian)		PICS		zero-filled		
SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	
Poisson	<b>0.9685 ± 0.0162</b>	<b>37.50 ± 3.607</b>	0.9674 ± 0.0163	36.94 ± 3.439	0.8781 ± 0.0531	32.23 ± 3.330	0.7639 ± 0.1351	16.38 ± 1.654
Gaussian	0.5982 ± 0.3483	19.05 ± 3.381	0.9673 ± 0.0165	36.91 ± 3.435	0.8736 ± 0.0510	31.90 ± 2.951	0.7382 ± 0.1344	15.96 ± 1.856

Table 5.1: SSIM and PSNR values of 10x undersampled 3D T1 reconstructions using the test set, comparing the algorithms used and displayed in figure 7. The values are averaged over all slices.

Figure 7 shows a reconstructed representative slice from the 3D T1 test set. The 64F 2D RIM trained with Gaussian masking, the 64F 2D RIM trained with Poisson masking, PICS, and zero-filled methods are assessed.

Poisson masking results in higher SSIM and PSNR compared to the Gaussian masking (Table 5.1). Therefore, the rest of the experiments were performed using only Poisson masking for training, validation and testing.

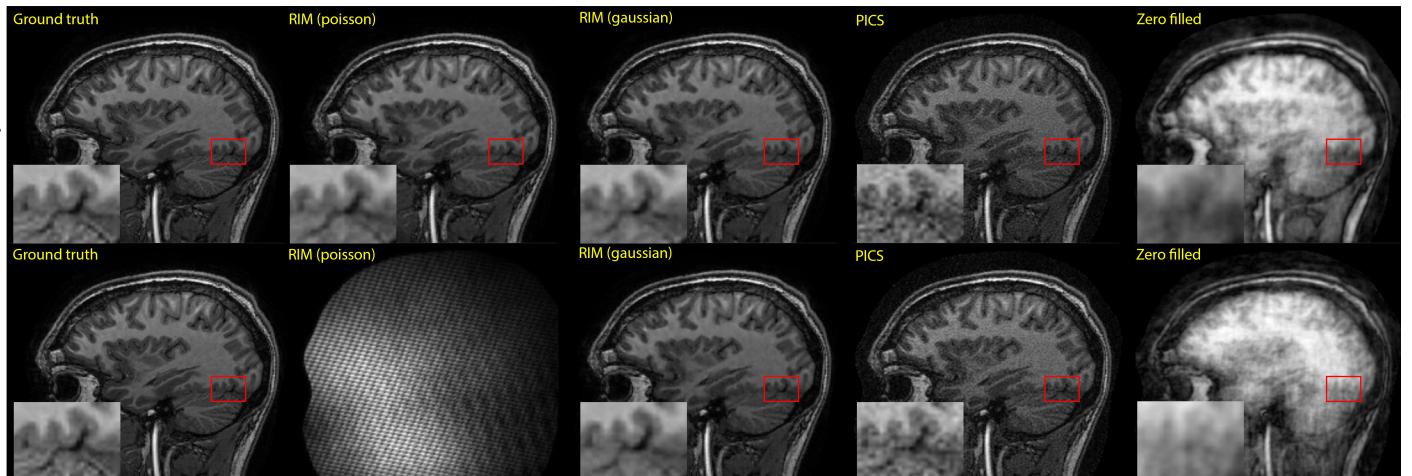


Figure 7: Comparison of 10x retrospectively undersampled 3D T1 sagittal slice reconstruction with the 64F 2D RIM. The leftmost column contains the ground truth, which is the target for reconstruction. The region marked in red is scaled up 4x in size and scaled in brightness for visual clarity. Row **A**: Inference tested on a Poisson masked slice. Row **B**: Inference tested on a Gaussian masked slice.

## 5.2 2.5D and 2D model comparison

	2D 96F RIM	2D 64F RIM	2.5D RIM (1 slice)	2.5D RIM (5 consec)
SSIM	$0.9677 \pm 0.0215$	<b><math>0.9685 \pm 0.0162</math></b>	$0.9124 \pm 0.0648$	$0.9073 \pm 0.0649$
PSNR	$36.20 \pm 6.300$	<b><math>37.50 \pm 3.607</math></b>	$31.26 \pm 3.904$	$30.18 \pm 1.768$

Table 5.2: SSIM and PSNR values of 10x undersampled 3D T1 reconstructions in the test set, comparing the 2D 96F RIM, the 2D 64F RIM, the 2.5D RIM feeding one slice, and the 2.5D RIM feeding five consecutive slices. The values are averaged over all slices. For the 2.5D, a reconstruction is made with 5 slices at the same time, just as it is trained, and with 1 slice, for comparing the direct impact of feeding 3D data.

