ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

INGEGNERIA ELETTRONICA, TELECOMUNICAZIONI E TECNOLOGIE DELL'INFORMAZIONE

Ciclo 37

Settore Concorsuale: 09/E3 - ELETTRONICA

Settore Scientifico Disciplinare: ING-INF/01 - ELETTRONICA

NEURAL COMPRESSED SENSING FOR MRI AND ECG DATA ACQUISITION AND COMPRESSION

Presentata da: Filippo Martinini

Coordinatore Dottorato

Davide Dardari

Supervisore

Riccardo Rovatti

Esame finale anno 2025

“기억하네”

Hyunjin

ALMA MATER STUDIORUM — UNIVERSITY OF BOLOGNA

Abstract

School of Engineering
Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi" (DEI)

Doctor of Philosophy

Neural Compressed Sensing
for MRI and ECG Data Acquisition and Compression

by Filippo MARTININI

Biomedical signal acquisition often occurs in resource-constrained environments, necessitating advanced encoding or acquisition algorithms. In such contexts, Compressed Sensing (CS) offers a promising solution but faces performance challenges, especially in practical implementations.

This dissertation explores the integration of Neural Network and Compressed Sensing techniques for the efficient acquisition and compression of biomedical signals, specifically focusing on Electrocardiogram (ECG) and Magnetic Resonance Imaging (MRI) data.

For ECG, two innovative approaches are presented. The first approach introduces a data-driven binary encoding to develop a lightweight encoding mechanism. The second approach introduces an adaptive, incremental compression scheme that uses a performance predictor to dynamically adjust the number of transmitted measurements.

For MRI data acquisition, the work delves into advanced undersampling techniques. The first part builds on the state-of-the-art LOUPE architecture, incorporating CS-derived constraints into the training framework to improve the quality of reconstructed MRI images. The second part, introduces the concept of incremental acquisition, where the number of acquired k-space samples is dynamically adjusted based on real-time quality assessments.

This dissertation demonstrates how combining model-based CS with data-driven DNN holds the potential to revolutionize acquisition methodologies for biomedical signals, making advanced diagnostics efficient even in resource-hungry settings.

Contents

List of Figures

List of Tables

List of Acronyms

ECG	**E**lectrocardiogram
CS	**C**ompressed **S**ensing
DNN	**D**eep **N**eural **N**etwork
MRI	**M**agnetic **R**esonance **I**maging
AFE	**A**nalog **F**ront **E**nd
MCU	**M**icrocontroller **U**nit
BPDN	**B**asis **P**ursuit **D**enoising
CNN	**C**onvolutional **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
TCSSO	**T**rained **C**ompressed **S**ensing with **S**upport **O**racle
PCR	**P**robability of **C**orrect **R**econstruction
SNR	**S**ignal-to-**N**oise **R**atio
ISNR	**I**ntrinsic **S**ignal-to-**N**oise **R**atio
RSNR	**R**econstructed **S**ignal-to-**N**oise **R**atio
MAC	**M**ultiply-**AC**cumulate (operation)
BLE	**B**luetooth **L**ow **E**nergy
SPGL1	**S**pectral **P**rojected **G**radient for **L1** minimization
RF	**R**adio **F**requency
TE	**E**cho **T**ime
TR	**R**epetition **T**ime
DAQ	**D**ata **A**cquisition
TV	**T**otal **V**ariation
GAN	**G**enerative **A**dversarial **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
SVD	**S**ingular **V**alue **D**ecomposition
LOUPE	**L**earning-based **O**ptimization of the **U**ndersampling **P**attern
MoDL	**M**odel-based **D**eep **L**earning
AUC	**A**rea **U**nder the **C**urve
ROC	**R**eceiver **O**perating **C**haracteristic
FPR	**F**alse **P**ositive **R**ate
TPR	**T**rue **P**ositive **R**ate
MSE	**M**ean **S**quared **E**rror
MAE	**M**ean **A**bsolute **E**rror
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio

SSIM	**S**tructural **S**imilarity **I**ndex
NR-IQA	**N**o-**R**eference **I**mage **Q**uality **A**ssessment
4S	**S**elf **s**hrinking of the **S**igmoid **S**lope

List of Symbols

$\mathbb{N}$	set of the natural numbers
$\mathbb{Z}$	set of the integer numbers
$\mathbb{R}$	set of the real numbers
$.^\top$	transposition operator
$\Pr\{\cdot\}$	probability
$\mathbf{E}[\cdot]$	expectation
$\mathrm{diag}(\times)$	diagonal matrix with $\times$ as diagonal vector
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert_p$	ℓ_p norm
cov	covariance matrix
$.^\dagger$	Moore-Penrose pseudo-inversion operator
$\boldsymbol{x}$	input signal
$\hat{\boldsymbol{x}}$	reconstructed signal
$\boldsymbol{y}$	encoded signal
n	input signal dimension
m	encoded signal dimension
k	number of non-null elements in the sparse representation
$\boldsymbol{\xi}$	sparse representation
D	sparsity basis
A	encoding matrix
$\hat{\xi}$	estimated signal in the sparse representation
supp	function to extract non-null elements from sparse representation
L	loss function
U	random matrix extracted from uniform distribution
$\sigma(\cdot)$	sigmoid function
t, s	slope parameters for the sigmoid function
$\mathcal{F}$	Fourier transform
$\mathcal{F}^{-1}$	inverse Fourier transform
$\boldsymbol{M}$	binary mask for undersampling
r	acquisition rate
R	undersampling ration
$\boldsymbol{\delta}$	error
ρ	Pearson correlation function
$Q(\cdot)$	quality function

$\hat{Q}(\cdot)$	quality estimation function

Introduction

Biomedical signal acquisition and compression are pivotal in modern healthcare, enabling the effective diagnosis, monitoring, and treatment of various medical conditions. Advances in medical technology have made it possible to collect diverse types of physiological data from the human body, such as electrical, biochemical, and mechanical signals. These signals provide critical information about the state of different organs and systems, facilitating early detection of diseases and guiding therapeutic interventions.

Biomedical signals are captured using a wide range of sensors and devices, tailored to the nature of the physiological process being monitored. These sensors may be embedded in wearable devices, placed on the skin, or even positioned inside the body to collect internal measurements. The acquisition process varies significantly based on the type of signal, the environment in witch it is recorded, and the intended purpose. For example, a wearable device for daily health monitoring must prioritize energy efficiency and convenience, whereas a clinical diagnostic setup may prioritize signal quality and detail [59].

To achieve this, compression techniques are a fundamental tool, helping to manage the large volumes of data generated by prolonged monitoring and high-resolution imaging. However, the requirements for compression can differ significantly depending on the specific biomedical application. Each scenario requires a tailored approach that takes into account the unique constraints and goals associated with the specific type of signal. In this thesis, we focus on the analysis and optimization of biomedical signal acquisition and compression in two specific scenarios where resources are limited or encoding requires to be efficient: Electrocardiogram (ECG) compression and Magnetic Resonance Imaging (MRI) acquisition optimization.

Compressed Sensing (CS) is well-regarded for its ability to alleviate the computational demands of encoding. This characteristic makes CS particularly suitable for scenarios where acquisition or encoding is constrained, such as low-energy or low-speed acquisition systems. However, CS has traditionally been confined to solving optimization problems, witch imposes certain limitations on its applicability. The advent of Machine Learning, and Deep Neural Networks in particular, has shifted the landscape towards data-driven approaches.

In this thesis, we aim to advance the current research on integrating CS with DNNs, with the goal of combining the strengths of both methodologies. This proposed framework seeks to develop encoding and acquisition techniques that incorporate the rigor of CS's prior-based models while leveraging the adaptive, data-driven capabilities of DNNs. This dissertation is organized as follows:

Part I focuses on ECG, a signals representing the electrical activity of the heart over time. In Chapter 1 we give an overview of ECG acquisition and transmission. ECG sensors are deployed in low-power, resource-constrained environments, where efficient data compression and transmission are crucial to extend battery life. To overcome this difficulties, in Chapter 2 we develop a binary compressed sensing technique, where DNN are used to optimize binary sensing matrices. This way, only accumulation operations are used for encoding, drastically reducing the computational burden. Standard CS is leveraged to decode the transmitted signal, and both encoder and decoder are jointly optimized [49]. In Chapter 3 we introduce an iterative compression scheme, witch leverages neural performance predictors to adapt the number of transmitted measurements in a standard CS-based scheme. Through real-time adaptability, we achieve improved performances in terms of energy consumption [46].

Part II addresses MRI signals; high-resolution images of the internal structures of the human body. As better explained in Chapter 4, MRI are complex systems and acquisition occurs in a resource-rich setting, that does not face power or computational resources shortages. However, the challenge lies in accelerating the data acquisition process itself to improve patient comfort, reduce costs, and increase throughput. One effective way to optimize MRI acquisition is to reduce the number of samples collected, a strategy known as undersampling [39]. A process that can be viewed as a from of embedded compression, where the decoding part, starting from incomplete data, outputs complete high-resolution images.

This work follows [50, 48, 47, 51] and organically presents the following main contributions. In Chapters 5 and 7 we improve a state-of-the-art DNN-based MRI encoder-decoder system by integrating CS-based solutions in the training system [50] and in the model itself [48]. In Chapter 4 we designing a novel callback [47] to mitigate the backpropagation issue introduced by undersampling (designed as a quasi binary matrix with poor intrinsic derivability). Finally, in Chapter 8 an iterative acquisition scheme enables to decide at acquisition-time the number of samples to acquire, so to stop as soon as the estimated acquisition quality meets the minimum acquisition target [51].

Part I

Neural Compressed Sensing for ECG

Chapter 1

ECG Compression and Transmission

Electrocardiogram (ECG) monitoring is one of the most widely used techniques in healthcare, particularly for the prevention and diagnosis of cardiovascular diseases. The continuous and reliable monitoring of heart signals helps in early detection of conditions like arrhythmias and ischemia. With the proliferation of wearable health devices, efficient signal acquisition and transmission have become increasingly important, especially given the energy-constrained environments these devices often operate in. Wearable sensors need to ensure accurate data acquisition while minimizing power consumption and computational demands, witch presents unique challenges in the design and deployment of these systems.

Compressed Sensing (CS) has emerged as a powerful method to address these challenges by reducing the amount of data that needs to be acquired and transmitted. This not only minimizes energy consumption but also ensures that continuous monitoring can be maintained over long periods without frequent battery recharging. CS allows for efficient encoding of ECG signals by projecting them onto a lower-dimensional space while preserving enough information for successful reconstruction.

This chapter introduces to encoding and decoding techniques for ECG data based on CS. Further chapters will explore two advanced neural compressed sensing methods. The first approach focuses on binary compressed sensing, witch employs neural networks to optimize sensing matrices, thus reducing the complexity of the encoding process while ensuring high reconstruction quality. The second approach introduces an adaptive compression scheme using a performance predictor, witch dynamically adjusts the number of measurements based on signal complexity to optimize energy use.

1.1 The ECG Signal

ECG is one of the most widely studied biomedical signals for clinical purposes. Continuous and rigorous monitoring of ECG signals is critical in the prevention of unexpected cardiovascular diseases and health issues. This makes the acquisition and transmission of ECG signals a heavily researched area within biomedical applications, particularly given the hardware limitations associated with wearable and mobile health devices. The design of these systems requires special attention to reduce computational and transmission costs, especially at the sensor side, witch must be as efficient as possible to minimize energy consumption and computational overhead.

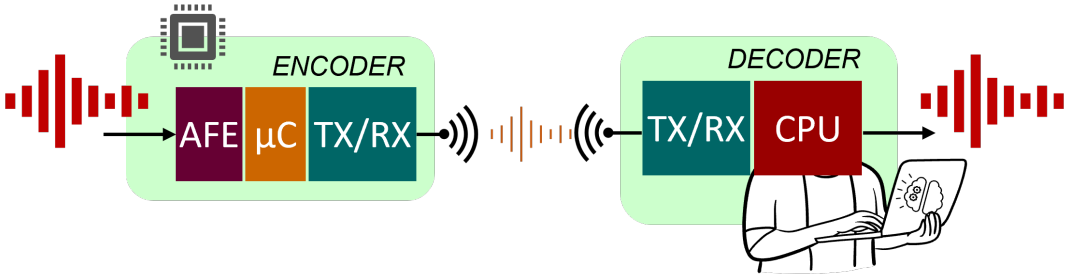


FIGURE 1.1: Scheme of a transmission setup with encoding and decoding stages.

In such systems, reducing power consumption while remaining within the limited computational capabilities of a microcontroller is one of the main objectives. This challenge has led to the development of various signal compression techniques, among witch Compressed Sensing (CS) stands out due to its ability to minimize encoding complexity. CS encodes a signal by projecting it onto random sensing waveforms, witch make up the sensing matrix. The number of waveforms (or measurements) can potentially be much lower than the original signal dimension, significantly reducing the data to be transmitted. At the encoder stage, this involves a simple matrix multiplication, making the operation lightweight in terms of power consumption and memory usage. On the other hand, decoding is often computationally demanding, as it involves solving an optimization problem to find the sparsest representation of the signal on a predefined basis. A standard method to reconstruct the encoded signal is Basis Pursuit Denoising [5, 63, 79, 58].

The design of a typical sensor node adheres to a well-established architectural structure [21], witch comprises three main components: (i) a sensing unit, often implemented as an Analog Frontend (AFE) with an integrated analog-to-digital converter, (ii) a Microcontroller Unit (MCU) responsible for processing and managing the digitized data, and (iii) a transmission stage that sends the processed output to a remote receiver (refer to Fig. 1.1). The overall energy consumption of the sensor node can be modeled as the cumulative energy cost of these individual components. Specifically, the energy requirements of the processing and transmission stages can be balanced against one another to achieve reduced power consumption by employing efficient and well-established compression algorithms.

Although CS lightens the computational load at the encoder, the decoder must solve a complex optimization problem to reconstruct the signal. This trade-off makes CS an appealing choice for low-power devices, where encoding complexity is the primary concern. However, the optimization involved in traditional CS reconstruction can be computationally prohibitive, and the performances might be not competitive with standard algorithms. This limitation has motivated the use of Deep Neural Networks (DNNs) to augment the optimization process in CS.

By leveraging the powerful approximation capabilities of DNNs, CS reconstruction can be made significantly faster and more adaptable. DNNs can learn the inherent characteristics of signals from data, allowing them to provide efficient, real-time reconstruction without the need for explicit optimization, thus bridging the gap between lightweight encoding and

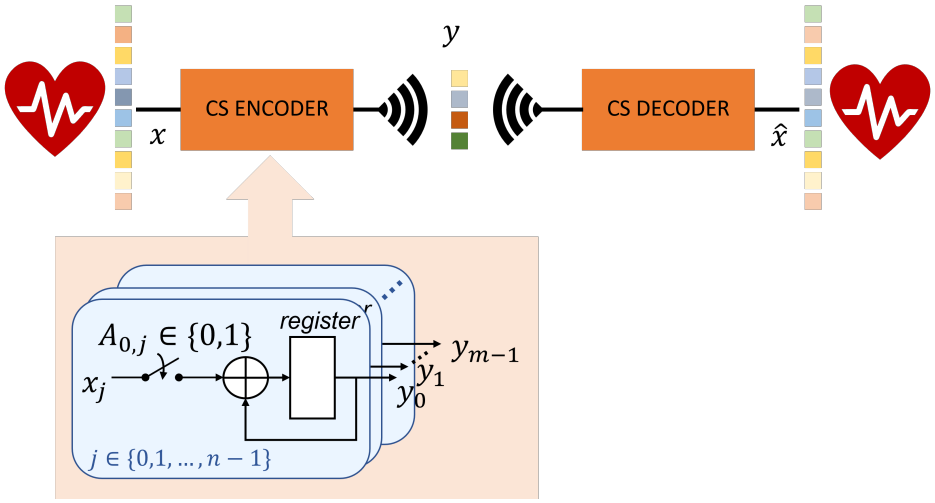


FIGURE 1.2: Compressed sensing framework incorporating a binary encoding phase

computationally intensive reconstruction.

This work explores two approaches that combine CS and DNN for ECG signal acquisition and compression, aimed at optimizing both encoding and transmission processes.

The first approach focuses on minimizing the complexity of the encoder by using a binary sensing matrix. Binary sensing matrices contain only 0s and 1s, witch drastically reduce the computation required during the encoding process. This results in a system where only accumulations are necessary to produce the compressed output, as illustrated in Fig. 1.2. Binary sensing matrices are advantageous in low-power environments because they reduce the multiply and accumulate operations with simple accumulation operations, minimizing the computational load.

The second approach tackles the problem of adapting the number of transmitted measurements dynamically based on the complexity of each signal instance. In real-world scenarios, the complexity of ECG signals can vary, and a fixed number of measurements may either be insufficient for complex signals or unnecessarily high for simpler ones. To address this, we propose an adaptive incremental encoding scheme where the number of transmitted measurements is adjusted in real-time.

1.2 Compressed Sensing

Consider a signal instance $x \in \mathbb{R}^n$, where n represents the number of samples in the signal acquired at the Nyquist rate. CS exploits the fact that many real-world signals, including ECGs, are sparse when expressed on a proper basis. A signal is said to be κ-sparse if it can be represented as $x = D\xi$, where $D \in \mathbb{R}^{n\times n}$ is a sparsity basis (e.g., a wavelet or Fourier basis) and $\xi \in \mathbb{R}^n$ is the sparse representation, with at most $\kappa \ll n$ non-zero entries.

To compress the signal, CS uses a random sensing matrix $A \in \mathbb{R}^{m\times n}$ (with $m < n$) that projects the signal onto a lower-dimensional space, resulting in a measurement vector

$y \in \mathbb{R}^m$ as follows:

$$y = A(x + \nu) = AD\xi + A\nu \tag{1.1}$$

Here, ν accounts for acquisition noise and non-idealities. The goal is to recover x (or equivalently, its sparse representation ξ) from the compressed measurements y, even though the number of measurements m is much smaller than the original signal length n.

Since the system is underdetermined, meaning there are infinitely many possible vectors ξ that could produce the same y, CS exploits the sparsity of the signal to guide the reconstruction. A well-established approach for reconstruction is the *Basis Pursuit with Denoising (BPDN)* method, witch solves the following optimization problem:

$$\hat{\xi} = \arg\min_{\xi \in \mathbb{R}^n} \|\xi\|_1 \quad \text{s.t.} \quad \|y - AD\xi\|_2 \leq \tau \tag{1.2}$$

where $\|\cdot\|_1$ is the ℓ_1-norm promoting sparsity, and τ is a threshold accounting for noise ν. The reconstructed signal is then obtained as $\hat{x} = D\hat{\xi}$.

The effectiveness of CS depends heavily on the design of encoder and decoder. The sensing matrix A is a crucial choice in the encoder design. To ensure accurate recovery of the sparse signal, the matrix must satisfy certain properties, such as incoherence with the sparsity basis D. Random Gaussian matrices, whose entries are drawn from a normal distribution, are often used to satisfy this requirement [15, 13], as they provide good incoherence guarantees. Alternatively, matrices with values drawn from a discrete set can be used. Notoriously, matrices with antipodal values (± 1) or binary values (0 and 1) can be employed, providing a balance between randomness and computational simplicity, as they reduce the complexity of matrix multiplications during encoding [20].

1.3 CS for ECG

Standard CS techniques and their extensions can be applied directly to ECG signals. In the standard CS framework, the high-dimensional ECG signal is projected onto a lower-dimensional space using a random sensing matrix and reconstructed usings standard algorithms.

The application of CS to ECG signals involves two main stages witch take place in different places: acquisition happens at sensor side and reconstruction at server side. A low-power sensor node uses a sensing matrix to project the high-dimensional ECG signal onto a lower-dimensional measurement vector. At the receiver (or in the cloud), an optimization algorithm recovers the sparse coefficient vector from the received compressed signal and reconstructs the signal.

It is possible to specialize CS to the statistics of the ECG signal, by carefully designed sensing matrix and by exploiting the inherent sparsity of the ECG in a proper transform domain. Many examples in the literature explore how CS can be extended by incorporating additional prior knowledge about the structure of ECG, e.g., by exploring the wavelet transforms.

One approach is to incorporate structured sparsity into the recovery process. For instance, once could enforce block-sparsity in the sparse representation vector to obtain improved reconstructions. For example, block sparse Bayesian learning methods have been successfully applied to ECG signals to exploit inter-beat correlations [32, 73].

Another direction is to combine model-based CS with adaptive dictionary learning. Since the ECG morphology may vary from patient to patient, adaptive dictionaries can capture personalized features more effectively than fixed bases such as the Discrete Wavelet Transform (DWT). Recent studies have shown that tailoring the sparsifying basis to the ECG data can yield better reconstruction quality at lower compression ratios [14].

Another way to leverage the structure of ECG data is to utilize wavelet transforms, witch exploit the quasi-periodic nature of ECG. CS techniques combined with wavelet-based transforms can achieve higher compression ratios than standard CS [42].

In addition to these approaches, several adaptations have been proposed in the literature to further optimize the sensing matrix. For example, the *Rakeness* framework adapts the sensing matrix to the second-order statistics of the input signal, ensuring better reconstruction for specific types of signals [43, 45].

1.4 CS and DNN in ECG

By projecting the signal onto a lower-dimensional space via a sensing matrix, CS achieves compression directly during acquisition, reducing data storage and transmission needs. Although encoding is lightweight, CS decoding requires optimization-based reconstruction, witch can be computationally demanding and depends on the sensing matrix and signal sparsity properties.

DNNs, on the other hand, have become highly effective for tasks requiring adaptive, data-driven solutions. Unlike CS, witch relies on mathematical formulations to ensure reconstruction quality, DNNs learn patterns directly from data. This data-driven approach enables DNNs to approximate complex functions, making them particularly suitable for tasks that involve large datasets or high variability. In the context of signal processing, DNNs have been used for denoising, feature extraction, and compression, as they can learn representations that are optimized for the specific characteristics of the signal. In recent years, DNNs have also been explored for reconstructing compressed signals, effectively replacing traditional optimization-based methods in CS and achieving real-time performance improvements.

Combining CS with DNNs offers the potential to harness the efficiency of CS-based encoding with the adaptability of DNN-based decoding. This hybrid approach allows encoding to remain lightweight, while DNNs can provide fast and accurate signal reconstruction, particularly beneficial in applications like real-time ECG monitoring. This fusion leverages the rigor of CS in the acquisition phase and the adaptive power of DNNs in the decoding phase, making it suitable for low-power, real-time applications in fields such as wearable health monitoring. The integration of CS and DNNs represents an exciting area of research, aiming to optimize both encoding and decoding while meeting the energy and performance constraints of modern devices.

CS-Net [72] combines CS-based sparse encoding with a deep network for decoding, designed specifically for ECG data. The encoding stage uses a traditional CS sensing matrix to reduce the signal dimension, while the DNN-based decoder reconstructs the ECG signals efficiently. The model is trained end-to-end, allowing the decoder to adapt to the compressed features, improving reconstruction accuracy for sparse ECG signals.

ReconNet [56] leverages a convolutional neural network (CNN)-based decoder to reconstruct ECG signals from CS-encoded measurements. ReconNet adapts well to ECG data by learning its structure, enabling faster and more accurate signal recovery compared to traditional optimization-based methods. The use of CNNs allows ReconNet to exploit spatial correlations in ECG signals, yielding improved performance in real-time applications.

DeepCS [86] introduces a hybrid CS-DNN approach, where the encoder uses a light sensing matrix, and the decoder consists of a multi-layer CNN trained to reconstruct compressed ECG signals. This approach demonstrates high accuracy and low complexity, making it suitable for mobile ECG monitoring devices. The decoder's design specifically caters to non-linearities in ECG signals, enhancing reconstruction fidelity.

[84] employs a recurrent neural network (RNN) as the CS decoder for ECG signals, where temporal dependencies in the ECG data are explicitly modeled. The RNN-based decoder improves upon traditional CS by better capturing signal dynamics, essential for reconstructing the inherently sequential structure of ECG data. This approach achieves high compression ratios with robust reconstruction quality.

Most notably, *Trained CS with Support Oracle (TCSSO)* [44], witch has been extensively tested with encoding-decoding of ECG signals, introduces a two-stage decoding scheme. TCSSO simplifies the decoding process by breaking the sparse representation ξ into two components:

- the *support* $s = \text{supp}(\xi)$, a binary vector that indicates the positions of the non-zero elements in ξ, i.e., $s \in \{0,1\}^n$ with $s_j = 1$ if $\xi_j \neq 0$ and $s_j = 0$ otherwise;
- the *amplitudes* of the non-zero elements of ξ, denoted as $\xi_{|s}$, witch represent the actual values of the non-zero entries.

TCSSO leverages a *support oracle*, implemented as a Deep Neural Network (DNN), to return a support $\hat{s}$ witch is an estimate of the support s. Once the support, the sparse signal is reconstructed by inverting $y = AD_{|\hat{s}}\hat{\xi}_{|\hat{s}}$. Anytime s has $k < m$ ones, $(AD_{|s})^\dagger$ allows a unique solution. In this case, using a simple pseudoinverse solves Eq. (1.1) without ambiguities:

$$\hat{\xi}_{|\hat{s}} = \left(AD_{|\hat{s}}\right)^\dagger y \tag{1.3}$$

where $\cdot_{|\hat{s}}$ represents the submatrix (subvector) composed of the columns (elements) corresponding to $s_j = 1$ of the support $\hat{s}$, and $\cdot^\dagger$ indicates the Moore-Penrose pseudoinverse. The final reconstructed signal is then $\hat{x} = D\hat{\xi}$.

The support is predicted a neural network, called *oracle*, witch can be implemented as a fully-connected deep neural network, with an input layer of dimension m, three hidden layers of size $2n, 2n, n$, and a sigmoid-activated output layer of dimension n. The output of

the oracle is a vector $p \in [0,1]^n$, where each entry p_j represents the probability that the corresponding element ξ_j is non-zero. The estimated support $\hat{s}$ is obtained by thresholding the probability vector p at a value $\theta = 0.1$, turning the probabilities into binary values.

The loss function $\mathcal{L}$ used to train the oracle is the categorical cross-entropy loss:

$$\mathcal{L}(p,s) = -\sum_{j=0}^{n-1} \left[s_j \log\left(p_j\right) - (1-s_j) \log\left(1-p_j\right) \right] \tag{1.4}$$

Note that, the sensing matrix A in TCSSO can be learned during training, enabling joint optimization of both the sensing and decoding processes.

1.5 Limitations

While standard Compressed Sensing is reduces data acquisition and transmission costs for ECG signals, it comes with several challenges and limitations that affect its practical implementation.

While CS simplifies the encoding process (making it computationally lightweight for wearable devices), the reconstruction phase remains computationally stressful, as CS recovery methods often rely on iterative convex optimization techniques (e.g., Basis Pursuit), witch can be computationally expensive and slow. In the next chapters, mixed solutions based on Deep Learning and CS are proposed to replace traditional decoders with neural networks able to recover a signal in a single feedforward pass.

ECG signals are often contaminated by several noises from muscle activity, electrode displacement, etc. Standard CS assumes that measurement noise is Gaussian and additive, but real-world ECG noise is often more difficult to model. The introduction of DNN improves adaptability to real data and improves robustness.

Standard CS typically employs a fixed number of measurements across all ECG signals, regardless of their complexity. However, ECG signals vary significantly depending on the patient and her/his conditions. A fixed compression ratio can lead to inefficient sampling‘Ă‘Ťwasting resources on simple signals while providing insufficient data for complex ones. In the next chapters adaptive CS strategies are proposed. Novel solutions, able to dynamically adjust the number of measurements based on signal complexity will be discussed.

The sensing matrix in CS plays a crucial role in ensuring stable signal recovery. Traditional choices include Gaussian random matrices, but as it will be shown in the next chapters, more Hardware-friendly structured sensing matrices can be designed to improve efficiency while maintaining reconstruction performance.

Chapter 2

Binary Compressed Sensing with Neural Optimization

We introduce a novel approach that combines binary compressed sensing with deep neural network optimization. Traditional compressed sensing methods often struggle producing binary encoding matrices without degrading performances because of the lack of adaptation. On the other hand, data-driven models, such as DNN, clash with the binary nature of the sensing matrix. Gradients computed over discrete values hamper the backpropagation of the gradient. Our method addresses this limitation by designing trainable binary sensing matrices, significantly reducing computational complexity during signal acquisition without compromising the performance. Our implementation also allows a joint optimization of the encoding and decoding.

2.1 CS encoder with a Trainable Binary Sensing Matrix

The introduction of a trainable binary sensing matrix presents challenges for the typical gradient-based optimization process. Standard training methods rely on differentiable outputs, while binary matrices, with values of either 0 or 1, yield gradients that are either undefined or zero, thus hindering the backpropagation algorithm. To overcome this, we follow the methodology outlined in [2, 3], witch introduces a multi-stage process for producing a quasi-binary matrix. The matrix is turned binary by the conclusion of training by applying a threshold.

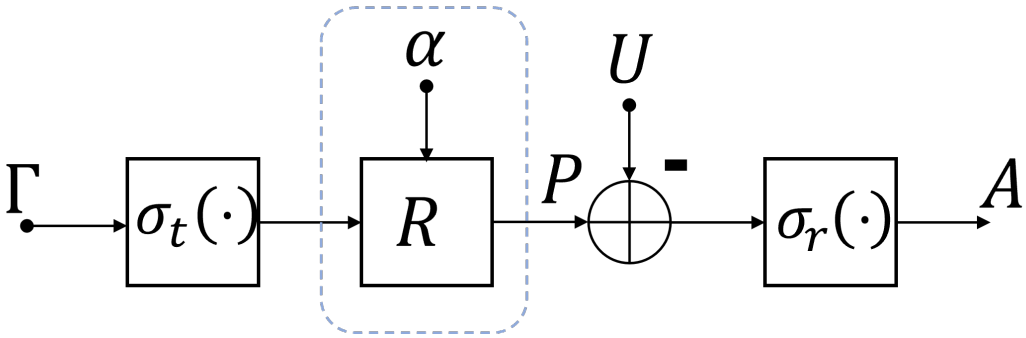


FIGURE 2.1: Block diagram of the sub-network generating a binary sensing matrix.

The key idea is to initialize the sensing matrix with random values within the range of $[0,1]$ and progressively guide the training process towards values close to either 0 or 1. This is achieved by generating the $m \times n$ quasi-binary matrix A using two auxiliary matrices: a *probability* matrix $P \in [0,1]^{m\times n}$, witch dictates the likelihood of each element in A being 1, and a random matrix $U \in [0,1]^{m\times n}$, whose elements are sampled from a uniform distribution. The quasi-binary matrix A is then computed as:

$$A = \sigma_r (P - U) \tag{2.1}$$

where $\sigma_r(\cdot)$ is the sigmoid function:

$$\sigma_r (\cdot) = \frac{1}{1 + \exp(-r\,\cdot)} \tag{2.2}$$

with r a so called *slope* scaling parameter. The role of the sigmoid function is to convert every element to a probability value. In Eq. (2.1), the parameter r controls the degree of resemblance between A and a binary matrix such that the higher the r, the closer to binary the A. To definitely turn A into a binary matrix, at the end of the training a threshold $\gamma = 0.5$ is applied to P.

The matrix P, witch governs the probabilities, is learned during the training phase. There are two potential configurations for this process. First, if the number of ones in the matrix is allowed to vary, P is computed from a trainable real matrix $\Gamma \in \mathbb{R}^{m\times n}$ by applying a sigmoid function to each element, i.e., $P = \sigma_t(\Gamma)$. Alternatively, if a fixed fraction α of ones is required, the probability matrix is rescaled accordingly:

$$P = \begin{cases} \frac{\alpha}{\langle P' \rangle} P' & \text{if } \langle P' \rangle \geq \alpha \\ J - \frac{1-\alpha}{1-\langle P' \rangle} (J - P') & \text{if } \langle P' \rangle < \alpha \end{cases} \tag{2.3}$$

where $P' = \sigma_t(\Gamma)$, $\langle P' \rangle$ is the mean value of the elements in P', and J is an $m \times n$ matrix of ones. The rescaling process ensures that the desired fraction α of ones in the binary matrix is achieved. The process is illustrated in Fig. 2.1, where the optional rescaling stage R is depicted.

Finally, it's worth highlighting that the only parameters learned during training are the entries of the matrix Γ, witch can be trained in conjunction with the support oracle to jointly optimize the sensing and decoding stages.

2.2 Numerical evidence

In this section, we evaluate the performance of our proposed binary encoding matrix trained along with the two-stage decoder. Synthetic ECG data is used to illustrate the effectiveness of our approach, and we compare it against both the original TCSSO (as in [44]) and traditional CS methods.

Specifically, for the standard CS comparison, we utilize BPDN as the decoder with two types of sensing matrices: *i*) a real-valued matrix (CS), where the elements are sampled from

a random normal distribution, and *ii*) a binary matrix, created by thresholding the real matrix to achieve a specified fraction of ones α. We evaluate cases with $\alpha = 0.1$ (CS-BM10) and $\alpha = 0.4$ (CS-BM40). Additionally, we consider a TCSSO variant generalized to real-valued encoding matrices (TCSSO) for a more comprehensive comparison with our proposed method, witch includes a fixed $\alpha = 0.1$ (TCSSO-BM10) and a case where α is automatically learned during training (TCSSO-BM).

For the numerical simulations and neural network training, we use a synthetic ECG dataset. This class of signals exhibits sparsity in the wavelet domain with respect to the set of orthonormal vectors from the Symmlet6 family. To generate the noiseless waveforms, we utilize MATLAB code[1], following the parameter settings outlined in [43]. Specifically, heart-rate values are uniformly sampled between 60 and 100 beat per minute. We introduce noise by adding Gaussian white noise with a power level set to achieve an intrinsic SNR (ISNR) of 60 dB. Using a sampling frequency of 256 sample per second, the signals are divided into 2 s chunks, witch are further split into windows of $n = 128$ samples (with $\kappa = 24$). From these chunks, we generate a total of 8×10^5 instances.

For training the neural networks, 80% of the dataset is randomly allocated for training, while the remaining 20% is used to evaluate the model's performance during testing.

The training of the models employs the ADAM optimization algorithm [29], with a batch size of 100 and an initial learning rate of 0.001, witch is reduced when the loss function ceases to decrease. Training is conducted for a maximum of 10,000 epochs, with early stopping applied if convergence is detected. The generation of the sensing matrix A uses slope parameters for the sigmoid functions set to $t = 5$ and $r = 200$.

We evaluate the model performance by measuring the signal quality in terms of Reconstruction Signal-to-Noise Ratio (RSNR):

$$\mathrm{RSNR} = \left(\frac{\|x\|_2}{\|x - \hat{x}\|_2} \right)_{\mathrm{dB}} \tag{2.4}$$

While the average RSNR offers an overall indication of the reconstruction quality, it does not provide insight into the variability of the performance across different instances. To address this, we also consider the Probability of Correct Reconstruction (PCR), witch is defined as:

$$\mathrm{PCR} = \Pr\{\mathrm{RSNR} \geq 55\,\mathrm{dB}\} \tag{2.5}$$

where 55 dB is the minimum RSNR value required to classify an instance as correctly reconstructed.

Fig. 2.2 and Fig. 2.3 illustrate the comparison between methods in terms of average RSNR and PCR, respectively, across various values of m. Reconstruction quality improves as m approaches n. Interestingly, the TCSSO-based methods outperform standard CS although using binary matrices.

For values of $m > 55$, the performance of both TCSSO-BM and TCSSO-BM10 is nearly identical, with both methods achieving an average RSNR of approximately 62 dB and a PCR

[1] Visit http://physionet.org/content/ecgsyn/

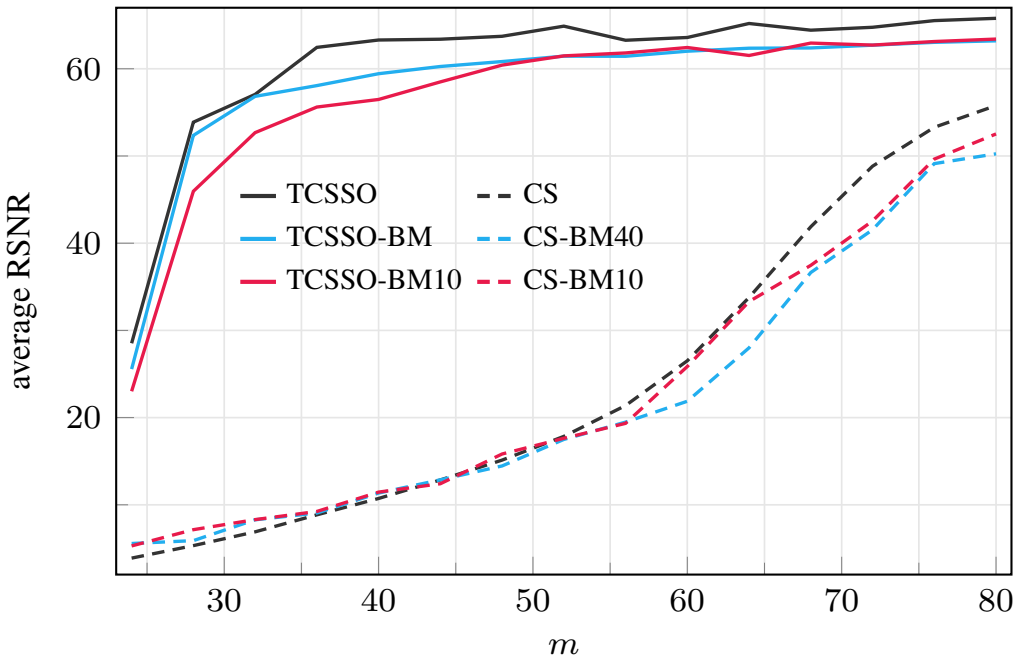


FIGURE 2.2: Performance in terms of RSNR as function of m.

around 0.95. However, when compared to the original TCSSO, these binary-based methods experience a reduction of about 3 dB in RSNR and a 3% decrease in PCR. As m decreases, the model with the learned fraction of ones ($\alpha = 0.47$) exhibits better performance, particularly in terms of PCR, where the differences become more pronounced.

