UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MEDIZINISCHE INFORMATIK

Aus dem Institut für Medizinische Informatik der Universität zu Lübeck
Direktor: Prof. Dr. rer. nat. habil. Heinz Handels

Fast and Efficient Registration of Subsampled MRI Data Using Neural Networks Utilizing K-Space Properties

Schnelle und Effiziente Registrierung Unterabgetasteter MRT Daten Mittels Neuraler Netze Unter Ausnutzung von K-Raum Eigenschaften

Masterarbeit
im Rahmen des Studienganges Medizinische Informatik
der Universität zu Lübeck

vorgelegt von
Jan Meyer

ausgegeben und betreut von
Prof. Dr. Mattias Heinrich

mit Unterstützung von
M.Sc. Ziad Al-Haj Hemidi und M.Sc. Eytan Kats

Lübeck, den 21. Dezember 2024

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Lübeck, den 21. Dezember 2024

Abstract

While magnetic resonance imaging is a widely used medical imaging technology with many benefits, it suffers from long acquisition times that hamper its effectiveness. To combat this, the k-space data acquired from the scanner is subsampled to speed up the process. This, however, leads to artifacts in the reconstructed images proportionate to the amount of subsampling. Long acquisition times also lead to potential movement between image frames as motion from e.g. the lung and heart that cannot be suppressed leading to further motion artifacts. These problems can be addressed by image registration which was traditionally done by computationally expensive and slow iterative algorithms. In this thesis, the usage of specialized neural networks for efficient and fast image registration of subsampled magnetic resonance imaging data is explored. The research question is approached from multiple different angles examining registration performance as well as usability in a larger motion-compensated reconstruction pipeline. The foundations of magnetic resonance imaging, deep learning and image registration are explained and multiple experiments described. These include diverse parameter tests and ablation studies as well as downstream tests on two different datasets to find optimal model parameters and test the applicability of the networks. The registration performance is compared to traditional registration algorithms and state-of-the-art neural networks using segmentations for accurate assessment across four different amounts of subsampling. We were able to outperform both traditional registration algorithms and state-of-the-art neural networks in terms of Dice score computed on the segmentations even for strongly subsampled data. In this thesis we showed that specialized neural networks perform best for both pure registration as well as usage within an motion-compensated reconstruction pipeline while being fast and efficient. In future work we plan to extend the networks to 3D and test the performance on more diverse datasets.

Kurzfassung

Obwohl Magnetresonanztomographie eine weit verbreitete medizinische Bildgebungstechnologie mit vielen Vorteilen ist, hat es Probleme mit langen Aufnahmezeiten. Um dies zu bekämpfen werden die k-Raum Daten, welche vom Scanner aufgenommen werden, unterabgetastet, um den Prozess zu beschleunigen. Dies führt allerdings zu Artefakten in den rekonstruierten Bilder proportional zu der Menge an Unterabtastung. Lange Aufnahmezeiten führen außerdem zu mehr potentieller Bewegung durch z.B. Lunge und Herz in aufeinanderfolgenden Bildern, die nicht unterdrückt werden können, wodurch Bewegungsartefakte entstehen. Diese Probleme können mittels Bildregistrierung vermindert werden, wobei traditionell rechenintensive und langsame iterative Algorithmen verwendet wurden. In dieser Arbeit wird die Nutzung spezialisierter Neuraler Netwerke für effiziente und schnelle Bildregistrierung von unterabgetasteter Magnetresonanztomographie Daten untersucht. Diese Forschungsfrage wurde aus mehreren Blickwinkeln betrachtet und untersucht die Registrierungsqualität sowie Nutzbarkeit in einer bewegungskompensierten Rekonstruktions-Pipeline. Die Grundlagen von Magnetresonanztomographie, Deep Learning und Bild-Registrierung werden erklärt und mehrere Experimente beschrieben. Diese beinhalten diverse Parameter-Tests und Ablationsstudien sowie Downstream-Tests auf zwei verschiedenen Datensätzen, um optimale Netzwerk-Parameter zu finden und die Anwendbarkeit der Netzwerke zu testen. Die Registrierungsqualität wird mit traditionellen Registierungsalgorithmen und aktuellen Neuralen Netzwerken verglichen, wobei Segmentierungen für eine akkurate Beurteilung über verschiedene Unterabtastungsraten ermöglicht. In dieser Arbeit konnten wir zeigen, dass spezialisierte Neurale Netzwerke am besten für pure Registrierung und als Teil einer Bewegungskompensierten Rekonstruktions-Pipeline funktionieren, während sie gleichzeitig sehr schnell und effizient sind. In der Zukunft planen wir die Netzwerke auf 3D zu erweitern und diese auf diversen weiteren Datensätzen zu testen.

Contents

1 Introduction **1**
- 1.1 Motivation and Challenges of MRI Acquisition . . . 1
- 1.2 Related Work . . . 2
- 1.3 Contributions and Structure of the Thesis . . . 3

2 Fundamentals **5**
- 2.1 Magnetic Resonance Imaging . . . 5
  - 2.1.1 Magnetic Excitation and Relaxation . . . 5
  - 2.1.2 Image Acquisition and the Concept of k-Space . . . 9
  - 2.1.3 Imaging Acceleration and Reconstruction . . . 14
  - 2.1.4 Motion-Compensated Image Reconstruction . . . 18
- 2.2 Image Transformations and Registration . . . 19
  - 2.2.1 Image Transformations . . . 20
  - 2.2.2 Image Registration . . . 21
- 2.3 Deep Learning . . . 22
  - 2.3.1 Deep Learning Architectures . . . 22
  - 2.3.2 Deep Learning for Image Registration . . . 23
  - 2.3.3 Network Training and Testing . . . 24

3 Materials and Methods **27**
- 3.1 Efficient Registration using Neural Networks . . . 27
  - 3.1.1 Problem Formulation . . . 27
  - 3.1.2 Fourier-Net . . . 28
  - 3.1.3 Fourier Net+ . . . 35
- 3.2 Datasets . . . 38
  - 3.2.1 ACDC Dataset . . . 38
  - 3.2.2 CMRxRecon Dataset . . . 40
  - 3.2.3 Simulated Motion . . . 42
- 3.3 Experiments . . . 43
  - 3.3.1 Ablation Studies and Parameter Tests . . . 44
  - 3.3.2 Registration Performance on Subsampled Data . . . 46
  - 3.3.3 Motion-Compensated Reconstruction Pipeline . . . 46

4 Results **48**
- 4.1 Ablation Studies and Parameter Tests . . . 48
  - 4.1.1 Fourier-Net versus Fourier-Net+ . . . 48
  - 4.1.2 Starting Channel Size . . . 50
  - 4.1.3 Fourier-Transform Crop Size . . . 51

4.1.4 Comparison with VoxelMorph 52
4.1.5 Dense Displacement on Accelerated Data 53
4.2 Registration Performance on Subsampled Data 54
4.3 Integration into a Motion-Compensated Reconstruction Pipeline 61
4.3.1 K-Space Line Swapping 61
4.3.2 Non-Linear Lung Transformations 63

5 Discussion **66**
5.1 Ablation Studies and Parameter Tests 66
5.2 Registration Performance on Subsampled Data 69
5.3 Integration into a Motion-Compensated Reconstruction Pipeline 70
5.4 Efficiency and Performance Trade-offs 72
5.5 Limitations and Outlook 72

6 Conclusion **74**

Bibliography **75**

1

Introduction

Magnetic resonance imaging (MRI) is a commonly used medical imaging technique based on measuring magnetic fields and radio waves [1, 2, 3]. Despite its many benefits, the method is weighed down by long acquisition times. Thus, MRI acquisition is usually accelerated by subsampling the k-space in which the raw MR data is recorded. This, however, leads to problematic image artifacts that hinder further processing. Another problem of the slow data acquisition are motion artifacts which are common for organs like lung and heart [4]. These challenges, which are further discussed in the next section, have been traditionally addressed with computationally intensive algorithms. New approaches based on deep neural networks promise to rectify these problems. Neural networks have seen a rise in popularity in recent years in fields such as image processing due to breakthroughs enabled through rapid improvements in computational hardware like graphics processing units (GPUs) [5]. While segmentation networks like the *U-Net* [6] were established quite early with great success, usage of these architectures for registration and even networks for aliasing-free MR reconstruction like [7, 8, 9] have recently come into focus [5, 10]. In this thesis, the possibility of using an unsupervised deep learning approach to align subsampled MRI data as well its potential usage in a motion reconstruction pipeline will be investigated.

1.1 Motivation and Challenges of MRI Acquisition

MRI has a variety of benefits because it is non-invasive, radiation-free and has great contrast for soft tissue. It has a variety of uses including generating high-resolution anatomical and functional images as it is sensitive to many physical properties, like T_1 and T_2 relaxation times, diffusion, and flow. Image contrast can be weighted by these properties through adjusting the scan settings to highlight different anatomical or physiological features [11]. While it is comparable in many regards to Computed Tomography (CT), it is often favored by physicians due to these reasons [12]. However, acquisition speed is its main weakness as a full-body scan can take up to 30 minutes depending on the number of slices scanned [1, 13]. This is a burden on patients due to needing to remain still for a long amount of time in the scanner as well as hindering general efficiency. In emergency situations, for example, CT will generally be used instead of MRI as the radiation is acceptable when in need of quick diagnostic imaging [12]. To understand the nuances of this comparison further, the general process of MR acquisition needs to be understood.

There are many challenges with MR acquisition due to the nature of the image technology. Firstly, the images are not acquired directly in image space, but the raw data is instead measured in the so-called k-space. This is a Fourier space holding the image frequencies from which the scans can be reconstructed using an inverse Fourier transform (iFT) [1]. The frequencies stem from the magnetic resonance of the measured subject. The magnetic excitation and relaxation, however, takes time which leads to large total acquisition times as the process needs to be repeated for each line in the k-space to acquire a full-resolution image [11]. For a spin-echo imaging sequence the acquisition time of 5 ± 10 ms for each of the 192 ± 256 phase-encoding steps which leads to a total net acquisition time of 1 ± 2 s per slice. However, the total acquisition time of the sequence is even longer due to the repetition time defined by e.g. the T_1-contrast [14]. Thus, the time needed to acquire a single image can range from hundreds of milliseconds or less for certain scans (e.g. gradient echo) up to several minutes for others (e.g. spin echo) [11].
An easy approach for acceleration is to simply measure less lines and fill the missing lines with zeros [15]. This is usually done for the higher frequencies as these hold information of edges and finer structures in the image, which are deemed less important than the image contrast stored in lower frequencies [11]. This missing information, however, leads to artifacts upon reconstruction as the Shannon-Nyquist sampling theorem [16] is violated. Consequently, this kind of acceleration is usually called under- or subsampling of the k-space. The more lines are zeroed out, the worse the artifacts become as more information is missing. This leads to problems for further processing as many algorithms struggle with these artifacts [15].
Another problem with the long acquisition times is patient movement. While most patient can remain relatively still for up to 30 minutes of measuring, the lung and heart movement cannot simply be stopped. Breath-holds (BH) can be imployed [17], however it is hard to exactly reproduce the anatomical positions by breathing in the same amount of air each time [18]. Instead, the motion is often estimated and compensated after reconstruction using registration algorithms or networks [7, 19, 20].

1.2 Related Work

Traditionally, mathematical optimization based algorithms like *DARTEL* [21], *Demons* [22], *NiftyReg* [23], *LAP* [24] and *FLASH* [25] have been used to iteratively solve registration tasks. Most of these use image similarity metrics like the mean squared error (MSE) or mutual information (MI) to judge the alignment of the images [26]. Once the similarity has reached an acceptable range, the algorithm stops and the registration is complete. While these algorithm are well-established and based on mathematical theory, they are also very time consuming and computationally expensive [19].
A simple way to solve the time problem is to use neural networks to learn the behavior of these algorithms as these are, once trained, fast in execution. Displacements can be generated with the slow mathematical algorithms as ground truth substitutes for the supervised training of networks like *LAPNet* [27] and *DeepFlash* [28]. The loss can then be calculated using e.g. the MSE between the displacement generated by the network

and the calculated ground truth. Both of these approaches, however, still do not solve the memory problem present as the networks are quite large due to complex values needing to be computed. The performance of these networks might also be limited by the quality of the ground truth displacements which need to be generated just for the training [5, 29].
Neural networks that can be trained in an unsupervised manner are a logical next step as they do not require ground truth displacements to be trained. Instead they use similarity metrics to compare the images after applying a transformation in order to calculate a loss similar to iterative algorithms. Such networks are *VoxelMorph* [30], *IC-Net* [31], *SYM-Net* [32], *Fourier-Net* [33] and *Fourier-Net+* [34]. The latter use the *SYM-Net* as part of their architecture and are the main networks used in this thesis as they are designed to not only be fast, but also more memory efficient compared to the other neural networks. This is due to only utilizing the real-valued image space and having a decoder without trainable weights, which makes the architecture more lightweight compared to similar networks [33, 34].
Another important topic is MRI reconstruction where again iterative algorithms are commonly used [11, 35, 36, 37]. However, more neural network based approaches have recently come into focus [15, 38]. A important aspect in this regard, as discussed in the previous section, is patient motion. While motion in time-series data can often be corrected after the reconstruction by registration of the individual frames, a growing interest in motion-compensated reconstruction can be observed [39]. In this process, the reconstruction and motion-correction are not done sequentially, but in one pipeline that optimizes both in a joint process [7]. Both the reconstruction and the motion-correction aspects of the pipeline can be done by either a traditional algorithm or neural networks. While there are a lot of works that cover this using only iterative algorithms [20], recent works have tried to use only neural networks due to time benefits [9, 40].

1.3 Contributions and Structure of the Thesis

As discussed in the previous section, there is a lack of efficient and fast registration methods. Traditional iterative algorithms fail in both regards, while most neural networks struggle with memory efficiency [33, 34]. Additionally, when tackling highly accelerated data, images have heavy artifacts due to the subsampling applied in the k-space. Most image based approaches struggle to deal with these, however, robust methods like *LAPNet* work directly in k-space [27]. This allows for avoiding the reconstruction artifacts, but makes the networks inherently less efficient due to needing to work with complex values which neural networks are not well equipped to handle [41]. Besides memory concerns, *LAPNet* is also a supervised approach, meaning that the ground truth displacements need to be calculated with the *LAP* algorithm for all the training data. This, however, has been rectified with the recent introduction of *LAPANet* [42], which is a unsupervised version with attention blocks that works without the need for ground truth displacements.
All of these challenges call for an efficient unsupervised neural network that can with-

stand subsampling artifacts despite working in the image space. For this, we choose to expand the already established work of [33, 34] as *Fourier-Net* and *Fourier-Net+* are unsupervised neural networks which we will use to efficiently align subsampled MR images. While the network architectures were already tested for e.g. inter-patient brain scan alignment in the literature, a new use-case with undersampled cardiac MR scan will be explored. The registration performance will also be compared to both traditional iterative registration algorithms and other neural networks. There, the applicability to this new task will be shown with better results compared to the other methods. Furthermore, a potential use-case in a motion-compensated image reconstruction pipeline will be tested demonstrating the networks potential. Main benefits compared to traditional algorithms are the additionally computation speed and low memory consumption compared to other neural networks while maintaining performance even on highly accelerated data.
In chapter 2, the foundations of the thesis are introduced, which includes general information about MRI such as basic magnetic principles in section 2.1.1 image acquisition (section 2.1.2), acceleration (section 2.1.3), and reconstruction (section 2.1.4). Building on the latter, image transformations (section 2.2.1) and registration (section 2.2.2) are introduced followed by deep learning basics such as common architectures (section 2.3.1), their use in image registration (section 2.3.2) and general principles of network training and testing (section 2.3.3).
In chapter 3, the materials and methods of the thesis are explained. This includes the specific network architectures in section 3.1, namely *Fourier-Net* (section 3.1.2) and *Fourier-Net+* (section 3.1.3), followed by the datasets in section 3.2. Lastly, the conducted experiments are explained in section 3.3.
In chapter 4 the results for these experiments are presented and discussed in chapter 5. This includes a look at limitations and future work followed by a brief summary and conclusion in chapter 6.

2

Fundamentals

In this chapter, the foundations of the thesis are explained. Starting with the principles behind MRI acquisition, acceleration and reconstruction as well as image registration and deep learning fundamentals.

2.1 Magnetic Resonance Imaging

MRI is a non-invasive, radiation-free, tomographic imaging technology based on measurements of a magnetic field. An MRI machine comprises four main components, as seen in Figure 2.1. The first component is a strong magnet powerful enough to generate a static magnetic field B_0 that is required to induce nuclear proton polarization. The second is a radio frequency (RF) system which generates an alternating magnetic field B_1 at the resonant frequency f and detects the MR signal that is returned from the patient. The third component is the set of gradient systems (oriented orthogonally in X, Y and Z directions) that generates linear magnetic field variations, which are then superimposed upon B_0 and are used to spatially encode the MR signal. In clinical MR scanners, the three gradient sets and whole-body RF coils are typically concentrically positioned inside the bore of the magnet. The fourth component is a computer providing the user interface, generating images to be displayed and interpreted on the console [2].

2.1.1 Magnetic Excitation and Relaxation

The base principles behind MR imaging are RF excitation and relaxation. Through these, the raw k-space data is measured which enables the reconstruction to an image. Terminology such as T_1 and T_2 relaxation times are introduced and explained. These are the basis for the pulse sequences discussed in section 2.1.2.

Radio Frequency Excitation

Individual nuclei precess around the field B_0 at a resonance frequency known as the Larmor frequency ω_0 of the net magnetization vector, which generally occurs in the RF range of the electromagnetic spectrum and is related to the external magnetic field as:

$$\omega_0 = \gamma \cdot B_0, \tag{2.1}$$

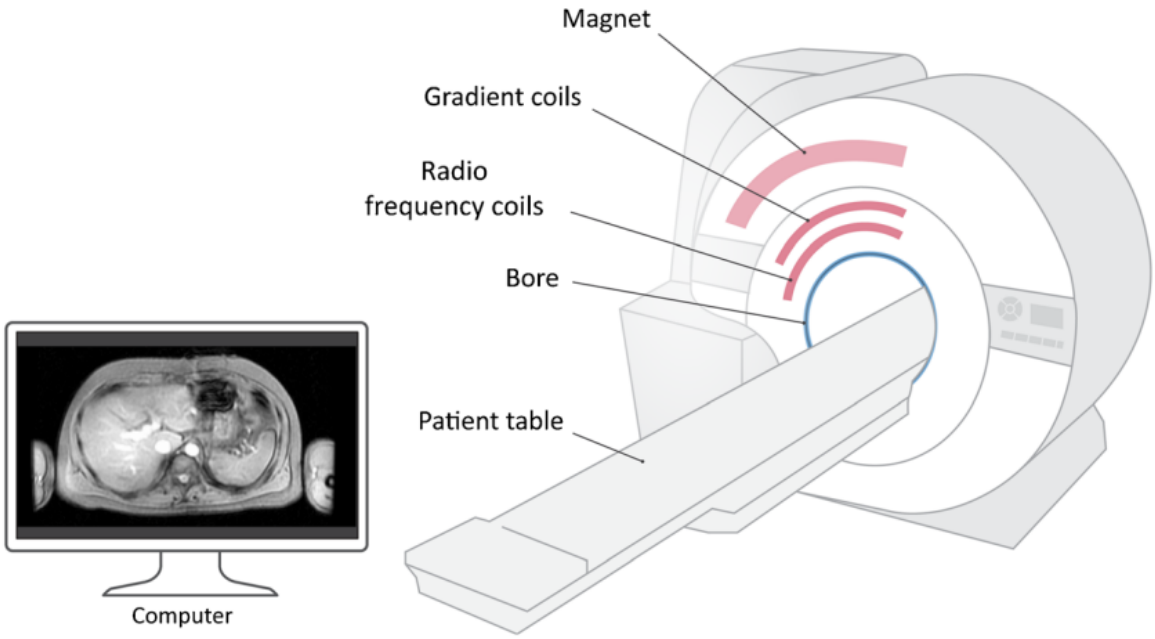


Figure 2.1: Schematic of an MRI scanner taken from [2].

with γ being the gyromagnetic ratio, which is a fixed value depending on the nuclei [14]. When nuclei are placed in the presence of a strong static magnetic field such as B_0 the nuclei split into two energy states, either aligned parallel to the magnetic field B_0 (called a spin-up state) or aligned anti-parallel to B_0 (called a spin-down state). The spin-up state has a slightly lower energy level as compared to the spin-down state and is therefore preferred. This slight difference in the spin states (0.001%) results in an overall net magnetization M aligned in the same direction as B_0 [2].
To create an MR signal, the spins are excited out of their resting equilibrium, i.e. tipping M away from B_0. To detect the signal from the hydrogen nuclei in the tissues an additional external field B_1 is introduced at the resonant Larmor frequency ω_0 that can affect magnetization vector, causing it to rotate into a plane orthogonal to its original orientation. The rotated vector continues to precess around the B_0. The precession of the magnetization vector in the transverse plane can be detected by an RF coil tuned to the resonant frequency ω_0. RF coils can be operated in a receive-only mode, in which case the inherent body coil is used as a transmitter; or the RF coils can be both transmit and receive. The purpose of the RF transmit coil is to create a time-varying B_1 field at right angles to B_0 that could be linearly or circularly polarized. The closer the receiving coil is to the source of the MR signal, the better the signal-to-noise ratio (SNR). Receiver coils generally comprise arrays of smaller individual coils or elements; however, each individual coil element has a limited depth penetration [2].
Multiple arrays of coils, termed phased-array coils, can be used together to achieve a higher coverage. The multiple coils are electronically decoupled from one another so that they do not appear as just a single large coil. The images from individual coils are independently reconstructed and then grouped together to create the final image. An RF pulse of amplitude B_1, called excitation pulse, is applied for a certain time

duration to tip the magnetization at an angle away from the B_0 field. The precessing transverse magnetization induces a voltage in the receiver RF coil; this induced voltage is known as the free induction decay (FID). After the pulse, the magnetization returns to thermal equilibrium by processes known as MR relaxation. To fully encode the spatial information within the field of view (FOV), pulse sequences must be iterated numerous times. The time between successive iterations of a pulse sequence is known as the repetition time (TR). The time between the application of the initial RF pulse and the middle of the detected echo is known as the echo time (TE). Due to this the overall time to acquire an MR image is quite long [2].