Scores in SSIM and PSNR are reported in table 5.2 Visual results of reconstructions on a representative slice in the coronal view from the test set can be found in figure 8. A region at the edge of the cerebellum is zoomed in for comparison in detail between reconstructions.

As the 2D 96F model with more parameters performed worse than the 2D 64F model, the 96F version is dropped from further experimentation.

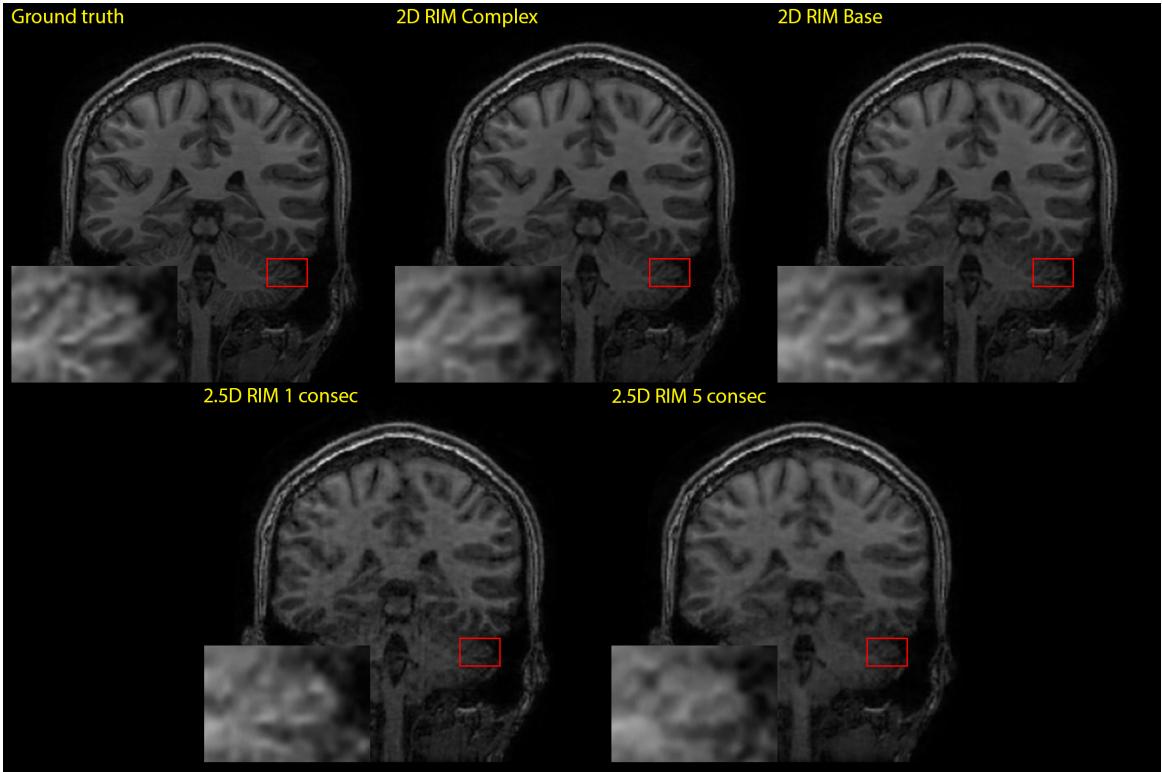


Figure 8: Comparison of reconstructing a 10x retrospectively undersampled slice for the 3D T1 test set reconstructed by the 2D 96F RIM (Complex), the 2D 64F RIM (Base), the 2.5D RIM feeding one slice, and the 2.5D RIM feeding five consecutive slices (denoted with 5 consec and 1 consec respectively). The inset within the red border is zoomed in four times and scaled up in brightness for comparison.