In contrast, standard CS with a real-valued sensing matrix consistently delivers higher reconstruction quality compared to its binary matrix counterparts, demonstrating that real matrices are generally more effective in this scenario.

A notable distinction between TCSSO and standard CS with binary matrices is the sharp decline in performance observed in standard CS when the fraction of ones, α, exceeds 0.4. In contrast, TCSSO consistently achieves optimal performance with α around 0.47, regardless of the value of m. For the range of m considered, TCSSO-BM automatically determines α to be within the interval $[0.46, 0.48]$.

To illustrate the differences in the sensing matrices produced by each method, Fig. 2.4 shows examples of matrices for both standard CS and TCSSO with $m = 48$. The matrices on the left correspond to $\alpha = 0.47$, as learned by TCSSO-BM, while those on the right represent $\alpha = 0.1$. A clear distinction can be seen: standard CS matrices display no discernible structure, while TCSSO matrices exhibit strong positive correlations between the elements in each row. This suggests that TCSSO is learning and adapting the sensing matrix to capture statistical characteristics of the input signals. For instance, with ECG signals, witch are inherently low-pass, the rows of the TCSSO sensing matrix show a low-pass behavior, while still maintaining the necessary randomness for the CS framework.

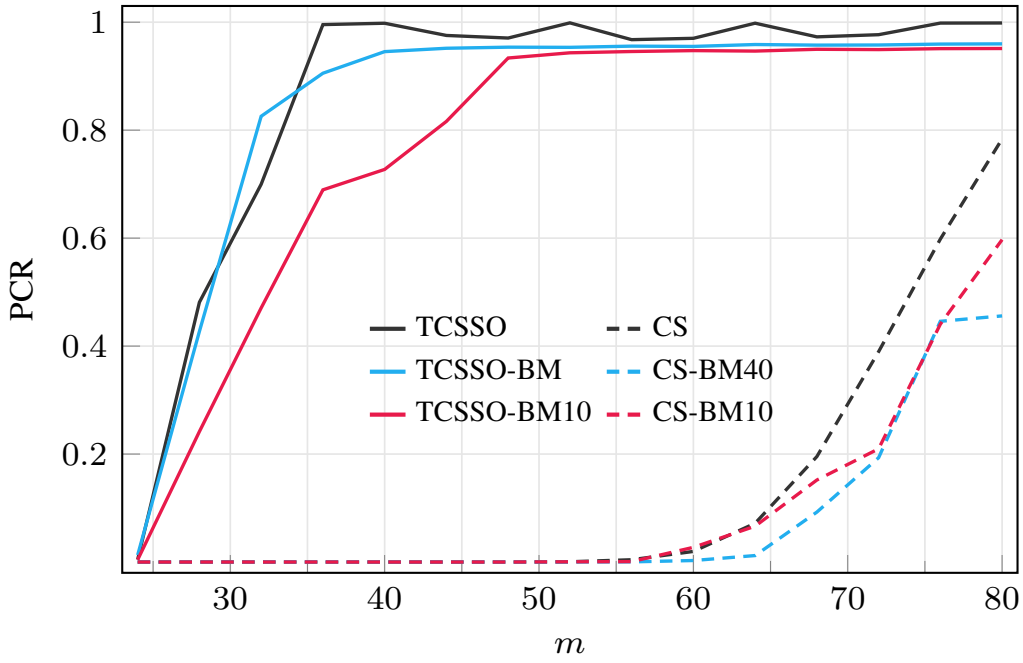


FIGURE 2.3: Performance in terms of PCR as function of m.

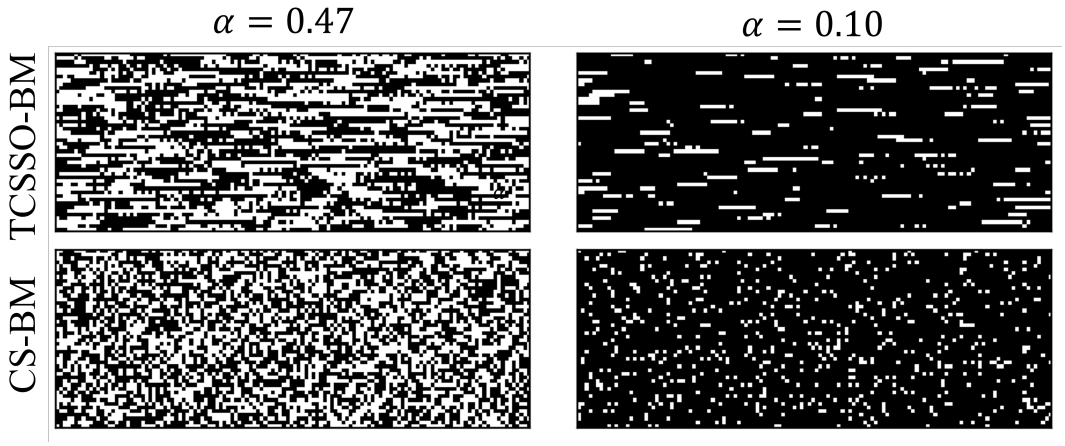


FIGURE 2.4: Examples of TCSSO (top) and standard CS (bottom) binary matrices with $m = 48$ and $\alpha = 0.47$ (left) and $\alpha = 0.1$ (right).

2.3 Trade-off Summary

The primary benefit of using a binary sensing matrix is the substantial reduction in computational complexity and power consumption during data acquisition. This is especially important for wearable devices where battery life and hardware simplicity are critical. However, this efficiency comes at the expense of a modest decrease in reconstruction fidelity. In some applications, a slight loss in signal quality might be acceptable if it means that the device can operate for longer periods or at lower cost.

Binarization simplifies the hardware implementation, but it introduces additional hyperparameters (such as the sigmoid slopes). The performance of the system thus depends on more factors.

In summary, the method strikes a balance between the need for low-power, computationally efficient encoding and the requirement for accurate signal recovery. The trade-offs involve accepting a potential decrease in reconstruction quality and increased sensitivity to hyperparameter choices.

Chapter 3

Incremental Adaptive Compressed Sensing with Performance Prediction

In this work, the compression process is not solely based on the statistical characteristics of the signal but is dynamically adjusted for each specific signal instance. Traditional ECG compression techniques often apply a fixed level of compression across all signal data, regardless of the underlying variability of each segment. However, real-world ECG signals are highly diverse, with some segments requiring more detailed representation than others to maintain critical diagnostic information. This calls for a more adaptive approach to balance efficient data compression with high signal fidelity.

The proposed method introduces an incremental, adaptive compression strategy that tailors the level of compression to the unique characteristics of each individual ECG segment in real time. By progressively adjusting the compression as more information about the signal is acquired, this approach ensures that segments containing complex cardiac features receive the necessary resolution while simpler, less significant portions are compressed more aggressively. This incremental strategy dynamically modulates the number of measurements based on real-time quality assessments, optimizing the trade-off between data reduction and reconstruction quality.

3.1 ECG Incremental Acquisition

We introduce a method where the encoder adapts the number of measurements m to each input x, ensuring that only the required measurements to achieve the desired reconstruction quality are computed. This adaptive strategy reduces the computational effort needed to generate the measurement vector y, as well as the bandwidth required for transmission.

More specifically, the appropriate value of m for each signal x is determined through an iterative process. Initially, the encoder compresses x using the sensing matrix $A^{(0)}$ with m_0 rows, resulting in the measurement vector $y^{(0)}$, witch is sent to the decoder. The decoder then performs an initial reconstruction, producing the signal estimate $\hat{x}^{(0)}$. If the estimated $\mathrm{RSNR}(x, \hat{x}^{(0)})$ exceeds the predefined quality threshold $\mathrm{RSNR}_{\mathrm{min}}$, the decoder signals the encoder to terminate the acquisition for the current instance. Otherwise, additional measurements are requested.

In cases where further measurements are needed, the encoder computes an additional set of m_Δ measurements, $y^{(1)} = A^{(1)}x$, using a new set of sensing sequences in the matrix $A^{(1)}$. Upon receiving $y^{(1)}$, the decoder performs a new reconstruction by concatenating the previous measurements $y^{(0)}$ and $y^{(1)}$, along with the matrices $A^{(0)}$ and $A^{(1)}$. This increases the total number of measurements to $m_0 + m_\Delta$, thereby improving reconstruction quality. The process repeats, adding m_Δ measurements in each iteration until the required RSNRis achieved.

Assuming the system must guarantee $\text{RSNR}(x, \hat{x}) > \text{RSNR}_{\text{min}}$ with a probability of over 99% (referred to as CS_{ref}), the traditional CS approach fixes a number of measurements m_{ref} that is transmitted at once. Conversely, the incremental method (CS_{inc}) begins by transmitting m_0 measurements and, only when necessary, sends additional measurements in batches of size m_Δ until the quality requirement is met.

To compare the energy consumption of these two methods, we consider both computation and transmission costs. The computational cost, measured in terms of Multiply-Accumulate (MAC) operations, is directly proportional to the number of measurements. Specifically, the number of MAC operations is given by:

$$\#\text{MAC} = n \cdot m \tag{3.1}$$

where m is either m_{ref} for the reference solution or $m_0 + n_{\text{TX}} m_\Delta$ for the incremental approach, with n_{TX} representing the number of additional transmissions. The transmission cost, on the other hand, depends on the number of bytes transmitted per measurement and the overhead for each transmission. If B_y is the number of bytes required to represent each measurement and B_H represents the overhead bytes for each transmission, the total number of bytes transmitted for each input x is:

$$B = B_H + m \cdot B_y \tag{3.2}$$

3.2 Performance Predictor

The effectiveness of the incremental approach (CS_{inc}) relies on accurately predicting the reconstruction quality on the decoder side. For this, we employ a performance predictor designed to estimate whether the reconstructed signal $\hat{x}$ satisfies the condition $\text{RSNR}(x, \hat{x}) > \text{RSNR}_{\text{min}}$ without knowing the original signal x.

This predictor is implemented as a DNN, witch acts as a binary classifier that outputs a positive result when the RSNRexceeds the target RSNR_{min}, and negative otherwise. The inputs to the DNN include the reconstructed signal $\hat{x}$, the received measurements y, and the sensing matrix A used for compression. Since both y and A vary in size, they are zero-padded into fixed dimensions $\tilde{y}$ and $\tilde{A}$, with sizes n and $n \times n$, respectively.

The architecture of the DNN, as illustrated in Fig. 3.1, is composed of three main components. The first component processes $\hat{x}$ and $\tilde{y}$ through five 1D-convolutional layers[1]. The

[1]The kernel size of the 1D-convolutional layers is 3, with filter sizes of 42, 51, 64, 85, and 128.

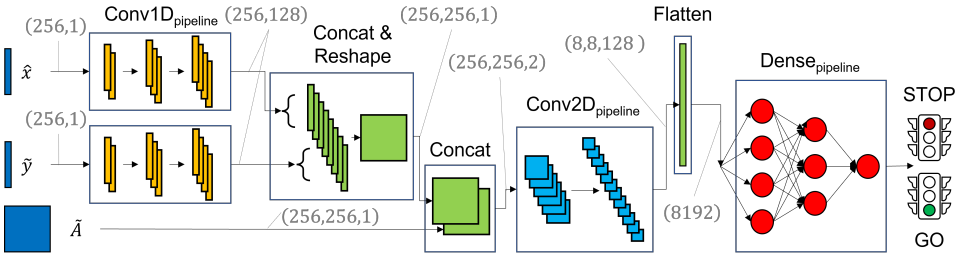


FIGURE 3.1: Architecture of the DNN used for performance prediction. The total number of parameters is 625,619 for $n = 256$.

output of these layers, with dimensions $n \times 128$, is concatenated and reshaped to match the dimensions of $\tilde{A}$. The second component processes the concatenated signal using five 2D-convolutional layers[2], followed by batch normalization and max-pooling[3] layers. Finally, the third component consists of four fully connected Dense layers[4], with batch normalization applied to all but the last layer. The Rectified Linear Unit (ReLU) activation function is used for all layers except the final one, witch uses a sigmoid function to produce the output probability. A threshold of 0.5 is applied to classify the result as positive or negative.

The DNN is trained using binary cross-entropy as the loss function and optimized with the Adam algorithm, starting with a learning rate of 0.001. Early stopping and learning rate reduction on plateau are applied to ensure convergence. To prevent overfitting, a dropout rate of 50% is applied to the Dense layers.

3.3 Numerical Evaluation

The Electrocardiogram (ECG) signals used for evaluation are synthetically generated using a realistic model [52], following the same configuration as in [43], with heart rates uniformly distributed between 60 and 100 beats per minute at a sampling rate of 256 sps. White noise is added to achieve an average signal-to-noise ratio of 40 dB. The Symmlet-6 wavelet basis [45] is used as the sparsity basis D. Signals are compressed using antipodal sensing sequences generated according to the rakeness framework, and reconstruction is performed by solving the Basis Pursuit Denoising (BPDN) problem[5] with the Spectral Projected Gradient for L1 minimization (SPGL1) toolbox [5].

The compression performance, measured in terms of RSNR as a function of the measurement vector dimension m, is shown in Fig. 3.2.

As expected, the average RSNR increases with the number of measurements m. However, the minimum value of m required to achieve a certain RSNR target varies widely. Fig. 3.3 illustrates this variability with histograms showing the distribution of m for two different RSNR_{min} targets.

[2]The 2D-convolutional layers have kernel sizes of 3×3, with filter sizes of 42, 51, 64, 85, and 128.

[3]The first three max-pooling layers have pool sizes of 3×3, while the last two have pool sizes of 5×5, all with a stride of 2.

[4]The Dense layers contain 50, 50, 25, and 1 neurons, respectively.

[5]The parameter τ, accounting for noise, is set to $\tau = 10^{-4}$.

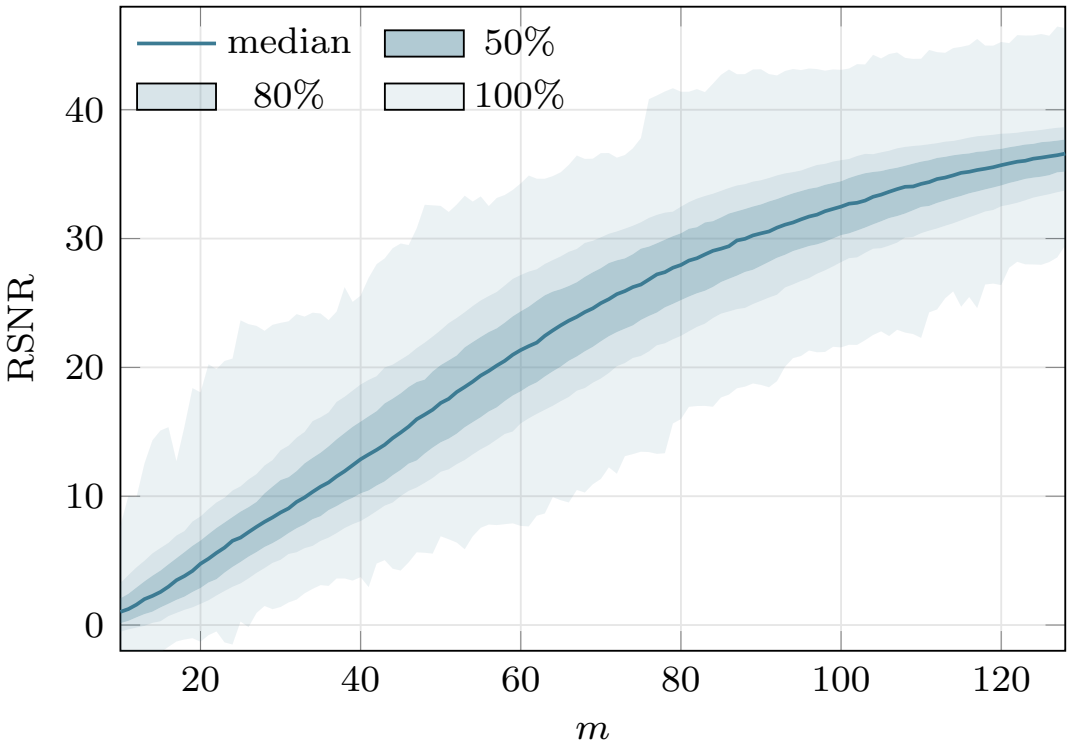


FIGURE 3.2: Reconstruction quality (RSNR) as a function of the number of measurements m. The solid line represents the median RSNR for different inputs x, while the shaded regions indicate the 50%, 80%, and 100% of the RSNR distribution.

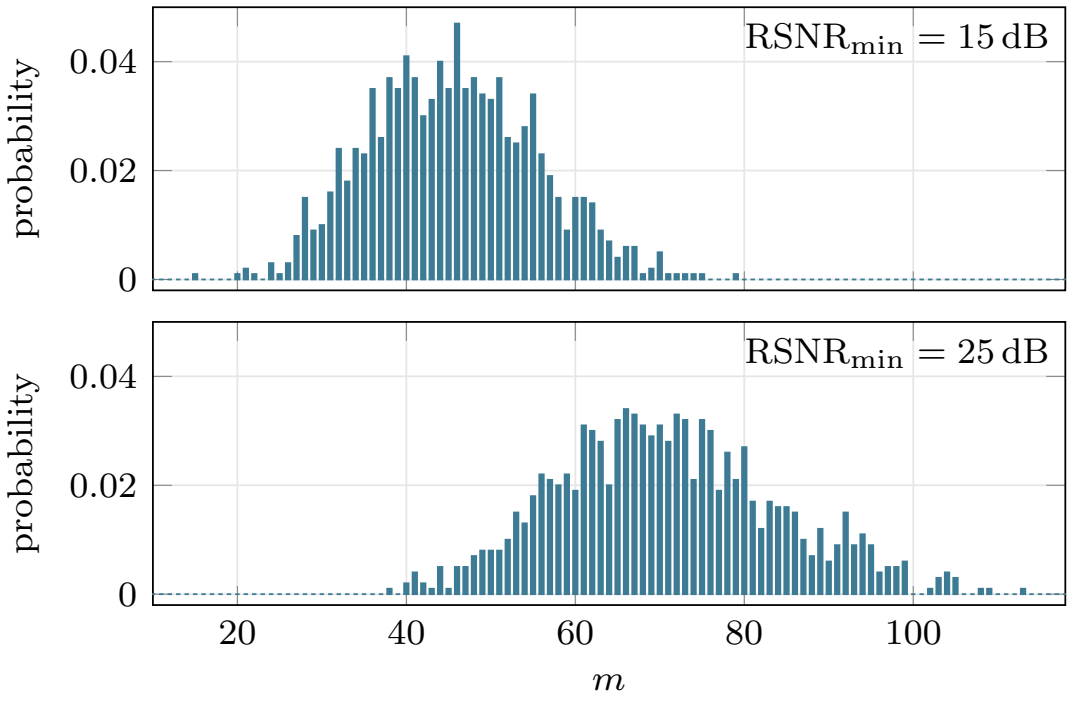


FIGURE 3.3: Histograms showing the minimum number of measurements m required to achieve an RSNRof 15 dB (top) and 25 dB (bottom).

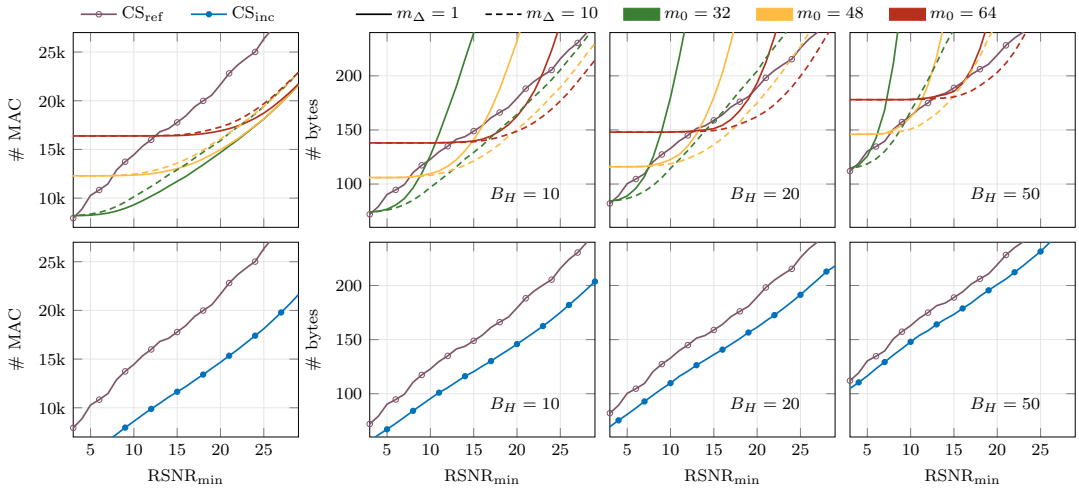


FIGURE 3.4: #MAC and #bytes vs RSNRtarget for different combinations of m_0 and m_Δ (top) and the optimal values of m_0 and m_Δ that minimize #bytes (bottom).

To satisfy the requirement for CS_{ref}, the value of m must ensure that 99% of instances meet or exceed the RSNR_{min}. In contrast, CS_{inc} determines the minimum m needed for each input.

The impact of the parameters m_0 and m_Δ on computation and transmission costs (as described in Eq. (3.1) and (3.2)) is depicted in the top plots of Fig. 3.4. The trends for both CS_{ref} and CS_{inc} are displayed for various combinations of m_0 and m_Δ, and for three values of the header size B_H.

The optimal values of m_0 and m_Δ, witch minimize transmission costs, are selected based on the size of the target RSNR_{min} and the corresponding values of B_H. The lower plots in Fig. 3.4 show that CS_{inc} offers considerable reductions in the number of #MAC and transmitted bytes compared to CS_{ref}, even when using non-optimal values of m_0 and m_Δ. These reductions are further highlighted in Fig. 3.5, where the savings range from 30% to 50% in terms of #MAC, and transmission savings depend significantly on the value of B_H.

Lastly, we evaluate the performance of the DNN predictor. The dataset for training the predictor is generated using 1,000 signal instances and the corresponding sensing matrices A. The measurements y are calculated for different values of m, and the reconstructions $\hat{x}$ are obtained. To balance the dataset, we select m values ranging from the minimum necessary to achieve at least 20% positive examples to the maximum ensuring all examples are positive. For target RSNRvalues of 15 dB and 25 dB, this corresponds to $m \in \{32, \ldots, 72\}$ and $m \in \{51, \ldots, 117\}$, respectively. The training set contains 32,800 examples, with 4,100 reserved for validation, and the test set includes 8,200 examples.

The predictor's performance is measured in terms of Accuracy, Precision, and Recall, as summarized in Table 3.1. For both RSNRtargets, the predictor achieves an accuracy of approximately 95%. The prediction errors are concentrated around RSNRvalues close to the target, as shown by the scatter plot in Fig. 3.6, where errors are marked in red.

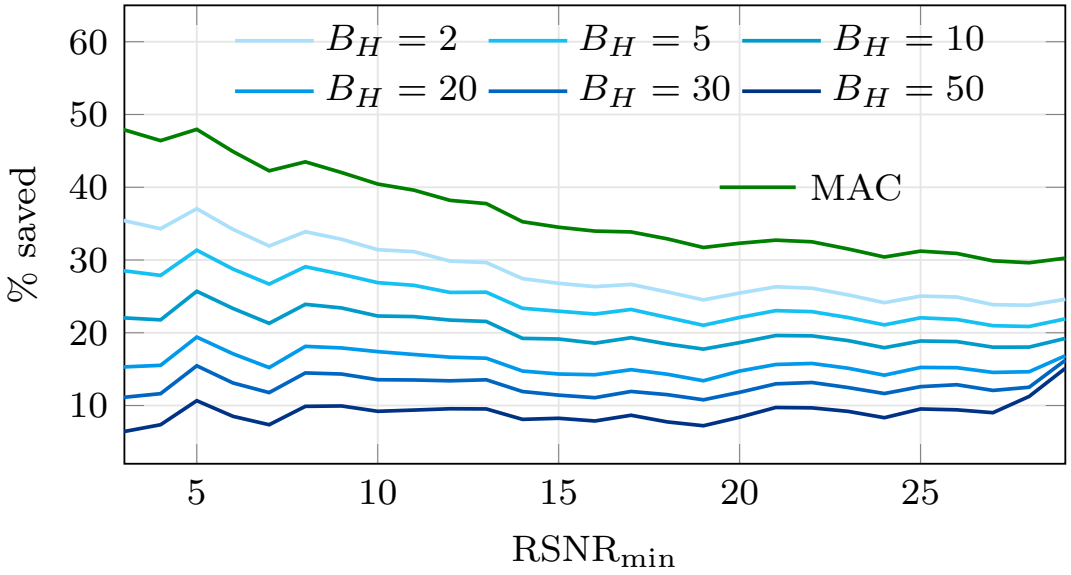


FIGURE 3.5: Percentage reduction in #MAC and #bytes as a function of RSNRtarget for various header sizes B_H.

TABLE 3.1: Performance of the predictor in terms of Accuracy, Precision, and Recall for target RSNRof 15 dB and 25 dB.

RSNR	Accuracy	Precision	Recall
15 dB	94.7%	95.9%	95.7%
25 dB	95.3%	96.7%	96.2%

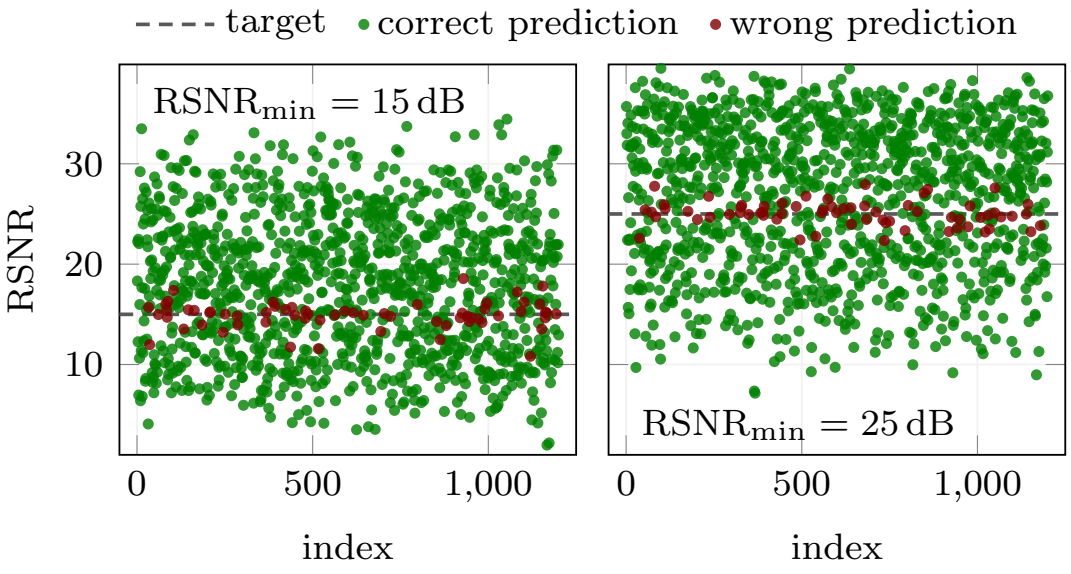


FIGURE 3.6: RSNRfor 1,200 examples, indicating correct and incorrect predictions.

TABLE 3.2: Optimal values of m_0 and m_Δ for RSNRtargets of 15 dB and 25 dB, with varying header sizes B_H.

RSNR	15 dB			25 dB		
B_H	10	20	50	10	20	50
m_0	46	49	55	69	73	81
m_Δ	9	9	12	9	12	14

3.3.1 Energy Consumption for Computation and Transmission

The system's energy consumption is mainly composed of three parts:

- the energy consumed by the AFE,
- the processing energy used by the MCU, and
- the transmission energy.

Since the AFE consumption is independent of the compression method, we focus our analysis on the latter two components.

The number of transmitted samples m_0 and m_Δ that better adapts to different B_H is summarized in Table 3.2.

TABLE 3.3: Average computation and transmission costs for different compression methods at specific RSNR_{min} targets.

RSNR_{min}	Method	Computation		Transmission	
		#MAC	Energy [μJ]	#bytes	Energy [μJ]
15 dB	CS_{ref}	17,780	19.90	150	13.82
	CS_{inc}	11,645	13.03	122	11.25
25 dB	CS_{ref}	26,342	29.48	217	19.98
	CS_{inc}	18,118	20.28	176	16.31
–	None	0	0	545	50.23

TABLE 3.4: Average total energy cost for different compression methods at specific RSNR_{min} targets.

RSNR_{min}	Method	Total Energy [μJ]
15 dB	CS_{ref}	33.72
	CS_{inc}	24.28
25 dB	CS_{ref}	49.46
	CS_{inc}	36.59
–	None	50.23

The computational cost for the encoder is largely determined by the matrix multiplication $y = Ax$, with an energy cost per MAC operation, E_{MAC}, as benchmarked on the nRF5340

microcontroller. By multiplying two $n \times n$ matrices, where $n = 100$, and using CMSIS-DSP with a DWT cycle counter, it was measured that 1,000,000 MAC operations require 6,141,510 clock cycles, translating to 6.14 cycles per MAC. Considering a clock speed of 64 M Hz (15.6 ns per cycle) and an average current consumption of 6.48 mA at 1.8 V, the energy consumption per MAC is:

$$\begin{aligned} E_{\text{MAC}} &= 6.48\,\text{mA} \cdot 1.8\,\text{V} \cdot 15.6\,\text{ns/cycle} \cdot 6.14\,\text{cycle/MAC} \\ &= 1.12\,\text{nJ/MAC} \end{aligned} \tag{3.3}$$

Table 3.3 summarizes the #MAC and corresponding energy consumption for both CS_{ref} and CS_{inc} at RSNRtargets of 15 dB and 25 dB. The table shows that CS_{inc} reduces computation energy costs by over 30% compared to CS_{ref}, while maintaining the same RSNRtarget.

For transmission energy, we focus on the energy required to send packets over BLE 5.3, assuming the most efficient settings for the 1M physical layer [74]. With a transmission rate of 1 µs/bit, each packet has an overhead of 11 B and a maximum payload of 250 B [8]. The nRF5340 draws 6.4 mA at 1.8 V, resulting in an energy cost per byte of:

$$\begin{aligned} E_{\text{B}} &= 6.4\,\text{mA} \cdot 1.8\,\text{V} \cdot 1\,\text{µs/bit} \cdot 8\,\text{bit/B} \\ &= 92.16\,\text{nJ/B} \end{aligned} \tag{3.4}$$

Using Eq. (3.2) and (3.4), along with the assumption that both the input samples x and the measurements y are represented with 16 bit, the energy required for transmission is calculated for CS_{ref}, CS_{inc}, and the no-compression case. Note that CS_{ref} only requires a single packet for transmission, while CS_{inc} may involve multiple packets, and the uncompressed case always requires three packets to transmit all $n = 256$ samples. Table 3.4 provides the average number of bytes transmitted and the corresponding energy consumption, with $B_H = 11$ and $B_y = 2$. The results show that CS_{inc} reduces transmission energy costs by 18% compared to CS_{ref}.

Overall, when considering both computation and transmission energy, CS_{inc} achieves energy savings of 28% and 26% for RSNRtargets of 15 dB and 25 dB, respectively. When compared to the uncompressed case, these savings increase to 52% and 27%, respectively.

3.4 Trade-off Summary

This novel method gives complex ECG segments more importance, and more measurements, while simpler segments are compressed more aggressively, thus optimizing both reconstruction fidelity and resource usage. It optimizes the transmission cost to the signal complexity and as a result, our comparative analysis shows that the incremental approach can reduce computation energy by over 30%.

This comes at the cost of increased complexity, as the incremental approach requires an iterative process in witch the decoder repeatedly assesses reconstruction quality and communicates with the encoder to request additional measurements. This feedback loop potentially

increases latency and requires more sophisticated synchronization between encoder and decoder.

Although incremental measurement can reduce the overall number of measurements, each additional batch of measurements incurs transmitting a header bytes. In scenarios where the signal is persistently complex (thus requiring many increments), the cumulative overhead might reduce the energy savings.

Moreover, the effectiveness of the approach hinges on the DNN predictor accurately estimating whether the reconstructed signal meets the RSNR threshold. In case where the DNN is trained on a dataset not matching the real acquired data, this affects the reliability of the method.

The incremental adaptive CS approach provides a balance between energy efficiency and reconstruction quality by dynamically adjusting the number of measurements based on the signal‘Ă‘Źs complexity. This results in significant savings in both computational and transmission energy, making it particularly attractive for wearable ECG applications where power and bandwidth are limited. However, these advantages come with the cost of increased system complexity, including the need for an iterative feedback loop between encoder and decoder and the integration of a fully data-driven performance predictor. The major trade-off remains balancing the benefits of adaptive, energy-efficient compression against increased system complexity and potential predictor inaccuracies.

3.5 Neural ECG Conclusion

The work presented in these two papers introduces significant advancements in the field of compressed sensing (CS) for biomedical applications, particularly for ECG signal compression and iterative adaptive signal acquisition. Both approaches showcase novel strategies to enhance the efficiency and flexibility of CS encoding and decoding while minimizing the computational and transmission costs, making them suitable for resource-constrained environments, such as wearable health-monitoring devices.

The first paper introduces a trainable binary sensing matrix optimized using deep neural networks (DNNs), witch dramatically reduces the computational complexity at the encoder stage. This binary matrix approach provides two primary advantages. First, it maintains the low power consumption characteristic of binary encoders while retaining a high level of reconstruction accuracy. Second, the flexibility of the learned binary matrix allows it to adapt to the statistical properties of the signal class, leading to performance improvements over standard random matrices, especially for sparse signals like ECGs. This methodology provides a balanced trade-off between computational efficiency and signal reconstruction quality, showcasing the potential for real-time, low-power ECG monitoring in wearable devices.

The second paper extends the compressed sensing framework by introducing an adaptive acquisition strategy that iteratively adjusts the number of measurements based on the signal complexity. This method optimizes energy consumption by dynamically tuning the number of transmitted measurements to achieve a target reconstruction quality. The primary novelty of this approach lies in its use of a deep learning-based performance predictor, witch allows

the system to stop acquiring additional measurements once the target quality is reached. This iterative process not only reduces the number of computations required at the encoder but also minimizes bandwidth usage, offering substantial energy savings in comparison to fixed-measurement CS methods. Furthermore, the incremental acquisition framework introduces the flexibility of dynamically adjusting to varying signal conditions, ensuring optimal performance across a wide range of ECG signals.

Both papers highlight the critical role of deep neural networks in improving the adaptability and performance of compressed sensing techniques, moving beyond static, model-driven approaches toward more data-driven, real-time solutions. By integrating these novel strategies, the works provide compelling solutions for the challenges of low-power, high-accuracy biomedical signal acquisition and processing. These innovations offer promising avenues for future research and practical applications in wearable devices and other resource-limited environments, where energy efficiency and real-time performance are paramount.

Part II

Neural Compressed Sensing for MRI Data

Chapter 4

MRI Physics and Acquisition

Magnetic Resonance Imaging is a non-invasive medical tool that leverages the magnetic properties of the particles composing the human body to find a mathematical representation of such body. It is probably the most interesting feature of MRI that such abstract description can be visually translated to obtain human-readable images.

An MRI machine can be schematized as the union of 4 fundamental components:

- Main Magnet: a massive solenoid coil maintained in superconductive state by a liquid helium cooling system, that generates a constant and homogeneous magnetic field. Because
- Radio Frequency emitter. Emits a homogeneous magnetic field with a frequency in the radio frequency (RF) range (around 100MHz) able to deposit energy in the nuclei.
- Radio Frequency receiver: a magnetic coil that collects the released energy of the excited atoms. It is built to be as sensible as possible to the emitted frequency, that is in the same range of the transmitted frequency.
- Magnetic field gradient coils. A collection of coils that linearly modify the main magnetic field along specified directions allowing enabling precise localization of the scanned anatomy.

Each of these components is utilized to stimulate, control or read a magnetic field.

4.1 MRI Physics

Every particle possess magnetic properties that define how such particle interacts with magnetic fields. By studying the interaction between magnetic fields and particles we can infer useful information about the particles themselves. It is not convenient in in this context to focus on the single particle; instead, it is more convenient to study the net magnetic field generated by or affecting such particles to infer information about a wider section.

A fundamental property of a particle is the spin. Spin is an intrinsic from of angular momentum. Accurate models to describe the possible spin interactions require relativistic quantum mechanics or quantum field theory. In this context, we limit our analysis to the magnetic field generated by the ensemble of the spin of the hydrogen nuclei in a specific location of

the body ($\overrightarrow{r} = [r_x, r_y, r_z]$), at time t. This is measured in terms of net magnetization $\overrightarrow{M}$:

$$\overrightarrow{M}(\overrightarrow{r}, t) = [M_x, M_y, M_z] \tag{4.1}$$

In general, the spins of elements that are not influenced by any external agent are uniformly randomly distributed along all directions, and the resulting net magnetization is null along all axis.

