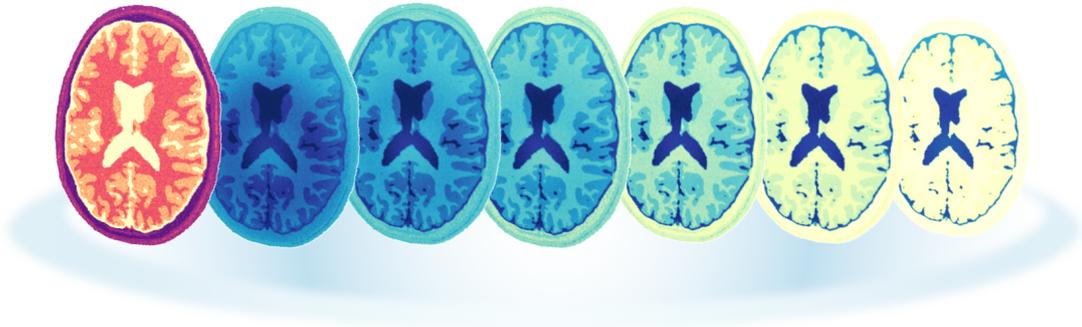




**TÉCNICO**  
LISBOA



## **Quantitative MRI Parameter Mapping with Extended Phase Graphs and Recurrent Inference Machines**

**Catarina Neves de Carvalho**

Thesis to obtain the Master of Science Degree in

**Biomedical Engineering**

Supervisor(s): Prof. Rita Homem de Gouveia Costanzo Nunes  
Prof. Teresa Margarida Matias Correia

### **Examination Committee**

Chairperson: Prof. Susana de Almeida Mendes Vinga Martins  
Supervisor: Prof. Rita Homem de Gouveia Costanzo Nunes  
Member of the Committee: Prof. Mário Alexandre Teles de Figueiredo

**30 November 2022**



**Declaration**

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

**Preface**

The work presented in this thesis was performed at the Institute of Systems and Robotics of Instituto superior Técnico (Lisbon, Portugal), during the period March-August 2020, under the supervision of Prof. Rita Nunes.

## **Acknowledgments**

I would like to express my gratitude and appreciation to my supervisors, for the invaluable guidance and feedback throughout this project. With my thanks to my friends and family, for their unwavering support.



## Resumo

A Ressonância Magnética Quantitativa é uma técnica de imagiologia que permite a quantificação de propriedades inerentes aos tecidos que frequentemente implica longos tempos de exame, limitando a utilização em contexto clínico. Um novo grupo de métodos de aprendizagem profunda, nomeados redes de aprendizagem profunda baseadas em modelos físicos, incorpora o modelo do sinal de ressonância no processo de aprendizagem para estimar mapas paramétricos dos tecidos, aliviando a necessidade de ter acesso a um grande número de dados de treino e acelerando o tempo necessário para obter os mapas. Contudo, a maioria destas redes adota uma função exponencial para modelar o decaimento do sinal, desprezando a contribuição de ecos estimulados e indiretos, ou heterogeneidades do campo  $B_1$ . Neste trabalho, sugere-se adaptar um desses modelos, as Máquinas de Inferência Recorrente (RIM), para mapear o  $T_2$  do cérebro com um modelo mais preciso, baseado em Grafos de Fase Estendidos (EPG), incluindo a influência do campo  $B_1$ . É também proposto um novo método, Máquinas de Inferência Recorrente com dicionários do modelo e seus gradientes (RIMFoGraD), que preserva a configuração original das RIM, mas acelera o processo de inferência através de dicionários de curvas de eco-modulação pré-calculadas para uma variedade de parâmetros. A RIMFoGraD estima mapas de  $T_2$  com  $320 \times 320$  pixels 380% mais rápido do que a RIM aliada ao modelo EPG sem perda de precisão, e 80% mais rápido do que um método de reconhecimento de padrões com precisão de 1 ms em  $T_2$ , com um erro mediano de 2.83 ms no parênquima cerebral.

**Palavras-chave:** Ressonância Magnética, Relaxometria, Mapeamento de  $T_2$ , Grafos de Fase Estendidos, Aprendizagem Profunda, Máquinas de Inferência Recorrente



## Abstract

Quantitative Magnetic Resonance Imaging is an imaging technique that allows the quantitative assessment of inherent tissue properties which often suffers from long scan times, preventing its use in routine clinical evaluation. A new class of deep learning frameworks, called model-based deep learning nets, incorporate the magnetic resonance signal model into the learning process to estimate parametric maps of the tissues, alleviating the need for a large number of training datasets and further pushing acceleration rates. However, most of the nets previously used in this context have employed a pure exponential curve to model the magnetic resonance signal, which does not account for stimulated or indirect echoes, or inhomogeneity of the  $B_1$  field. In this work, one of such models, the Recurrent Inference Machine (RIM), is adapted to perform  $T_2$  mapping with a more accurate signal model based on the Extended Phase Graphs (EPG) concept, while also considering the influence of the effective  $B_1$  field. Additionally, Recurrent Inference Machines with forward model and gradient dictionaries (RIMFoGraD) are proposed, which preserve the configuration of the original RIM, but speed up the inference process through dictionaries of pre-calculated echo-modulation curves and their gradients for a large range of parameters. RIMFoGraD was able to estimate  $T_2$   $320 \times 320$  pixel maps 380 times faster than the RIM implemented with the EPG model with no loss in accuracy, and was 80% faster than a pattern-based recognition approach with 1 ms  $T_2$  precision, with a median  $T_2$  difference of 2.83 ms from this method on the brain parenchyma.

**Keywords:** Magnetic Resonance Imaging, Relaxometry,  $T_2$  mapping, Extended Phase Graphs, Deep Learning, Recurrent Inference Machines



# Contents

Acknowledgments . . . . .	v
Resumo . . . . .	vii
Abstract . . . . .	ix
List of Tables . . . . .	xiii
List of Figures . . . . .	xv
Nomenclature . . . . .	xix
Glossary . . . . .	xxi
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Topic Overview . . . . .	2
1.3 Objectives and Contributions . . . . .	3
1.4 Thesis Outline . . . . .	4
<b>2 Background</b>	<b>5</b>
2.1 The Magnetic Resonance Signal . . . . .	5
2.1.1 The Extended Phase Graph Representation . . . . .	6
2.2 Deep Learning . . . . .	15
2.2.1 Recurrent Neural Networks and Gated Recurrent Units . . . . .	15
2.2.2 Recurrent Inference Machines . . . . .	15
<b>3 Implementation</b>	<b>19</b>
3.1 Magnetic Resonance Signal Models . . . . .	19
3.2 Signal and Gradient Dictionaries . . . . .	21
3.3 Training, Validation and Testing Data . . . . .	22
3.4 Network . . . . .	26
<b>4 Results</b>	<b>29</b>
4.1 Monoexponential Model . . . . .	30
4.1.1 Validation . . . . .	31
4.1.2 Testing . . . . .	37
4.2 Extended Phase Graphs Model . . . . .	38

4.2.1	Validation . . . . .	40
4.2.2	Testing . . . . .	41
4.3	Extended Phase Graphs including the effect of B1 . . . . .	43
4.3.1	Validation . . . . .	43
4.3.2	Testing . . . . .	49
4.4	Performance Analysis . . . . .	53
<b>5</b>	<b>Conclusions</b>	<b>59</b>
5.1	Achievements . . . . .	59
5.2	Future Work . . . . .	61
	<b>Bibliography</b>	<b>63</b>
<b>A</b>	<b>Statistics</b>	<b>67</b>

# List of Tables

- 3.1 Mean ground-truth PD and  $T_2$  values per tissue,  $\mu_{\text{tissue}}^{\text{GT}}$ , used to simulate realistic MRI data 24
- 3.2 Specifications of the simulated training, validation and testing datasets . . . . . 25



# List of Figures

2.1	Schematic representation of the general protocol to perform $T_2$ mapping . . . . .	6
2.2	Diagrams of the position of an isochromat in 3D and 2D space . . . . .	8
2.3	The partitioning effect of a radiofrequency pulse . . . . .	9
2.4	All possible isochromat population exchange pathways due to the application of a radiofrequency . . . . .	10
2.5	Graphical representation of the first three discrete configuration states for dephasing configurations, rephasing configurations and longitudinal configuration pairs . . . . .	12
2.6	Schematic representation of the effect of $B_1$ inhomogeneities . . . . .	13
2.7	Estimated $T_2$ maps with the pure-exponential signal model, the EPG concept, and the EPG concept accounting for the effect of the $B_1$ field . . . . .	13
2.8	Diagram showcasing the sequential steps to build an extended phase diagram . . . . .	14
2.9	General architecture of an RNN . . . . .	15
2.10	General configuration of the RIM . . . . .	17
3.1	Extended phase diagram of the ME-SE sequence used in this work . . . . .	21
3.2	Diagram of the EPG dictionaries . . . . .	22
3.3	Protocol for generating ground-truth parametric maps and weighted images from discrete anatomical models . . . . .	23
3.4	Simulated $B_1$ maps of a full $160 \times 160$ slice for the lower and upper bounds of $I^{B_1}$ , $\sigma^{B_1}$ and $\mu^{B_1}$ . . . . .	24
3.5	RIM configuration depicted for two inference steps . . . . .	27
3.6	RNNCell configuration . . . . .	28
4.1	$T_2$ maps estimated by two different networks on the same validation image, and respective NRMSE, PSNR and SSIM scores . . . . .	30
4.2	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noisy validation dataset by $M_{EXP}$ networks trained with 12, 24, 48 and 72 patches . . . . .	32
4.3	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noisy validation dataset by $M_{EXP}$ networks trained with 2, 4, 6 and 8 inference steps . . . . .	33
4.4	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noiseless validation dataset by $M_{EXP}$ networks trained with different acquisition noise amounts . . . . .	34

4.5	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noisy validation dataset by $M_{EXP}$ networks trained with different acquisition noise amounts . . . . .	35
4.6	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the very noisy validation dataset by $M_{EXP}$ networks trained with different acquisition noise amounts . . . . .	36
4.7	$T_2$ ground-truth maps of the simulated testing datasets, and corresponding estimates produced from exponential fitting and the $M_{EXP}$ RIM . . . . .	37
4.8	$T_2$ ground-truth maps of the simulated testing datasets with no intra-tissue variability, and corresponding estimates produced from exponential fitting and the $M_{EXP}$ RIM . . . . .	38
4.9	$T_2$ maps of the <i>in vivo</i> testing dataset estimated with the exponential fitting method and the $M_{EXP}$ RIM. . . . .	39
4.10	$T_2$ maps estimated on one of the <i>in vivo</i> testing images for $M_{EPG(T_2)}$ dictionaries with $T_1=600, 800, 1000$ and $1200$ ms . . . . .	39
4.11	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noisy validation dataset by $M_{EPG(T_2)}$ networks trained with 36, 48, 60 and 72 patches . . . . .	40
4.12	$T_2$ ground-truth maps of the simulated testing datasets and corresponding estimates obtained from dictionary matching and the $M_{EPG(T_2)}$ RIM . . . . .	41
4.13	$T_2$ ground-truth maps of the simulated testing datasets with no intra-tissue variability and corresponding estimates obtained from dictionary matching and the $M_{EPG(T_2)}$ RIM . . . . .	42
4.14	$T_2$ maps estimated from the <i>in vivo</i> testing dataset with the dictionary matching technique and the $M_{EPG(T_2)}$ RIM . . . . .	42
4.15	$T_2$ maps estimated on one of the <i>in vivo</i> testing images for $M_{EPG(T_2, B_1)}$ dictionaries with $T_1=600, 800, 1000$ and $1200$ ms . . . . .	43
4.16	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noisy validation dataset by $M_{EPG(T_2, B_1)}$ networks trained with 48, 60, 72 and 100 patches . . . . .	44
4.17	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noisy validation dataset by $M_{EPG(T_2, B_1)}$ networks trained with 2, 4, 6 and 8 inference steps . . . . .	45
4.18	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noiseless validation dataset by $M_{EPG(T_2, B_1)}$ networks trained with different acquisition noise amounts . . . . .	46
4.19	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the noisy validation dataset by $M_{EPG(T_2, B_1)}$ networks trained with different acquisition noise amounts . . . . .	47
4.20	NRMSE, PSNR and SSIM scores of the estimated $T_2$ maps on the very noisy validation dataset by $M_{EPG(T_2, B_1)}$ networks trained with different acquisition noise amounts . . . . .	48
4.21	$T_2$ ground-truth maps of the simulated testing datasets and corresponding estimates obtained with dictionary matching and the $M_{EPG(T_2, B_1)}$ RIM . . . . .	49
4.22	$T_2$ ground-truth maps of the simulated testing datasets with no intra-tissue variability and corresponding estimates obtained with dictionary matching and the $M_{EPG(T_2, B_1)}$ RIM . . . . .	50
4.23	$B_1$ and $T_2$ ground-truth maps of the simulated noisy testing dataset with three different $B_1$ fields and corresponding estimates produced from a $M_{EPG(T_2, B_1)}$ RIM . . . . .	51

4.24 $T_2$ maps of the <i>in vivo</i> testing dataset estimated with the dictionary matching technique and the $M_{\text{EPG}(T_2, B_1)}$ RIM . . . . .	52
4.25 Average time needed to obtain a $T_2$ map from a set of squared weighted images with varying dimensions, for protocols with 12 and 32 echoes, for the RIM, RIMFoGraD, and dictionary matching . . . . .	54
4.26 $T_2$ maps estimated from one image of the <i>in vivo</i> testing dataset with the RIM network and with the RIMFoGraD networks for varying dictionaries with decreasing precision, and difference maps between the RIM and respective RIMFoGraD . . . . .	55
4.27 Average time needed to obtain a $T_2$ map with the RIM and RIMFoGraD for a different number of inference steps . . . . .	56
4.28 Average time required to estimate a $T_2$ map with increasing dimensions for the RIM, the RIMFoGraD, exponential fitting and dictionary matching . . . . .	57



# Nomenclature

## Magnetic Resonance Imaging

$\alpha$	Flip angle of a radiofrequency pulse
$\gamma$	Gyromagnetic ratio of protons
$\omega_0$	Larmor frequency
$\Phi$	Phase angle of a radiofrequency pulse
$\phi$	Phase angle of a dephasing isochromat
<b>E</b>	Relaxation matrix
<b>k</b>	Angular wave vector
<b>M</b> = $[M_x, M_y, M_z]^T$	Isochromat
<b>m(x, y, t)</b>	Contrast-weighted image in image coordinates
<b>M</b> <sup>+</sup>	Isochromat after the application of an RF pulse
<b>M</b> <sup>-</sup>	Isochromat before the application of an RF pulse
<b>p</b> = $[\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]$	relaxometry parametric maps
<b>r</b>	Distance of an isochromat to the isocenter
<b>R</b> <sub>X</sub>	Rotation matrix describing the effect of a radiofrequency pulse along the <i>x</i> -axis
<b>R</b> <sub>Z</sub>	Rotation matrix describing the dephasing of an isochromat
<b>S</b>	Shift operator
<b>s(k<sub>x</sub>, k<sub>y</sub>, t)</b>	Contrast-weighted image in k-space coordinates
<b>S, S</b> <sup>-1</sup>	Pair of similarity matrices
<b>T</b> <sub>Φ</sub>	Mixing matrix
<b>T</b> <sub>X</sub>	Rotation matrix describing the effect of a radiofrequency pulse along the <i>x</i> -axis for the configuration states

$\mathbf{T}_Z$	Rotation matrix describing the dephasing of an isochromat for the configuration states
$\tilde{F}_+, \tilde{F}_-, \tilde{Z}$	Configuration states
$\tilde{F}_k, \tilde{F}_{-k}^*, \tilde{Z}_k$	Discrete configuration states
$B_0$	External magnetic field
$B_1$	Radiofrequency field
$G$	Linear gradient-field applied along $z$
$t, \tau$	Time variables

### Machine Learning

$\Lambda$	Loss function
$\mathcal{L}(\mathbf{m} \mid \mathbf{p})$	Negative log-likelihood function
$\mathcal{L}(\mathbf{p})$	Negative logarithm of the prior probability
$\varepsilon$	Noise associated with the measurements
$f(\mathbf{m} \mid \mathbf{p})$	Likelihood function
$J$	Number of inference steps
$M$	forward model of a signal

### Mathematical Operations

$\nabla$	Gradient operator
$\mathbf{F}$	Fourier Transform operator

# Acronyms

<b>CSF</b>	cerebrospinal fluid. 22, 24, 29–36, 39, 40, 42, 44–48, 52
<b>DCRNet</b>	deep complex residual network. 3
<b>DOPAMINE</b>	deep model-based magnetic resonance parameter mapping network. 3
<b>EMC</b>	echo-modulation curve. 2
<b>EPD</b>	extended phase diagram. 13, 14, 20, 21
<b>EPG</b>	extended phase graph. 2–7, 11, 13, 19–22, 27, 38, 43, 44, 53, 56, 59, 61
<b>ESP</b>	echo-spacing. 1, 19, 21, 25
<b>ETL</b>	echo-train length. 3, 19, 25, 53, 60
<b>FFT</b>	fast Fourier transform. 5
<b>FID</b>	free induction decay. 10
<b>FSE</b>	fast spin-echo. 30
<b>GD</b>	gradient descent. 16
<b>GRU</b>	gated recurrent unit. 15, 26, 28
<b>MANTISGAN</b>	model augmented neural network with incoherent k-space sampling generative adversarial network. 3
<b>MAP</b>	maximum a posteriori. 6, 15, 16, 67
<b>ME-SE</b>	multi-echo spin echo. 2, 6, 14, 19, 21, 25, 29
<b>MLE</b>	maximum likelihood estimator. 67
<b>MLP</b>	multilayer perceptron. 3
<b>MR</b>	magnetic resonance. 1–7, 9, 11–13, 19, 21, 22, 26, 67
<b>MRF-DRONE</b>	magnetic resonance fingerprinting deep reconstruction network. 3
<b>MRI</b>	magnetic resonance imaging. 1, 2, 5, 12, 24, 61
<b>MSE</b>	mean squared error. 26
<b>NRMSE</b>	normalized root mean square error. 29–38, 40–42, 44–51

<b>PD</b>	proton density. 2, 6, 19, 22–27, 53, 61
<b>PSNR</b>	peak signal-to-noise ratio. 29–36, 40, 44–48
<b>Q<sup>2</sup>MRI</b>	qualitative and quantitative magnetic resonance imaging. 3
<b>RF</b>	radiofrequency. 6, 9–14, 19, 21, 60
<b>RIM</b>	recurrent inference machines. 3–5, 15–17, 26, 27, 30, 35, 37–43, 47, 49–57, 59–61
<b>RIMFoGraD</b>	recurrent inference machines coupled with <b>f</b> orward model and <b>g</b> radient dictionaries. 4, 27, 30, 53–57, 59–61
<b>RNN</b>	recurrent neural network. 3, 15, 16, 26
<b>ROA</b>	rotation operator algorithm. 6
<b>SCQ</b>	<b>s</b> patially <b>c</b> onstrained tissue <b>q</b> uantification. 3
<b>SNR</b>	signal-to-noise ratio. 26, 59
<b>SSIM</b>	structural similarity index measure. 29–38, 40–42, 44–51
<b>TR</b>	repetition time. 1, 2, 25
<b>VN</b>	variational network. 3

# Chapter 1

## Introduction

In this chapter, the motivation and main objectives of this work are presented, along with a brief overview of the topic at hand.

### 1.1 Motivation

Magnetic resonance imaging (MRI) provides non-invasive imaging of the human body for a wide variety of clinical applications. The signal intensity of a magnetic resonance (MR) image depends on the tissue relaxation properties  $T_1$  and  $T_2$ , on acquisition parameters such as repetition time (TR), echo-spacing (ESP) and flip angle, and on the MR system conditions (homogeneities of the radiofrequency field  $B_1$  and magnetic field  $B_0$ , and uniformity of receiver coil sensitivity) [1]. MRI offers better soft-tissue characterization and more flexible contrast mechanisms without radiation exposure, but its day-to-day use is still very often limited to the qualitative assessment of contrast-weighted images, created based on the variation of underlying MR tissue parameters [2], which presents a bottleneck in image analysis and digital healthcare [1].

The term “MRI relaxometry” or “MRI mapping” refers to the quantitative assessment of inherent tissue  $T_1$  and  $T_2$  values. Relaxometry is more sensitive to different pathologies with more specific information on tissue composition and microstructure (enabling early clinical diagnosis) when compared with conventional  $T_1$  and  $T_2$  weighted images, and also more robust to surface coil effects [3]. Relaxometry maps depend less on hardware variations, acquisition settings and operator expertise when compared to qualitative weighted images, and are highly reproducible [4]. Typically, the extraction of quantitative tissue properties requires repeated acquisitions of datasets with varying MR parameters [1, 3], so relaxometry requires much longer scan times than qualitative contrast-weighted images. The speed of the imaging exam is crucial in many applications, hence a large volume of research in relaxometry focuses on accelerating parameter mapping and improving robustness to noise and artifacts.

## 1.2 Topic Overview

Quantitative MRI is accomplished by scanning data in k-space, which is reconstructed into a set of weighted images later fit into a relevant signal model. Multi-echo spin echo (ME-SE) sequences are commonly used for  $T_2$  mapping due to their reduced scanning time, since they scan multiple time points along the  $T_2$  decay for each k-space line during a single TR. A monoexponential curve is then typically fit to the measured time points, resulting in the estimation of the proton density (PD) and  $T_2$  relaxation constant of each tissue. However, the magnetization in a ME-SE sequence does not actually follow a pure exponential curve. In fact, each refocusing pulse separates the magnetization into three coherence pathways, contaminating the signal with stimulated and indirect echoes [5]. A more precise way to model the magnetization in  $T_2$  decay is to use a generalized echo-modulation curve (EMC) based on the Bloch equations, which depends not only on the relaxation values, but also on a variety of experimental and physical parameters. The extended phase graph (EPG) concept is an elegant tool to depict the magnetization response of a variety of MR sequences, but researchers and users often refrain from applying EPGs due to the increased complexity added to the signal model [6].

EPGs have been successfully applied to  $T_2$  estimation. For instance, Ben-Eliezer et al. [5] developed a post-processing approach where EPGs were used to build a dictionary of EMCs, each corresponding to a unique  $T_2$  value, and then matching the experimentally measured data against the dictionary on a pixel-by-pixel basis. This approach bypasses the erroneous estimation of  $T_2$  values when fitting a pure exponential curve to multi-echo EMC, and the results closely matched the maps produced from single spin-echo data.

Research on acceleration in relaxometry typically focuses on two aspects: reducing the time required to acquire the scanner data in k-space, and improving the speed of parameter fitting the signal model to the set of weighted images obtained from the measurements.

To reduce the scan time of quantitative MR parameter mapping, k-space data can be undersampled below the Nyquist rate instead of acquiring the entire k-space, and different techniques can be employed to reconstruct a set of weighted images from the incomplete k-space data [7]. In parallel MRI, multiple-coil arrays are used to skip certain k-space measurements; the information from multiple coils is used to fill-in the missing k-space data and unfold aliased undersampled images. However, the maximum acceleration that can be achieved with parallel MRI is limited by the number of coil elements and the design of coil arrays [2]. In Compressed Sensing, the reconstructed image is assumed to be sparse in a transform domain and regularization with the  $\ell_1$ -norm is incorporated into the framework [2].

Traditional MR parameter fitting consists in finding the least-squares solution that minimizes the error between the underlying signal evolution and the signal curve synthesized from the model parameters. It is typically implemented with an iterative algorithm, which is computationally taxing and time-consuming. To tackle this issue, numerous techniques have been developed, such as pattern recognition approaches, where a dictionary is simulated with signal evolutions from a range of possible parameters, by considering relevant signal models or the Bloch equations, and a pattern-based recognition method is employed to search for the dictionary signal that best matches the signal evolution observed for a given

pixel. The post-processing approach mentioned above, where EPGs were used to build a dictionary, is one of such methods.

Recently, with the introduction of deep learning, acceleration rates have been pushed further thanks to the variety of neural network configurations that find direct correlations between relaxometry images and parameter maps. Deep learning networks have been applied to (i) improve image reconstruction from k-space, such as the variational network (VN) [8], DeepT1 [9], DeepResolve [10], and deep complex residual network (DCRNet) [11]; (ii) directly estimate the parametric maps from the k-space data, such as the **m**agnetic **r**esonance **f**ingerprinting **d**eep **r**econstruction **n**etwork (MRF-DRONE) [12], **s**patially **c**onstrained tissue **q**uantification (SCQ) [13], and qualitative and quantitative magnetic resonance imaging (Q<sup>2</sup>MRI) [1]; (iii) estimate the parametric maps in a model-based approach, such as **m**odel **a**ugmented **n**eural **n**etwork with **i**ncoherent k-space **s**ampling **g**enerative **a**dversarial **n**etwork (MANTISGAN) [14], **d**eep **m**odel-based magnetic resonance **p**arameter **m**apping **n**etwork (DOPAMINE) [7], and recurrent inference machines (RIM) [4]. The main challenge of applying deep learning to MR reconstruction is the acquisition of training datasets. In particular, because MR relaxometry is not routinely performed in a clinical environment, the accumulation of training datasets is severely limited [2]. The latter class of networks, called model-based deep learning nets, incorporate a physical model of the signal into the deep learning network, thus reducing the size of the necessary training dataset.

Despite the promising results of model-based deep learning networks, most of the architectures developed for  $T_2$  mapping consider the pure exponential approximation, and none of the literature analysed included the EPG concept to produce the signal model in a model-based deep learning approach. Amidst the research done with deep learning and EPGs, Liu et al. [14] have developed a recurrent neural network (RNN) to speed up the computation of large scale MR signals and derivatives following an EPG-Bloch model, while Yu et al. [15] used the EPG concept to generate synthetic data to train a multilayer perceptron (MLP) for  $T_2$  relaxometry, and Cohen et al. [12] applied the EPG concept to generate training signals for cardiac imaging.

### 1.3 Objectives and Contributions

In this work, we propose adapting one of the model-based deep learning architectures, the RIM, to perform  $T_2$  relaxometry mapping of the brain, by implementing the EPG concept to calculate the signal model in lieu of the monoexponential approximation. The RIM has been applied to reconstruct weighted images from k-space data by Lønning et al. [16] and to perform  $T_1$  and  $T_2$  map estimation by Sabidussi et al. [4], using the pure exponential MR signal model, and won the 2019's fastMRI challenge in the single-coil track <sup>1</sup>. Its promising results are owed to the network's ability to learn the inference process whilst removing the need for domain knowledge, which results in low overfitting. The main objectives of this work are as follows: (i) adapt the RIM to a new acquisition protocol with longer echo-train length (ETL) keeping the pure exponential model, by taking advantage of the materials made publicly available by Sabidussi et al. [4]; (ii) implement the EPG framework and the partial derivatives

<sup>1</sup><https://fastmri.org/leaderboards/challenge/2019/>

of the corresponding MR signals to replicate the state-of-the-art dictionary matching technique; (iii) modify the RIM's signal model into the EPG concept; (iv) and finally, further increase the complexity of the model by including the effect of  $B_1$  inhomogeneities into the estimation process. Additionally, an alternative framework is proposed, **recurrent inference machines coupled with forward model and gradient dictionaries** (RIMFoGraD), which shares the RIM's configuration but uses the pre-calculated MR signal evolutions in the EPG dictionary to perform faster inference of the  $T_2$  maps.

The work developed for this thesis was submitted as an abstract to the International Society for Magnetic Resonance in Medicine 2023 annual meeting and exhibition with the title "Diving into Extended Phase Graph-based Deep Learning for accurate  $T_2$  mapping with PhasE graph sigNal and Gradients qUantitative Inference machiNe (PENGUIN)".

## 1.4 Thesis Outline

This thesis is organized in 5 chapters. Chapter 1 contains the motivation for our work, and a brief introduction to the problem and the state-of-the-art methods. Chapter 2 gives a detailed description of the MR signal acquisition, the EPG concept, and the main principles behind deep learning and the RIM. In Chapter 3, the proposed implementation of the EPG concept, the MR signal dictionaries, and the RIM configuration used in this work are described. The  $T_2$  maps produced by our network and their accuracy analysis are included in Chapter 4, as well as a performance comparison between RIM, RIMFoGraD, and the state-of-the-art techniques. Finally, we conclude with a discussion of our findings and suggestions for future work in Chapter 5.

# Chapter 2

## Background

This chapter presents a brief introduction on the acquisition of MR signals and on the EPG concept, and a breakdown on the general configuration of RIMs.

### 2.1 The Magnetic Resonance Signal

Typical MRI experiments are accomplished by manipulating the precession frequency of spins, called the Larmor frequency  $\omega_0$ , in a strong external magnetic field  $B_0$ , using on-resonant radio-frequency pulses and gradient fields [17]. The magnetization vectors behave as stipulated by the Bloch equations, which describe their dependence on physical parameters and externally controlled magnetic fields, and results in the MR signal that can be measured.

MR relaxometry is performed by acquiring a set of contrast-weighted images with varying contrast and parameters and fitting the signal evolution of each pixel to a relaxometry model across the parameter dimension, as illustrated by figure 2.1. The weighted signals  $\mathbf{s}(\mathbf{k}_x, \mathbf{k}_y, \mathbf{t})$  are measured in k-space, where  $k_x$  and  $k_y$  are the spatial-frequency variables in k-space, and  $\mathbf{t}$  is the time vector, and can be converted into a time-dynamic image series  $\mathbf{m}(x, y, \mathbf{t})$  in image space coordinates  $x$  and  $y$  using the Fourier Transform. This conversion is typically named image reconstruction. The MR forward model for acquiring MR data of a 2D time-dynamic image series can thus be written as equation 2.1 [2]. When discretized, equation 2.1 can be rewritten in matrix notation with the fast Fourier transform (FFT) operator  $\mathbf{F}$  (equation 2.2).

$$\mathbf{s}(\mathbf{k}_x, \mathbf{k}_y, \mathbf{t}) = \iint \mathbf{m}(x, y, \mathbf{t}) e^{-i2\pi(\mathbf{k}_x x + \mathbf{k}_y y)} dx dy \quad (2.1)$$

$$\mathbf{s} = \mathbf{F} \cdot \mathbf{m} \quad (2.2)$$

Having the reconstructed dynamic image series  $\mathbf{m}$ , the following step in relaxometry is to fit the data into a relevant signal model  $M$  to find the  $n$  MR parameters  $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2 \dots, \mathbf{p}_n]$  by performing least-squares minimization (equation 2.3):

$$\hat{\mathbf{p}} = \arg \min_{\mathbf{p}} \|M(\mathbf{p}) - \mathbf{m}\|_2^2 \quad (2.3)$$

Equation 2.3 can be reformulated as the optimization of the maximum a posteriori (MAP) estimator from statistics (equation 2.4), considering the likelihood  $f(\mathbf{m}|\mathbf{p})$  that model  $M(\mathbf{p})$  explains the observed data  $\mathbf{m}$ : the optimization is accomplished by minimizing the sum of the negative log-likelihood  $\mathcal{L}(\mathbf{m}|\mathbf{p})$  that and log-prior distributions  $\mathcal{L}(\mathbf{p})$ .

$$\hat{\mathbf{p}} = \arg \max_{\mathbf{p}} \{\mathcal{L}(\mathbf{m}|\mathbf{p}) + \mathcal{L}(\mathbf{p})\} \quad (2.4)$$

In  $T_2$  mapping, specifically, the points scanned with ME-SE pulse sequences are typically reconstructed into a series of weighted images, which is then used to fit the exponential model in equation 2.5. Here, two MR parameters  $\mathbf{p}_1, \mathbf{p}_2$  are estimated: the  $T_2$  decay constant and the PD of each tissue.

$$\mathbf{m}(\mathbf{x}, \mathbf{y}, t) = \text{PD}(\mathbf{x}, \mathbf{y}) \cdot \exp\left(-\frac{t}{T_2(\mathbf{x}, \mathbf{y})}\right) \quad (2.5)$$

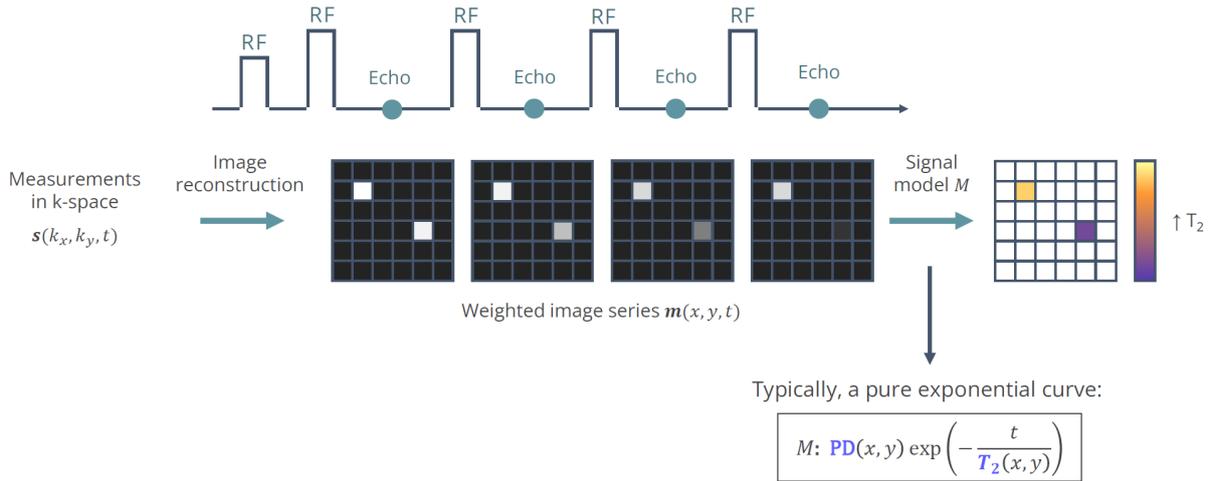


Figure 2.1: Schematic representation of the general protocol to perform  $T_2$  mapping.