## 5.3 Synthetic 3D TOF-MRA reconstruction

In this section PICS and zero-filled reconstruction are again included for a baseline comparison. SSIM and PSNR scores are calculated in table 5.3. A representative slice of the validation set

	2D RIM (Base) Synth TOF	2D RIM (Base) 3D T1	2.5D RIM (1 slice) Synth TOF	2.5D RIM (5 consec) Synth TOF	PICS	zero-filled
SSIM	$0.9633 \pm 0.0516$	$0.5252 \pm 0.8076$	$0.8723 \pm 0.1089$	$0.7701 \pm 0.1445$	<b><math>0.9944 \pm 0.0081</math></b>	$0.6003 \pm 0.1995$
PSNR	$37.13 \pm 11.59$	$24.81 \pm 28.19$	$30.73 \pm 7.621$	$26.36 \pm 7.257$	<b><math>47.26 \pm 13.99</math></b>	$19.13 \pm 1.812$

Table 5.3: SSIM and PSNR values on Synthetic 3D TOF-MRA comparing the 2D and 2.5D 64F RIM models, PICS and zero-filled reconstructions. The values are averaged over all slices within the validation set.

is displayed in figure 9. Since this is angiogram data, a Maximum Intensity Projection (MIP) is included for visual comparison in the figure as well.

In figure 10 the effects of 2D and 2.5D convolutions is included.

## 5.4 Out-of-training distribution 3D TOF-MRA reconstruction

	2D RIM (Base) Synth TOF	2D RIM (Base) 3D T1	PICS	zero-filled
SSIM	$0.9213 \pm 0.0338$	<b><math>0.9347 \pm 0.0327</math></b>	$0.7107 \pm 0.0803$	$0.6619 \pm 0.0339$
PSNR	$36.81 \pm 2.548$	<b><math>37.45 \pm 2.847</math></b>	$29.24 \pm 3.117$	$19.53 \pm 3.574$

Table 5.4: SSIM and PSNR values of out-of-training distribution 3D TOF-MRA, comparing the two trained 2D 64F RIM models, PICS and zero-filled reconstructions. The values are averaged over all slices within the scan.

Since the 2D models still score the highest in all setups so far, the real, in-vivo 3D TOF-MRA scan is tested on two setups of the 2D RIM with retrospective undersampling. Both use the 64F setup that has been scoring the highest, with the difference being the training data. One is trained on real, 3D T1 MRI data, while the other uses synthetic, 3D TOF-MRA data. As with the results on the synthetic data, we document a visual example of a representative reconstructed slice present within the scan, as well as MIP reconstructions for a visual comparison in vessel structure in figure 11. The SSIM and PSNR values are again calculated and averaged per slice in figure 11.

## 5.5 Synthetic-trained model on 3D T1 data

Since testing on the 3D T1 dataset was performed before introducing results with using the synthetic TOF-MRA dataset as training data, we include the results of using the 2D 64F RIM model trained on synthetic 3D TOF-MRA data tested on 3D T1 data as a final result. The same slice is used as in section 5.2 in figure 12 for visual comparison, as well as an error map of the reconstruction compared to the ground truth. The SSIM and PSNR scores are documented in the same figure.