At time $t = 0$, the main magnet generates a constant magnetic field of magnitude B_0 in the z direction (the main magnetic field direction defines the z direction), such that a homogeneous magnetic field is created. We represent the generic magnetic field along the three axis as:

$$\overrightarrow{B}(\overrightarrow{r}, t) = [B_x, B_y, B_z] \tag{4.2}$$

where at $t = 0$ we have $B_x = 0, B_y = 0, B_z = B_0$. We neglect the magnetic field non-idealities that might exist along the z axis. Common values for B_0 are 1.5 T or 3 T.

When B_0 is applied, the spins averagely tend to be more aligned along the z axis, and the net magnetization reflects such change along the same direction:

$$M_z = M_0(\overrightarrow{r}) = \frac{N(\overrightarrow{r})\overline{\gamma}^2 h^2 I_Z(I_Z + 1)B_z}{3KT} \tag{4.3}$$

where $N(\overrightarrow{r})$ is the spin density, $\overline{\gamma}$ is the gyromagnetic ratio ($42.58e6$ Hz/T for hydrogen), h is the Planck's constant ($6.63e-34$vJ/Hz), I_z is the spin number ($1/2$ for hydrogen), K is Boltzmann's constant ($1.38e-23$ J/K) and T is the temperature (human body temperature is 310 K). The net magnetization is linearly dependent on the density of spins at the specific location, i.e., the concentration of hydrogen nuclei in the material under observation; and by the strength of the main magnetic field. M_x and M_y remain null.

While the main magnet applies an homogeneous magnetic field, the magnetic gradient coils apply a magnetic field $\overrightarrow{G} = [G_x, G_y, G_z]$ that varies linearly with $\overrightarrow{r}$. The resulting magnetic field:

$$B_x = r_x G_x(t) \tag{4.4}$$

$$B_y = r_y G_y(t) \tag{4.5}$$

$$B_z = B_0 + r_z G_z(t) \tag{4.6}$$

Based on the magnetic field strength the hydrogen nuclei resonate at different frequencies of the magnetic field, with the resonance frequency f_{res} defined as:

$$f_{\text{res}} = \overline{\gamma} \left\| \overrightarrow{B} \right\| \tag{4.7}$$

In the MRI context f_{res} is usually close to 100MHz.

The RF emitter coil produces a magnetic field with tunable frequency. For simplicity, let us assume the gradient coil on the xy axis are off; by turning on the RF emitter coil on the

xy axis with frequencies f_x and f_y respectively, $\overrightarrow{B}$ becomes:

$$B_x = B_1^+(\overrightarrow{r}, t)\cos(2\pi f_x(t)t) \tag{4.8}$$

$$B_y = B_1^+(\overrightarrow{r}, t)\sin(2\pi f_y(t)t) \tag{4.9}$$

$$B_z = B_0 + \overrightarrow{r}\,\overrightarrow{G}(t) \tag{4.10}$$

where B_1^+ is the intensity of the magnetic field applied by the transmitter gradient coils, witch amplitude is typically in the range of 10μT.

When the the emitted frequency matches the resonance frequency at a certain time $t = \tilde{t}$, i.e., $f_x(\tilde{t}) = f_{\text{res}}$ or $f_y(\tilde{t}) = f_{\text{res}}$, the nuclei experience resonance. In particular, the emitted RF signal perturbs the net magnetization that loses its equilibrium. The direct consequence of the misalignment on the nuclei is the net magnetization precession (measure as the "flip angle" θ), responsible for a time-varying magnetic field.

Once the transmitted RF signal is turned off ($B_1^+ = 0$), the nuclei lose the only energy source to precess and return to their relaxation state following the Block equation, witch can be studied separately for the transverse magnetization:

$$M_{xy}(\overrightarrow{r}, t) = M_{xy}(\overrightarrow{r}, 0)\exp^{-\frac{t}{T_2(\overrightarrow{r})}} \tag{4.11}$$

And for the longitudinal direction:

$$M_z(\overrightarrow{r}, t) = M_z(\overrightarrow{r}, 0)\exp^{-\frac{t}{T_1(\overrightarrow{r})}} + M_0(\overrightarrow{r}, t)(1 - \exp^{-\frac{t}{T_1(\overrightarrow{r})}}) \tag{4.12}$$

This phenomena, called relaxation, describes the magnetization decays to their equilibrium state $\overrightarrow{M}(\overrightarrow{r}, 0)$, with an exponential time constant T_1 and T_2 for transverse and longitudinal directions respectively.

During relaxation the receiving RF coils are activated to read the magnetic field emitted by the relaxing nuclei. Such coils, typically in the range of 8 to 64 (also known as channels), are tailored to the anatomy of interest, e.g. knee coils, abdomen coils, head coils. Every coil has a sensitivity profile that characterizes the sensitivity to the emitted signals, typically influenced by the distance from the origin of the emitted centre. Every coil readout is combined, based on its sensitivity, to create a unique homogeneous readout.

Magnetic resonance experiments are described by a Pulse Sequence, witch is a timing diagram encoding how the different magnetic fields are manipulated. In particular, they describe when and how the RF signal, the transversal and longitudinal gradients are applied, and when the receiving coil is activated to read the emitted relaxation signal.

Fundamental parameters of a pulse sequence are:

- Flip Angle (θ): how much the transmitted RF magnetic signal, that resonate the nuclei, rotates the net magnetization.

- Echo Time (TE): the time between excitation and data acquisition, typically measured between the center of the RF excitation pulse and the data acquisition window.

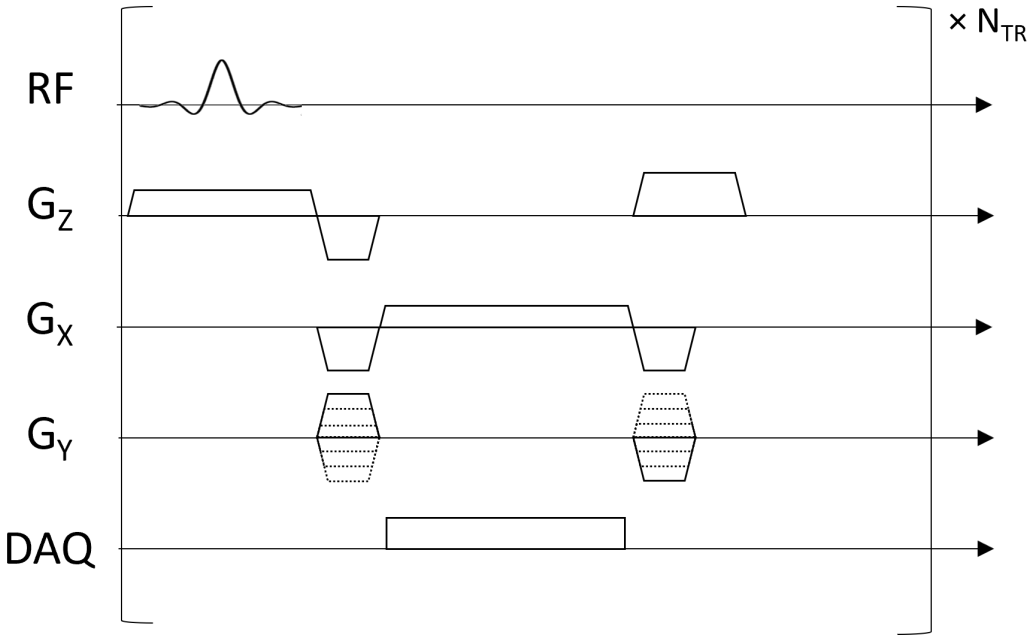


FIGURE 4.1: An example of a pulse sequence.

- Repetition Time (TR): time between repetitions of the main pulse sequence block
- Number of Repetitions (N_{TR}): how many times the pulse sequence block is repeated.
- Data Acquisition (DAQ): the interval of time when the scanner collects the magnetic resonance signals emitted by the relaxing nuclei.

An example of pulse sequence can be found in Fig. 4.1.

Because the MRI reads a signal over a portion of space, we define the "voxel" as the smallest section of the body that the MRI machine can investigate at the same moment (one sampled point refers to the properties of a voxel). The MRI punctual measurement is a sum of the transverse magnetization across the voxel:

$$s_{\text{voxel}} = \int_{\text{voxel}} M_{xy}\, d\vec{r} \tag{4.13}$$

when, in general, the net magnetization along the xy directions is decaying during relaxation with a time constant T_2.

A simple pulse is the so-called "gradient pulse": a sequence sending RF signal to produce $\theta = 90$ and manipulating the gradients along all directions to create spatial encoding, as visible in Fig. 4.1.

A more specialized MRI model takes into account the non-homogeneity of the main magnetic filed B_0. This phenomena is called off-resonance, because local variations in the magnetic field directly affects the resonance of the nuclei, as in Eq. (4.7). Indirectly, the off-resonance variations affect the phase of the spins of the relaxing nuclei by de-phasing them. Finally, this variation affects the signal s_{voxel} that is measured by the receiving coils. Instead of experiencing homogeneous alignment, the decaying nuclei see their magnetization rotating each with a different phase on the xy axis. Intuitively, this phenomena attenuates

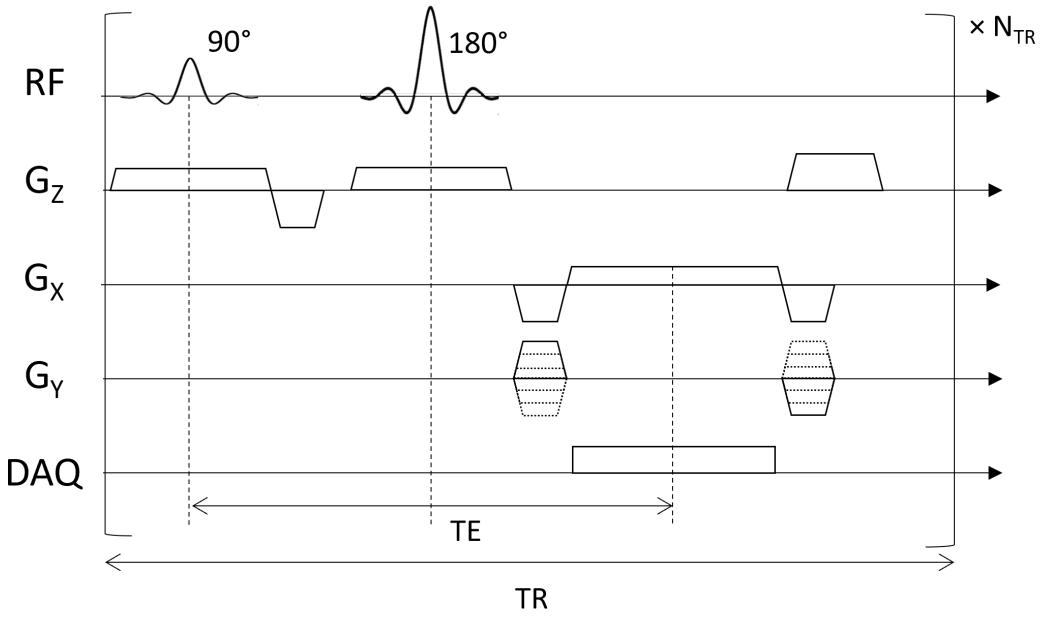


FIGURE 4.2: An example of a Gradient Echo Pulse Sequence

s_{voxel} because when two magnetization vectors on the xy axis are not aligned, it exists a component that negatively contributes to the sum, diminishing its final magnitude.

To compensate for such non-idealities, the "Gradient Echo Pulse Sequence" is a pulse sequence tailored to reduce the magnitude loss of the sum by re-aligning the spins. An example of Echo Pulse sequence can be observed in Fig. 4.2, where a second RF signal of $\theta = 180$ is injected to flip (and re-focus) the magnetization. This has the effect of inverting the phase accumulation of the off-resonance net magnetization, after witch the off-resonance phase begins to cancel out.

Based on the received signal s_{voxel}, it is possible to infer properties of the body tissues, that directly reflect in the final MRI image contrast. In particular the relaxation times T1 (spin-lattice relaxation time), T2 (spin-spin relaxation time), T2* (T2 also accounting for local magnetic inhomegeneities) in combination with the sequence parameters TE (echo time), TR (time of repetition) and θ (flip angle) are used to infer information about the tissue. For example, liquids and fluids have longer T1 and T2, while solids have intermediate T1, short T2 and fat has the shortest T1, intermediate T2. Typical values of T1 and T2 (ms) are known, given the magnitude of B_0.

In mathematical terms, the acquired signal $s_{\text{voxel}}^{\text{sequence}}$ is proportional to the desired time constant T_2 or T_2^* by selecting the sequence type:

- Gradient Echo (GE): $s_{\text{voxel}}^{\text{GE}} \propto M_0 \exp \frac{TE}{T_2^*}$
- Spin Echo (SE): $s_{\text{voxel}}^{\text{SE}} \propto M_0 \exp \frac{TE}{T_2}$

Differently, the T_1 constant is measured by repeating the same sequence with a given TR, such that there is incomplete recovery of the longitudinal magnetization and incomplete recovery appears in the transverse magnetization. Also the flip angle θ has a prominent role in reading T_1.

		TE	
		short	long
TR	short	T1-weighted	Not Used
	long	PD-weighted	T2 or T2* weighted

TABLE 4.1: Example of sequence tuning and relative dependence with tissue relaxation times.

To summarize, the parameters of the acquisition can be modified as in Tab. 4.1 to get the desired readout. Intuitively, by letting the system fully relax (long TR) and by reading after a long TE, we ensure that no T_1 affects the readout, thus we read T_2. Conversely, by using short TR and short TE we measure T_1.

Note that, pulse sequences are not limited to the ones shown, but there exist many more. For example the Inversion Recovery techniques use a 180-degree inversion pulse and are commonly used to null tissue types. For example, so-called short-time inversion recovery (STIR) is used to null fat signals, while fluid attenuated inversion recovery (FLAIR) is used to null fluids.

The gradient coils, by applying a difference in the gradient field affect the phase of the recessing nuclei such that:

$$M_{xy}(\overrightarrow{r},t) \approx M_{xy}(\overrightarrow{r},0)\exp\left(-i\overline{\gamma}\int_0^t \overrightarrow{G}(\tau)\cdot\overrightarrow{r}\,d\tau\right) \tag{4.14}$$

i.e., their effect is to create phase accumulation depending on the cumulative gradient. This allows to turn the gradients on and off to start and stop the phase accumulation, and also reverse the gradient polarity to undo any prior phase accumulation.

Here, we introduce the concept of k-space, witch is a simplified representation of the phase accumulation due to magnetic field gradients. It is defined as

$$\overrightarrow{k}(t) = -i\frac{\overline{\gamma}}{2\pi}\int_0^t \overrightarrow{G}(\tau)d\tau \tag{4.15}$$

hence, the net magnetization can be written as:

$$M_{xy}(\overrightarrow{r},t) = M_{xy}(\overrightarrow{r},0)\exp\left(-i2\pi\overrightarrow{k}(t)\overrightarrow{r}\right) \tag{4.16}$$

The MRI signal is the result of the sum of any precessing magnetization within the sensitive volume (vol) of the RF receive coils:

$$s_{\text{vol}}(t) = \int_{\text{vol}} M_{xy}(\overrightarrow{r},t)d\overrightarrow{r} \tag{4.17}$$

by integrating Eq. (4.16) and Eq. (4.17), this becomes:

$$s_{\text{vol}}(t) = \int_{\text{vol}} M_{xy}(\overrightarrow{r},0)\exp\left(-i2\pi\overrightarrow{k}(t)\overrightarrow{r}\right)d\overrightarrow{r} \tag{4.18}$$

where the integral, if expanded to the general range $[-\inf,+\inf]$, encloses the Fourier Transform $\mathcal{F}$ of the net magnetization, thus:

$$s(t) = \mathcal{F}\left[M_{xy}\right]\left(\overrightarrow{k}(t)\right) \tag{4.19}$$

where, at time t, $\overrightarrow{k}$ expresses the spatial frequency and $s(t)$ the magnitude of such frequencies in the volume of interest. Over time, $s(t)$ is stored in a data matrix (2D example) with known k-space locations, i.e., every $\overrightarrow{k}$ is associated to its magnitude $\mathcal{F}\left[M_{xy}\right]\left(\overrightarrow{k}\right)$ in a grid that we call $\boldsymbol{y}$. It is possible to retrieve the image in the spatial domain $\boldsymbol{x}$ (as opposed to the image in the spatial frequency domain $\boldsymbol{y}$) by simply applying the inverse Fourier Transform $\mathcal{F}^{-1}$:

$$\boldsymbol{x} = \mathcal{F}^{-1}(\boldsymbol{y}) \tag{4.20}$$

A commonly adopted spatial encoding is the so-called "Cartesian" encoding, where the k-space is sampled along a regularly spaced grid. The acquisition relies on two types of encoding:

- Frequency encoding: one dimension is encoded using a constant gradient during data acquisition. By convention, this is applied in the x-direction.
- Phase encoding: other dimensions are encoded using gradient applied after RF excitation and before data acquisition. This gradient is incremented to provide complete spatial encoding. To fill a line on the k-space grid this pulse is conventionally repeated with different amplitudes.

While frequency encoding is always applied along one dimension, phase encoding can be applied in one dimension for 2D imaging, and in two dimensions for 3D imaging.

We tune every component of $\overrightarrow{G}$ to obtain the desired $\overrightarrow{k}$. We call "k-space trajectory" the continuous pattern created by the gradients over time. Note that k-space trajectories always start at the center of k-space. The most common trajectories are Cartesian trajectories, in witch parallel lines of k-space are covered to sample a 2D (or 3D) grid. K-space trajectories with other patterns, such as radial lines, spirals, rastered lines (echo-planar trajectories), or blades are also possible.

Despite its numerous advantages, MRI has several complications and challenges that limit its widespread use:

- **Long Acquisition Times:** One of the primary challenges of MRI is the lengthy acquisition time, witch can range from several minutes to over an hour, depending on the scan type. This is a major limitation in clinical practice, as it can lead to patient discomfort, motion artifacts, and increased costs.
- **Noise and Patient Motion:** The MRI scanning process often generates considerable noise from the machine, witch can be unsettling for patients. The confined space inside the MRI tube, where patients are required to lie completely still, can intensify feelings of claustrophobia. As a result, some patients may choose to avoid undergoing the procedure altogether.

- **High Cost and Maintenance:** The complex structure of the MRI machine, particularly the superconducting magnet, requires constant cooling (often with liquid helium) and regular maintenance. The operational and maintenance costs make MRI one of the more expensive imaging modalities.

4.2 Undersampling in MRI

Magnetic Resonance Imaging (MRI) is widely known for its ability to produce detailed images of internal structures without ionizing radiation. However, MRI acquisition times are notoriously long, witch can be uncomfortable for patients and susceptible to motion artifacts. One of the most promising approaches to accelerate MRI acquisition is *undersampling*, where fewer data points are collected during acquisition compared to traditional, fully sampled k-space. In this section, we discuss why undersampling is feasible and advantageous, and we explore how it affects the physical aspects of MRI acquisition.

When undersampling is employed, fewer samples of k-space are collected, leading to gaps in the acquired data. These gaps introduce *aliasing artifacts* in the reconstructed images if not properly addressed. However, by utilizing advanced reconstruction techniques, it is possible to mitigate these artifacts and reconstruct images that are visually similar to those obtained from fully sampled data. The effect of undersampling on MRI physics can be understood as a trade-off: fewer measurements reduce acquisition time but make the reconstruction problem more challenging, requiring sophisticated algorithms to recover the missing information. Undersampling, by reducing scan time, enhances patient comfort and minimizes motion artifacts, resulting in higher-quality images [7].

From a physical perspective, the process of undersampling modifies the timing and gradients used during data acquisition. In traditional MRI, gradients are used to encode spatial information systematically over time, ensuring uniform coverage of k-space. In undersampled acquisitions, gradient trajectories may be altered to focus on acquiring the most important spatial frequencies first or to create non-Cartesian trajectories such as radial or spiral paths, witch inherently provide better coverage of the central, low-frequency regions of k-space [39, 7].

From a mathematical point of view, MRI machines acquire k-space samples, witch are points in the k-space: a spatial frequency domain represented as a Cartesian grid, denoted as $y \in \mathbb{C}^{d \times d}$. These $d \times d$ samples represent the spatial frequency coefficients of the image. The acquired data is then transformed into an image through the Inverse Fourier Transform, i.e., $x = F^{-1}(y)$. In this context, a k-space sample refers to an individual value of the k-space representing the frequency domain information. A standard full acquisition samples every elemnt of $\boldsymbol{y}$. In this context, undersampling means that only certain values are sampled. We model such operation with binary masks M, or undersampling patter, that specify witch k-space samples to acquire and witch to discard $M \in \{0,1\}^d \times d$.

To design the undersampling pattern and reconstruct the missing samples, Compressed Sensing (CS) has been introduced as a promising approach. CS enables the reconstruction of a signal from fewer measurements than those required by the classical Shannon-Nyquist

sampling theory, making it feasible to accelerate MRI while maintaining image quality [15, 37]. Notably, the adoption of CS for MRI has been approved by the FDA, paving the way for its widespread clinical application [18].

4.2.1 Compressed Sensing Methods in MRI

Compressed Sensing is based on the premise that many natural signals, such as medical images, are sparse in some transform domain. Mathematically, a generic input signal $a \in \mathbb{R}^d$ is assumed to be κ-sparse in an orthonormal basis $\mathbf{D} \in \mathbb{R}^{d\times d}$, meaning it can be expressed as $a = \mathbf{D}\xi$, where $\xi \in \mathbb{R}^d$ contains only κ non-zero entries, with $\kappa \ll d$. The signal acquisition process can be represented by a sensing matrix $\mathbf{A} \in \mathbb{R}^{m\times d}$, resulting in the acquisition vector $b = \mathbf{A}a = \mathbf{AD}\xi$ [40].

The reconstruction of ξ can be achieved by solving an optimization problem that enforces sparsity:

$$\xi^* = \arg\min_{\xi\in\mathbb{R}^n} \|\xi\|_1 \quad \text{s.t.} \quad \mathbf{AD}\xi = b, \tag{4.21}$$

where the ℓ_1-norm promotes sparsity in the solution. Regularization terms such as the total variation (TV) penalty [7] or wavelet-based sparsifying terms can be used to enhance reconstruction quality by leveraging domain-specific knowledge [37]. In MRI, the input signal is typically an image represented by $x \in \mathbb{R}^{d\times d}$, witch is acquired in the frequency domain using the Fourier transform $F(x)$. The total number of elements in the k-space is defined as $n = d^2$. The undersampling process limiting acquisition to $m < n$ samples can be described using a binary mask $\mathbf{M} \in \{0,1\}^{d\times d}$, with $\sum\mathbf{M} = m$, that selects witch frequency components to acquire:

$$\boldsymbol{y} = \mathbf{M} \circ \mathcal{F}(x), \tag{4.22}$$

where $\circ$ denotes the Hadamard (element-wise) product between matrices and Enc is intrinsically defined as the encoder and $\boldsymbol{y}$ as its output, i.e., the undersampled (encoded) MRI acquisition.

We define $r = \frac{m}{n}$ as the acquisition rate and $R = \frac{n}{m}$ as the undersampling ratio. The undersampled data $\boldsymbol{y}$ are then used by a decoder to reconstruct an image that closely approximates the fully sampled one [38, 27, 61]. The decoder commonly utilizes sparsifying transforms or deep neural networks (DNNs) to compensate for the undersampling artifacts.

It is important to notice that in this dissertation, when dealing with undersampling patterns, each k-space sample is assumed to be acquired independently of the others and in the same amount of time. This assumption simplifies the analysis by allowing us to isolate the contribution of each k-space sample to the image reconstruction. However, this is a significant idealization, as acquisition times can vary due to the chosen strategy and physical constraints.

Common and simple undersampling patterns are, for example:

- **Cartesian Sampling**[6, 61]: This method samples entire rows or columns in k-space, making it compatible with conventional MRI hardware. Cartesian trajectories are commonly used in clinical settings.

- **Radial Sampling** [34, 60]: This method samples k-space along radial lines from the center, providing good coverage of low-frequency components, witch are essential for capturing image contrast. Radial sampling is particularly robust to motion artifacts, making it suitable for dynamic imaging [7].
- **Spiral Sampling** [53]: Spiral sampling involves trajectories that cover k-space in a spiral manner, ensuring dense coverage of the center.
- **Unconstrained Patterns** [37]: Unconstrained undersampling of points in k-space is often used to demonstrate the potential of the models. These patterns provide a balance between incoherence and computational efficiency, but their implementation in practice can be challenging.

The design of an optimal undersampling pattern is a key challenge in CS-based MRI. In this dissertation, we limit our analysis to two types of undersampling patterns: unconstrained sampling, witch allows for flexible selection of k-space samples, and line-constrained sampling, where k-space is sampled along straight lines. The constraint in line-constrained sampling is an example of specific requirements imposed by the imaging system, whereas unconstrained sampling provides more freedom in selecting k-space samples.

4.2.2 Model-based MRI Undersampling and Reconstruction

To improve MRI acquisition efficiency, various undersampling and reconstruction methods have been developed, witch focus on exploiting the inherent properties of MRI signals. This section introduces model-based reconstruction techniques for MRI undersampling:

The concept of CS was first introduced by Donoho [15], providing the foundational theory for reconstructing signals from fewer measurements than dictated by the Nyquist-Shannon theorem. This idea was later extended to MRI by Lustig et al. [37], where the authors demonstrated that MRI images, witch are often sparse in transform domains like wavelets, can be reconstructed accurately using CS. This work laid the groundwork for CS-based MRI, significantly reducing the acquisition time while maintaining image quality.

Block et al. [7] explored the use of *Total Variation (TV)* constraints for MRI reconstruction from undersampled k-space data. Their work focused on iterative image reconstruction using a total variation constraint, witch effectively preserved edges and reduced aliasing artifacts, particularly in undersampled radial MRI.

Authors in [64] proposed a method based on *Dictionary Learning* for MRI reconstruction. Their approach trains a dictionary to represent image patches sparsely, witch allows for the efficient recovery of images from highly undersampled k-space data, thereby improving artifact suppression and preserving fine details.

Seeger et al. [70] introduced an approach for optimizing k-space trajectories for MRI using *Singular Value Decomposition (SVD)* and Bayesian experimental design. This method is particularly effective for dynamic MRI, such as cardiac imaging, where it is crucial to capture dynamic processes while maintaining high image quality.

Yang et al. [83] introduced an optimization method for MRI reconstruction using *Lp-norm minimization*. By employing ℓ_p-norm minimization (with $0 < p < 1$), they demonstrated improved recovery of sparse signals compared to the conventional ℓ_1-norm, allowing for better reconstruction quality under high undersampling ratios.

Bredies et al. [11] proposed the use of *Total Generalized Variation (TGV)* for MRI undersampling, witch extends the total variation approach by incorporating multiple levels of image derivatives. TGV is particularly effective in reducing staircasing artifacts, making it suitable for preserving fine details in undersampled MRI reconstructions.

4.2.3 Data-Driven MRI Undersampling and Reconstruction

Deep Learning (DL) has revolutionized the field of image processing and analysis, particularly in medical imaging. It leverages deep neural networks to automatically learn hierarchical feature representations from raw image data, allowing for high accuracy in various tasks such as image classification, segmentation, and object detection. Unlike traditional methods, witch often rely on handcrafted features and domain-specific expertise, DL can extract features automatically, significantly improving performance and efficiency. DL has shown significant potential for enhancing diagnostic accuracy and reducing workloads. Example of applications include segmentation of anatomical structures, detection of abnormalities, and prediction of disease outcomes based on medical images [67]. A fundamental contribution in the field of image processing is the use of Convolutional Layers. The breakthrough first hit the biomedical field in the shape of a fully convolutional image analyses [36] and gradually evolved. A milestone in this path is represented by the so-called U-Net, witch is particularly effective in segmenting medical images [66].

The U-Net utilizes an encoder-decoder structure with skip connections to preserve spatial details throughout the reconstruction. U-Net has been used extensively as a backbone for various MRI reconstruction models, such as in the fastMRI competition, witch aimed to develop and benchmark DL-based reconstruction methods that significantly speed up MRI scans while maintaining diagnostic quality [28].

Generative Adversarial Networks (GANs) have also been adapted to address MRI reconstruction problems, with notable success in improving visual fidelity. GAN-based models have been extended to include various loss functions to stabilize training and achieve better reconstruction quality. For example, Yang et al. [82] proposed a GAN-based method that uses a cyclic consistency loss to ensure consistency between the reconstructed and original images, improving artifact reduction.

Another important line of research is the incorporation of domain-specific adaptations to deep convolutional networks. In [68], Schlemper et al. introduced a deep cascade of convolutional neural networks for MRI reconstruction. Their model iteratively refines the reconstructed images by employing multiple stages of CNNs, achieving state-of-the-art results by effectively learning a mapping from undersampled to fully sampled data.

Additionally, models incorporating multi-domain and recurrent neural network components have demonstrated significant success. A notable example is the multi-domain approach

presented in [75], witch processes both the k-space and image domain in a unified framework, allowing the model to exploit redundancy across both domains for better reconstructions. This approach, adopted in the fastMRI challenge, demonstrates the ability of DL-based multi-domain networks to achieve high-quality image reconstruction.

The use of attention mechanisms in MRI reconstruction has also shown promise. Qin et al. [62] introduced a method that utilizes attention to focus on important spatial features, improving the quality of reconstructions. This approach helps prioritize diagnostically relevant regions during the reconstruction process, leading to improved artifact suppression and overall image quality.

While significant advancements have been made in deep learning-based reconstruction methods, there is also a growing body of work focused on optimizing the undersampling patterns used in MRI, witch directly impacts acquisition efficiency and image quality. This section highlights notable methods that introduce innovation specifically on the encoding side.

One noteworthy approach for optimizing the k-space undersampling pattern is the work by Knoll et al. [30], where the authors use the power spectrum of a set of example images as a reference to derive adapted random sampling patterns. This approach leads to better allocation of k-space samples and improved image quality compared to uniformly random undersampling patterns.

Another important contribution is from Seeger et al. [70], who proposed optimizing k-space trajectories through a Bayesian experimental design framework. This method formulates the problem as an optimization task, aiming to find the most informative k-space locations to acquire, thus maximizing the quality of image reconstruction for a given acquisition budget. Bayesian inference-based optimization has proven effective in reducing redundancy in k-space sampling.

Recently, several data-driven approaches have been proposed to jointly learn the undersampling pattern with the reconstruction network in an end-to-end manner. In particular, LOUPE (Learning-based Optimization of the Undersampling PattErn) introduced by Bahadir et al. [2, 3] utilizes a deep neural network to learn optimal sampling masks. This work demonstrated that undersampling masks could be trained alongside the reconstruction network to achieve better performance than predefined masks.

Another approach is the Reinforcement Learning-based framework by Zhang et al. [88], witch treats the selection of k-space samples as a decision-making process. The authors use reinforcement learning to optimize the acquisition pattern, achieving significant acceleration without compromising image quality. This adaptive selection process allows the model to explore different sampling configurations, thereby identifying an optimal pattern for different anatomy or imaging settings.

In addition, the method presented by Sriram et al. [75] integrates the learning of the sampling mask into the reconstruction process by treating the mask as a trainable parameter within the network. This approach allows for simultaneous optimization of the sampling pattern and the reconstruction process, leading to superior overall performance compared to separate optimization.

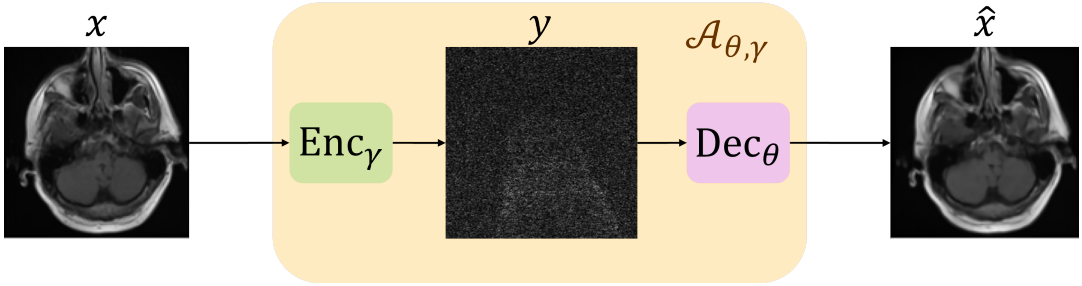


FIGURE 4.3: The encoder-decoder framework represented as a unified autoencoder structure $\mathcal{A}_{\gamma,\theta}(\cdot)$.

Another significant contribution is from Zhang et al. [87], who focused on radial undersampling, a pattern that is known for its robustness to motion artifacts. They proposed a learnable radial subsampling strategy that dynamically adjusts sampling density to focus on important regions of the k-space, thereby achieving high reconstruction quality with fewer measurements.

Following the success of LOUPE, Aggarwal et al. [1] proposed MoDL (Model-based Deep Learning), witch incorporates both model-based reconstruction and deep learning components to jointly optimize the undersampling pattern and reconstruction model. MoDL utilizes a deep network with multiple iterative updates to enhance the quality of MRI reconstruction, while simultaneously optimizing the sampling pattern to improve acquisition efficiency.

Another significant advancement was presented by Sun et al. [77], who introduced a method for the joint optimization of sampling patterns and deep priors using a model-based deep learning framework. Their approach integrates the learning of k-space trajectories as trainable parameters in the reconstruction network, enabling the simultaneous optimization of both aspects and resulting in an overall reduction of aliasing artifacts and improved image sharpness.

Additionally, Schlemper et al. [69] developed a method that employs an unrolled optimization strategy, where both the sampling mask and reconstruction network are iteratively refined in a cascaded manner. By updating the sampling pattern based on feedback from the reconstruction model, the approach adapts k-space acquisition strategies to best suit the anatomical structure being imaged, leading to significant improvements in reconstruction quality.

4.3 Learning-based Optimization of the Undersampling Pattern

LOUPE (Learning-based Optimization of the Undersampling Pattern) is designed as a deep learning framework for jointly optimizing the undersampling pattern and the reconstruction in MRI, integrating both an encoder for sampling and a decoder for reconstruction. The encoder and decoder stages are modeled as a generalized autoencoder structure, denoted as $\mathcal{A}_{\gamma,\theta}(\cdot) = \text{Dec}_\theta(\text{Enc}_\gamma(\cdot))$, effectively representing the entire process as a single deep neural network. Subscripts indicate parameter dependencies, meaning that any term with $\cdot_\theta$ or $\cdot_\gamma$ is dependent on the respective variables θ or γ.

The original input, a normalized image $\boldsymbol{x} \in [0,1]^{d\times d}$, undergoes encoding and decoding to produce the reconstructed image $\hat{\boldsymbol{x}} = \mathcal{A}_{\gamma,\theta}(\boldsymbol{x})$. The autoencoder is trained to approximate the identity function, ideally resulting in $\hat{\boldsymbol{x}} \approx \boldsymbol{x}$.

The **encoder** Enc generates an undersampling mask, denoted by $\boldsymbol{M}_\gamma \in [0,1]^{d\times d}$, witch controls the locations in k-space that are sampled:

$$\boldsymbol{y} = \text{Enc}(\boldsymbol{x}) = \boldsymbol{M}_\gamma \circ \mathcal{F}\boldsymbol{x}, \tag{4.23}$$

Such structure is also sketched in Fig. 4.3.

The generation of $\boldsymbol{M}_\gamma$ differs between the training and inference stages. During training, the mask is stochastic, while at inference it becomes deterministic.

At training-time, the mask is generated using a matrix $\boldsymbol{T}(\gamma) \in [0,1]^{d\times d}$, witch varies randomly during training but is fixed during inference. The matrix $\boldsymbol{T}(\gamma)$ is computed as follows:

- The matrix
$$\boldsymbol{S}(\gamma) = \sigma_t(\gamma) \in [0,1]^{d\times d} \tag{4.24}$$
is generated by applying the sigmoid function to each element of $\gamma \in \mathbb{R}^{d\times d}$. This operation maps the trainable parameters γ into the $[0,1]$ range.

- $\boldsymbol{T}(\gamma)$ is computed based on the average value of $\boldsymbol{S}(\gamma)$, $\langle \boldsymbol{S}(\gamma)\rangle = \frac{1}{N^2}\sum_{i,j} \boldsymbol{S}_{i,j}(\gamma)$:
$$\boldsymbol{T}(\gamma) = \begin{cases} \frac{r}{\langle S(\gamma)\rangle}\boldsymbol{S}(\gamma), & \text{if } \langle \boldsymbol{S}(\gamma)\rangle \geq r, \\ \boldsymbol{1} - \frac{1-r}{1-\langle S(\gamma)\rangle}(\boldsymbol{1} - \boldsymbol{S}(\gamma)), & \text{if } \langle \boldsymbol{S}(\gamma)\rangle < r, \end{cases} \tag{4.25}$$
where $\boldsymbol{1}$ is an $d \times d$ matrix of ones, ensuring that the expected value of $\boldsymbol{T}(\gamma)$ equals r.