However, the monoexponential model in equation 2.5 is only an approximation of the magnetization behaviour when an ME-SE sequence is used for  $T_2$  mapping. In fact, each refocusing pulse separates the magnetization into three coherence pathways, which contaminate the signal with stimulated and indirect echoes [5]. A more precise way of modelling the magnetization in  $T_2$  decay is to use the EPG concept, which is introduced in the following section.

### 2.1.1 The Extended Phase Graph Representation

The common way to describe magnetization is using the rotation operator algorithm (ROA): the Bloch equations are converted into rotations of classical magnetization vectors, named isochromats. Applying a radiofrequency (RF) pulse along the  $x$ -axis rotates all isochromats around the  $x$ -axis by the flip angle

$\alpha$  (rotation matrix 2.6a), while dephasing rotates each isochromat around the  $z$ -axis by a phase angle  $\phi$  (rotation matrix 2.6b). The magnetization response to an MR sequence can thus be estimated from an ensemble of isochromats, each with a different dephasing angle  $\phi$ .

$$\mathbf{R}_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (2.6a)$$

$$\mathbf{R}_Z(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & -\cos \phi \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6b)$$

The main drawbacks of this approach are the fact that estimating each isochromat is very expensive computationally-wise, and the fact that diagrams drawn with this method are very difficult to interpret when it comes to deducing information about echo intensity, number of echoes and time points.

EPGs are a representation of the magnetization based on matrix operations applied over configuration states which are easier to interpret. The following subsections describe the deduction of the EPG representation.

### Configuration states

A dephasing isochromat  $\mathbf{M} = [M_x, M_y, M_z]^T$  rotates around the  $z$ -axis with a frequency  $\omega \neq 0$ . Let us name  $\mathbf{r}$  the distance of the isochromat to the isocenter, and  $\phi$  the phase-angle, which depends on time  $t'$  and position. To describe dephasing, the phase-angle of the isochromat is given by equation 2.7, where  $\gamma$  is the gyromagnetic ratio of protons and  $G$  is a linear gradient-field applied along  $z$ . The farther from the isocenter the isochromat is, the larger the phase-angle  $\phi$  will be. The position of the isochromat can be described in  $xy$  space by equations 2.8a and 2.8b. The angular wave vector  $\mathbf{k} = \gamma \int_0^t \mathbf{G}(t') dt'$  constitutes a quantitative measure of dephasing, and is the origin of the term “k-space”.

$$\phi = \gamma \mathbf{r} \int_0^t G(t') dt' \quad (2.7)$$

$$M_x(\mathbf{r}) = M_T \cos \phi = M_T \cos \left( \gamma \mathbf{r} \int_0^t \mathbf{G}(t') dt' \right) = M_T \cos(\mathbf{k}\mathbf{r}) \quad (2.8a)$$

$$M_y(\mathbf{r}) = M_T \sin \phi = M_T \sin \left( \gamma \mathbf{r} \int_0^t \mathbf{G}(t') dt' \right) = M_T \sin(\mathbf{k}\mathbf{r}) \quad (2.8b)$$

The magnetization components  $M_x$  and  $M_y$  have mutual magnetization exchange over time, so it is reasonable to find a more compact way of describing them. This can be accomplished with a change of basis to complex space  $[M_x, M_y, M_z]^T \rightarrow [M_+, M_-, M_z]^T$  (equations 2.9a and 2.9b). The position of the isochromat in 3D and 2D space is illustrated in figure 2.2.

$$M_+(\mathbf{r}) = M_x(\mathbf{r}) + iM_y(\mathbf{r}) = Me^{i\phi(\mathbf{r})} = Me^{i\mathbf{k}\mathbf{r}} \quad (2.9a)$$

$$M_-(\mathbf{r}) = M_x(\mathbf{r}) - iM_y(\mathbf{r}) = Me^{-i\phi(\mathbf{r})} = Me^{-i\mathbf{k}\mathbf{r}} \quad (2.9b)$$

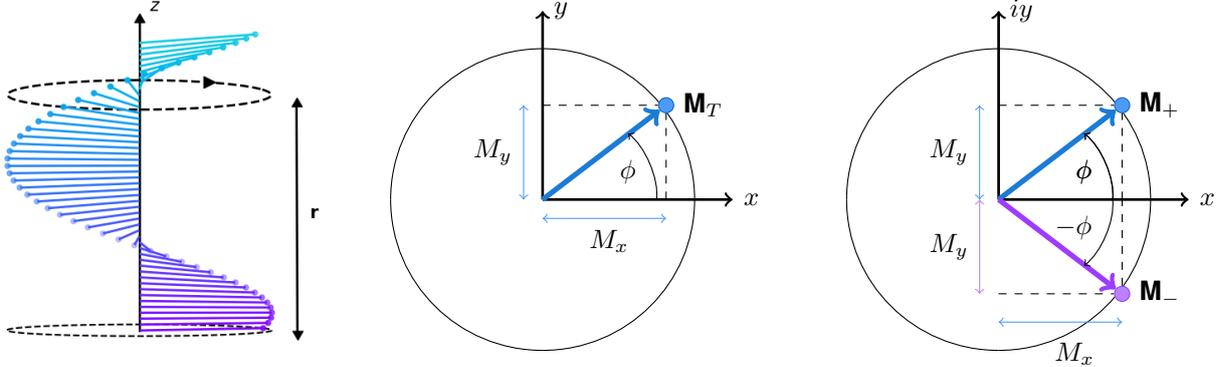


Figure 2.2: Position of an isochromat in 3D and 2D space. A linear gradient is assumed to be applied along the  $z$  direction. Isochromats rotate around the  $z$ -axis and are progressively more dephased (i.e., larger  $\phi$ ) as they move away from the isocenter ( $z = 0$ ). The position of the isochromat can be described in Cartesian coordinates (middle diagram) or in complex space coordinates (right diagram).

The net magnetization of each spatial coordinate ( $\tilde{F}_+$ ,  $\tilde{F}_-$  and  $\tilde{Z}$ ) is calculated as the sum of all isochromats over a macroscopic volume  $V$  (equations 2.10a, 2.10b and 2.10c). These equations are equivalent to the Fourier transform definition.

$$\tilde{F}_+(\mathbf{k}) = \int_V M_+(\mathbf{r})e^{-i\mathbf{k}\mathbf{r}}d^3r \Leftrightarrow M_+(\mathbf{r}) = \int_V \tilde{F}_+(\mathbf{k})e^{i\mathbf{k}\mathbf{r}}d^3k \quad (2.10a)$$

$$\tilde{F}_-(\mathbf{k}) = \int_V M_-(\mathbf{r})e^{-i\mathbf{k}\mathbf{r}}d^3r \Leftrightarrow M_-(\mathbf{r}) = \int_V \tilde{F}_-(\mathbf{k})e^{i\mathbf{k}\mathbf{r}}d^3k \quad (2.10b)$$

$$\tilde{Z}(\mathbf{k}) = \int_V M_z(\mathbf{r})e^{-i\mathbf{k}\mathbf{r}}d^3r \Leftrightarrow M_z(\mathbf{r}) = \int_V \tilde{Z}(\mathbf{k})e^{i\mathbf{k}\mathbf{r}}d^3k \quad (2.10c)$$

Now, the magnetization is described as a sum of complex spatial harmonics, termed the configuration states, with different wave vectors  $\mathbf{k}$ . A change of  $\mathbf{k}$  results in a change of the transverse magnetization of the whole isochromat ensemble.  $\tilde{F}_+$  represents dephasing transverse magnetization and is illustrated by a right-handed helix, while  $\tilde{F}_-$  represents the counterpart, and is illustrated by a left-handed helix.  $\tilde{Z}$  is a complex transform of the real-valued longitudinal magnetization  $M_z$ , defined for convenience.

### Partitioning Effect of a radiofrequency pulse

Let us describe the matrices 2.6a and 2.6b in the configuration states notation. Applying a pair of similarity matrices  $\mathbf{S}$  and  $\mathbf{S}^{-1}$  (2.11) to perform a change of basis, the new rotation matrices are defined as  $\mathbf{T} = \mathbf{S}\mathbf{R}\mathbf{S}^{-1}$  (equations 2.12b and 2.12a).

$$\mathbf{S} = \begin{bmatrix} 1 & +i & 0 \\ 1 & -i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{S}^{-1} = \begin{bmatrix} 1 & 1 & 0 \\ -i & +i & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (2.11)$$

$$\mathbf{T}_X(\alpha) = \begin{bmatrix} \cos^2 \frac{\alpha}{2} & \sin^2 \frac{\alpha}{2} & -i \sin \alpha \\ \sin^2 \frac{\alpha}{2} & \cos^2 \frac{\alpha}{2} & i \sin \alpha \\ -\frac{i}{2} & \frac{i}{2} \sin \alpha & \cos \alpha \end{bmatrix} \quad (2.12a)$$

$$\mathbf{T}_Z(\phi) = \begin{bmatrix} e^{i\phi} & 0 & 0 \\ 0 & e^{-i\phi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12b)$$

A general RF pulse with an initial angle  $\Phi \neq 0$  acting on a complex magnetization vector  $[M_+, M_-, M_z]^T$  will transform it according to equation 2.13.

$$\begin{bmatrix} M_+ \\ M_- \\ M_z \end{bmatrix}^+ = \begin{bmatrix} \cos^2 \frac{\alpha}{2} & e^{2i\Phi} \sin^2 \frac{\alpha}{2} & -ie^{i\Phi} \sin \alpha \\ e^{-2i\Phi} \sin^2 \frac{\alpha}{2} & \cos^2 \frac{\alpha}{2} & ie^{-i\Phi} \sin \alpha \\ -\frac{i}{2} e^{-i\Phi} \sin \alpha & \frac{i}{2} e^{i\Phi} \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} M_+ \\ M_- \\ M_z \end{bmatrix}^- \quad (2.13)$$

$\mathbf{M}^+$  and  $\mathbf{M}^-$  represent the magnetization vector after and before the RF pulse has been applied, respectively. Equation 2.13 represents a very important observation in the MR field: after an RF pulse is applied, each magnetization component behaves as a superposition of three different parts ( $M_+^+$ ,  $M_-^+$  and  $M_z^+$ ).  $M_+^+$  represents dephasing transverse magnetization,  $M_-^+$  represents rephasing transverse magnetization that can produce an echo, and  $M_z^+$  represents longitudinal magnetization. This phenomenon is called the Woessner Decomposition or the Partitioning effect of an RF pulse, and it is illustrated in figure 2.3. Moreover, the fraction of the three different coherence pathways only depends on the flip angle  $\alpha$ ; the RF pulse phase  $\Phi$  is only present in complex phase terms and defines the choice of the rotation system around the  $z$ -axis [6].

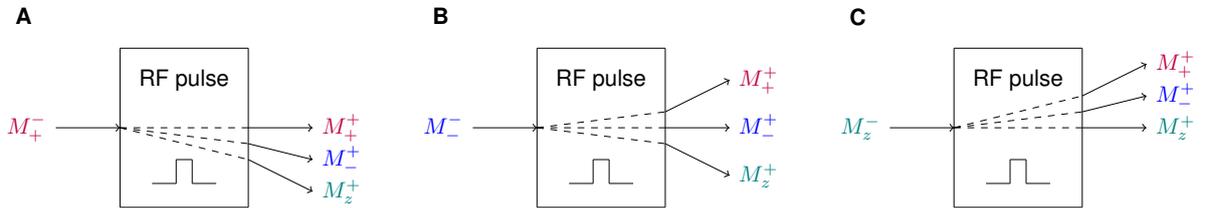


Figure 2.3: The partitioning effect of an RF pulse: when a pulse is applied, each magnetization component is split into three different parts. **a)** From the perspective of initial transverse magnetization ( $M_+^-$ ), the first pathway remains unaffected, while the second part is refocused ( $M_-^+$ ) and the third part becomes longitudinal magnetization ( $M_z^+$ ). **b)** From the perspective of rephasing transverse magnetization ( $M_-^-$ ), the first part becomes dephasing transverse magnetization ( $M_+^+$ ), the second part remains unaffected and the third part becomes longitudinal magnetization ( $M_z^+$ ). **c)** From the perspective of longitudinal magnetization ( $M_z^-$ ), the first part becomes dephasing transverse magnetization ( $M_+^+$ ), the second part becomes rephasing transverse magnetization ( $M_-^+$ ) and the third part remains unaffected [6].

## Population exchange between configuration states

Equation 2.13 applied on the configuration states becomes equation 2.14, which defines all the possible pathways for magnetization populations to exchange. Note that RF pulses only mix configuration states with the same dephasing order  $\mathbf{k}$ .

$$\begin{bmatrix} \tilde{F}_+(\mathbf{k}) \\ \tilde{F}_-(\mathbf{k}) \\ \tilde{Z}(\mathbf{k}) \end{bmatrix}^+ = \begin{bmatrix} \cos^2 \frac{\alpha}{2} & e^{2i\Phi} \sin^2 \frac{\alpha}{2} & -ie^{i\Phi} \sin \alpha \\ e^{-2i\Phi} \sin^2 \frac{\alpha}{2} & \cos^2 \frac{\alpha}{2} & ie^{-i\Phi} \sin \alpha \\ -\frac{i}{2}e^{-i\Phi} \sin \alpha & \frac{i}{2}e^{i\Phi} \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \tilde{F}_+(\mathbf{k}) \\ \tilde{F}_-(\mathbf{k}) \\ \tilde{Z}(\mathbf{k}) \end{bmatrix}^- \quad (2.14)$$

When populations move from  $\tilde{Z}(0)$  to  $\tilde{F}_\pm(0)$ , there is a free induction decay (FID): the equilibrium magnetization is excited into coherent transverse magnetization, which starts dephasing. Storing is an exchange from  $\tilde{F}_+(\mathbf{k})$  and  $\tilde{F}_-(\mathbf{k})$  to  $\tilde{Z}(\mathbf{k})$ ,  $\mathbf{k} \neq 0$ : non-coherent transverse magnetization is converted into longitudinal magnetization and stops dephasing.

When any repetitive sequence of RF pulses is applied to an ensemble of isochromats, there is a combination of echoes. A spin-echo results from a phase reversal  $\tilde{F}_+(\mathbf{k}) \rightarrow \tilde{F}_-(-\mathbf{k})$ ,  $\mathbf{k} \neq 0$ , whereas a stimulated-echo results from an exchange from  $\tilde{Z}(\mathbf{k}) \rightarrow \tilde{F}_-(-\mathbf{k})$ ,  $\mathbf{k} \neq 0$ . When admitting the monoexponential signal model approximation, the contribution of stimulated-echoes is not taken into account. Figure 2.4 displays a graphical representation of the possible isochromat population exchanges between configuration states due to the effect of an RF pulse.

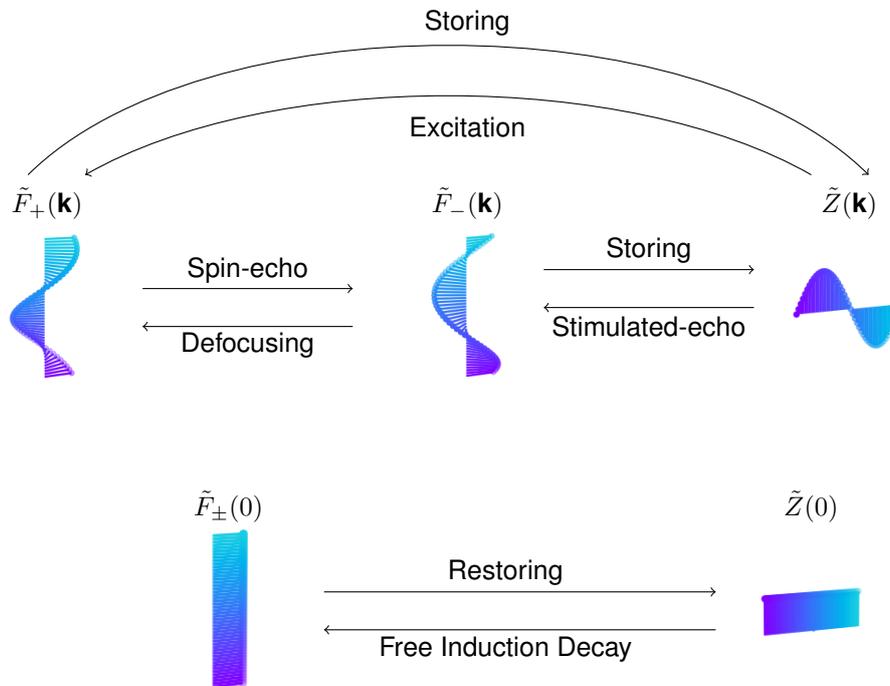


Figure 2.4: All possible isochromat population exchange pathways due to the application of an RF pulse.

## Discrete configuration states

When the MR sequence has equidistant timings, it becomes more practical to use a dephasing angle  $\theta(\mathbf{r})$  in place of the spatial coordinate  $\mathbf{r}$  [6]. The magnetization is then described by equations 2.15a-2.15c, where  $\mathcal{R}$  represents the real part of a complex quantity, and  $\tilde{F}_{-k}^*$  is the complex conjugate of  $\tilde{F}_k$ .

$$M_+(\theta) = \sum_{k=-\infty}^{+\infty} \tilde{F}_k e^{i\mathbf{k}\theta} \quad (2.15a)$$

$$M_-(\theta) = \sum_{k=-\infty}^{+\infty} \tilde{F}_{-k}^* \cdot e^{i\mathbf{k}\theta} \quad (2.15b)$$

$$M_z(\theta) = \sum_{k=-\infty}^{+\infty} \tilde{Z}_k \cdot e^{i\mathbf{k}\theta} = \mathcal{R} \left( \sum_{k=0}^{\infty} \tilde{Z}_k \cdot e^{i\mathbf{k}\theta} \right) \quad (2.15c)$$

The effect of an RF pulse in a sequence with equidistant timings can thus be described by equation 2.16.  $\mathbf{k}$  is now an integral number describing the state's dephasing in units of  $2\pi$ ; in other words, each state  $k$  will have a dephasing of  $2\pi k$ . A visual representation of the first three states of  $\tilde{F}_k$ ,  $\tilde{F}_{-k}^*$  and  $\tilde{Z}_k$  can be observed in figure 2.5.

$$\begin{bmatrix} \tilde{F}_k \\ \tilde{F}_{-k}^* \\ \tilde{Z}_k \end{bmatrix}^+ = \begin{bmatrix} \cos^2 \frac{\alpha}{2} & e^{2i\Phi} \sin^2 \frac{\alpha}{2} & -ie^{i\Phi} \sin \alpha \\ e^{-2i\Phi} \sin^2 \frac{\alpha}{2} & \cos^2 \frac{\alpha}{2} & ie^{-i\Phi} \sin \alpha \\ -\frac{i}{2} e^{-i\Phi} \sin \alpha & \frac{i}{2} e^{i\Phi} \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \tilde{F}_k \\ \tilde{F}_{-k}^* \\ \tilde{Z}_k \end{bmatrix}^- \quad (2.16)$$

## Relaxation and dephasing

Dephasing caused by a gradient with 0th moment  $\Delta k = \gamma \int_{t'}^t G(t') dt'$  can be implemented with the EPG representation as a shift operator  $\mathbf{S}$  (equation 2.17) [6]. Only transverse states experience phase evolution.

$$\mathbf{S}(\Delta k) : \tilde{F}_k \rightarrow \tilde{F}_{k+\Delta k} \quad \text{and} \quad \tilde{Z}_k \rightarrow \tilde{Z}_k \quad (2.17)$$

$T_1$  and  $T_2$  relaxation can be described using a matrix operator  $\mathbf{E}$  defined in 2.18. All configuration states that are off-equilibrium ( $k \neq 0$ ) decay inevitably due to relaxation; longitudinal equilibrium magnetization is also subjected to a  $T_1$  recovery term towards  $M_z = M_0$ .

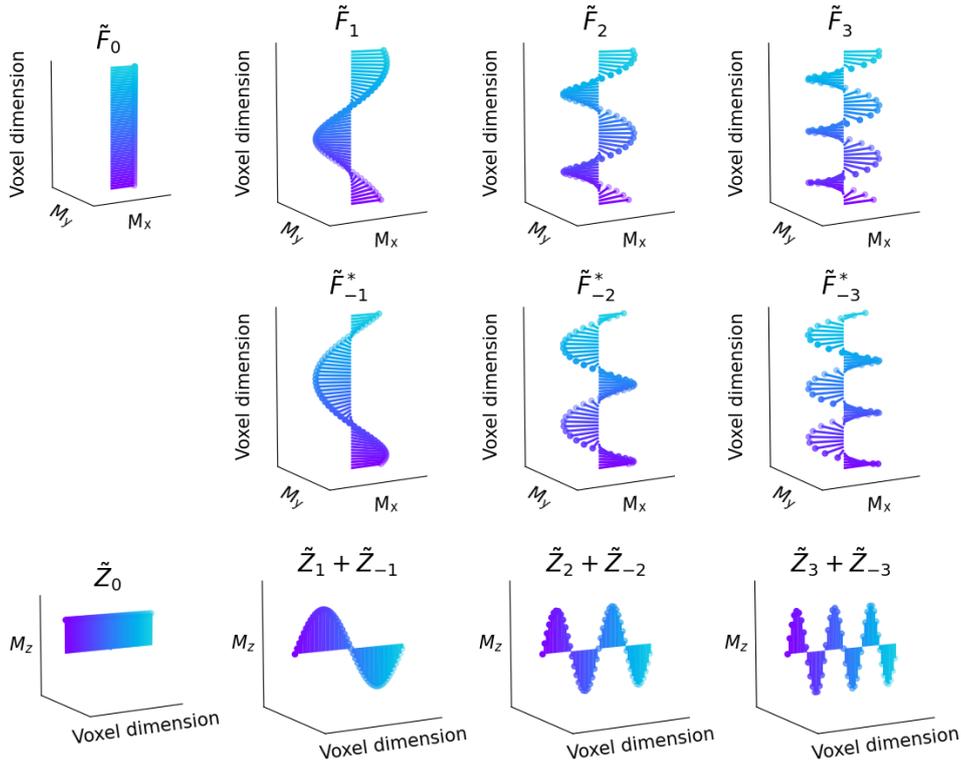


Figure 2.5: Graphical representation of the first three discrete configuration states for dephasing configurations ( $\tilde{F}_k$ ), rephasing configurations ( $\tilde{F}_{-k}^*$ ) and longitudinal configuration pairs ( $\tilde{Z}_k + \tilde{Z}_{-k}$ ).

$$\begin{aligned}
 \mathbf{E}(T_1, T_2, \tau, k \neq 0) &= \begin{bmatrix} e^{-\frac{\tau}{T_2}} & 0 & 0 \\ 0 & e^{-\frac{\tau}{T_2}} & 0 \\ 0 & 0 & e^{-\frac{\tau}{T_1}} \end{bmatrix} \\
 \mathbf{E}(T_1, T_2, \tau, k = 0) &= \begin{bmatrix} e^{-\frac{\tau}{T_2}} & 0 & 0 \\ 0 & e^{-\frac{\tau}{T_2}} & 0 \\ 0 & 0 & e^{-\frac{\tau}{T_1}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M_0 \left(1 - e^{-\frac{\tau}{T_1}}\right) \end{bmatrix} \quad (2.18)
 \end{aligned}$$

The evolution of the MR signal can thus be described by multiplying a matrix  $\Omega$  containing the discrete conjugation states by the matrices that account for the application of RF pulses ( $\mathbf{T}_\Phi$ , as defined in equation 2.5), relaxation ( $\mathbf{E}$ ) and dephasing ( $\mathbf{S}$ ).

### Influence of the $B_1$ field

In MRI, the  $B_1$  field is responsible for tipping the magnetization by the flip angle  $\alpha$  from its initial alignment with  $\vec{B}_0$ . When the  $B_1$  field is homogeneous, the magnetization will be tipped by exactly  $\alpha$ , but this is not true if the  $B_1$  field contains inhomogeneities. In this case, if we use relative units taking as reference the nominal  $B_1$  value that would produce the design flip angle  $\alpha$ , the flip angle of the magnetization will be given by  $\alpha B_1$ , that is, the larger the magnitude of the  $B_1$  field, the larger the flip

angle, and vice-versa. Figure 2.6 schematically illustrates this effect. To model this effect with the EPG concept, the flip angles  $\alpha$  should be multiplied by the effective  $B_1$  field in percentage.

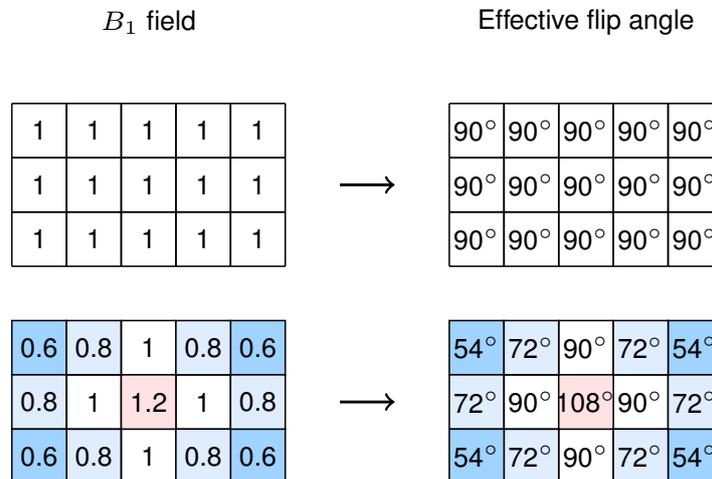


Figure 2.6: Schematic representation of the effect of  $B_1$  inhomogeneities, with a design flip angle  $\alpha = 90^\circ$ . When  $B_1$  is homogeneous, the effective flip angle is  $\alpha=90^\circ$ . When  $B_1$  is inhomogeneous, the effective flip angle will be a percentage of the design flip angle.

Figure 2.7 compares estimated  $T_2$  maps of the same image data using the pure exponential approximation, the EPG concept, and the EPG concept accounting for the effect of the  $B_1$  field.

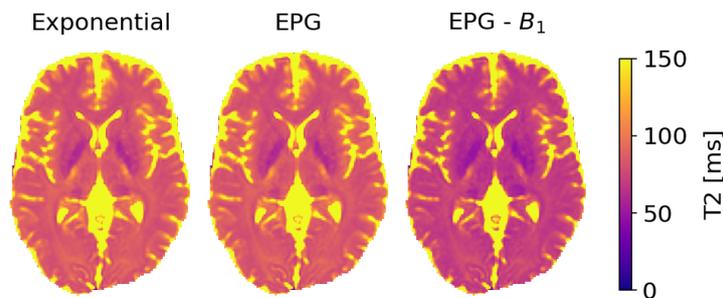


Figure 2.7: Estimated  $T_2$  maps with the pure-exponential signal model, the EPG concept, and the EPG concept accounting for the effect of the  $B_1$  field.

### Extended Phase Diagrams

The magnetization response during an MR sequence can be traced with an extended phase diagram (EPD). These describe the configuration states the system of isochromats can occupy in response to RF pulses and dephasing. The configuration states are represented in the  $y$ -axis, while time is located on the  $x$ -axis. Vertical solid black lines indicate an RF pulse; the other solid black lines correspond to  $\tilde{F}$  states and have a slope equal to the gradient being applied including  $B_0$  inhomogeneity effects. Dashed lines represent  $\tilde{Z}$  states, and are horizontal since longitudinal magnetization does not have

phase evolution. Transitions to the  $\tilde{F}_{-k}^*$  and  $\tilde{Z}_{-k}$  states are typically omitted for readability purposes. However, their influence is still visible in the diagram — when they intersect the  $x$ -axis ( $\tilde{F}_0$  state), the resulting echo is marked with a disc marker. Figure 2.8 showcases the main steps to build an EPD of a ME-SE sequence.

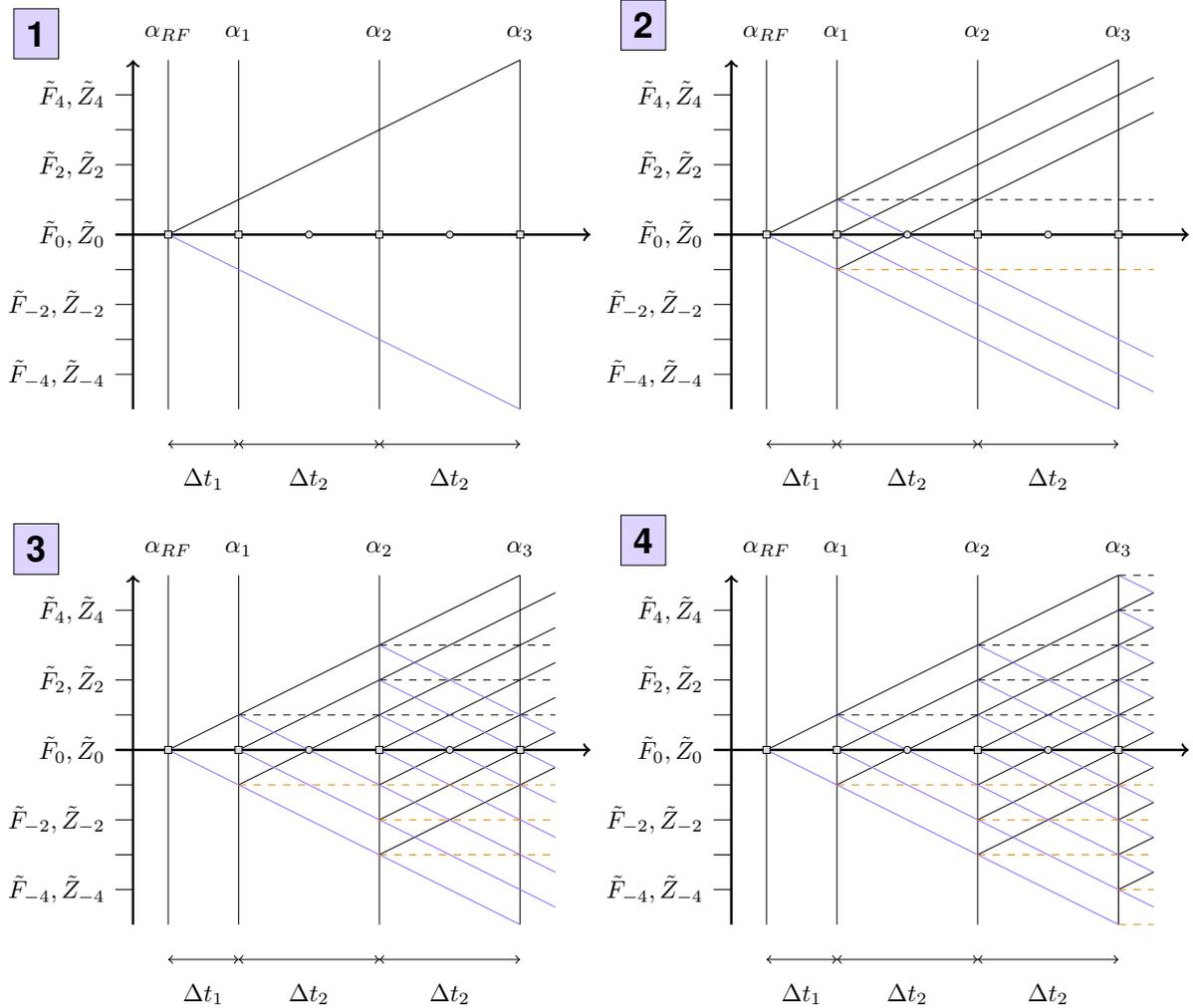


Figure 2.8: Diagram showcasing the sequential steps to build an EPD. The black-coloured lines are the ones typically drawn in EPDs. The purple lines correspond to the  $\tilde{F}_{-k}^*$  pathways and the orange lines represent transitions to  $\tilde{Z}_{-k}$  states. RF pulses are marked with a square on the horizontal axis, while echoes are marked with a disk. At each step, a new RF pulse is applied and each magnetization component is divided into three pathways:  $\tilde{F}_{+k}$  (solid black lines),  $\tilde{F}_{-k}^*$  (purple lines) and  $\tilde{Z}$  (dashed lines). To calculate the signal measured at each echo, the contributions of each line towards the corresponding disk marker are summed. The slope of the lines is proportional to the gradient strength applied along  $z$  including  $B_0$  inhomogeneity effects [6].

## 2.2 Deep Learning

Deep learning is a form of machine learning that takes advantage of non-linear functions to learn complex patterns [2]. The deep learning model used in this work was the RIM, which is based on RNNs.