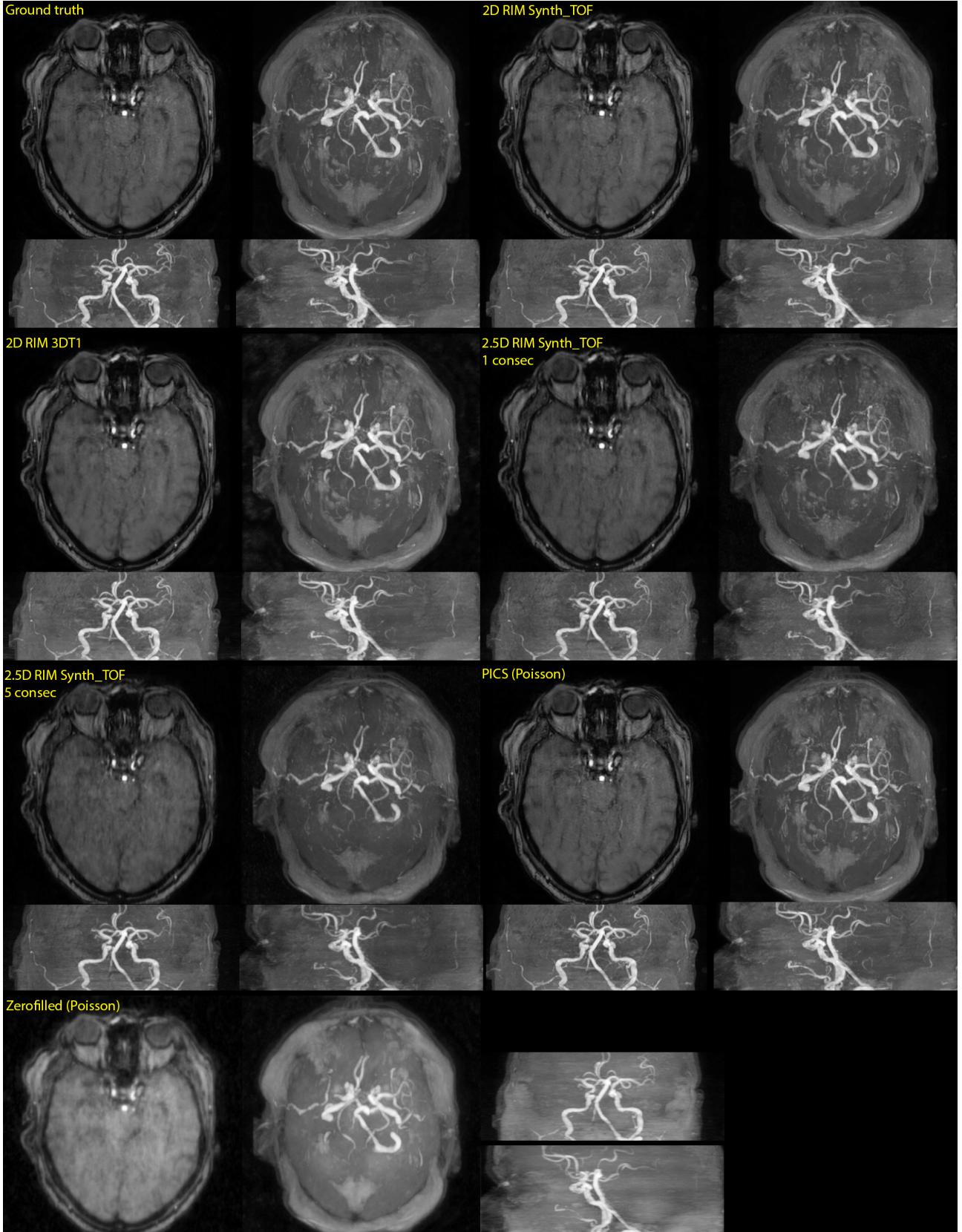


Figure 9: Slice- and MIP comparison between 2D and 2.5D 64F RIM models, PICS and zero-filled reconstructions on the synthetic 3D TOF-MRA validation data. The data are 10x retrospectively undersampled. The MIP is taken over all reconstructed axial slices. For the coronal and sagittal MIPs (bottom left and bottom right per reconstruction respectively), only the reconstruction in axial view is used to capture the impact of 2D reconstructions from different viewpoints in the slice dimension. For the 2.5D RIM, a reconstruction is made with 5 slices at the same time, just as it is trained, and with 1 slice, for comparing the direct impact of feeding 3D data. This is denoted with 5 consec and 1 consec respectively.