- During training, a random matrix $\boldsymbol{U} \in [0,1]^{d\times d}$ is drawn with independent entries uniformly distributed $\boldsymbol{U} \sim \mathcal{U}(0,1)$, and the mask $\boldsymbol{M}_\gamma$ is generated as:
$$\boldsymbol{M}_\gamma = \sigma_s(\boldsymbol{T}(\gamma) - \boldsymbol{U}), \tag{4.26}$$
where $\sigma_s(\cdot)$ is another sigmoid function with slope s. The value of s determines the trade-off between smooth gradients for easier backpropagation and sharper, more binary-like masks.

The mask generation mechanism is implemented in four layers. The first layer computes $\boldsymbol{S}(\gamma) = \sigma_t(\gamma)$, the second layer calculates $\boldsymbol{T}(\gamma)$ using Eq. (4.25), and the last two layers generate $\boldsymbol{U}$ and apply Eq. (4.26).

Similarly, at inference-time, once training is complete, a deterministic binary mask is generated by thresholding $\boldsymbol{T}$:

$$(\boldsymbol{M}_\gamma)_{j,k} = \begin{cases} 0 & \text{if } \boldsymbol{T}_{j,k}(\gamma) < 0.5, \\ 1 & \text{if } \boldsymbol{T}_{j,k}(\gamma) \geq 0.5, \end{cases} \quad j,k = 0,\ldots,N-1. \tag{4.27}$$

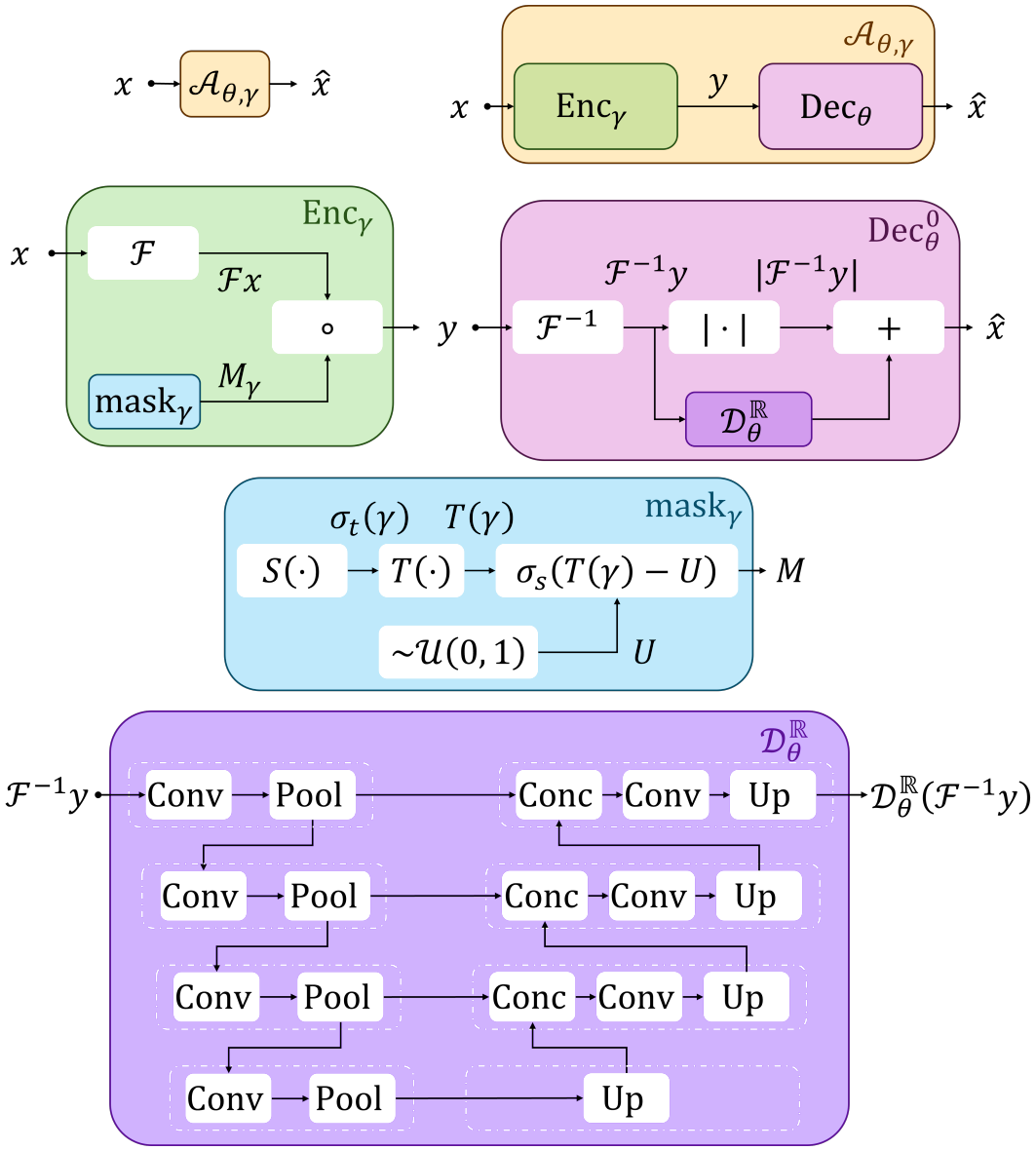


FIGURE 4.4: LOUPE schematic representation.

Fig. 4.4 illustrates the LOUPE's encoder-decoder flow during training.

The decoder Dec_θ^0 used in LOUPE is a slightly modified U-Net [66] designed to refine the initial reconstruction obtained from the undersampled data. The decoder operates in two stages. First, it reconstructs a baseline estimate $|\mathcal{F}^{-1}\boldsymbol{y}|$ from the encoded data $\boldsymbol{y}$, where $\mathcal{F}^{-1}\cdot$ represents the inverse Fourier transform, and the modulus is used to ensure real-valued outputs.

In the second stage, a sub-network $\mathcal{D}_\theta^{\mathbb{R}}$ applies pixel-wise corrections to the initial estimate:

$$\hat{\boldsymbol{x}} = \text{Dec}_\theta^0(\boldsymbol{y}) = |\mathcal{F}^{-1}\boldsymbol{y}| + \mathcal{D}_\theta^{\mathbb{R}}(\mathcal{F}^{-1}\boldsymbol{y}) \tag{4.28}$$

where $\boldsymbol{\theta}$ are trainable parameters.

The reconstruction error between the reconstructed image $\hat{\boldsymbol{x}}$ and the ground truth $\boldsymbol{x}$ can be implemented with a general ℓ_p norm difference:

$$\delta \boldsymbol{x}_l = \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_l \tag{4.29}$$

The parameters of the LOUPE architecture, including both $\boldsymbol{\gamma}$ for the encoder and $\boldsymbol{\theta}$ for the decoder, are trained using $l = 1$ (the ℓ_1 norm $\|\cdot\|_1$) to minimize Eq. (4.29), with the corresponding loss function defined as:

$$L^0(\boldsymbol{\gamma}, \boldsymbol{\theta}) = \mathbf{E}_{\boldsymbol{x}}[\delta \boldsymbol{x}_1] \tag{4.30}$$

where $\mathbf{E}_{\boldsymbol{x}}[\cdot]$ denotes the expectation over all possible images $\boldsymbol{x}$.

4.4 Limitations

One of the major drawbacks of conventional MRI is its long acquisition time, witch can lead to patient discomfort and motion-induced artifacts. To address these issues, undersampling is employed, wherein only a fraction of the full k-space data is acquired. This results in a significant reduction in scan time and mitigates motion artifacts, but it introduces challenges in the from of artifacts in the reconstructed images.

Undersampling drastically reduces scan time, however, the fewer samples acquired, the more challenging the reconstruction problem becomes. Advanced iterative algorithms and deep learning methods are required to recover high-quality images from incomplete data. As it will be shown in the next chapters, the trade-off between acceleration and reconstruction quality is fundamental and much of the effort of this thesis is dedicated to optimizing this balance.

Various undersampling strategies exist. Each has its own advantages and challenges. For instance, Cartesian sampling is hardware-friendly and common in clinical settings, while radial and spiral (up to unconstrained strategy) offer better coverage of important components but may require more complex gradient designs. Complex acquisitions can be designed to optimize acquisition, as it will be the case in the next chapters, but this comes at the cost of a more complex hardware implementation. In the next chapter it will be investigated the difference between unconstrained and Cartesian sampling strategies.

In MRI fast acquisition scenario, the need to simplify the model comes at the cost of losing precision. The typical assumption that each k-space sample is acquired in the same amount of time is a significant idealization. In practice, acquisition times may vary due to the specific sampling strategy and hardware limitations. While this assumption simplifies analysis, it may not fully capture the physical constraints and variability inherent in real-world MRI systems.

Chapter 5

Regularized LOUPE

Despite the notable success of existing deep learning methods for MRI reconstruction, they possess a fundamental limitation: they do not ensure that the original measured frequencies (i.e., the k-space measurements) are preserved throughout the reconstruction process. This shortcoming can lead to discrepancies between the actual acquired data and the reconstructed images, ultimately affecting the accuracy and reliability of the MRI scans.

To address this limitation, we introduce an approach that explicitly embeds the preservation of the measured frequencies as a constraint within the reconstruction pipeline. By ensuring that the encoder retains these measurements throughout the encoding and decoding processes, we can significantly enhance the quality of the reconstructed images. An additional advantage of our proposed method is the ability to incorporate a self-assessment mechanism directly into the MRI reconstruction process. By embedding a quality evaluation module that operates in parallel with the image reconstruction, our method can provide an estimate of the reconstruction quality in real time. This self-assessment capability enables users to determine whether the reconstructed image meets predefined quality standards, facilitating more informed decisions during the scanning procedure. Furthermore, this mechanism allows for the estimation of the reconstruction error at inference time, offering a valuable metric to guide further adjustments in the acquisition process. Similarly, next Sections further explore the idea of imposing the measurement constraint into standard data-driven MRI encoding-decoding scheme.

Although the current work is grounded in the LOUPE framework, it is important to note that the principles and techniques we introduce are not restricted to this specific architecture. The measurement-preserving mechanism, quality assessment capability, and overall framework are generally applicable to a broader range of MRI reconstruction models. This flexibility makes our contributions potentially useful for a wide array of deep learning-based reconstruction methods, providing a foundational approach to enhance reliability and performance across different architectures.

5.1 Measurement-Constrained Loss Regularization

As previously outlined, LOUPE shares similarities with Compressed Sensing (CS) in its approach to reconstructing the original instance from the measurement vector, leveraging the redundancy of information in the original domain. This implies that the sensing process

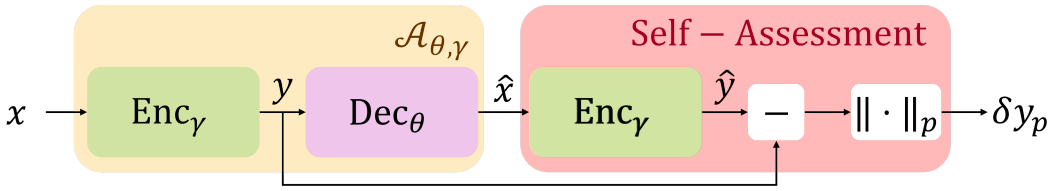


FIGURE 5.1: Schematic representation of $\delta \boldsymbol{y}_p$ implementation

retains sufficient information to enable effective reconstruction. However, unlike traditional CS, LOUPE does not require $\boldsymbol{y}$ to be obtained strictly through a linear projection of $\hat{\boldsymbol{x}}$.

Nevertheless, if reconstruction is successful, then $\hat{\boldsymbol{x}}$ should approximate $\boldsymbol{x}$ closely enough such that:

$$\delta \boldsymbol{y}_p = \|\boldsymbol{y} - \boldsymbol{M} \circ \mathcal{F}\hat{\boldsymbol{x}}\|_p \tag{5.1}$$

becomes negligible. Therefore, minimizing $\delta \boldsymbol{y}_p$ is desirable for improving reconstruction quality. This objective can be incorporated into network training by adding a regularization term to the loss function, resulting in an alternative formulation:

$$L'_p(\gamma, \boldsymbol{\theta}) = \mathbf{E}_x \left[\phi \delta \boldsymbol{x}_1 + (1-\phi)\delta \boldsymbol{y}_p\right] \tag{5.2}$$

where ϕ is a tunable weight that balances the contributions of each term. This approach allows us to approximate the constraint defined in Eq. (4.21). We refer to a model trained using L'_p as a regularized-LOUPE (r-LOUPE$_p$), with $p \in \{1, 2\}$.

5.2 Introducing Self-Assessment Capabilities

Ideally, for properly reconstructed signals, the residual error $\delta \boldsymbol{y}_p$ defined in Eq. (5.1) should approach zero. However, when the reconstruction fails, $\delta \boldsymbol{y}_p$ is expected to deviate significantly from zero, making its magnitude a useful indicator of decoder performance. Since $\delta \boldsymbol{y}_p$ can be computed directly from the available measurements $\boldsymbol{y}$, as sketched in Fig. 5.1, it can serve as a proxy for assessing the quality of the reconstruction.

This concept opens the door to a self-assessment mechanism integrated into the reconstruction process. After the image $\boldsymbol{x}$ is reconstructed, $\delta \boldsymbol{y}_p$ is calculated. The self-assessment stage can then either compare $\delta \boldsymbol{y}_p$ to a predefined threshold ϵ to signal a potential reconstruction issue if $\delta \boldsymbol{y}_p > \epsilon$, or use $\delta \boldsymbol{y}_p$ to estimate the reconstruction quality through a function $\delta \hat{\boldsymbol{x}}_l = \delta \hat{\boldsymbol{x}}_l(\delta \boldsymbol{y}_p)$.

For simplicity, we focus on an affine relationship between $\delta \boldsymbol{y}_p$ and the estimated output quality, expressed as:

$$\delta \hat{\boldsymbol{x}}_l(\delta \boldsymbol{y}_p) = c_0 \delta \boldsymbol{y}_p + c_1$$

where c_0 and c_1 are constants determined during the training process, witch help to fine-tune the estimation based on the behavior of $\delta \boldsymbol{y}_p$.

This approach provides a practical tool for users, offering additional insight into the accuracy of the reconstruction. It can enhance the interpretability of the system, making it possible to detect when an image may not be reliable due to poor reconstruction quality, without needing access to the original image $\boldsymbol{x}$.

5.3 Numerical Evidence

The numerical evidences presented in this section aim to validate the proposed methodologies through rigorous experimental evaluation.

5.3.1 Dataset

For training and testing our models, we used the publicly available "Brain MRI Segmentation" dataset[1], witch consists of brain MRI images.

This dataset includes 2D brain scans from 110 patients as part of The Cancer Genome Atlas (TCGA) lower-grade glioma collection[2]. The number of slices per patient varies between 20 and 88. To from the training set, we excluded some of the least informative images, resulting in a dataset comprising 2753 scans for training (80%) and 688 scans for testing (20%). All original images, witch have dimensions of 256×256, were resized to 128×128. Each scan was also re-scaled so that pixel values fall within the range [0,1].

5.3.2 Results

The Peak Signal-to-Noise Ratio (PSNR) is the primary metric used to assess the quality of reconstruction. It is defined as:

$$\text{PSNR} = 10\log_{10}\left(\frac{\max\{\boldsymbol{x}\}}{\delta\boldsymbol{x}_2}\right) = -10\log_{10}\left(\delta\boldsymbol{x}_2\right)$$

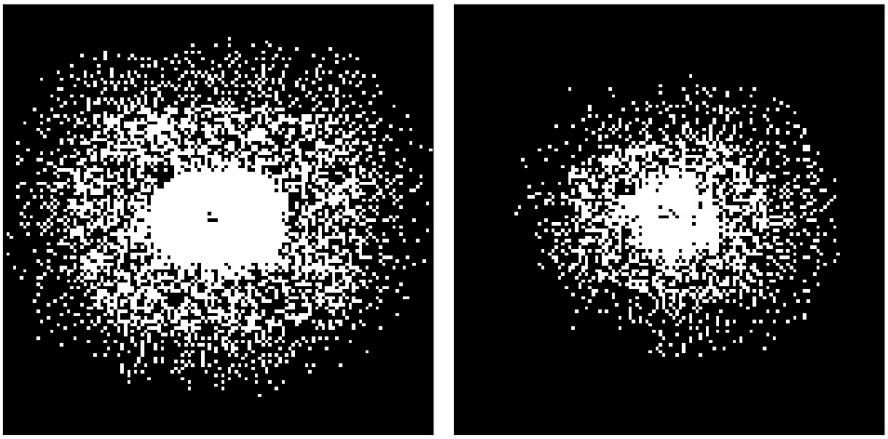
where $\max\{\boldsymbol{x}\} = 1$ in our case, and $\delta\boldsymbol{x}_2$ represents the ℓ_2-norm of the reconstruction error.

We evaluated our methods with acquisition ratios of $R \in \{4, 8\}$, corresponding to undersampling where 25% and 12.5% of the entries in $\boldsymbol{M}$ are ones. For the regularization weights that define r-LOUPE$_p$, we selected values such that $0.99 \leq \phi < 1$.

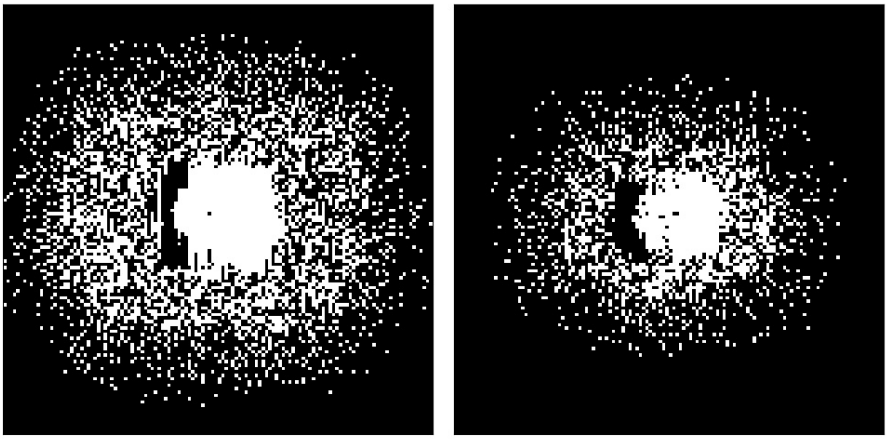
The undersampling masks $\boldsymbol{M}$ for the different methods were generated using the trained mask generator blocks as described in [3]. Specifically, each entry in the random instance of $\boldsymbol{M}$ is selected based on the probability defined by the corresponding entry in an $N \times N$ matrix that was trained during the model's learning process. In Fig. 5.2, we show the probability matrices for LOUPE, r-LOUPE$_1$, and r-LOUPE$_2$ for both $R = 4$ (top row) and $R = 8$ (bottom row).

Table 5.1 presents the average PSNR for all methods, along with the percentage of cases where r-LOUPE$_1$ or r-LOUPE$_2$ achieves a higher PSNR compared to LOUPE. Both r-LOUPE$_1$ and r-LOUPE$_2$ outperform LOUPE in at least 97.7% of the cases, in terms of average PSNR.

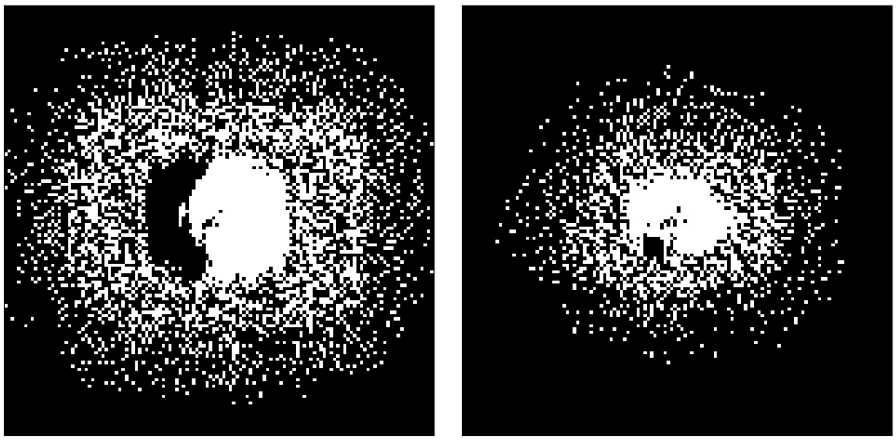
[1] https://www.kaggle.com/mateuszbuda/lgg-mri-segmentation
[2] https://www.cancer.gov/tcga



(A) LOUPE



(B) r-LOUPE$_1$



(C) r-LOUPE$_2$

FIGURE 5.2: Matrices representing the probability that each element of $\boldsymbol{M}$ equals one are depicted, with white denoting a probability of one and black indicating a probability of zero. The majority of values lie very close to these extreme probabilities. The first column plots correspond to $R = 4$, while the second column plots are for $R = 8$.

TABLE 5.1: Average PSNR and fraction of MRI images for witch r-LOUPE$_p$ achieves higher PSNR with respect to LOUPE

Model	Aver. PSNR [dB]		% of success	
	$R=4$	$R=8$	$R=4$	$R=8$
LOUPE	38.65	32.67	-	-
r-LOUPE$_1$	39.26	34.15	98.7	97.7
r-LOUPE$_2$	39.86	34.23	99.9	98.4

TABLE 5.2: AUC of the ROC for all considered models with $\delta \boldsymbol{y}_1$ or $\delta \boldsymbol{y}_2$ and with $R=4$ or $R=8$

Model	AUC using $\delta \boldsymbol{y}_1$		AUC using $\delta \boldsymbol{y}_2$	
	$R=4$	$R=8$	$R=4$	$R=8$
LOUPE	0.971	0.934	0.969	0.955
r-LOUPE$_1$	0.992	0.978	0.991	0.979
r-LOUPE$_2$	0.992	0.983	0.993	0.986

Fig. 5.3 depicts the probability density functions (PDF) for PSNR values, showing distributions for $R=4$ (Fig. 5.3a) and $R=8$ (Fig. 5.3b).

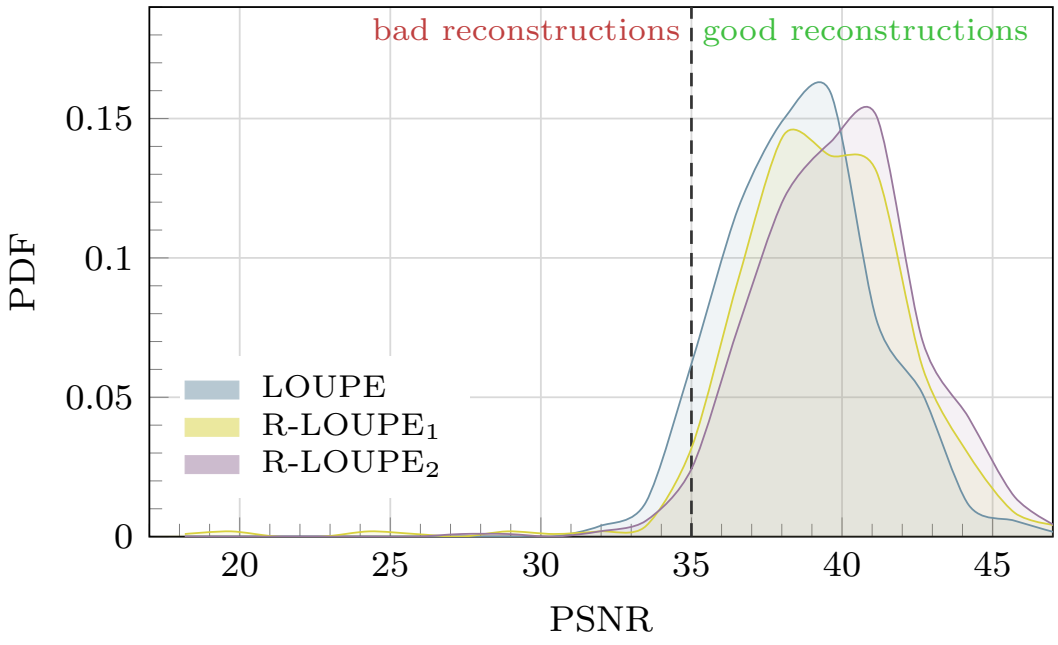
The dashed vertical line indicates a threshold PSNR, dividing the reconstructions into "bad" and "good". The density curves for the three methods are similar in general shape, but there are notable differences in their positions. LOUPE has a more significant portion of its distribution below the threshold compared to r-LOUPE$_1$ and r-LOUPE$_2$. This implies that LOUPE is more prone to producing lower-quality reconstructions, as a greater part of its distribution falls into the "bad reconstruction" region. r-LOUPE$_1$ and r-LOUPE$_2$, on the other hand, demonstrate improved performance.

In Fig. 5.4, we provide a visual comparison by showing an original MRI scan alongside its corresponding r-LOUPE$_2$ reconstructions for acquisition ratios $R=4$ and $R=8$.

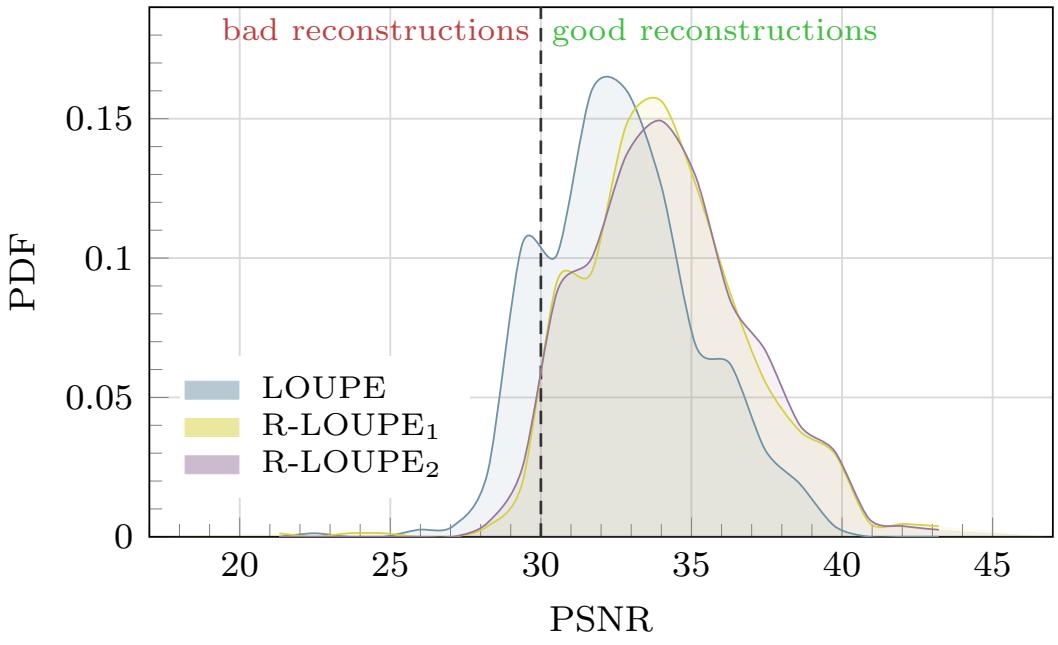
To evaluate the self-assessment capability of our method, we set PSNR thresholds to differentiate between high-quality and low-quality reconstructions. Specifically, for $R=4$ and $R=8$, reconstructions are classified as *good* if the PSNR exceeds 35dB and 30dB, respectively. The classifier attempts to distinguish between good and bad reconstructions by comparing $\delta \boldsymbol{y}_p$ with a threshold value ϵ. By adjusting ϵ, we can calculate the fraction of good reconstructions that are incorrectly classified as bad (False Positive Rate, FPR) and the fraction of bad reconstructions that are correctly identified as bad (True Positive Rate, TPR). Both FPR and TPR take values in the range $[0,1]$.

The overall performance of the classifier is evaluated using the Receiver Operating Characteristic (ROC) curve, witch plots TPR against FPR for various ϵ values. An ideal classifier would achieve FPR$=0$ and TPR$=1$ for all ϵ, producing a ROC curve that degenerates into the point (0,1). A common figure of merit for classifier performance is the Area Under the ROC Curve (AUC), where a higher AUC (closer to 1) indicates a better-performing classifier.

Table 5.2 presents AUC values for LOUPE, r-LOUPE$_1$, and r-LOUPE$_2$, showing that the regularization introduced during the training phase improves the self-assessment capability.

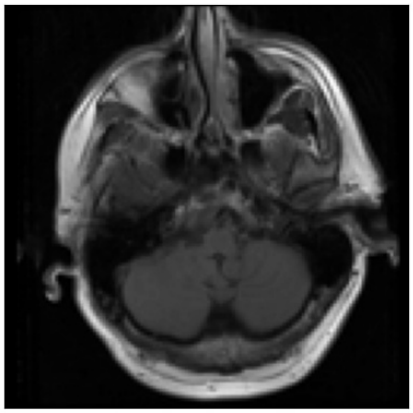


(A)

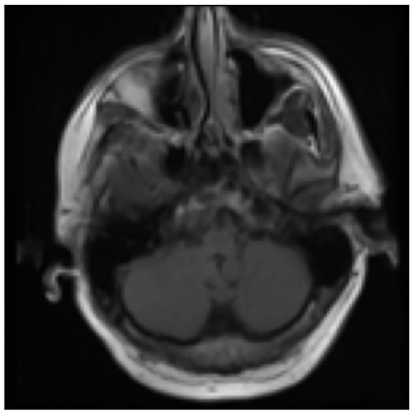


(B)

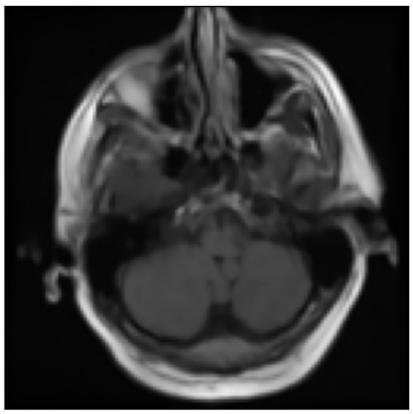
FIGURE 5.3: Estimated probability density functions of PSNR for LOUPE, r-LOUPE$_1$ and r-LOUPE$_2$. In case of (a) $R = 4$ and PSNR values lower than 35 dB are labeled as *bad reconstruction*. In case of (b) $R = 8$ and the PSNR threshold is 30 dB.



(A) $R = 0$



(B) $R = 4$



(C) $R = 8$

FIGURE 5.4: Original MRI scan (a) and reconstructed images with r-LOUPE$_2$ with: (b) $R = 4$ and PSNR=37.5 dB, (c) $R = 8$ and PSNR=31.6 dB.

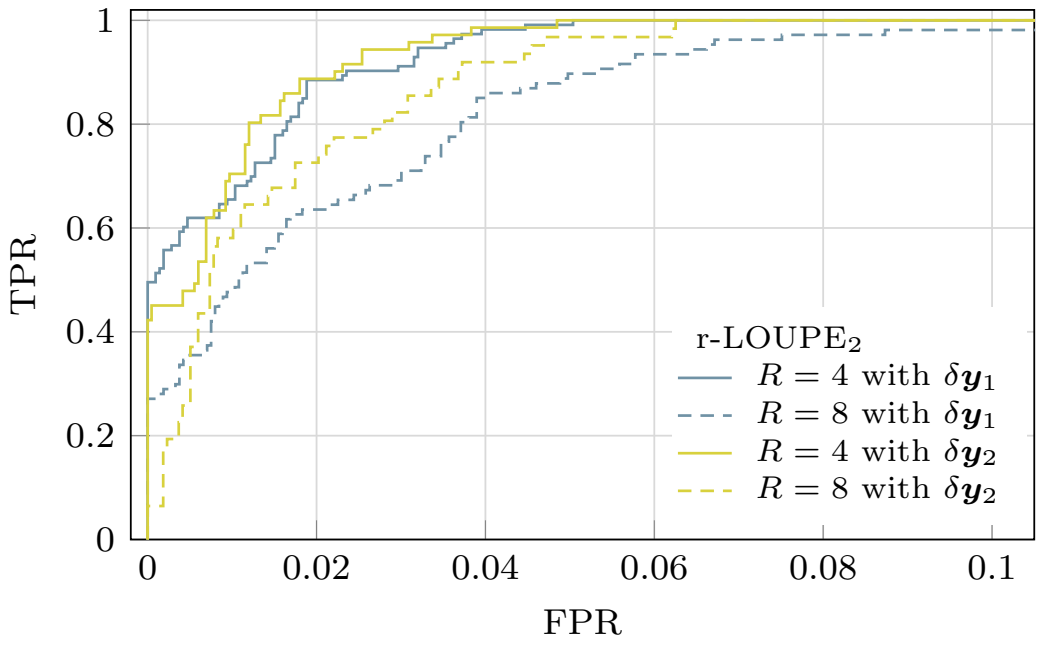
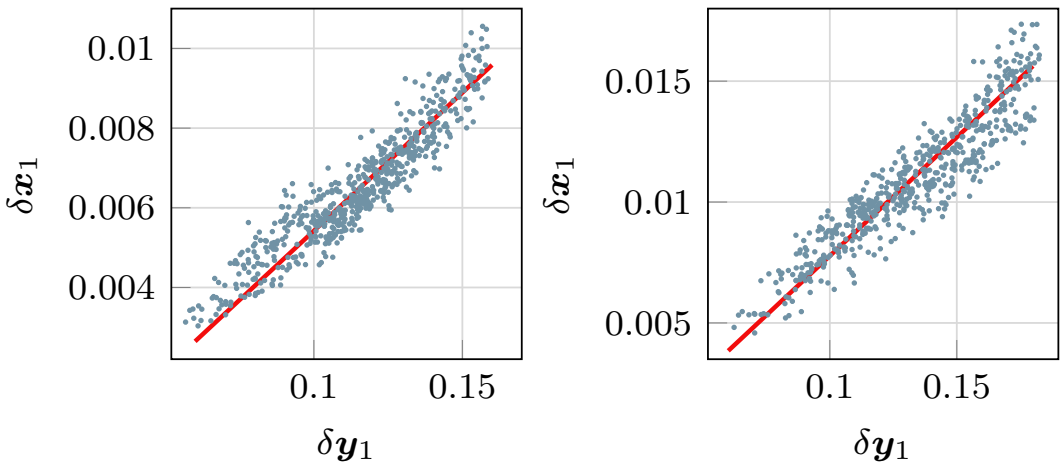


FIGURE 5.5: ROC curve for r-LOUPE$_2$ with $R = \{4, 8\}$ and both $\delta \boldsymbol{y}_1$ and $\delta \boldsymbol{y}_2$.



(A) $R = 4$ (B) $R = 8$

FIGURE 5.6: Scatter plots reporting couples of $\delta \boldsymbol{x}_1$ and $\delta \boldsymbol{x}_2$ associated to r-LOUPE$_2$. Plots also report lines $\delta \hat{\boldsymbol{x}}_1(\delta \boldsymbol{y}_1) = c_0 \delta \boldsymbol{y}_1 + c_1$, used to predict $\delta \boldsymbol{x}_1$. We found: (a) $R = 4$ with $c_0 = 0.06937$ and $c_1 = -0.001515$; (b) $R = 8$ with $c_0 = 0.09815$ and $c_1 = -0.002042$.

TABLE 5.3: Average absolute and relative errors in estimating $\delta \boldsymbol{x}_l$ from $\delta \boldsymbol{y}_p$

l	p	$\lvert\delta \boldsymbol{x}_l - \delta \hat{\boldsymbol{x}}_l\rvert$		$\lvert\delta \boldsymbol{x}_l - \delta \hat{\boldsymbol{x}}_l\rvert / \lvert\delta \boldsymbol{x}_l\rvert$	
		$R = 4$	$R = 8$	$R = 4$	$R = 8$
1	1	4.58e-04	8.19e-04	0.0752	0.0776
1	2	4.67e-05	1.01e-04	0.530	0.350
2	1	6.39e-04	1.06e-03	0.104	0.103
2	2	3.50e-05	8.61e-05	0.434	0.266

The table also reveals minimal differences in AUC when comparing $\delta \boldsymbol{y}_1$ and $\delta \boldsymbol{y}_2$, with r-LOUPE$_2$ consistently outperforming r-LOUPE$_1$. Consequently, we focus on r-LOUPE$_2$ for further analysis, and its ROC curve is displayed in Fig. 5.5.

As mentioned earlier, self-assessment can also be used to estimate $\delta \boldsymbol{x}_l$ by analyzing $\delta \boldsymbol{y}_p$ with $l, p \in \{1, 2\}$. For r-LOUPE$_2$, we use a linear predictor to estimate $\delta \boldsymbol{x}_l$ based on the Eq. $\delta \hat{\boldsymbol{x}}_l = c_0 \delta \boldsymbol{y}_p + c_1$. To mitigate the impact of outliers, we compute the coefficients c_0 and c_1 using the Theil-Sen linear regression method [78, 71] on the training set. Table 5.3 summarizes the prediction performance on the test set in terms of prediction error.

Additionally, Fig. 5.6 shows a visual representation of the estimation of $\delta \boldsymbol{x}_1$ based on $\delta \boldsymbol{y}_1$ for both $R = 4$ and $R = 8$. Across all cases, the relative error in predicting $\delta \boldsymbol{x}_1$ remains below 8%, indicating that the proxy $\delta \boldsymbol{y}_1$ is an effective predictor of reconstruction quality. The scatter plots show a relatively tight clustering around the regression line, indicating a strong linear relationship between the quality estimation error and the reconstruction error. The strength of this relationship suggests that quality estimation errors are reliable predictors of image reconstruction errors, allowing effective self-assessment of reconstruction quality.

5.4 Trade-off Summary

By explicitly enforcing a measurement-preservation constraint within the loss function, Regularized LOUPE ensures that the original k-space frequencies are maintained throughout the reconstruction process. This constraint helps to reduce discrepancies between the acquired data and the reconstructed image, thereby improving the overall accuracy.