Magnetic Relaxation

Once the RF pulse is turned off, M continues to precess as it returns to its thermal equilibrium state. During this time, two types of relaxation occur: T_1 (longitudinal) and T_2 (transverse). One attribute that makes T_1 and T_2 so valuable for determining the signal in MRI is their sensitivity to the presence and type of tissue. It is this tissue dependence property that gives MR its excellent soft-tissue contrast [2]. T_1 relaxation describes the recovery of the longitudinal magnetization back to thermal equilibrium following a perturbation by an RF pulse. The longitudinal component regrows along the Z direction with a time constant T_1. In other words, after the RF pulse is turned off, the protons that were disturbed give their energy to the surrounding environment and return back to their original equilibrium state, realigning with B_0. Hence, T_1 relaxation is also called longitudinal relaxation. Furthermore, because T_1 relaxation involves the loss of energy that was put into the spin system by the RF pulse, it is also referred to as spin-lattice relaxation, the lattice consisting of surrounding macromolecules. This loss of energy is stimulated by the fluctuating magnetic fields associated with the dipole–dipole interactions of neighboring magnetic moments. T_1 relaxation can only occur when these magnetic field fluctuations occur at the resonant frequency ω_0. The rate at which the spin magnetization M_z recovers to M_0 at time t is called T_1 relaxation time [2]. It can be expressed as follows:

$$M_z = M_0 \cdot \left(1 - e^{-\frac{t}{T_1}}\right). \tag{2.2}$$

A preferred method of measuring T_1 relaxation is the Look-Locker method, which is based on the principle that one does not need to wait for the net magnetization vector to equilibrate in order to measure T_1. Instead, an RF pulse is used with a small flip, which can be repeatedly applied. The acquisition pulse sequence is designed to generate a train of signals that gradually approaches a steady-state recovery [2].

Cine CMR Imaging

Cardiac cine MRI is the gold standard for the assessment of cardiac morphology and function [43, 44]. To capture a motion-free image of the heart requires the image to be acquired in just a few tens of milliseconds. This means both limiting the number of

phase encoding steps (and thus spatial resolution) as well as making the TR as short as possible [45, 46]. Whilst this can be done, it is at the cost of significantly reducing the image quality. To achieve acceptable image quality, the image acquisition times would become too long to effectively freeze heart motion in the image. For routine cardiovascular magnetic resonance (CMR) therefore, the MR signals are acquired over multiple heart beats, synchronizing the pulse sequence in the cardiac cycle [18]. Cardiac synchronization is achieved by using the electrocardiogram (ECG) signal of the patient. The R wave is detect in the ECG and used to generate a synchronization pulse for the MR data acquisition [47]. This enables images of the beating heart to be obtained either at a single time point (still imaging) or at multiple time points through the cardiac cycle (cine imaging) [45]. A schematic of this process can be seen in Figure 2.2.

Conventional imaging techniques need several minutes to generate a full image [45]. This, however, leads to degradation by respiratory motion. The motion can be reduced by e.g. respiratory compensation methods (respiratory gating), cardiac synchronized fast imaging techniques combined with patient BHs or ultra-fast imaging techniques. Patient BHs combined with fast imaging techniques are most common in practice [45]. This technique often requires multiple patient BHs during data acquisition, which can be difficult for very sick or sedated patients, further complicating the work-flow and extending exam time [18]. Additionally, free-breathing cine imaging is essential and significant in specific conditions such as exercise-stressed CMR exams [48]. Therefore, free-breathing CMR with good image quality and comparable spatio-temporal resolution to traditional BH CMR is clinically desirable [49] for expanded CMR application, improved exam work-flow, and patient tolerance. Real-time (RT) cine imaging, which suppresses potential respiratory motion artifacts by acquiring each cardiac phase in one shot, allows shorter BH times or fully free-breathing acquisition but sacrifices spatio-temporal resolution [18].

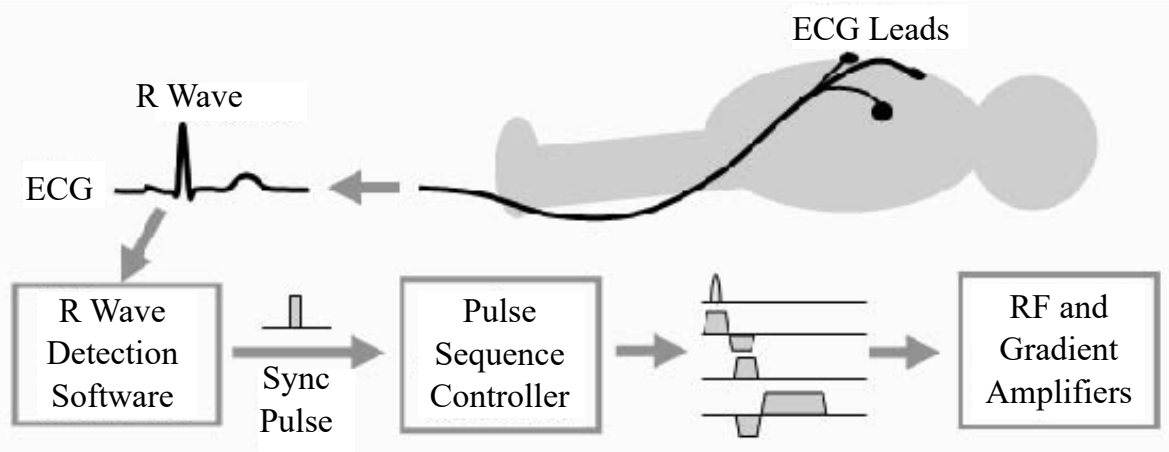


Figure 2.2: Synchronization of imaging pulse sequences via ECG, taken from [45]. The R wave is detected in the ECG, which is then used to align the pulse sequences for imaging.

2.1.2 Image Acquisition and the Concept of k-Space

After discussing the physical mechanisms behind MRI, it is time to look at the actual process of acquiring the image as well as the concept of the k-space. The image contrast in MR imaging arises from tissues generating MR signals with different intensities due to their physical properties. Contrast weighting of the MR signal is obtained by the design of pulse sequences, which consist of repetitive trains of RF pulses. Two of the most common pulse sequences are gradient echo and spin echo sequences.

Gradient Echo Sequence

In a gradient echo (GE) sequence, the FID signal is manipulated by a bi-polar gradient. The excitation pulse tilts the magnetization by α degrees. For $\alpha = 90°$, the longitudinal magnetization is rotated in the transverse plane. The data is sampled during a GE at the echo time (TE) after the excitation pulse. This GE is achieved by dephasing the spins with a negative gradient before they are rephased by an opposite gradient with opposite polarity to generate an echo. A schematic overview of the process can be seen in Figure 2.3.

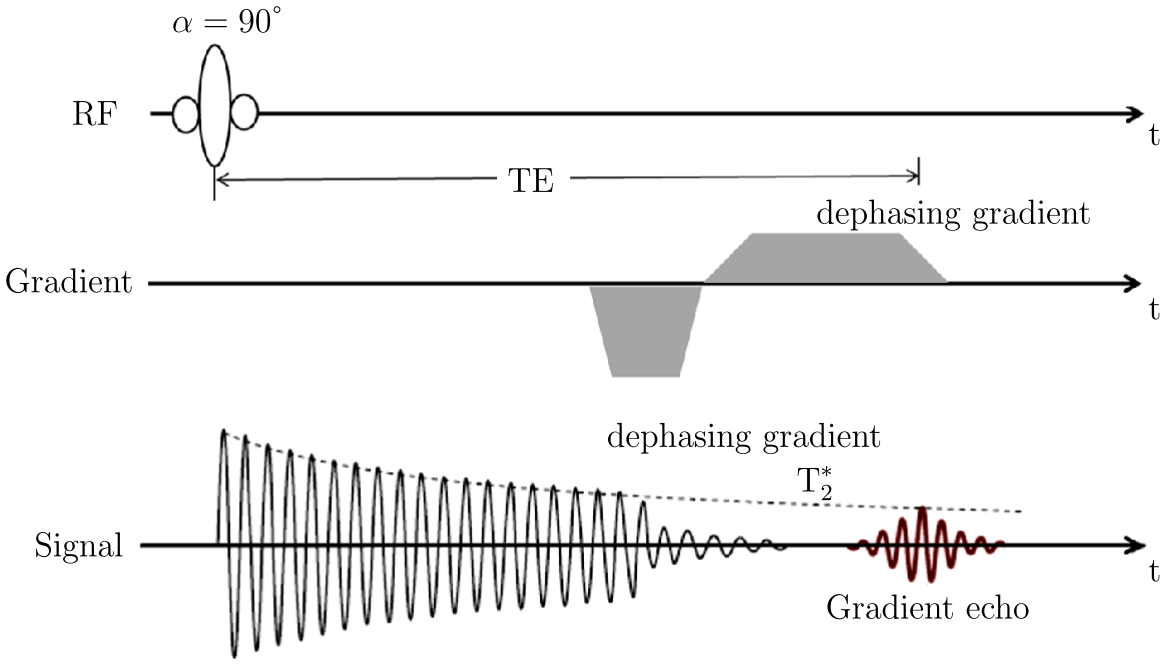


Figure 2.3: Schematic of an gradient echo sequence taken from [13]. The FID signal is manipulated by a bi-polar gradient to create a gradient echo. The excitation RF pulse tilts the magnetization by 90° and the signal is sampled during a gradient echo at time TE which is caused by a bi-polar gradient.

Spin Echo Sequence

With a spin echo (SE) sequence, pure T_2-weighted contrast can be generated. When a 90° pulse rotates the magnetization into the transverse plane, the resulting FID signal quickly decays due to the strong T_2^* dephasing, which is the T_2 decay shortened by extra phase dispersion [2]. If after a time $\frac{TE}{2}$ a 180° pulse is applied, the spins will be flipped and start to rephase. After another time, $\frac{TE}{2}$, a measurable echo signal is created. The spin dephasing due to static magnetic field inhomogeneities is compensated by inverting the spins with the 180° refocusing pulse. This process is visualized in Figure 2.4.
Consequently, the decay of the signal at time TE will solely originate from the T_2 relaxation. A SE sequence can also be used to generate proton density or T_1-weighted signals by using a short TE and a long or short TR, respectively [13].

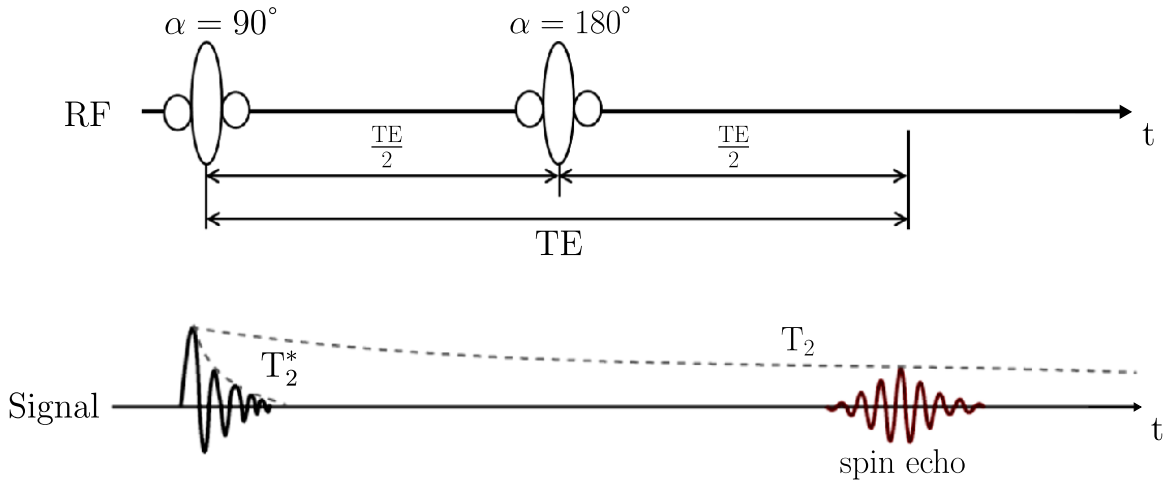


Figure 2.4: Schematic of an spin echo sequence taken from [13]. A 90° pulse rotates the magnetization resulting in a quickly decaying FID signal. At $\frac{TE}{2}$ the spins are flipped by applying a 180° pulse and give rise to a spin echo at time TE after waiting another $\frac{TE}{2}$ following the 180° pulse.

Introduction to k-Space

The concept of the k-space is a generalization of the simple relation of a time-variant signal to a spectrum of two or more dimensions. An 2D image is related to a 2D k-space data set by a 2D FT, as seen in Figure 2.5. As the FT is an information-preserving operation, the k-space data contains exactly the same information as the image data. Thus, in order to get the full image information, the full k-space data needs to be measured. The task of forming an image by this approach is then converted to the task of finding a way to measure the necessary corresponding k-space data. Using the Larmor frequency from Equation 2.1 the coordinates in 2D k-space will thus be given as $k_x = \gamma \cdot G_x \cdot t$ and $k_y = \gamma \cdot G_y \cdot t$ [14].
In order to acquire an MR image, methods need to be developed to traverse the k-space. There are only two possibilities, as seen in Figure 2.6. The first one is to apply

a gradient, in which case the k-space trajectory will be a line defined by the orientation of the gradient (see Figure 2.6a). As long as the gradient is kept constant, the k-space trajectory will be a straight line. The second method for traversing the k-space is the application of a refocusing pulse, which will lead to a jump of the k-space trajectory around the origin (see Figure 2.6b). It is clear that a SE formation alone will not allow coverage of all of the k-space, since the trajectory would only jump between the two mirror symmetric points. A combination of SEs and gradients can produce very efficient k-space trajectories. All of the existing k-space-based imaging techniques are based on some combination of these two basic means of traversing the k-space [14].
Is should be noted that the signals are measured at discrete time intervals called the dwell-time of data acquisition. This discrete sampling leads to an ambiguous assignment of frequencies above a given threshold which is called the Nyquist frequency [14]. The Nyquist frequency therefore determines the acquisition bandwidth inside which the signal should occur. The definition of the k-space coordinates implies that these are invariant with respect to the actual strength of the gradient used, as long as G and t are constant. Mathematically, data acquisition under a strong gradient and in a shorter acquisition time will yield the same k-space data and thus the same image as acquisition under a weaker gradient and a longer acquisition time. A shorter acquisition time and a closer spacing of sampling points are equivalent to a higher-acquisition bandwidth. Since the received noise grows with the square root of the bandwidth, a faster imaging technique will therefore by principle always carry the penalty of a lower SNR. For a conventional SE imaging sequence a typical acquisition time of 5 to 10 ms for each phase-encoding step is used. Acquisition of 192 to 256 phase-encoding steps therefore requires a total net acquisition time of 1 to 2 s. However, this is much shorter than the actual acquisition time of such a sequence, which is determined by the TR defined by the T_1 contrast. It follows that the SNR of a fast imaging sequence leading to acquisition times of approximately 50 ms, will be lowered by at least a factor of 5 to 10, regardless of the actual sequence used. Fast imaging thus relies on radio-frequency coil designs improving the SNR [14].
The discrete data sampling has some additional consequences regarding the sampling density and coverage of data in k-space. Different parts of the k-space encode different features of the image: The center of the k-space hold the lower frequencies which represent the image contrast, whereas the outer parts encode the higher frequencies for sharp structures. Due to the symmetry of the 2D FT, this statement can also be reversed: The image center will be encoded by low-resolution k-space data. Therefore, sampling of sparsely distributed k-space data will reduce the effective field of view of the final image [11, 14].
Mathematically, the final image will look exactly the same, irrespective of the way the data is sampled in the k-space. In practice, however, different approaches can have a major impact on image quality as the data has to sampled sequentially. The observed spins evolve not only as a function of the gradients defining the intended k-space trajectory, but are also influenced by other mechanisms unrelated to image encoding. It is noteworthy that the coordinates in k-space are expressed in units of a phase angle across space. Any mechanism affecting the phase of the signal along this trajectory will

therefore alter the k-space trajectory. Some such commonly occurring mechanisms are flow and motion effects, magnetic field inhomogeneities, as well as susceptibility and chemical shift effects. The consequence of such phase effects in terms of imaging properties is dependent on the particular sequence and data sampling speed. In addition to phase effects, the signal decay with T_2 needs to be taken into account also, especially if the data acquisition time is similar or even longer than the T_2 of the observed tissues under construction [14].