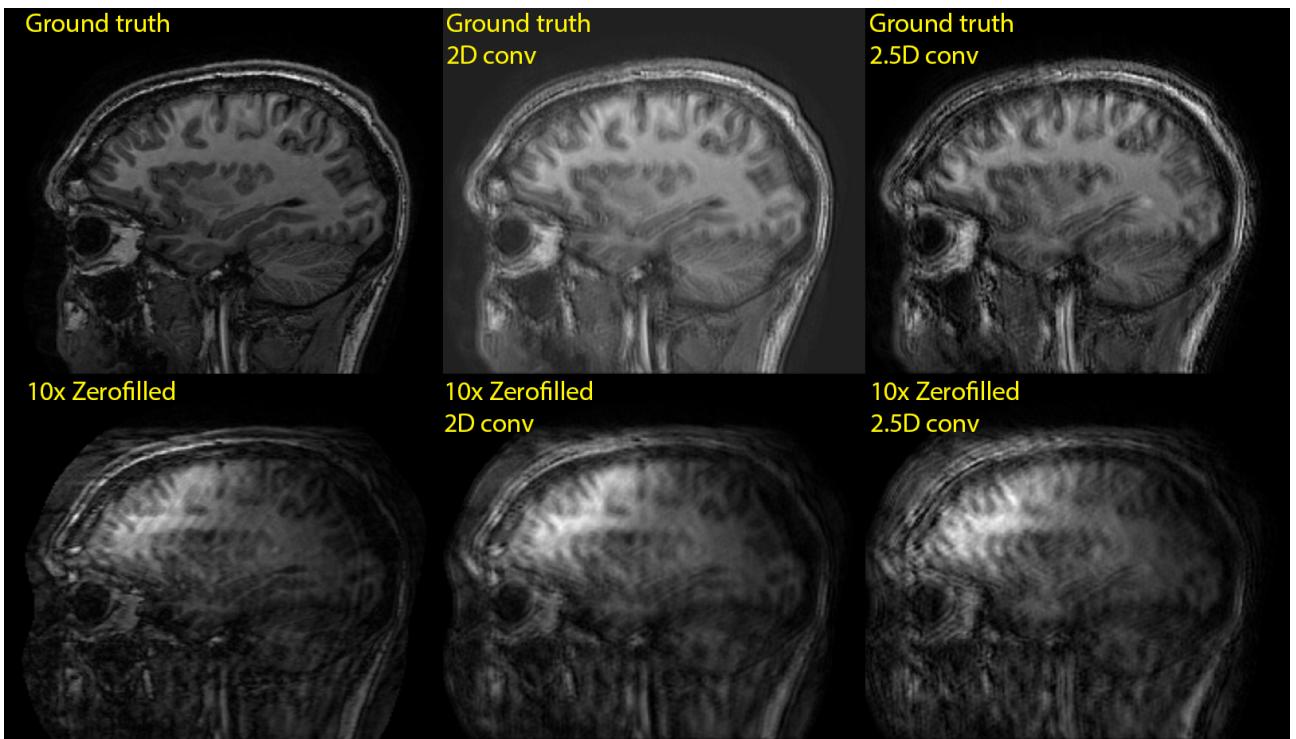


Figure 10: Visualization of the effect of 2.5D convolutions using consecutive slices. For the 2D convolution, a random-weighted  $5 \times 5$  convolution is used in the 2D slice-plane. For 2.5D, 4 extra consecutive slices after the present slice are stacked and convoluted in the same way, and another randomly weighted  $5 \times 5$  convolution used, however this time permuted into the slice-and height dimension. In the top row, we can see the effects of 2D and 2.5D convolutions on the ground truth slice, while in the bottom row, the slices are first 10x undersampled, to represent the effect of artifacts present across slices before convoluting over multiple slices, as happens in the 2.5D RIM.