Moreover, the incorporation of a self-assessment mechanism via a dedicated deep neural network allows the system to estimate reconstruction quality, adding a value to the returned image. In this context, the estimated quality is used to increase reliability, i.e., enables the identification of potential reconstruction issues. The quality estimation technique has broader potential applications; for instance, it could be utilized to initiate corrective actions or adjustments in the acquisition process. This aspect will be further explored in subsequent chapters of this thesis.

The addition of a regularization term introduces extra components into the reconstruction pipeline, causing the complexity to increase. This may lead to more challenging training dynamics and necessitates additional hyperparameters tuning, such as the weight in the loss.

Chapter 6

Training Binary Layers by Self-Shrinking of Sigmoid Slope

DNNs are governed by a common set of learning principles, in witch training progresses through two fundamental stages. First, a batch of input samples is passed through the network to generate corresponding output predictions, witch are then compared against the true reference labels to compute an average loss or error. In the subsequent stage, this loss is used to calculate gradients through a process known as backpropagation (BP), and the network parameters are adjusted in a direction that aims to minimize the loss.

For effective learning to occur, it is critical to manage the magnitude of these propagated gradients. Ideally, gradients must neither vanish nor explode. Vanishing gradients, characterized by values close to zero, result in negligible parameter updates and thereby hinder learning. Exploding gradients, on the other hand, can lead to unstable updates, adversely affecting the convergence of the model. These two challenges are commonly referred to as the vanishing gradient problem and the exploding gradient problem [22, 4].

A widely adopted strategy to mitigate these problems is the tuning of the learning rate, witch determines the size of parameter updates during training. However, in cases where the architecture itself inherently causes vanishing or exploding gradients, adjusting the learning rate alone often proves insufficient. In these situations, targeted interventions in the network architecture are required to ensure stable gradient propagation.

One such architectural design challenge emerges in the context of custom trainable layers, such as a binary mask layer. When constructing a trainable binary mask, as proposed in [2, 3], it is necessary to replace the hard threshold responsible for generating binary values with a differentiable approximation, often using a soft sigmoid-like function. While a hard threshold effectively stops gradient flow, leading to vanishing gradients, a soft approximation allows the gradients to flow smoothly, supporting efficient backpropagation. The use of a soft threshold introduces a hyperparameter that controls the sharpness of binarization, essentially dictating the trade-off between effective training and strict adherence to binary constraints.

We can think of this hyperparameter as a knob that regulates the "flow" of gradients, with [3] progressively tuning it from a completely relaxed setting to a more rigid one as training advances. However, challenges remains:

1. the optimal strategy for controlling this knob throughout the training process.

2. the degree of "gradients fluidity" to achieve an optimal balance between model performance and adherence to constraints.

To address these questions, we propose a novel training strategy termed *Self-Shrinking of Sigmoid Slope*. This approach involves gradually decreasing the softness of the sigmoid during training, akin to gently turning a knob to restrict the flow of gradients in a controlled manner. By progressively reducing the softness, the network gradually transitions from a relaxed state to a more constrained one, enhancing stability and allowing for more effective parameter updates.

In the following chapter, we will delve into the details of this method, demonstrating how it contributes to more stable and efficient training of binary mask layers, ultimately leading to improved model performance and reduced sensitivity to gradient-related challenges.

6.1 Self-Shrinking of Sigmoid Slope

Although $\boldsymbol{M}_\gamma$ is generated by applying hard thresholding to $\boldsymbol{P}_r(\boldsymbol{S}(\gamma))$ during inference, the sigmoid function involved in the mask generation process during training, as defined in Eq. (4.22), introduces a hyper-parameter s. This parameter can significantly influence the model's final performance. Poorly tuned values of s can lead to two types of errors:

- *i*) When s is too low, many mask entries will not converge to 0 or 1 during training, so the hard thresholding process will alter the mask's behavior significantly at inference time, leading to performance degradation.

- *ii*) When s is too high, the sigmoid function risks experiencing vanishing gradient problems, making training inefficient.

It is clear that the tuning of s is a critical and time-consuming task, often requiring numerous training processes. Moreover, it is essential to note that s cannot be treated as a trainable parameter. If it were, the network would naturally favor $s = 0$, resulting in a degenerate scenario where all entries in $\boldsymbol{M}_\gamma$ are approximately 0.5, leading to an absence of undersampling.

In this work, we introduce a novel method to mitigate the burden of tuning s by implementing a callback function, witch can automatically adjust the slope of the sigmoid function during the training process. While this method is demonstrated in the context of the LOUPE framework, it is equally applicable to a wide variety of problems, as discussed in [26, 19].

A callback function in the context of deep neural networks is a procedure that affects the training process only during specific stages, such as at the end of an epoch. These functions act independently of the backpropagation algorithm and are widely used in modern machine learning workflows. Examples of callback functions include:

- *i*) Early Stopping (ES), witch monitors the model's loss at the end of each epoch and halts training if the loss does not improve for a given number of epochs.

- *ii*) Reduce Learning Rate on Plateau (RLRP), witch lowers the learning rate when the loss remains stagnant for a set number of epochs.

- *iii*) Model Checkpointing (MC), witch saves the model's weights each time the loss decreases, ensuring that the best-performing model is retained.

To alleviate the need for manual tuning of s, we propose a callback function that automatically adjusts the sigmoid slope by gradually increasing s as needed. Specifically, when a set number of epochs (patience) pass without improvement in loss, the callback increases the value of s. This process continues until a predefined maximum slope is reached. Throughout the training process, model checkpointing ensures that only the best-performing weights, associated with the lowest loss, are retained, and the model with the optimal auto-tuned s is returned at the end of training.

We call this method Self-Shrinking of Sigmoid Slope (4S). The parameters of 4S include the patience p_{4S} and a multiplication factor m_{4S} used to increment s after each trigger, such that $s_{i+1} = m_{4S}\ s_i$. In practice, the choice of these parameters is not critical, and we suggest setting $p_{4S} = 40$ and $m_{4S} = 1.5$ as reasonable defaults.

The 4S method has a significant impact for several reasons:

- *i*) As training progresses, the mask $\boldsymbol{M}_\gamma$ becomes more "fossilized" as s increases, resulting in a near-binary mask towards the end of training. This reduces the potential performance drop typically associated with transitioning from a soft mask during training to a hard mask during inference.

- *ii*) The manual tuning of s is eliminated, simplifying the model development process and enhancing training efficiency.

6.2 Numerical evidences

To evaluate the performance of our proposed strategy, we follow the structure outlined in [3] and make use of a subset of the publicly available fastMRI dataset [76], witch has been widely adopted in international competitions such as [57]. This subset includes 2269 normalized grayscale MRI knee images, each with a resolution of 320×320 pixels. These images are divided into three subsets: 1895 images (83.5%) for training, 188 images (8.3%) for validation, and 186 images (8.2%) for testing. The dataset naturally groups scans into volumes, where each volume represents a different knee, with approximately 40 slices per volume. Every slice is normalized by the highest magnitude value in its respective volume to ensure consistency across samples.

In alignment with our proposed model, the undersampling mask $\boldsymbol{M}_\gamma$ is not pre-defined by the user but is entirely learned by the model during training. For inference tests, a deterministic, binarized version of the trained masks is used.

We trained our models on an Nvidia V100 GPU using the Adam optimizer, starting with an initial learning rate of 0.01 and a batch size of 16. The Peak Signal-to-Noise Ratio (PSNR) is used as the primary figure of merit for evaluating reconstruction quality, witch is defined as:

$$\text{PSNR} = -10 \log_{10} (\text{MSE}) \tag{6.1}$$

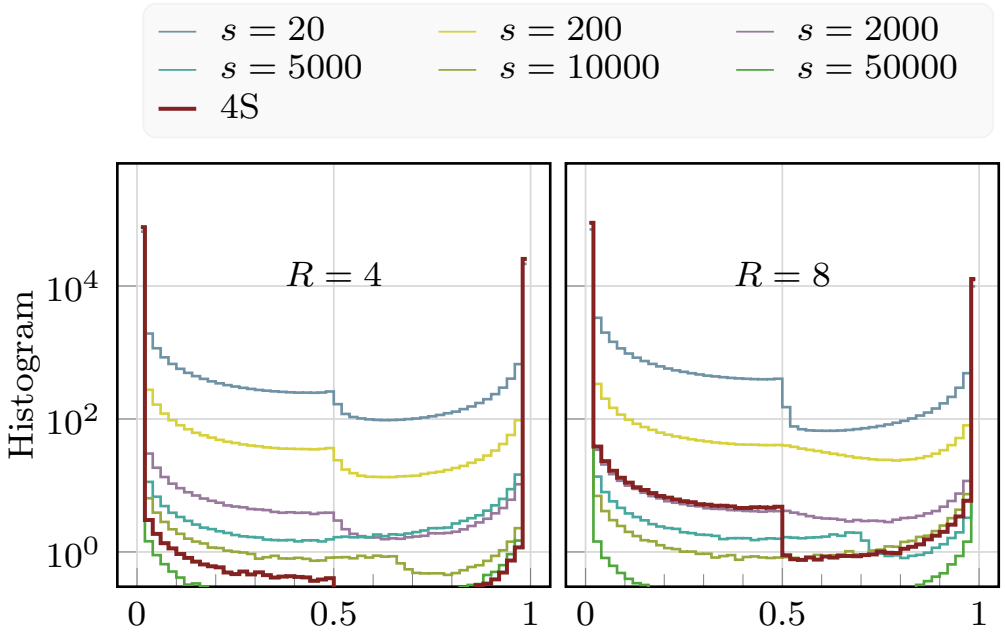


FIGURE 6.1: Histograms for $R = 4$ and $R = 8$ of LOUPE trained with several s values and 4S, that returns two models with $s = 18800$ and $s = 1675$ respectively for $R = 4$ and $R = 8$.

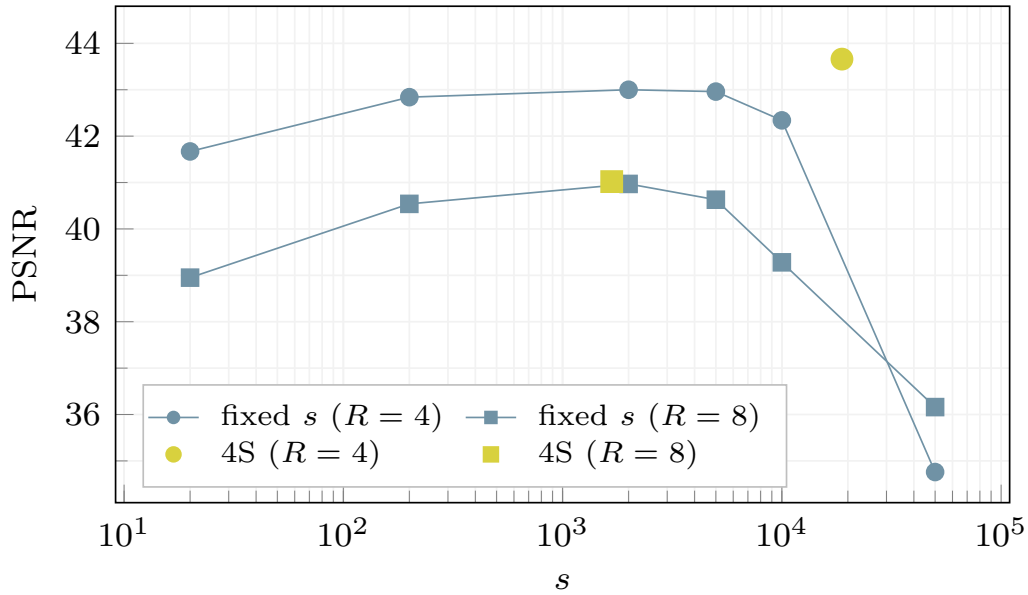


FIGURE 6.2: PSNR for $R = 4$ (circles) and $R = 8$ (squares), trained with several fixed s values (blue points) and 4S (yellow points). 4S models return $s = 18800$ and $s = 1675$ respectively for $R = 4$ and $R = 8$.

where MSE represents the mean squared error between the ground truth image $\boldsymbol{x}$ and the reconstructed image $\hat{\boldsymbol{x}}$ produced by the decoder.

6.2.1 Results and Comparison of Models with 4S

In this section, we compare the performance of models trained with and without our proposed Self-Shrinking Sigmoid Slope (4S) technique. The models trained without 4S follow a traditional training process using Reduce Learning Rate on Plateau (RLRP), Early Stopping (ES), and Model Checkpointing (MC). In contrast, the models trained with 4S first undergo a training phase using 4S, ES, and MC, followed by a second training round where 4S is replaced by RLRP. This two-stage strategy allows the model to explore various slope configurations of s, resulting in more robust performance.

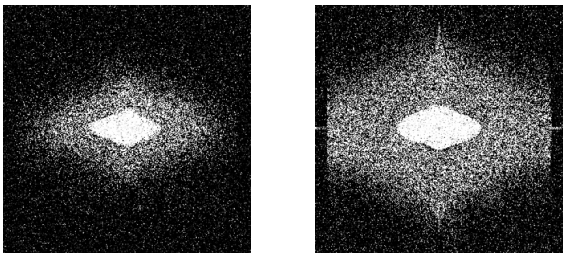
To illustrate how 4S operates, Fig. 6.1 presents histograms of the mask entries for different configurations of $s = \{20, 200, 2000, 5000, 10000, 50000\}$, both with and without 4S. These histograms depict the values of the trained masks (before applying hard thresholding) generated for $R = 4$ and $R = 8$. Specifically, we randomly sampled 1000 masks $\boldsymbol{M}_\gamma$ for each s configuration and plotted the average distribution of their values. As expected, larger values of s yield masks that are nearly binary, while lower s values result in masks with many entries deviating from 0 and 1, thus reducing their binarized quality.

Notably, the 4S technique achieves excellent preservation of the binary structure in the masks, with final s values of 18800 for $R = 4$ and 1675 for $R = 8$. This strong binary structure in conjunction with the PSNR performance, as depicted in Fig. 6.2, further underscores the effectiveness of our approach and the importance of proper s tuning. The figure suggests that the 4S method effectively identifies suitable settings that improve the image reconstruction quality, highlighting its advantage over the fixed sampling approach.

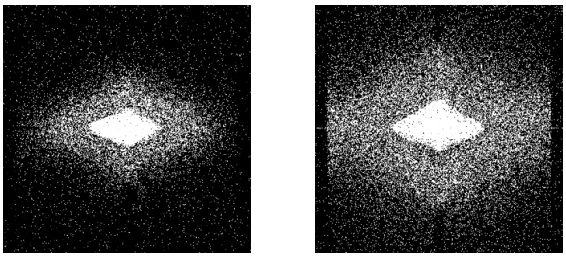
Lastly, Fig. 6.3 provides a visual comparison of the masks generated for models trained with $R = 4$ and $R = 8$, using $s = \{20, 200, 2000, 50000\}$ when applying 4S. As expected, the masks associated with $s = 20$ exhibit many non-binary values, compromising their structure, while the masks generated with $s = 50000$ maintain a more binary configuration but lack adaptability to the signal. In contrast, 4S successfully balances both aspects, producing masks with a robust binary structure that are well-adapted to the input signal.

6.3 Trafe-off Summary

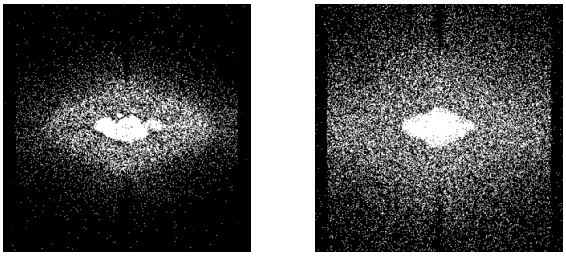
The Self-Shrinking of Sigmoid Slope makes training binary mask layers easier by automating the tuning of a critical hyperparameter, i.e., it ensures a smooth transition to a near-binary state.However, this comes at the cost of increased training complexity.



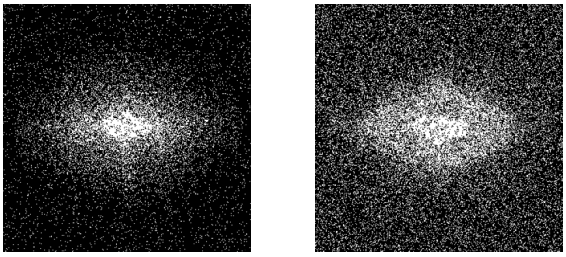
(A) $s = 20$



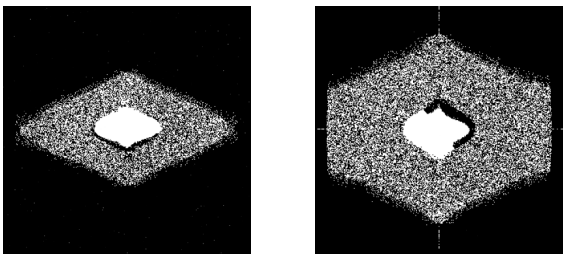
(B) $s = 200$



(C) $s = 2000$



(D) $s = 5000$



(E) 4S

FIGURE 6.3: Probability masks for $R = 8$ (first column) and $R = 4$ (second column), for cases where s is a hyperparameter (fixed $s = \{20, 200, 2000, 50000\}$) and for 4S (e, j) ($s$ is automatically determined at training time). Without 4S, low values of s produce mask values far from either 0 or 1, so that the thresholding operation at inference time becomes critical. High values of s hamper the training, and produce poor masks.

Chapter 7

Embedding CS Constraint by Losses and Projections

The foundational LOUPE architecture has made significant strides in the field of accelerated MRI by introducing a learnable undersampling mask through its encoder-decoder structure. However, despite its innovative approach, LOUPE faces certain limitations that hinder its full potential. One of the main issues lies in the decoder design, witch, while effective, does not fully exploit the information content provided by the undersampled measurements. As a result, the quality of the reconstructed images may fall short, particularly when the undersampling is aggressive.

The encoder in LOUPE plays a critical role in determining witch parts of the frequency space to sample, and it does so in a way that is optimized for the reconstruction process. However, the decoder in LOUPE is relatively standard, focusing primarily on reversing the effects of undersampling without adequately incorporating sophisticated corrections for the artifacts that inevitably arise. This results in a potential loss of fine details, affecting the overall quality and reliability of the reconstructed images.

To address these limitations, we propose a new approach that maintains the strengths of LOUPE's encoder while redesigning the decoder to better utilize the available measurements. Our new decoder structure introduces additional layers specifically aimed at reducing reconstruction artifacts and enhancing the quality of the final output. By incorporating deeper correction mechanisms, our model is capable of achieving more accurate and detailed reconstructions, witch are crucial for high-quality MRI applications.

Furthermore, our approach introduces a novel method to estimate the reconstruction error directly from the measurements. This estimation serves as a proxy for image quality and allows us to add a predictive capability to our model. By leveraging this feature, the model can anticipate the quality of its reconstructions in real-time, offering a from of self-assessment. This is particularly valuable in clinical settings, where ensuring the quality of reconstructed images is paramount for accurate diagnosis.

In summary, our proposed model builds on the strengths of LOUPE but addresses its key shortcomings by:

- Modifying the decoder to more effectively correct for undersampling artifacts, thereby significantly enhancing the quality of the reconstructed images.

- Introducing an error estimation mechanism that allows for real-time quality assessment, paving the way for adaptive acquisition strategies that ensure consistent image quality.

This new approach aims to push the boundaries of accelerated MRI, making it not only faster but also more reliable, with built-in mechanisms for quality control. By addressing both the reconstruction and the assessment aspects of MRI, we provide a more holistic solution that promises better performance and robustness in practical applications.

7.1 Enhanced LOUPE

The original architecture and its training procedure can be improved in several directions. To do so, we begin by defining a set of convex subsets of the space of complex-valued images $\mathbb{I}_{\mathbb{C}} = \mathbb{C}^{d\times d}$:

$$\begin{aligned} \mathbb{I}_{\boldsymbol{y}} &= \{\boldsymbol{x} \mid \boldsymbol{y} = \boldsymbol{M}_{\gamma} \circ \mathcal{F}(\boldsymbol{x})\} \\ \mathbb{I}_{\mathbb{R}} &= \mathbb{R}^{d\times d} \\ \mathbb{I}_{[0,1]} &= [0,1]^{d\times d} \end{aligned}$$

Notice that the original decoder Dec0$\cdot$ ensures that the reconstruction $\hat{\boldsymbol{x}}$ belongs to $\mathbb{I}_{\mathbb{R}}$, i.e., it is real-valued, but it does not guarantee that $\hat{\boldsymbol{x}} \in \mathbb{I}_{[0,1]}$, meaning the reconstruction may not always be within the valid image range $[0,1]$.

For a generic subset $\mathbb{I} \subset \mathbb{I}_{\mathbb{C}}$, we define $\Pi_{\mathbb{I}}(\cdot)$ as the projection operator that projects any input onto $\mathbb{I}$. The projections onto $\mathbb{I}_{\mathbb{R}}$ and $\mathbb{I}_{[0,1]}$ are given by:

$$\begin{aligned} \Pi_{\mathbb{I}_{\mathbb{R}}}(\boldsymbol{x}) &= \operatorname{Re}\{\boldsymbol{x}\} \\ \Pi_{\mathbb{I}_{[0,1]}}(\boldsymbol{x}) &= \max\{0, \min\{1, \operatorname{Re}\{\boldsymbol{x}\}\}\} \end{aligned}$$

where all operations are applied element-wise over the matrix.

For the projection onto $\mathbb{I}_{\boldsymbol{y}}$, observe that the encoding operator $\operatorname{Enc}(\cdot) = \boldsymbol{M}_{\gamma} \circ \mathcal{F}(\cdot)$ is a non-injective linear operator that maps onto a (rn)-dimensional subspace (where r is the undersampling ratio). The projection $\Pi_{\mathbb{I}_{\boldsymbol{y}}}(\boldsymbol{x})$ can be computed as:

$$\Pi_{\mathbb{I}_{\boldsymbol{y}}}(\boldsymbol{x}) = \arg\min_{\varsigma} \|\varsigma - \boldsymbol{x}\|_2 \quad \text{s.t.} \quad \operatorname{Enc}(\varsigma) = \boldsymbol{y} \tag{7.1}$$

The solution to this minimization is given by:

$$\Pi_{\mathbb{I}_{\boldsymbol{y}}}(\boldsymbol{x}) = \boldsymbol{x} + \operatorname{Enc}^{+}(\boldsymbol{y} - \operatorname{Enc}(\boldsymbol{x})) \tag{7.2}$$

where $\operatorname{Enc}^{+}(\cdot)$ denotes the Moore-Penrose pseudo-inverse of $\operatorname{Enc}(\cdot)$.

To derive this projection, note that the encoding operator can be split into two components: $\operatorname{Enc}^{\perp}(\cdot) = (\mathbf{1} - \boldsymbol{M}_{\gamma}) \circ \mathcal{F}(\cdot)$, witch complements $\operatorname{Enc}(\cdot)$ by operating on the masked

out Fourier components. The Fourier transform of $\Pi_{\mathbb{I}_y}(\boldsymbol{x})$ is then:

$$\begin{aligned}\mathcal{F}(\Pi_{\mathbb{I}_y}(\boldsymbol{x})) &= \text{Enc}(\boldsymbol{x}) + \text{Enc}^{\perp}(\boldsymbol{x}) + \boldsymbol{y} - \text{Enc}(\boldsymbol{x}) + \text{Enc}^{\perp}(\text{Enc}^{+}(\boldsymbol{y} - \text{Enc}(\boldsymbol{x}))) \\ &= \text{Enc}^{\perp}(\boldsymbol{x}) + \boldsymbol{y}\end{aligned}$$

where we used the fact that $\text{Enc}^{\perp}(\text{Enc}^{+}(\cdot)) = 0$. This gives the final projection in the image domain:

$$\Pi_{\mathbb{I}_y}(\boldsymbol{x}) = \mathcal{F}^{-1}((\mathbf{1} - \boldsymbol{M}_{\gamma}) \circ \mathcal{F}(\boldsymbol{x}) + \boldsymbol{y}) \tag{7.3}$$

Thus, $\Pi_{\mathbb{I}_y}(\boldsymbol{x})$ is obtained by performing a Fourier transform, substituting the available measurements $\boldsymbol{y}$ into the transformed image, and then applying an inverse Fourier transform.

We propose several improvements to the original decoder architecture, starting with a slight modification of its structure (illustrated in Fig. 7.1), witch produces the estimate:

$$\hat{\boldsymbol{x}}_{\star} = \text{Dec}^{1}(\boldsymbol{y}) = |\bar{\boldsymbol{x}}| = \left|\mathcal{F}^{-1}(\boldsymbol{y}) + \mathcal{D}_{\boldsymbol{\theta}}^{\mathbb{C}}\left(\mathcal{F}^{-1}(\boldsymbol{y})\right)\right| \tag{7.4}$$

where $\bar{\boldsymbol{x}}$ is an intermediate complex estimate, and $\mathcal{D}_{\boldsymbol{\theta}}^{\mathbb{C}}$ is a neural network responsible for correcting both the real and imaginary parts of the reconstruction. In practice, $\mathcal{D}_{\boldsymbol{\theta}}^{\mathbb{C}}$, maintains the same architecture as $\mathcal{D}_{\boldsymbol{\theta}}^{\mathbb{R}}$ but its output is complex valued (two channels output, one for the real and one for the imaginary part).

This modification introduces an additional error metric, $\delta\bar{\boldsymbol{y}}$, witch measures the difference between the observed measurements $\boldsymbol{y}$ and those produced by the intermediate estimate $\bar{\boldsymbol{x}}$:

$$\delta\bar{\boldsymbol{y}} = \|\boldsymbol{y} - \text{Enc}(\bar{\boldsymbol{x}})\|_2 \tag{7.5}$$

A good reconstruction should yield a small value of $\delta\bar{\boldsymbol{y}}$. This new information can be incorporated into the training process by defining a revised loss function:

$$L^{1}(\boldsymbol{\gamma}, \boldsymbol{\theta}) = \mathbf{E}_{\boldsymbol{x}}\left[(1 - \phi)\delta\boldsymbol{x} + \phi\delta\bar{\boldsymbol{y}}\right] \tag{7.6}$$

where $\phi \in [0, 1]$ is a user-defined parameter that balances the importance of $\hat{\boldsymbol{x}}$ being close to the ground truth $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$ reproducing the correct measurements. This multi-term loss allows us to train the decoder with more attention to the accuracy of the measurements, thus improving the overall quality of the reconstruction.

As an alternative approach, one can integrate the measurement matching criterion directly into the signal chain, leading to the definition of the following decoder structure (see third decoder architecture in Fig. 7.1):

$$\hat{\boldsymbol{x}} = \text{Dec}^{2}(\boldsymbol{y}) = \left|\Pi_{\mathbb{I}_y}\left(\mathcal{F}^{-1}(\boldsymbol{y}) + \mathcal{D}_{\boldsymbol{\theta}}^{\mathbb{C}}(\mathcal{F}^{-1}(\boldsymbol{y}))\right)\right| \tag{7.7}$$

This decoder, $\text{Dec}^{2}(\cdot)$, incorporates a projection onto $\mathbb{I}_y$ of $\bar{\boldsymbol{x}}$, thus improving the measurement matching criterion during the processing phase. It is typically trained using the loss function $L^{0}(\boldsymbol{\gamma}, \boldsymbol{\theta})$.

A significant advantage of using $\text{Dec}^{2}(\cdot)$ is its ability to enhance the alignment of the

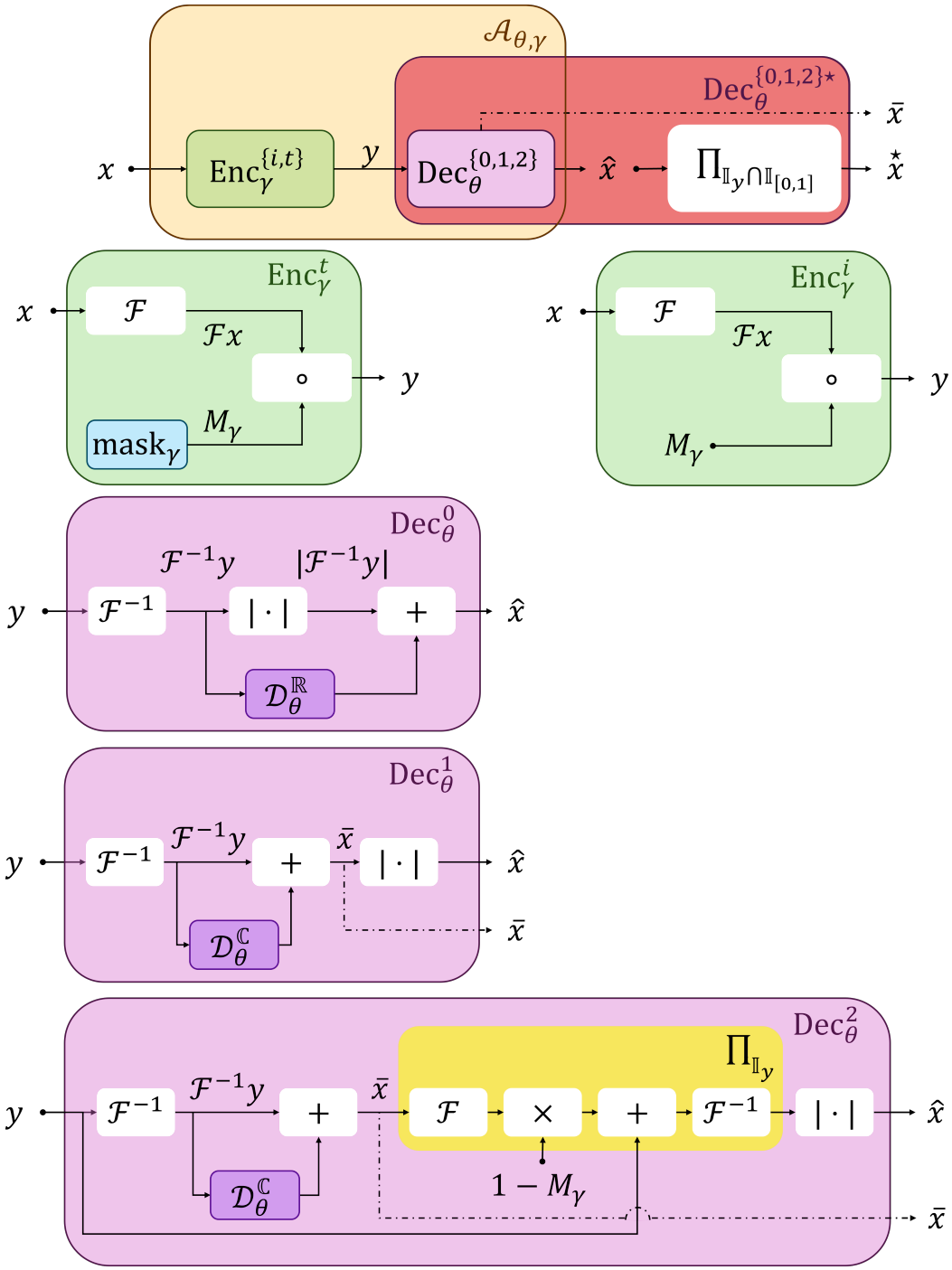


FIGURE 7.1: Schematic representation of all LOUPE adaptations.

estimated image with the actual measurements. This inclusion directly refines the decoder's ability to match the measurements observed during the acquisition phase.

Additionally, regardless of the specific inner decoder employed, a final stage can be appended to ensure that the reconstructed image meets all the necessary criteria, defined as:

$$\overset{\star}{\boldsymbol{x}} = \mathrm{Dec}^{i\star}(\boldsymbol{y}) = \Pi_{(\mathbb{I}_{\boldsymbol{y}} \cap \mathbb{I}_{[0,1]})} \left(\mathrm{Dec}^{i}(\boldsymbol{y}) \right) \tag{7.8}$$

The projection $\Pi_{(\mathbb{I}_{\boldsymbol{y}} \cap \mathbb{I}_{[0,1]})}$ can be computed using Dykstra's alternating projections algorithm [16, 12], leveraging the elementary projections onto $\mathbb{I}_{\boldsymbol{y}}$ and $\mathbb{I}_{[0,1]}$.

Incorporating a final projection stage guarantees that the final image reconstruction is consistent with the observed measurements and lies within the feasible image range $[0, 1]$. This strategy ensures that the reconstruction respects the available Fourier coefficients. However, it is important to note that during the decoding stage, the Fourier components of $\boldsymbol{x}$ that are not directly measured remain unknown and cannot be utilized in the reconstruction process.

Nevertheless, during training, all Fourier components of the ground-truth image $\boldsymbol{x}$ are available. To take advantage of this, one can improve the performance of $\mathrm{Dec}^{i\star}(\cdot)$ by incorporating a loss function that accounts for the non-measured Fourier components. This leads to the definition of the error term:

$$\delta \boldsymbol{y}^{\perp} = \left\| \mathrm{Enc}^{\perp}(\bar{\boldsymbol{x}}) - \mathrm{Enc}^{\perp}(\boldsymbol{x}) \right\|_2$$

This error captures how well the intermediate estimate $\bar{\boldsymbol{x}}$ matches the Fourier components that $\boldsymbol{x}$ would have produced if they had been acquired. This criterion can be used to define an alternative loss function:

$$L^2(\boldsymbol{\gamma}, \boldsymbol{\theta}) = \mathbf{E}_{\boldsymbol{x}} \left[(1 - \psi)\delta \boldsymbol{x} + \psi \delta \boldsymbol{y}^{\perp} \right] \tag{7.9}$$

where $\psi \in [0, 1]$ is a user-defined weight that balances the reconstruction error and the mismatch of the non-acquired Fourier components.

It is worth mentioning that this approach has shown to yield substantial performance improvements when combined with the decoder $\mathrm{Dec}^{1\star}(\cdot)$.

7.2 Self-assessment

All the enhanced decoders we propose take advantage of the fact that high-quality reconstructions should exhibit small values of $\delta \bar{\boldsymbol{y}}$.

In an ideal reconstruction pipeline, each estimated image should closely approximate the original image, thereby generating measurements that align with the acquired ones. Thus, we can reasonably assume that defining:

$$\delta \hat{\boldsymbol{y}} = \| \boldsymbol{y} - \mathrm{Enc}(\hat{\boldsymbol{x}}) \|_2 \tag{7.10}$$

when the reconstruction process fails, both $\delta \bar{\boldsymbol{y}}$ and/or $\delta \hat{\boldsymbol{y}}$ will be significantly larger than zero.

Consequently, $\delta\bar{\boldsymbol{y}}$ and $\delta\hat{\boldsymbol{y}}$, both computable using only the actual measurements $\boldsymbol{y}$, have magnitudes that serve as proxies for the decoder's performance. This insight can be leveraged in a self-assessment stage, witch follows the reconstruction process. The purpose of this stage is to provide the user with additional feedback about the quality of the reconstructed image during inference, i.e., when the ground truth image $\boldsymbol{x}$ is unknown, and the system relies solely on the frequency components present in $\boldsymbol{y}$.

To evaluate the effectiveness of $\delta\bar{\boldsymbol{y}}$ and/or $\delta\hat{\boldsymbol{y}}$ as predictors of decoder performance, we compute the Pearson correlation coefficient,

$$\rho(a,b) = \frac{\mathrm{cov}(a,b)}{\sigma_a \sigma_b},$$

where $\mathrm{cov}(\cdot,\cdot)$ denotes the covariance between two vectors, and $\sigma_\cdot$ refers to their standard deviation.

We adopt $\rho(\delta\hat{\boldsymbol{x}}, \delta\hat{\boldsymbol{y}})$ for $\mathrm{Dec}^0(\cdot)$, $\rho(\delta\overset{\star}{\boldsymbol{x}}, \delta\hat{\boldsymbol{y}})$ for $\mathrm{Dec}^{0\star}(\cdot)$, and $\rho(\delta\overset{\star}{\boldsymbol{x}}, \delta\bar{\boldsymbol{y}})$ for the other proposed decoders, where $\delta\overset{\star}{\boldsymbol{x}} = \left\| \boldsymbol{x} - \overset{\star}{\boldsymbol{x}} \right\|_1$.

7.3 Numerical Evidence

In this section, we present numerical evidence supporting the effectiveness of the proposed approach for MRI undersampling and reconstruction. We provide comparisons with standard undersampling methods, highlighting improvements in reconstruction quality.

7.3.1 Dataset

To ensure consistency with [3], we evaluated our models using a subset of the publicly available NYU fastMRI dataset [76], witch has been widely used in international MRI reconstruction competitions [31, 57].

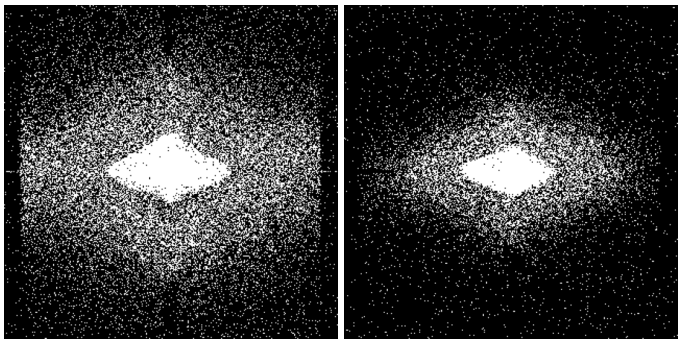
The selected dataset consists of single-coil proton density-weighted scans, emulated from the original 15-channel multi-coil raw data, acquired using the Biograph mMR scanner.