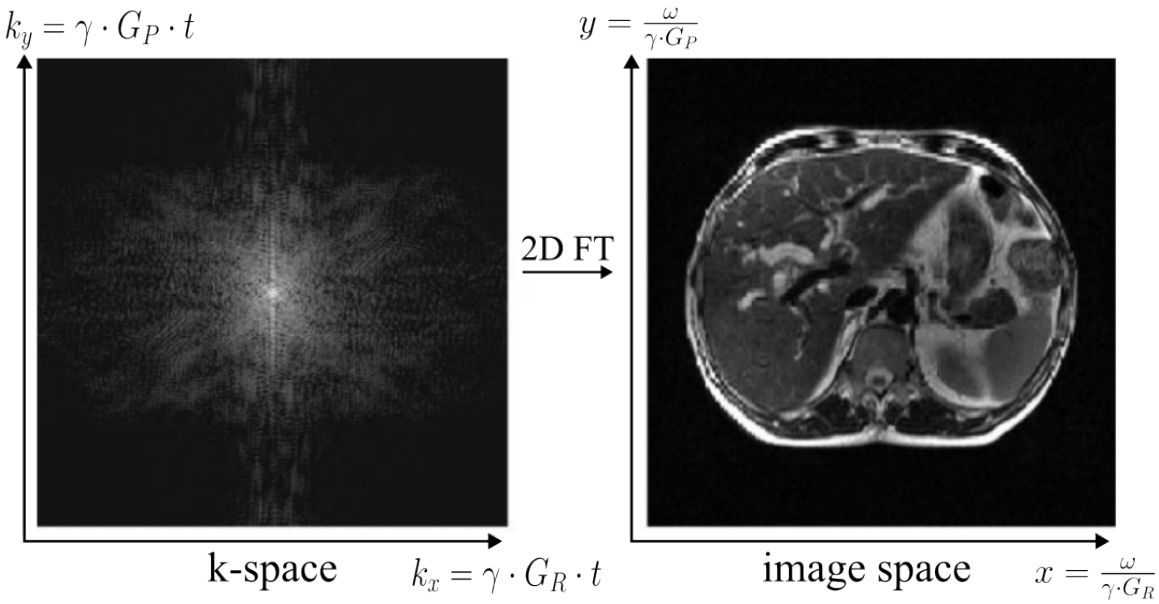
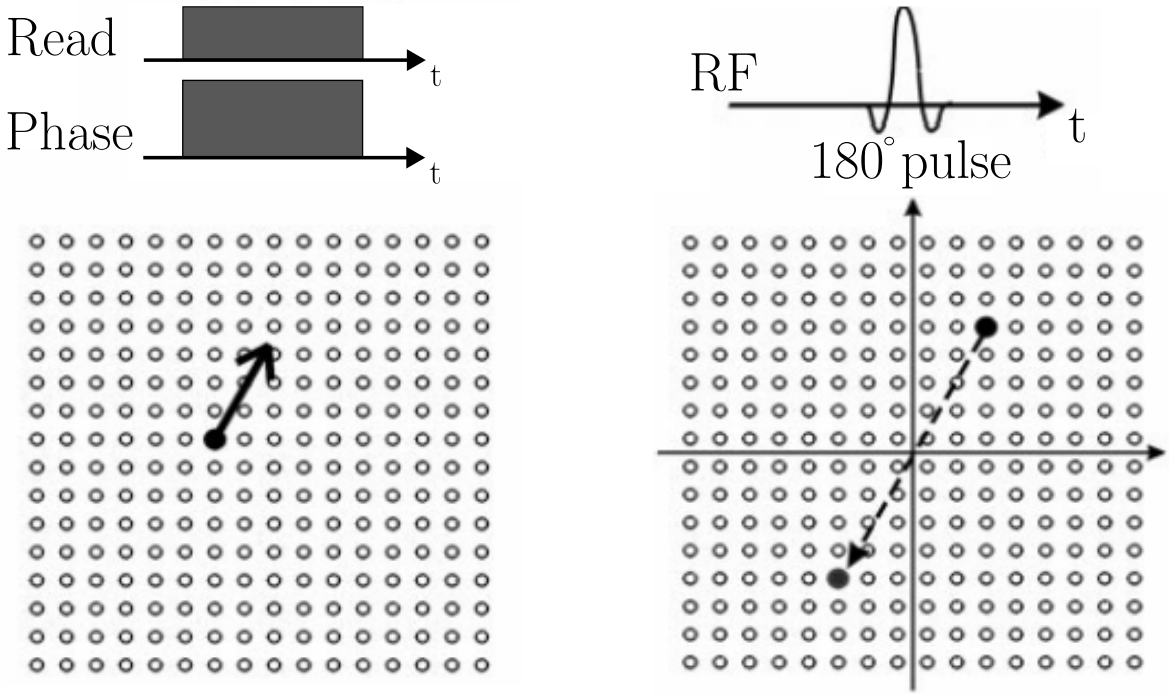


Figure 2.5: Correspondence of the image domain and the k-space via a 2D FT, adapted from [14]. Consequently, a 2D iFT can be used to go from image space into k-space. In practice, a 2D FFT and iFFT are used due to faster computation. The coordinates of each domain are defined by the Larmor frequency according to Equation 2.1.

Rectilinear K-Space Sampling

Almost all MR imaging sequences are based on rectilinear k-space sampling, meaning the sampling points are placed on a rectangular grid. This ia also referred to as Cartesian sampling and allows the usage of the fast Fourier transform (FFT) which enables image reconstruction in a very short time (under 1 s). For comparison, using the 2D FT to transform a non-rectilinear grid can take about 5 to 50 min per image [14]. In a Cartesian acquisition, one or more lines of k-space parallel to a Cartesian axis are collected following application of an RF pulse and generation of an RF echo. Adjacent points in a single line are collected in rapid succession in a single echo. The time between echoes and adjacent lines is much slower. The direction of fast acquisition is known as the frequency direction and the other one or two directions are known as the phase-encode direction or directions. Following line-by-line filling of k-space in a



(a) Schematic of a constant gradient creating a straight k-space trajectory.

(b) Schematic of an refocusing pulse inverting the phase of the k-space.

Figure 2.6: Methods of moving in the k-space using a) constant gradients to move along straight lines or b) mirror the k-space at the origin using a refocusing pulse. Adapted from [14].

Cartesian sequence, each layer of a filled grid is subsequently used to reconstruct a single image slice [50].

Non-Rectilinear K-Space Sampling

A common problem of rectilinear sampling techniques is the very heterogeneous nature of the k-space trajectories used. Data is acquired along straight k-space lines under a constant gradient. Going around the corner of the trajectory requires very fast switching for a brief period. This is very demanding on the gradient power amplifiers, which have to be able to alternate between the two modes. A more common load on the gradients is offered by the use of curved k-space trajectories. Especially spiral trajectories [51, 52] have won considerable interest due to their very efficient use of the gradient system. A practical problem relates to image reconstruction which requires algorithms that can take several seconds per image on a typical scanner [14]. The ultimate success of spiral imaging will then be determined by its inherent imaging properties, which are significantly different from rectilinear scans. Constant off-resonance effects, such as chemical shift, field inhomogeneity, and susceptibility, will not lead to a displacement in the phase-encoding direction, but to blurring of the corresponding structures [14]. A problem of spiral imaging is the fact that severe artifacts can arise when the k-space

trajectory produced by the gradient system is not exactly identical to the trajectory used for image reconstruction. It is thus necessary to accurately calibrate the gradient system or to even measure the actual trajectory used. A favorable property of spirals is the motion correction inherent to the trajectory [53]. Apart from spirals, other non-rectilinear k-space trajectories have been suggested [54, 55]. Back projection uses a star-like trajectory and allows the realization of extremely short echo times. Its inhomogeneous k-space coverage with a high sampling density for points at the center of k-space and its somewhat awkward artifact behavior have limited its success for conventional imaging [14].
Other trajectories, such as rosettes or even random k-space trajectories, have also been explored. A common feature of all non-rectilinear trajectories relates to their non-periodic nature, which produces severe artifacts when the field of view of the data acquisition is less than the size of the object. For rectilinear scans this leads to severe fold-over artifact, which can be tolerated as long as it does not affect relevant structures or even is totally avoided when it occurs in the readout direction, where oversampling can be applied without penalty in the data acquisition time. For non-rectilinear scans oversampling is not an option, since the field of view is determined by the distance between adjacent parts of the trajectory, rather than by the sampling rate along the trajectory. For an insufficient field of view of the k-space data, misregistration artifacts can be very severe, depending on the trajectory used. Spirals produce a circular rim artifact around the image. It is therefore mandatory to ensure that the k-space trajectory is sufficiently dense to always cover the whole object of interest [14].

2.1.3 Imaging Acceleration and Reconstruction

To alleviate the slow image acquisition times of the MR imaging, different MRI acceleration techniques have been proposed. These cause strong image artifacts when using a simple iFT for reconstruction. As a consequence more sophisticated approaches for image reconstruction have been developed.

Image Acceleration via k-Space Subsampling

Most acceleration techniques included scanning less k-space lines during signal acquisition, omitting certain frequencies by zero-padding. This is because total acquisition time T_{Ac} for the 2D case is determined by two factors:

$$\mathrm{T}_{Ac} = \mathrm{TR} \cdot \mathrm{N}_{PE}, \tag{2.3}$$

where N_{PE} is the number of phase encoding lines. The TR determines the contrast in the image while N_{PE} determines the resolution [36]. To reduce the acquisition time, either the k-space data must be collected more quickly (reducing TR) or the amount of collected k-space data must be decreased (reducing N_{PE}). The speed at which k-space data can be collected is determined by the desired image contrast and the strength of the magnetic field gradients needed to encode the k-space data [36]. For some sequences like SE, the TR must be left long in order to generate the desired

contrast. For others, it is possible to reduce the TR while still maintaining image contrast. However, in these cases the electrical power required to run the magnetic field gradients faster would be massive [36]. There is also a physiological limit as rapidly switching high-strength magnetic field gradients on and off can induce electrical currents in the patient, potentially causing peripheral nerve stimulation [56, 57]. In addition, hardware limitations also restrict the TR, like maximum gradient amplitude and slew rate [11].
Instead of shortening TR, another option is to reduce the number of k-space lines that are collected. The amount of k-space lines that are left out and zero-filled is usually given by a reduction factor R, defined as the ratio between the number of k-space points in the fully-sampled data compared to the subsampled data [11, 36]. While there are different subsampling techniques, most involve fully sampling a center region containing the lower frequencies and dropping higher frequencies. These are deemed less important as they contain mostly details and fine image structures, whereas the low frequencies encode the general image structure and contrast [14]. All of these methods create artifacts during image reconstruction, like image blur and aliasing, as seen in Figure 2.7. The more lines are missing/zeroed, the more intense these artifacts become. In order to still reconstruct images with as few artifacts as possible, different reconstruction algorithms have been developed such as parallel imaging and compressed sensing, which are discussed in the following sections.
Recently, using neural network to find better k-space subsampling patterns has been proposed. These try to learn new subsampling strategies in a data-driven manner like pruning unimportant k-space frequencies [58] as well as deep learning based radial [38] and other non-Cartesian [15] subsampling for MRI acceleration. These promise better image quality and easier reconstruction for the same amount of deleted k-space data. However, these have yet to be used in a clinical application [58].

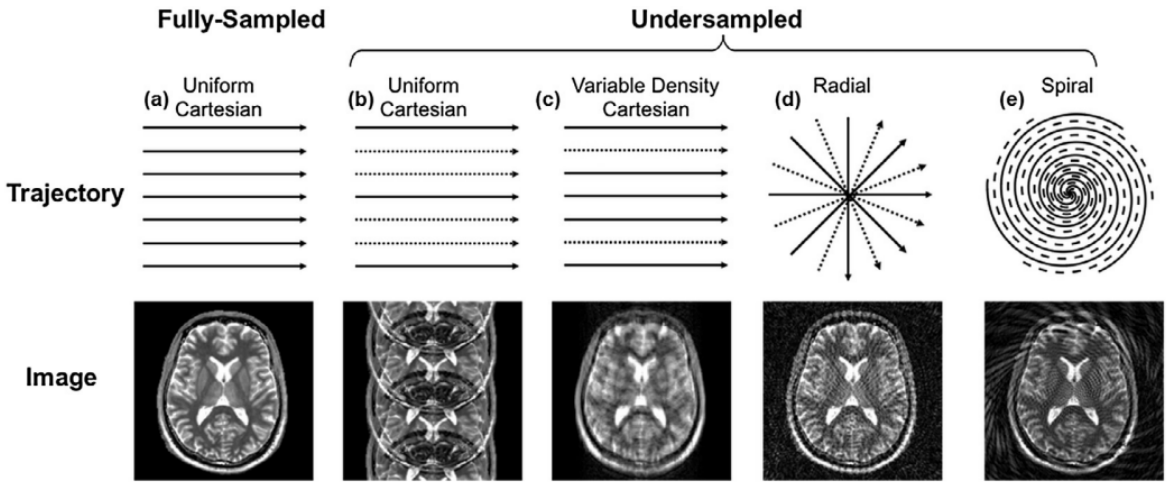


Figure 2.7: Examples of different subsampled k-space trajectories (top row) and their associated aliasing artifacts in image space (bottom row) taken from [11]. Omitted/zeroed data represented by dashed lines.

Parallel Imaging

Parallel imaging (PI) is a robust method for accelerating the acquisition of MRI data by subsampling the k-space data while still being able to reconstruct high quality images using an array of receiver coils. One of several PI algorithms can then be used to reconstruct artifact-free images from either the aliased images (SENSE-type reconstruction) or from the subsampled k-space data (GRAPPA-type reconstruction) [11]. PI requires special hardware known as phased array coils that contain multiple independent receiver channels. Each coil element is most sensitive to the magnetization closest to it and less sensitive to magnetization further away. Individually, each coil has high local SNR but inhomogeneous coverage. To expand the FOV while maintaining high SNR, several coils are organized in an array. The images from each channel are usually combined into a single image with relatively homogeneous intensity by taking a root sum-of-squares (RSS) combination [15]. Theoretically, the maximum acceleration factor is limited by the number of coils, thus, for PI to be successful, each coil must have a unique sensitivity variation along the direction that is accelerated. For example, an array with four coils arranged in a line may be able to attain a maximum R = 4 along one direction, but no acceleration would be possible in the perpendicular direction [11].
As mentioned before, PI techniques fall into one of two classes, depending on whether aliased pixels are separated in the image domain (SENSE) or missing phase encoding lines are reconstructed in k-space (GRAPPA). Sensitivity encoding (SENSE) [59] is a PI technique which unfolds superimposed pixels in the image domain. For Cartesian k-space sampling with a uniform acceleration factor of R = 3 and a receiver array with four coils the undersampling reduces the FOV threefold, such that three pixels from the fully-sampled image fold onto the same pixel in the aliased image [11]. SENSE uses prior knowledge of the coil sensitivity profiles to separate folded pixels and recover the full FOV image. If the acceleration factor exceeds the number of coils, then the SENSE algorithm will not be able to recover an un-aliased image [36]. In practice, the largest achievable acceleration factor is usually smaller than the theoretical limit because of coil sensitivity overlap and coil not being orthogonal [11].
One common feature of PI is the amplification of noise in the reconstructed image, though with varying degrees:

$$\mathrm{SNR}_{\mathrm{SENSE}}(x,y) = \frac{\mathrm{SNR}_{\mathrm{full}}(x,y)}{\sqrt{R}\cdot g(x,y)}, \tag{2.4}$$

where the geometry factor $g(x,y)$ describes the spatial pattern of noise enhancement, $\mathrm{SNR}_{\mathrm{SENSE}}(x,y)$ the SNR in the reconstructed SENSE image and $\mathrm{SNR}_{\mathrm{full}}(x,y)$ the SNR of the fully sampled image [11]. In addition to the g-factor losses, which depend on variables like number of coils, array configuration, etc., the SNR decreases with the square root of R, which is known as Fourier averaging [11].
While SENSE unfolds aliased signals in the image domain, generalized partially parallel acquisitions (GRAPPA) [37] synthesizes missing data points directly in k-space [36]. There, the use of inhomogeneous receiver coils effectively spreads information from one k-space point to nearby k-space points. GRAPPA exploits these k-space redundancies

across coils to reconstruct missing k-space data using neighboring acquired points. A single missing k-space data point (target point) is synthesized as a linear combination of acquired neighboring k-space points (source points) with the spatial arrangement of source and target points being called GRAPPA kernel [36]. Each acquired source point is multiplied by a coefficient (GRAPPA weight) and the results are added to estimate the target point. A single target point for one coil is reconstructed using source points from all other coils. For Cartesian sampling, the weights are shift invariant to a first approximation, so the same GRAPPA weights can be applied throughout k-space. The reconstruction can be described as convolving or sliding the GRAPPA kernel throughout the k-space [11].
While SENSE uses additional information in the form of coil sensitivity profiles to unfold aliased pixels, GRAPPA requires extra data to estimate the weights. GRAPPA is considered to be autocalibrating because several additional phase encoding lines, called the auto-calibration signal (ACS), are collected near the k-space origin for calculating the weights [36]. These lines can be used as a reference for calibration as the center is usually fully sampled. The SNR of images reconstructed using GRAPPA is also reduced according to Equation 2.4, however, because GRAPPA does not require an explicit estimate of the coil sensitivities, it tends to be more robust than SENSE to inconsistencies between the calibration and subsampled data [11].
SENSE and GRAPPA can be combined in what is called iterative self-consistent parallel imaging (SPIRiT) [60]. Like GRAPPA k-space kernels are used to recover missing information by exploiting correlations between neighboring k-space points, though the reconstruction is framed as an inverse problem similar to SENSE. Regardless of the original sampling trajectory, SPIRiT outputs a Cartesian k-space. The reconstruction is typically initialized with the subsampled zero-filled k-space and is solved iteratively. The algorithm minimizes and balances the errors of two terms: calibration consistency and data consistency [11]. The first is calculated using a so-called SPIRiT kernel in the k-space similar to GRAPPA, while the second term enforces consistency with the subsampled data as the difference between the reconstructed data and acquired data should be zero at the originally sampled positions. Thus reconstruction should only recover missing k-space points without changing the known data points [11]. ESPIRiT [61] is an extension of SPIRiT that uses k-space kernel operations to derive a set of eigenvector maps that behave like coil sensitivities, which can be incorporated in a generalized SENSE reconstruction. This requires calibration data from the fully sampled region at the center of k-space. Unlike SENSE and GRAPPA, SPIRiT and ESPIRiT reconstruct images iteratively and can require long computation times which can be addressed by using e.g. parallelized GPUs [11].

Compressed Sensing

The idea behind compressed sensing (CS) is that sparse or compressible signals can be acquired in an efficient way by applying compression already in the data acquisition process. According to CS theory, sparse or compressible signals can be recovered from fewer samples than required by the Shannon-Nyquist sampling theorem [35].

This is achieved by applying an appropriate sampling scheme and reconstruction that employs signal sparsity to recover the signal. MRI fulfills two important requirements for the application of CS: medical imaging is naturally compressible by sparse coding and MRI scanners acquire samples of the encoded image in spatial frequency, rather than direct pixel values. CS emerged as an abstract mathematical idea that if one measures a relatively small number of random linear combinations of the signal values the signal can be reconstructed with good accuracy from these few measurements by a non-linear procedure due to the underlying signal being compressible. In MRI, the sampled linear combinations are the individual Fourier coefficients (k-space samples) and CS is able to make accurate reconstructions from a small subset of k-space, rather than an entire k-space grid [35]. It should be noted that unlike PI approaches, the k-space is incoherently sampled; thus, noise-like artefacts appear in the image when a direct inverse Fourier transform is performed. To remove the artefacts caused by the undersampling, iterative reconstruction is used. There are a number of sparsifying transforms that can be used for CS such as wavelets, which are very common, and total variation, which enforces the sparsity of the image gradients. The advantages of total variation include its simplicity, rotation invariance, and capability of preserving edges and providing good image quality [13]. There are also several methods combining CS and PI trying to achieve higher acceleration factors [62, 63].

2.1.4 Motion-Compensated Image Reconstruction

Patient motion during acquisition is one of the major impediments of high-quality MRI scans. This is especially true for thoracic and abdominal imaging, as organs move during breathing. Such motion artifacts can be compensated either during or after reconstruction using image registration.

Motion in MRI Acquisition

Due to the long acquisition times, motion is one of the major extrinsic factors influencing MR image quality. Patient and physiological motion induces aliasing along the phase-encoding direction and blurring of the image content [7]. This motion can induce several consequences on MR signal formation. Intraview and interview motion have to be distinguished between: motion is intraview when occurring during individual MR experiments (between RF excitation and echo formation), whereas motion is interview when occurring between individual MR experiments. Whenever the period of motion is slow compared to the period of MR acquisition TR, the assumption can be made that motion is interview. This is often a reasonable assumption when considering pseudo-periodic motion induced by respiration, and also possibly by cardiac contraction, which are the two common sources of motion in cardiac and abdominal imaging (typically, the adult respiratory period is about 4 to 5 s, and TR $\approx$ 10 ms for fast imaging) [20]. Interview motion results in spatial encoding inconsistencies, and thus in image deterioration which can take complex forms (blurring/ghosting artifacts) as acquisition is performed in k-space. Several strategies can be employed in order to handle patient motion better. Patient cooperation is the most commonly used

method [20]. However, BHs cannot last much longer than 20 s and physiological drifts cannot be completely avoided. This leads to a limitation on the time-period of signal recording and thus SNR. Moreover, the position of organs in successive BHs may not be reproducible. Synchronization techniques are well-established and systematically used in clinical protocols, but they require a high-level of motion reproducibility. This is often a limiting factor considering heart rate variability (whether in free breathing or during a BH), and respiratory variability in terms of amplitude and frequency [20].

Reconstruction Pipelines

Motion-resolved data acquisition is usually accelerated by PI or CS techniques yielding subsampled k-space data. In order to reconstruct aliasing-free images, these methods rely on reconstruction schemes that incorporate sparsity or low-rank constraints to solve the ill-posed problem [35, 36, 64]. Fixed sparsity assumptions in CS are often too restrictive and incapable of fully modeling spatio-temporal dynamics [7]. After reconstruction, non-rigid motion fields can be estimated in image space from reconstructed images by solving a registration problem. A particular interest and challenge lies in the derivation of reliable motion fields which capture the spatio-temporal non-rigid deformations, such as respiratory or cardiac movement. Instead of performing these two steps sequentially, motion-compensated image reconstruction schemes like *GRICS* [20] integrate both motion field estimation and motion modeling into the reconstruction process. These methods require reliable motion-resolved images from which the motion fields can be estimated. Motion field estimation can be controlled or supported by external motion surrogate signals [65], initial motion field estimates [66, 67], from motion-aliased images [68] or low-frequency image contents [69]. Moreover, spatio-temporal redundancies can be exploited to achieve an aliasing-free image. While these methods have been proven to be more robust against registration errors, they can require a significantly increased computational demand and/or limit imaging acceleration [7].
There are two major parts of a motion-compensated reconstruction pipeline: a non-rigid registration model to reliably estimate the motion fields between frames and an unrolled reconstruction model that reconstructs the motion-corrected frames. Both of these can use either conventional iterative algorithms like in GRICS [20] or neural networks that learn the given task [7].