### 2.2.1 Recurrent Neural Networks and Gated Recurrent Units

RNNs are a class of neural networks designed to model sequence data, such as audio, text, or numerical time series [18]. RNNs have sequential memory achieved by a looping mechanism that allows previous information from a previous step to flow to the next step. This concept of “memory” allows the RNN to generate an input considering information of previous inputs. When training an RNN, the forward pass is unfolded in time and the weight parameters are adjusted through backpropagation, but where each layer shares the same weight parameter across nodes (figure 2.9).

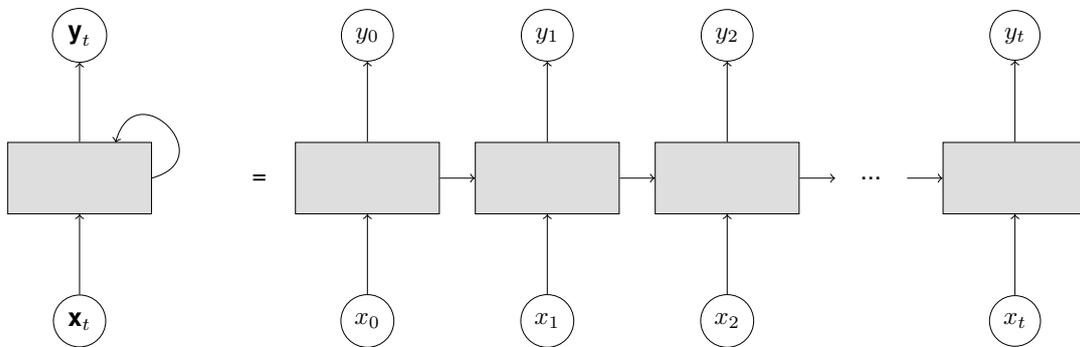


Figure 2.9: General architecture of an RNN. The input  $\mathbf{x}_t$  contains sequential information that is fed into the network, which uses a looping mechanism to calculate the output  $\mathbf{y}_t$ . This is equivalent to an unfolded representation, where the individual layers are drawn, each mapped to an element  $x_i$  of the input.

Due to their configuration, RNNs suffer from short-term memory, caused by vanishing gradients during backpropagation, and thus have difficulty learning earlier time-steps. Gated recurrent units (GRUs) are a specialized neural network created to mitigate the short-term memory issue of RNNs [19]. GRU contain a reset gate and an update gate that take advantage of sigmoid layers to allow the network to learn what data can be forgotten, and what data needs to be memorized.

### 2.2.2 Recurrent Inference Machines

RIMs are a class of neural networks first proposed by Putzky and Welling [20] which employ an RNN architecture in conjunction with a signal model  $\mathbf{m} = M(\mathbf{p}) + \varepsilon$  to learn a parameter inference method, instead of a standard direct mapping between input and estimates. RIMs were developed to solve inverse problems, i.e. finding the inverse transformation of a forward model that transforms variables  $\mathbf{p}$  into measurements  $\mathbf{m}$ , with  $\varepsilon$  the noise associated with the measurements.

The main distinctive characteristic of the RIM is that its neural network portion attempts to maximize the MAP to reconstruct the signal in an iterative procedure, while implicitly learning prior information that

typically has to be hand-chosen in other networks [20].

Let the measurements  $\mathbf{m}$  be described by a distribution with a probability density function  $f(\mathbf{m}|\mathbf{p})$ , from which the negative log-likelihood  $\mathcal{L}(\mathbf{m}|\mathbf{p})$  can be defined. Inverse problems are typically solved by optimizing the MAP in equation 2.19, where  $\mathcal{L}(\mathbf{p})$  is negative the log-prior probability. This can be accomplished by performing gradient descent (GD) (equation 2.20), where  $\alpha_j$  is the learning rate.

$$\hat{\mathbf{p}} = \arg \min_{\mathbf{p}} (\mathcal{L}(\mathbf{m}|\mathbf{p}) + \mathcal{L}(\mathbf{p})) \quad (2.19)$$

$$\mathbf{p}_{j+1} = \mathbf{p}_j + \alpha_j (\nabla_{\mathbf{p}} \mathcal{L}(\mathbf{m}|\mathbf{p}) + \nabla_{\mathbf{p}} \mathcal{L}(\mathbf{p})) \quad (2.20)$$

Since the likelihood is calculated separately from the prior, a model trained with a set of measurements  $\mathbf{m}$  can be applied to different sets of measurements successfully. However, the success of the optimization heavily depends on the assumptions made about the model, namely the prior distribution.

RIMs tackle this setback by integrating the prior into the model and jointly learning the prior distribution of the parameters and the inference model [4]. At a given time-step  $j \in \{1, \dots, J\}$ , the gradient of the negative log-likelihood  $\nabla_{\mathbf{p}} \mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j)$  and the current estimate  $\hat{\mathbf{p}}_j$  are passed as inputs of an RNN block, as well as a vector of memory states  $\mathbf{h}_j$ . The RNN block then yields an incremental step  $\Delta \hat{\mathbf{p}}_j$  and the new vector of memory states  $\mathbf{h}_{j+1}$  as output. The update equations for this framework are written in equations 2.21:

$$\{\Delta \hat{\mathbf{p}}_j, \mathbf{h}_{j+1}\} = \text{RNN}(\hat{\mathbf{p}}_j, \nabla_{\mathbf{p}} \mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j), \mathbf{h}_j) \quad (2.21a)$$

$$\hat{\mathbf{p}}_{j+1} = \hat{\mathbf{p}}_j + \Delta \hat{\mathbf{p}}_j \quad (2.21b)$$

The RNN portion is responsible for evaluating what the measurements  $\hat{\mathbf{m}}$  would be if the estimate  $\hat{\mathbf{p}}_j$  contained the real model parameters, and how much these hypothetical measurements differ from the real ones.

The total loss  $\Lambda^{\text{total}}$  to minimize is given by the weighted sum of the loss function  $\Lambda(\mathbf{p}, \hat{\mathbf{p}}_j)$  over all time-steps (equation 2.22).  $w_\tau$  is a weight parameter which determines the emphasis put on each reconstruction  $j$  relative to the other estimates.

$$\Lambda^{\text{total}} = \frac{1}{J} \sum_{j=1}^J w_j \Lambda(\mathbf{p}, \hat{\mathbf{p}}_j) \quad (2.22)$$

Figure 2.10 displays a diagram of the general configuration of a RIM. Seeing that RIMs attempt to learn the inference process of reconstructing a given signal and use internal states as part of their recurrent architecture, they have a low tendency to overfit and a high capacity to generalize to unseen types of data [4].

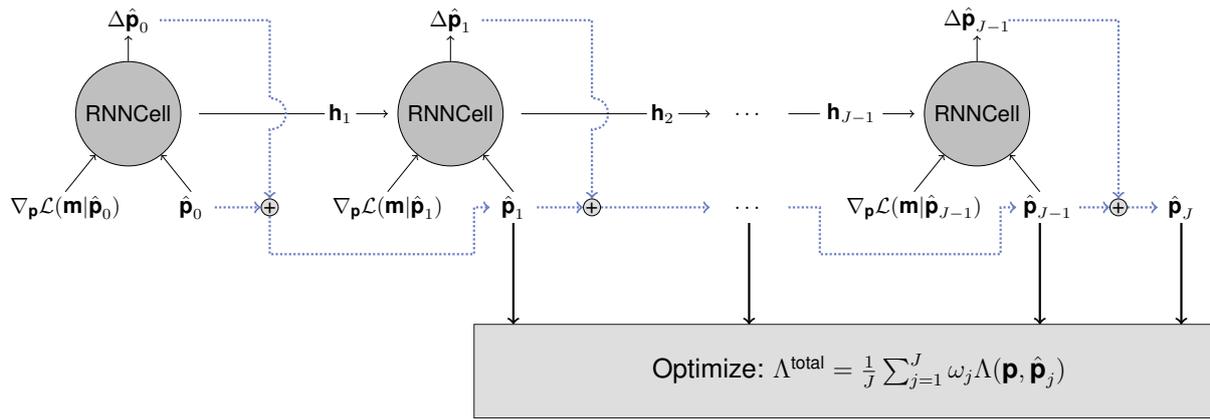


Figure 2.10: General configuration of the RIM. The RIM performs  $J$  inference steps to obtain  $J$  estimates of the variables  $\mathbf{p}$ , taking the previous estimate  $\hat{\mathbf{p}}_{j-1}$ , the gradient of the negative log-likelihood  $\nabla_{\mathbf{p}} \mathcal{L}(\mathbf{m} | \hat{\mathbf{p}}_{j-1})$ , and a vector of memory states  $\mathbf{h}_j$ .



# Chapter 3

## Implementation

The present chapter describes the implementation of the MR signal models considered, the structure of the dictionaries containing the EPG signal evolution, the configuration of the proposed networks and the training and testing frameworks. Network models were implemented with PyTorch 1.10.2, trained on a NVIDIA TESLA P100 GPU, and tested on an Intel Core i7 2.8 GHz CPU.

### 3.1 Magnetic Resonance Signal Models

The performance and accuracy of the proposed networks were assessed for three progressively more complex MR signal models: the exponential model  $M_{EXP}$  (as defined in equation 2.5), the EPG model  $M_{EPG(T_2)}$  (as described in section 2.1.1) and the EPG model considering also the influence of the  $B_1$  field  $M_{EPG(T_2, B_1)}$ . For the first two models, the networks attempted to estimate the PD and  $T_2$  maps; for the latter, the networks aimed to additionally estimate the  $B_1$  maps. It should be noted that the PD maps estimated with the EPG models  $M_{EPG(T_2)}$  and  $M_{EPG(T_2, B_1)}$  do not correspond to the physical definition of proton density values in the tissues, but are instead included in the signal model as a simple normalization constant and henceforth named “PD maps” for convenience.

The EPG forward model was implemented by defining a matrix of discrete configuration states  $\Omega$  where each column contains each configuration state  $\tilde{F}_+(\mathbf{k})$ ,  $\tilde{F}_-(\mathbf{k})$  and  $\tilde{Z}(\mathbf{k})$ . The EPG signal measured after the application of an ETL number of RF pulses can be obtained by recursively multiplying the matrix  $\Omega$  with the relaxation matrix  $\mathbf{E}$ , the shifting operator  $\mathbf{S}$  and the mixing matrix  $\mathbf{T}_\Phi$ , in intervals equal to the ESP of the ME-SE pulse sequence, as defined in section 2.1.1.

The partial derivatives of the EPG signal with respect to PD,  $T_2$  and  $B_1$  were hand-coded following the work of Layton et al. [21]: they can be calculated recursively in conjunction with the EPG signal by applying the product rule and defining the corresponding matrices  $\partial\mathbf{T}_\Phi/\partial B_1$  and  $\partial\mathbf{E}/\partial T_2$ .

Algorithm 1 illustrates the calculation of the EPG signal evolution and the corresponding partial derivatives, used in both  $M_{EPG(T_2)}$  and  $M_{EPG(T_2, B_1)}$  models, where  $\times$  denotes matrix multiplication. In the former signal model,  $B_1$  was fixed as 1, while in the latter  $B_1$  corresponded to the effective  $B_1$  value observed in the pixel in question. This algorithm was applied to all image pixels simultaneously by cod-

ing an additional third dimension corresponding to the pixel dimension into all matrices and vectors. This vectorization is crucial to greatly speed up the calculations in lieu of looping through each pixel. Figure 3.1 displays the EPD of our protocol, which offers a complementary understanding of the EPG algorithm.

---

**Algorithm 1** Extended Phase Graph forward model and partial derivatives

---

- 1:  $\alpha_{\text{exc}} = \alpha_{\text{exc}} \times B_1$  ▷ Flip angle of the excitation pulse
  - 2: Initialize matrix of configuration states  $\Omega$
  - 3:  $\Omega_{\{\mathbf{z}, k=0\}} = 1$  ▷  $\tilde{Z}(0) = 1$ , equilibrium condition
  - 4: Calculate  $\mathbf{T}_{\Phi}(|\alpha_{\text{exc}}|, \angle\alpha_{\text{exc}})$  ▷ Mixing matrix
  - 5:  $\Omega = \mathbf{T}_{\Phi}(|\alpha_{\text{exc}}|, \angle\alpha_{\text{exc}}) \times \Omega$  ▷ Apply excitation pulse
  
  - 6: Calculate  $\partial\mathbf{T}_{\Phi}(|\alpha_{\text{exc}}|, \angle\alpha_{\text{exc}})/\partial B_1$  ▷ Partial derivative of mixing matrix
  - 7:  $\partial\Omega/\partial B_1 = \partial\mathbf{T}_{\Phi}(|\alpha_{\text{exc}}|, \angle\alpha_{\text{exc}})/\partial B_1 \times \Omega$
  
  - 8: Calculate  $\mathbf{S}$  ▷ Shift operator
  - 9: Calculate  $\mathbf{E}(T_1, T_2, \text{ESP}/2)$  ▷ Relaxation matrix
  - 10: Calculate  $\partial\mathbf{E}(T_1, T_2, \text{ESP}/2)/\partial T_2$  ▷ Partial derivative of relaxation matrix
  
  - 11: **for** ech = {1, ..., ETL} **do**
  
  - 12:  $\Omega = \mathbf{E}(T_1, T_2, \text{ESP}/2) \times \Omega$
  - 13:  $\Omega = \mathbf{S} \times \Omega$
  
  - 14:  $\partial\Omega/\partial T_2 = \mathbf{S} \times \partial\mathbf{E}(T_1, T_2, \text{ESP}/2)/\partial T_2 \times \Omega + \mathbf{S} \times \mathbf{E} \times \partial\Omega/\partial T_2$
  
  - 15:  $\alpha_{\text{refoc}} = \alpha_{\text{refoc}} \times B_1$  ▷ Flip angle of the refocusing pulse
  - 16: Calculate  $\mathbf{T}_{\Phi}(|\alpha_{\text{refoc}}|, \angle\alpha_{\text{refoc}})$  ▷ Mixing matrix
  - 17:  $\Omega = \mathbf{T}_{\Phi}(|\alpha_{\text{refoc}}|, \angle\alpha_{\text{refoc}}) \times \Omega$  ▷ Apply refocusing pulse
  
  - 18: Calculate  $\partial\mathbf{T}_{\Phi}(|\alpha_{\text{refoc}}|, \angle\alpha_{\text{refoc}})/\partial B_1$  ▷ Partial derivative of mixing matrix
  - 19:  $\partial\Omega/\partial B_1 = \partial\mathbf{T}_{\Phi}(|\alpha_{\text{refoc}}|, \angle\alpha_{\text{refoc}})/\partial B_1 \times \Omega + \mathbf{T}_{\Phi}(|\alpha_{\text{refoc}}|, \angle\alpha_{\text{refoc}}) \times \partial\Omega/\partial B_1$
  
  - 20:  $\Omega = \mathbf{E}(T_1, T_2, \text{ESP}/2) \times \Omega$  ▷ Apply relaxation
  - 21:  $\Omega = \mathbf{S} \times \Omega$
  
  - 22:  $\partial\Omega/\partial T_2 = \mathbf{S} \times \partial\mathbf{E}(T_1, T_2, \text{ESP}/2)/\partial T_2 \times \Omega + \mathbf{S} \times \mathbf{E} \times \partial\Omega/\partial T_2$
  
  - 23: **end for**
  
  - 24: EPG signal =  $\Omega_{\{\mathbf{Fp}, k=0\}}$  ▷  $\tilde{F}(0)$
  - 25:  $\partial\text{EPG signal}/\partial\text{PD} = \text{EPG signal}$
  - 26: EPG signal = PD  $\times$  EPG signal ▷ Normalize signal
  - 27: **return** EPG signal,  
 $\partial\text{EPG signal}/\partial\text{PD},$   
 $\partial\text{EPG signal}/\partial T_2,$   
 $\partial\text{EPG signal}/\partial B_1$
-

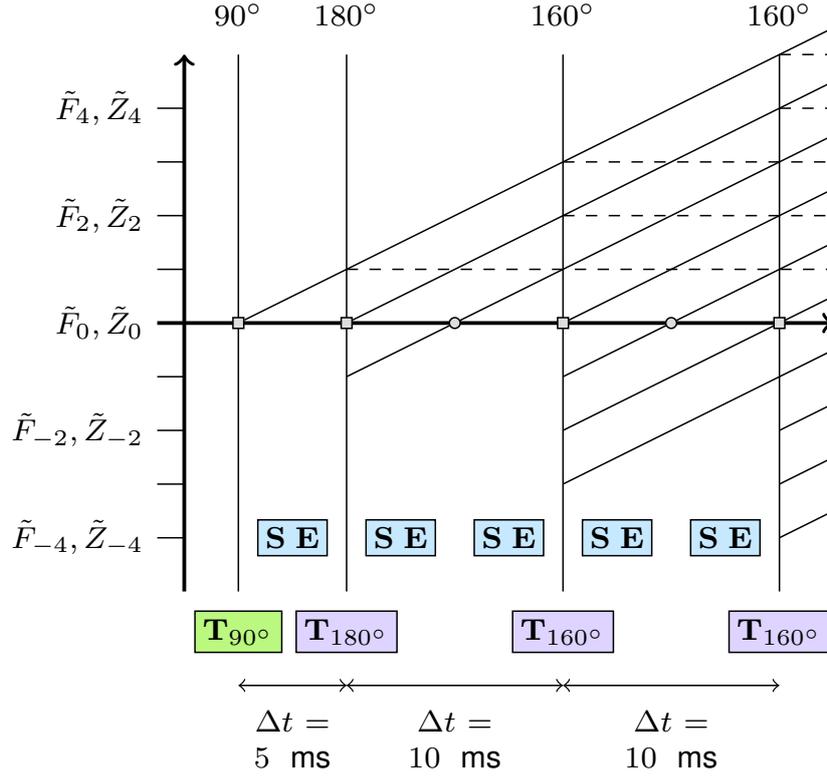


Figure 3.1: Extended phase diagram (EPD) of the ME-SE sequence used in this work, truncated after 25 ms. The initial  $90^\circ$  RF pulse excites the system at  $t = 0$  s, corresponding to the application of matrix  $\mathbf{T}_\phi$ . A first  $180^\circ$  refocusing pulse is applied after 5 ms, registered in the application of a new  $\mathbf{T}_\phi$ . The following refocusing pulses are applied in intervals of ESP = 10 ms, with a flip angle of  $160^\circ$ . Matrices  $\mathbf{E}$  and  $\mathbf{S}$  are calculated for half of the ESP (5 ms), so they should be applied every 5 ms to update the matrix of configuration states. Echoes are measured in-between refocusing pulses, as is expected in ME-SE sequences. The weighted set of images  $\mathbf{m}$  is obtained by collecting  $\tilde{F}_0$  from each echo.

## 3.2 Signal and Gradient Dictionaries

A set of EPG signal evolution dictionaries were created by calculating the forward signal model and its partial derivatives for all combinations of a range of  $T_2$  and  $B_1$  values. Figure 3.2 shows a diagram depicting the structure of the dictionaries. Each dictionary entry contains the evolution of the MR signal for a given  $T_2$  and  $B_1$ , as well as the signal's partial derivatives with respect to  $T_2$  and  $B_1$ .

The state-of-the-art dictionary matching technique was implemented by looping through each pixel  $(x, y)$  on the weighted images  $\hat{\mathbf{m}}$  and assigning it the parametric map  $\mathbf{p}$  whose signal in the dictionary  $\mathbf{m}_{\text{dict}}(\mathbf{p})$  maximizes its inner product with that pixel's signal evolution  $\hat{\mathbf{m}}(x, y)$  (equation 3.1).

$$\hat{\mathbf{p}}(x, y) = \arg \max_{\mathbf{p}} (\hat{\mathbf{m}}(x, y) \cdot \mathbf{m}_{\text{dict}}(\mathbf{p})) \quad (3.1)$$

		$B_1$		0.6				0.61				...		
		$T_2$		0	1	...	1999	2000	0	1	...	1999	2000	...
Echoes	1	o	o	...	o	o	o	o	o	o	...	o	o	...
	2	o	o	...	o	o	o	o	o	o	...	o	o	...
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	31	o	o	...	o	o	o	o	o	o	...	o	o	...
	32	o	o	...	o	o	o	o	o	o	...	o	o	...

Figure 3.2: Diagram of the EPG dictionaries. A dictionary contains all the combinations of  $T_2$  and  $B_1$  values with a selected range and precision. Each dictionary entry, marked in this diagram with a circle, corresponds to the MR signal calculated with the EPG framework at each measured echo. Alternatively, each entry can contain the partial derivative of the EPG signal with respect to  $T_2$  or  $B_1$ .

### 3.3 Training, Validation and Testing Data

The networks were fully trained with simulated data created from BrainWeb’s discrete anatomical models <sup>1</sup>. BrainWeb [22–25] is a simulated brain database that contains twenty three-dimensional (3D) anatomical brain models with 362 slices and matrix dimensions (362, 434), where each voxel is labeled with an integer representing the tissue that contributes the most to that voxel (0=background, 1=cerebrospinal fluid (CSF), 2=grey matter, 3=white matter, 4=fat, 5=muscle, 6=muscle/skin, 7=skull, 8=vessels, 9=connective tissue, 10=dura matter, 11=bone marrow). The anatomical models were used to simulate pairs of ground-truth maps  $\mathbf{p}$  and weighted images  $\mathbf{m}$  that constituted the training data. At each new training epoch, each anatomical model in the training set was randomly split into a minibatch, which was used as the network’s ground-truth. The total number of training images can thus be calculated as the number of epochs multiplied by the size of the minibatch, multiplied by the number of anatomical models in the training dataset. The procedure for generating the ground-truth simulated weighted images followed the methodology described in Sabidussi et al. [4], outlined in detail in this section and illustrated schematically in figure 3.3.

Firstly, each 3D anatomical model was randomly split into a chosen number of two-dimensional patches of  $40 \times 40$  pixels each, whose centers were uniformly drawn from the model’s brain mask. Ground-truth values for PD and  $T_2$ ,  $\text{PD}^{\text{GT}}$  and  $T_2^{\text{GT}}$  respectively, were attributed to each tissue in a patch, following a normal distribution with mean  $\mu_{\text{tissue}}^{\text{GT}}$  according to the literature tissue properties in table 3.1, and standard deviation  $\sigma^{\text{GT}} = 0.3$ . This distribution was applied to guarantee the presence of variability between the ground-truth values of each tissue across different patches, in order to simulate inter-subject variability of  $T_2$  and PD values. An additional normal distribution  $\mathcal{N}(\mu = 0, \sigma^2)$  was added to the resulting PD and  $T_2$  ground-truth maps, followed by convolution with a Gaussian smoothing kernel  $\mathcal{G}(\sigma = 0.4)$  to simulate intra-tissue variation within each patch.

<sup>1</sup>[https://brainweb.bic.mni.mcgill.ca/anatomic\\_normal\\_20.html](https://brainweb.bic.mni.mcgill.ca/anatomic_normal_20.html)

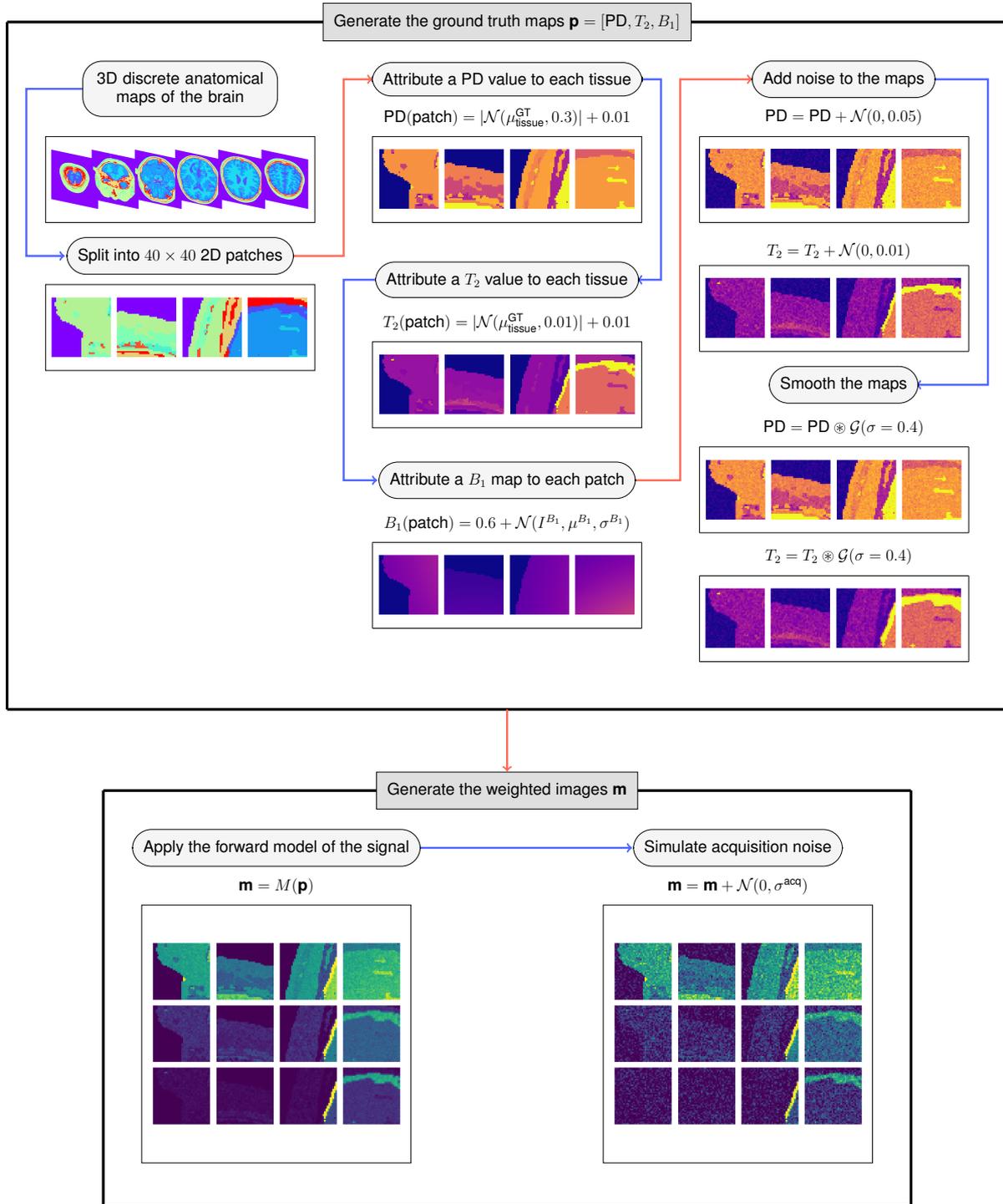


Figure 3.3: Protocol for generating simulated pairs of ground-truth weighted images  $\mathbf{m}$  and parametric maps  $\mathbf{p}$ . Three-dimensional discrete anatomical maps are split into a chosen number of two-dimensional patches, and ground-truth parameters PD and  $T_2$  (in seconds) are attributed to each tissue in a patch, following a normal distribution to simulate inter-subject variability. Gaussian  $B_1$  maps are simulated separately for each patch. In this diagram,  $B_1$  maps were masked from the background region, but this is representative only: in the following chapters, we will specify the values set on the background regions for these maps.  $T_2$  and PD maps are smoothed after including noise addition to simulate intra-tissue variability. The forward signal model is then applied to the ground-truth maps to obtain a set of echoes for each patch. This diagram illustrates the process for 4 patches, represented horizontally, and 3 echoes, represented vertically.  $B_1$  maps are only estimated when using the signal model  $M_{\text{EPG}}(T_2, B_1)$ .

Table 3.1: Mean ground-truth PD and  $T_2$  values per tissue,  $\mu_{\text{tissue}}^{\text{GT}}$ , used to simulate realistic MRI data [4].

	PD	$T_2$ [ms]
CSF	1	2 000
Grey matter	0.85	110
White matter	0.65	80
Fat	0.9	70
Muscle	0.7	50
Muscle skin	0.7	50
Skull	0.9	30
Vessels	1.0	275
Marrow	0.8	50
Connective tissue	0.7	80
Dura matter	0.7	70

The ground-truth  $B_1$  maps were randomly created from a Gaussian distribution as defined in equation 3.2, where each variable was obtained from a uniform distribution  $\mathcal{U}$ .  $I^{B_1}$  represents the  $B_1$  map's maximum intensity, and was drawn from  $\mathcal{U}(0.5, 0.8)$ ;  $\sigma^{B_1}$  is the distribution's standard deviation, drawn from  $\mathcal{U}(0.2, 0.3)$  as a percentage of the image smallest dimension;  $\mu^{B_1}$  determines the position of the Gaussian in the image and it was drawn from  $\mathcal{U}(0.4, 0.6)$  as a percentage of the image dimension. A new  $B_1$  map of the full 2D slice was generated for each patch and then cropped according to the patch's coordinates in the full anatomical model. The chosen distributions assure that all  $B_1$  values are limited to the range  $[0.6, 1.4]$ , a realistic range for this variable [26]. Figure 3.4 showcases the effect of varying each parameter  $I^{B_1}$ ,  $\sigma^{B_1}$  and  $\mu^{B_1}$  on the  $B_1$  maps produced.

$$B_1(x, y) = 0.6 + I^{B_1} \exp\left(-\frac{(x - \mu^{B_1})^2}{2(\sigma^{B_1})^2}\right) \exp\left(-\frac{(y - \mu^{B_1})^2}{2(\sigma^{B_1})^2}\right) \quad (3.2)$$

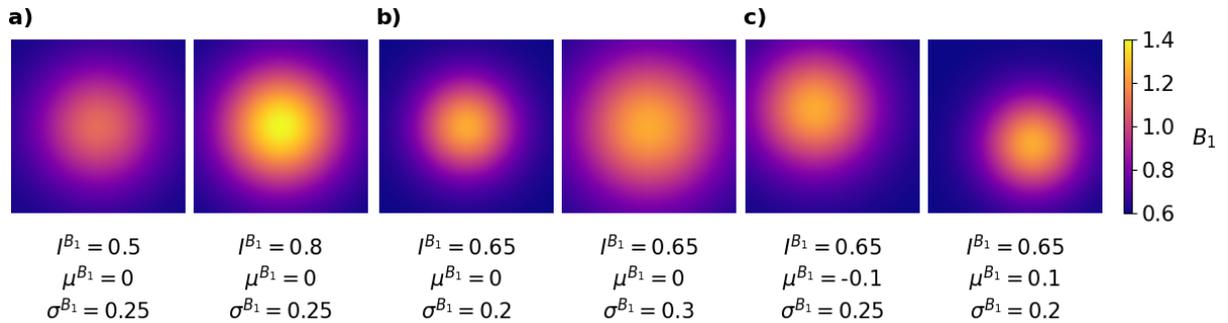


Figure 3.4: Simulated  $B_1$  maps of a full  $160 \times 160$  slice for the lower and upper bounds of  $I^{B_1}$ ,  $\sigma^{B_1}$  and  $\mu^{B_1}$ . **a)**  $I^{B_1}$  controls the maximum  $B_1$  value achieved in the map. The lower bound,  $I^{B_1}=0.5$ , designs maps with values in the interval  $[0.6, 1.1]$ , while the upper bound,  $I^{B_1}=0.8$ , designs maps with values in  $[0.6, 1.4]$ . **b)**  $\sigma^{B_1}$  controls the spread of the Gaussian distribution, and it is defined as a percentage of the image dimension. **c)**  $\mu^{B_1}$  controls the position of the Gaussian in the image. The lower bound,  $\mu^{B_1}$ , places the Gaussian 10% northwest from the image center, whereas the upper bound,  $\mu^{B_1}$ , places it 10% southeast from the image center.

Secondly, the resulting ground-truth maps  $\mathbf{p} = [\text{PD}, T_2, B_1]$  were used as the input of the forward signal model  $M$ , outputting a set of ground-truth weighted images  $\mathbf{m}$ . All ground-truth weighted images will have intensity values ranging from 0 to 1.3, the maximum value the PD variable can take. Finally, a normal distribution  $\mathcal{N}(0, \sigma^{\text{acq}})$  was added to the weighted images to simulate the presence of acquisition noise. The standard deviation of this distribution was drawn from a log-normal distribution with underlying mean  $\mu = 0$  and standard deviation  $\sigma = 1$ , scaled by a factor  $l^{\text{acq}}$  that determines the distribution's width (equation 3.3), to ensure that the network was trained with minibatches with varying levels of noise.

$$\sigma^{\text{acq}} \in \left\{ \frac{1}{l^{\text{acq}}} \log \mathcal{N}(\mu = 0, \sigma = 1) \right\} = \left\{ \frac{1}{\sigma^{\text{acq}} \cdot \sigma \cdot l^{\text{acq}} \cdot \sqrt{2\pi}} \exp \left( -\frac{(\ln \sigma^{\text{acq}})^2}{2\sigma^2} \right) \right\} \quad (3.3)$$

All networks were trained with a total of 72 000 patches, and 10 distinct anatomical models. Validation images were simulated following the same procedure, but considering a single subject, manually-selected full two-dimensional slices instead of random patches, a user-defined  $\sigma^{\text{acq}}$ , and  $B_1$  maps created with  $I^{B_1} = 0.65$ ,  $\sigma^{B_1}=25$  and  $\mu^{B_1}=0$ . Testing datasets were created in a similar fashion to the validation datasets, pertaining to a different subject. The training, validation and testing simulated datasets specifications are included in table 3.2. Note that each simulated dataset consists of three distinct sub-datasets: one where the images were simulated considering the signal model  $M_{\text{EXP}}$ , another considering  $M_{\text{EPG}(T_2)}$  and a third considering  $M_{\text{EPG}(T_2, B_1)}$ .