The dataset is structured in *volumes*, with each volume consisting of multiple grayscale slices. Each slice is of size $N = 320 \times 320$, and pixel values are normalized into the range $[0,1]$ by dividing by the maximum magnitude within the respective volume.

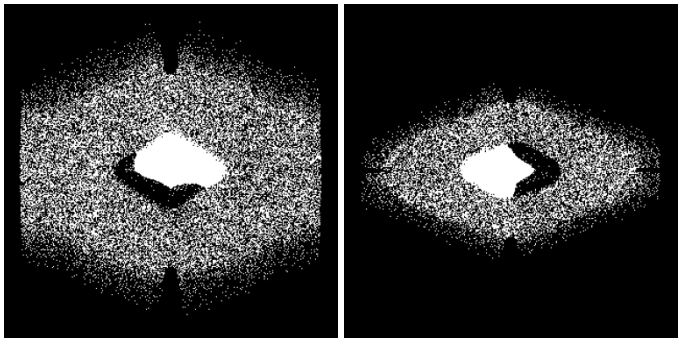
The dataset was split into three parts: 50 volumes for training, 5 volumes for validation, and 5 volumes for testing. Training volumes contain between 34 to 42 slices each, validation volumes between 34 to 40 slices, and test volumes between 35 to 45 slices. This yields a total of 1895 training slices, 188 validation slices, and 186 test slices.

7.3.2 Training and Evaluation

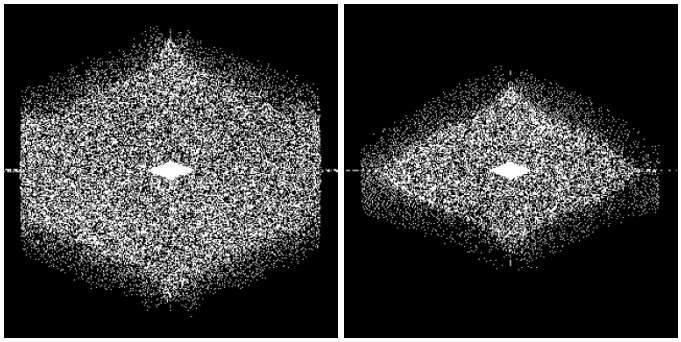
The original $\mathcal{D}_\theta^{\mathbb{R}}$ and our modified $\mathcal{D}_\theta^{\mathbb{C}}$ architectures have nearly the same number of trainable parameters (31,504,323 vs. 31,504,388). Training was performed on an Nvidia V100 GPU using the Adam optimizer, with an initial learning rate of 0.01 and a batch size of 16.



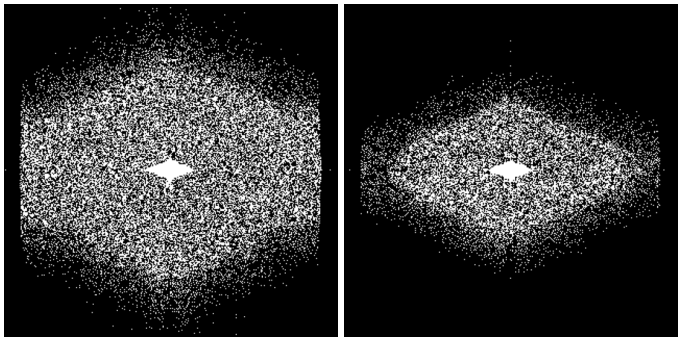
(A) $\mathrm{Dec}^0|L^0$



(B) $\mathrm{Dec}^1|L^1$



(C) $\mathrm{Dec}^1|L^2$



(D) $\mathrm{Dec}^2|L^0$

FIGURE 7.2: Undersampling masks $\boldsymbol{M}_\gamma$ for all considered decoders (starred and non-starred versions share the same undersampling mask). The first column is trained with $R = 4$, the second with $R = 8$.

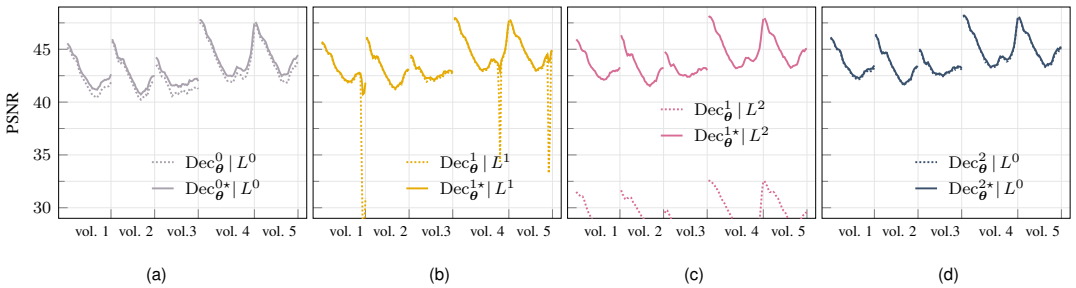


FIGURE 7.3: Effect of final projection on various decoders for $R = 4$.

To evaluate the reconstruction quality, we used the Peak Signal-to-Noise Ratio (PSNR) metric, witch is defined as:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\max\{\boldsymbol{x}\}}{\mathrm{MSE}} \right) \tag{7.11}$$

where MSE is the mean square error between the ground truth image $\boldsymbol{x}$ and the reconstructed image $\hat{\boldsymbol{x}}$.

We evaluated our models using $R = 4$ and $R = 8$. For each R, we tested four configurations of decoder architectures and training loss functions: $\mathrm{Dec}^0|L^0$ (original LOUPE [3]), $\mathrm{Dec}^1|L^1$ (witch emphasizes the ability to reproduce physical measurements), $\mathrm{Dec}^1|L^2$ (witch focuses on predicting the Fourier coefficients not captured during acquisition), and $\mathrm{Dec}^2|L^0$ (witch incorporates an approximate measurement constraint directly in the network architecture rather than the loss function).

In all cases, we also evaluated the performance of the "starred" version $\mathrm{Dec}^{i\star}$, witch includes a final projection stage to ensure that the reconstructed image satisfies the physical measurement constraints.

The best values of ϕ in $\mathrm{Dec}^1|L^1$ were determined experimentally as $\phi = 10^{-5}$ for $R = 4$ and $\phi = 10^{-4}$ for $R = 8$. For $\mathrm{Dec}^1|L^2$, the optimization yielded $\psi = 1$ for both $R = 4$ and $R = 8$, suggesting that the best performance is obtained by focusing solely on the regularization term, witch aims to predict the Fourier coefficients not captured during acquisition. This approach performs best when combined with a final projection stage.

7.3.3 Undersampling Masks

To demonstrate how encoder adaptation functions in different architectures, Fig. 7.2 shows the resulting fixed binary masks $\boldsymbol{M}_{\gamma}$ from Eq. (4.27) for each trained model. The white dots indicate the Fourier coefficients that are retained by the system during acquisition. As expected, the number of retained coefficients decreases with increasing R.

It is also evident that different decoder architectures and training strategies result in different undersampling masks, as the encoder is co-optimized with the decoder. This highlights the interplay between undersampling patterns and decoder performance in CS-based MRI acquisition frameworks.

7.3.4 Results

The performance plots for the overall signal chain have a consistent format across different configurations. Each plot shows the results for all the slices in the 5 test set volumes. The horizontal axis corresponds to slice positions, where the first slice of volume $i+1$ directly follows the last slice of volume i. Each slice's reconstruction quality is represented by a point on the vertical axis, corresponding to the PSNR. The non-uniformity of the slices and the volumes' complexity results in a varied PSNR profile across the plots.

Fig. 7.3 illustrates the impact of adding the projection operator $\Pi_{\mathbb{I}_y \cap \mathbb{I}_{[0,1]}}$ to modify Dec^i into $\text{Dec}^{i\star}$ for $R = 4$. Dotted lines represent the non-starred configurations, while solid lines correspond to the starred versions. The comparison highlights the effect of the final projection.

The final projection step shows minimal impact for $\text{Dec}^2|L^0$, as the architecture of Dec^2 already includes $\Pi_{\mathbb{I}_y}$, the primary component of $\Pi_{\mathbb{I}_y \cap \mathbb{I}_{[0,1]}}$. For the other architectures, $\Pi_{\mathbb{I}_y \cap \mathbb{I}_{[0,1]}}$ proves especially effective for the more challenging slices, with dramatic improvements when the training loss function is L^2. This is expected because, with $\psi = 0$, the network focuses on predicting the Fourier coefficients not acquired, witch complements the role of $\Pi_{\mathbb{I}_y \cap \mathbb{I}_{[0,1]}}$.

Overall, the highest performance on this dataset is achieved by $\text{Dec}^{1\star}|L^2$, followed closely by $\text{Dec}^{2\star}|L^0$. To quantify these improvements over the original LOUPE ($\text{Dec}^0|L^0$), we present the average, minimum, and maximum PSNR values in Table 7.1 for $R = 4$ and in Table 7.2 for $R = 8$. Notably, the proposed architectures deliver a PSNR improvement of up to 2.1 dB for $R = 4$ and 1.7 dB for $R = 8$ compared to the original LOUPE.

Additionally, Table 7.3 provides the comparison matrix for the test set, showing how many slices were better reconstructed with the configuration in the column compared to the row configuration. This table applies to $R = 4$ and shows a similar trend for $R = 8$. Furthermore, Fig. 7.4 compares the proposed methods (solid lines) with LOUPE ($\text{Dec}^0|L^0$) (dashed line) for both $R = 4$ and $R = 8$. As expected, increasing the acquisition ratio from $R = 4$ to $R = 8$ results in a PSNR reduction of approximately 3 dB (as also shown in Tables 7.1 and 7.2).

These numerical results underscore the significant improvements of the proposed architectures over the baseline LOUPE model in [3]. Specifically, $\text{Dec}^{2\star}|L^0$ consistently delivers the best performance across all test set slices. For a visual comparison, Fig. 7.5 shows reconstructed images using $\text{Dec}^{2\star}|L^0$ and the original LOUPE model, along with the corresponding ground truth images.

Table 7.4 reports the Pearson correlation coefficient between the reconstruction errors at the decoders' outputs and the measurement approximation errors computed from the same decoder outputs.

Excluding the case of $\text{Dec}^{1*}|L^1$, witch shows less promising results, the correlation values are generally strong, indicating a good agreement between decoder performance and the adherence to the measurement constraint. These results suggest that the error in the approximation of the measurements could serve as a proxy for evaluating the quality of the reconstructed image. This ability to self-assess the decoder's output may be useful for designing a

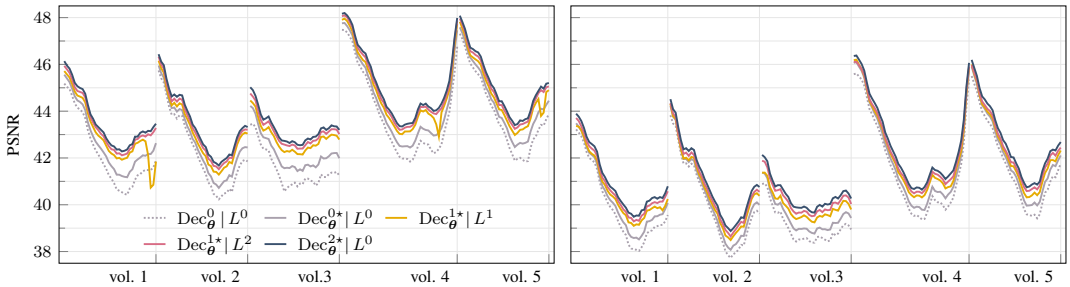
TABLE 7.1: Average (μ), minimum (min) and maximum (max) values of PSNR [dB] for the five volumes composing the test set and $R = 4$.

		$\text{Dec}^0\|L^0$	$\text{Dec}^{0\star}\|L^0$	$\text{Dec}^{1\star}\|L^1$	$\text{Dec}^{1\star}\|L^2$	$\text{Dec}^{2\star}\|L^0$
vol.1	μ	42.2	42.7	43.1	43.4	43.6
	min	40.4	41.2	40.7	42.1	42.3
	max	45.2	45.6	45.7	45.9	46.1
vol.2	μ	42.3	42.7	43.1	43.3	43.4
	min	40.2	40.7	41.3	41.5	41.7
	max	45.7	46.0	46.2	46.3	46.4
vol.3	μ	41.5	42.2	42.8	43.1	43.3
	min	40.6	41.4	42.1	42.4	42.5
	max	43.5	44.2	44.5	44.8	45.0
vol.4	μ	43.9	44.3	44.7	45.0	45.1
	min	41.8	42.4	42.9	43.2	43.3
	max	47.5	47.8	48.0	48.1	48.2
vol.5	μ	43.9	44.4	44.7	45.0	45.1
	min	41.8	42.5	43.0	43.2	43.4
	max	47.3	47.6	47.8	48.0	48.1

detector capable of identifying poorly reconstructed images, a topic we intend to explore in future work.

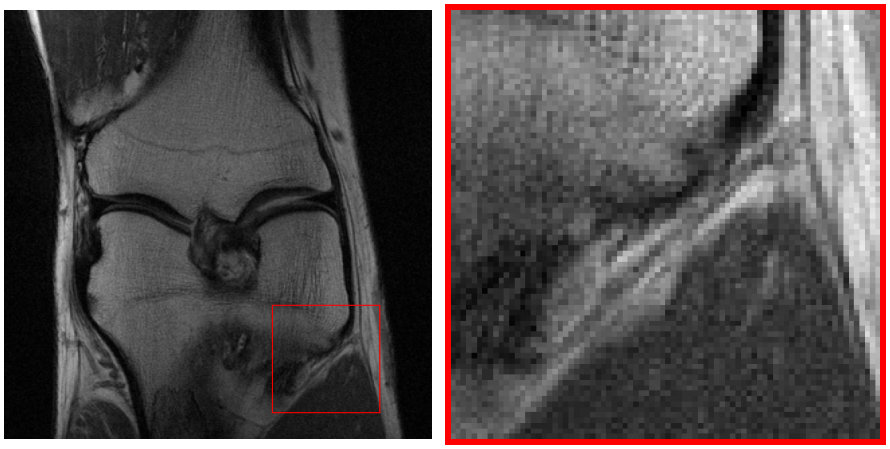
To illustrate the meaning of the computed correlation coefficients, Fig. 7.6 presents scatter plots showing the relationship between reconstruction errors and the errors in fitting the measurement constraint. [1] Interestingly, all plots in Fig. 7.6 share the same y-axis scale, but the x-axis scales vary significantly. This is due to the fact that, during training, $\text{Dec}^0|L^0$ directly minimizes $\delta\hat{\boldsymbol{y}}$, while $\text{Dec}^1|L^2$ and $\text{Dec}^2|L^0$ do not explicitly minimize $\delta\tilde{\boldsymbol{y}}$.

[1] The results for $\text{Dec}^{1*}|L^1$ are not included as they do not provide meaningful insights.

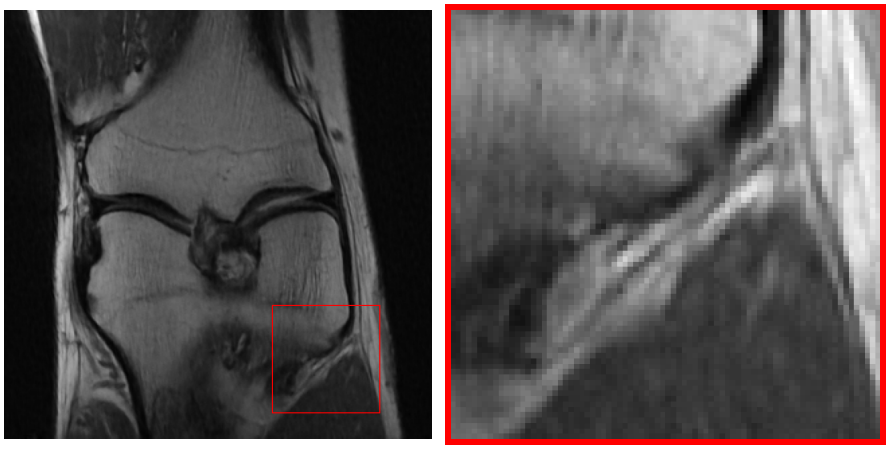


(a) $R = 4$ (b) $R = 8$

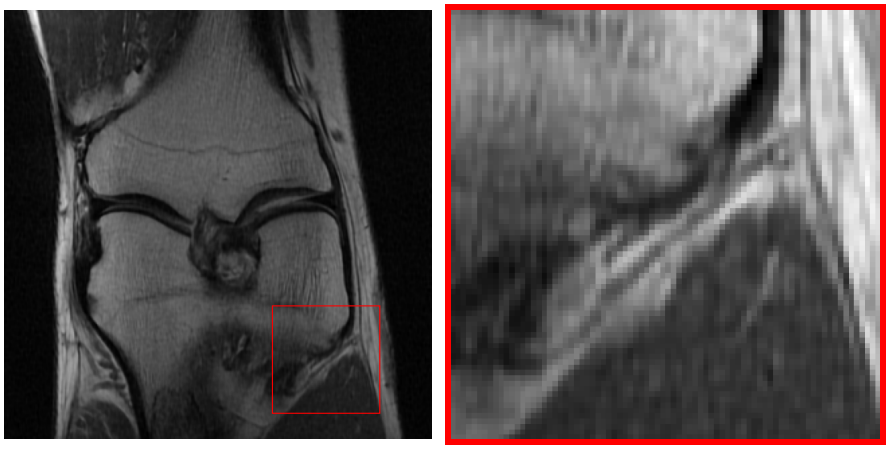
FIGURE 7.4: PSNR values for all considered decoders and for the 5 volumes composing the test set. Volumes number 1,2,3 and 5 include 35 slices while 45 slides are in volume 4.



(A) ground truth



(B) $\mathrm{Dec}^{0}|L^{0}$



(C) $\mathrm{Dec}^{2\star}|L^{0}$

FIGURE 7.5: Left side of (a) represents a single Knee image from the dataset "fastMRI", volume 2 and slice number 24 in the test set. Left sides of (b) and (c) correspond to reconstructed images by $\mathrm{Dec}^{0}|L^{0}$ with PSNR equal to 37.7 dB, and $\mathrm{Dec}^{2\star}|L^{0}$ with 38.9 dB. Reconstructions refer to $R = 8$. Right side images highlight the region of the images in the left sides in the red box.

TABLE 7.2: Average (μ), minimum (min) and maximum (max) values of PSNR [dB] for the five volumes composing the test set and $R = 8$.

		$\text{Dec}^0\|L^0$	$\text{Dec}^{0\star}\|L^0$	$\text{Dec}^{1\star}\|L^1$	$\text{Dec}^{1\star}\|L^2$	$\text{Dec}^{2\star}\|L^0$
vol.1	μ	39.9	40.3	40.6	40.8	41.0
	min	38.0	38.5	39.1	39.3	39.5
	max	43.2	43.5	43.4	43.7	43.9
vol.2	μ	40.1	40.4	40.6	40.8	41.0
	min	37.7	38.1	38.5	38.6	38.9
	max	43.8	44.2	44.3	44.3	44.5
vol.3	μ	39.1	39.5	39.9	40.2	40.4
	min	38.4	38.8	39.2	39.5	39.7
	max	40.9	41.4	41.4	41.9	42.1
vol.4	μ	41.6	42.0	42.3	42.4	42.6
	min	39.4	39.8	40.3	40.5	40.6
	max	45.6	46.1	46.2	46.1	46.4
vol.5	μ	41.7	42.1	42.3	42.5	42.7
	min	39.4	40.0	40.3	40.6	40.7
	max	45.5	45.9	46.0	46.0	46.2

7.4 Trade-off Summary

Our method explicitly incorporates a measurement-preservation constraint into both the loss function and the decoder architecture. By enforcing this constraint, our approach enhances image reconstruction quality, ensuring that the recovered signals remain faithful to the acquired measurements while still allowing for learned optimizations in the encoding and decoding process.

One of the key advantages of our approach is its ability to discover new undersampling patterns. In particular, as demonstrated for $\text{Dec}^{2\star}|L^0$, learn acquisition patterns in a novel manner. Our approach jointly optimizes the encoder and decoder, hence the new patters are a result of double optimization.

Another advantage is that the improved reconstruction quality comes with a negligible increase in computational cost and model complexity, in particular for the case of $\text{Dec}^{2\star}|L^0$. The new incorporation does not introduce significant overhead, making the approach as feasible as its predecessor.

One drawback of our experiments is on undersampling patterns: without extensive evaluation on commonly used undersampling patterns, the demonstrated performance need further validation to improve generalization. Nevertheless, as a side result, in the following chapter the method will be demonstrated to work on Cartesian sampling.

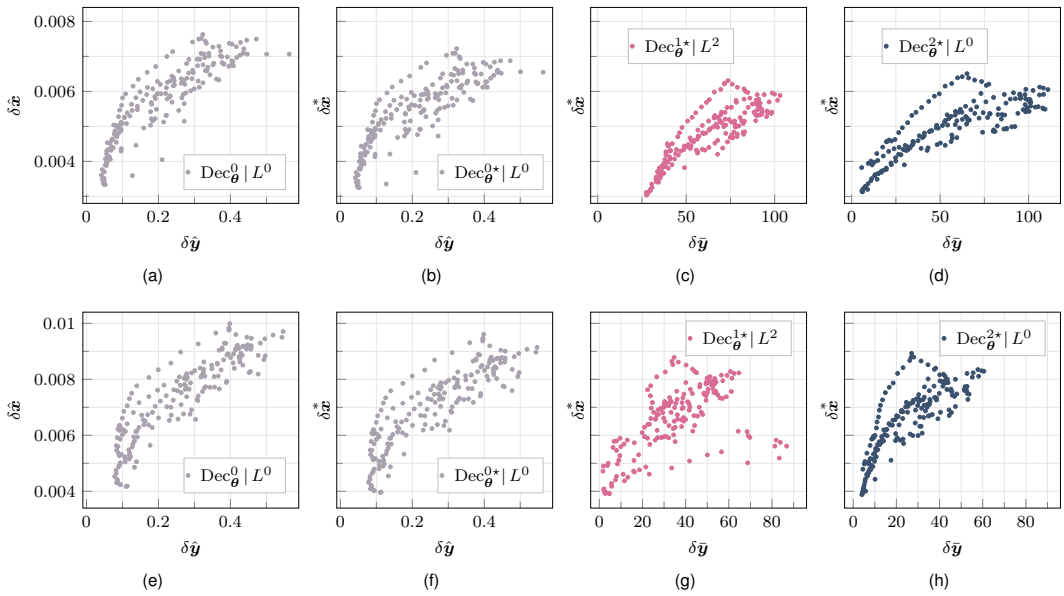


FIGURE 7.6: Error in the measurement constraint fitting versus reconstruction error for the images composing the test set. Results are for all considered approaches except for $\text{Dec}^{1*}|L^1$. Top plots are for $R = 4$ while bottom plots are for $R = 8$.

TABLE 7.3: The improvement matrix for $R = 4$. Each cell contains the number of slices in the test set for witch the configuration labeling the corresponding column performs better than the configuration labeling the corresponding row.

	$\text{Dec}^0\|L^0$	$\text{Dec}^{0\star}\|L^0$	$\text{Dec}^{1\star}\|L^1$	$\text{Dec}^{1\star}\|L^2$	$\text{Dec}^{2\star}\|L^0$
$\text{Dec}^0\|L^0$	·	186	183	186	186
$\text{Dec}^{0\star}\|L^0$	0		179	186	186
$\text{Dec}^{1\star}\|L^1$	3	7		186	186
$\text{Dec}^{1\star}\|L^2$	0	0	0		186
$\text{Dec}^{2\star}\|L^0$	0	0	0	0	

TABLE 7.4: Pearson correlation coefficient between reconstruction error and the measurement mismatch.

	$\text{Dec}^0\|L^0$	$\text{Dec}^{0\star}\|L^0$	$\text{Dec}^{1\star}\|L^1$	$\text{Dec}^{1\star}\|L^2$	$\text{Dec}^{2\star}\|L^0$
	$\rho(\delta\hat{\mathbf{y}}, \delta\hat{\mathbf{x}})$	$\rho(\delta\hat{\mathbf{y}}, \delta\hat{\mathbf{x}}^\star)$	$\rho(\delta\bar{\mathbf{y}}, \delta\hat{\mathbf{x}}^\star)$	$\rho(\delta\bar{\mathbf{y}}, \delta\hat{\mathbf{x}}^\star)$	$\rho(\delta\bar{\mathbf{y}}, \delta\hat{\mathbf{x}}^\star)$
$R = 4$	0.877	0.837	0.138	0.853	0.821
$R = 8$	0.888	0.866	0.198	0.535	0.863

Chapter 8

Incremental Acquisition

Typically, the fast MRI acquisition problem is approached by optimizing the acquisition and reconstruction process over a dataset consisting of multiple volumes, where each volume contains several slices. Generally, the central slices tend to be more intricate and feature-rich, whereas the peripheral slices are simpler and contain fewer details. Thus, applying the same acquisition effort uniformly to all slices within a volume is inefficient. Nevertheless, this is exactly how the conventional acquisition process operates-disregarding individual slice optimization, ultimately limiting adaptability and quality control.

We propose a novel acquisition strategy called *Incremental Adaptive MRI Acquisition*, witch aims to address these limitations by introducing a more flexible, quality-driven approach to MRI acquisition. Instead of using a fixed acquisition pattern, our method divides the acquisition of each slice into batches, where each batch consists of three main steps:

- *Sensing*: Partial acquisition of k-space information.
- *Reconstruction*: Reconstruction of the current approximation of the slice from the acquired k-space data.
- *Quality Estimation*: Evaluation of the reconstructed slice's quality.

Based on the estimated quality, the sequence continues with additional sensing steps or terminates once the desired quality is achieved. This iterative process is illustrated in Fig. 8.1, showcasing the adaptive nature of our approach, witch tailors the acquisition to the content of each slice, ultimately gaining control over reconstruction quality while overcoming the rigid limitations of classic acquisition methods.

We build our experiments on top of the model that in Chapter 7 guaranteed the best performances, i.e., $\text{Dec}^{2\star}$ trained with the loss L^0. To alleviate notation, we will now refer to $\text{Dec}^{2\star}|L^0$ as Dec and to its output as $\hat{\boldsymbol{x}}$.

This work is not the first to propose slice-adaptive k-space acquisition patterns, commonly referred to as *masks* or *binary masks*, in the context of MRI [23, 85, 24]. In [23], the authors introduce slice-specific sampling strategies based on an initial acquisition batch focused on a low-frequency segment of the k-space. Similarly, [85, 24] propose iterative adaptation of k-space acquisition across a predefined number of K iterations, where each iteration is informed by the results of the previous one. Specifically, [85] uses a single unrolled model throughout all iterations, whereas [24] utilizes separate networks for sampling and reconstruction. Unlike these approaches, our method dynamically determines when to

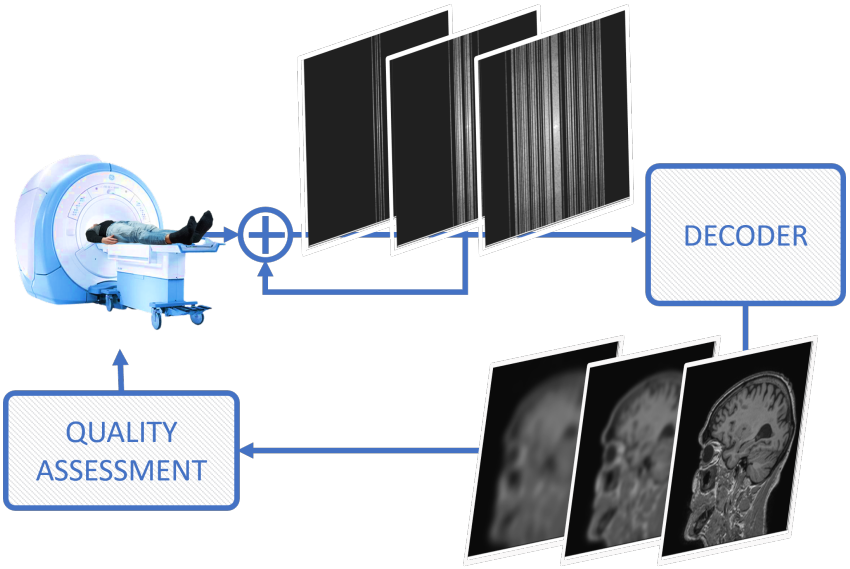


FIGURE 8.1: Incremental Acquisition scheme. At every iteration the MRI machine acquires a batch of samples, that are accumulated with the previously acquired samples in the k-space. A decoder block recreates the anatomical image witch quality is assessed by a specific block. If a target quality is reached the acquisition stops, otherwise another iteration is launched.

terminate the acquisition based on a quality threshold, allowing the number of iterations K to vary, thereby removing the necessity for a predefined K.

Additionally, we are not the first to associate a quality score to MRI reconstructions. In [17], uncertainty is quantified using the Stein's Unbiased Risk Estimator (SURE), witch evaluates the effect of adding Gaussian noise to the encoded image by means of a Monte Carlo simulation. SURE computes second-order statistics of the reconstructed image set to provide an uncertainty measure. While the authors demonstrate that SURE can serve as an indicator of reconstruction uncertainty, they also note that SURE is not necessarily a measure of reconstruction quality. For example, adversarial loss training might yield lower SURE values, suggesting lower reliability, even though it tends to improve the reconstruction quality compared to standard loss training.

In general, implementing a quality estimator falls within the broad field of No-Reference Image Quality Assessment (NR-IQA), whose goal is to estimate the quality of a corrupted image without reference to the original image [54, 55]. However, unlike NR-IQA, we benefit from prior domain-specific knowledge, albeit limited. NR-IQA applied to MRI acquisitions often focuses on developing new metrics to better approximate human perceptual quality [35].

In this work, we utilize well-established metrics such as Mean Absolute Error (MAE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) for quality assessment. Nevertheless, we believe that alternative evaluation metrics, such as those proposed in [65, 17], metrics focused on detecting motion-related artifacts [33], or metrics from pure NR-IQA approaches [81], could be integrated into our framework to further enhance the overall acquisition process.

This study explores unconventional strategies for MRI acquisition, focusing on simplified scenarios-namely, simplified MRI physics-to emphasize conceptual clarity over technical complexity. Our aim is to inspire future advances in MRI acquisition methodologies, rather than presenting a single fully-optimized model.

8.1 Model

In this section, we extend the traditional acquisition process by introducing a novel approach called *Incremental MRI Acquisition*.

For undersampling factors $R > 1$, the autoencoder $\mathcal{A}_{\gamma,\theta}$ inevitably introduces reconstruction errors, resulting in quality degradation measured by a quality function $Q(\boldsymbol{x}, \hat{\boldsymbol{x}})$. Given an input $\boldsymbol{x}$, it is reasonable to assume that there exists a monotonically decreasing relationship between s and $Q(\boldsymbol{x}, \hat{\boldsymbol{x}})$.

One approach is to determine a maximum undersampling factor $\tilde{R}$ such that $Q(\boldsymbol{x}, \hat{\boldsymbol{x}}) \geq q_{\min}$ holds for all images $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X}$ denotes the set of all possible input instances. However, instead of using a fixed value of $\tilde{R}$, it may be advantageous to develop a strategy that dynamically identifies, for each image $\boldsymbol{x}$, the highest s that still satisfies the minimum quality requirement.

To achieve this, we consider a set of undersampling factors $\mathcal{R} = \{R^{(1)}, \ldots, R^{(N)}\}$, with each $R^{(i)}$ corresponding to a different autoencoder $\mathcal{A}^{(i)}_{\gamma,\theta}$ and associated undersampling mask $\boldsymbol{M}^{(i)}$. We then provide a procedure that, for each incoming image, selects the highest undersampling factor that meets the quality constraint.

The task of determining the highest $R \in \mathcal{R}$ that satisfies the quality condition $Q(\boldsymbol{x}, \hat{\boldsymbol{x}}) \geq q_{\min}$ for a given image $\boldsymbol{x}$ is formulated as an optimization problem, with the solution referred to as an *oracle-based* approach.

$$\begin{aligned} R^* = &\operatorname*{argmax}_{R \in \mathcal{R}} R \\ &\text{subject to} \quad Q(\boldsymbol{x}, \hat{\boldsymbol{x}}) \geq q_{\min} \end{aligned} \tag{8.1}$$

The above problem cannot be solved at inference time because $\boldsymbol{x}$ is unknown, hence we rewrite Eq. (8.1) as

$$\begin{aligned} R^* = &\operatorname*{argmax}_{R \in \mathcal{R}} R \\ &\text{subject to} \quad \hat{Q}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \cdots) \geq q_{\min} \end{aligned} \tag{8.2}$$

where $\hat{Q}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \cdots)$, referred to as the *Quality Assessment*, serves as an estimate of the true quality metric $Q(\boldsymbol{x}, \hat{\boldsymbol{x}})$. The function $\hat{Q}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \cdots)$ leverages all available information, such as $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$, to provide an approximation of image quality, potentially incorporating additional inputs like $\overline{\boldsymbol{x}}$ from Eq. (7.4).

Consider the elements of $\mathcal{R}$ arranged in descending order such that $R^{(i)} > R^{(i+1)}$, and construct the sequence of undersampling masks $\{\boldsymbol{M}^{(1)}, \ldots, \boldsymbol{M}^{(N)}\}$ in a manner that respects the incremental constraint $\boldsymbol{M}^{(i)} - \boldsymbol{M}^{(i-1)} \in \{0,1\}^{d \times d}$. This condition implies that any mask

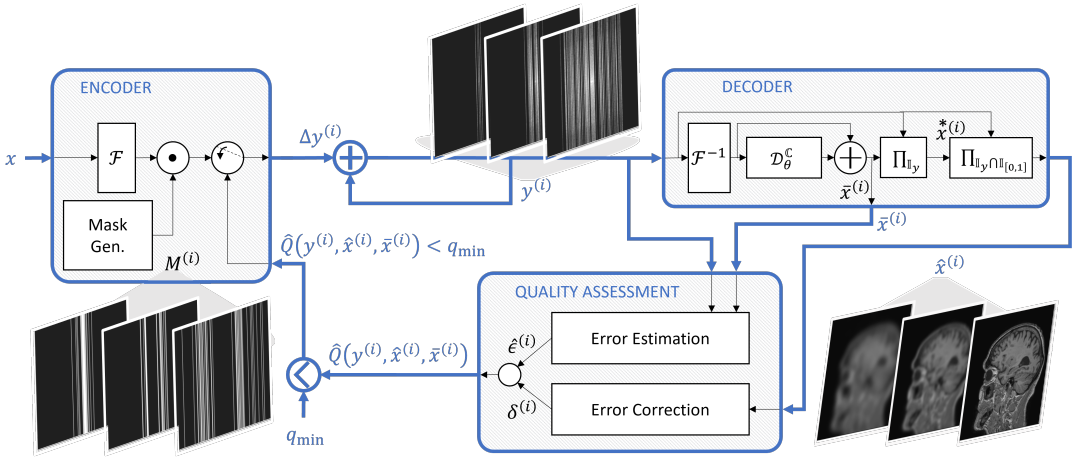


FIGURE 8.2: Incremental Acquisition scheme. At every iteration the MRI machine acquires a batch of samples, that are accumulated with the previously acquired samples in the k-space. A decoder block recreates the anatomical image witch quality is assessed by a specific block. If a target quality is reached the acquisition stops, otherwise another iteration is launched.

$\boldsymbol{M}^{(i)}$ includes all k-space samples selected by the preceding mask $\boldsymbol{M}^{(i-1)}$, along with a batch of additional samples.

Our approach to solving Eq. (8.2) involves an incremental method, where we utilize all autoencoders $\mathcal{A}^{(i)}_{\gamma,\theta}(\boldsymbol{x})$ corresponding to undersampling factors $R^{(i)} \geq R^*$. The proposed algorithm for implementing this procedure is illustrated in Fig. 8.2 and detailed in Algorithm 8.1, where $\mathbf{0}$ represents the null matrix, $\boldsymbol{M}^{(0)} = \mathbf{0}$, and get_sample$(\cdot)$ denotes the MRI sampling operation, modeled by Eq. (4.23) during design, witch takes the undersampling mask as input.

Algorithm 8.1 Incremental Adaptive Acquisition (slice)

Require: q_{min}: a minimum quality level
Require: $\{\boldsymbol{M}^{(1)}, \ldots, \boldsymbol{M}^{(N)}\}$: a sequence of incremental masks with corresponding undersampling ratio $\{R^{(1)}, \ldots, R^{(N)}\}$, with $\boldsymbol{M}^{(0)} = \mathbf{0}$
  $i \leftarrow 1$
  $\hat{q} \leftarrow 0$
  $\boldsymbol{y} \leftarrow \mathbf{0}$
  while $\hat{q} < q_{\text{min}}$ **do**
    $\Delta\boldsymbol{y}^{(i)} \leftarrow$ get_sample$(\boldsymbol{M}^{(i)} - \boldsymbol{M}^{(i-1)})$
    $\boldsymbol{y} \leftarrow \boldsymbol{y} + \Delta\boldsymbol{y}^{(i)}$
    $\hat{\boldsymbol{x}}, \bar{\boldsymbol{x}} \leftarrow \text{Dec}(\boldsymbol{y})$
    $\hat{q} \leftarrow \hat{Q}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \bar{\boldsymbol{x}})$
    $i \leftarrow i + 1$
  end while
  $R^* = R^{(i)}$
  return $R^*, \hat{\boldsymbol{x}}$

Building upon the encoder design presented in the work [48], we introduce an incremental constraint on the sequence of binary masks that define the sequence of autoencoders. The set of incremental masks $\{\boldsymbol{M}^{(1)}, \ldots, \boldsymbol{M}^{(N)}\}$ is constructed by training the corresponding

autoencoders $\{\mathcal{A}^{(1)}_{\gamma,\theta}, \ldots, \mathcal{A}^{(N)}_{\gamma,\theta}\}$ in a decremental fashion[1], beginning from $i = N$ (representing the lowest undersampling ratio s) and proceeding to $i = 1$ (the highest R). Notably, each mask $\boldsymbol{M}^{(i)}$ is derived from $\boldsymbol{M}^{(i+1)}$ through a specialized layer within the encoder that ensures the entries with value one in $\boldsymbol{M}^{(i)}$ are a subset of those in $\boldsymbol{M}^{(i+1)}$.