2.2 Image Transformations and Registration

In this chapter, a brief overview over different kinds of image transformations is given, followed by an introduction into the challenges of image registration.
The goal of image registration is to compute a transformation that aligns two images - moving and fixed. This transformation is then applied to the moving image to create an warped image that more closely resembles the fixed image. But first, the different kinds of image transformations must be discussed.

2.2.1 Image Transformations

In medical image registration, there are three basic transformation types that are typically used: rigid, affine and non-linear (deformable) [70].

Rigid Transformations

Rigid transformations are linear and global transformations that affect the whole image. As these are global operations, one can express them as matrix and vector operations. A rigid transformation includes translation and rotation and can be represented by:

$$\mathbf{T}_{rigid}(x) = \mathbf{R} \cdot \overrightarrow{p} + \overrightarrow{t}, \tag{2.5}$$

with $\mathbf{R}$ being the rotation matrix, $\overrightarrow{p}$ a point in the image and $\overrightarrow{t}$ the translation vector. For 3D images, the rotation matrix and the translation vector require three parameters each. Thus, six parameters have to be calculated for a rigid transformation for 3D images [70].

Affine Transformations

Affine transformations are also linear and global transformations, however they include translation, rotation, scaling and shearing. Matrix multiplication can be used to merge all of these individual transformations into a single matrix:

$$\overrightarrow{p}' = \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{t} \cdot \overrightarrow{p} \tag{2.6}$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.7}$$

$$= \begin{bmatrix} s \cdot \cos(\theta) & -s \cdot \sin(\theta) & t_x \\ s \cdot \sin(\theta) & s \cdot \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, \tag{2.8}$$

with θ being the angle of the rotation for the rotation matrix $\mathbf{R}$, s the scaling factor for the scaling matrix $\mathbf{S}$ as well as t_x and t_y being the translations in x- and y-direction. This can be further generalized as a single matrix multiplication:

$$\overrightarrow{p}' = \mathbf{T}_{affine} \cdot \overrightarrow{p} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, \tag{2.9}$$

with $\mathbf{T}_{affine}$ being the general affine transformation matrix [70].

Non-Linear Transformations

Non-linear or deformable transformations are used to model local deformations in images that rigid and affine transformations cannot capture as these transformations are capable of locally warping the image. Non-linear transformations include radial

basis functions, physical continuum models, and large deformation models like diffeomorphisms. These transformations are complex and do not preserve straightness or parallelism:

$$\mathbf{T}_{deformable}(x) = x + \phi(x), \tag{2.10}$$

with ϕ being the deformation or displacement field. Deformable image registration is an ill-posed problem, often requiring regularization to ensure a smooth and plausible displacement field [70].

2.2.2 Image Registration

Image registration is a challenging, yet important task for image processing. In the field of medical image analysis, image registration remains one of the main research topics and challenges [19]. This task consists of transforming a moving image to match a fixed image [23]. In the medical field this can be used for clinical applications such as disease diagnosis and monitoring, image-guided treatment delivery, and post-operative assessment [19]. Medical image registration is typically used to pre-process data for tasks like object detection and segmentation. Thus, the performance of these methods is dependent on the quality of image registration [5].
Medical image registration was historically done manually by medical professionals, however, the quality of manual alignments is dependent on the expertise of the clinicians [10]. These manual registrations are thus not only time consuming, but also hardly reproducible leading to high interobserver-variability. While the need for automatic registration is very much apparent, the computational cost of traditional registration algorithms prohibited their usage. With the rise of deep learning, neural networks gained popularity and now provide an alternative to conventional algorithms and manual registration that is fast during execution [10]. However, it should be noted that these networks still need a lot of data and computational power to be trained.
In pair-wise image registration, two images, called moving (M) and fixed (F), are aligned with a spatial transformation T. As discussed in section 2.2.1, there are three types of transformations: rigid, affine, and non-linear (deformable). While the latter is the most difficult to compute, these are also the ones most common in clinical practice [29]. Additionally, deformable image registration can also be utilized for various computer-assisted interventions like biopsy [71] and (MRI-guided) radiotherapy [72, 73]. Registration can be described as an optimization problem:

$$\mathrm{T}' = \underset{\mathrm{T}}{\operatorname{argmax}}\, S(F, \mathrm{T}(M)), \tag{2.11}$$

where T' is the best transformation maximizing the similarity S between the two images. This can be done iteratively, continuously improving the estimates for the desired T, maximizing the similarity [5]. Intuitively, deformable image registration is an ill-posed problem, making it fundamentally different from other computer vision tasks such as object localization, segmentation or classification [26]. Given two images, deformable image registration aims to find a spatial transformation that warps the moving image to match the fixed image as closely as possible. However, there is no

ground-truth available and without enforcing any constraints on the properties of the spatial transformation, the resulting cost function is ill-conditioned and highly non-convex [5]. In order to address the latter and ensure tractability, all image registration algorithms regularize the estimated displacement, based on some prior assumptions on the properties of the underlying unknown deformation [5].
While there are many registration algorithms that are proven to work, they lack computation speed, which can be alleviated by new deep learning approaches [26].

2.3 Deep Learning

Deep learning, and neural networks in general, have seen a significant rise in usage over the last couple of years due to many breakthroughs in areas of image recognition [74], segmentation [6] and also registration [30]. After discussing different network architectures, specific use-cases for image registration are discussed and a brief overview of network training and testing, including evaluation metrics, is given.

2.3.1 Deep Learning Architectures

Neural networks, despite their theoretical foundations being around for decades, have seen a drastic rise in popularity over the last few years as constraints on computational power and data size have been alleviated [5]. Especially deep neural networks have become more popular and are often summarized under the term deep learning. In recent years the development and use of deep neural networks has grown substantially, sustained by rapid improvements in computational hardware like GPUs [19]. Consequently, clinical applications requiring image classification, segmentation, registration, or object detection/localization, have witnessed significant improvements in algorithmic performance, in terms of accuracy and/or efficiency [5]. The following network architectures are widely used for different tasks including medical image registration.

Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of neural networks with regularized multi-layer perceptrons, which are mainly used for image processing [75]. CNNs use convolution operations instead of general matrix multiplications in typical neural networks [76, 77]. These convolutional filters make CNNs very suitable for visual signal processing. Because of their excellent feature extraction capabilities in combination with further operations, CNNs are some of the most successful architectures for image analysis. Different variants have been proposed achieving state-of-art performances for various image processing tasks [77]. A typical CNN usually consists of multiple convolutional layers, max pooling layers, batch normalization layers, optional dropout layers, a sigmoid or softmax layer. In each convolutional layer, multiple channels of feature maps are extracted by sliding trainable convolutional kernels across the input feature maps. Hierarchical features with high-level abstraction are extracted using multiple convolutional layers. These feature maps are passed through multiple fully connected

layer before reaching the final decision layer. Max pooling is often used to reduce the image size and promote spatial invariance [5]. Batch normalization is used to reduce internal covariate shift among the training samples. Weight regularization and dropout layers are used to prevent data overfitting [26]. The loss is often defined as the difference between predicted and target output. CNNs are usually trained by minimizing this loss via gradient back propagation using optimization methods like Adam [78].

U-Net

The U-Net architecture [6] is a CNN variant first used for image segmentation, but has been proven to also be effective for image registration tasks [79]. It adopts symmetrical contractive and expansive paths with skip connections between them. The encoding blocks extract important features from the image using convolution layers and max pooling, which are then stored in a latent space between the paths. From there the image is reconstructed using upsampling and (de-)convolutions in the decoding blocks. Additionally, skip connections are used to improve the spatial resolution [6]. The U-Net enables effective feature learning from a small number of training data [26].

Autoencoders

An autoencoder (AE) is a CNN that learns to reconstruct an image from its input without supervision. AEs usually consists of an encoder which extracts the input features, which are stored in a low-dimensional latent space, similar to a U-Net, and a decoder which restore the original input from the latent space. To prevent an AE from learning an identity function, regularized autoencoders were invented, which can be used for e.g. denoising AEs. Variational AEs (VAEs) are generative models that learn latent representation using a variational approach, which constrains the variability of the outputs. VAEs can been used for anomaly detection and image generation [26].

2.3.2 Deep Learning for Image Registration

Recently, there has been a surge in the use of deep learning based approaches for medical image registration [19]. Their success is largely due to their ability to perform fast inference, and the flexibility to leverage auxiliary information such as anatomical masks as part of the training process [26]. The most effective methods, such as *VoxelMorph* [30], typically employ a U-Net architecture to estimate dense spatial deformation fields. These methods require only one forward pass during inference, making them orders of magnitude faster than traditional iterative methods [26]. Following the success of *VoxelMorph*, numerous deep neural networks have been proposed for various registration tasks [34]. Other approaches also utilize e.g. CNNs and AEs [19].

Supervised Registration

Supervised registration describes training a network with a ground truth displacement field that is either real (created by a medical professional) or synthetic (generated via

registration algorithms). Thus, the loss can easily be calculated as the difference in the displacements of the network prediction and ground truth (see Figure 2.8). These methods have achieved notable results with real displacement fields as supervision [29]. However, this approach is limited by the size and the diversity of the dataset. As the displacements are often calculated by conventional algorithms their effectiveness might be limited for difficult problems with which the traditional algorithms struggle. Fully supervised methods are widely studied and have notable results, but the generation of real or synthetic displacement fields is hard, and these displacements fields might be different from the real ground truth, which can impact the accuracy and efficiency of these kinds of methods [29]. Notable approaches include *BIRNet* [80] and *LAPNet* [27], however the latter works in the k-space domain.

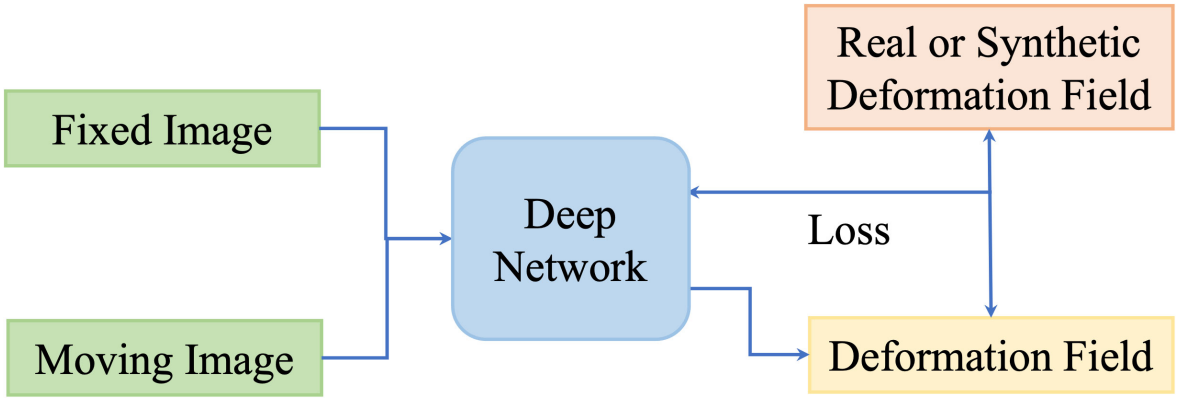


Figure 2.8: Schematic of the training process for a supervised network calculating the loss directly to a ground truth displacement, taken from [29].

Unsupervised Registration

As the preparation of ground truth displacements for supervised methods is difficult and time consuming, limitations in generalizing results for various domains and registration tasks are inevitable [19]. Thus, unsupervised registration has a more convenient training process with paired images as inputs, but without a ground truth to compare against. Typically, the loss function computes the similarity between the aligned images (see Figure 2.9) as well as the smoothness of the displacement field, rather than the difference to a ground truth [29]. Example are the *IC-Net* [31], *VoxelMorph* [30], *TransMorph* [81] and *SYMNet* [32].

2.3.3 Network Training and Testing

In order to train any neural network, a large amount of data is needed. This data can be videos, images, audio, text and many more. However, the data is usually divided into three subsets: the training, validation and test subset.

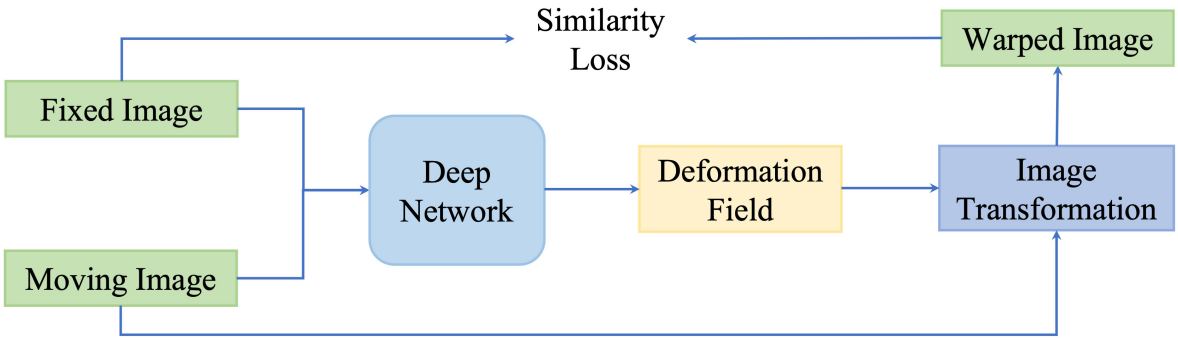


Figure 2.9: Schematic of the training process for an unsupervised network with only a image similarity loss (no ground truth), taken from [29].

Training and Back-Propagation

The first subset is used for the training of the network. It is often the largest subset and should include a lot of variance for the network to learn. Neural networks in general learn by adapting their weights to perform better at a specific task on a specific dataset/domain. This is done in a process called back-propagation, where the loss calculated between the current output of the model. This is acquired by a simple forward pass through the network, and the desired output is propagated backwards through the model and the weights are optimized using e.g. gradient decent [77]. The second subset is also used during training, but is not learned as no back-propagation is performed. This is done to validate the networks training progress and to prevent overfitting which is a common problem, especially for larger networks. This occurs when a network learns every example from the training set, but not general properties as intended. To prevent this, the unseen validation data can be repeatedly used, as it is not learned, to see whether the network improves on unseen data. Once it has been trained for a long time and the validation performance stagnates, or drops, the training is stopped. This is known as early stopping and is a very common strategy to prevent overfitting while optimizing training time [76].

Testing and Evaluation Metrics

The last subset is the test set, which is used for the final evaluation of the network. It can be used for evaluating performance, as well as memory consumption and inference/execution time. This set cannot be used for training and needs to be completely unseen by the network. For datasets with a lot of patients, a single patient should be withheld in order to truly test on unseen data. Different metrics can be used for evaluation of a networks registration performance including metrics computed on the images and on the generated displacements. The mean squared error (MSE) [5] is calculated between a fixed (ground truth) image and the warped image giving a pixel-wise comparison:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} |F(x,y) - W(x,y)|^2, \tag{2.12}$$

where F is the fixed image and W is the warped image with N being the number of pixels in the images. The lower the MSE the higher the similarity with 0 being a perfect match (the images are exactly the same). The MSE can also be used to calculate the peak SNR (PSNR) [8], which describes the ratio between the maximum possible power of a signal and the power of corrupting noise. It is defined as:

$$\text{PSNR} = 20 \cdot \log_{10}\left(\frac{2^B - 1}{\sqrt{\text{MSE}}}\right), \tag{2.13}$$

with B being the bit depth of the image and the metric being defined in decibel (dB) due to the logarithmic scale. Another common image metric is the structural similarity index measure (SSIM) [82], which ranges from 1 (complete similarity) to 0 (no similarity). It tries to evaluate the general similarity instead of a pixel-wise comparison like the MSE making it more robust against contrast changes:

$$\text{SSIM} = \frac{(2\mu_F\mu_W + c_1) \cdot (2\sigma_{FW} + c_2)}{(\mu_F^2 + \mu_W^2 + c_1) \cdot (\sigma_F^2 + \sigma_W^2 + c_2)}, \tag{2.14}$$

with μ_F, μ_W being the mean values of the images F and W; σ_F^2, σ_W^2 the variances of F and W as well as σ_{FW} the covariance for F and W, with c_1, c_2 being constants derived from the dynamic range of the images. For comparison of segmentations the Dice score [75] is a commonly used metric to estimate the similarity of two segmentations. A score of 1 indicates a complete overlap/match and a score of 0 indicates no overlapping of the segmentations. It is calculated as follows:

$$\text{Dice} = \frac{2|M_F \cap M_W|}{|M_F| + |M_W|}, \tag{2.15}$$

with M_F and M_W being segmentations corresponding to F and W. M_W is obtained by warping the manual segmentation M_F [23]. Aside from image similarity and the evaluation of segmentations, the displacement itself can be evaluated. This is based on the assumption that the displacement should be smooth as folding could result in unrealistic anatomic structures, thus indicating errors. Jacobian matrices are the derivatives of the displacement field ϕ that form a second order tensor field:

$$J_\phi(p) = \nabla\phi(p) = \begin{pmatrix} \frac{\partial\phi_1(p)}{\partial x_1} & \frac{\partial\phi_1(p)}{\partial x_2} & \frac{\partial\phi_1(p)}{\partial x_3} \\ \frac{\partial\phi_2(p)}{\partial x_1} & \frac{\partial\phi_2(p)}{\partial x_2} & \frac{\partial\phi_2(p)}{\partial x_3} \\ \frac{\partial\phi_3(p)}{\partial x_1} & \frac{\partial\phi_3(p)}{\partial x_2} & \frac{\partial\phi_3(p)}{\partial x_3} \end{pmatrix}, \tag{2.16}$$

encoding the local stretching, shearing and rotating of the displacement field for a point p. The determinants of such matrices, also called Jacobian determinants $|J_\phi|$, must be positive everywhere to avoid folding as a region of negative determinants would indicate that the one-to-one mapping has been lost [21]. Thus the percentage of non-positive Jacobian determinants of the deformation $\% \, |J_\phi| \leq 0$ can be used to evaluate the quality of the generated displacement field [19].

3

Materials and Methods

In this chapter, the materials and methods at the core of this thesis are discussed. Starting with a section on efficient image registration using neural networks, the problem of efficient registration is explained and the network architectures of *Fourier-Net* and *Fourier-Net+* are presented in detail. After this, the pipeline used for the motion-compensated reconstruction is explained followed by the datasets used in this thesis. Lastly, the experiments conducted in this thesis are described. These include ablation studies, parameter tests, tests for registration performance and the evaluation of a motion-compensated reconstruction pipeline for cardiac MR data.

3.1 Efficient Registration using Neural Networks

As a starting point, *Fourier-Net* [33] and its successor *Fourier-Net+* [34] were used. These networks, which are explained in the following pages, enable fast and accurate registration while needing less resources compared to similar approaches. These attributes are very beneficial for a potential application like motion correction because the current networks, e.g. *LAPNet*, are usually supervised and require large computational resources.

3.1.1 Problem Formulation

While there are many different traditional iterative algorithms that are proven to work [21, 22, 23, 24, 25] these all suffer from heavy computational requirements and long registration times [19] as discussed previously in section 1.2. While neural networks like [27, 28] could be trained in a supervised manner using these conventional algorithms as a basis to alleviate the time problems [5], these were still very memory intensive as well as being unwieldy to train [29]. While unsupervised networks like [30, 31, 32] could circumvent the latter problem, memory efficiency remained a difficult problem. All of these networks and algorithms predict a dense displacement, some even use complex values which essentially kills their performance. However, a new type of architecture has recently been proposed [33, 34] that is far more efficient due to predicting only a band-limited displacement, without the need for complex values, while promising similar performance. These networks will be discussed in the next sections.