Table 3.2: Anatomical model identifier, slices and standard deviation of the acquisition noise of the simulated datasets created for training, validation and testing. Each dataset contains three sub-datasets, one for each signal model studied.

	Subjects' ID	Slices	Acquisition Noise
Training	4, 5, 6, 18, 20, 38 41, 42, 43 and 44	Random	(Depends on model)
Validation (noiseless)	45	100, 120, 140, 160, 180, 200, 220, 240, 260, 280	None
Validation (noisy)	45	100, 120, 140, 160, 180, 200, 220, 240, 260, 280	$\sigma^{\text{acq}} = 0.001$
Validation (very noisy)	45	100, 120, 140, 160, 180, 200, 220, 240, 260, 280	$\sigma^{\text{acq}} = 0.05$
Testing (noiseless)	50	180	None
Testing (noisy)	50	180	$\sigma^{\text{acq}} = 0.001$
Testing (very noisy)	50	180	$\sigma^{\text{acq}} = 0.05$

All weighted images were simulated following a ME-SE acquisition protocol with an ETL of 32 echoes, an ESP of 10 ms ( $T_2 = 10:10:320$  ms) and flip angles  $[180^\circ, 160^\circ, \dots, 160^\circ]$ . In addition to the simulated testing images, a testing set of *in vivo* brain data of a healthy subject previously acquired in a Philips Achieva 3T scanner, using the same imaging protocol, with spatial resolution of  $1.6 \times 1.6 \times 4.0$  mm<sup>2</sup>, field-of-view  $250 \times 250$  mm<sup>2</sup>, and TR of 4 s was considered. Networks were evaluated on both the simulated

testing datasets and on four *in vivo* datasets.

### 3.4 Network

Our RIM configuration was adapted from Sabidussi et al. [4]. At each optimization step, a new ground-truth parametric map  $\mathbf{p}$ , which can be the  $T_2$  map for instance, is simulated and the network performs  $J$  inference steps to obtain  $J$  estimates  $\hat{\mathbf{p}}_j$ , where each  $\hat{\mathbf{p}}_{j+1}$  is more accurate than the previous  $\hat{\mathbf{p}}_j$ . We used the mean squared error (MSE) as a loss function to evaluate each estimate  $\hat{\mathbf{p}}_j$  against the true map  $\mathbf{p}$  (equation 3.4). The total loss to minimize, defined in equation 3.5, corresponds to the weighted sum of the MSE obtained between  $\mathbf{p}$  and each  $\hat{\mathbf{p}}_j$ , where all weights  $w_j$  were set to 1.

$$\text{MSE}(\mathbf{p}, \hat{\mathbf{p}}_j) = \sum_{\text{pixel}} (\mathbf{p}_{\text{pixel}} - \hat{\mathbf{p}}_{j \text{ pixel}})^2 \quad (3.4)$$

$$\Lambda^{\text{total}} = \frac{1}{J} \sum_{j=1}^J w_j \cdot \text{MSE}(\mathbf{p}, \hat{\mathbf{p}}_j) = \frac{1}{J} \sum_{j=1}^J \left( \sum_{\text{pixel}} (\mathbf{p}_{\text{pixel}} - \hat{\mathbf{p}}_{j \text{ pixel}})^2 \right) \quad (3.5)$$

At each inference step  $j$ , the network takes the current maps estimate  $\hat{\mathbf{p}}_j$  into the forward signal model  $M$  to calculate a set of weighted images  $\hat{\mathbf{m}}_j$ . Since the noise of the acquired signal  $\mathbf{m}$  can be well described by a Gaussian distribution when the images have signal-to-noise ratio (SNR) larger than three [4, 27], the negative log-likelihood  $\mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j)$  can be written as equation 3.6. This is demonstrated in Appendix A.

$$\mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j) = \sum_{\text{pixel}} (\mathbf{m}_{\text{pixel}} - M(\hat{\mathbf{p}}_{j \text{ pixel}}))^2 = \sum_{\text{pixel}} (\mathbf{m}_{\text{pixel}} - \hat{\mathbf{m}}_{j \text{ pixel}})^2 \quad (3.6)$$

The negative log-likelihood is used to obtain its partial derivatives with respect to each map in  $\hat{\mathbf{p}}_j$ ,  $\nabla_{\hat{\mathbf{p}}_j} \mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j)$ , in a posterior step. This gradient is then concatenated with the estimates  $\hat{\mathbf{p}}_j$ , and submitted to the RNN portion of the network, named RNNCell. In the experiments with the MR signal models  $M_{\text{EXP}}$  or  $M_{\text{EPG}(T_2)}$ , this results in four input channels: the PD map estimate, the  $T_2$  map estimate, the gradient of the likelihood function with respect to PD and the gradient of the likelihood function with respect to  $T_2$ . When the signal model is  $M_{\text{EPG}(T_2, B_1)}$ , there are two additional input channels: the  $B_1$  map estimate and the gradient of the likelihood function with respect to  $B_1$ . In addition, the RNNCell receives two hidden-state vectors  $\mathbf{h}_j^1$  and  $\mathbf{h}_j^2$  as inputs, and outputs the two new hidden-state vectors  $\mathbf{h}_{j+1}^1$  and  $\mathbf{h}_{j+1}^2$ , and the incremental update of the maps  $\Delta \hat{\mathbf{p}}_j$  (figure 3.5).

The RNNCell block consists of a sequence of convolutional layers and GRU cells and is represented in figure 3.6. All convolutions are padded to retain the patch size throughout the network, and are followed by ReLu activation functions.

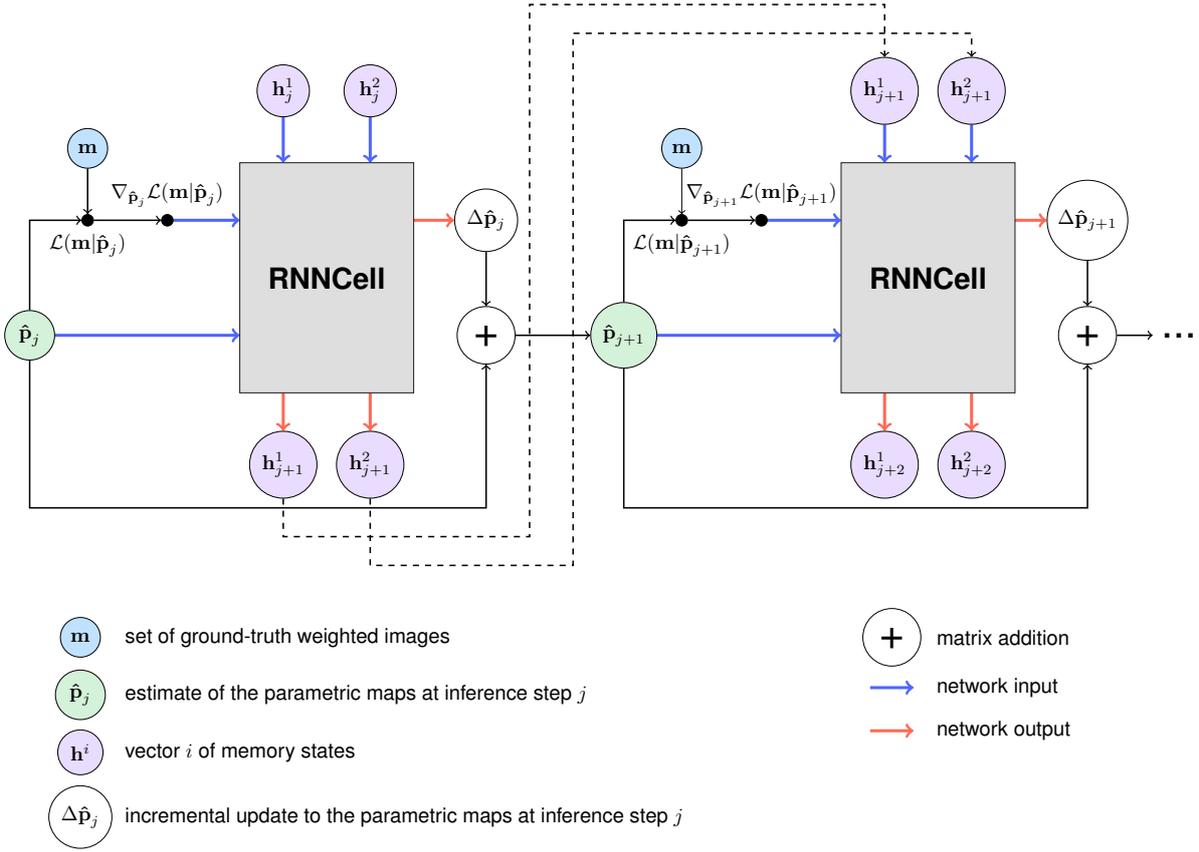


Figure 3.5: RIM configuration depicted for two inference steps. The current estimate  $\hat{\mathbf{p}}_j$  is both an input to the RNNCell and used to calculate the gradients  $\nabla_{\hat{\mathbf{p}}_j} \mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j)$ , alongside the simulated ground-truth weighted images  $\mathbf{m}$ . Two vectors of memory states  $\mathbf{h}_j^1$  and  $\mathbf{h}_j^2$  are present in this configuration. The RNNCell block outputs the incremental update of the maps  $\Delta \hat{\mathbf{p}}_j$ , which is summed to  $\hat{\mathbf{p}}_j$  to obtain the new estimate  $\hat{\mathbf{p}}_{j+1}$ . Estimates  $\hat{\mathbf{p}}_1, \dots, \hat{\mathbf{p}}_J$  are then used to perform the network optimization.

When evaluating the network,  $\mathbf{m}$  represents the weighted images reconstructed from the scanner data. The network initializes the parametric maps estimate  $\hat{\mathbf{p}}_0$  and performs  $J$  inference steps to obtain  $\hat{\mathbf{p}}_J$ , which corresponds to the estimate that the RIM outputs, while the intermediate estimates  $\hat{\mathbf{p}}_1, \dots, \hat{\mathbf{p}}_{J-1}$  are discarded.

All networks were trained using the Adam optimizer, and a learning-rate of 0.005. The  $T_2$  estimate maps were initialized with 1000 ms, the  $B_1$  maps with 1 and the the PD maps with the maximum value in the ground-truth PD maps.

In the original implementation by Sabidussi et al. [4], the partial derivatives  $\nabla_{\hat{\mathbf{p}}_j} \mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j)$  were calculated with Pytorch's automatic differentiation package. Since the RIM requires the calculation of the forward signal model and its partial derivatives  $J$  times during inference, it is expected that inference becomes a time-consuming process when the signal model is too complex. We defined a new framework, RIMFoGraD, which shares RIM's training scheme, but uses pre-defined dictionaries of the signal model and its partial derivatives to speed up the inference stage. With RIMFoGraD,  $\hat{\mathbf{m}}_j$  is obtained by simply taking the EPG signal evolution in the dictionary that was simulated with the  $\hat{\mathbf{p}}_j$  of each pixel; the gradient  $\nabla_{\hat{\mathbf{p}}_j} \mathcal{L}(\mathbf{m}|\hat{\mathbf{p}}_j)$  is obtained by taking the respective  $\nabla_{\hat{\mathbf{p}}_j} M(\hat{\mathbf{p}}_j)$  in the dictionary, and performing the remaining differentiation operations during inference.

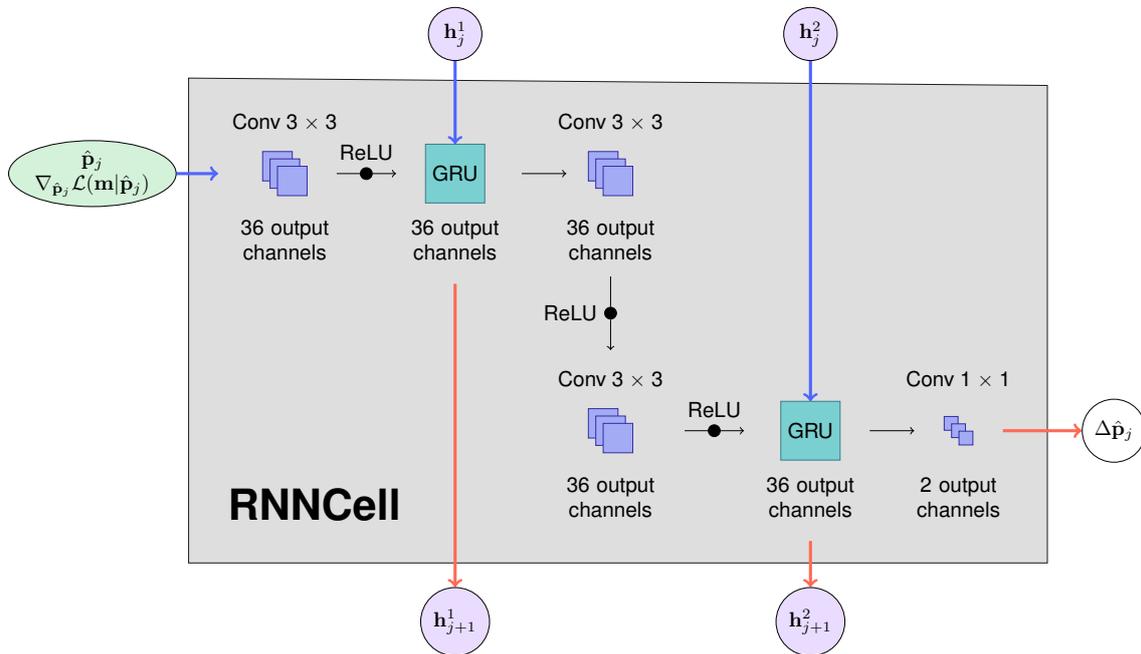


Figure 3.6: RNNCell configuration. The current parametric map estimate  $\hat{\mathbf{p}}_j$  is concatenated with the log-likelihood gradient  $\nabla_{\hat{\mathbf{p}}_j} \mathcal{L}(\mathbf{m} | \hat{\mathbf{p}}_j)$  along the channel dimension, resulting in 4 or 6 input channels depending on the model, and passed through the first convolutional layer. This layer has a kernel size of  $3 \times 3$  and produces 36 feature maps, which are activated with a ReLU function, and passed to a GRU cell. The GRU cell returns the first updated vector of memory states  $\mathbf{h}_{j+1}^1$  and the 36 feature maps, which go through two additional  $3 \times 3$  convolutional layers. The second GRU cell returns the second updated vector of memory states  $\mathbf{h}_{j+1}^2$  and the updated 36 feature maps, which go through a final convolutional layer with kernel size  $1 \times 1$ . This layer outputs 2 or 3 channels, depending on the model, which correspond to the incremental update to the parametric maps  $\Delta \hat{\mathbf{p}}_j$ .

# Chapter 4

## Results

This chapter presents an analysis of the accuracy and performance of the proposed frameworks.

Firstly, networks were optimized for a range of hyper-parameters. Networks of each signal model  $M_{\text{EXP}}$ ,  $M_{\text{EPG}(T_2)}$  and  $M_{\text{EPG}(T_2, B_1)}$  were trained with distinct hyper-parameters and compared with each other based on their performance on the validation datasets. As performance metrics, the normalized root mean square error (NRMSE), the peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) scores are reported. The NRMSE and PSNR scores quantify the pixel-by-pixel difference between images and are defined in equations 4.1 and 4.2, respectively. The PSNR score was defined by setting the maximum  $T_2$  value of the true maps  $\mathbf{p}$  as the peak signal. The SSIM score aims to quantify loss of structure in an image compared to a reference [28, 29]. It is defined in equation 4.3, where  $\mu_{\mathbf{p}}$  corresponds to the pixel mean in image  $\mathbf{p}$ ,  $\sigma_{\mathbf{p}}$  is the variance of image  $\mathbf{p}$ ,  $\sigma_{\mathbf{p}\hat{\mathbf{p}}}$  represents the covariance between the images  $\mathbf{p}$  and  $\hat{\mathbf{p}}$ , and  $c_1$  and  $c_2$  are constants that stabilize division and depend on the dynamic image range, here defined as the maximum  $T_2$  value observed in the reference images.

$$\text{NRMSE} = \sqrt{\frac{\sum (\mathbf{p} - \hat{\mathbf{p}})^2}{\sum \mathbf{p}^2}} \quad (4.1)$$

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{\max^2(\mathbf{p}) \cdot \#\text{pixels}}{\sum (\mathbf{p} - \hat{\mathbf{p}})^2} \right) \quad (4.2)$$

$$\text{SSIM} = \frac{(2 \cdot \mu_{\mathbf{p}} \cdot \mu_{\hat{\mathbf{p}}} + c_1) (2 \cdot \sigma_{\mathbf{p}\hat{\mathbf{p}}} + c_2)}{(\mu_{\mathbf{p}}^2 + \mu_{\hat{\mathbf{p}}}^2 + c_1) (\sigma_{\mathbf{p}}^2 + \sigma_{\hat{\mathbf{p}}}^2 + c_2)} \quad (4.3)$$

Since we aim to find the true  $T_2$  values of each tissue in the images, we gave precedence to the NRMSE and PSNR scores to validate the networks, while the SSIM score was used as a secondary selection metric. Performance scores focused on evaluating the  $T_2$  maps estimated within the brain parenchyma, and excluding the CSF region, since the long  $T_2$  relaxation of the CSF cannot be measured with the selected ME-SE protocol. For all validation tests, unless mentioned otherwise, NRMSE and PSNR scores were calculated separately on three distinct regions (white matter, grey matter and whole image excluding the CSF), while SSIM scores were calculated on the whole image, excluding the CSF

region. None of the scores took into account the background region. Figure 4.1 showcases the  $T_2$  maps estimated by two distinct networks over the same validation image and the corresponding NRMSE, PSNR and SSIM scores, highlighting the image features that each performance metric promotes.

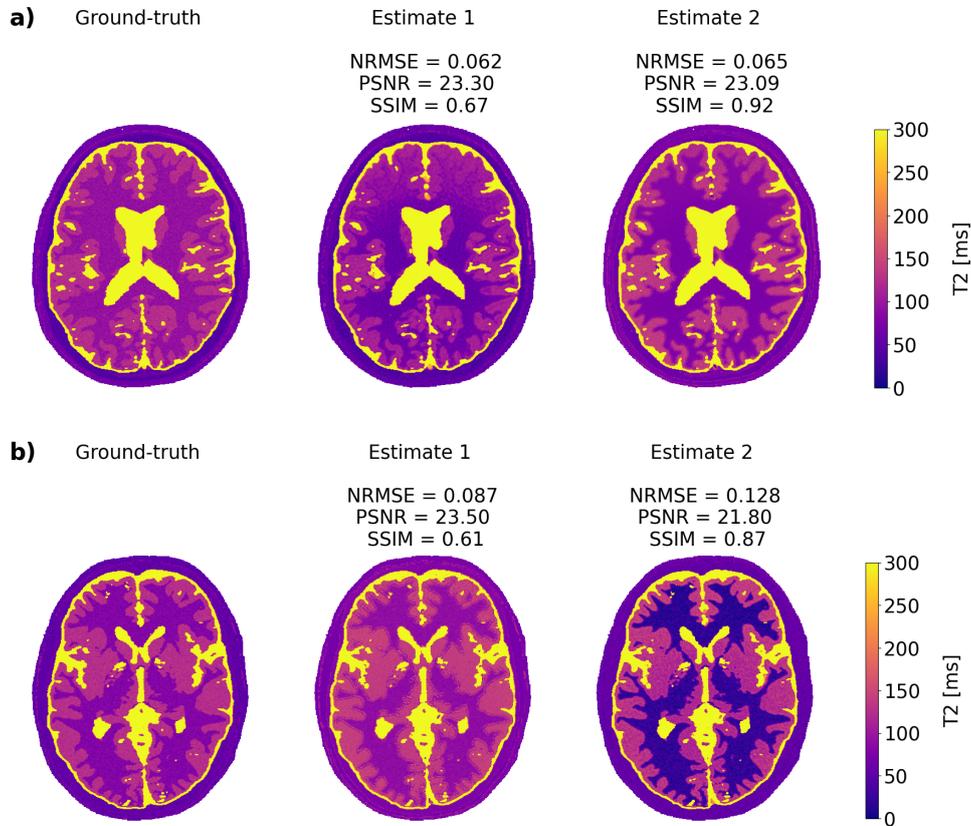


Figure 4.1:  $T_2$  maps estimated by two different networks on the same validation image, and respective NRMSE, PSNR and SSIM scores. NRMSE and PSNR scores were calculated on the image region excluding the CSF, while SSIM score was calculated on the whole image. **a)** Estimates 1 and 2 achieved similar NRMSE and PSNR scores, but Estimate 2 has a higher SSIM. In this case, the SSIM score is essential to select Estimate 2 over Estimate 1, where the boundaries between grey and white matter lost definition. **b)** Estimate 1 has a better NRMSE score, while Estimate 2 achieved a higher SSIM score. In this case, the  $T_2$  values of Estimate 1 in grey and white matter are closer to the reference, but because this estimate lost some structural definition, it would not be selected by the SSIM score.

Once the optimal hyper-parameters were identified, the selected networks of each signal model were evaluated on the testing datasets and assessed by visual inspection. RIM and RIMFoGraD share the same training process, so all the validation results are valid for both; furthermore, it will be verified that the same is true for the testing results. The validation and testing results are organized per signal model studied. Finally, a performance analysis was carried out to compare the time required to produce a parametric map for the state-of-the-art methods, the RIM, and the RIMFoGraD.

## 4.1 Monoexponential Model

The first step in this work consisted in verifying the performance of the RIM with the monoexponential signal model  $M_{EXP}$ , as developed by Sabidussi et al. [4]. The signal model was changed from a fast spin-

echo (FSE) sequence with 6 echoes to the acquisition protocol described in section 3.3. Networks with the exponential signal model took on average 1.5 h to train.

### 4.1.1 Validation

Networks with the  $M_{\text{EXP}}$  signal model were trained by varying the following hyper-parameters: size of the minibatch, number of inference steps, and standard deviation of the acquisition noise added to the training weighted images.

#### Size of the minibatch

Networks were trained with 12, 24, 48 and 72 patches in each minibatch. Figure 4.2 shows the NRMSE, PSNR and SSIM scores obtained on the estimated maps of the noisy validation dataset for networks trained with different minibatch sizes. All networks were trained with  $J=6$  inference steps, and variable standard deviation of the acquisition noise with  $l^{\text{acq}}=60$ .

All performance metrics display worse scores with the lowest number of patches. The NRMSE score is the best for the networks trained with 24 and 72 patches, while the PSNR is higher with 24 patches. We hypothesize that networks trained with 12 patches do not have enough information to perform optimization at each step, while networks trained with 24 patches achieve the best trade-off between enough information and small enough gradient variance to perform optimization. The SSIM score is similar across all hyper-parameters. Since the PSNR score favours 24 patches in the grey matter and brain parenchyma regions, we selected 24 patches for the final evaluation model.

#### Number of inference steps

Networks were trained using  $J = 2, 4, 6$  and 8 inference steps. The NRMSE, PSNR and SSIM scores obtained in the noisy validation dataset are represented in figure 4.3. The optimal number of inference steps was  $J=6$ , with the corresponding network achieving the best NRMSE and the PSNR scores in the grey matter region and the whole image excluding the CSF. Contrarily to what was suggested in the literature [16], increasing the number of inference steps does not necessarily guarantee a more accurate estimate of the overall map, and results in a more unstable estimation of the white matter  $T_2$  values.

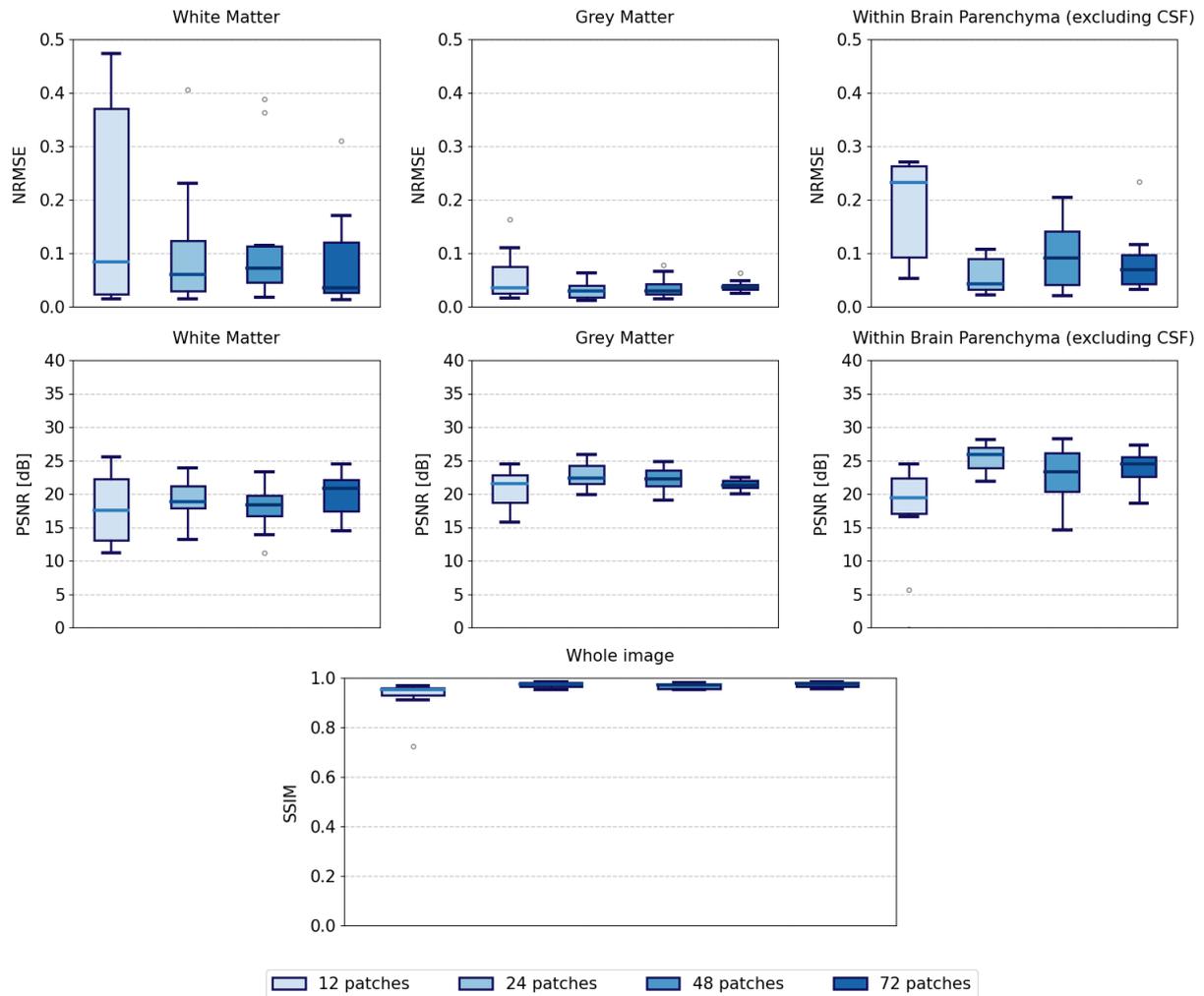


Figure 4.2: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noisy validation dataset by  $M_{\text{EXP}}$  networks trained with 12, 24, 48 and 72 patches. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and brain parenchyma excluding the CSF), whereas the SSIM score was calculated on the whole map. All networks were trained with  $J=6$  and  $l^{\text{acq}}=60$ .

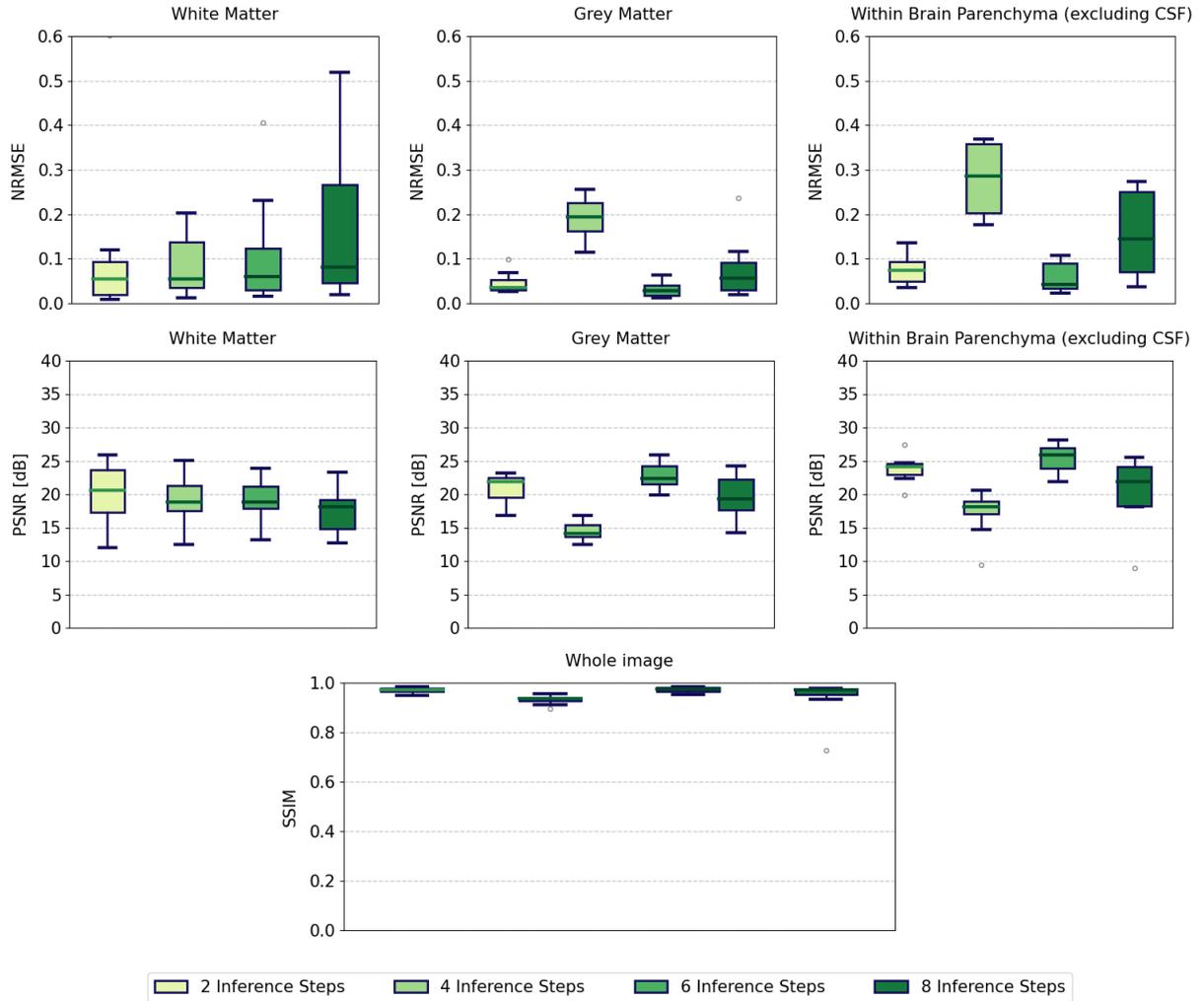


Figure 4.3: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noisy validation dataset by  $M_{\text{EXP}}$  networks trained with 2, 4, 6 and 8 inference steps. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the whole image. All networks were trained with 24 patches and  $l^{\text{acq}}=60$ .

## Acquisition noise of the training images

To evaluate the effect of the acquisition noise added to the training weighted images, we trained two groups of networks: a group with a fixed value for the standard deviation of the acquisition noise  $\sigma^{\text{acq}}$  for all minibatches, and another group with a variable  $\sigma^{\text{acq}}$  between each minibatch. Networks of the former group were trained with  $\sigma^{\text{acq}}=0.001, 0.005, 0.01$  and  $0.05$ . Networks of the latter group were trained with  $l^{\text{acq}}=30, 45, 60$  and  $100$ , to assess the influence of the variation of  $\sigma^{\text{acq}}$  between minibatches. All of these networks were evaluated on the noiseless, noisy and very noisy validation datasets in an attempt to select a model that is robust to a large range of noise intensities in the evaluation images. All networks were trained with 24 patches in each minibatch, and 6 inference steps.

Figure 4.4 shows the NRMSE, PSNR and SSIM scores obtained on the noiseless validation images. The first group of networks achieved the best scores on all performance metrics when training with the lowest  $\sigma^{\text{acq}}$ . The second group achieved the best NRMSE and PSNR scores with  $l^{\text{acq}}=60$ . On this validation dataset, both groups achieved similar NRMSE and PSNR scores.