To make this approach effective, the function $\hat{Q}(\cdots)$ must be capable of accurately predicting the reconstruction quality at each stage of the incremental acquisition process. Inspired by recent advances in No-Reference Image Quality Assessment (NR-IQA) [41, 9], we implement $\hat{Q}(\cdots)$ using a deep neural network. However, we deviate from the conventional NR-IQA approach by utilizing additional information beyond just $\hat{\boldsymbol{x}}$. Our architecture is composed of two primary blocks: the *Error Estimation* block and the *Error Correction* block. These two blocks produce scalar outputs that are subsequently combined to estimate the final image quality. A schematic overview of the incremental acquisition, including the neural quality predictor, is depicted in Fig. 8.2.

The first block takes as input the error $\epsilon = \|\mathrm{Enc}(\overline{\boldsymbol{x}}) - \boldsymbol{y}\|_1$, where $\|\cdot\|_1$ represents the standard ℓ_1 norm, and uses it within a quadratic function to generate an output $\hat{\epsilon} = a\epsilon^2 + b\epsilon + c$. As previously demonstrated [50, 48], this estimated value $\hat{\epsilon}$ serves as an initial, albeit rough, proxy for the reconstruction quality. The *Error Correction* block then extracts additional features:

$$\gamma = \mathrm{gpool}\left(\mathbf{B}_5\left(\mathbf{B}_4\left(\mathbf{B}_3\left(\mathbf{B}_2\left(\mathbf{B}_1\left(\mathbf{B}_0\left(\hat{\boldsymbol{x}}\right)\right)\right)\right)\right)\right)\right) \tag{8.3}$$

where gpool denotes a global average pooling operation that outputs a scalar feature $\delta \in \mathbb{R}$, and each block $\mathbf{B}_j$ performs the following:

$$\mathbf{B}_j(\cdot) = \mathrm{pool}\left(\mathrm{BN}_{\beta_j}\left(\mathrm{ReLu}(\mathrm{Conv}_{\alpha_j}(\cdot))\right)\right) \tag{8.4}$$

where j represents the block index; pool indicates the average pooling operation with a window size of 2×2; BN_β represents the batch normalization process with parameter β; ReLU refers to the Rectified Linear Unit activation function; and Conv_α denotes the convolution operation with a 5×5 kernel, parameterized by α. The number of filters in each convolution layer Conv_{α_j} is set to $[10, 8, 6, 4, 2, 1]$.

Subsequently, the outputs from the two branches are concatenated and passed to a fully connected Dense layer, parameterized by $\omega_{\hat{\epsilon}}$ and ω_γ, witch yields the estimated quality:

$$\hat{\boldsymbol{Q}}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \bar{\boldsymbol{x}}) = \omega_{\hat{\epsilon}}\hat{\boldsymbol{\epsilon}} + \omega_\gamma \boldsymbol{\gamma} \tag{8.5}$$

All parameters, namely a, b, c, α_i, β_i, $\omega_{\hat{\epsilon}}$, and ω_γ, are trainable and are optimized during the training process by minimizing the following loss function:

$$L_Q(a, b, c, \alpha, \beta, \omega_{\hat{\epsilon}}, \omega_\gamma) = \mathbb{E}_{\boldsymbol{x}}\left[\left\|Q(\boldsymbol{x}, \hat{\boldsymbol{x}}) - \hat{Q}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \overline{\boldsymbol{x}})\right\|_1\right] \tag{8.6}$$

[1] Empirical evidence indicates that training in a decremental order yields better performance compared to training in incremental order.

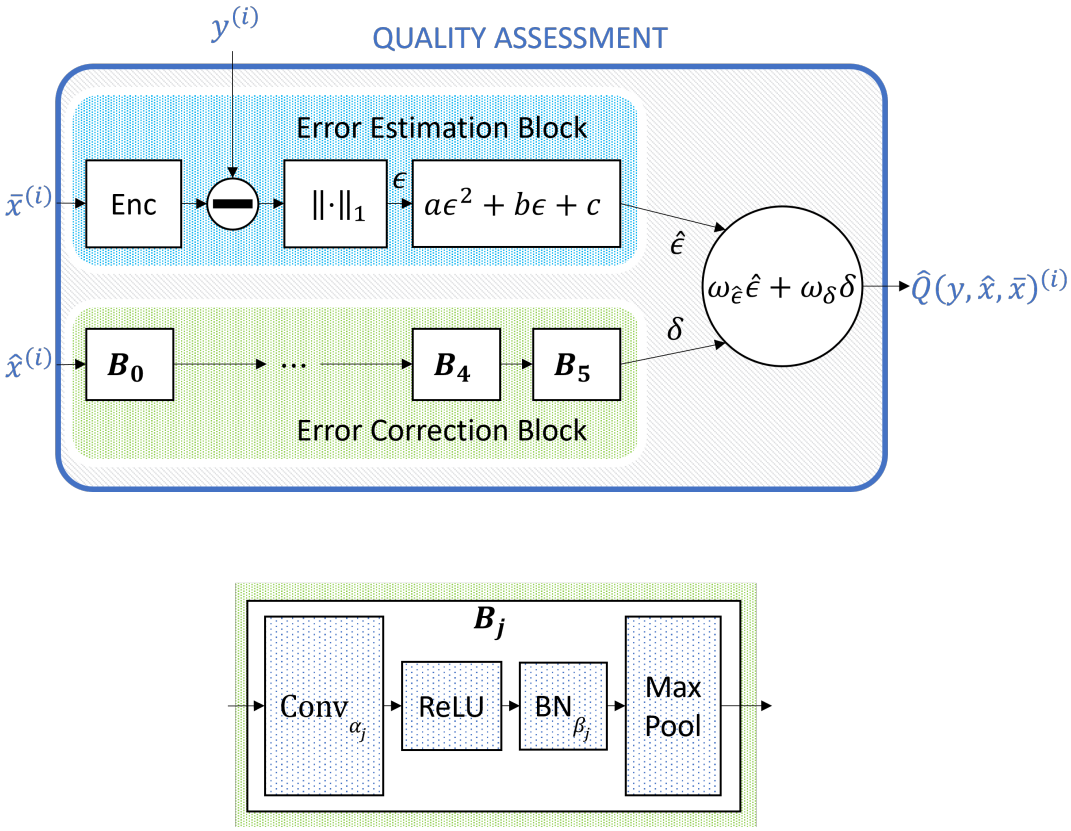


FIGURE 8.3: Incremental Acquisition scheme. At every iteration the MRI machine acquires a batch of samples, that are accumulated with the previously acquired samples in the k-space. A decoder block recreates the anatomical image witch quality is assessed by a specific block. If a target quality is reached the acquisition stops, otherwise another iteration is launched.

Since z, and consequently $\hat{x}$ and $\bar{x}$, depend on the undersampling mask $\boldsymbol{M}$, a separate quality estimator $\hat{Q}$ is required for each mask $\boldsymbol{M}^{(i)}$ in our approach.

A detailed visual depiction of the quality assessment block is provided in Fig. 8.3.

8.2 Numerical Evidence

In this section, we present the results of our incremental MRI undersampling approach, witch dynamically adjusts the k-space acquisition based on the estimated reconstruction quality. The proposed method is evaluated against traditional fixed-pattern undersampling, showcasing significant improvements in efficiency by reducing the number of acquired samples while maintaining or exceeding the desired image quality. The results demonstrate the adaptability and effectiveness of the incremental approach in optimizing the MRI acquisition process, ensuring a balance between acquisition time and reconstruction quality.

8.2.1 Dataset

To evaluate our proposed approach, we utilize two publicly available datasets: the NYU fastMRI dataset [28] and the IXI dataset [10]. The fastMRI dataset contains fully-sampled

knee scans, while the IXI dataset provides fully-sampled brain scans. In accordance with the workflow presented in Fig. 8.2, we use the reconstructed images provided by these datasets rather than the raw k-space data. We generate k-space by applying the Fourier transform to each image from all volumes for both datasets considered in this study.

In our experiments, we represent the k-space as a two-dimensional Cartesian grid and evaluate our method using two different sampling strategies: *i*) Line-Constrained, witch follows straight-line sampling paths, aligning with typical 2D imaging approaches, such as Spin Echo sequences where the frequency-encoding dimension is represented on the x-axis and the phase-encoding direction on the y-axis of the k-space; *ii*) Unconstrained, witch provides more flexibility and enables an exploratory analysis of the model's potential capabilities. Although this unconstrained approach could theoretically be extended to 3D imaging (using one frequency encoding dimension and two phase encoding dimensions), its practical deployment would require additional investigation.

By combining the two datasets with the two sampling strategies, we derive four distinct scenarios, named as follows: *i*) `FastMRI-U`; *ii*) `FastMRI-C`; *iii*) `IXI-U`; *iv*) `IXI-C`, where `U` and `C` denote Unconstrained and Line-Constrained sampling strategies, respectively. We use the 320×320 single-coil proton density-weighted images from the FastMRI dataset, and similarly, we work with 256×256 single-coil T1-weighted images from the IXI dataset[2].

Both datasets are naturally organized into *volumes*, each containing multiple grayscale slices. To address the inherent variability in intensity across volumes acquired independently, each volume is normalized by dividing its pixel values by the maximum intensity within that volume. This ensures consistent contrast across different patients without causing distortions. The FastMRI dataset is divided into training, validation, and test sets, comprising 50, 5, and 5 volumes, respectively, with each volume containing between 34 to 45 slices. The IXI dataset is similarly split into training, validation, and test sets consisting of 166, 11, and 8 volumes, respectively, with each volume containing 150 slices.

For the `IXI-U` and `IXI-C` experiments, the data split consists of 166, 11, and 8 volumes, and 166, 12, and 7 volumes for training, validation, and test, respectively. An additional validation volume was used for `IXI-C` to estimate and compensate for an unexpected bias in the quality assessment.

Within each volume, initial and terminal slices exhibit fewer anatomical details, with higher concentrations of information towards the center of the volume. Fig. 8.4 illustrates slices 5, 15, 25, and 35 (for FastMRI) and slices 30, 60, 90, and 130 (for IXI), confirming this pattern.

8.2.2 Training Parameters

For training, we utilize the Adam optimizer with an initial learning rate of 0.01, combined with early stopping and a learning rate reduction on plateau to ensure stable convergence. During testing, binary masks are drawn based on the generated probabilities from training,

[2]While this study focuses on single-coil data, our proposed acquisition strategy is extendable to multi-coil scenarios, witch will be addressed in future work.

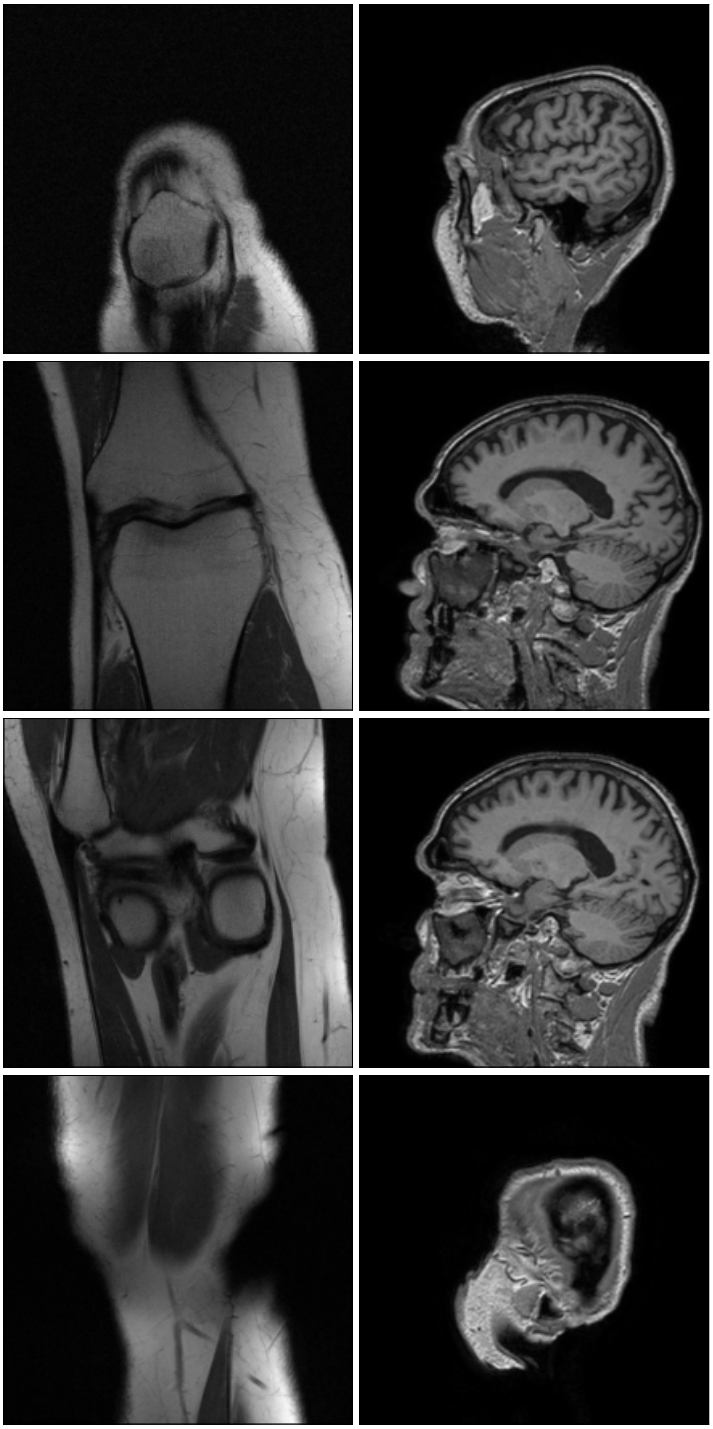


FIGURE 8.4: Some slices from the first volume of both FastMRI and IXI datasets (first and second columns respectively). Slice index for FastMRI are, in order $[5, 15, 25, 35]$, while for IXI are $[30, 60, 90, 130]$. These sequences show how a volume is made by images of different information content. For improved visualization, all images have been linearly rescaled to enhance visual contrast.

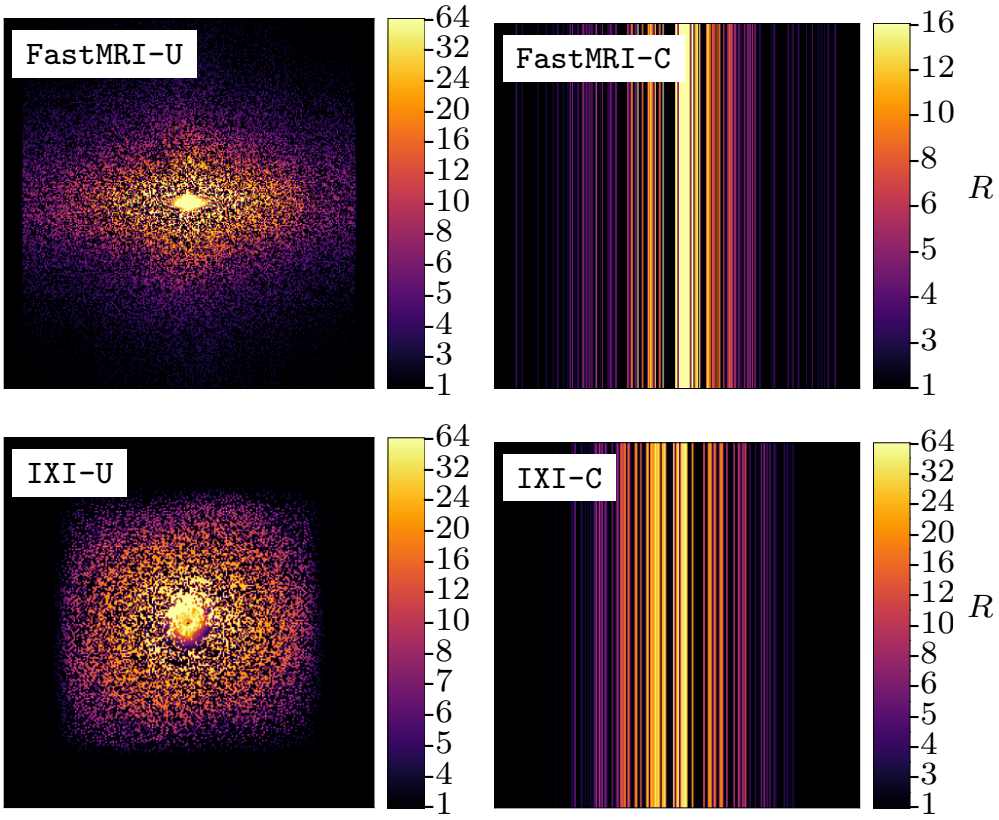


FIGURE 8.5: Sampling pattern for the four use-cases. Each pixel of the image is a sample of the k-space, that is painted proportionally with the undersampling ratio R at witch it is acquired. The higher the associated undersampling ratio (the sooner it is acquired), the brighter the pixel.

where each mask element (pixel or line) represents the probability of being non-zero. Once a mask is drawn, it remains fixed throughout the evaluation phase.

The autoencoder model consists of approximately 31.5 million trainable parameters. Each of the N self-assessment models, corresponding to different undersampling ratios $R \in \mathcal{R}$, contains roughly 5.5 thousand parameters. For each model $\mathcal{A}^{(i)}_{\gamma,\boldsymbol{\theta}}$, we perform training across l different random weight initializations and subsequently select the model with the highest average reconstruction quality. The value of l is set to 5 for the FastMRI dataset and 2 for the IXI dataset.

$$\mathcal{R} = \{64, 32, 24, 20, 16, 12, 10, 8, 6, 5, 4, 3\}.$$

For `IXI-U` we consider

$$\mathcal{R} = \{64, 32, 24, 20, 16, 12, 10, 9, 8, 7.5, 7, 6.75, 6.5, 6.25, 6, 5.75, 5.5, 5.25, 5, 4.75, 4.5, 4.25, 4, 3\}.$$

while `IXI-C` refers to

$$\mathcal{R} = \{64, 32, 24, 20, 16, 12, 10, 9, 8, 7, 6.5, 6, 5.75, 5.5, 5.25, 5, 4.75, 4.5, 4.25, 4, 3.75, 3.5, 3.25, 3\}.$$

For the `FastMRI-U` and `FastMRI-C` scenarios, we use a subset of undersampling ratios $\mathcal{R}$[3]. In a real-world application, the parameter $q_{\min}$, witch represents the minimum acceptable reconstruction quality, should be defined by domain experts. For experimental purposes, we adjusted $\mathcal{R}$ to achieve competitive performance within the range $q_{\min} \in [38, 44]$. To extend or modify this range, it is sufficient to adjust the set $\mathcal{R}$.

8.2.3 Evaluation Metrics

To assess the quality of each reconstruction we adopt three different metrics:

- Peak Signal-to-Noise Ratio:

$$\mathrm{PSNR}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = 10 \log_{10} \left(\frac{\max(\boldsymbol{x})^2}{\mathrm{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}})} \right) \tag{8.7}$$

  where $\mathrm{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = 1/n \sum_{j=0}^{n-1} ([\boldsymbol{x}]_j - [\hat{\boldsymbol{x}}]_j)^2$ is the Mean Square Error between $\boldsymbol{x}$ and the decoder output and $[\boldsymbol{x}]_j$ refers to the j-th pixel of the flattened matrix. Notice that, $\max(\boldsymbol{x}) = 1$.

- Inverse dB Mean Absolute Error:

$$\mathrm{MAE}^{-1}_{\mathrm{dB}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = 20 \log_{10} \left(\frac{\max(\boldsymbol{x})}{\mathrm{MAE}(\boldsymbol{x}, \hat{\boldsymbol{x}})} \right) \tag{8.8}$$

[3]For `FastMRI-C`, the final evaluation set is reduced to $\mathcal{R} = \{16, 12, 10, 8, 6, 5, 4, 3\}$ due to failure in convergence of the quality estimator for the remaining four undersampling ratios.

Where $\mathrm{MAE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = {}^1/_n \sum_{j=0}^{n-1} |[\boldsymbol{x}]_j - [\hat{\boldsymbol{x}}]_j|$ is the Mean Absolute Error between $\boldsymbol{x}$ and the decoder output.

- Structural Similarity Index:

$$\mathrm{SSIM}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{(2\mu_x\mu_{\hat{x}} + c_1)(2\sigma_{x\hat{x}} + c_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)} \tag{8.9}$$

where μ_x and $\mu_{\hat{x}}$ denote the *local* mean values for the original and reconstructed images, respectively; σ_x and $\sigma_{\hat{x}}$ represent the *local* variances; and $\sigma_{x\hat{x}}$ is the *local* covariance between the decoder output and the ground-truth. The term *local* implies that these statistics are calculated over sliding windows of size 11×11, where each pixel is weighted by a Gaussian kernel with a standard deviation of 1.5 pixels. In the above formula, $c_1 = (k_1 D)^2$ and $c_2 = (k_2 D)^2$, with D being the dynamic range of pixel values, and constants $k_1 = 0.01$ and $k_2 = 0.03$ as recommended in [80].

8.2.4 Introduction to Experiments

To validate the effectiveness of our proposed approach, we conducted a series of experiments aimed at evaluating different facets of the acquisition and reconstruction process. The experiments are centered around three main aspects:

- **Feasibility:** We present all sampling patterns (acquisition masks) in a single figure, particularly highlighting those corresponding to higher undersampling ratios (s). Additionally, we provide examples of actual reconstructed images to demonstrate the efficacy visually.
- **Quantitative Advantage Analysis:** We illustrate the performance gains in terms of acceleration achieved by our approach for different target reconstruction quality levels. The improvements are analyzed both in terms of reconstruction quality and undersampling ratios for a fixed acquisition objective.
- **Quality Estimation Performance:** We evaluate the accuracy of the quality estimator and assess the reliability of the chosen metrics for image quality estimation.

In this section, we compare three distinct approaches to MRI acquisition:

- **Oracle-Based Approach:** This approach assumes perfect knowledge of the target quality of the reconstructed image and selects the optimal undersampling ratio R to meet the given quality requirement. The values of R are derived based on the solution of Eq. (8.1). This method serves as an ideal upper bound for comparison.
- **Incremental Approach:** In this strategy, the undersampling ratio is progressively increased based on real-time quality estimates, as described in Algorithm 8.1. This reflects a more practical and adaptive acquisition process that adjusts according to the specific characteristics of each acquired slice.

- **Conventional Acquisition Approach:** This baseline approach utilizes a fixed undersampling mask with a preselected value $\tilde{R}$. Here, the undersampling ratio remains constant throughout the acquisition, serving as a reference point for evaluation.

To ensure a consistent comparison across all methods, we define a minimum quality threshold, q_{min}, representing the lowest acceptable reconstruction quality achieved across the entire test set. The oracle-based approach provides the benchmark for optimal performance, while the incremental approach offers a practical, adaptable solution that dynamically modifies the acquisition process to meet quality requirements.

8.2.5 Results

In this section, we present the experimental results comparing the performance of the proposed methods across various acquisition strategies. The results are analyzed in terms of undersampling ratios, reconstruction quality, and the accuracy of quality estimation. Below, we summarize the main findings of our experiments.

Fig. 8.5 visualizes the association between each frequency in inthe k-space and its corresponding undersampling ratio R for all four use cases. Frequencies corresponding to higher values of R are prioritized and acquired before those associated with lower values of R, consistently giving precedence to central frequencies. This observation underlines the model's adaptability to different types of signals, as evident from the noticeable structural differences across use cases.

Results for the undersampling ratio R as a function of the minimum quality threshold q_{min} $\left[\text{MAE}_{\text{dB}}^{-1}\right]$ are depicted in Fig. 8.6. Across all approaches, q_{min} displays a monotonic relationship with s. The reported undersampling ratio values are computed as the reciprocal of the average acquisition rates r over the entire test set: $\overline{R} = \left(\frac{1}{L}\sum_{j=0}^{L-1} r_j^*\right)^{-1}$, where L represents the size of the test set and $r^* = 1/R^*$.

The results clearly demonstrate that for all four use cases, the incremental acquisition method consistently outperforms others in terms of the undersampling ratio for a fixed q_{min}. The gap between the incremental and oracle-based curves provides an indication of the performance of the quality assessment block. Due to inevitable errors introduced by $\hat{Q}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \overline{\boldsymbol{x}})$, the incremental acquisition curve tends to fall below the oracle-based curve.

Table 8.1 compares the average undersampling ratio $\overline{R}$ of the incremental approach to the fixed undersampling ratio $\tilde{R}$ of the classic approach for all use cases, with reference to specific q_{min} values. These results offer a more detailed understanding of the improvement achieved by the incremental acquisition method. As expected, the improvements in the Line-Constrained cases are less pronounced compared to the Unconstrained cases, witch can be attributed to the limited degrees of freedom imposed by the Line-Constrained sampling trajectories.

Fig. 8.7 illustrates the reconstruction quality Q for the acquisition of 5 volumes from the FastMRI and IXI test sets, with a fixed target quality of $q_{\text{min}} = 40$ dB. The x-axis represents the indices of each slice across the 5 volumes, while the y-axis indicates the reconstruction quality in terms of $\text{MAE}_{\text{dB}}^{-1}$. The periodic pattern observed reflects the anatomical structure

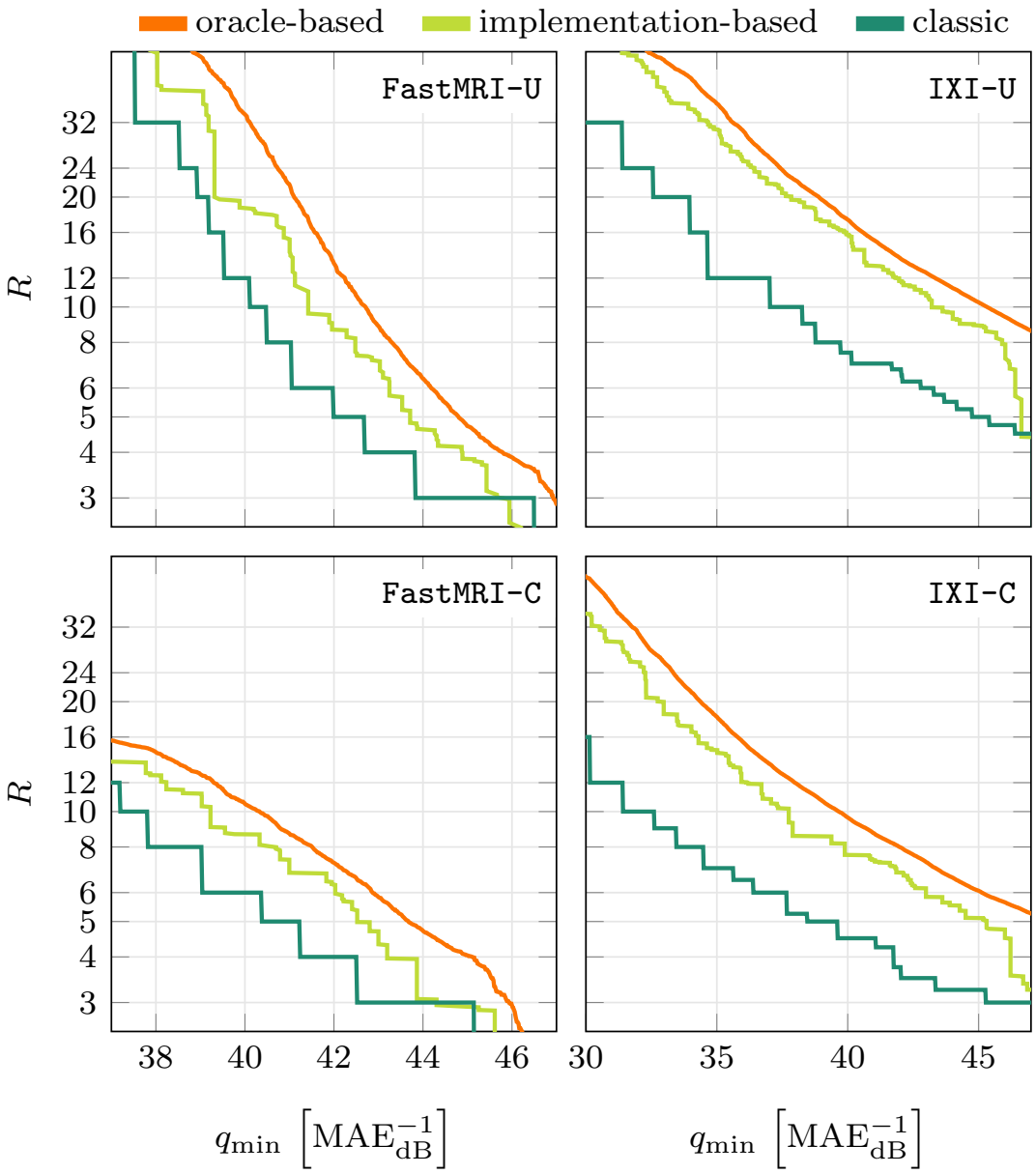


FIGURE 8.6: $\overline{R}$ (average R over the test set) for incremental acquisition and $\tilde{R}$ for the classic acquisition, are plotted for many different $q_{\min}$. $\overline{R}$ is displayed for both the oracle-based and incremental acquisition; and for the four selected use cases.

of the volumes, as shown in Fig. 8.4. As we move through the slices within a volume, they transition from less detailed regions that are easier to reconstruct (i.e., higher PSNR) to more complex regions (i.e., lower PSNR), and then back to simpler regions. When concatenating 5 volumes, this results in pseudo-periodic variations in PSNR. This effect is more pronounced with the classic acquisition approach, witch maintains a fixed undersampling ratio across all slices. In contrast, the incremental acquisition curve adjusts R according to the complexity of each slice, resulting in PSNR values that more closely match the desired target $q_{\min}$.

The difference between the classic and incremental acquisition curves is more evident at the volume boundaries than at the center. This indicates the incremental acquisition's ability

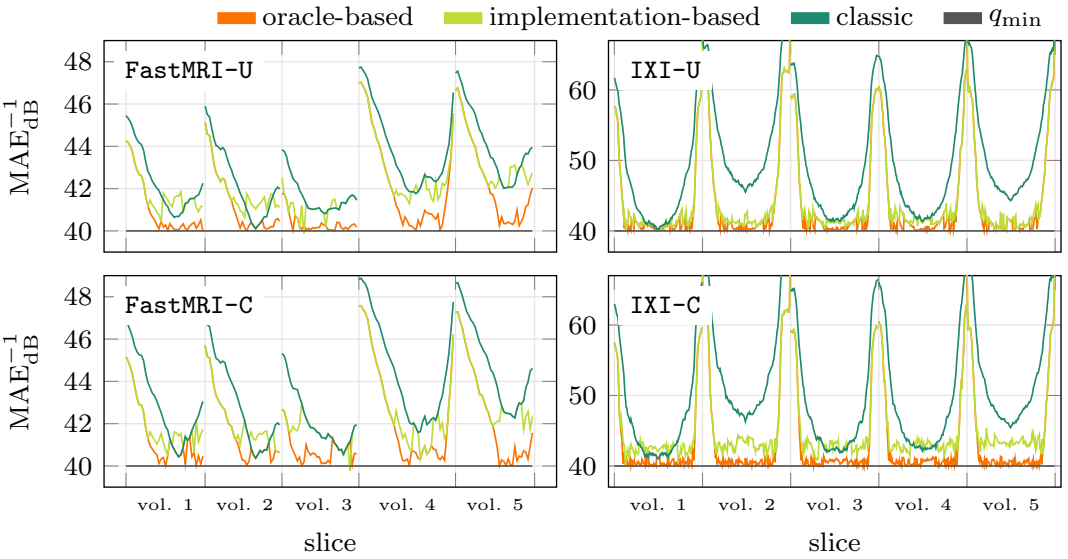


FIGURE 8.7: Quality of the slices with classic, incremental, and oracle-based incremental acquisitions, given a target quality $q_{\min} = 40\mathrm{MAE}_{\mathrm{dB}}^{-1}$ (solid horizontal line) for all use cases FastMRI-U, FastMRI-C, IXI-U, IXI-C. FastMRI displays all the volumes of the test set, while IXI displays a subset of five volumes.

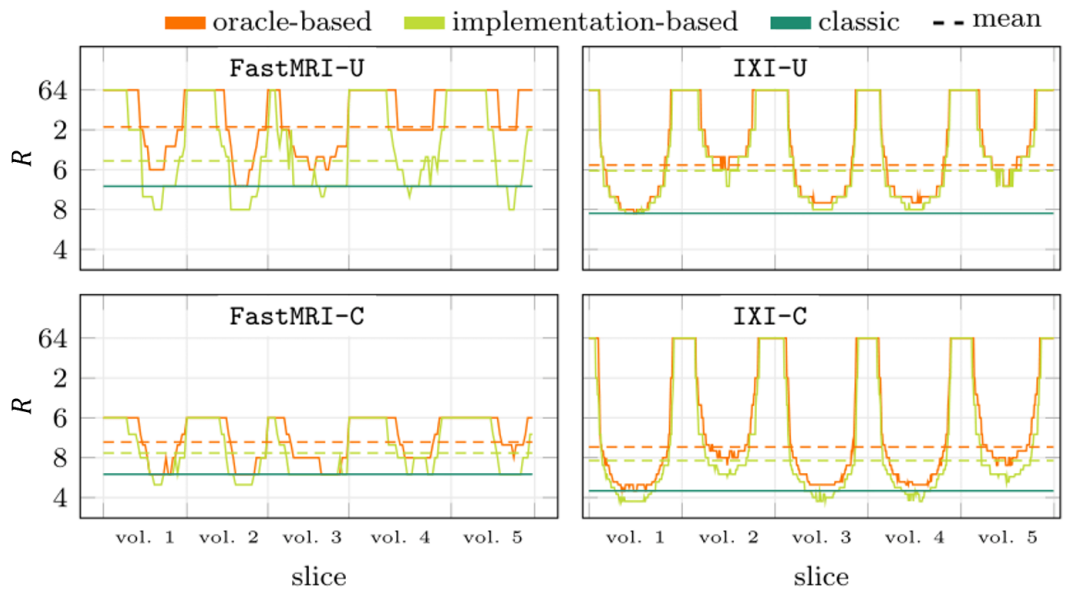


FIGURE 8.8: Undersampling ratio R of the slices with classic, incremental and oracle-based incremental acquisitions, given a target quality $q_{\min} = 40\mathrm{MAE}_{\mathrm{dB}}^{-1}$ for all use cases FastMRI-U, FastMRI-C, IXI-U, IXI-C. The average R over the whole test set is reported with dashed lines. FastMRI displays all the volumes of the test set, while IXI displays a subset of five volumes.

to adapt to the information content of the slices-specifically, adopting a higher undersampling ratio R when the slice contains less information (typically at the extremities of the volume). Due to the estimation error in assessing the reconstruction quality, the incremental approach tends to over-sample the central slices, deviating from the optimal oracle-based trend. In the oracle-based approach, the quality is consistently maintained just above the target threshold.

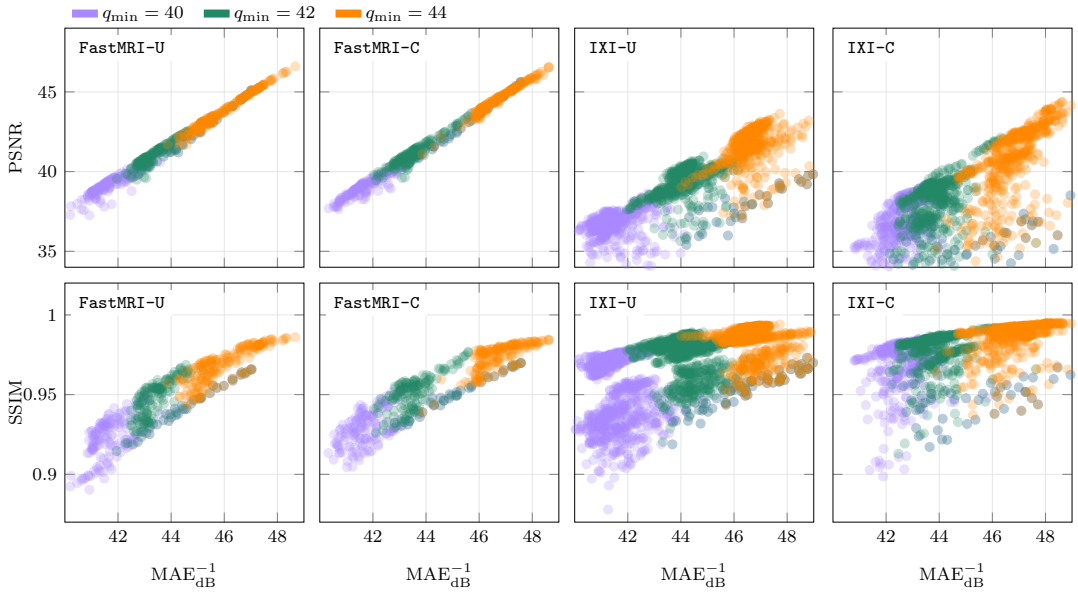


FIGURE 8.9: PSNR and SSIM as a function of $\mathrm{MAE}_{\mathrm{dB}}^{-1}$, for every sample of the test set, for incremental acquisition, for three different target qualities.