3.1.2 Fourier-Net

Fourier-Net is a new unsupervised approach that aims to learn a low-dimensional representation of the displacement field in a band-limited Fourier domain instead of the full field in the spatial domain. This band-limited representation is then decoded by a model-driven decoder to a dense, full-resolution displacement field in the spatial domain. The zero-padding is necessary to keep the spatial resolution in image space without adding any further information. This band-limiting allows for fewer parameters and computational operations, resulting in faster inference speeds [33]. The architecture is based on the U-Net, like many deep registration approaches, but replaces the expanding path with a parameter-free model-driven decoder to save memory as mentioned before. The encoder of *Fourier-Net* consists of a CNN, which takes two images (fixed and moving) as inputs. The output is a displacement field that is then converted from the spatial domain into the Fourier domain via a discrete Fourier transformation (DFT). From there, this band-limiting representation is padded with zeros to the full resolution of the original displacement field. The field is then recovered by using the inverse DFT (iDFT) to convert it back into the spatial domain. This displacement field is then used to warp the moving image into the fixed image. Additionally, squaring and scaling layers [83] can be added before warping the image in order to encourage a diffeomorphism in final deformation.

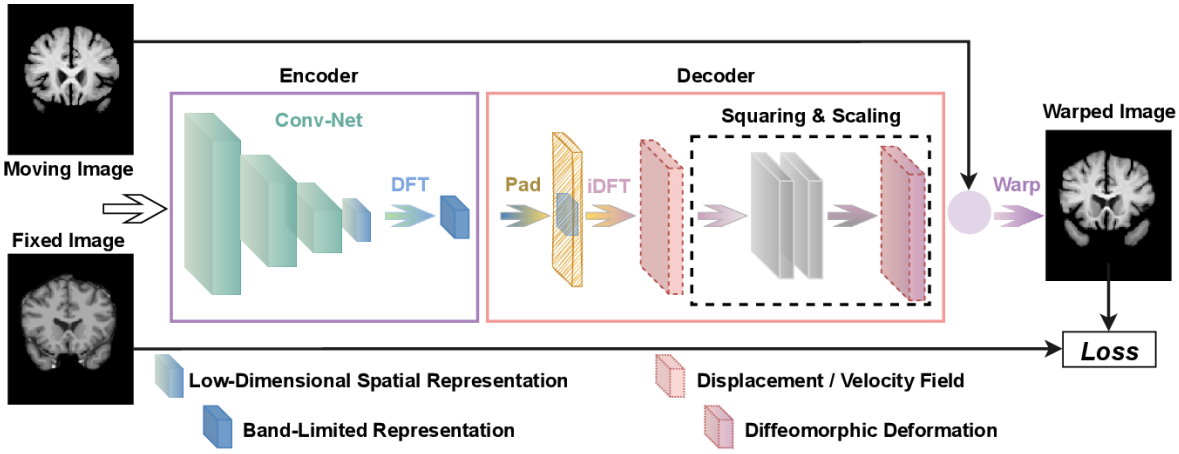


Figure 3.1: Architecture of *Fourier-Net* taken from [33]. The encoder takes two images as an input and produces a band-limited displacement which is then padded and transformed to image space in the decoder. This output is the full-resolution displacement which is then used to warp the moving image for loss calculation based on its similarity to the fixed image.

Encoder

The encoder of *Fourier-Net* consists of a CNN that generates the displacement field between the two inputs followed by a DFT layer that produces a band-limited representation of the full displacement field. The fully convolutional neural network (FCN)

from *SYMNet* [32] was modified to function as the CNN of the encoder. Its (original) architecture (see Figure 3.2) is again based on the U-Net with a contracting and expanding path. The FCN concatenates the inputs images X and Y as a single 2-channel input and estimates two dense, non-linear displacement fields ϕ_{XY} and ϕ_{YX}. However only the displacement field for the moving image, denoted as $\mathbb{S}_\phi$, is needed since we are not interested in transforming the fixed image. This is actually a low dimensional representation because *Fourier-Net* does not utilize the last two levels of the FCNs expanding path that would be needed to reconstruct the actual full-resolution displacement field ϕ.

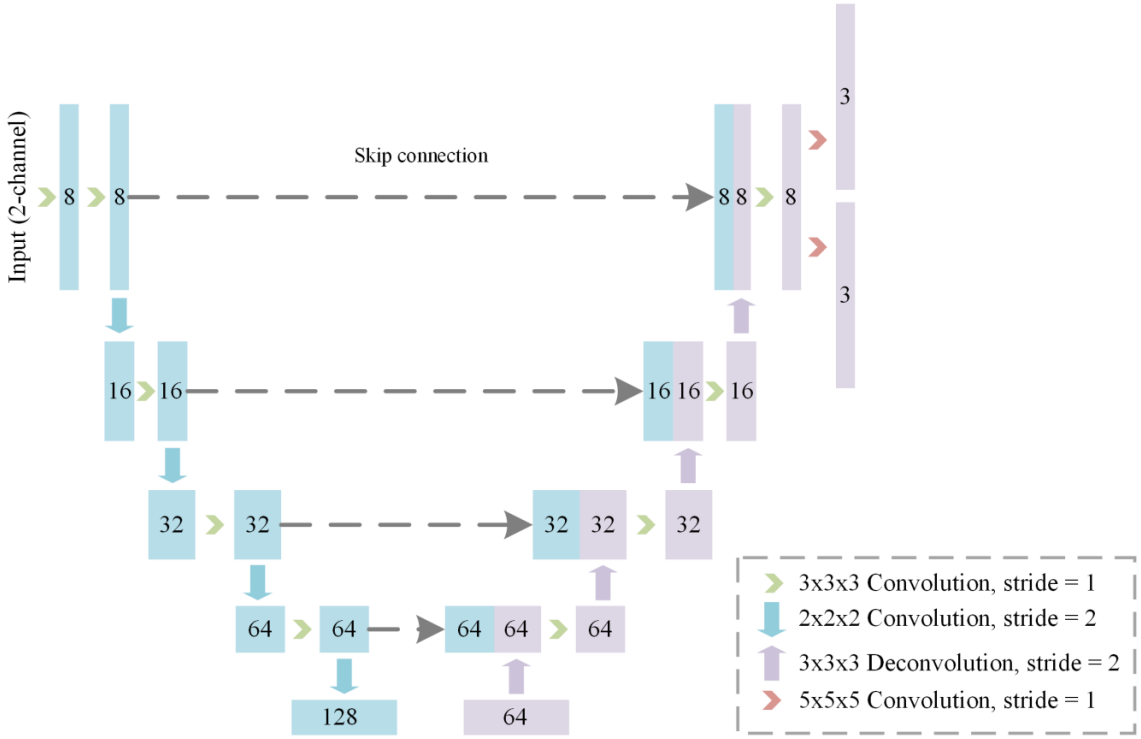


Figure 3.2: Architecture of the FCN from *SYMNet* [32]. The contracting path of the encoder on the left is connected to the expanding path of the decoder on the right via skip connections. The blocks in blue and purple indicate the feature maps from the encoder and decoder respectively.

For each level in the contracting path of the FCN, two successive convolution layers are applied, which contain one $3 \times 3 \times 3$ convolution layer with a stride of 1, followed by a $3 \times 3 \times 3$ convolution layer with a stride of 2 to further compute the high-level features between the input images as well as downsampling the features by half until the lowest level of the network is reached. For each level in the expanding path of the FCN, the feature maps from the contracting path are concatenated through skip connections and apply $3 \times 3 \times 3$ convolution with a stride of 1 and $2 \times 2 \times 2$ de-convolution layer for upsampling the feature maps to twice of its size. At the end of the expanding path, two $5 \times 5 \times 5$ convolution layers with a stride of 1 are appended to the last convolution

layer and generate the displacement fields θ_{XY} and θ_{YX} [32]. Each convolution layer in the FCN is followed by a rectified linear unit (ReLU) activation, except for the output convolution layer that does not have an activation function because *Fourier-Net* only uses the first two levels of the expanding path, thus leading the FCN to generate a low dimensional representation $\mathbb{S}_\phi$ of the full-resolution displacement field ϕ.
As discussed previously, the encoder aims to learn a displacement (or velocity) field in the band-limited Fourier domain. Intuitively, this may require convolutions to be able to handle complex-valued numbers, which can be done by using complex-valued CNNs [41], which are suitable when both input and output contain complex values, however these complex-valued operations sacrifice computational efficiency. Other approaches like *DeepFlash* [28] tackle this problem by converting the input images to the Fourier domain and using two individual real-valued CNNs to learn the real and imaginary parts separately. This, however, increases training and inference cost. To bridge the domain gap between real-valued spatial images and complex-valued band-limited displacement fields without increasing complexity, *Fourier-Net* uses a DFT layer after the FCN. This is a simple and effective way to produce complex-valued band-limited displacement fields without the network needing to be able to handle complex values or learn a mapping between the image and frequency domain. The DFT applied to the displacement field ϕ can be defined as follows:

$$[\mathcal{F}(\phi)]_{k,l} = \sum_{n=0}^{H-1} \sum_{m=0}^{W-1} \phi_{n,m} \cdot \exp\left(i \cdot \left(\frac{2\pi k}{H} \cdot n + \frac{2\pi l}{W} \cdot m\right)\right), \tag{3.1}$$

where ϕ has size $H \times W$ with H and W being the spatial dimensions in 2D. $n \in [0, H-1]$ and $m \in [0, W-1]$ are the discrete indices in the spatial domain, and $k \in [0, H-1]$ and $l \in [0, W-1]$ are the discrete indices in the frequency domain with i being the imaginary unit. However, ϕ in this equation is actually the low dimensional representation of the displacement field output by the modified FCN, which can be formulated as follows:

$$\mathbb{S}_\phi = \mathrm{FCN}(M, F; \Theta), \tag{3.2}$$

with M being the moving and F the fixed image, as well as Θ representing the parameters of the FCN. Thus, the whole encoder can be defined as:

$$\mathbb{B}_\phi = \mathcal{F}(\mathbb{S}_\phi) = \mathcal{F}(\mathrm{FCN}(M, F; \Theta)), \tag{3.3}$$

with the DFT layer $\mathcal{F}$, full-resolution spatial displacement field ϕ and the complex band-limited displacement field $\mathbb{B}_\phi$. The low dimensional representation $\mathbb{S}_\phi$ actually contains all the information of the band-limited Fourier coefficients in $\mathbb{B}_\phi$. As such, *Fourier-Net* does not need to learn the coefficients of $\mathbb{B}_\phi$, but instead only the real-valued coefficients in $\mathbb{S}_\phi$, which is the low dimensional spatial representation of the full-resolution spatial displacement field ϕ, which is then reconstructed by the decoder.

Decoder

The decoder contains no learnable parameters, as the expansive path is replaced with a zero-padding layer, an iDFT layer, and an optional squaring and scaling [84] layer.

The output from the encoder is the band-limited representation $\mathbb{B}_\phi$ in the frequency domain of the low dimensional displacement field $\mathbb{S}_\phi$ in the spatial domain. To recover the full-resolution displacement field ϕ in the spatial domain, we first pad the patch $\mathbb{B}_\phi$, containing mostly low frequency signals, to the original image resolution with zeros. We then feed the zero-padded complex-valued Fourier coefficients, denoted as $\mathcal{F}(\phi)$, to an iDFT layer consisting of two steps: shifting the Fourier coefficients from centers to corners and then applying the iDFT to convert them into the spatial domain:

$$\phi_{n,m} = \frac{1}{HW} \sum_{k=0}^{H-1} \sum_{l=0}^{W-1} \mathcal{D}_{k,l} [\mathcal{F}(\phi)]_{k,l} \cdot \exp\left(i \cdot \left(\frac{2\pi n}{H} \cdot k + \frac{2\pi m}{W} \cdot l \right) \right). \tag{3.4}$$

The $H \times W$ sized sampling mask $\mathcal{D}$ is a low-pass filter that has zeros as entries if they are on the positions of high-frequency signals in ϕ and ones if they are on the low-frequency positions. Thus we can reconstruct the full spatial displacement field ϕ from $\mathbb{B}_\phi$ despite the latter being band-limited. Approaching the problem from the other side we can also think about working backwards from the final displacement. For this, after applying equation (3.1), the low-frequency signals are shifted to a center patch of size $\frac{H}{a} \times \frac{W}{b}$ with $a = 2 \cdot Z_a$ and $b = 2 \cdot Z_b$ where $Z_a, Z_b \in \mathbb{Z}^+$, which is then center-cropped to get $\mathbb{B}_\phi$. This crop of $\mathcal{F}(\phi)$ can be reconstructed using the iDFT from equation (3.4) with the cropping functioning resembling a low-pass filter:

$$[\mathbb{S}_\phi]_{n,m} = \frac{ab}{HW} \sum_{k=1}^{\frac{H}{a}-1} \sum_{l=1}^{\frac{W}{b}-1} [\mathbb{B}_\phi]_{k,l} \cdot \exp\left(i \cdot \left(\frac{2\pi an}{H} \cdot k + \frac{2\pi bm}{W} \cdot l \right) \right), \tag{3.5}$$

with $n \in [0, \frac{H}{a} - 1]$ and $m \in [0, \frac{W}{b} - 1]$ being the indices of the spatial domain, while $k \in [0, \frac{H}{a} - 1]$ and $l \in [0, \frac{W}{b} - 1]$ are the indices of the frequency domain with i being the imaginary unit. Thus $\mathbb{S}_\phi$ actually contains all the necessary information from ϕ, as long as they have the same low-frequency coefficients $\mathbb{B}_\phi$. This can be formulated as:

$$[\mathbb{S}_\phi]_{n,m} = ab \cdot \phi_{an,bm}, \tag{3.6}$$

because most entries of $\mathcal{F}(\phi)$ are zeros, and the remaining values are exactly the same as in $\mathbb{B}_\phi$, which means that $\mathbb{S}_\phi$ contains all the information ϕ can provide. For this, however, $\mathbb{B}_\phi$ needs to be padded to the original image size first in order to get the full-resolution displacement and not a low dimensional representation. This ultimately shows that there is a unique mapping between $\mathbb{S}_\phi$ and ϕ, which means that it is reasonable to use a network to learn $\mathbb{S}_\phi$ directly from image pairs and then reconstruct the displacement field in a very efficient manner [33]. The complete reconstruction process is visualized in Figure 3.3.

As both padding and iDFT layers are differentiable, *Fourier-Net* can be optimized via back-propagation. *Fourier-Net* can use extra squaring and squaring layers [84, 83] in the decoder to turn the displacement field into a stationary velocity field. Typically seven scaling and squaring layers are used to impose such a diffeomorphism [33, 83].

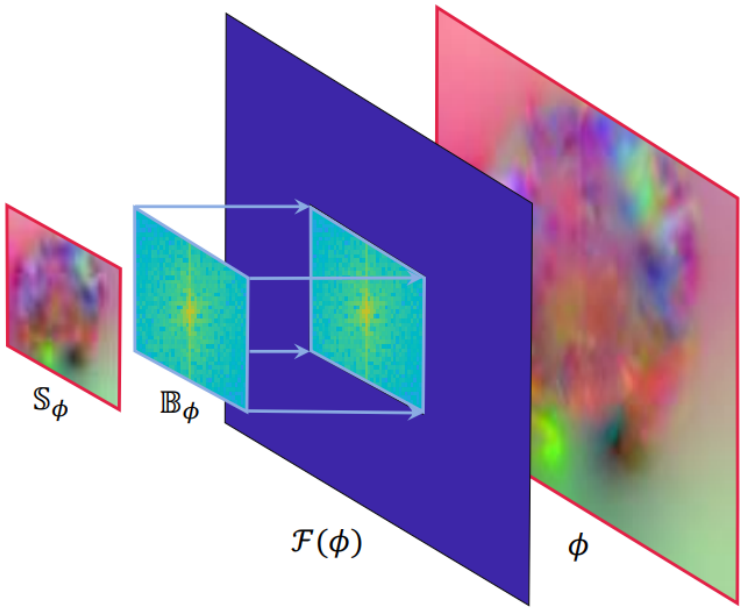


Figure 3.3: Reconstruction of the band-limited displacement field via the decoder taken from [34]. The band-limited representation $\mathbb{B}_\phi$ is padded with zeros to the desired spatial resolution $\mathcal{F}(\phi)$ and then transformed to the full-resolution displacement ϕ using an iDFT.

Diffeomorphic Transforms

Diffeomorphic deformations are differentiable and invertible, thus preserving topology, which is a desirable property for transformations. $\phi : \mathbb{R}^N \rightarrow \mathbb{R}^N$ represents the deformation that maps the coordinates from one image to coordinates in another image, as long as both images have the dimension N. When using a stationary velocity field representation like e.g. *DARTEL* [21], the deformation field is defined through the following ordinary differential equation (ODE) [83]:

$$\frac{\partial \phi^{(t)}}{\partial t} = v(\phi^{(t)}), \tag{3.7}$$

where $\phi^{(0)} = \mathrm{Id}$ is the identity transformation and t is time. The final registration field $\phi^{(1)}$ can be obtained by integrating the stationary velocity field v over $t = [0, 1]$. This is typically done by integration numerically using scaling and squaring [84]. The integration of a stationary ODE represents a one-parameter subgroup of diffeomorphisms. In group theory, v is a member of the Lie algebra and is exponentiated to produce $\exp(v) = \phi^{(1)}$, which is also a member of the Lie group. From the properties of one-parameter subgroups, for any scalars t and t', $\exp((t+t')\cdot v) = \exp(t\cdot v) \circ \exp(t'\cdot v)$, where $\circ$ is a composition map associated with the Lie group. Starting from $\phi^{(1/2^T)} = p + v(p)$ where p is a map of spatial locations, we use the recurrence $\phi^{(1/2)^{t1}} = \phi^{(1/2^t)} \circ \phi^{(1/2^t)}$ to obtain $\phi^1 = \phi^{(1/2)} \circ \phi^{(1/2)}$. T is chosen so that $v \approx 0$ [83]. As diffeomorphic deformations are defined as smooth and invertible deformations, the output of the iDFT layer

in *Fourier-Net* can be regarded as a stationary velocity field v instead of a displacement field ϕ. In Figure 3.1 scaling and squaring layers are visualized. These apply the diffeomorphic transformation in three steps [84]:

1. Scaling: Divide the velocity field v by a factor 2^N , so that $\frac{v}{2^N}$ is close to zero (depending on the desired accuracy).

2. Exponentiation: Compute $\exp\left(\frac{v}{2^N}\right) = \phi^{(1)}\left(\frac{v}{2^N}\right)$ with a first-order explicit numerical scheme.

3. Squaring: N recursive squarings of $\phi^{(1)}\left(\frac{1}{2^N}\right) = \exp\left(\frac{v}{2^N}\right)$ to yield an accurate estimation of $\phi^{(1)}(1) = \phi^{(1)}\left(\frac{1}{2^N}\right)^{2^N} = \exp\left(\frac{v}{2^N}\right)^{2^N} = \exp(v)$.

Thus, the diffeomorphic deformation can be efficiently calculated.

Spatial Transformer

The warping layer of *Fourier-Net* utilizes the *Spatial Transformer* [85], which allows for spatial image manipulation within the network. This is a differentiable and learnable module for neural networks which applies a spatial transformation to a feature map during a single forward pass. The spatial transformer mechanism is split into three parts as seen in Figure 3.4. First is the localization network, which takes the input and outputs the parameters for the transformation. These are then used to create a sample grid using the grid generator. Lastly, the sampler produces the output feature map based on the input at the grid points.
From the input feature map $U \in \mathbb{R}^{H \times W \times C}$ with width W, height H and channels C the localization network f_{loc} computes the parameters $\theta = f_{\text{loc}}(U)$ of the transformation $\mathcal{T}_\theta$ which is later applied to the feature map. Thus the size of θ varies depending on the transformation. The localization network function can both be implemented as a fully-connected network or as a CNN, but should include a final regression layer to produce the transformation parameters.
In order to warp the input feature map, each output pixel is computed by applying a sampling kernel centered at a particular location in the input feature map. The output pixels are defined to lie on a regular grid $G = G_i$ of pixels, forming an output feature map $V \in \mathbb{R}^{H' \times W' \times C}$, where H' and W' are the height and width of the grid with C again being the number of channels, which is the same for input and output. In order to perform a spatial transformation of the input feature map U, the sampler must take the set of sampling points $\mathcal{T}_\theta(G)$, along and produce the sampled output feature map V. Each coordinate (x_i^s, y_i^s) in $\mathcal{T}_\theta(G)$ defines the spatial location in the input where a sampling kernel is applied to get the value at a particular pixel in the output:

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y), \tag{3.8}$$

with Φ_x and Φ_y being the parameters for a generic sampling kernel k that defines the image interpolation, U_{nm}^c is the value of the input feature maps at location (n, m) in

the channel $c \in [1, ..., C]$ and V_i^c is the value for every pixel $i \in [1, ..., H'W']$ for the output feature map. Any sampling kernel can be used, as long as (sub-) gradients can be defined with respect to (x_i^s, y_i^s) to allow the loss gradients to flow back not only to the input feature map, but also to the sampling grid coordinates and therefore back to the transformation parameters θ and localization network, thus enabling back-propagation [85].