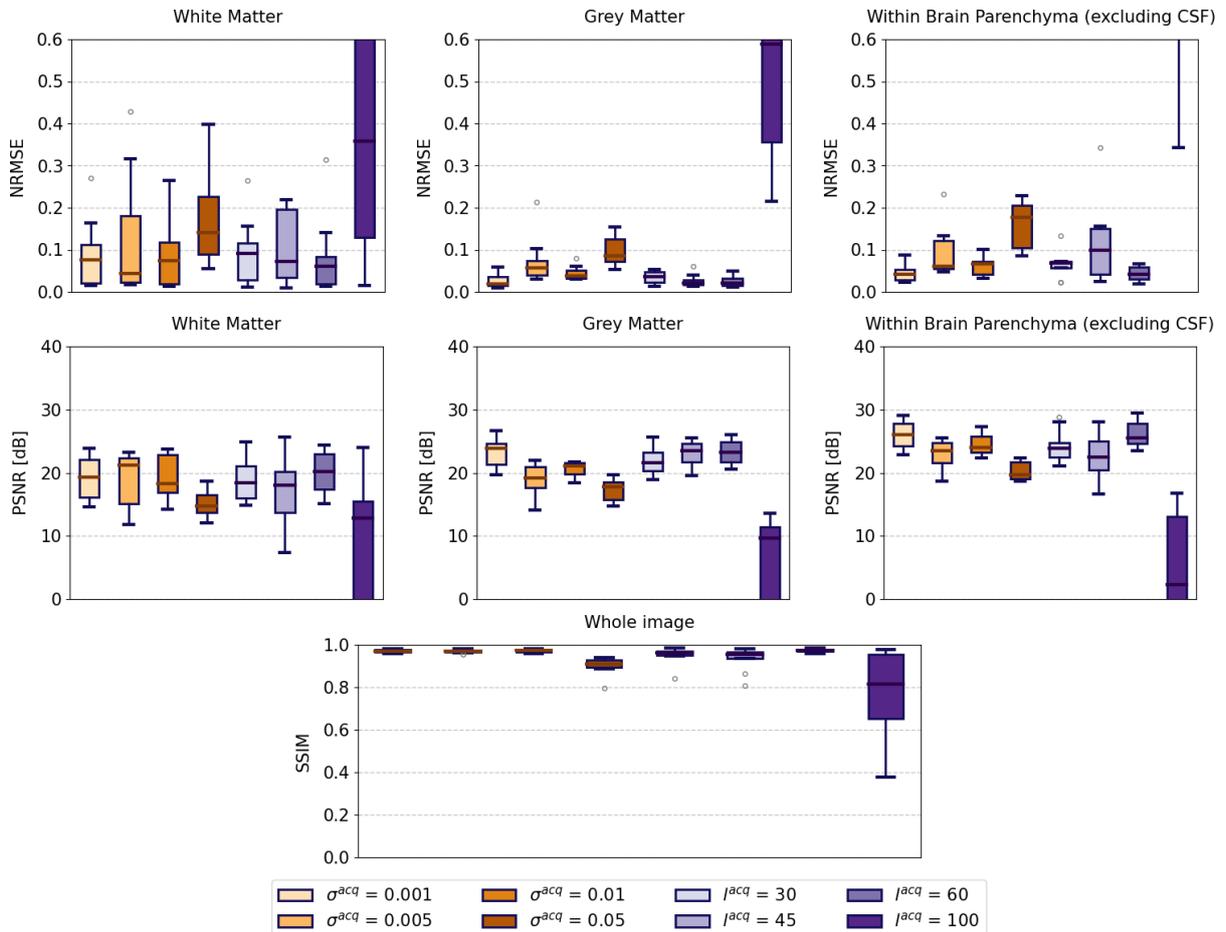


Figure 4.4: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noiseless validation dataset by  $M_{\text{EXP}}$  networks trained with different acquisition noise amounts. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with 24 patches and  $J=6$ .

Figure 4.5 shows the NRMSE, PSNR and SSIM scores obtained on the noisy validation images. The networks with  $\sigma^{\text{acq}}=0.001$  and  $l^{\text{acq}}=60$  remain the best-scoring models of their respective groups in terms of NRMSE and PSNR.

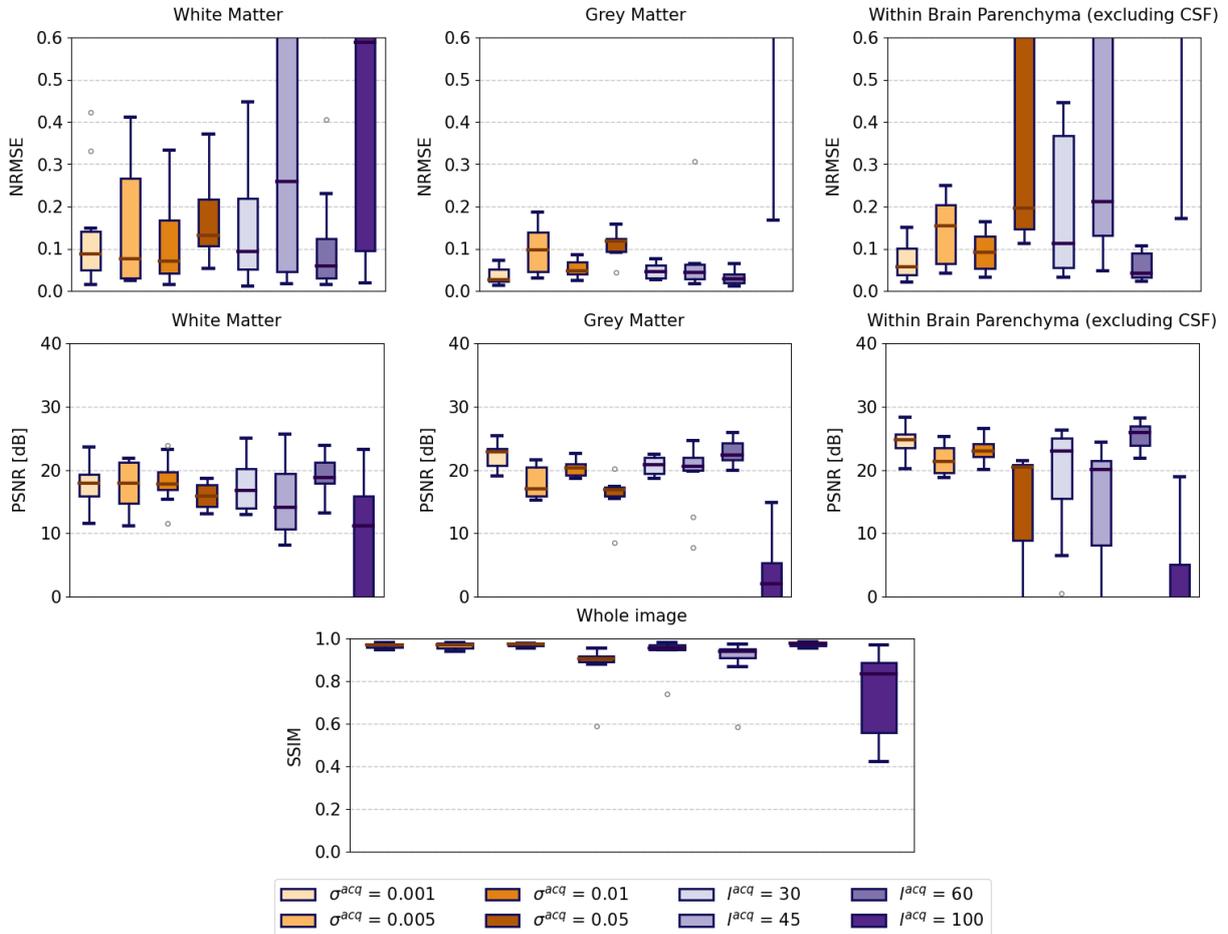


Figure 4.5: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noisy validation dataset by  $M_{\text{EXP}}$  networks trained with different acquisition noise amounts. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with 24 patches and  $J=6$ .

Figure 4.6 shows the NRMSE, PSNR and SSIM scores obtained on the very noisy validation images. Networks trained with  $\sigma^{\text{acq}}=0.001$  and  $l^{\text{acq}}=60$  remain the best-scoring networks in their respective groups in terms of NRMSE and PSNR. Despite the fact that this validation dataset was created with the same amount of noise that the network  $\sigma^{\text{acq}}=0.05$  used to train, this network achieves the worst NRMSE and PSNR scores in its group.

The good performance scores across all validation datasets for the network trained with  $\sigma^{\text{acq}}=0.001$  seems to suggest the RIM favours training with noiseless weighted images to produce maps that are more robust to noise in the evaluation dataset. However, the poor scores of network  $l^{\text{acq}}=100$  indicate that using a narrow distribution of low-valued  $\sigma^{\text{acq}}$  damages the estimation process, presumably because the RIM cannot generalize to other noise values. Given these results, we selected  $\sigma^{\text{acq}}=0.001$  for the evaluation network.

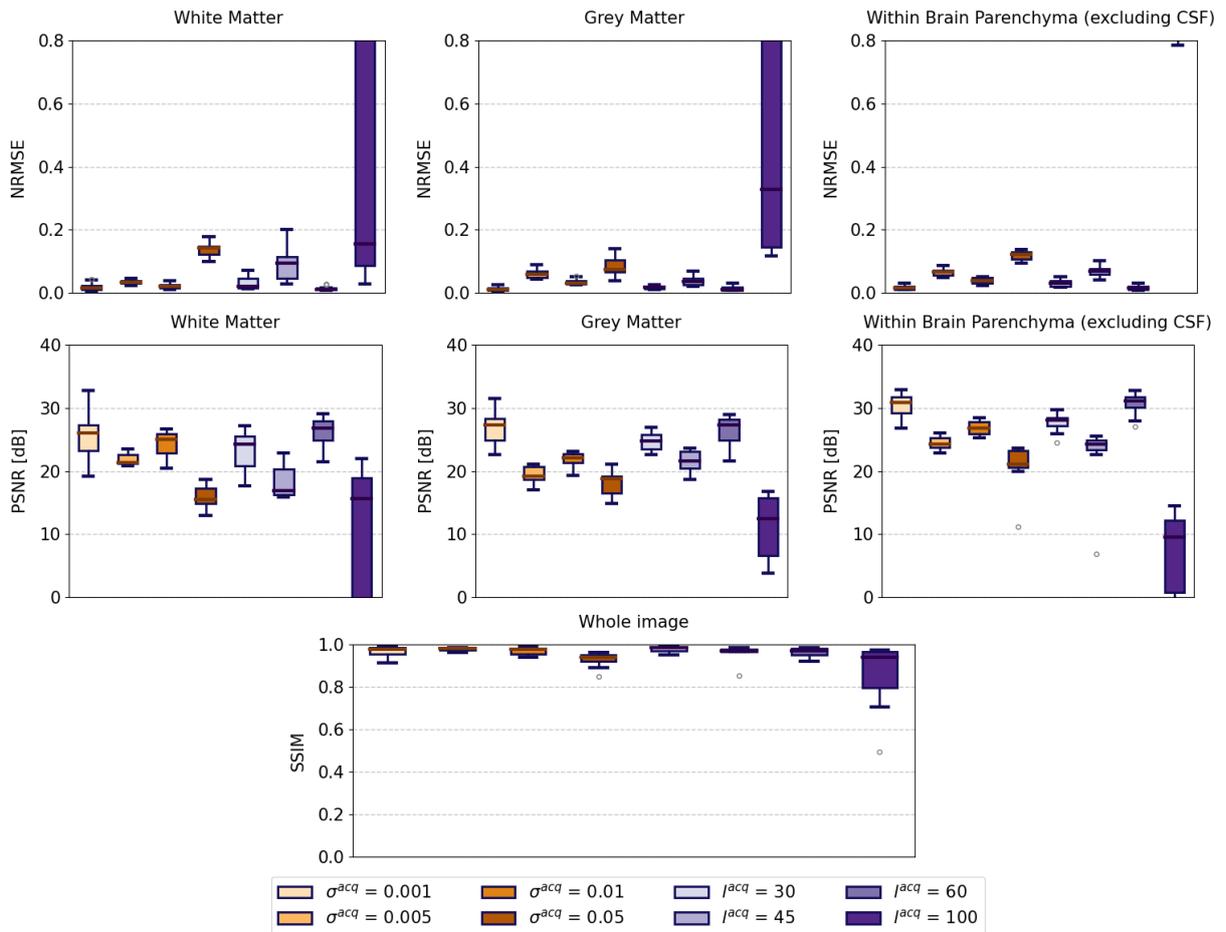


Figure 4.6: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the very noisy validation dataset by  $M_{EXP}$  networks trained with different acquisition noise amounts. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with 24 patches and  $J=6$ .

## 4.1.2 Testing

Taking into account the results of the validation stage, a network was trained with the model  $M_{\text{EXP}}$ , a minibatch of 24 patches, 6 inference steps and training images with acquisition noise with  $\sigma^{\text{acq}} = 0.001$ . The estimated  $T_2$  maps on the simulated testing datasets were evaluated and compared with those estimated with standard exponential fitting, implemented with non-linear least squares and a maximum number of iterations of 5000 (figure 4.7). The RIM produced a highly accurate  $T_2$  map on the noisy simulated dataset and achieved a similar performance to the exponential fitting on the very noisy dataset. On the noiseless images, the RIM could not estimate the  $T_2$  of the white matter region, possibly because all the training images contained some noise.

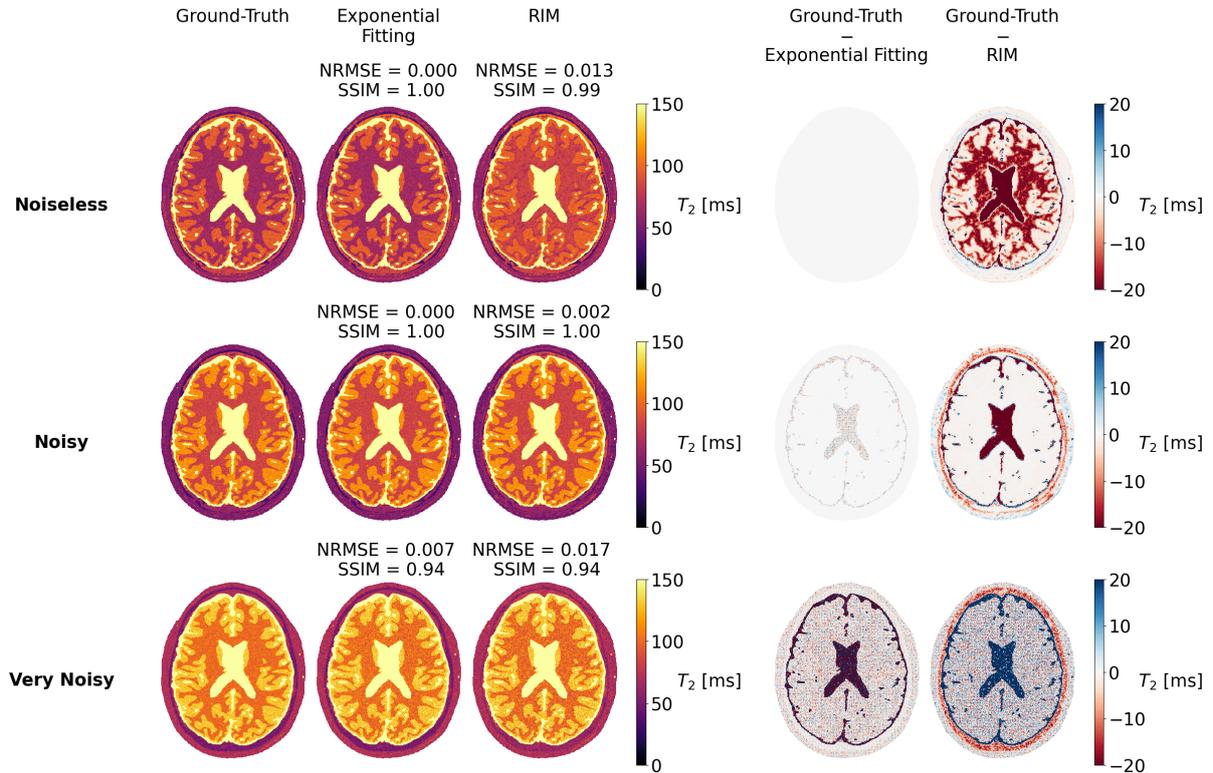


Figure 4.7:  $T_2$  ground-truth maps of the simulated testing datasets and corresponding estimates produced from exponential fitting and a  $M_{\text{EXP}}$  RIM trained with 24 patches,  $J=6$  and  $\sigma^{\text{acq}}=0.001$ . Difference maps between ground-truth and exponential fitting, and ground-truth and RIM. NRMSE scores were calculated on the brain parenchyma, while SSIM scores were calculated on the whole image.

To evaluate whether the intra-tissue variability of the ground-truth parametric maps of the testing images affects these results, these tests were repeated on testing datasets where the ground-truth maps were generated with homogenous  $T_2$  values per tissue (figure 4.8). In this case, both the RIM and the exponential fitting method estimated near-perfect  $T_2$  maps regardless of the noise of the ground-truth weighted images.

The  $T_2$  maps estimated on the *in vivo* datasets are compared with those estimated with the exponential fitting method in figure 4.9. In the *in vivo* datasets, the RIM successfully estimated the  $T_2$  maps with a median error of 3.41 ms relative to the exponential fitting maps in the worst case.

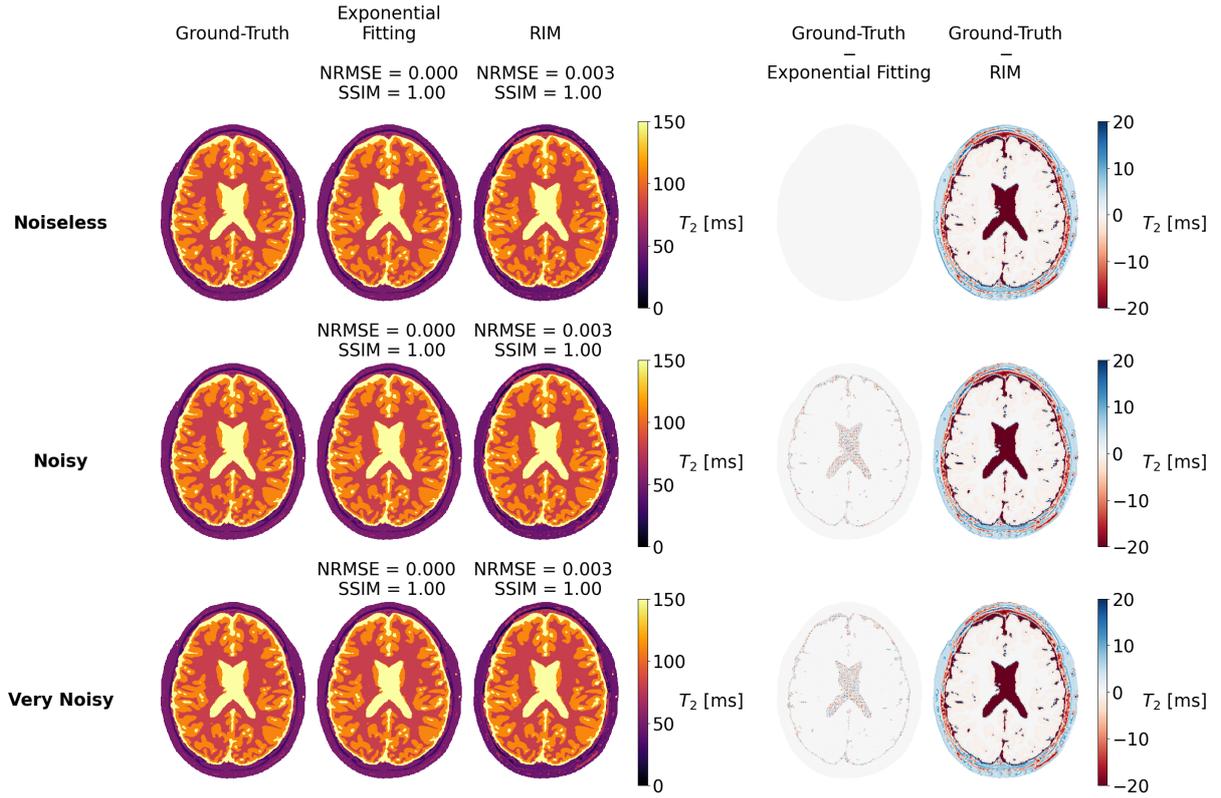


Figure 4.8:  $T_2$  ground-truth maps of the simulated testing datasets and corresponding estimates produced from exponential fitting and a  $M_{\text{EXP}}$  RIM trained with 24 patches,  $J=6$  and  $\sigma^{\text{acq}}=0.001$ . Difference maps between ground-truth and exponential fitting, and ground-truth and RIM. Ground-truth maps were generated with no intra-tissue variability. NRMSE scores were calculated on the brain parenchyma, while SSIM scores were calculated on the whole image.

## 4.2 Extended Phase Graphs Model

Once it was verified that the RIM could produce an accurate  $T_2$  map with the selected acquisition protocol, the complexity of the signal model was progressively increased. New networks were trained in a similar fashion, now with the model  $M_{\text{EPG}(T_2)}$ . These networks took on average 10 h to train.

The EPG forward model requires a predefined  $T_1$  value in order to perform  $T_2$  estimation, which may induce errors in the estimated maps. This effect is evaluated in figure 4.10, where  $T_2$  maps from the *in vivo* testing dataset were estimated with the state-of-the-art dictionary matching technique, using dictionaries created for distinct  $T_1$  values. All dictionaries were created following the  $M_{\text{EPG}(T_2)}$  signal model, with  $T_2$  values in the range 0-300 ms, with precision 1 ms, and  $B_1=1$ . For  $T_1$  values from 600 ms to 1200 ms, realistic  $T_1$  values in the brain [4], the estimated maps differ from the map estimated with  $T_1=1000$  ms by 1 ms at most. Thus, we can be confident that the chosen  $T_1$  does not affect the estimated  $T_2$  maps significantly, which is in accordance with the literature findings [5].

We proceeded to train networks with the  $M_{\text{EPG}(T_2)}$  signal model, considering  $T_1=1000$  ms and  $B_1=1$  on all tissues.

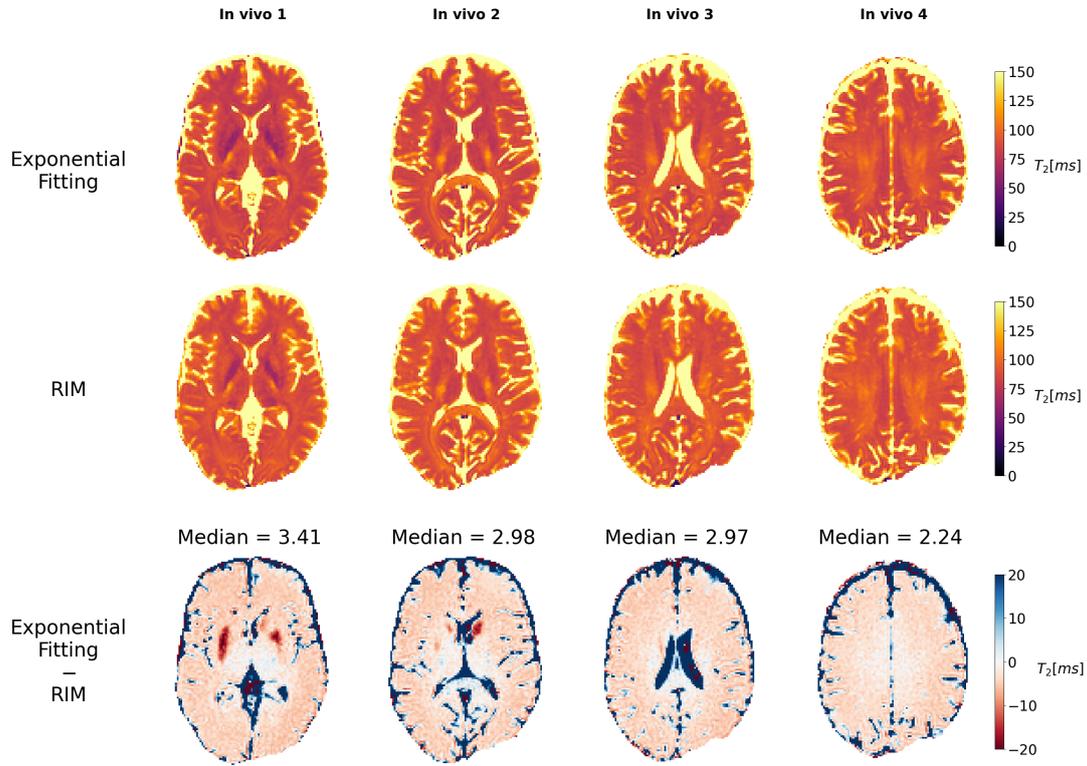


Figure 4.9:  $T_2$  maps of the *in vivo* testing dataset estimated with the exponential fitting method and the  $M_{\text{EXP}}$  RIM, and difference map between them. The dictionary matching used a dictionary with  $T_2$  values from 0 to 300 ms and a precision of 1 ms. The RIM was trained with 24 patches,  $J=6$  and  $\sigma^{\text{acq}}=0.001$ . Median scores of the difference maps were calculated in the brain parenchyma, excluding the CSF regions.

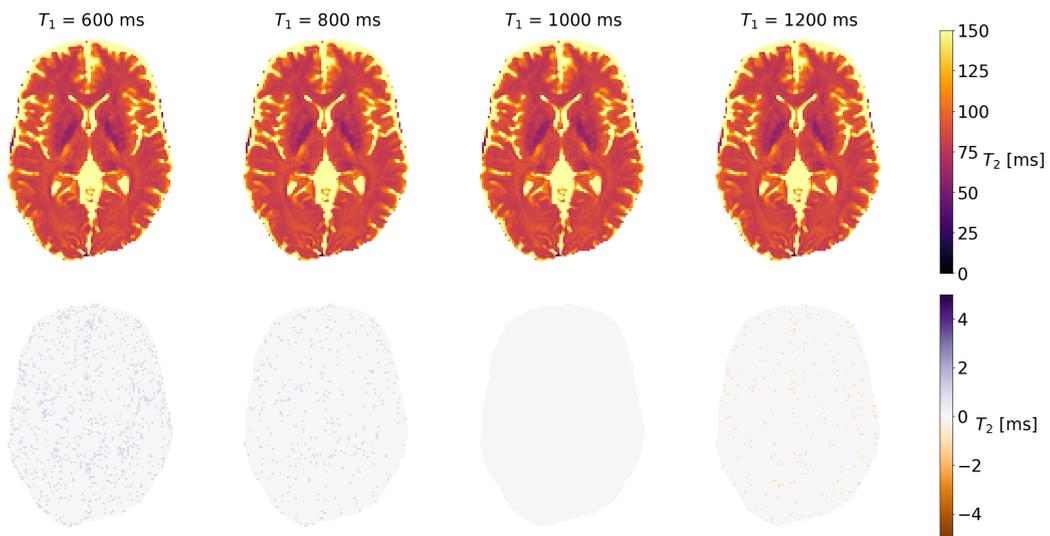


Figure 4.10:  $T_2$  maps estimated on one of the *in vivo* testing images for  $M_{\text{EPG}(T_2)}$  dictionaries with  $T_1=600, 800, 1000$  and  $1200$  ms, and difference maps between them and the map estimated with  $T_1=1000$  ms. All dictionaries were created for  $T_2$  ranging from 0 to 300 ms, with a precision of 1 ms, and  $B_1=1$ .

## 4.2.1 Validation

Since it is expected that the RIM estimates  $T_2$  maps similar to those achieved with the  $M_{EXP}$  model, the networks in this section were only validated for different sizes of the minibatch. Networks were trained with 36, 48, 60 and 72 patches, using 6 inference steps and a fixed simulated acquisition noise across patches with  $\sigma^{acq}=0.001$ .

Figure 4.11 showcases the NRMSE, PSNR and SSIM scores obtained in the noisy validation dataset for networks trained with different minibatch sizes. As the number of patches increases, so does the NRMSE score improve in the overall image excluding the CSF region. The model trained with 72 patches seems to achieve the best NRMSE scores, but contains the most extreme outliers in this performance metric. The network trained with 60 patches is the second best in terms of NRMSE and PSNR scores in the grey matter region and the overall image excluding the CSF, whereas the network trained with 48 patches achieves the second best PSNR score in the white matter region, and the best when considering the outliers. The network trained with 60 patches was selected for the testing stage.

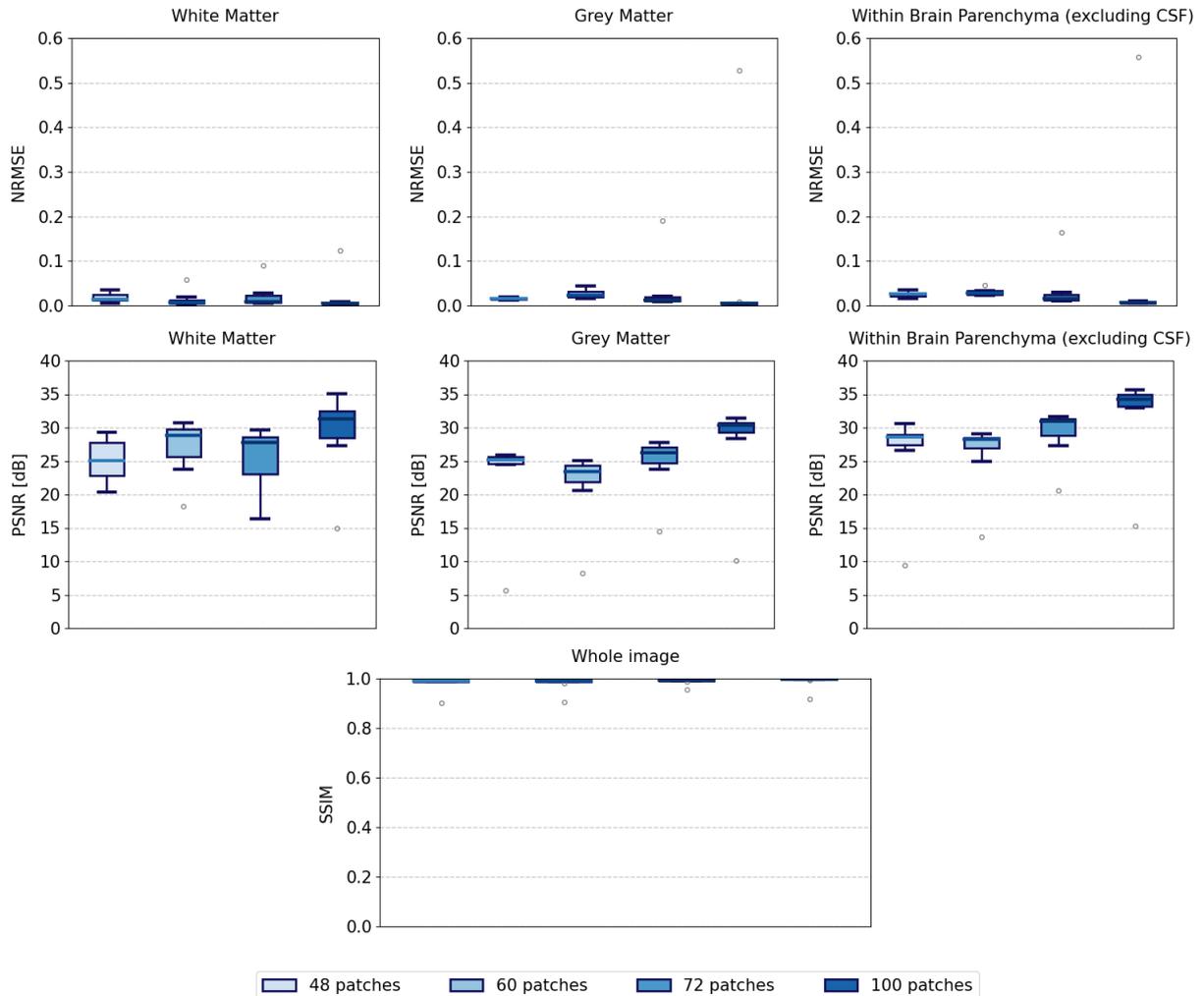


Figure 4.11: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noisy validation dataset  $M_{EPG(T_2)}$  networks trained with 36, 48, 60 and 72 patches. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with  $J=6$  and  $\sigma^{acq}=0.001$ .

## 4.2.2 Testing

Figure 4.12 displays the estimated  $T_2$  maps on the simulated testing datasets by a RIM trained with 60 patches per minibatch, 6 inference steps and  $\sigma^{\text{acq}}=0.001$ . Similarly to the exponential model, the RIM loses accuracy when estimating the white matter on noiseless images. The estimates on these datasets are less accurate than those obtained with dictionary matching, but the latter loses accuracy on the very noisy dataset, while the RIM can still estimate the  $T_2$  values of white matter. Once again, it can be verified that the RIM cannot capture the grainy texture of the ground-truth  $T_2$  maps corresponding to the variability of  $T_2$  values in each tissue. Figure 4.13 repeats these tests, but using testing datasets with no intra-tissue variability in the ground-truth maps. The RIM is able to estimate  $T_2$  maps of the all the datasets with minimal error, although less accurate than the ones obtained with dictionary matching. We can thus conclude the RIM struggles with estimating maps with large intra-tissue variability, which is exacerbated when the ground-truth weighted images contain noise.