TABLE 8.1: $\overline{R}$ for incremental and classic acquisitions, for different $q_{\min}$, for all the use cases.

$q_{\min}$ $\left[\mathrm{MAE}_{\mathrm{dB}}^{-1}\right]$	$\overline{R}$: incremental / classic			
	FastMRI-U	FastMRI-C	IXI-U	IXI-C
38	49.4 / 32.0	12.6 / 8.0	20.1 / 10.0	9.3 / 5.25
39	39.0 / 20.0	11.2 / 8.0	17.4 / 8.00	8.6 / 5.00
40	19.6 / 12.0	8.7 / 6.0	15.8 / 7.50	8.2 / 4.50
41	15.4 / 8.0	7.4 / 5.0	13.0 / 7.00	7.4 / 4.50
42	9.1 / 5.0	6.3 / 4.0	12.0 / 6.75	6.8 / 3.75
43	7.1 / 4.0	4.7 / 3.0	10.9 / 6.00	6.2 / 3.50
44	4.7 / 3.0	3.9 / 3.0	9.7 / 5.50	5.6 / 3.25

To further emphasize this point, Fig. 8.8 provides a visual comparison of the resulting undersampling ratios R for the three acquisition approaches with a target quality of $q_{\min} = 40$ dB. In the classic acquisition strategy, the undersampling ratio remains constant, represented by the horizontal line ($\tilde{R}$), whereas the curves for the adapted approaches exhibit a pronounced step-like behavior. The incremental acquisition curve consistently stays below the oracle-based curve, and only occasionally falls below the classic acquisition line. This observation once again underscores the adaptability and superiority of the incremental acquisition method. Additionally, the figure includes dashed lines that indicate the average undersampling ratios across the entire test set for both adaptive approaches.

The results are presented in terms of $\mathrm{MAE}_{\mathrm{dB}}^{-1}$, as MAE serves as the loss function used during the training of $\mathcal{A}_{\gamma,\theta}$, in line with [48]. Furthermore, we explore the relationship between $\mathrm{MAE}_{\mathrm{dB}}^{-1}$ and the other two evaluation metrics.

The scatter plots in Fig. 8.9 illustrate, for every slice in the test set, the relationship between reconstruction quality in terms of $\mathrm{MAE}_{\mathrm{dB}}^{-1}$ and the corresponding quality in terms of

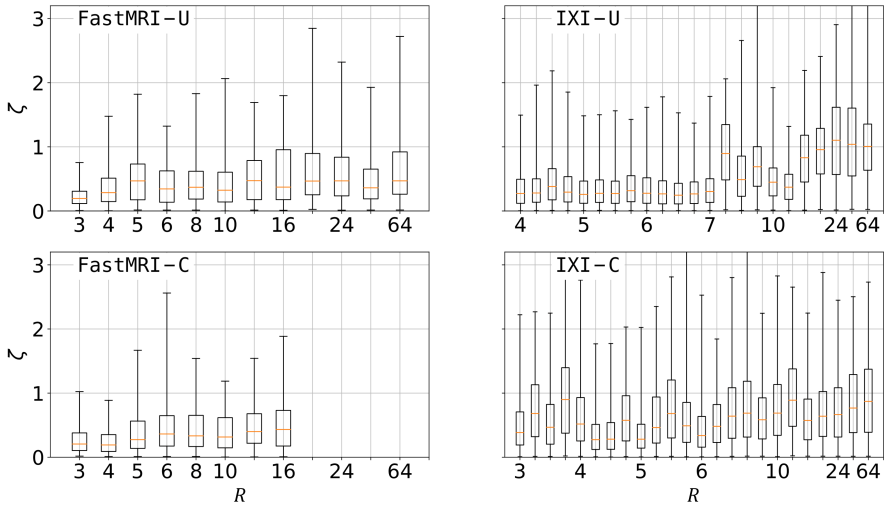


FIGURE 8.10: Estimation error in terms of ζ or all the implemented quality estimators. The central line within each box represents the median, while the edges of the box correspond to the 1st and 3rd quartiles. Whiskers extend to the 1st and 99th percentiles.

PSNR (top row) and SSIM (bottom row). The comparison is made for images acquired with three different target qualities, each represented by a distinct color.

Interestingly, the results confirm that setting a target $q_{\min}$ in terms of $\text{MAE}_{\text{dB}}^{-1}$ influences the reconstruction quality of slices in terms of both PSNR and SSIM. Moreover, in the case of FastMRI, a strong correlation can be observed between $\text{MAE}_{\text{dB}}^{-1}$ and the other two metrics.

Lastly, we evaluate the performance of the quality assessment model $\hat{Q}(\boldsymbol{y}, \hat{\boldsymbol{x}}, \overline{\boldsymbol{x}})$ in estimating $\text{MAE}_{\text{dB}}^{-1}$. As a figure of merit, we compute the absolute error

$$\zeta = \frac{1}{L} \sum_{j=0}^{L-1} |Q(\boldsymbol{x}_j, \hat{\boldsymbol{x}}_j) - \hat{Q}(\boldsymbol{y}_j, \hat{\boldsymbol{x}}_j, \overline{\boldsymbol{x}}_j)| \tag{8.10}$$

. The results for all the quality assessment models are presented in Fig. 8.10. The plots reveal a general trend of increasing difficulty in quality estimation as the undersampling ratio R grows.

The encoder-decoder network processes images with an average runtime of 4.3 ms for FastMRI and 2.7 ms for IXI, using an NVIDIA A100 GPU. The quality estimation step adds 0.78 ms for FastMRI and 0.36 ms for IXI, resulting in total processing times of 5.1 ms and 3.1 ms, respectively. In terms of computational requirements, the quality estimator requires approximately 85 M MAC operations for FastMRI images and 54 M MAC operations for IXI images.

Before concluding, Fig. 8.11 provides a visual comparison of slices acquired with both the classic and incremental acquisition approaches, where the target quality is set to $q_{\min} =$ 43 dB. Each image also displays (in red) the corresponding undersampling ratio R and the actual reconstruction quality. As expected, the slices acquired with the classic method share the

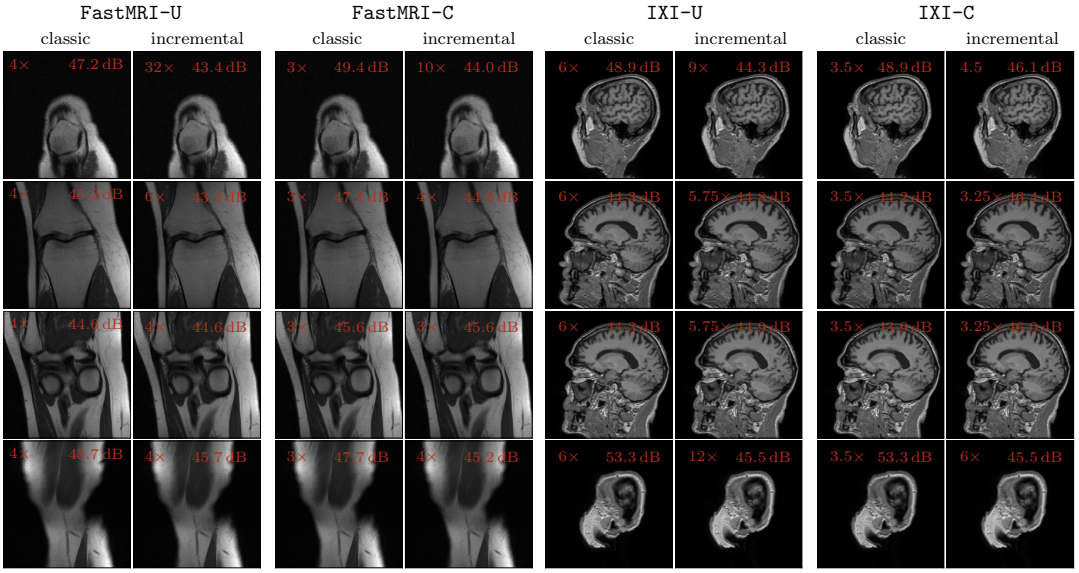


FIGURE 8.11: Classically and incrementally acquired test slices (indexes $[5, 15, 25, 35]$) for the FastMRI dataset, and $[30, 60, 90, 130]$ for the IXI dataset. In red, are both the acquisition undersampling ratio s and the real reconstruction quality $Q(\boldsymbol{x}, \hat{\boldsymbol{x}})$[dB]. For improved visualization, all images have been linearly rescaled to enhance visual contrast.

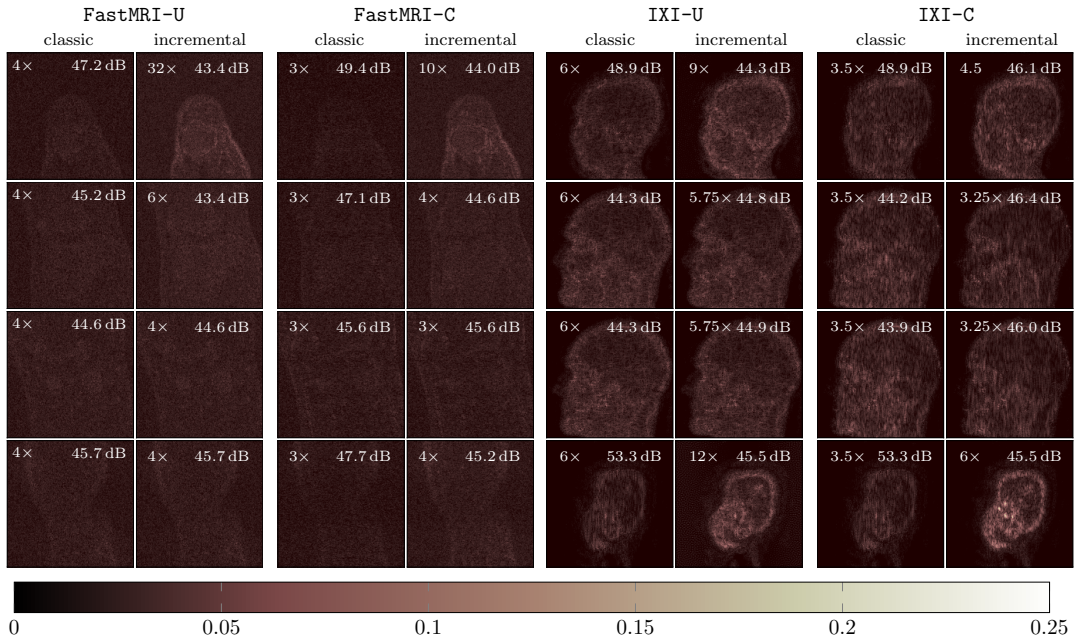


FIGURE 8.12: Absolute difference between classically and incrementally acquired slices w.r.t. the ground truth slices. Slices from the test set (indexes $[5, 15, 25, 35]$) for the FastMRI dataset, and $[30, 60, 90, 130]$ for the IXI dataset. In red, the acquisition undersampling ratio R and the real reconstruction quality $Q(\boldsymbol{x}, \hat{\boldsymbol{x}})$[dB]. Images have been linearly rescaled to enhance visual contrast.

same fixed value of R. In contrast, the undersampling ratio R in the incrementally-acquired slices varies depending on the information content of the slice. It can be observed that, when a sufficiently high $q_{\min}$ is chosen, incremental acquisition enables faster scanning with negligible degradation in quality. Indeed, as shown in the figure, even when the reconstruction quality varies by more than 3 dB, no noticeable degradation can be observed, and similar behavior is seen across the rest of the dataset. Fig. 8.12 further supports this analysis. In this figure, we visualize the absolute differences between the reconstructed images and the ground truth, with a color scale ranging from black to pink to white to highlight areas of minimal to maximal differences, respectively. The results confirm that incremental acquisition introduces no significant perceptual degradation, with differences remaining minimal even in cases where reconstruction quality deviates by over 3 dB.

The proposed method achieves a total processing time of 5.1 ms for knee slices of size 320×320 and 3.1 ms for brain slices of size 256×256. These runtimes encompass both the reconstruction and quality assessment stages, and are significantly faster than the typical Echo Times (TE) for standard MRI sequences, witch range from 22 ms to 65 ms for Spin Echo sequence protocols used in knee imaging [25]. These results indicate that our method has the potential to be integrated into real-time MRI acquisition workflows, enabling on-the-fly adjustments of the undersampling ratio.

8.3 Trade-off Summary

The incremental adaptive MRI acquisition method optimizes k-space sampling by dynamically adjusting undersampling based on slice complexity, ensuring efficient data acquisition while maintaining a predefined quality threshold. This adaptability enhances acceleration without compromising image fidelity, making it a promising alternative to fixed undersampling techniques.

A major strength of this approach is real-time quality estimation, allowing adjustments during acquisition. However, relying on a single metric may not fully capture image quality. While preliminary results suggest generalizability across different metrics, a more comprehensive evaluation is needed. Additionally, quality estimation errors may lead to overestimated quality, potentially impacting diagnosis. Setting a higher quality threshold could mitigate this but at the cost of reduced overall acceleration.

The method also introduces flexibility, adapting undersampling to individual patients rather than applying a fixed pattern. This personalization enhances imaging efficiency and could be particularly beneficial for complex anatomies. Furthermore, it achieves low processing times (a few milliseconds per slice) making real-time implementation feasible within standard MRI protocols.

However, its validation has been limited to unconstrained undersampling and Cartesian sampling, raising concerns about applicability to structured sampling patterns commonly used in clinical settings. Future work should explore its performance across broader acquisition strategies to ensure integration into real-world MRI workflows.

8.4 Conclusion

The method presented in this work is based on the LOUPE architecture for subsampled MRI acquisition. We show that the reconstruction quality can be significantly improved by embedding a measurement constraint via a regularization term, resulting in an average PSNR increase of up to 1.6 dB for $R = 4$ and $R = 8$. Moreover, we demonstrate that the error between re-acquisitions and actual acquisitions serves as an effective proxy for estimating reconstruction quality against the unknown ground truth. This insight has led to the development of two self-assessment tools: one to determine if a reconstructed scan meets a quality standard at inference time, and another to estimate the image reconstruction error. These tools enhance the robustness of the reconstruction process, enabling reliable quality predictions.

We also present a callback-based novel method that simplifies the hyper-parameters tuning in a deep neural network, transforming a typically challenging task into a straightforward process. Specifically, we demonstrate how this approach can be applied within the LOUPE framework, witch involves a mask generator whose output is critically dependent on a hyper-parameter. Our results indicate that using this novel approach achieves competitive performance without requiring extensive parameter tuning.

Building on LOUPE, a state-of-the-art deep neural network capable of optimizing both undersampling patterns and image recovery [2], we demonstrate that minor architectural changes can further enhance reconstruction performance. In particular, we introduce three methods to incorporate measurement constraints, each resulting in improved reconstruction quality. Additionally, we solve an optimization problem to identify the best reconstructed image within the image domain and its set of counterimages. Our experiments show average PSNR improvements of 2.1 dB and 1.7 dB for acceleration rates of 4 and 8, respectively, compared to the unoptimized version of LOUPE. We also establish a link between the reconstruction error over the entire image and the measurement-only reconstruction error, suggesting that it is possible to estimate the former by examining the latter. We plan to further explore this direction in future work.

Finally, we present a novel acquisition method capable of dynamically adapting the sampling rate during inference acquisition. The acquisition process is divided into N temporally ordered batches, with each batch acquiring a portion of the k-space. After each batch, a reconstructed slice is computed, and its quality is estimated using a neural predictor. As additional batches are processed, the reconstruction quality improves monotonically. The acquisition stops once a target quality is reached, often requiring fewer than N iterations. We validate this method using a state-of-the-art deep neural encoder-decoder [48], demonstrating that our approach achieves higher acceleration than standard acquisition methods. The tests, conducted on the FastMRI and IXI datasets using both Unconstrained and Line-Constrained sampling strategies, highlight the effectiveness of the proposed approach.

Since our acquisition method relies on a quality estimator, we provide a detailed description of the adopted architecture. Our experiments demonstrate that the proposed neural quality estimator achieves performance comparable to an oracle-based solution, proving its viability for practical applications.

Bibliography

[1] Hemant Kumar Aggarwal and Mathews Jacob. “J-MoDL: Joint Model-Based Deep Learning for Optimized Sampling and Reconstruction”. In: *IEEE Journal of Selected Topics in Signal Processing* 14.6 (2020), pp. 1151–1162. DOI: 10.1109/JSTSP.2020.3004094.

[2] Cagla Deniz Bahadir, Adrian V. Dalca, and Mert R. Sabuncu. “Learning-Based Optimization of the Under-Sampling Pattern in MRI”. In: *Information Processing in Medical Imaging*. Cham: Springer International Publishing, 2019, pp. 780–792. ISBN: 978-3-030-20351-1.

[3] Cagla Deniz Bahadir et al. “Deep-Learning-Based Optimization of the Under-Sampling Pattern in MRI”. In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 1139–1152. ISSN: 23339403. DOI: 10.1109/TCI.2020.3006727.

[4] Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. “Advances in optimizing recurrent networks”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8624–8628. ISBN: 978-1-4799-0356-6. DOI: 10.1109/ICASSP.2013.6639349. arXiv: 1212.0901. URL: http://ieeexplore.ieee.org/document/6639349/.

[5] Ewout van den Berg and Michael P. Friedlander. *SPGL1: A solver for large-scale sparse reconstruction*. https://www.cs.ubc.ca/labs/scl/spgl1. Accessed: 2019-01-01.

[6] Johan Berglund, Billy Zaar, and Freddy St‘ĕhlberg. “Parallel imaging techniques: k-space Row and column Sampling Schemes”. In: *Journal of Magnetic Resonance Imaging* 14.3 (2001), pp. 287–293.

[7] Kai Tobias Block, Martin Uecker, and Jens Frahm. “Undersampled radial MRI with multiple coils. Iterative image reconstruction using a total variation constraint”. In: *Magnetic Resonance in Medicine* 57.6 (2007), pp. 1086–1098. DOI: 10.1002/mrm.21236.

[8] *Bluetooth Core Specification*. Revision v5.3. Core Specification Working Group, 2021. URL: https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=521059.

[9] Sebastian Bosse et al. “Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment”. In: *IEEE Transactions on Image Processing* 27.1 (2018), pp. 206–219. DOI: 10.1109/TIP.2017.2760518.

[10] *brain-development.org*. Mar. 2022. URL: https://brain-development.org/.

[11] Kristian. Bredies, Karl Kunisch, and Sbatta Pock. "Total Generalized Variation". In: *SIAM Journal on Imaging Sciences* 3.3 (2010), pp. 492–526. DOI: 10.1137/090769521.

[12] Lev M Bregman. "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming". In: *USSR Computational Mathematics and Mathematical Physics* 7.3 (1965), pp. 200–217.

[13] Emmanuel Candes, Jianni Romberg, and Tsing Tao. "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information". In: *IEEE Trans. on Inf. Theory* 52.2 (2006), pp. 489–509. ISSN: 0018-9448. DOI: 10.1109/TIT.2005.862083.

[14] Darren Craven et al. "Adaptive Dictionary Reconstruction for Compressed Sensing of ECG Signals". In: *IEEE Journal of Biomedical and Health Informatics* 21.3 (2017), pp. 645–654. DOI: 10.1109/JBHI.2016.2531182.

[15] David Donoho. "Compressed sensing". In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306. ISSN: 0018-9448.

[16] Richard Dykstra. "An algorithm for restricted least squares regression". In: *Journal of the American Statistical Association* 78.384 (1983), pp. 837–842.

[17] Vineet Edupuganti et al. "Uncertainty Quantification in Deep MRI Reconstruction". In: *IEEE Transactions on Medical Imaging* 40.1 (2021), pp. 239–250. DOI: 10.1109/TMI.2020.3025065.

[18] FDA. *FDA Clears Compressed Sensing MRI Acceleration Technology From Siemens Healthineers*. Feb. 2017.

[19] Mohammad Golilarz and Dewen Hu. "DCT-DWT based image denoising and compression approach for image processing applications". In: *IEEE Access* 6 (2018), pp. 464–472.

[20] Javier Haboba et al. "A Pragmatic Look at Some Compressive Sensing Architectures With Saturation and Quantization". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2.3 (Sept. 2012), pp. 443–459. ISSN: 2156-3357. DOI: 10.1109/JETCAS.2012.2220392.

[21] Michael Healy, Thomas Newe, and Elfed Lewis. "Wireless Sensor Node hardware: A review". In: *SENSORS, 2008 IEEE*. 2008, pp. 621–624. DOI: 10.1109/ICSENS.2008.4716517.

[22] Sepp Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen". MA thesis. Technische Universit'd't M'ijnchen, 1991.

[23] Zhishen Huang and Saiprasad Ravishankar. "Single-Pass Object-Adaptive Data Undersampling and Reconstruction for MRI". In: *IEEE Transactions on Computational Imaging* 8 (2022), pp. 333–345. DOI: 10.1109/TCI.2022.3167454.

[24] Kyong Hwan Jin, Michael A. Unser, and Kwang Moo Yi. "Self-Supervised Deep Active Accelerated MRI". In: *ArXiv* abs/1901.04547 (2019).

[25] Patricia M. et al. Johnson. "Deep Learning Reconstruction Enables Prospectively Accelerated Clinical Knee MRI". In: *Radiology* 0.0 (0). PMID: 36648347, p. 220425. DOI: 10.1148/radiol.220425. eprint: https://doi.org/10.1148/radiol.220425. URL: https://doi.org/10.1148/radiol.220425.

[26] Fangyi Juefei-Xu, Xiaoyu Wu, and Marios Savvides. "A comprehensive study on countermeasures against adversarial examples". In: *IEEE International Conference on Computer Vision (ICCV) Workshops*. IEEE. 2017, pp. 1–10.

[27] Ho-Lim Jung et al. "k-t FOCUSS: A general compressed sensing framework for high resolution dynamic MRI". In: *Magnetic Resonance in Medicine* 61.1 (2009), pp. 103–116.

[28] Zbontar et al. Jure. *fastMRI: An Open Dataset and Benchmarks for Accelerated MRI*. 2018. eprint: arXiv:1811.08839.

[29] Diederik P Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[30] Florian Knoll et al. "Adapted Random Sampling Patterns for Accelerated MRI". In: *Magnetic Resonance Materials in Physics, Biology and Medicine* 24.1 (2011), pp. 43–50. DOI: 10.1007/s10334-010-0231-9.

[31] Florian Knoll et al. "Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge". In: *Magnetic Resonance in Medicine* 84.6 (Dec. 2020), pp. 3054–3070. ISSN: 0740-3194. DOI: 10.1002/mrm.28338. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/mrm.28338https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.28338https://onlinelibrary.wiley.com/doi/10.1002/mrm.28338.

[32] Sushant Kumar, Bhabesh Deka, and Sumit Datta. "Block-Sparsity Based Compressed Sensing for Multichannel ECG Reconstruction". In: *Pattern Recognition and Machine Intelligence*. Ed. by Bhabesh Deka et al. Cham: Springer International Publishing, 2019, pp. 210–217. ISBN: 978-3-030-34872-4.

[33] Thomas K'ijstner et al. "Automatic Motion Artifact Detection for Whole-Body Magnetic Resonance Imaging". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 995–999. DOI: 10.1109/ICASSP.2018.8462414.

[34] Peder E Z Larson, Peter T Gurney, and Dwight G Nishimura. "Self-gated cardiac cine MRI". In: *Magnetic Resonance in Medicine* 51.1 (2004), pp. 93–102.

[35] Siyuan Liu et al. "Real-Time Quality Assessment of Pediatric MRI via Semi-Supervised Deep Nonlocal Residual Neural Networks". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 7697–7706. DOI: 10.1109/TIP.2020.2992079.

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[37] Michael Lustig, David Donoho, and John M Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging". In: *Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195.

[38] Michael Lustig and John Pauly. "SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space". In: *Magnetic Resonance in Medicine* 64.2 (2010), pp. 457–471.

[39] Michael Lustig et al. "Compressed Sensing MRI". In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 72–82. ISSN: 1053-5888. DOI: 10.1109/MSP.2007.914728. URL: http://ieeexplore.ieee.org/document/4472246/.

[40] Michael Lustig et al. "k-t SPARSE: High frame rate dynamic MRI exploiting spatio-temporal sparsity". In: *Proceedings of the International Society for Magnetic Resonance in Medicine*. 2008.

[41] Kede Ma et al. "End-to-End Blind Image Quality Assessment Using Deep Neural Networks". In: *IEEE Transactions on Image Processing* 27.3 (2018), pp. 1202–1213. DOI: 10.1109/TIP.2017.2774045.

[42] Hossein Mamaghanian et al. "Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes". In: *IEEE Transactions on Biomedical Engineering* 58.9 (2011), pp. 2456–2466. DOI: 10.1109/TBME.2011.2156795.

[43] Mauro Mangia, Riccardo Rovatti, and Gianluca Setti. "Rakeness in the Design of Analog-to-Information Conversion of Sparse and Localized Signals". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 59.5 (2012), pp. 1001–1014. DOI: 10.1109/TCSI.2012.2191312.

[44] Mauro Mangia et al. "Deep Neural Oracles for Short-Window Optimized Compressed Sensing of Biosignals". In: *IEEE Transactions on Biomedical Circuits and Systems* 14.3 (June 2020), pp. 545–557. ISSN: 19409990. DOI: 10.1109/TBCAS.2020.2982824.

[45] Mauro Mangia et al. "Rakeness-based design of low-complexity compressed sensing". In: *IEEE Trans. on Circuits and Systems I: Reg. Papers* 64.5 (2017), pp. 1201–1213.

[46] Alex Marchioni et al. "Adapted Compressed Sensing with Incremental Encoder and Deep Performance Predictor for Low-Power Sensor Node Design". In: *2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. 2023, pp. 1–6. DOI: 10.1109/I2MTC53148.2023.10175954.

[47] F. Martinini et al. "Training Binary Layers by Self-Shrinking of Sigmoid Slope: Application to Fast MRI Acquisition". In: *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. 2022, pp. 665–669. DOI: 10.1109/BioCAS54905.2022.9948688.

[48] Filippo Martinini et al. “A Deep Learning Method for Optimal Undersampling Patterns and Image Recovery for MRI Exploiting Losses and Projections”. In: *IEEE Journal of Selected Topics in Signal Processing* 16.4 (2022), pp. 713–724. DOI: 10.1109/JSTSP.2022.3171082.

[49] Filippo Martinini et al. “Binary Compressed Sensing of ECG by Neural Matrix Optimization and Support Oracle”. In: *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. 2022, pp. 660–664. DOI: 10.1109/BioCAS54905.2022.9948666.

[50] Filippo Martinini et al. “Compressed Sensing Inspired Neural Decoder for Undersampled MRI with Self-Assessment”. In: *2021 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. 2021, pp. 01–06. DOI: 10.1109/BioCAS49922.2021.9644958.

[51] Filippo Martinini et al. “Incremental undersampling MRI acquisition with neural self assessment”. In: *Signal Processing* (2024), p. 109746. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2024.109746. URL: https://www.sciencedirect.com/science/article/pii/S0165168424003669.

[52] Patrick E McSharry et al. “A Dynamical Model for Generating Synthetic Electrocardiogram Signals”. In: *IEEE Transactions on Biomedical Engineering* 50.3 (2003), pp. 289–294.

[53] Craig H Meyer et al. “Spiral MR imaging”. In: *Magnetic Resonance in Medicine* 28.2 (1992), pp. 202–213.

[54] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. “No-Reference Image Quality Assessment in the Spatial Domain”. In: *IEEE Transactions on Image Processing* 21.12 (2012), pp. 4695–4708. DOI: 10.1109/TIP.2012.2214050.

[55] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. “Making a Completely Blind Image Quality Analyzer”. In: *IEEE Signal Processing Letters* 20.3 (2013), pp. 209–212. DOI: 10.1109/LSP.2012.2227726.

[56] Amir Mousavi, Qiang Fu, and Trung Bao Le. “ReconNet: A Fast and Efficient Deep Neural Network for Compressed Sensing in ECG Signal Reconstruction”. In: *IEEE Journal of Biomedical and Health Informatics* 23.6 (2019), pp. 2400–2409. DOI: 10.1109/JBHI.2019.2897050.

[57] Matthew J. Muckley et al. “Results of the 2020 fastMRI Challenge for Machine Learning MR Image Reconstruction”. In: *IEEE Transactions on Medical Imaging* (2021), pp. 1–1. ISSN: 0278-0062. DOI: 10.1109/TMI.2021.3075856. URL: https://ieeexplore.ieee.org/document/9420272/.

[58] Deanna Needell and Joel A Tropp. “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. In: *Applied and Computational Harmonic Analysis* 26.3 (2009), pp. 301–321.

[59] Anteo Pantelopoulos and Nanni Giacomo Bourbakis. “A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.1 (2010), pp. 1–12. DOI: 10.1109/TSMCC.2009.2032660.

[60] Nicholas R Pipitone and Craig K Yue. “High temporal resolution radial k-space trajectory for steady-state free precession MRI”. In: *Magnetic Resonance in Medicine* 41.3 (1999), pp. 529–536.

[61] Klaas Pruessmann et al. “SENSE: Sensitivity encoding for fast MRI”. In: *Magnetic Resonance in Medicine* 42.5 (1999), pp. 952–962.

[62] Chen Qin et al. “Attention-Guided Network for Ghost-Free High Dynamic Range Imaging”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2901–2910. DOI: 10.1109/CVPR.2018.00307.

[63] Sundeep Rangan. “Generalized approximate message passing for estimation with random linear mixing”. In: *2011 IEEE International Symposium on Information Theory Proceedings*. 2011, pp. 2168–2172. DOI: 10.1109/ISIT.2011.6033942.

[64] Saiprasad Ravishankar and Yoram Bresler. “MR image reconstruction from hzighly undersampled k-space data by dictionary learning”. In: *IEEE Transactions on Medical Imaging* 30.5 (May 2011), pp. 1028–1041. ISSN: 02780062. DOI: 10.1109/TMI.2010.2090538.

[65] Abdul Rehman and Zhou Wang. “Reduced-Reference Image Quality Assessment by Structural Similarity Estimation”. In: *IEEE Transactions on Image Processing* 21.8 (2012), pp. 3378–3389. DOI: 10.1109/TIP.2012.2197011.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9351 (2015), pp. 234–241. ISSN: 16113349. DOI: 10.1007/978-3-319-24574-4{_}28.

[67] Ravindran Sandeep. “Five Ways Deep Learning Has Transformed Image Analysis”. In: *Nature* (2024). DOI: 10.1038/nature12345.

[68] Jo Schlemper et al. “A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction”. In: *IEEE Transactions on Medical Imaging* 37.2 (2018), pp. 491–503. DOI: 10.1109/TMI.2017.2760978.

[69] Jo Schlemper et al. “Enabling Joint Optimization of Sampling Patterns and Reconstruction for Accelerated MRI”. In: *IEEE Transactions on Medical Imaging* 39.12 (2020), pp. 4477–4488. DOI: 10.1109/TMI.2020.3035687.

[70] Matthias Seeger et al. “Optimization of K-space Trajectories for Compressed Sensing by Bayesian Experimental Design”. In: *Magnetic Resonance in Medicine* 63.1 (2010), pp. 116–126. DOI: 10.1002/mrm.22187.

[71] Pranab Kumar Sen. "Estimates of the Regression Coefficient Based on Kendall's Tau". In: *Journal of the American Statistical Association* 63.324 (1968), pp. 1379–1389. ISSN: 1537274X. DOI: 10.1080/01621459.1968.10480934.

[72] Yunfeng Shi et al. "CS-Net: An End-to-End Compressed Sensing Network for ECG Signal Reconstruction". In: *IEEE Transactions on Biomedical Engineering* 67.7 (2020), pp. 1750–1760. DOI: 10.1109/TBME.2019.2958751.

[73] Anurag Singh and Samarendra Dandapat. "Exploiting multi-scale signal information in joint compressed sensing recovery of multi-channel ECG signals". In: *Biomedical Signal Processing and Control* 29 (2016), pp. 53–66. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2016.05.008. URL: https://www.sciencedirect.com/science/article/pii/S1746809416300507.

[74] Michael Spork, Carlo Alberto Boano, and Kay Romer. "Performance and Trade-Offs of the New PHY Modes of BLE 5". In: *Proceedings of the ACM MobiHoc Workshop on Pervasive Systems in the IoT Era*. PERSIST-IoT '19. Association for Computing Machinery, 2019, 7‘Ă‘Ş12. ISBN: 9781450368056. DOI: 10.1145/3331052.3332471. URL: https://doi.org/10.1145/3331052.3332471.

[75] Anuroop Sriram et al. "End-to-End Variational Networks for Accelerated MRI Reconstruction". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2020), pp. 64–73. DOI: 10.1007/978-3-030-59710-8_7.

[76] Anuroop Sriram et al. "GrappaNet: Combining Parallel Imaging With Deep Learning for Multi-Coil MRI Reconstruction". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020, pp. 14303–14310. ISBN: 978-1-7281-7168-5. DOI: 10.1109/CVPR42600.2020.01432. URL: https://ieeexplore.ieee.org/document/9157643/.

[77] Liang Sun et al. "High-Order Feature Learning for Multi-Atlas Based Label Fusion: Application to Brain Segmentation With MRI". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 2702–2713. DOI: 10.1109/TIP.2019.2952079.

[78] Henri Theil. "A Rank-Invariant Method of Linear and Polynomial Regression Analysis". In: *Proceedings of the Royal Netherlands Academy of Sciences 53 (1950) Part I: 386-392, Part II: 521-525, Part III: 1397-1412*. Springer, Dordrecht, 1950. DOI: 10.1007/978-94-011-2546-8{_}20. URL: https://link.springer.com/chapter/10.1007/978-94-011-2546-8_20.

[79] Joel A Tropp and Anna C Gilbert. "Signal recovery from random measurements via orthogonal matching pursuit". In: *Information Theory, IEEE Transactions on* 53.12 (2007), pp. 4655–4666.

[80] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[81] Qingsen Yan, Dong Gong, and Yanning Zhang. "Two-Stream Convolutional Networks for Blind Image Quality Assessment". In: *IEEE Transactions on Image Processing* 28.5 (2019), pp. 2200–2211. DOI: 10.1109/TIP.2018.2883741.

[82] Guang Yang et al. "DAGAN: Deep De-Aliasing Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1310–1321. DOI: 10.1109/TMI.2017.2785879.

[83] Jing Yang, Yu Zhang, and Wo Yin. "A Fast Alternating Direction Method for TVL1-L2 Signal Reconstruction from Partial Fourier Data". In: *IEEE Journal of Selected Topics in Signal Processing* 4.2 (2010), pp. 288–297. DOI: 10.1109/JSTSP.2010.2042333.

[84] Zhao Yang et al. "An RNN-Based Compressed Sensing Decoder for ECG Signal Reconstruction". In: *IEEE Transactions on Biomedical Engineering* 67.9 (2020), pp. 2658–2669. DOI: 10.1109/TBME.2020.2975165.

[85] Tianwei Yin et al. "End-to-End Sequential Sampling and Reconstruction for MRI". In: *Proceedings of Machine Learning for Health*. Vol. 158. Proceedings of Machine Learning Research. PMLR, 2021, pp. 261–281.

[86] Hongyu Zhang et al. "DeepCS: Compressed Sensing of ECG Signals Using Convolutional Neural Networks". In: *IEEE Transactions on Biomedical Engineering* 68.5 (2021), pp. 1481–1492. DOI: 10.1109/TBME.2021.3055365.

[87] Yiran Zhang, Jiaming Liu, and Xiaofang Wang. "Learnable Radial Subsampling for MRI Reconstruction". In: *IEEE Transactions on Medical Imaging* (2022). DOI: 10.1109/TMI.2022.3147978.

[88] Yiran Zhang, Guang Yang, and Shanshan Wang. "Deep Reinforcement Learning Based Unrolling Network for MRI Undersampling Pattern Optimization". In: *IEEE Transactions on Medical Imaging* (2021). DOI: 10.1109/TMI.2021.3067864.

Acknowledgments

This thesis entirely goes to my PhD peers Andriy, Silvia, Livia, Clemence, and Lorenzo for every test, paper, Friday call, deadline, review, figure, missing reference, credit, document to sign, and academic struggle we endured together. It would have been terrible if not for the memes we forged from it all.

In this moment of commotion, I realize I am leaving the lab for good. We made a great job and I hope our legacy will remain for the years to come. It is a duty for the ones who will come after us not to lose the memory of what we created. Starting from The Great Tale of Fiuggi, or the teaching of Christian, even the genocide of Poline or the Braccacoglioni, the conquer of Angola, the song of Bibidi, the hemorrhoid of Bernoulli... and so many more.

Goodbye, peers. It was the best time of life.