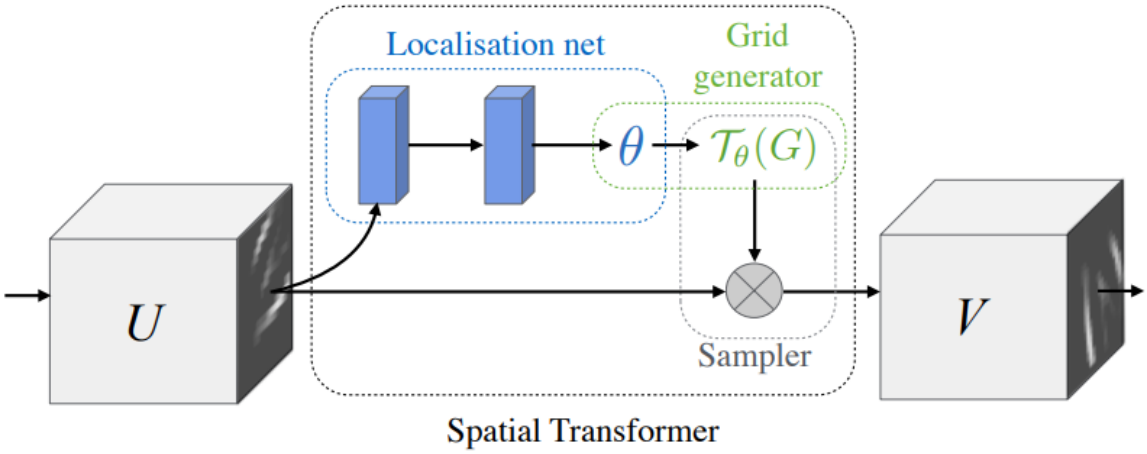


Figure 3.4: Architecture of the *Spatial Transformer* taken from [85]. The input feature map U is passed to a localization network which outputs the transformation parameters θ. The regular spatial grid G over V is transformed to the sampling grid $\mathcal{T}_\theta(G)$, which is applied to U producing the warped output feature map V.

Loss Function

The loss function consists of two parts to enable unsupervised learning, which are balanced using the scalar parameter λ. The first, $\mathcal{L}_1$, measures the similarity between the fixed image and the moving image after warping, while the second, $\mathcal{L}_2$, ensures a smooth displacement field. Thus, the unsupervised loss $\mathcal{L}$ can be calculated as follows:

$$\begin{aligned} \mathcal{L}(\Theta) &= \min \Big(\mathcal{L}_1(\phi(\Theta)) + \lambda \cdot \mathcal{L}_2(\phi(\Theta)) \Big) \\ &= \min \Big(\mathcal{L}_1(v(\Theta)) + \lambda \cdot \mathcal{L}_2(v(\Theta)) \Big), \end{aligned} \tag{3.9}$$

for both displacement fields ϕ and velocity fields v. The first part of the loss function consists of:

$$\mathcal{L}_1(\phi(\Theta)) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{Sim}(M_i \circ (\phi_i(\Theta) + \mathrm{Id}) - F_i), \tag{3.10}$$

where $\circ$ denotes the warping operation, N the number of training pairs with moving images M_i and fixed images F_i, Θ the network parameters, ϕ_i the displacement field and

Id the identity grid. $\mathcal{L}_{Sim}$ determines the similarity between warped moving images and fixed images via MSE or NCC, and the second term of the unsupervised loss, $\mathcal{L}_2$, defines the smoothness regularization function that controls smoothness of the displacement fields:

$$\mathcal{L}_2(\phi(\Theta)) = \frac{1}{N}\sum_{i=1}^{N} ||\nabla\phi_i(\Theta)||_2^2, \tag{3.11}$$

with ∇ denoting the first order gradient and $||\cdot||_2^2$ denoting the squared L_2-Norm. When using the squaring and scaling layers, thus making the deformation of the moving image diffeomorphic, the loss needs to be modified by replacing the displacement field θ with the velocity field v. Thus, both parts of the of the loss function need to be changed:

$$\mathcal{L}_1(v(\Theta)) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{Sim}(M_i \circ Exp(v_i(\Theta) - F_i), \tag{3.12}$$

$$\mathcal{L}_2(v(\Theta)) = \frac{1}{N}\sum_{i=1}^{N} ||\nabla v_i(\Theta)||_2^2 \tag{3.13}$$

3.1.3 Fourier Net+

Fourier-Net+, as the name suggests, is an extension of Fourier-Net which takes the band-limited spatial representation of the images as input, instead of their original full-resolution counterparts. This leads to further reduction in the number of convolutional layers in the contracting path of the network, resulting in a decrease of parameters, memory usage, and computational operations. This makes *Fourier-Net+* even more efficient than its predecessor [34].

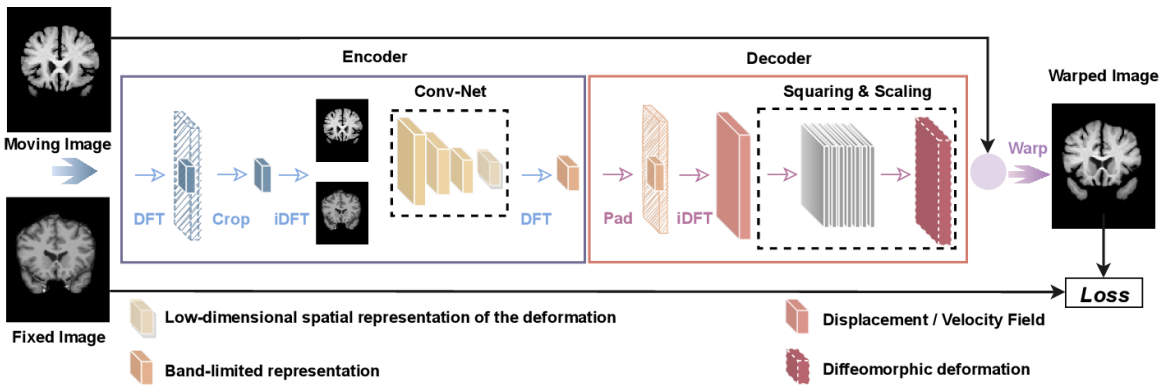


Figure 3.5: Architecture of *Fourier-Net+* taken from [34]. The encoder has a additional compression step to reduce the input size of the images compared to *Fourier-Net*. This is achieved by center-cropping the k-space of the images obtained via a DFT and then reconstructing them via an iDFT to get a compressed version of the inputs without the higher frequencies present. The rest of the network is the same as for *Fourier-Net*.

As seen in Figure 3.5, the network architecture is almost the same as for *Fourier-Net* (see Figure 3.1 for comparison). However, while the decoder, and thus the loss function, remain the same, the encoder is slightly altered to make the network even more efficient. For this, similarly to the decoder, a DFT is used, however this time the idea is applied to the input images. These are first transformed into the Fourier domain, then low-pass filtered by center-cropping and finally reconstructed from their band-limited representation back into the spatial domain via an iDFT. The two images, now compressed, are the input for the encoder of *Fourier-Net*, meaning the CNN and following DFT. However, due to the band-limiting before the CNN, the latter can be made much more light-weight, thus reducing computational cost. This is visualized in Figure 3.7. Thus, *Fourier-Net+* too is overall lighter than the baseline *Fourier-Net* in terms of the number of parameters and computations. However, such a light network may face limitations in accurately capturing complex deformations. To counter this potential weakness, the authors propose a cascaded version of *Fourier-Net+*, which uses multiple versions of *Fourier-Net+* cascaded one after the other to achieve a better overall displacement field [34]. A schematic for this can be seen in Figure 3.8.

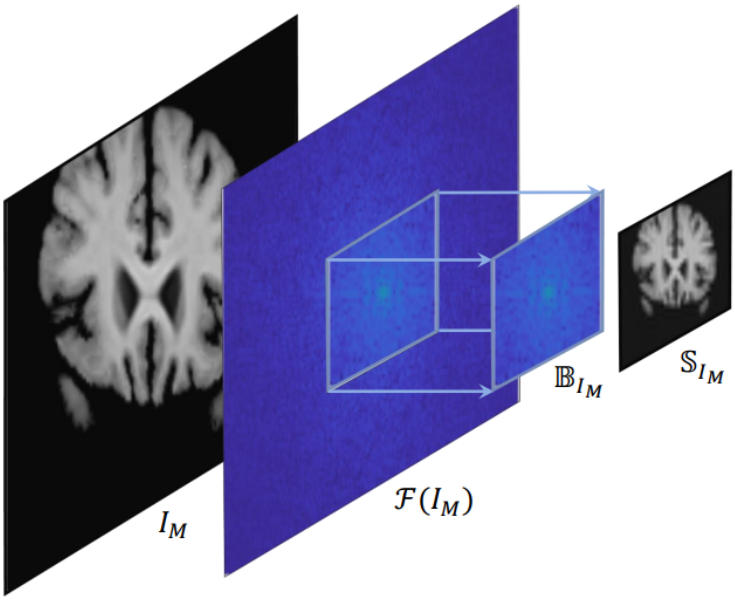


Figure 3.6: Compression in the frequency domain of the encoder used in *Fourier-Net+* taken from [34]. The frequencies $\mathcal{F}(I_M)$ are obtained via a DFT and then center-cropped to obtain a band-limited version $\mathbb{B}_{I_M}$. The compressed image $\mathbb{S}_{I_M}$ is then reconstructed via a iDFT.

Changes to the Encoder

In order to further reduce the amount of computational operations, *Fourier-Net+* discards the early layers of the encoder. Instead, a DFT $\mathcal{F}(I_M)$ followed by a center-crop to produce the band-limited representation $\mathbb{B}_{I_M}$ in the frequency domain and iDFT are

used to get the spatial patch $\mathbb{S}_{I_M}$, while the rest of the encoder from *Fourier-Net* stays the same. The input I_M is thus compressed to a lower resolution (i.e. band-limited) using the frequency space, which reduces the computational cost. This process is visualized in Figure 3.6. The encoder of *Fourier-Net+* has several convolutional layers less in the contracting path compared to *Fourier-Net*, which leads to a further accelerated registration process while reducing the memory footprint. These advantages are visualized in Figure 3.7 where the amount of different layers between a conventional U-Net, *Fourier-Net* (with the smaller decoder) and *Fourier-Net+* (with a smaller encoder and decoder) are shown.

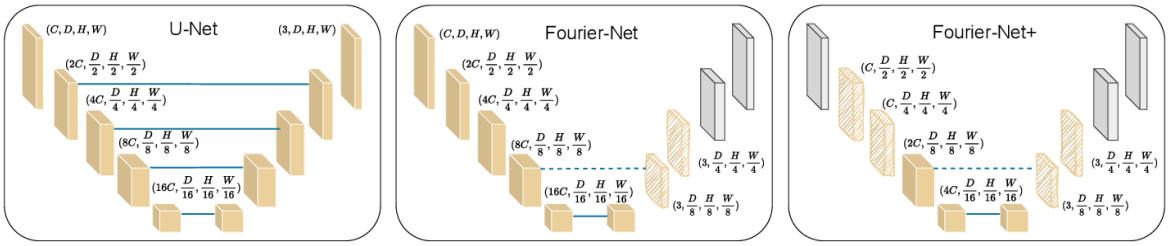


Figure 3.7: Architecture of the CNN for a typical U-Net, *Fourier-Net* and *Fourier-Net+* taken from [34]. The latter have far less trainable network parameters due to only computing a band-limited displacement. *Fourier-Net+* additionally has a smaller encoder due to compressing the input images.

Effects of Cascading

As seen in the previous section, *Fourier-Net+* is lighter than *Fourier-Net* due to the band-limited representation of both images and deformations lowering the number of parameters and computations. This, however, can lead to limitations when trying to accurately capture complex deformations. To this end, a cascaded version of *Fourier-Net+* called $K\times$*Fourier-Net+* (see Figure 3.8), with K denoting the amount of cascades, can be used where the warped image of one cascade is used as the moving image of the next. It is important to note that the weights are not shared between cascades. The squaring and scaling layers are applied after the last cascade, in case a diffeomorphic deformation is wanted. The same is true for the calculation of the loss function.

In order to more accurately describe this version of *Fourier-Net+* one can look at the in- and outputs of the different cascades. The first cascade of $K\times$*Fourier-Net+* has the moving image I_M and fixed image I_F as inputs. While the latter always stays the same for all cascades the warped image $I_M^{w(1)}$ from the first cascade is used as input for the second cascade instead of the original moving image. Thus, in general $I_M^{w(k-1)}$ and I_F are the inputs for a cascade $k \in [1, K]$ with output $\delta\phi^{(k)}$. Furthermore, $I_M^{w(k)}$ can be defined as:

$$I_M^{w(k)} = ((((I_M \circ \delta\phi^{(1)}) \circ \delta\phi^{(2)}) \circ \dots) \circ \delta\phi^{(k-1)}) \circ \delta\phi^{(k)} = I_M \circ \phi^{(k)}, \tag{3.14}$$

with $\phi^{(k)}$ being the displacement field computed by composing the outputs of the first cascade up to the k-th cascade:

$$\phi^{(k)} = \delta\phi^{(1)} \circ \delta\phi^{(2)} \circ \cdots \circ \delta\phi^{(k-1)} \circ \delta\phi^{(k)}. \tag{3.15}$$

Thus, the output displacement field of the K-th cascade $\phi^{(K)}$ is the final displacement used to warp I_M, the original moving image, in order to compute the loss [34].

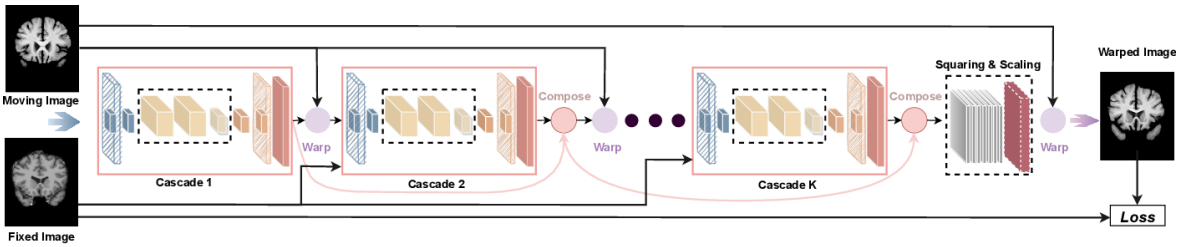


Figure 3.8: Cascaded version of *Fourier-Net+* taken from [34]. Multiple instances of *Fourier-Net+* can be cascaded with independent weights for better performance as the network itself is very memory efficient.

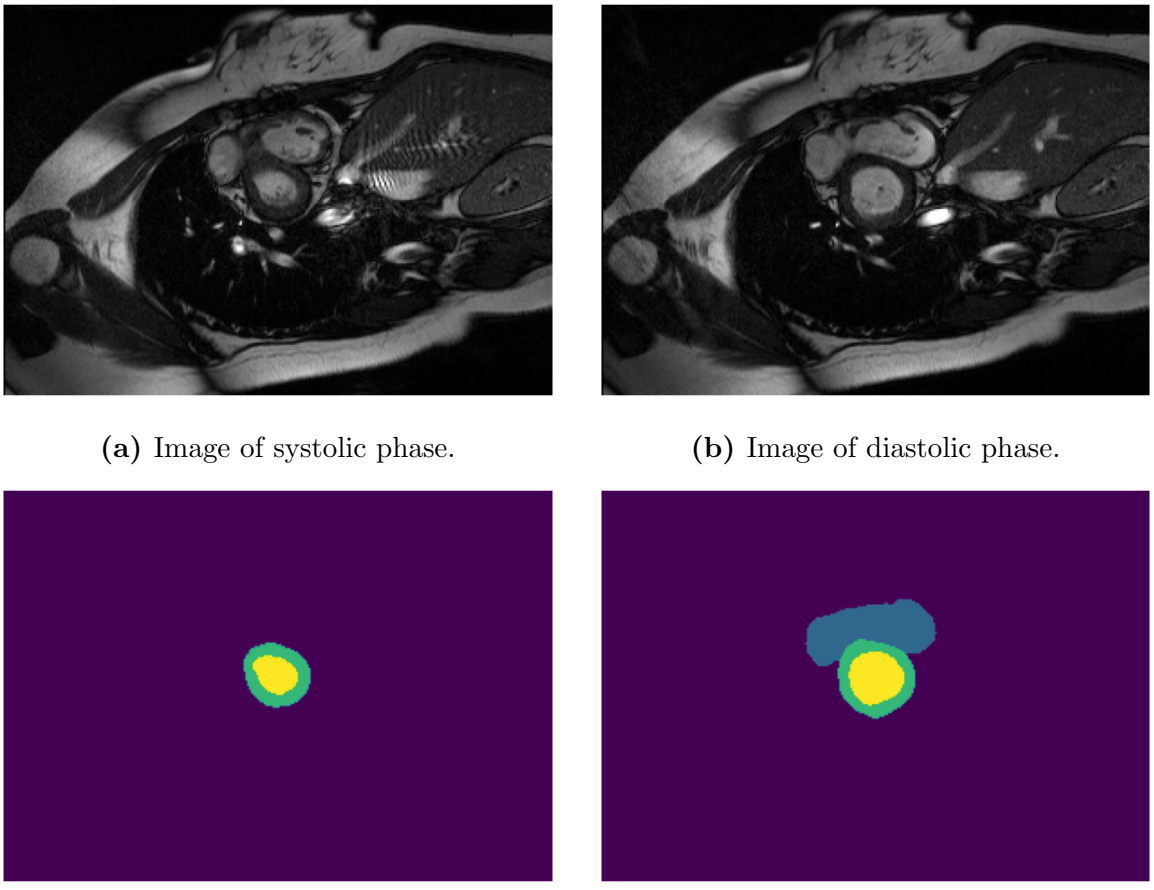
3.2 Datasets

In the following chapter, the datasets used in this thesis are presented. The *Automated Cardiac Diagnosis Challenge (ACDC)* dataset [86] from the *MICCAI 2017* conference was mainly used for ablation studies and parameter tests. While it contains no k-space data, segmentations are given for the end-systolic and end-diastolic frames of the test set, which can be used to evaluate the registration performance using the Dice score. The *CMRxRecon* dataset [87] from the *CMRxRecon2023* challenge specializes in CMR imaging. It was used for downstream-tests regarding MR image reconstruction. The cardiac MRI k-space data was used for the evaluation motion-compensated reconstruction pipeline. It contains subsampled data, but does not provide segmentations for multi-coil data.

3.2.1 ACDC Dataset

The *ACDC* dataset contains cardiac cine MRI short-axis data from 150 subjects that were divided into 5 subgroups (4 pathological, 1 healthy). For each subject systolic (see Figure 3.9a) and diastolic frames (see Figure 3.9b) are provided with corresponding segmentations (see Figure 3.9c and 3.9d), which enables direct comparison via e.g. the Dice score. The segmentations contain only four values with 0, 1, 2 and 3 representing pixels located in the background, in the right ventricle (RV) cavity, in the myocardium, and in the left ventricle (LV) cavity. The frames themselves are 3D volumes with size $216 \times 256 \times 10$, thus ten image slices can be extracted which each have size 216×256. The same holds true for the segmentation, which means that both the image data and segmentations can generated in the same manner.

For the training data, the original 4D data was used as we do not require the segmentations for the end-systolic and end-diastolic frames. As the 4D data has size $216 \times 256 \times 10 \times 30$ we can extract 30 frames for each of the ten slices. These can be sorted into 251376 image pairs for training. For the validation and test data the end-systolic and end-diastolic frames with their segmentations were used giving us 641 image pairs each.
As the dataset only contained images reconstructed from fully sampled data, the data had to manually be subsampled. This was done by first obtaining the k-space data via a FFT and then subsampling the k-space for $= 4$, $R = 8$ and $R = 10$ before going back to image space using a iFFT. This naive reconstruction lead to the typical artifacts (see Figure 3.10) that the networks will be tested against.



(a) Image of systolic phase.

(b) Image of diastolic phase.

(c) Segmentation of systolic phase.

(d) Segmentation of diastolic phase.

Figure 3.9: Example images for systolic (a) and diastolic frames (b) with corresponding segmentations (c), (d) taken from the *ACDC* dataset [86].

(a) Manuel subsampling for $R = 4$.