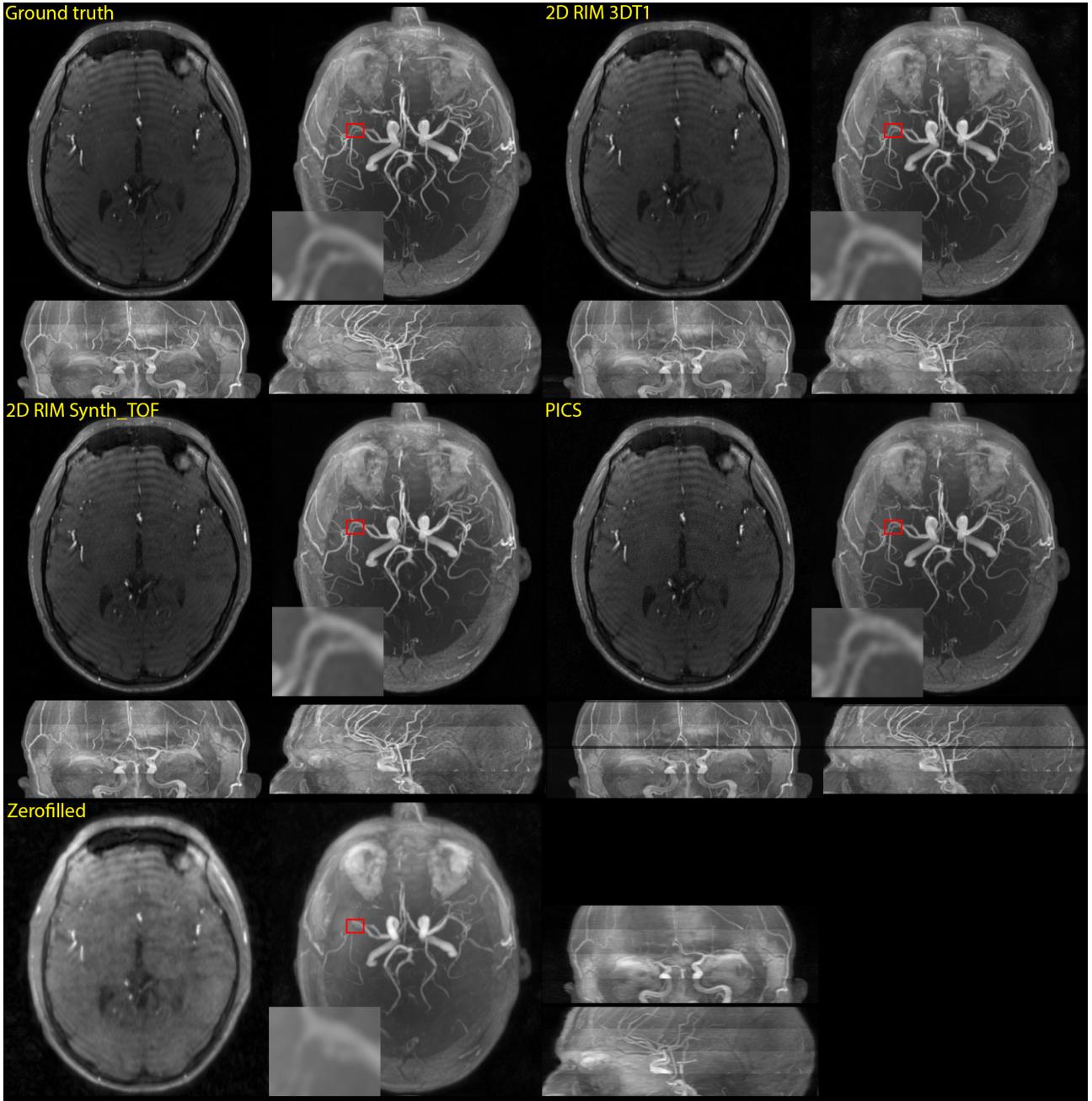


Figure 11: Slice and MIP comparison between the two trained 2D 64F RIM models, PICS and zero-filled reconstructions on the fully-sampled 3D TOF-MRA dataset. The Axial MIP is taken over all slices reconstructed. The inset within the red border is scaled x4 and scaled up in brightness for comparison. For the coronal and sagittal MIPs, only the reconstruction in axial view is used to display impact of 2D reconstructions from different viewpoints. For the 2.5D RIM, a reconstruction is made with 5 slices at the same time, just as it is trained, and with 1 slice, for comparing the direct impact of feeding 3D data. This is denoted with 5 consec and 1 consec respectively.

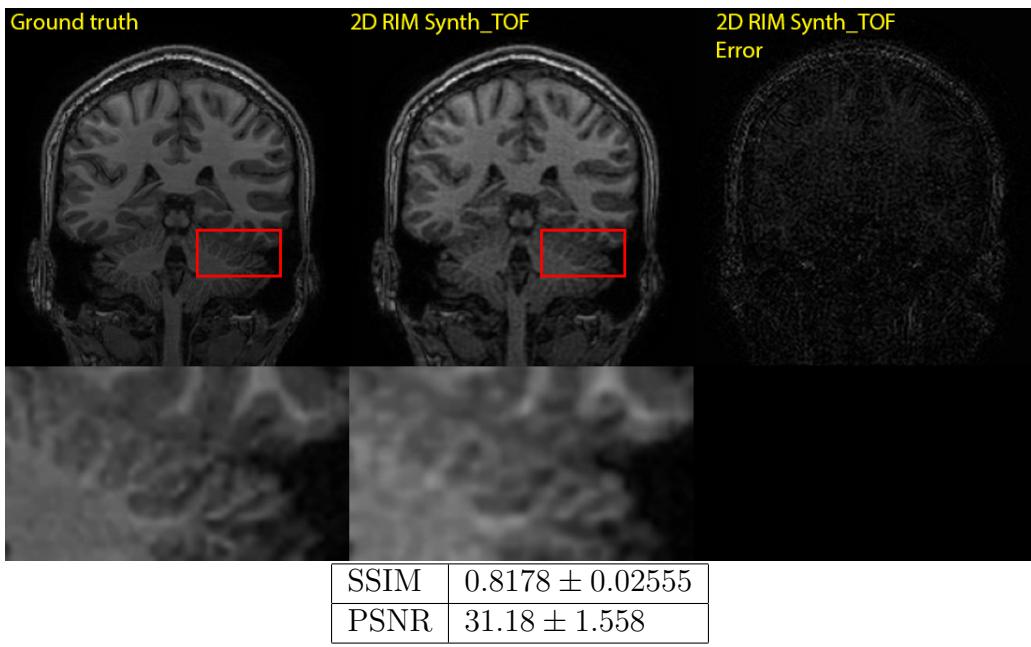


Figure 12: Reconstruction of a 10x undersampled slice from the 3D T1 test set with the 2D 64F RIM trained on the synthetic 3D TOF-MRA dataset, with the target on the left and the error map on the right. The error is calculated by taking the absolute of subtracting the reconstruction from the ground truth. The table represents SSIM and PSNR values calculated and averaged over the 3D T1 test set.