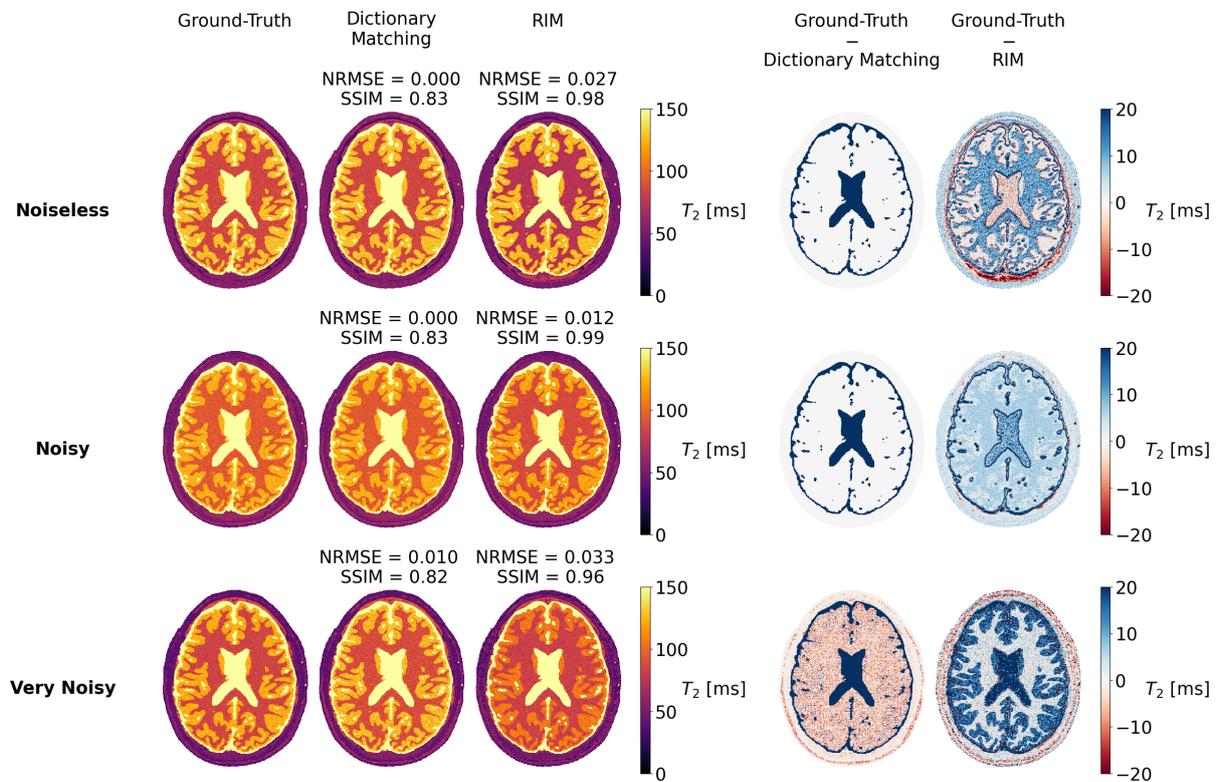


Figure 4.12:  $T_2$  ground-truth maps of the simulated testing datasets and corresponding estimates obtained from dictionary matching and a  $M_{\text{EPG}(T_2)}$  RIM trained with 60 patches,  $J=6$  and  $\sigma^{\text{acq}}=0.001$ . Difference maps between ground-truth and dictionary matching, and ground-truth and RIM. The dictionary covered  $T_2$  values from 0 to 300 ms, with a precision of 1 ms. NRMSE scores were calculated on the brain parenchyma, while SSIM scores were calculated on the whole image.

Figure 4.14 showcases the maps estimated by the RIM on the *in vivo* testing datasets, compared with those produced with dictionary matching. The RIM successfully estimates  $T_2$  maps with a median error of 1.66 ms in the worst case when compared to those estimated by the dictionary matching technique, with a larger difference in the grey matter regions.

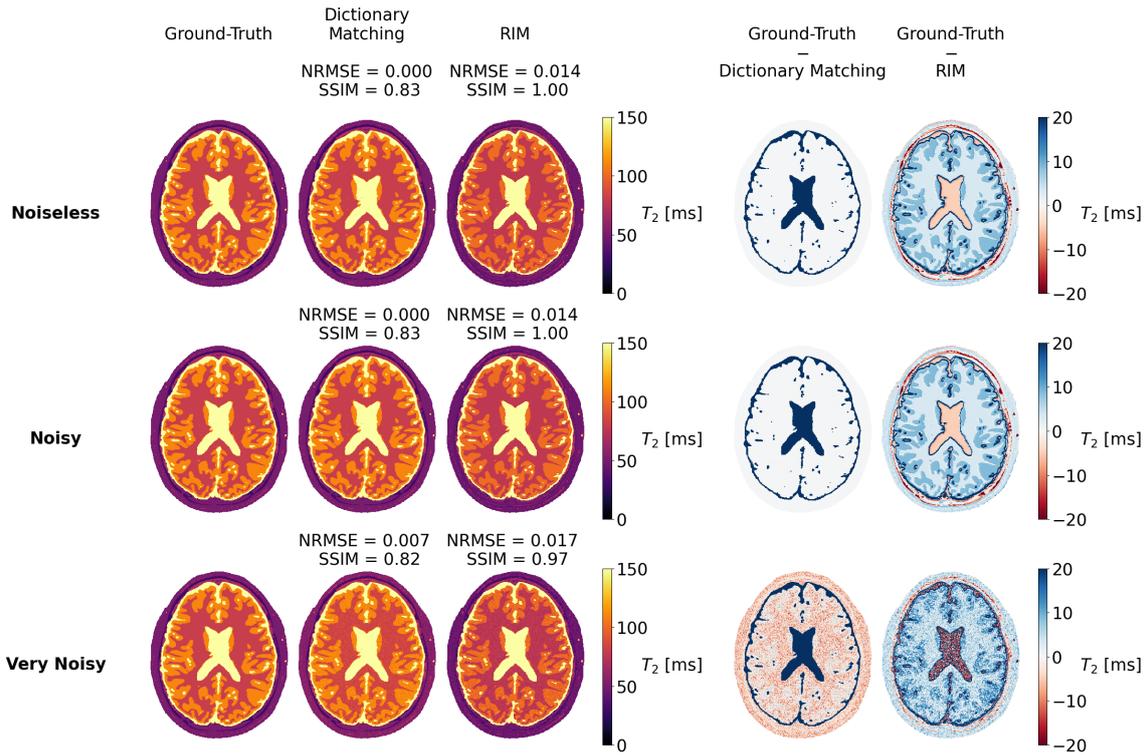


Figure 4.13:  $T_2$  ground-truth maps of the simulated testing datasets and corresponding estimates produced from dictionary matching and a  $M_{\text{EPG}}(T_2)$  RIM trained with 60 patches,  $J=6$  and  $\sigma^{\text{acq}}=0.001$ . Difference maps between ground-truth and dictionary matching, and ground-truth and RIM. The dictionary covered  $T_2$  values from 0 to 300 ms, with a precision of 1 ms. Ground-truth maps were simulated with no intra-tissue variability. NRMSE scores were calculated on the brain parenchyma, while SSIM scores were calculated on the whole image.

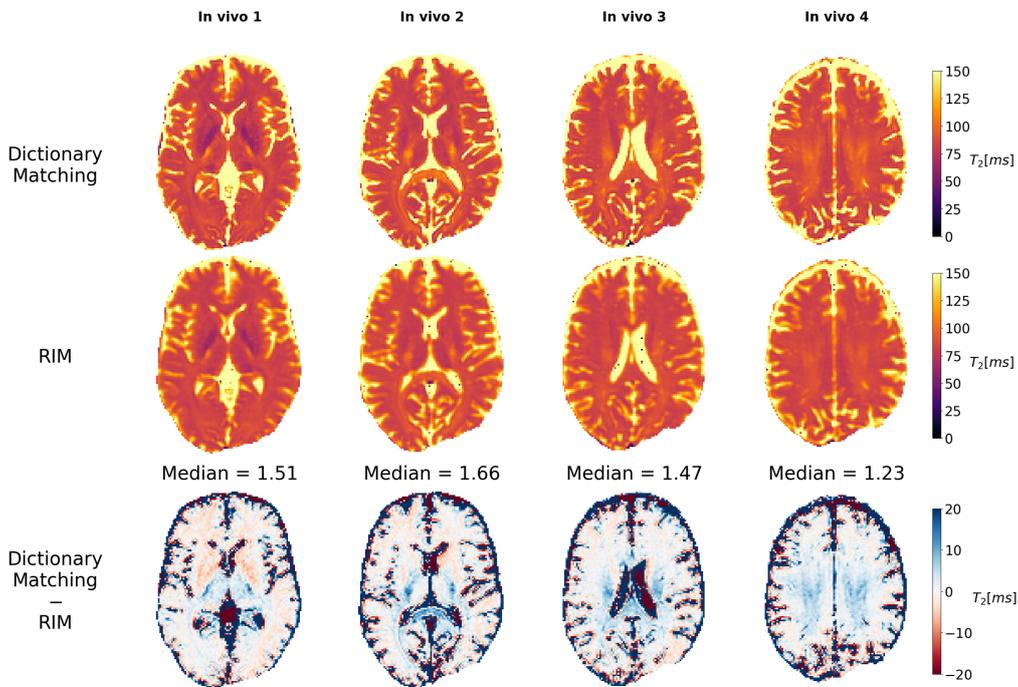


Figure 4.14:  $T_2$  maps estimated from the *in vivo* testing dataset with the dictionary matching technique and the  $M_{\text{EPG}}(T_2)$  RIM, and difference map between them. The dictionary matching used a dictionary with  $T_2$  values from 0 to 300 ms and a precision of 1 ms. The RIM was trained with 60 patches,  $J=6$  and  $\sigma^{\text{acq}}=0.001$ . Median scores of the difference maps were calculated in the brain parenchyma, excluding the CSF regions.

### 4.3 Extended Phase Graphs including the effect of B1

Once the RIM successfully estimated images that follow the EPG model, the complexity of the problem was increased by attempting to additionally estimate the  $B_1$  maps to apply their effect on the estimated  $T_2$  maps. Networks with this signal model took on average 10 hours to train. Once again, the effect of selecting different  $T_1$  values on the estimated  $T_2$  maps was evaluated (figure 4.15). Compared to the EPG model that disregards the effect of the  $B_1$  field, the selected  $T_1$  has a larger influence on the  $T_2$  maps. However, the  $T_2$  values on the grey and white matters only differ 1 ms at most from the reference  $T_1=1000$  ms for  $T_1$  values in the range 600-1200 ms. Thus, this approximation can be considered to have a negligible effect and the models  $M_{\text{EPG}(T_2, B_1)}$  were built with  $T_1 = 1000$  ms.

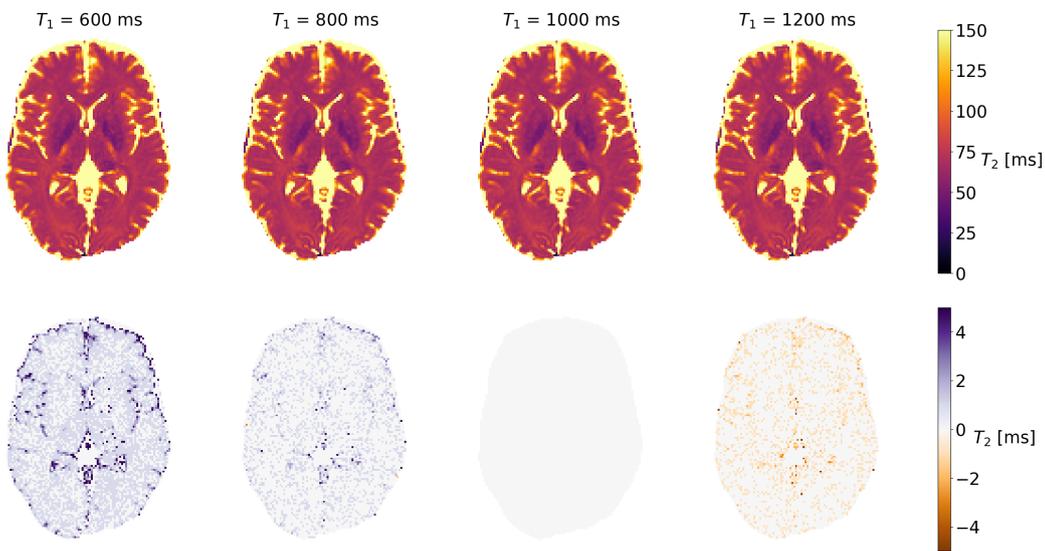


Figure 4.15:  $T_2$  maps estimated on one of the *in vivo* testing images for  $M_{\text{EPG}(T_2, B_1)}$  dictionaries with  $T_1=600, 800, 1000$  and  $1200$  ms, and difference maps between them and the map estimated with  $T_1=1000$  ms. All dictionaries were created for  $T_2$  ranging from 0 to 300 ms, with a precision of 1 ms, and  $B_1$  ranging from 0.6 to 1.4, with a precision of 0.01.

#### 4.3.1 Validation

Networks with the  $M_{\text{EPG}(T_2, B_1)}$  signal model were trained by varying the following hyper-parameters: size of the minibatch, number of inference steps, and standard deviation of the noise added to the training weighted images. In addition, the influence of the  $B_1$  ground-truth values set in the background region of the images was also analyzed. Models identified with “ $B_1$  masked with 0” correspond to  $B_1$  maps that follow the Gaussian function defined in equation 3.2 in the image regions with anatomical information, and are set to 0 in the background region. Similarly, models identified with “ $B_1$  masked with 0.6” have a  $B_1$  of 0.6 in the background region. All other models do not distinguish between background and brain regions.

## Size of the minibatch

Networks were trained with 48, 60, 72 and 100 patches in each minibatch. Figure 4.16 shows the NRMSE, PSNR and SSIM scores obtained on the estimated maps of the noisy validation dataset for networks trained with a different minibatch size. All these networks were trained with  $J=6$  inference steps, and a standard deviation of the acquisition noise equal to  $\sigma^{\text{acq}}=0.001$ . The networks trained with the EPG signal model performed better for a larger number of patches than those trained with the exponential model. All networks demonstrated similar NRMSE scores in the white matter region, whereas networks trained with 48 and 72 patches achieved the best scores in the grey matter region and in the overall image excluding the CSF. All networks scored similarly for the SSIM metric. The network trained with 72 patches was selected for the evaluation stage.

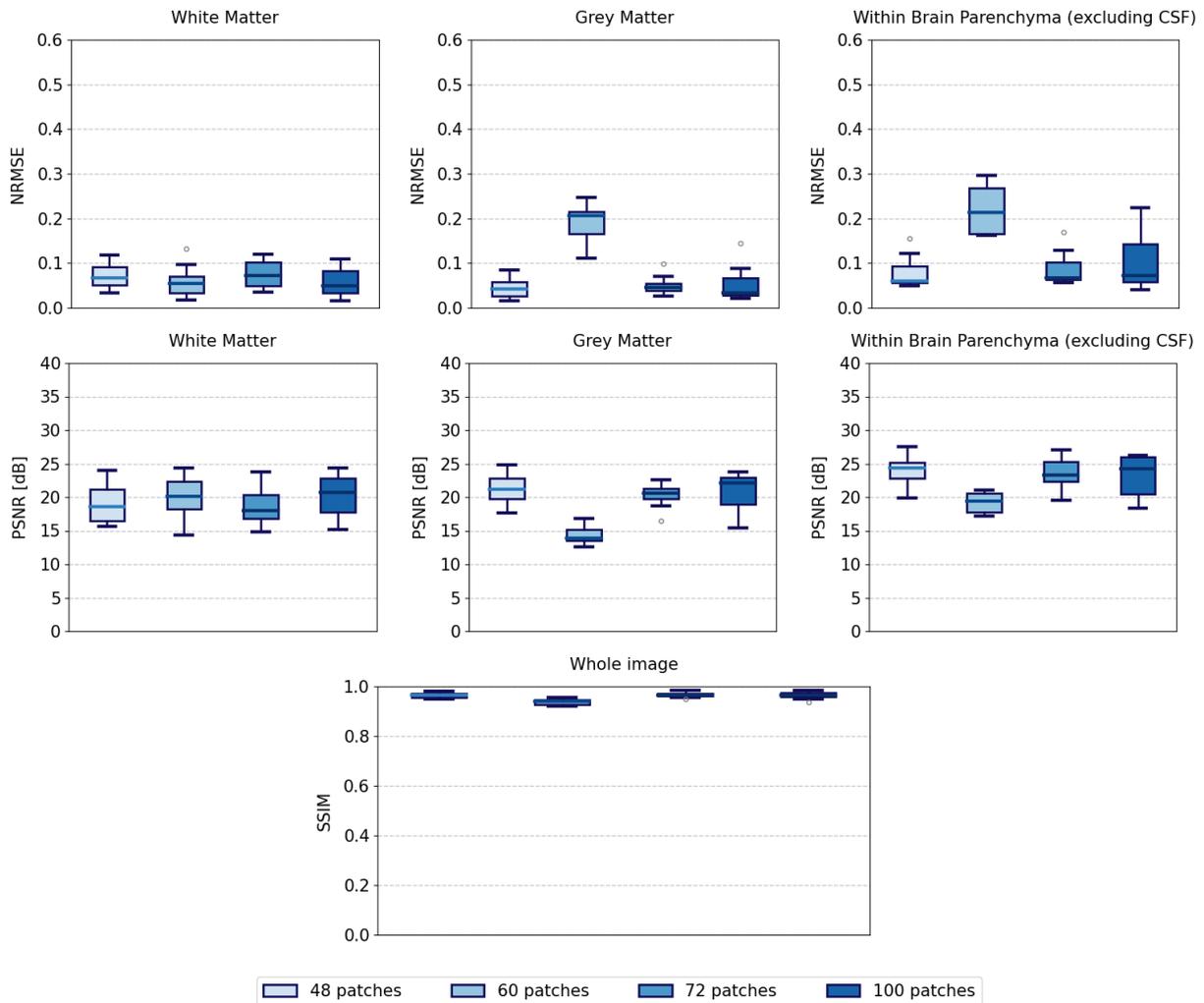


Figure 4.16: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noisy validation dataset by  $M_{\text{EPG}(T_2, B_1)}$  networks trained with 48, 60, 72 and 100 patches. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with  $J=6$  and  $\sigma^{\text{acq}}=0.001$ .

## Number of inference steps

Networks were trained with  $J=2, 4, 6$  and 8 inference steps. The NRMSE, PSNR and SSIM scores obtained in the noise validation set are represented in figure 4.17. All networks were trained with 72 patches and  $\sigma^{\text{acq}}=0.001$ . Once again, networks trained with 6 inference steps achieved the best scores overall. The network trained with 2 inference steps achieved the best NRMSE score in the overall image, but did not outperform the network with 6 inference steps on the white and grey matter regions.  $J=6$  inference steps was selected for the final network.

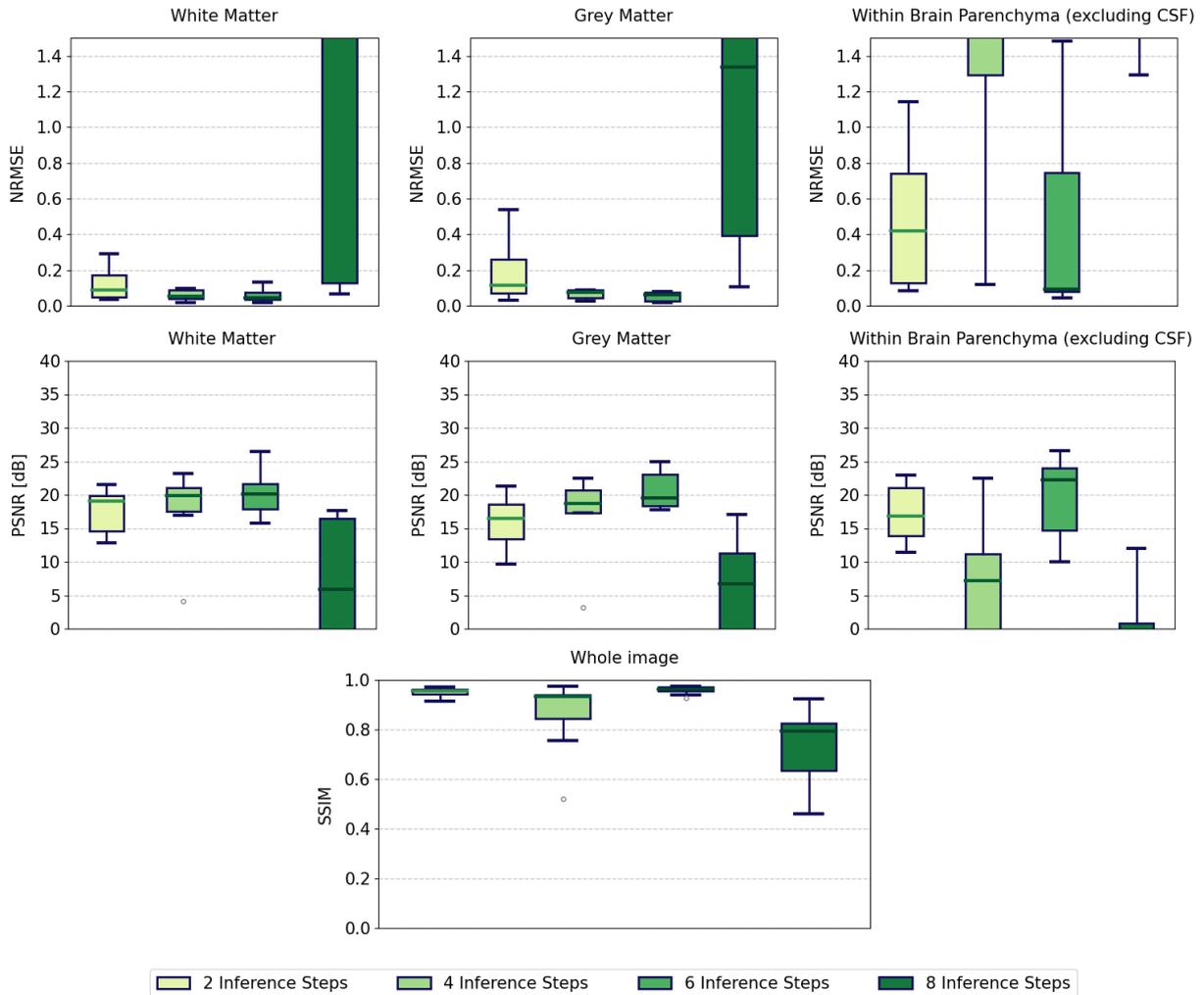


Figure 4.17: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noisy validation dataset by  $M_{\text{EPG}}(T_2, B_1)$  networks trained with 2, 4, 6 and 8 inference steps. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with 72 patches and  $\sigma^{\text{acq}}=0.001$ .

## Acquisition noise of the training images

Considering the observations made with the exponential signal model, networks were trained with  $\sigma^{\text{acq}}=0.001$  and  $l^{\text{acq}}=60$ . Each of these groups contained three variants: unmasked  $B_1$ ,  $B_1$  masked with

0 and  $B_1$  masked with 0.6. Since results point to a better estimation when training with low acquisition noise values, a network without noise addition to the ground-truth weighted images was also trained. All networks were trained with 72 patches and  $J=6$  inference steps.

Figure 4.18 shows the NRMSE, PSNR and SSIM scores obtained on the noiseless validation dataset. For the networks trained with a fixed value of  $\sigma^{\text{acq}}$  across minibatches, masking the  $B_1$  map with 0 in the background region results in poorer NRMSE and PSNR scores, whereas masking with 0.6 improves the PSNR score on white matter and in the overall image. For the networks trained with a distribution of  $\sigma^{\text{acq}}$ , masking the  $B_1$  map resulted in worse performance scores for both types of mask. The network trained with noiseless weighted images and the network trained with  $\sigma^{\text{acq}}=0.001$  and  $B_1$  masked with 0.6 achieved the best performance.

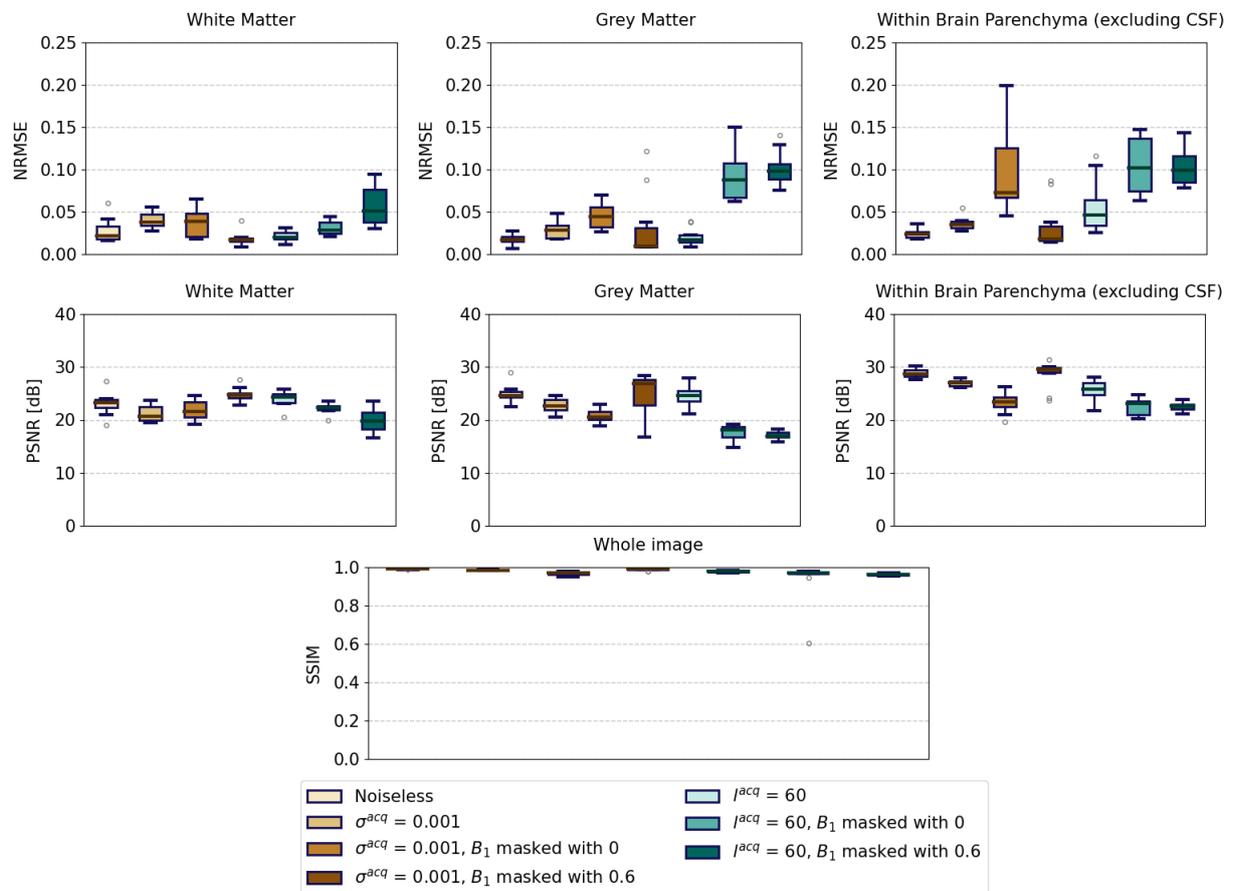


Figure 4.18: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noiseless validation dataset by  $M_{\text{EPG}}(T_2, B_1)$  networks trained with different acquisition noise amounts. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with 72 patches and  $J=6$ .

The NRMSE, PSNR and SSIM scores obtained in the noisy validation dataset are represented in figure 4.19. When evaluated on a noisier dataset, the network trained with  $\sigma^{\text{acq}}=0.001$  and  $B_1$  masked with 0.6 shows more unstable performance scores. The network trained with noiseless weighted images remains the best performing model in the overall image, while the networks trained with a distribution of

$\sigma^{\text{acq}}$  show poorer results.

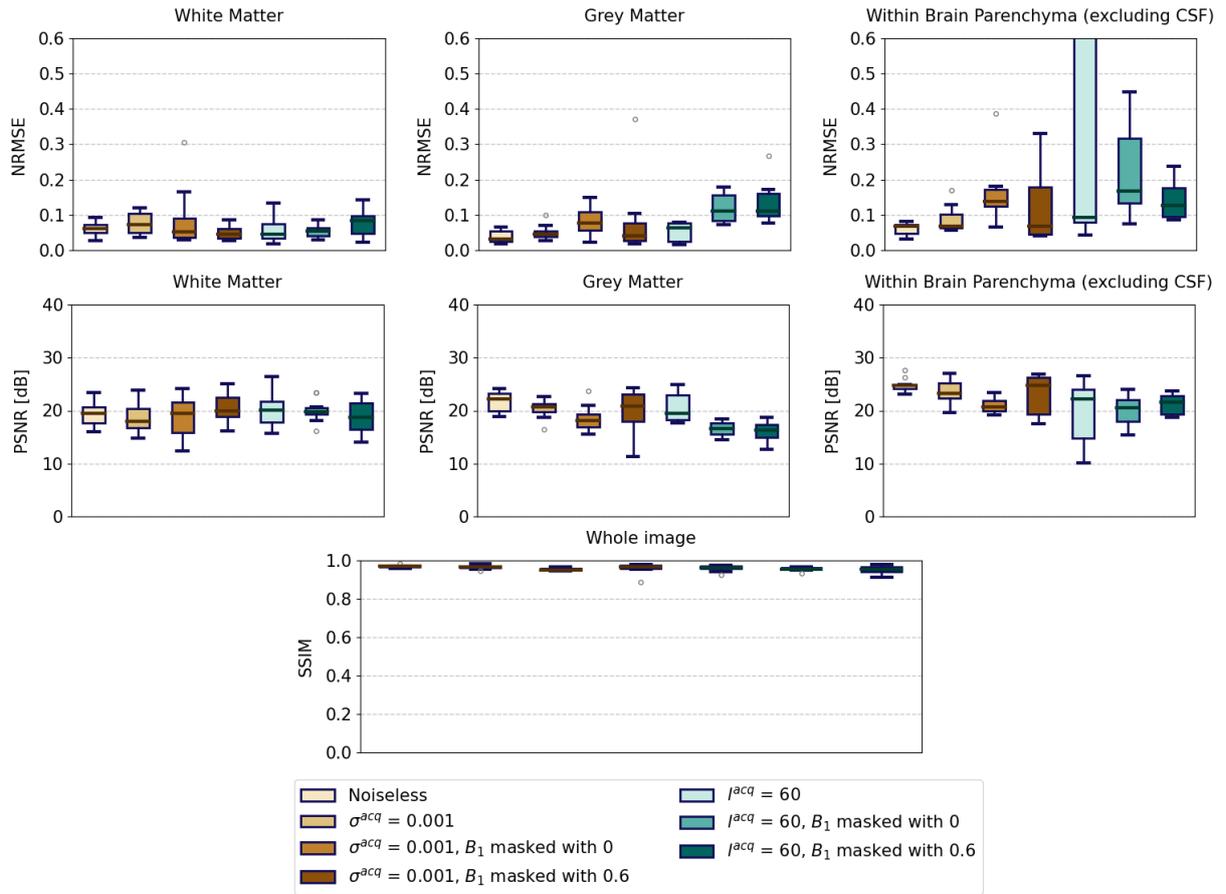


Figure 4.19: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the noisy validation dataset by  $M_{\text{EPG}}(T_2, B_1)$  networks trained with different acquisition noise amounts. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with 72 patches and  $J=6$ .

Figure 4.20 shows the NRMSE, PSNR and SSIM scores obtained on the very noisy validation dataset. The network trained with  $\sigma^{\text{acq}}=0.001$  and  $B_1$  masked with 0.6 becomes one of the worst-performing models in NRMSE and PSNR scores. In both types of networks, masking the  $B_1$  in the background region deteriorates the results. The network trained with noiseless echoes remains the best scoring network.

From this analysis, we confirm the RIM's preference to train with noiseless weighted images. Furthermore, the results suggest that training with an unmasked  $B_1$  field is more robust to estimation on increasingly noisier evaluation datasets.