(b) Manuel subsampling for $R = 8$.

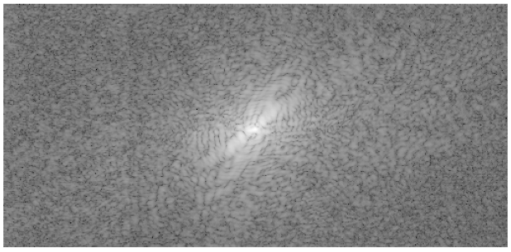
(c) Manuel subsampling for $R = 10$.

Figure 3.10: Example images for the manual subsampling $R = 4$, $R = 8$, and $R = 10$ of the *ACDC* data.

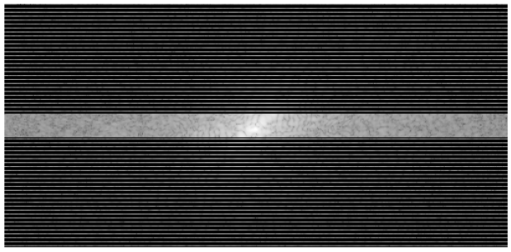
3.2.2 CMRxRecon Dataset

The *CMRxRecon* dataset includes fully sampled and subsampled multi-coil k-space data, as well as auto-calibration lines from the center region. This includes imaging of different anatomical views like long-axis (2-chamber, 3-chamber, and 4-chamber) and short-axis (SAX). There is a total of 120 training data, 60 validation data, and 120 test data from healthy volunteers. In contrast to the *ACDC* dataset, *CMRxRecon* contains only the raw k-space data. One of the goals of the challenge is the reconstruction from subsampled k-space data, e.g. accelerated MRI scans.
As the data was recorded in the k-space (see Figure 3.11) and stored as *.mat* files it first needs to reconstructed to images in order to use them for training and testing of neural networks. First, an iFFT needs to be performed to get from k-space to image space, however the multi-coil data needs further processing. The effect of different coil sensitivities in their respective images can be seen in Figure 3.12, where each coil focuses on a specific area of the image. These images are then stitched together during reconstruction to produce a coil-combined image with great overall contrast. Often the simple RSS method is used where the resulting image is simply the root of the sum of the squared coil images [88].
The dataset, as mentioned before, contains fully sampled as well as subsampled k-space data (see Figure 3.11a and Figure 3.11b), with the latter being used to accelerate the MRI acquisition process. This is done by subsampling the k-space data with mask. The center region of the k-space is usually fully sampled, but the outer regions are subsampled [14]. The dataset contains subsampled k-space data for 4x, 8x and 10x acceleration with the latter of course leading to more artifact in the reconstructed image due to the subsampling. This can be seen in Figure 3.11d, where the image reconstructed from 4x accelerated ($R = 4$) k-space data seems blurred when compared to the fully sampled ($R = 0$) one in Figure 3.11c.
For all experiments, the SAX view data was used. As image sizes between patients can vary slightly, interpolation was used to standardize the images to a size of 246×512 to avoid further problems with e.g. loss calculation. Additionally, min-max-normalization

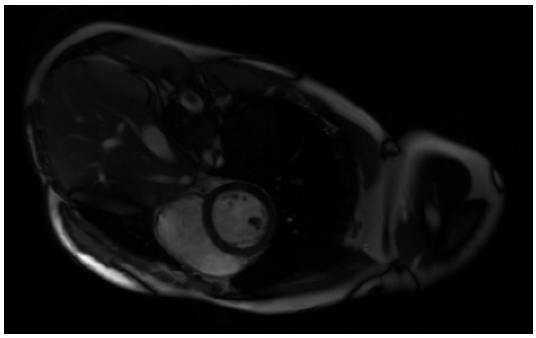
is used to standardize the data range for all images to $[0, 1]$. The reconstructed images were stored for every image slice for every patient. Thus, all frames for an image slice were in a single folder to enable easy access for later data load-in.



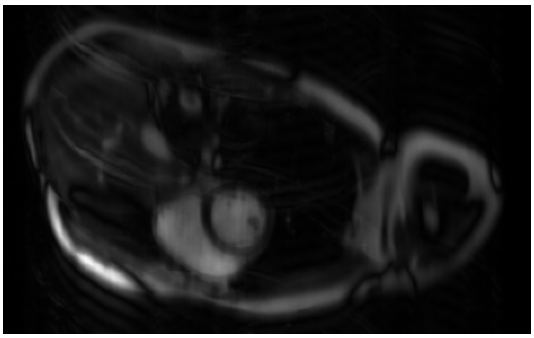
(a) A fully sampled k-space with low frequencies in the center and higher frequencies in the outer regions.



(b) A 4x accelerated k-space. The center region remains fully sampled while higher frequencies are omitted.



(c) Corresponding high quality image reconstructed from the fully sampled k-space.



(d) Corresponding image reconstructed from the subsampled k-space with blurring and aliasing artifacts.

Figure 3.11: Fully sampled and subsampled k-space data from the from the *CMRxRecon* dataset [87] with corresponding images (empty background was cropped out).

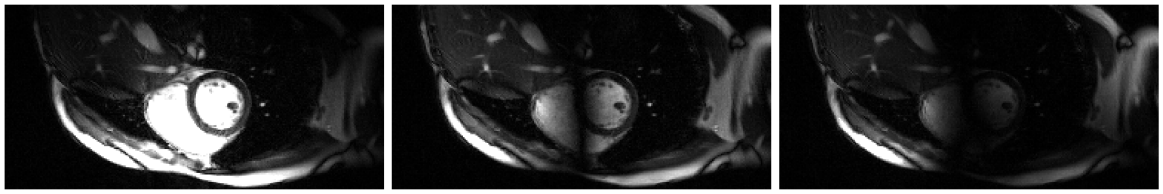


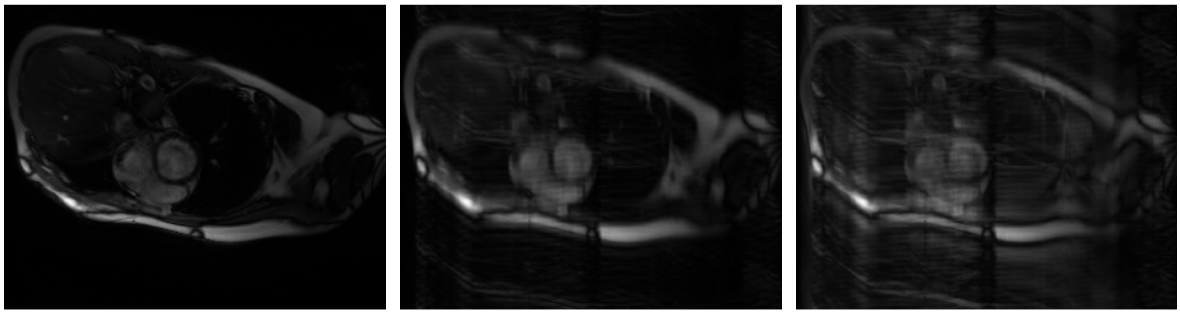
Figure 3.12: MRI images for three different coils from the *CMRxRecon* dataset [87]. The different sensitive areas can clearly be seen as they have significantly higher pixel values compared to the rest of the image. The background was cropped out from the images as the effect is not visible there. Note that the sensitivities are slightly corrupted as the coils were compressed to 10 coils for all the scans.

3.2.3 Simulated Motion

In order to test the motion-compensated reconstruction pipeline, strongly motion-corrupted data is needed. As none of the datasets described previously meet these demands, the motion needed to be simulated manually. To simulate this motion or mis-triggering, two different strategies were used. First, line swapping in the k-space, and second, non-linear lung transformations.

K-space Line Swapping

First, a strategy similar to [39] was used, swapping z k-space lines between the frames in k-space. The frames used for this all stem from the same slice and the information is swapped for all coils. The new fully sampled frames were then subsampled to the needed acceleration using the subsampling masks. Then, an iFT and RSS was used to generate the corrupted images. An example for $z = \{16, 32\}$ can be seen in Figure 3.13.



(a) Original image. **(b)** Line swapping $z = 16$. **(c)** Line swapping $z = 32$.

Figure 3.13: Examples for line swapping for $z = \{16, 32\}$ with $R = 4$ subsampling compared to the original fully sampled image.

Non-Linear Lung Transformations

Second, a simulation of non-linear lung movement was performed by randomly deforming the images and then generating the motion-corrupted k-space data from them. For this, a random displacement generator from [27] was used to generate random displacements in image space with a maximum velocity of 0.1. These displacements were then applied to each coil image before using an FT to convert from image to k-space, thereby modeling lung motion by expanding a cardiac time point into different respiratory time points for the multi-coil k-space data. This was done for $L = 4$ additional lung motion frames for each cardiac frame. An example of such additional frames with artificial lung movement can be as seen in Figure 3.14.

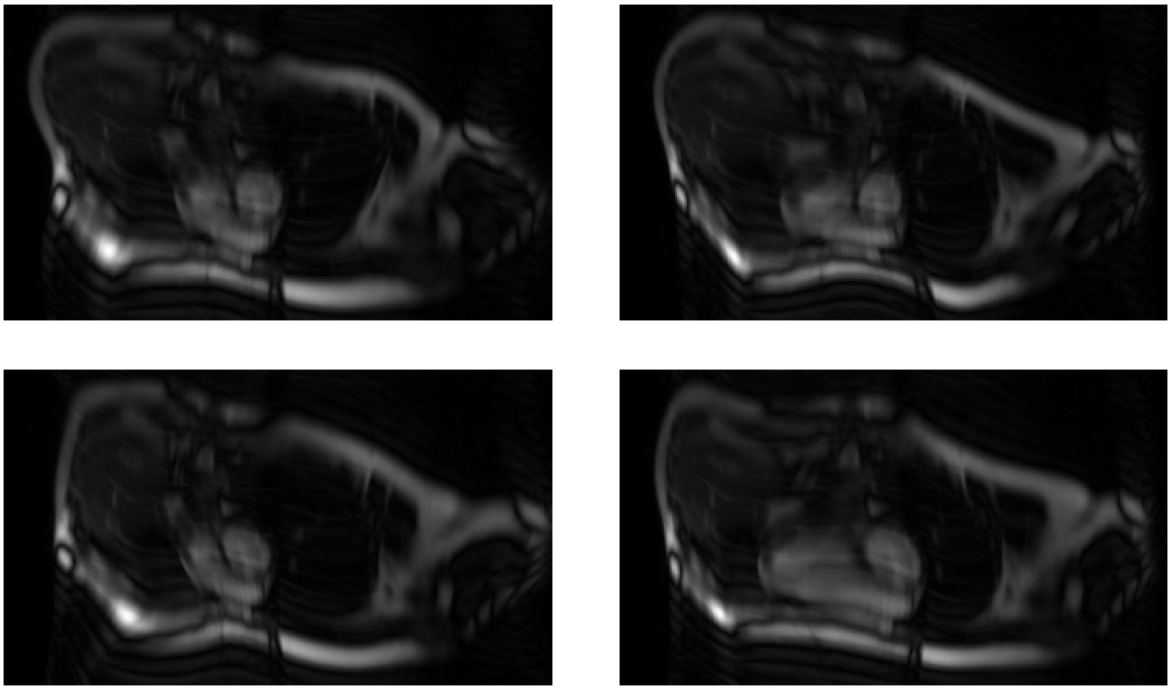


Figure 3.14: $L = 4$ images with simulated lung motion for the same cardiac frame (background was cropped out).

3.3 Experiments

In the following chapter the different experiments that were conducted are described and explained. First, experiments were conducted on the *ACDC* dataset to find the optimal parameters for our model using ground truth segmentations. Then a motion-compensated reconstruction pipeline was used as an downstream task on the *CMRxRecon* dataset.

3.3.1 Ablation Studies and Parameter Tests

As the evaluation on the *CMRxRecon* dataset is limited to image similarity measures, further testing was done on the *ACDC* dataset which contains cardiac data with segmentations. These can be used to better evaluate the registration performance by using e.g. the Dice score. The tests include comparisons between *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net*, parameter tests for the starting channel size, FT crop size as well as comparisons with VoxelMorphs and dense versions of the three networks.

Fourier-Net versus Fourier-Net+

First, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net*, as introduced in sections 3.1.2 and 3.1.3, are compared using both baseline and diffeomorphic versions. Additionally, network versions creating dense instead of band-limited displacements were used for

comparison. The registration performance was evaluated on the test set using the percentage of Dice scores calculated with and without the background label, percentage of SSIM, MSE multiplied by 10^{-3} (denoted by m for milli) and the percentage of negative Jacobian determinant of deformation. The mean inference time on the GPU (NVIDIA GeForce RTX 2080 Ti) together with memory consumption for each model are also added to the comparison with the latter containing number of trainable model parameters, mult-add operations (in millions or billion as denoted by M or G) and the total memory in Megabyte (MB). The latter is only given for the normal versions as the diffeomorphic version require the same amount of memory as well as the baseline dense versions. The first network was trained for a maximum of 15 epochs with early stopping activating after 3 epochs without improvement of the validation Dice score. The other variants were than trained with the same number of epochs, which worked out to be only 6 epochs due to the large training set in the experiments without any data augmentation. All networks were trained on the fully sampled *ACDC* data with MSE as the similarity loss, a channel size of 8, learning rate of 0.0001 and $\lambda = 0.01$. *Fourier-Net+* and *4xFourier-Net* used a FT crop size of 48×48 for these experiments. All of these parameters were constant for the diffeomorphic and dense versions of the networks. The results can be found in section 4.1.1.

Starting Channel Size

Next, the impact of the starting channels (see section 3.1.2) on the *ACDC* data was examined as these dictate the number of features that the network uses. For this, *Fourier-Net+* and *4xFourier-Net+* was trained with MSE-loss, a learning rate of 0.0001, $\lambda = 0.01$, FT crop of 48×48 and no diffeomorphic transform. The experiments were again only conducted on the fully sampled data. Four different channel sizes were used for training: 8, 16, 32 and 64. The first network variant was trained with early stopping, which ended at 6 epochs. For better comparability, the other network variants were also trained with the same number of epochs. The metrics from the previous experiment were again used. The results can be found in section 4.1.2.

Fourier-Transform Crop Size

The FT crop size, used for compressing the input images, is a parameter specific to *Fourier-Net+* and *4xFourier-Net+*. Four different sizes of the FT crop were analyzed: 80×168, 40×84, 48×48 and 24×24. A larger crop would obviously compress the images less, thus leading to less efficiency, however, a very small crop, while very efficient, might lose some of the image details leading to a decrease in performance. The experiments were again only conducted on the fully sampled data as one would expect similar results on subsampled data. The network variants were trained with MSE as the similarity loss, a channel size of 8, learning rate of 0.0001 and $\lambda = 0.01$. Again, the networks were trained for only 5 epochs, as this is were the first model ended training due to early stopping. The registration performance was evaluated on the test set using the same metrics as before. The results can be found in section 4.1.3.

Comparison with VoxelMorph

After finding parameters which optimize registration performance while trying to minimize memory consumption, a comparison to another commonly used registration network was made. For this, a diffeomorphic version [89] of the well known *VoxelMorph* [30] was used with integration steps $T = 7$. It was trained with the default number of layers, MSE as the similarity loss, a learning rate of 0.0001, $\lambda = 0.01$. It provides a dense displacement field to align the image pair. For comparison *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* were trained for 6 epochs with MSE as the similarity loss, a channel size of 16, learning rate of 0.0001 and $\lambda = 0.01$ as well as a FT crop size of 48×48, which provide a good balance of performance and memory imprint for the networks. To evaluate registration performance the same metrics as before were used. All networks were trained on the fully sampled *ACDC* data and the results can be seen in section 4.1.4.

Dense Displacement on Accelerated Data

The difference between a dense displacement field and a band-limited one was already explored in a previous section, but only on fully sampled, not accelerated data. Potential performance changes on the latter were investigated in theses tests with the hypothesis that the network variants with the dense displacement will perform worse than the variants with the band-limited displacement field for strongly subsampled data. This expectation stems from the fact that frequencies outside the center region of the k-space are dropped for the accelerated data thus reducing the advantage of the dense displacement in comparison to the band-limited (i.e. center-cropped) displacement. The tests again included *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+*, which were all trained for 6 epochs with MSE as the similarity loss, a channel size of 16, 48×48 FT crop, learning rate of 0.0001 and $\lambda = 0.01$. The registration performance was evaluated using the same metrics as before. The results can be seen in section 4.1.5.

3.3.2 Registration Performance on Subsampled Data

After comparing *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* with both dense and band-limited displacement fields as well as another comparison with *VoxelMorph* on fully sampled data an extension to subsampled data needs to be made. For this *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* were compared to a baseline consisting of the unaligned test image pairs as well as *NiftyReg* [23], a traditional registration algorithm, and *VoxelMorph* [30], a dense registration network.
Fourier-Net, *Fourier-Net+* and *4xFourier-Net+* were trained for 6 epochs with MSE as the unsupervised similarity loss, a channel size of 16, learning rate of 0.0001 and $\lambda = 0.01$ as well as a FT crop size of 48×48 for all subsampling data. VoxelMorph was trained with default layers and MSE as the similarity loss, a learning rate of 0.0001 and $\lambda = 0.01$. Three subsamplings $R = 4$, $R = 8$ and $R = 10$ were tested. For reference, the data for the fully sampled data ($R = 0$) is also provided. The registration performance of all methods was evaluated on the test set using the Dice score (with and without

the background label), SSIM and MSE. All times are computed on CPU for better comparability as *NiftyReg* currently only works on CPU as the GPU capable versions were deprecated and the other networks would gain an unfair time advantage when ran on GPU. The baseline has no time associated as it only denotes the metrics on the test data before registration. The results are provided in section 4.2.

3.3.3 Motion-Compensated Reconstruction Pipeline

After multiple ablation studies and parameter tests to find optimal network parameters, a downstream experiment is conducted to show the usability of the networks in real-world applications. For this a motion-compensated reconstruction pipeline was chosen where the network can be used to compensate the motion-corrupted data. As the *ACDC* dataset only contains already reconstructed images, the next tests were all conducted on the *CMRxRecon* dataset.
As discussed in section 2.1.4, neural networks can be used in motion-compensated MRI reconstruction pipelines for either the reconstruction or the motion-compensation. As *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* were already trained for frame-to-frame registration, they could be easily used for the latter. ESPIRiT [61] was used to simulate the coil sensitivity maps needed for a SENSE-type reconstruction from the provided k-space data using an eigenvalue decomposition [61]. For the SENSE [59] reconstruction, an iterative conjugate gradient algorithm was used. This made the pipeline very memory effective as *Fourier-Net+* is a lightweight network. Two experiments were conducted on the *CMRxRecon* dataset as it provides the needed k-space data (both fully sampled and accelerated) as well as the subsampling masks required for reconstruction. However, the cardiac scans only show a small amount of movement between frames due to cardiac movement no larger movement of e.g. the lung. Thus, the input images were further motion-corrupted as the cardiac motion present was deemed not severe enough. This process is detailed in section 3.2.3. An overview of this augmentation incorporated into the reconstruction pipeline can be seen in Figure 3.15. *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* have been pre-trained on image reconstructed from this data via an iFT for frame-to-frame registration using the optimal settings obtained in section 3.3.1 (MSE-loss, 16 channels, 48×48 FT crop, learning rate 0.0001, $\lambda = 0.01$). As the *CMRxRecon* dataset does not include segmentations for Dice calculation, image similarity metrics need to be utilized. For this, SSIM and MSE were used again, however, the PSNR (see section 2.3.3) and HaarPSI [90] were also added. For comparison *VoxelMorph* was used. The results can be seen in section 4.3.