# Chapter 6

## Discussion

This thesis aims to progress accelerated 3D TOF-MRA reconstruction using RIM models. We explored how undersampling schemes and training data affects performance of the models. Additionally, we investigated 2D, 2.5D, and 3D approaches for implementing the RIM. As can be seen in figure 7 and the inset containing the edges between the cerebellum, temporal and occipital lobe, the models trained with variable density Poisson undersampling retain the most detail compared to Gaussian while appearing less grainy compared to the other reconstruction techniques. This idea is confirmed when comparing the SSIM and PSNR values of the reconstruction algorithms in table 5.1, where undersampling this way scores better for all set-ups, even when the 2D 64F RIM is trained on Gaussian undersampling and tested on variable density Poisson.

One odd result from using variable density Poisson undersampling is that trained models on this undersampling technique failed to reconstruct Gaussian undersampled data as shown in 7. The reason behind this could be that Poisson sampling ensuring a specific distribution in undersampling with minimum and maximum distances between samples. Gaussian does not have this, which could be why training on Gaussian does not break on Poisson undersampled data, but vice versa it does. Still, since undersampling done through scanners is often also a variation on variable density Poisson, including Philips scanners [7], this does not pose an issue in the clinical setting. Additionally, since the mask generation is fast enough to generate new masks for every batch using a fast algorithm [6], it also does not limit training and validation inference times for ML.

To explore the effects of convolutional layers to the RIM, 2.5D and a 3D convolutions have been compared to the 2D approach. This appeared not to be beneficial for reconstruction performance, on neither 3D TOF-MRA nor 3D T1 MRI data. This can be attributed to how the RIM uses convolutions for constructing the initial estimation. By taking undersampled data that contain artifacts, and applying convolutions across the slice dimension, artifacts are shared between slices worsening the initial estimation. This effect is visualized in figure 10, where the difference between undersampled data before and after 2.5D convolutions is significantly worse compared to the same convolution process on fully sampled data. However, this does not mean that 2.5D convolutions would not be able to improve the model after aliasing is able to be solved first. As introduction of more artifacts is intrinsically harder to solve for the model than blurriness introduced by only using 2D convolutions, this is a design flaw in the setup of the 2.5D and 3D RIM. This line of argumentation is supported by the fact that in both synthetic 3D TOF-MRA and 3D T1 reconstructions, taking 1 slice at a time performed better than 5 slices for the 2.5D model, even while training occurred purely on 5 slices as input. In [15], the authors explore the invertible RIM, a variation of the RIM model. This model also shows that using a 3D model is not beneficial for reconstruction, aligning with our findings.

Interestingly, PICS reconstructions were almost flawless on the synthetic dataset when un-

dersampling with a 10x factor as documented in table 5.3. This could be due to the synthetic phase added to the data. The typical grainy reconstruction that PICS has does occur on the non-synthetic TOF-MRA data which contains a real phase. With the phase of the input data stemming solely from the sensitivity map and a relatively simple gradient scale added, PICS might have an easier time dealing with this than a trained ML model.

One of the reasons for AI research on MRI reconstruction being scarce is that data are not easily acquired by any researcher. Acquiring own data would of course require access to an MRI scanner. MRI data are protected by privacy laws for patients as any medical data are, and so first requires permission by patients/test subjects to be shared. Manufacturers of MRI scanners did not design the user interface for these scanners to be easy for researching raw data, making some parameters a challenge to resurface for preprocessing and aligning data with present separate real and imaginary components. The fastMRI dataset [18] aims to resolve some of these issues around researching MRI data, by being a publicly available and complete dataset of MRI data from both brain and knee scans. Even so, this dataset exceeds 1.3 terabytes, making storing and sharing datasets harder.

As a result, it can become even trickier to not only acquire this type of data, but acquire it as raw  $k$ -space data, across multiple coils, with an aligned sensitivity map, also containing the appropriate phase and magnitude. Using the steps documented for preprocessing the synthetic data in section 4.1.2, we have shown that when we lack raw multi-coil data, we can still reverse-engineer data that is viable for training, using only the sensitivity map and the magnitude of the target image, and even reconstruct undersampled real multi-coil data as shown in figure 11. This is supported by the results documented in table 5.4 for the synthetically trained RIM, getting close to the non-synthetic 3D T1 data trained RIM in performance on real 3D-TOF MRA data. Looking at the MIP reconstructions, there is no clear winner in terms of vessel clarity, as it depends on what region and vessel you look at. The 64F 2D RIM model, trained on non-synthetic 3D T1 data still performs best, even when tested on TOF data, in terms of SSIM and PSNR values on the slices recorded in table 5.4. This speaks to the generalizability of the RIM, which is performing well across multiple training setups, even when the training and test set is weighted and acquired differently. This is essential for the use of ML models in the clinical setting, as accelerated reconstruction algorithms must be robust and versatile to this end.