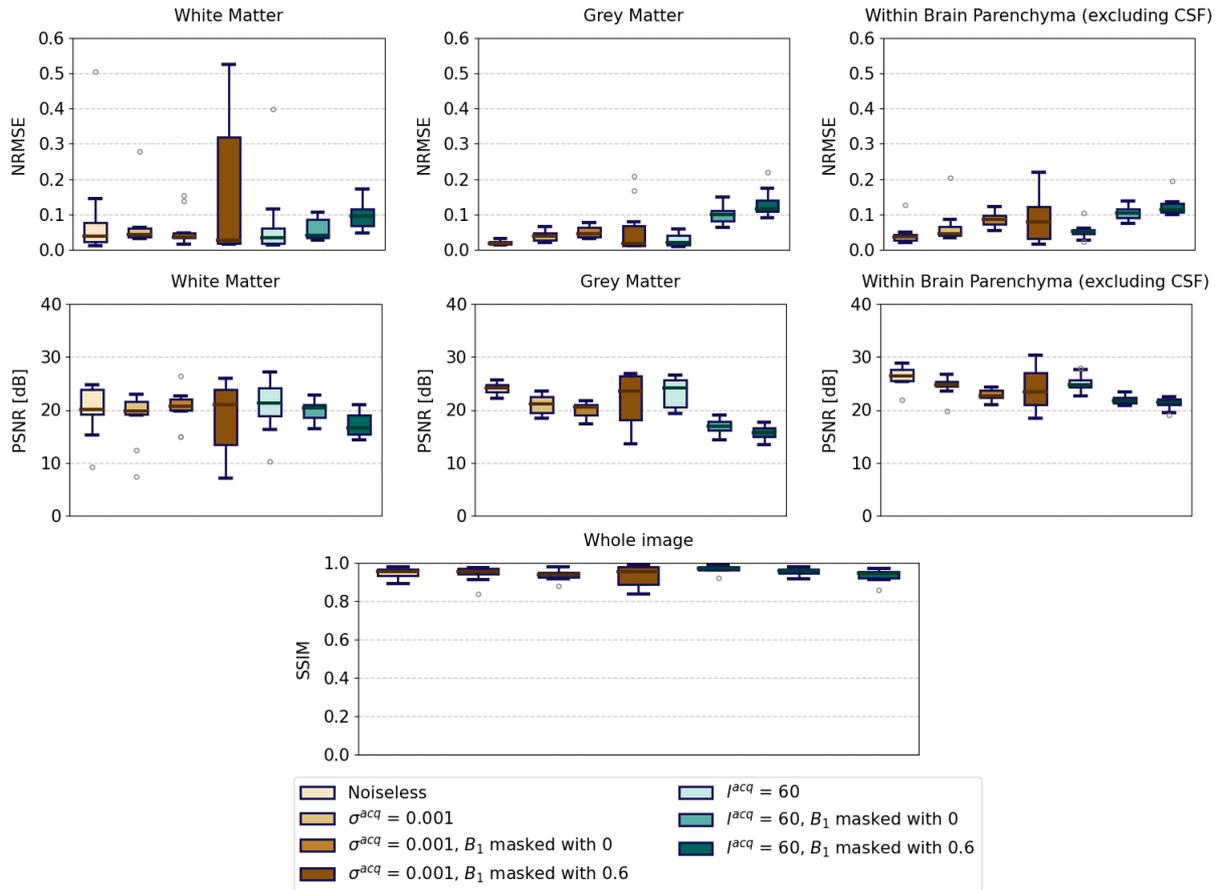


Figure 4.20: NRMSE, PSNR and SSIM scores of the estimated  $T_2$  maps on the very noisy validation dataset by  $M_{EPG}(T_2, B_1)$  networks trained with different acquisition noise amounts. NRMSE and PSNR scores were calculated in three separate regions (white matter, grey matter and whole image excluding the CSF), whereas the SSIM score was calculated on the full map. All networks were trained with 72 patches and  $J=6$ .

### 4.3.2 Testing

Based on the validation results, a RIM using 72 patches, 6 inference steps, weighted images without noise addition and ground-truth  $B_1$  fields that do not distinguish between background and brain regions in the image was trained.

Figures 4.21 and 4.22 showcase the maps estimated by the RIM on the simulated testing datasets, as well as the maps estimated with the standard dictionary matching technique, using testing datasets with and without intra-tissue variability in the parametric maps, respectively. The maps obtained with the RIM show a clear difference in accuracy between regions with a higher  $B_1$  and regions with a lower  $B_1$ : the accuracy is lower where the ground-truth  $B_1$  field is closer to 0.6. In the remaining regions, the RIM proves to be robust to the noise on the testing set. On the other hand, the dictionary matching technique has a much higher accuracy on the noiseless and noisy datasets, regardless of the presence of intra-tissue variability on the ground-truth maps, but shows poor performance on very noisy images when compared to the RIM, achieving lower NRMSE and SSIM scores. When removing the intra-tissue variability of the ground-truth maps, the artifacts induced by a wrong estimation of the  $B_1$  field become more clear. In this case, the dictionary method outperforms the RIM in all datasets.

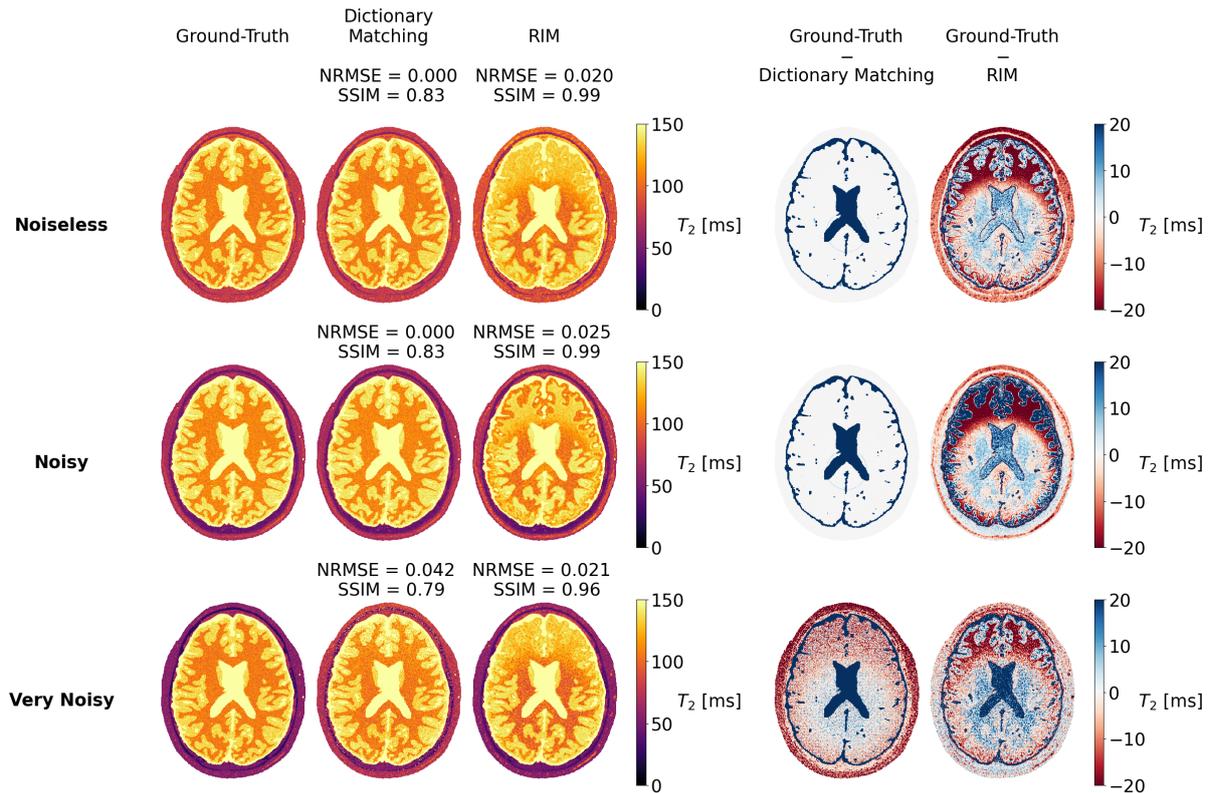


Figure 4.21:  $T_2$  ground-truth maps of the simulated testing datasets and corresponding estimates obtained with dictionary matching and a  $M_{EPG}(T_2, B_1)$  RIM trained with 72 patches,  $J=6$  and noiseless weighted images. Difference maps between ground-truth and dictionary matching, and ground-truth and RIM. The dictionary covered  $T_2$  values from 0 to 300 ms, with a precision of 1 ms, and  $B_1$  values from 0.6 to 1.4, with a precision of 0.01. NRMSE scores were calculated on the brain parenchyma, while SSIM scores were calculated on the whole image.

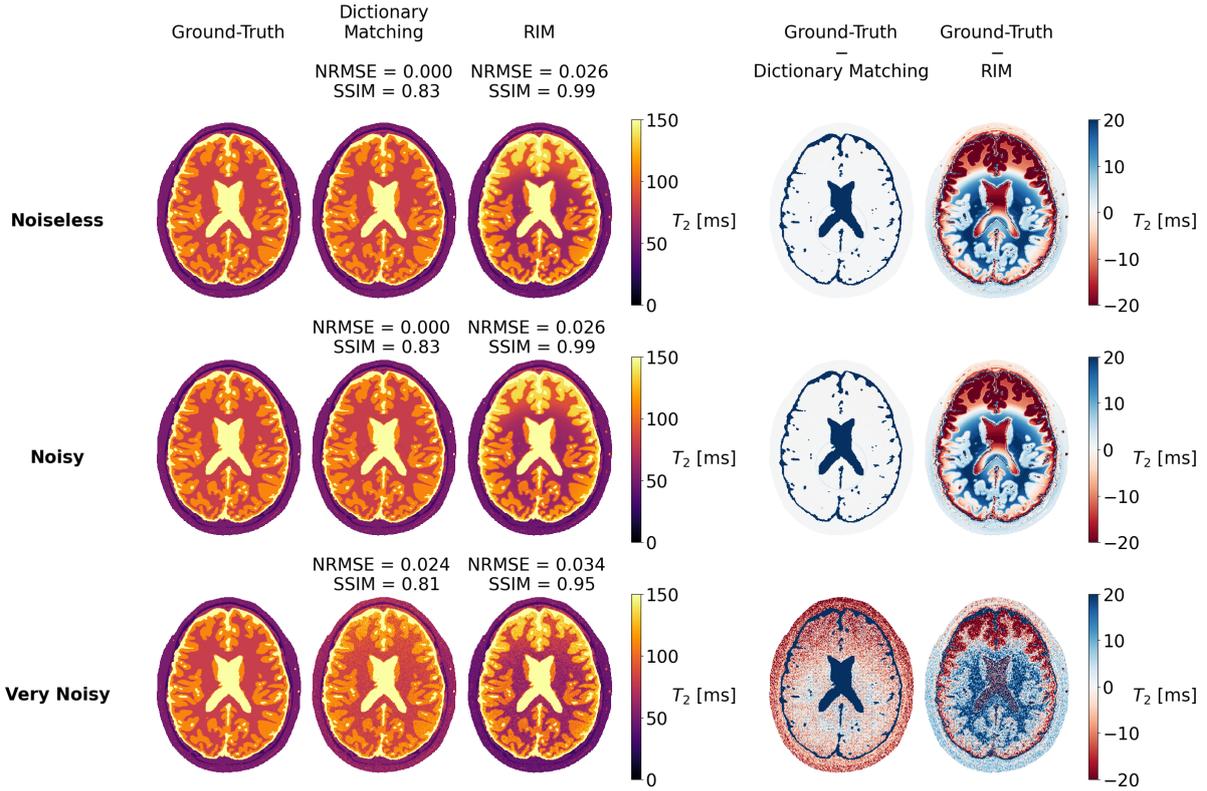


Figure 4.22:  $T_2$  ground-truth maps of the simulated testing datasets and corresponding estimates produced from dictionary matching and a  $M_{\text{EPG}(T_2, B_1)}$  RIM trained with 72 patches,  $J=6$  and noiseless weighted images. Difference maps between ground-truth and dictionary matching, and ground-truth and RIM. The dictionary covered  $T_2$  values from 0 to 300 ms, with a precision of 1 ms, and  $B_1$  values from 0.6 to 1.4, with a precision of 0.01. Testing datasets parametric maps were simulated with no intra-tissue variability. NRMSE scores were calculated on the brain parenchyma, while SSIM scores were calculated on the whole image.

The RIM's performance for variants of the noisy testing dataset with different  $B_1$  fields, including intra-tissue variability in the ground-truth parametric maps, is evaluated in figure 4.23. The RIM shows a clear difficulty to accurately estimate the  $B_1$  maps, and the estimate particularly loses quality on images where the  $B_1$  presents a large range of values, whereas with smoother  $B_1$  fields the estimates become more accurate. It is also possible to observe that the regions where the RIM's map estimate is more accurate are the ones corresponding to a  $B_1$  field closer to 1, and the ones with less accuracy correspond to more extreme values. This suggests that the network did not train with enough distinct  $B_1$  maps to learn how to estimate  $T_2$  values for a wide range of  $B_1$  values.

To assess whether the RIM is capable of estimating maps from realistic images, the network was tested on the *in vivo* datasets in figure 4.24(a). The ground-truth  $T_2$  map of these datasets is not known, so the RIM was compared with the maps estimated through dictionary matching. The RIM finds the overall range of  $T_2$  values in grey and white matter with a maximum median error of 4.20 ms in relation to the maps estimated with dictionary matching, but seems to slightly overestimate the larger  $T_2$  values.

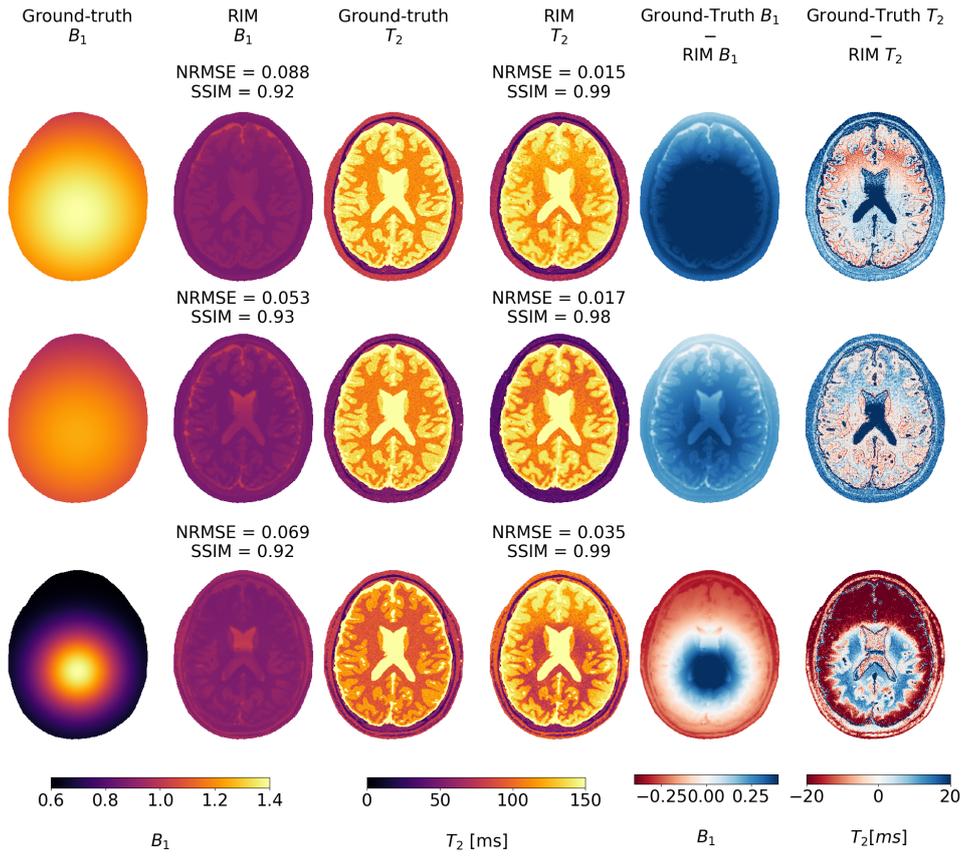
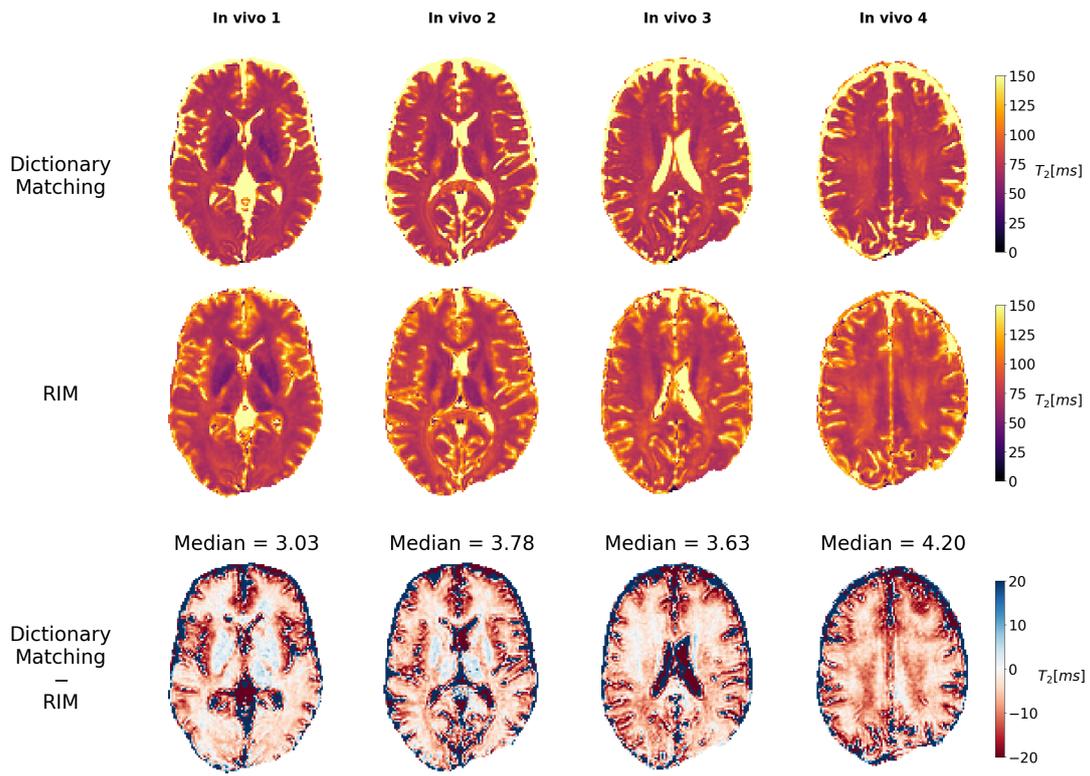
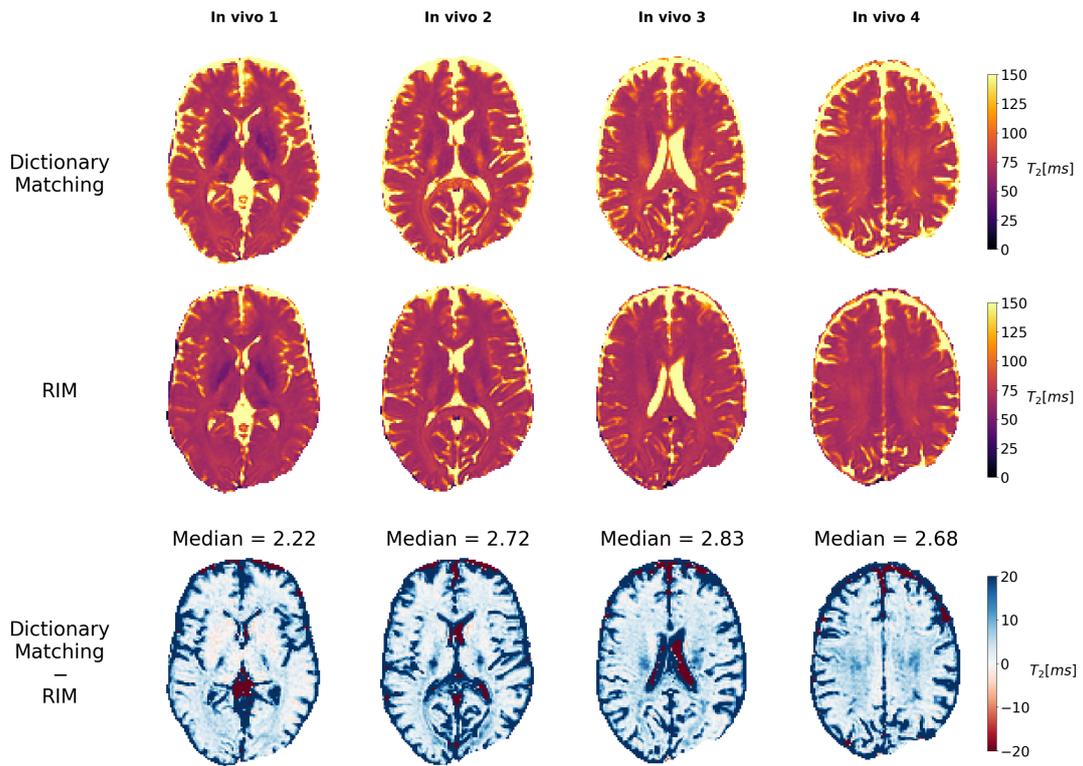


Figure 4.23:  $B_1$  and  $T_2$  ground-truth maps of the simulated noisy testing dataset with three different  $B_1$  fields and corresponding estimates produced from a  $M_{\text{EPG}(T_2, B_1)}$  RIM trained with 72 patches,  $J=6$  and noiseless weighted images. Difference maps between ground-truth and RIM. NRMSE scores were calculated on the brain parenchyma, while SSIM scores were calculated on the whole image.

It can be verified that the *in vivo*  $T_2$  values in the grey and white matter regions were lower than those reported in section 3.3 and used to train the networks. In figure 4.24(b), the previous analysis is repeated for a network trained with lower ground-truth  $T_2$  values for grey and white matter,  $\mu_{\text{grey matter}}^{\text{GT}} = 85$  ms and  $\mu_{\text{white matter}}^{\text{GT}} = 65$  ms respectively. The estimated  $T_2$  maps are now closer to those obtained through dictionary matching, with a maximum median error of 2.83 ms, now with some underestimation of  $T_2$  values in grey matter. We conclude that the training data was simulated with insufficient variability in the  $T_2$  values, thus increasing the network's sensitivity towards  $\mu^{\text{GT}}$ . Nevertheless, given the small range of  $T_2$  values of each tissue that the RIM could access during training, the network can still produce accurate  $T_2$  maps that are very close to those produced with the dictionary matching technique.



(a)  $\mu_{\text{white matter}}^{\text{GT}} = 80 \text{ ms}$ ,  $\mu_{\text{grey matter}}^{\text{GT}} = 100 \text{ ms}$



(b)  $\mu_{\text{white matter}}^{\text{GT}} = 65 \text{ ms}$ ,  $\mu_{\text{grey matter}}^{\text{GT}} = 85 \text{ ms}$

Figure 4.24:  $T_2$  maps of the *in vivo* testing dataset estimated with the dictionary matching technique and the  $M_{\text{EPG}}(T_2, B_1)$  RIM, and difference map between them. The dictionary matching technique used a dictionary with  $T_2$  values from 0 to 300 ms and a precision of 1 ms, and  $B_1$  values from 0.6 to 1.4 with a precision of 0.01. The RIM was trained with 72 patches,  $J=6$ , and noiseless weighted images. Median scores of the difference maps were calculated in the brain parenchyma, excluding the CSF regions.

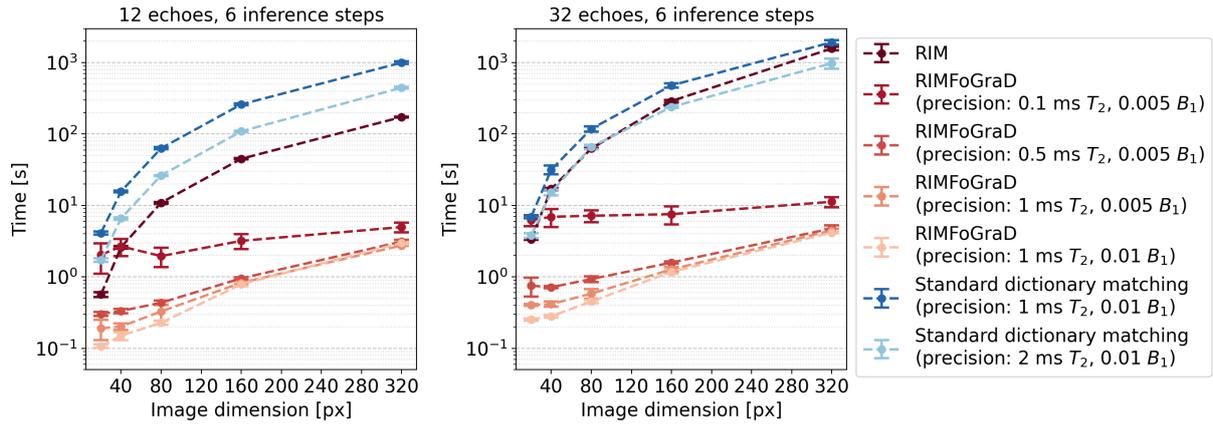
## 4.4 Performance Analysis

During the training, validation and testing experiments, we verified that the time required to produce a  $T_2$  map with the RIM using EPG signal models was quite large when compared with the exponential model. This was to be expected since not only is the EPG model more complex, thus requiring a larger number of calculations, but the signal measured in each echo can also only be computed once the signals of all the previous echoes have been calculated. Contrary to the exponential model, where only the time point and the  $T_2$  and PD maps are required, the EPG needs to calculate the measured signals of all echoes before arriving at the desired time-point, which results in larger computational times, especially when calculating the derivatives of the signal with respect to  $T_2$  and  $B_1$  using automatic differentiation methods due to the complexity of the computational graph. Furthermore, because of the very complex structure of the RIM, the forward signal model and its partial derivatives need to be calculated every inference step. For these reasons, RIMs may not be suited for very complex signal models like the EPG framework: despite the accuracy of the estimated maps, there may not be an advantage in performance compared to state-of-the-art methods.

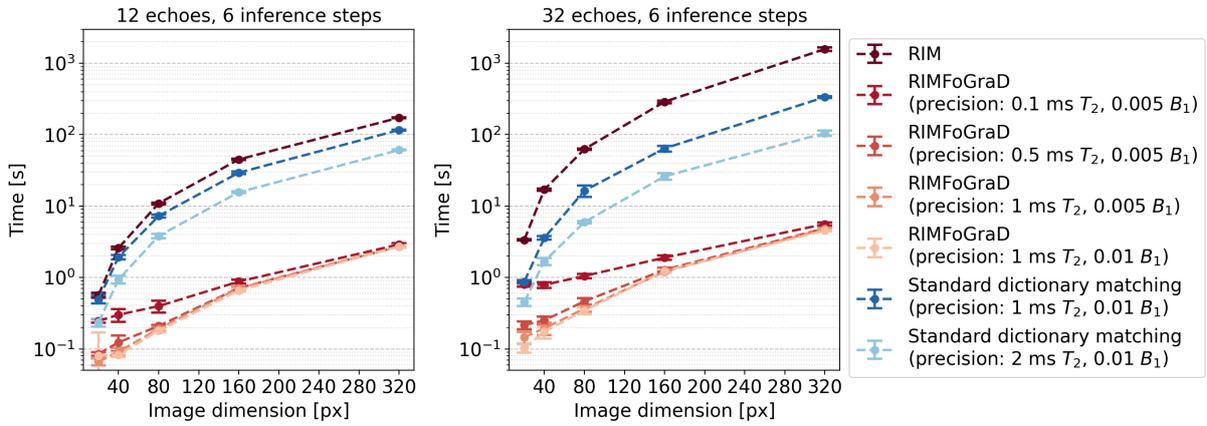
In this work, RIMFoGraD was introduced to tackle this setback. RIMFoGraD networks share the same configuration as the RIM, but use pre-calculated dictionaries containing the forward signal model and its partial derivatives for a wide range of parameters (as defined in section 3.2) to avoid complex calculations during the inference stage. The performance of the RIMFoGraD compared to the RIM will only depend on the time required to load the dictionary into the network and to access the signal and derivatives corresponding to each pixel in the set of weighted images. All RIMFoGraDs were trained in a similar fashion to RIMs, calculating the signal model at each step and using automatic differentiation tools to obtain the derivatives, in order to preserve the precision of the network; dictionaries were only applied during evaluation. Therefore, all results presented so far are also valid for the RIMFoGraD.

In figure 4.25, the time required to obtain a  $T_2$  map with varying dimensions with the  $M_{\text{EPG}(T_2, B_1)}$  signal model, for protocols with 12 and 32 echoes, by the RIM, RIMFoGraDs with dictionaries with varying precision, and the state-of-the-art dictionary matching technique with dictionaries with varying sizes is compared. In 4.25(a), the dictionaries were created for  $T_2$  values ranging 0-2000 ms, which corresponds to the range of values considered when training; in 4.25(b), the dictionaries were created for  $T_2$  values ranging 0-300 ms, which is a smaller interval that still allows a precise estimation of the  $T_2$  maps on grey and white matter using standard dictionary matching.

As expected, the time required to produce a  $T_2$  map from a set of weighted images increases exponentially as the image dimensions increase. The average time to produce a  $320 \times 320$  pixel  $T_2$  map of a sequence with 32 echoes was  $1931 \pm 130$  s and  $1578 \pm 82$  s for the state-of-the-art dictionary matching and for the RIM, respectively. If we account for potential errors in the RIM estimate, we can compare its performance with matching with a dictionary with a precision of 2 ms, which required an average of  $980 \pm 149$  s to produce the same map. In sequences with a lower ETL, the RIM can outperform the dictionary matching: for a sequence with 12 echoes, the RIM is 2.6% faster than the low-precision dictionary matching. However, when considering dictionaries with a lower range of  $T_2$  values, the RIM



(a)  $T_2$  dictionaries created for  $T_2$  values from 0 ms to 2000 ms;  $B_1$  dictionaries created for  $B_1$  values from 0.6 to 1.4.



(b)  $T_2$  dictionaries created for  $T_2$  values from 0 ms to 300 ms;  $B_1$  dictionaries created for  $B_1$  values from 0.6 to 1.4.

Figure 4.25: Average time needed to obtain a  $T_2$  map from a set of squared weighted images with varying dimensions, for protocols with 12 and 32 echoes, for the following models: **RIM** — standard RIM, where the forward signal model and the partial derivatives are calculated during inference; **RIMFoGraD** — RIM coupled with dictionaries with varying  $T_2$  and  $B_1$  precision; **standard dictionary matching** — standard matching with dictionaries with varying  $T_2$  and  $B_1$  precision. Points and error bars result from the mean and standard deviation over set of 10 repeated measurements, respectively.

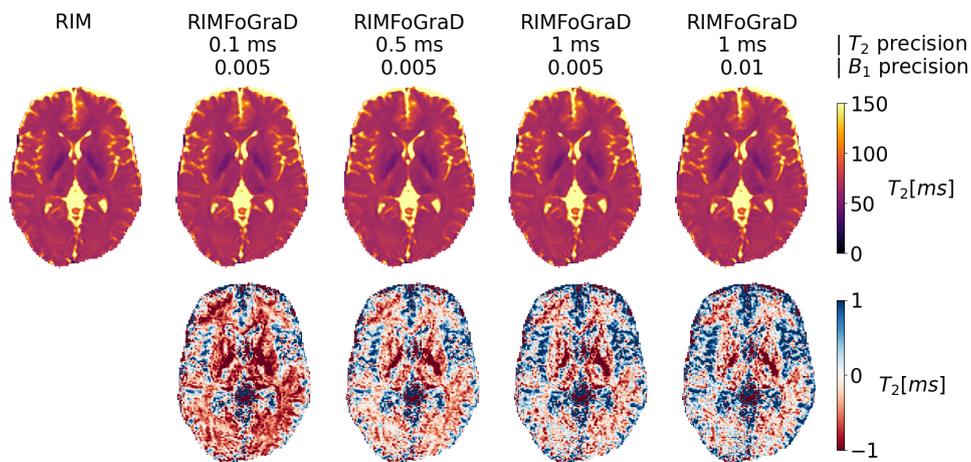
underperforms, as expected.

On the other hand, all the RIMFoGraD networks outperform both the RIM and the dictionary matching method in all cases. The RIMFoGraD with the most precise dictionary (0.1 ms in  $T_2$  and 0.005 in  $B_1$ ) can produce a  $320 \times 320$   $T_2$  map from 32 echoes in  $11.27 \pm 1.77$  s on average, whereas the one with less precision (1 ms in  $T_2$  and 0.01 in  $B_1$ ) accomplishes the same task in  $4.16 \pm 0.10$  s. The counterpart RIMFoGraD networks with dictionaries with a smaller range of  $T_2$  values produce the same  $T_2$  maps in  $5.60 \pm 0.26$  s and  $4.56 \pm 0.09$  s, respectively. Furthermore, we verify that the time required to produce a map with the RIMFoGraD does not increase as steeply with the map dimension as the other methods: creating a  $20 \times 20$  map versus creating a  $320 \times 320$  map results in a 1574 s (471%) increase with the RIM, a 979 s (2177%) increase with the smallest dictionary matching, and a 5 s (1.8%) increase with the RIMFoGraD with the largest dictionary.

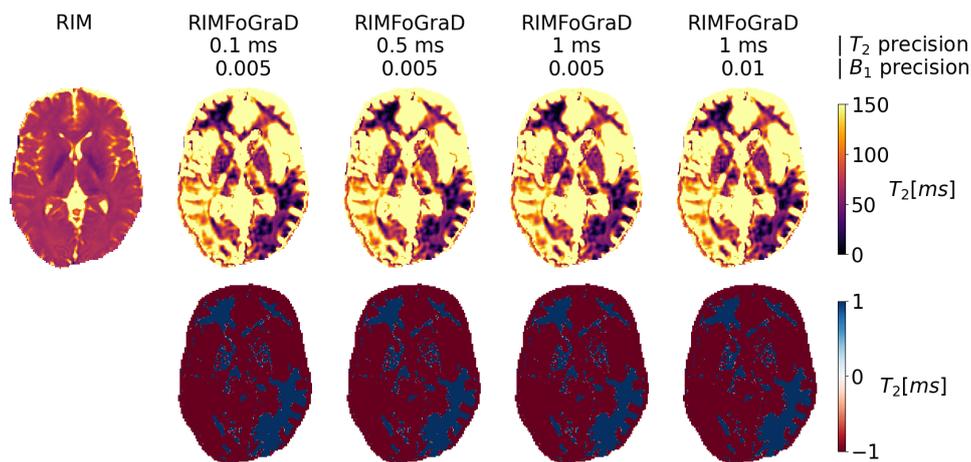
To evaluate the loss of precision of the estimates produced by the RIMFoGraD with varying dictionaries, we compared the  $T_2$  maps obtained from the *in vivo* testing dataset between the RIM and

RIMFoGraDs with dictionaries with decreasing precision (figure 4.26(a)). Surprisingly, the RIMFoGraD with the most precise dictionary is the one that differs the most from the automatic calculation. We hypothesize this is the case due to losses of precision in the intermediate steps of the automatic method. Nevertheless, we confirm that all RIMFoGraD networks tested produce  $T_2$  maps without any significant loss of precision from the ones produced by RIMs. Thus, we can use the least precise dictionary to achieve the best time-performance without loss of accuracy.