## 6.1 Conclusion

Variable density Poisson masking significantly improves results on both training and testing undersampled reconstruction. This holds not just for trained models like the RIM, but for algorithms like Compressed Sensing and simple zero-filled reconstructions as well. It is a relatively simple to implement improvement on the model, and with the shown general improvement could prove to be beneficial for training other accelerated MRI reconstruction models. Synthetic data proves to be useful in training the RIM for MRI reconstruction, as long as the properties of complex data are accounted for. This is a valuable property as MRI data can be scarce, hard to distribute, and memory-demanding. Though the approach used here for training the RIM with 3D data did not result in a fruitful model with the intent of accelerating 3D TOF-MRA data, it plants new ideas for exploiting the 3D property of different 3D MRI acquisition techniques in training the RIM. As described in [8], the RIM can be expanded using Cascades of Independently Recurrent Inference Machines (CIRIM). By applying reconstructions in 3D reconstructing one slice per cascade, we can remove the inherent drawback of 3D convolutions while benefitting from the generalizability of the RIM and potentially exploiting the property of 3D-acquired data that 3D T1 and 3D TOF-MRA data has. We leave this line of experimentation for future research.

# Bibliography

- [1] Bart toolbox for computational magnetic resonance imaging. 10.5281/zenodo.592960.
- [2] Thesis GitHub repository. <https://github.com/jerke123/mridc>.
- [3] Hyungjin Chung, Eunju Cha, Leonard Sunwoo, and Jong Chul Ye. Two-stage deep learning for accelerated 3d time-of-flight mra without matched training data. *Medical Image Analysis*, 71:102047, 2021.
- [4] Andrew M Demchuk, Bijoy K Menon, and Mayank Goyal. Comparing vessel imaging: noncontrast computed tomography/computed tomographic angiography should be the new minimum standard in acute disabling stroke. *Stroke*, 47(1):273–281, 2016.
- [5] J Ding, Y Duan, Z Zhuo, Y Yuan, G Zhang, Q Song, B Gao, B Zhang, M Wang, L Yang, et al. Acceleration of brain tof-mra with compressed sensitivity encoding: A multicenter clinical study. *American Journal of Neuroradiology*, 42(7):1208–1215, 2021.
- [6] Nicholas Dwork, Corey A Baron, Ethan MI Johnson, Daniel O’Connor, John M Pauly, and Peder EZ Larson. Fast variable density poisson-disc sample generation with directional variation for compressed sensing in mri. *Magnetic Resonance Imaging*, 77:186–193, 2021.
- [7] L Geerts-Ossevoort, E de Weerdt, A Duijndam, G van IJperen, H Peeters, M Doneva, M Nijenhuis, and A Huang. Compressed sense. speed done right. every time. philips healthcare, netherlands, 2018, 2020.
- [8] Dimitrios Karkalousos, Samantha Noteboom, Hanneke E Hulst, Frans M Vos, and Matthan WA Caan. Assessment of data consistency through cascades of independently recurrent inference machines for fast and robust accelerated mri reconstruction. *Physics in Medicine & Biology*, 2022.
- [9] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5457–5466, 2018.
- [10] Kai Lønning, Patrick Putzky, Matthan WA Caan, and Max Welling. Recurrent inference machines for accelerated mri reconstruction. 2018.
- [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] Elisa Moya-Sáez, Óscar Peña-Nogales, Rodrigo de Luis-García, and Carlos Alberola-López. A deep learning approach for synthetic mri based on two routine sequences and training with synthetic data. *Computer Methods and Programs in Biomedicine*, 210:106371, 2021.

- [13] Frank Ong and Michael Lustig. Sigpy: a python package for high performance iterative reconstruction. In *Proceedings of the ISMRM 27th Annual Meeting, Montreal, Quebec, Canada*, volume 4819, 2019.
- [14] 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.
- [15] Patrick Putzky and Max Welling. Invert to learn to invert. *Advances in neural information processing systems*, 32, 2019.
- [16] Nahian Siddique, Sidiqe Paheding, Colin P Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *Ieee Access*, 9:82031–82057, 2021.
- [17] Andrew J Wheaton and Mitsue Miyazaki. Non-contrast enhanced mr angiography: physical principles. *Journal of magnetic resonance imaging*, 36(2):286–304, 2012.
- [18] 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.