To evaluate the legitimacy of using dictionaries with a lower  $T_2$  value range than the values encountered during the training process, we compared the  $T_2$  maps obtained from the *in vivo* dataset between the RIM and RIMFoGraDs with different dictionaries (figure 4.26(b)). We verified that none of the  $T_2$  maps produced accurately describes the  $T_2$  values of the tissues. We conclude that the dictionaries in the RIMFoGraD must match the range of values observed during the network training.



(a)  $T_2$  dictionaries created for  $T_2$  values from 0 ms to 2000 ms;  $B_1$  dictionaries created for  $B_1$  values from 0.6 to 1.4.



(b)  $T_2$  dictionaries created for  $T_2$  values from 0 ms to 300 ms;  $B_1$  dictionaries created for  $B_1$  values from 0.6 to 1.4.

Figure 4.26:  $T_2$  maps estimated from one image of the *in vivo* testing dataset with the RIM network and with the RIMFoGraD networks for varying dictionaries with decreasing precision, and difference maps between the RIM and respective RIMFoGraD.

Since the precision of the dictionaries in the RIMFoGraD networks evaluated does not affect the quality of the results, we selected the network with the dictionary with smallest precision (1 ms in  $T_2$  and

0.01 in  $B_1$ ) as our final model. Thus, we conclude that RIMFoGraD can accurately produce a  $320 \times 320$  pixel map from 32 weighted images in  $4.16 \pm 0.10$  s, and it is 25% faster than the dictionary matching method with a dictionary with  $T_2$  ranging from 0-300 ms and precision 2 ms, and the same  $B_1$  range and precision.

The RIMFoGraD is also more reliable than the RIM when applied to protocols that eventually promote a larger number of inference steps. Figure 4.27 compares the average time needed to produce a  $160 \times 160$  pixel map between the two methods as the number of inference steps increases. An increase from 2 to 8 inference steps results in a 3.9% slower RIM and a 2.9% slower RIMFoGraD.

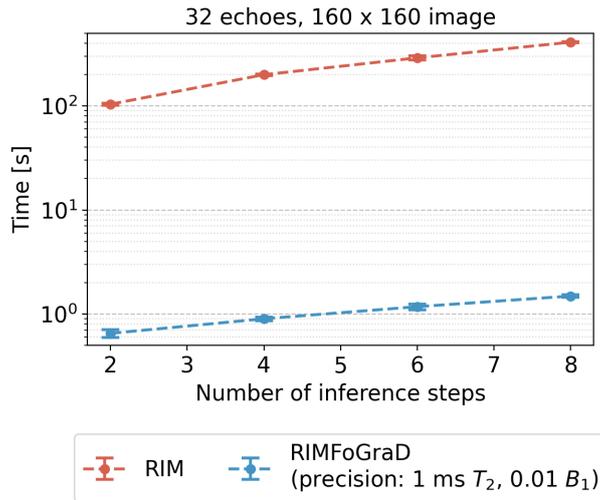


Figure 4.27: Average time needed to obtain a  $T_2$   $160 \times 160$  map from a set of squared weighted images with varying dimensions, for a protocol with 32 echoes, with the RIM and the RIMFoGraD. Points and error bars result from the mean and standard deviation over set of 10 repeated measurements, respectively.

Finally, figure 4.28 is included as a comparison of the average time needed to produce  $T_2$  maps with increasing dimensions between: the exponential fitting method, the standard EPG dictionary matching method, the RIM with the exponential signal model, the RIM with the EPG signal model considering the  $B_1$  field and the RIMFoGraD. When estimating a  $320 \times 320$  pixel map from 32 echoes, the RIMFoGraD is 380% faster than the RIM and 80% faster than the dictionary matching technique, and estimates  $T_2$  maps that are more accurate than the exponential RIM with only 1.2 s of difference in the estimation time.

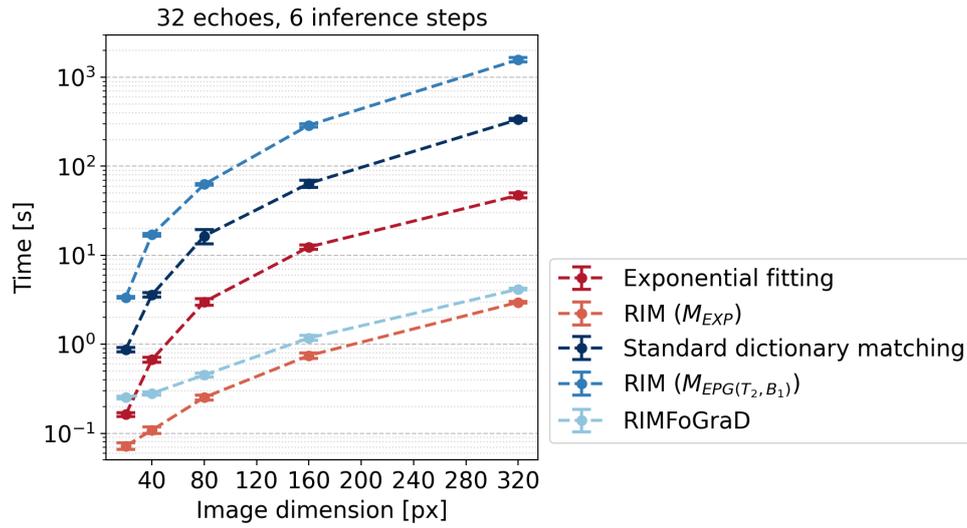


Figure 4.28: Average time required to estimate a  $T_2$  map with increasing dimensions, for a protocol with 32 echoes, with the  $M_{EXP}$  RIM, the  $M_{EPG(T_2, B_1)}$  RIM, the RIMFoGraD, exponential fitting and dictionary matching. Points and error bars result from the mean and standard deviation over set of 10 repeated measurements, respectively.



# Chapter 5

## Conclusions

This chapter presents a discussion of this project's findings and suggestions of future work.

### 5.1 Achievements

In this work, the EPG signal model was successfully implemented into the RIM to produce more accurate  $T_2$  maps compared with state-of-the-art methods admitting the monoexponential approximation. With RIMFoGraD, the RIM's ability to learn the inference process was combined with the dictionary matching technique's speed to maintain the RIM's performance when applied to a more complex signal model. We verified that the RIM is not suitable for complex signal models due to the need to calculate the forward model and respective gradients repeatedly for a chosen number of inference steps, whereas the RIMFoGraD preserved the performance of a RIM implemented with simple signal models, and was able to estimate a  $T_2$  map 380-fold faster than a RIM implemented with the EPG framework. Seeing that the RIMFoGraD infers parametric maps using pre-calculated signal evolutions, it is possible to add as much complexity to the signal model as desired without a loss of estimation speed, while taking advantage of RIM's low tendency to overfit and high capability of estimating the inference process. Furthermore, we verified that the time required to produce a  $T_2$  map with the RIMFoGraD does not increase as steeply with the map's image dimensions as with the other state-of-the-art methods, which constitutes a major advantage when applying these techniques to a clinical setting.

While the RIMFoGraD can estimate  $T_2$  maps in a reduced time, the accuracy of the maps estimated on simulated data was inferior to the state-of-the-art dictionary matching technique for data with large SNR, presumably due to the difficulties in estimating the  $B_1$  field map. However, the RIMFoGraD was slightly more robust to datasets strongly contaminated with noise and with larger intra-tissue  $T_2$  variability. Evaluating the estimated maps on *in vivo* data is complicated by the fact that the ground-truth map of the data is not known, which limits the analysis. For this reason, we cannot easily assess whether the dictionary matching estimate or the RIMFoGraD estimate are indeed accurate on *in vivo* datasets, especially when admitting the presence of noise: the difference of the estimated maps between the two methods may result from either one method being more robust to noise, or one method producing less

accurate estimates. Nevertheless, we confirmed the RIMFoGraD estimates maps with a similar range of values of the state-of-the-art dictionary matching methods. Caution is needed when comparing the RIMFoGraD's performance with the dictionary matching technique: we believe our dictionary matching implementation may be further optimized with parallel processing tools, for instance, which would make this method more competitive time-wise, depending on the hardware available to the user. Moreover, the larger the error considered for the RIMFoGraD's estimates, the smaller the dictionary that produces an equivalent map will be, and the faster the dictionary matching technique becomes.

A minor limitation of the RIMFoGraD is the time-consuming setup required for each acquisition protocol. Not only does this method share the deep learning tools' need to train a new network for each new acquisition protocol, but it also requires the creation of dictionaries containing the signal evolution and partial derivatives for a desired combination of  $T_2$  and  $B_1$  values, which may burden the user with the need of expensive hardware. In this work, the RIMFoGraD was trained using the RIM framework, but it is possible that the training time can be reduced when also taking advantage of dictionaries during that stage. However, additional studies are needed to confirm that this does not lead to a precision loss in the estimated  $T_2$  maps. Further, we verified that the optimal RIM hyper-parameters of our protocol varied from those applied in the literature [4], which suggests new validation tests may need to be performed when changing the signal acquisition protocol. In particular, the RIM benefits from training with larger minibatches as the forward signal model complexity increases. We hypothesize this may also be true when changing the ETL of the protocol, which would explain why the tests in this work promoted training with a larger number of patches per minibatch than those reported for similar applications with a shorter ETL. Nevertheless, the training process and the dictionary creation only need to be performed once per protocol, and we believe this is a good compromise for the short inference times.

The choice of hyper-parameters was accomplished with an analysis of performance metrics across a number of different datasets, but it is unclear whether these metrics are a good evaluation of the quality of the estimated maps. We verified that distinct performance metrics promote different image characteristics, which may not always correlate to a good quality image. Furthermore, selecting hyper-parameters with performance scores entails choosing a metric to give preference to in detriment of the other metrics, and taking into account the presence of outliers, which increases the uncertainty of the metrics' reliability. Validation results suggest networks trained with noiseless sets of weighted images are more robust to noise in the evaluation images, yet the networks that were trained with images with a selected noise standard deviation demonstrated to be unable to estimate maps from noiseless data with the same level of accuracy, and produced more accurate maps on datasets that matched the noise used during training. Thus, it remains unclear whether the RIM should be trained with noiseless weighted images, or images with a range of distinct acquisition noise amounts.

This work's implementation also considered a number of approximations. Firstly, we implemented the RF pulses while admitting the small tip angle approximation, where the Bloch equations are used to compute a pulse via the Fourier Transform of the desired slice profile. When using large flip angles, this results in excitation error that could be reduced by decomposing each pulse into multiple pulses applied during shorter time intervals. The result of each sub-RF pulse can once again be calculated using the

tip angle approximation [30]. Secondly, the EPG implementation also requires the user to select a  $T_1$  value valid for all tissues. While we confirmed the chosen  $T_1$  had a negligible effect on the estimated  $T_2$  maps of the brain, this may not remain true when acquiring images of other anatomical regions. Finally, the objective function of the optimization process considered all estimates from the inference stage to be equally meaningful, despite the latter estimates being more accurate than the former. We believe the accuracy of the estimated maps may somewhat improve when giving more weight to the latter estimates in the objective function [16].

Because the RIM has a high generalization capability, it can be trained with simulated data that can contain any combination of parameter values and be arbitrarily large [4], yet this encumbers the user with additional decisions on the ground-truth  $T_2$  values of each tissue. We verified that the RIM is somewhat sensitive to the chosen distribution of  $T_2$  relaxation times, which may be a limitation when considering the reported  $T_2$  values of the same tissue in the literature vary depending on the signal model considered, the number of sample points, the presence of noise, differences in hardware and sequence design protocol and [31]. Nevertheless, given the lack of MRI annotated datasets that can be used in machine learning applications, we consider the ability to train with simulated data an advantage over other deep learning methods.

## 5.2 Future Work

Future work may focus on further improving the RIMFoGraD parametric map estimation with the EPG framework. It was verified the RIM was somewhat able to estimate an additional  $B_1$  map in *in vivo* data to produce more accurate  $T_2$  maps without altering the network configuration or employing regularization. Follow-up studies may focus on adjusting the RNNCell configuration and employing various forms of regularization to improve the accuracy of maps estimated with the EPG framework for a wider range of  $B_1$  fields. Concurrently, it may be worthwhile to compare other schemes for the creation of the  $B_1$  ground-truth fields, namely using a uniform field per patch, or increasing the range of  $B_1$  values observed in a patch to improve the network's performance on images with a high  $B_1$  contrast.

Secondly, given the network's ability to generalize to unseen data, the training simulated images may be created with PD and  $T_2$  values pertaining to other anatomical regions so that the RIM is able to estimate parametric maps on other anatomical regions while training with brain data. Future work should also focus on studying the ability of the RIMFoGraD to detect pathologies unseen during training.

Finally, we propose applying the RIMFoGraD framework directly to the acquisition data from the MRI scanner instead of their image reconstruction, by including the Fourier Transform on the signal model. The dictionaries will contain the k-space measurements obtained for each combination of  $T_2$  and  $B_1$ , and the corresponding partial derivatives. The RIMFoGraD may also be combined with other acceleration techniques such as compressed sensing.



# Bibliography

- [1] Y. Wu, Y. Ma, Y. Kee, N. Kovalchuk, D. Capaldi, H. Ren, S. Hancock, E. Chang, M. Alley, J. Pauly, J. Du, S. Vasanaawala, and L. Xing. Quantitative parametric mapping of tissues properties from standard magnetic resonance imaging enabled by deep learning. *arXiv preprint arXiv:2108.04912*, 2021.
- [2] L. Feng, D. Ma, and F. Liu. Rapid MR relaxometry using deep learning: An overview of current techniques and emerging trends. *NMR in Biomedicine*, 2020. ISSN 10991492. doi: 10.1002/nbm.4416.
- [3] F. Liu, L. Feng, and R. Kijowski. MANTIS: Model-augmented neural network with incoherent k-space sampling for efficient MR parameter mapping. *Magnetic Resonance in Medicine*, 82:174–188, 7 2019. ISSN 15222594. doi: 10.1002/mrm.27707.
- [4] E. Sabidussi, S. Klein, M. Caan, S. Bazrafkan, A. den Dekker, J. Sijbers, W. Niessen, and D. Poot. Recurrent inference machines as inverse problem solvers for mr relaxometry. *Medical Image Analysis*, 74:102220, 2021. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2021.102220>. URL <https://www.sciencedirect.com/science/article/pii/S1361841521002656>.
- [5] N. Ben-Eliezer, D. Sodickson, and K. Block. Rapid and accurate t2 mapping from multi spin echo data using bloch-simulation-based reconstruction. *Magnetic Resonance in Medicine*, 73:809–817, 02 2015. doi: 10.1002/mrm.25156.
- [6] M. Weigel. Extended phase graphs: Dephasing, RF pulses, and echoes - pure and simple. *Journal of Magnetic Resonance Imaging*, 41:266–295, 2 2015. ISSN 15222586. doi: 10.1002/jmri.24619.
- [7] Y. Jun, H. Shin, T. Eo, T. Kim, and D. Hwang. Deep model-based magnetic resonance parameter mapping network (dopamine) for fast t1 mapping using variable flip angle method. *Medical Image Analysis*, 70, 5 2021. ISSN 13618423. doi: 10.1016/j.media.2021.102017.
- [8] M. Zibetti, A. Sharafi, K. Hammernik, F. Knoll, and R. Regatte. Comparing learned variational networks and compressed sensing for T1 $\rho$  mapping of knee cartilage. 2018. ISMRM Workshop on Machine Learning, Part II ; Conference date: 25-10-2018 Through 28-10-2018.
- [9] H. Jeelani, Y. Yang, R. Zhou, C. M. Kramer, M. Salerno, and D. S. Weller. A myocardial T1-Mapping framework with recurrent and u-net convolutional neural networks. *Proceedings - International*

- Symposium on Biomedical Imaging*, 2020-April:1941–1944, 4 2020. ISSN 19458452. doi: 10.1109/ISBI45749.2020.9098459.
- [10] A. S. Chaudhari, Z. Fang, F. Kogan, J. Wood, K. J. Stevens, E. K. Gibbons, J. H. Lee, G. E. Gold, and B. A. Hargreaves. Super-resolution musculoskeletal MRI using deep learning. *Magnetic Resonance in Medicine*, 80:2139–2154, 11 2018. ISSN 15222594. doi: 10.1002/mrm.27178.
- [11] Y. Gao, M. Cloos, F. Liu, S. Crozier, G. B. Pike, and H. Sun. Accelerating quantitative susceptibility and R2\* mapping using incoherent undersampling and deep neural network reconstruction. *NeuroImage*, 240, 10 2021. ISSN 10959572. doi: 10.1016/j.neuroimage.2021.118404.
- [12] O. Cohen, B. Zhu, and M. S. Rosen. MR fingerprinting deep reconstruction network (DRONE). *Magnetic Resonance in Medicine*, 80:885–894, 9 2018. ISSN 15222594. doi: 10.1002/mrm.27198.
- [13] Z. Fang, Y. Chen, M. Liu, L. Xiang, Q. Zhang, Q. Wang, W. Lin, and D. Shen. Deep learning for fast and spatially-constrained tissue quantification from highly-accelerated data in magnetic resonance fingerprinting. *IEEE Transactions on Medical Imaging*, 38(10):2364–2374, Oct 2019. doi: 10.1109/tmi.2019.2899328. URL <http://dx.doi.org/10.1109/TMI.2019.2899328>.
- [14] H. Liu, O. Heide, C. Berg, and A. Sbrizzi. Fast and accurate modeling of transient-state gradient-spoiled sequences by recurrent neural networks. 08 2020.
- [15] T. Yu, E. J. Canales-Rodríguez, M. Pizzolato, G. F. Piredda, T. Hilbert, E. Fisch-Gomez, M. Weigel, M. Barakovic, M. Bach Cuadra, C. Granziera, T. Kober, and J.-P. Thiran. Model-informed machine learning for multi-component t2 relaxometry. *Medical Image Analysis*, 69:101940, 2021. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2020.101940>. URL <https://www.sciencedirect.com/science/article/pii/S1361841520303042>.
- [16] K. Lønning, P. Putzky, J.-J. Sonke, L. Reneman, M. W. Caan, and M. Welling. Recurrent inference machines for reconstructing heterogeneous mri data. *Medical Image Analysis*, 53:64–78, Apr. 2019. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2019.01.005>. URL <https://www.sciencedirect.com/science/article/pii/S1361841518306078>.
- [17] X. Wang, Z. Tan, N. Scholand, V. Roeloffs, and M. Uecker. Physics-based reconstruction methods for magnetic resonance imaging. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379, 6 2021. ISSN 1364503X. doi: 10.1098/rsta.2020.0196.
- [18] R. M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. *ArXiv*, abs/1912.05911, 2019.
- [19] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL <https://aclanthology.org/W14-4012>.

- [20] P. Putzky and M. Welling. Recurrent inference machines for solving inverse problems. *ArXiv*, abs/1706.04008, 2017.
- [21] K. J. Layton, M. Morelande, D. Wright, P. M. Farrell, B. Moran, and L. A. Johnston. Modelling and estimation of multicomponent  $t_2$  distributions. *IEEE Transactions on Medical Imaging*, 32(8): 1423–1434, 2013. doi: 10.1109/TMI.2013.2257830.
- [22] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, and A. C. Evans. Brainweb: Online interface to a 3d mri simulated brain database. *NeuroImage*, 1997.
- [23] R.-S. Kwan, A. Evans, and G. Pike. Mri simulation-based evaluation of image-processing and classification methods. *IEEE Transactions on Medical Imaging*, 18(11):1085–1097, 1999. doi: 10.1109/42.816072.
- [24] R. K. S. Kwan, A. C. Evans, and G. B. Pike. An extensible mri simulator for post-processing evaluation. pages 135–140, 1996.
- [25] A. Tristán-Vega and S. Aja-Fernández. Design and construction of a realistic dwi phantom for filtering performance assessment. pages 951–958, 2009.
- [26] A. Calisto De Freitas, A. Gaspar, I. Sousa, R. A. G. Teixeira, J. Hajnal, and R. Nunes. Improving  $b_1 +$  parametric estimation in the brain from multispin-echo sequences using a fusion bootstrap moves solver. *Magnetic resonance in medicine*, 86(5):2426–2440, Nov. 2021. ISSN 0740-3194. doi: <https://doi.org/10.1002/mrm.28878>. Funding Information: The authors acknowledge the following funding sources: Portuguese Foundation for Science and Technology (FCT—IF/00364/2013, UID/EEA/50009/2019, SFRH/BD/120006/2016, PTDC/EMD-EMD/29686/2017, UIDB/50009/2020), and POR Lisboa 2020 (LISBOA-01-0145-FEDER-029686). Publisher Copyright: © 2021 International Society for Magnetic Resonance in Medicine.
- [27] J. Sijbers, A. J. den Dekker, M. Verhoye, E. R. Raman, and D. V. Dyck. Optimal estimation of T2 maps from magnitude MR images. In K. M. Hanson, editor, *Medical Imaging 1998: Image Processing*, volume 3338, pages 384 – 390. International Society for Optics and Photonics, SPIE, 1998. doi: 10.1117/12.310915. URL <https://doi.org/10.1117/12.310915>.
- [28] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.
- [29] A. Mason, J. Rioux, S. E. Clarke, A. Costa, M. Schmidt, V. Keough, T. Huynh, and S. Beyea. Comparison of objective image quality metrics to expert radiologists’ scoring of diagnostic quality of mr images. *IEEE Transactions on Medical Imaging*, 39(4):1064–1072, 2020. doi: 10.1109/TMI.2019.2930338.
- [30] C. S. Aigner, C. Clason, A. Rund, and R. Stollberger. Efficient high-resolution rf pulse design applied to simultaneous multi-slice excitation. *Journal of Magnetic Resonance*, 263:33–44, 2016. ISSN

1090-7807. doi: <https://doi.org/10.1016/j.jmr.2015.11.013>. URL <https://www.sciencedirect.com/science/article/pii/S109078071500302X>.

- [31] J. Zavala, S. Bricq, C. Acqutter, F. Brunotte, P. Walker, and A. Lalande. What are normal relaxation times of tissues at 3 tesla? *Magnetic Resonance Imaging*, 35, 08 2016. doi: 10.1016/j.mri.2016.08.021.

# Appendix A

## Statistics

In this chapter, we provide a demonstration of the log-likelihood expression defined in equation 3.6. Let  $\mathbf{p}$  be the population parameter we aim to estimate from a set of  $I$  observations  $\mathbf{m}$  with a probability distribution  $f$ . In the case of MR relaxometry,  $\mathbf{p}(\mathbf{x})$  denotes the parametric maps, while  $\mathbf{m}(\mathbf{x})$  corresponds to the set of weighted images obtained from the scanner measurements at each spatial coordinate  $\mathbf{x}$ . The likelihood function that corresponds to the probability of observing  $\mathbf{m}$  when the population parameter is  $\mathbf{p}$  is expressed by  $f(\mathbf{m}|\mathbf{p})$ .

The population parameter  $\mathbf{p}$  can be estimated with the maximum likelihood estimator (MLE) method following equation A.1, where  $\mathcal{L}$  stands for the negative log-likelihood.

$$\hat{\mathbf{p}} = \arg \max_{\mathbf{p}} f(\mathbf{m}|\mathbf{p}) = \arg \max_{\mathbf{p}} \log(f(\mathbf{m}|\mathbf{p})) = \arg \min_{\mathbf{p}} \mathcal{L}(\mathbf{m}|\mathbf{p}) \quad (\text{A.1})$$

Alternatively, we can use the MAP estimator (equation A.2) to find  $\mathbf{p}$ . This technique consists instead on maximizing the posterior probability  $f(\mathbf{p}|\mathbf{m})$ , which can be calculated with the Bayes theorem. The MAP estimate takes into account the prior probability  $f(\mathbf{p})$  and coincides with the MLE when this probability is uniform. Notice that  $f(m)$  is always positive and does not play a role in the optimization process.

$$\hat{\mathbf{p}} = \arg \max_{\mathbf{p}} f(\mathbf{p}|\mathbf{m}) = \arg \max_{\mathbf{p}} f(\mathbf{m}|\mathbf{p})f(\mathbf{p}) = \arg \max_{\mathbf{p}} \{\log(f(\mathbf{m}|\mathbf{p})) + \log(f(\mathbf{p}))\} \quad (\text{A.2})$$

Let  $M$  be the forward model that transforms the population parameters into the observations such that  $\mathbf{m} = M(\mathbf{p}) + \varepsilon$ , with  $\varepsilon$  representing the noise associated with the measurements. Let us further assume the observations  $\mathbf{m}$  are corrupted by independent and identically distributed measurement errors that follow a normal distribution with variance  $\sigma^2$ . The observations  $\mathbf{m}$  will themselves follow a normal distribution defined in A.3.

$$f(\mathbf{m}) = (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{1}{2} \frac{(\mathbf{m} - M(\mathbf{p}))^2}{\sigma^2}\right) \quad (\text{A.3})$$

The negative log-likelihood function  $\mathcal{L}(\mathbf{m}|\mathbf{p})$  is then calculated in A.4.

$$\begin{aligned}
f(\mathbf{m}|\mathbf{p}) &= \prod_{i=1}^I f(\mathbf{m}_i|\mathbf{p}) \\
&= \prod_{i=1}^I (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{1}{2} \frac{(\mathbf{m}_i - M(\mathbf{p}))^2}{\sigma^2}\right) \\
&= (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^I (\mathbf{m}_i - \mathbf{p})^2\right) \\
\Rightarrow \log f(\mathbf{m}|\mathbf{p}) &= \log\left((2\pi\sigma^2)^{-n/2}\right) + \log\left(\exp\left(-\frac{1}{2} \frac{(\mathbf{m} - M(\mathbf{p}))^2}{\sigma^2}\right)\right) \quad (\text{A.4}) \\
&= -\frac{I}{2} \log(2\pi) - \frac{I}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^I (\mathbf{m}_i - M(\mathbf{p}))^2 \\
\Rightarrow \mathcal{L}(\mathbf{m}|\mathbf{p}) &= \frac{I}{2} \log(2\pi) + \frac{I}{2} \log(\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=1}^I (\mathbf{m}_i - M(\mathbf{p}))^2
\end{aligned}$$

The minimization of  $\mathcal{L}(\mathbf{m}|\mathbf{p})$  with respect to the unknown parameter  $\mathbf{p}$  is equivalent to least squares fitting (equation A.5).

$$\hat{\mathbf{p}} = \arg \min_{\mathbf{p}} \sum_{i=1}^I \|M(\mathbf{p}) - \mathbf{m}_i\|^2 \quad (\text{A.5})$$

The intermediate results of this project were displayed in the International Society for Magnetic Resonance in Medicine's 2022 Iberian Chapter as a poster presentation, included in the next page.

Catarina Carvalho\*<sup>1</sup>, Teresa M Correia<sup>2,3</sup>, Rita G Nunes<sup>1</sup>

1. ISR-Lisboa and Department of Bioengineering, Instituto Superior Técnico – Universidade de Lisboa, Lisbon, Portugal; 2. School of Biomedical Engineering Imaging Sciences, King's College London, United Kingdom; 3. Centre for Marine Sciences - CCMAR, Faro, Portugal

\*catarina.neves.carvalho@tecnico.ulisboa.pt

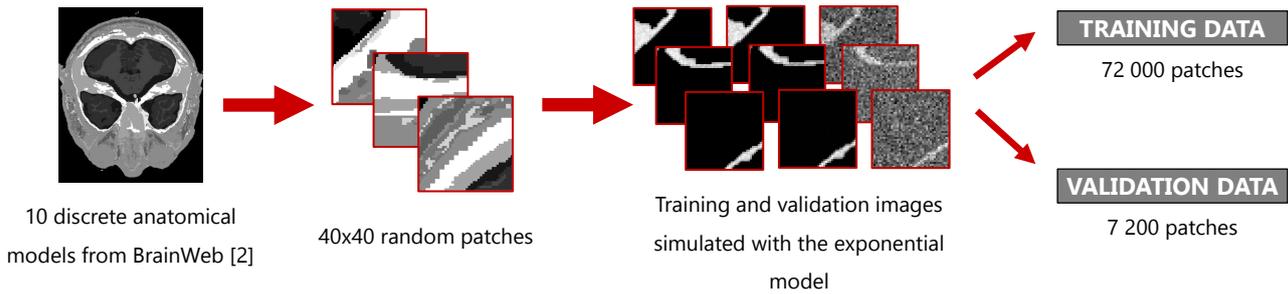
## INTRODUCTION

**Quantitative MRI** requires a time-consuming parameter fitting operation. In this work, we attempt to reduce the fitting time in T2 maps with a model-based recurrent inference machine (RIM), which incorporates the MRI signal model into its framework.

The methodology developed by Sabidussi et al [1] was adapted to a distinct multi spin-echo (MSE) acquisition protocol, and the results were compared with standard exponential fitting.

**GOAL:** To obtain T2-maps of the brain from MSE acquisitions using a RIM neural network.

## METHODS

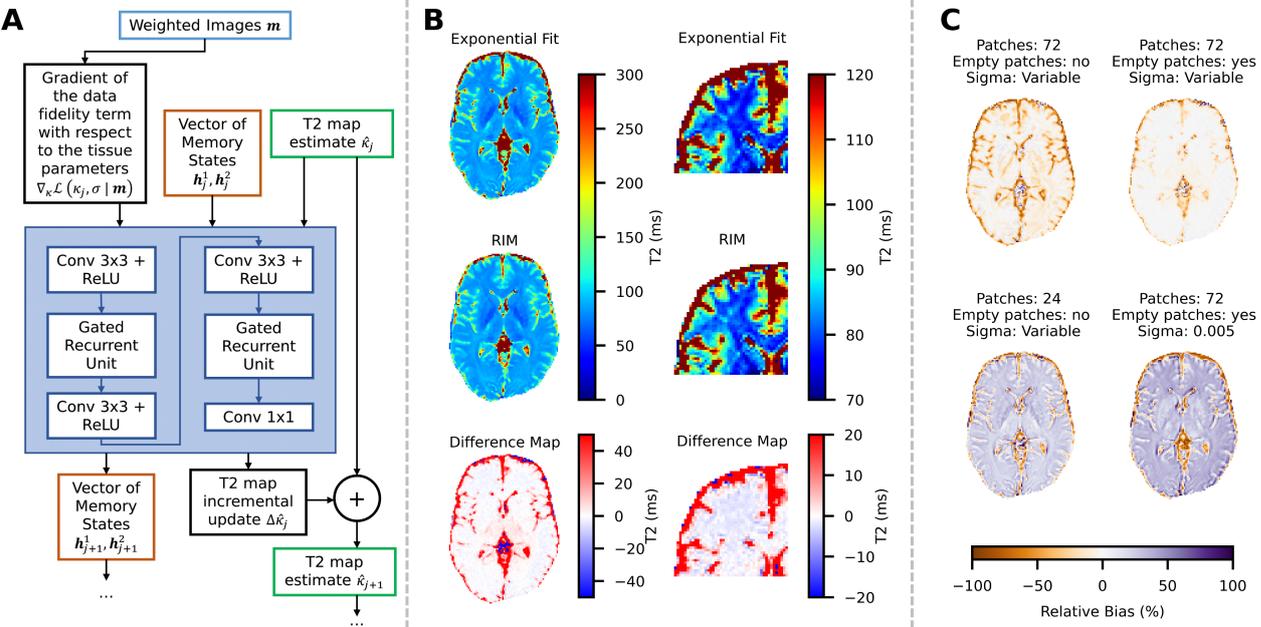


SAVE POSTER

### HYPERPARAMETER TESTS

- Number of patches in each batch (24, 72 and 200);
- Inclusion/absence of empty patches corresponding to background noise;
- Amount of simulated noise in the images (standard deviation with a fixed value for all patches or chosen from a standard log-normal distribution).

## RESULTS AND DISCUSSION



**Optimal training hyperparameters:**

- **72 patches** per batch;
- Including patches without anatomical information;
- Acquisition noise with a **variable standard deviation** across patches.

Generating a T2 map with a trained RIM took an average of 0.92 s across 15 experiments, **3.6 times faster** than performing exponential fitting.

## CONCLUSION

- T2 maps obtained with the RIM closely resemble the maps obtained by exponential fitting for grey and white matter.
- Training hyperparameters strongly influence the quality of the maps and may need to be adjusted when applying the RIM to different acquisition protocols.

## FUTURE WORK

- Modify the RIM MRI signal model from a monoexponential curve into the Extended Phase Graph representation [3] for better accuracy.
- Include *in vivo* images in the training stage.