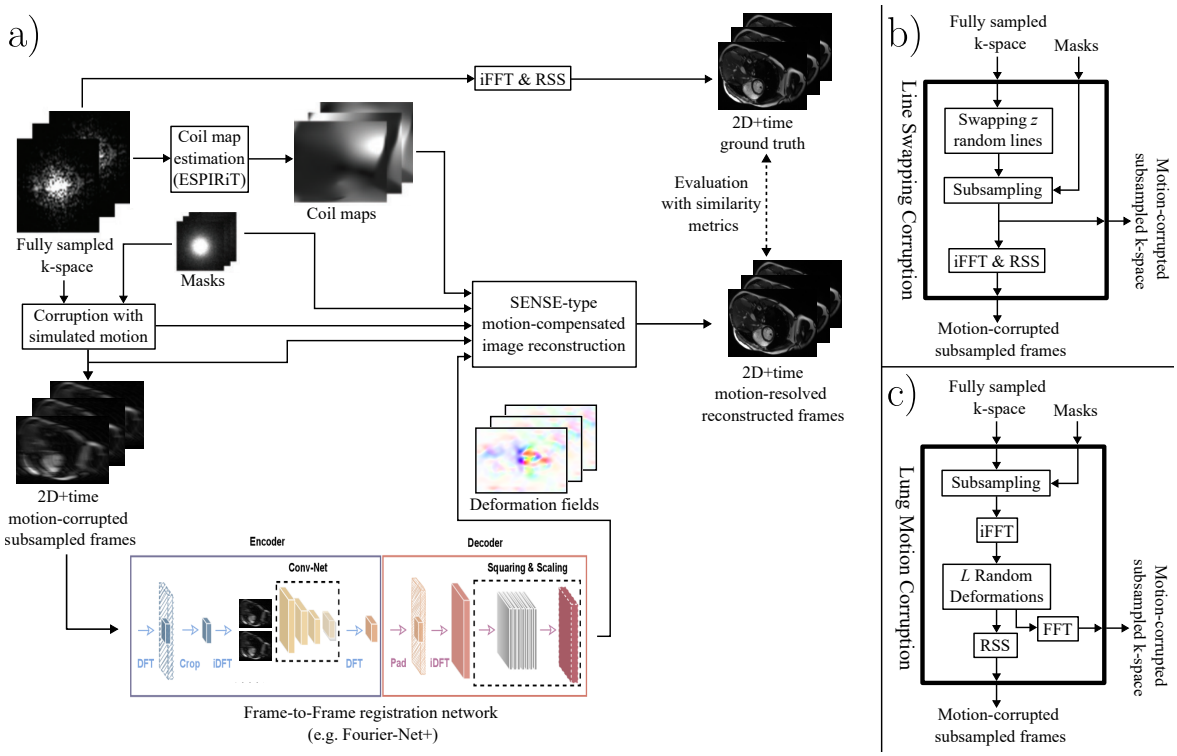


Figure 3.15: Overview of the generalized reconstruction pipeline in a) using *Fourier-Net+* as a registration network (Figure inspired by [7]). The k-space data is corrupted with synthetic motion, either through swapping z random line (steps shown in b) or by via L random deformation frames to simulate lung motion (shown in c)).

4

Results

In this chapter, the results for the experiment described in section 3.3 are reported. This includes ablation studies, parameter tests and registration performance assessment on the *ACDC* dataset as well as a motion-compensated reconstruction pipeline on the *CMRxRecon* dataset.

4.1 Ablation Studies and Parameter Tests

In this section, the results for the ablation studies and parameter tests are described in section 3.3.1 are presented.

4.1.1 Fourier-Net versus Fourier-Net+

First, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net* were compared as described in section 3.3.1.1. The results can be seen in Table 4.1. For the dense displacement variants, the diffeomorphic transform increases the Dice score (with and without the background label) slightly for *Fourier-Net* and *4xFourier-Net*, but decreases for *Fourier-Net+*. The SSIM decreases for all models slightly, while the MSE remains unaffected. The percentage of non-positive Jacobian determinants is lower for the diffeomorphic variants. Times for the baseline and diffeomorphic versions are similar with the *Fourier-Net* being faster without the diffeomorphism by 1.13 ms, while *Fourier-Net+* and *4xFourier-Net* are faster with the diffeomorphism by 3.28 ms and 1.56 ms respectively.
The results for the band-limited versions are very similar with *Fourier-Net* and *Fourier-Net+* having a lower Dice (with background) with the diffeomorphism, while *4xFourier-Net* increases by almost 2%. When excluding the background label *Fourier-Net* is again slightly worse, but *Fourier-Net+* and *4xFourier-Net* both improve, the latter with almost 3% Dice more than the baseline version. The SSIM values are slightly lower for *Fourier-Net* and *Fourier-Net+* with the diffeomorphism, but again higher for *4xFourier-Net*. The MSE is slightly lower for *Fourier-Net* and *4xFourier-Net*, but slightly higher for *Fourier-Net+*. The percentage of non-positive Jabobian determinants again decreases for the diffeomorphic variants. The times again vary slightly with *Fourier-Net* and *Fourier-Net+* being slower by 0.5 ms and 6.26 ms, while *4xFourier-Net* is a lot faster with 14.68 ms difference.
Now to the difference between the dense and band-limited displacements. Overall, the dense displacement versions have a higher Dice score for all models both with and

without the background label. The dense displacements also have better SSIM and MSE metrics, however the difference is not quite as large as with the Dice scores. Only the percentage of non-positive Jacobian determinants is notably lower for the band-limited displacement. In terms of inference time, the dense displacement variants are slightly faster despite having a larger encoder, however all of the models are in the milliseconds (7 ms up to 33.5 ms). Last but not least, the memory consumption needs to be addressed. The dense displacment variants have a larger encoder, as discussed before, and thus have far more parameters, especially for the more optimized *Fourier-Net+* and *4xFourier-Net* (about 5 times more). The number of Mult-Adds and the amount of total memory are more than doubled (almost tripled for *4xFourier-Net*) for the dense displacement versions compared to those with a band-limited displacement across all models.

Table 4.1: Results for the *Fourier-Net versus Fourier-Net+* experiment. The networks are compared using both dense and band-limited displacement fields as well as diffeomorphic transforms on the fully sampled *ACDC* test data.

Metrics	Dense Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	78.22 ± 13.84	78.43 ± 13.96	$\mathbf{79.34 \pm 14.03}$
% Dice* ↑	71.03 ± 18.87	70.50 ± 19.17	$\mathbf{72.02 \pm 18.94}$
% SSIM ↑	$\mathbf{91.92 \pm 3.32}$	89.09 ± 4.10	89.65 ± 3.85
MSE (m) ↓	$\mathbf{0.09 \pm 0.06}$	0.17 ± 0.13	0.15 ± 0.12
$\% \lvert J_\phi \rvert \leq 0$ ↓	0.53 ± 0.52	0.32 ± 0.55	$\mathbf{0.04 \pm 0.09}$
Time [ms] ↓	**7.03**	10.74	21.89
	Diff-Fourier-Net	Diff-Fourier-Net+	Diff-4xFourier-Net+
% Dice ↑	78.56 ± 14.13	77.88 ± 13.81	$\mathbf{79.49 \pm 14.26}$
% Dice* ↑	71.31 ± 19.13	70.19 ± 18.82	$\mathbf{72.12 \pm 19.33}$
% SSIM ↑	$\mathbf{91.79 \pm 3.38}$	89.08 ± 4.10	89.62 ± 3.98
MSE (m) ↓	$\mathbf{0.09 \pm 0.06}$	0.16 ± 0.13	0.15 ± 0.12
$\% \lvert J_\phi \rvert \leq 0$ ↓	0.03 ± 0.06	0.00 ± 0.00	0.00 ± 0.00
Time [ms] ↓	8.16	**7.46**	20.33
Parameters ↓	645,216	**380,470**	1,521,880
Mult-Adds (G) ↓	1.12	**0.04**	0.18
Memory [MB] ↓	97.63	**5.81**	21.90

Metrics	Band-limited Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	**77.95 ± 14.71**	75.35 ± 14.28	76.59 ± 14.84
% Dice* ↑	**69.96 ± 21.55**	64.74 ± 21.12	66.82 ± 21.55
% SSIM ↑	**91.35 ± 3.51**	88.42 ± 3.94	88.59 ± 4.23
MSE (m) ↓	**1.00 ± 0.71**	2.06 ± 1.54	1.92 ± 1.46
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.36 ± 0.38	0.13 ± 0.35	**0.03 ± 0.12**
Time [ms] ↓	**14.46**	**14.46**	33.49
	Diff-Fourier-Net	Diff-Fourier-Net+	Diff-4xFourier-Net+
% Dice ↑	77.93 ± 14.93	75.17 ± 13.43	**78.20 ± 14.01**
% Dice* ↑	**69.91 ± 21.99**	65.67 ± 18.61	69.61 ± 19.11
% SSIM ↑	**91.24 ± 3.56**	87.81 ± 3.94	88.81 ± 4.13
MSE (m) ↓	**0.98 ± 0.67**	2.29 ± 1.61	1.85 ± 1.41
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.01 ± 0.03	**0.00 ± 0.00**	**0.00 ± 0.00**
Time [ms] ↓	**14.96**	20.72	18.81
Parameters ↓	434,519	**75,429**	301,716
Mult-Adds (M) ↓	595.04	**10.98**	43.91
Memory [MB] ↓	44.69	**2.25**	7.66

4.1.2 Starting Channel Size

Next, the impact of the starting channel size was examined as explained in section 3.3.1.2. The results can be seen in Table 4.2. The Dice (with and without the background label) increased for both *Fourier-Net+* and *4xFourier-Net+* for larger channel sizes. A marginal increase in SSIM and MSE was observed, while the percentage of non-positive Jabobian determinants decreased. The inference time is not heavily impacted by channel size as the times remain very similar for all sizes. Overall, *4xFourier-Net+* is slower than *Fourier-Net+* by roughly a factor of 3 to 4. The number of parameters drastically increases with starting size leading to an increase in Mult-Add and thus overall memory. As *4xFourier-Net+* is just a cascaded version of *Fourier-Net+* a channel size of 16 for the latter is about the same as channel size 8 for the cascaded version and so on. This can also be seen in the direct comparison between *Fourier-Net+* and *4xFourier-Net+*, where the latter has a better performance, but also needs more memory. While an increase of channel size also increases performance in terms of Dice (with the background label) by 0.36%, 0.43% and 0.41% for the latter, which is quite consistent, it can be observed that an increase of channel size for *Fourier-Net+* has an bigger impact at the beginning (1.38%, 0.78% and 0.15%) and fades towards larger channels sizes. This effect can also be seen when excluding the background label (0.69%, 0.78% and 0.42% for *4xFourier-Net+* compared to 2.16%, 1.09% and 0.10% for *Fourier-Net+*), but is less pronounced for the SSIM and MSE. Note that all versions of *Fourier-Net+* and *4xFourier-Net+* are smaller in terms of total

memory than *Fourier-Net* with channel size 8 (44.69 MB), except for *4xFourier-Net+* with channel size 64.

Table 4.2: Results for the different starting channel sizes of *Fourier-Net+* and *4xFourier-Net+* on the fully sampled *ACDC* test data.

Metrics	Starting Channels - Fourier-Net+			
	8	16	32	64
% Dice ↑	75.50 ± 13.79	76.88 ± 13.86	77.66 ± 13.60	**77.81 ± 13.76**
% Dice* ↑	65.70 ± 18.96	67.86 ± 19.06	68.95 ± 18.65	**69.05 ± 18.80**
% SSIM ↑	88.05 ± 3.89	88.67 ± 3.88	88.83 ± 3.83	**89.02 ± 3.67**
MSE (m) ↓	0.22 ± 0.16	0.20 ± 0.15	**0.19 ± 0.15**	**0.19 ± 0.15**
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.07 ± 0.25	0.04 ± 0.14	0.01 ± 0.04	**0.00 ± 0.03**
Time [ms] ↓	**7.18**	7.77	7.99	7.79
Parameters ↓	**75,429**	300,477	1,199,469	4,793,037
Mult-Adds (M) ↓	**10.98**	42.89	169.54	674.10
Memory [MB] ↓	**2.25**	4.64	11.22	31.57
Metrics	Starting Channels - 4xFourier-Net+			
	8	16	32	64
% Dice ↑	77.54 ± 13.73	77.90 ± 13.92	78.33 ± 14.14	**78.74 ± 13.91**
% Dice* ↑	68.52 ± 18.62	69.21 ± 19.02	69.99 ± 19.21	**70.41 ± 19.02**
% SSIM ↑	88.70 ± 4.17	88.89 ± 4.05	89.08 ± 4.01	**89.29 ± 3.74**
MSE (m) ↓	0.19 ± 0.14	0.19 ± 0.14	**0.18 ± 0.14**	**0.18 ± 0.14**
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.02 ± 0.07	0.01 ± 0.04	0.01 ± 0.05	**0.00 ± 0.00**
Time [ms] ↓	30.13	**24.61**	29.65	27.73
Parameters ↓	**301,716**	1,201,908	4,797,876	19,172,148
Mult-Adds (G) ↓	**0.04**	0.17	0.68	2.70
Memory [MB] ↓	**7.66**	17.23	43.56	124.94

4.1.3 Fourier-Transform Crop Size

The results for the FT crop size experiment, as described in section 3.3.1.3, can be seen in Table 4.3. The Dice score, both with and without the background label, decreases drastically with a smaller crop size. The SSIM and MSE values also get worse (SSIM decreases, MSE increases), however, the percentage of non-positive Jacobian determinants decreases for smaller crop sizes. It should be noted that the crop size of 48×48 is an outlier in this regard breaking the trend with a higher percentage as 40×84. The inference time also decreases slightly with a smaller crop size, although this is quite noisy as the time from 80×168 to 40×84 is halved for *Fourier-Net+*, but the time needed for 48×48 and 24×24 increases again very slightly. For *4xFourier-Net+* the time decrease for all smaller crop sizes, except for 24×24. The number of parameters

does not change for different crop sizes as the networks themselves do not change, however the number of Mult-Adds and the total memory still change with the image size. Both decrease with a larger crop size as the image gets smaller. This effect is again not linear as a reduction in image size yields a larger reduction in memory going from 80×168 to 40×84 than from 48×48 to 24×24.

Table 4.3: Results for four different FT crop sizes for *Fourier-Net+* and *4xFourier-Net+* examined on the fully sampled *ACDC* test data.

Metrics	FT crop size - Fourier-Net+			
	80×168	40×84	48×48	24×24
% Dice ↑	**78.24 ± 14.43**	76.61 ± 13.90	75.27 ± 13.49	73.77 ± 14.42
% Dice* ↑	**70.66 ± 19.70**	67.60 ± 19.24	65.48 ± 18.76	63.37 ± 20.07
% SSIM ↑	**89.81 ± 4.09**	88.58 ± 3.87	87.98 ± 3.91	87.00 ± 3.99
MSE (m) ↓	**0.15 ± 0.12**	0.20 ± 0.15	0.22 ± 0.16	0.27 ± 0.19
$\% \lvert J_\phi \rvert \leq 0$ ↓	0.22 ± 0.46	0.05 ± 0.15	0.09 ± 0.25	**0.00 ± 0.02**
Time [ms] ↓	**6.99**	7.71	7.18	8.14
Parameters ↓	75,429	75,429	75,429	75,429
Mult-Adds (M) ↓	64.03	16.37	10.98	**2.74**
Memory [MB] ↓	9.51	2.97	2.25	**1.12**
Metrics	FT crop size - 4xFourier-Net+			
	80×168	40×84	48×48	24×24
% Dice ↑	**78.12 ± 15.01**	77.49 ± 14.67	74.95 ± 14.15	72.58 ± 14.67
% Dice* ↑	**70.08 ± 21.92**	68.03 ± 21.57	64.54 ± 20.75	61.19 ± 21.63
% SSIM ↑	**89.91 ± 4.01**	89.03 ± 3.98	87.98 ± 3.94	87.02 ± 3.95
MSE (m) ↓	**1.45 ± 1.11**	1.81 ± 1.37	2.22 ± 1.61	2.64 ± 1.84
$\% \lvert J_\phi \rvert \leq 0$ ↓	0.12 ± 0.23	0.03 ± 0.15	0.06 ± 0.23	**0.02 ± 0.15**
Time [ms] ↓	39.03	30.05	30.13	**28.86**
Parameters ↓	301,716	301,716	301,716	301,716
Mult-Adds (M) ↓	256.14	65.50	43.91	**10.98**
Memory [MB] ↓	36.70	10.54	7.66	**3.15**

4.1.4 Comparison with VoxelMorph

Next is the comparison with *VoxelMorph*, as described in section 3.3.1.4. The results can be seen in Table 4.4. *VoxelMorph* has the lowest Dice score with the background label, however it does have a higher value than *Fourier-Net+* for the Dice without the background label. *Fourier-Net* has the highest Dice scores (both with and without the background label), followed by *4xFourier-Net+*. *VoxelMorph* has the best SSIM and MSE values, followed by *Fourier-Net*. The percentage of non-positive Jacobian determinants for *VoxelMorph* is high with 1% while the other models are all under

0.25%. In terms of time, *4xFourier-Net+* is slowest followed by *VoxelMorph*, *Fourier-Net+* and *Fourier-Net*, although all models are very fast (less than 25 ms per image pair). The model with the least amount of parameters and Mult-Adds is *VoxelMorph*, but it still requires quite a lot of memory with almost 40 MB. *Fourier-Net* is the largest model in terms of model parameters, Mult-Adds and total memory (90 MB). *Fourier-Net+* and *4xFourier-Net+* both have a higher number of parameters and Mult-Adds, but a lower total amount of memory is needed (5 MB and 17 MB).

Table 4.4: Comparison of *Fourier-Net*, *Fourier-Net+*, *4xFourier-Net+* and *VoxelMorph* with similarity metrics and memory consumption on the fully sampled *ACDC* test data.

	Fourier-Net	Fourier-Net+	4xFourier-Net+	VoxelMorph
% Dice ↑	**78.31 ± 13.96**	76.88 ± 13.86	77.90 ± 13.92	75.84 ± 13.46
% Dice* ↑	**71.17 ± 18.99**	67.86 ± 19.06	69.21 ± 19.02	68.25 ± 18.36
% SSIM ↑	91.53 ± 3.49	88.67 ± 3.88	88.89 ± 4.05	**93.53 ± 3.30**
MSE (m) ↓	0.09 ± 0.07	0.20 ± 0.15	0.19 ± 0.14	**0.06 ± 0.04**
$\%\,\lvert J_\phi\rvert \leq 0 \downarrow$	0.24 ± 0.45	0.02 ± 0.14	**0.00 ± 0.04**	1.00 ± 0.57
Time [ms] ↓	**7.58**	7.78	24.61	10.82
Parameters ↓	1,735,447	300,477	1,201,908	**84,322**
Mult-Adds (G) ↓	2.35	0.04289	0.17157	**0.00157**
Memory [MB] ↓	90.08	**4.64**	17.23	39.04

4.1.5 Dense Displacement on Accelerated Data

The difference between a dense displacement field and a band-limited one was already explored in section 4.1.1. However, the results in Table 4.1 only include fully sampled data, not accelerated data. These new tests again included *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+*, but are extended to subsampled data. Results for $R = 4$ can be seen in Table 4.5, $R = 8$ in Table 4.6 and $R = 10$ in Table 4.7.
For $R = 4$, the dense version of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* have higher Dice scores than the band-limited versions, though the differences are less than a percent with and less than two percent without the background label. The same is true for the SSIM and MSE values with the dense versions being better. It should be noted that *Fourier-Net* performs best for all the aforementioned metrics for the dense and band-limited displacements with the exception of the Dice with the background label where the band-limited *4xFourier-Net+* is better. The percentage of non-positive Jacobian determinants is higher for the band-limited *Fourier-Net*, while band-limited *Fourier-Net+* and *4xFourier-Net+* show almost no folding. Band-limited *Fourier-Net* and *Fourier-Net+* are faster (1.6 ms and 1.76 ms), likely due to the smaller network size as the encoder lack some layers compared to the dense versions, however, band-limited *4xFourier-Net+* is slower than its dense counterpart by 4.93 ms. Overall, *Fourier-Net+* is the fastest network for both dense and band-limited displacements.

For $R = 8$, all band-limited networks have a lower Dice and SSIM values with a higher MSE. Band-limited *4xFourier-Net+* again has the highest Dice with the background for the band-limited networks while *Fourier-Net* has the highest Dice without background, SSIM and lowest MSE. While it also previously performed best for all these metrics for the dense displacements, *4xFourier-Net+* now has the highest Dice scores for all dense networks. The percentage of non-positive Jacobian determinants is again higher for the band-limited *Fourier-Net* compared to the dense version, while the band-limited *Fourier-Net+* and *4xFourier-Net+* show almost no folding. Band-limited *Fourier-Net* and *Fourier-Net+* are faster by 2.49 ms and 3.08 ms while *4xFourier-Net+* is slower compared to its dense counterpart by 8.03 ms. For both dense and band-limited networks, *Fourier-Net+* is repeatedly the fastest network.
For $R = 10$, the dense networks have higher Dice and SSIM values compared to the band-limited versions. For the MSE, band-limited *Fourier-Net* has similar values to its dense counterpart, while band-limited *Fourier-Net+* and *4xFourier-Net+* are slightly higher compared to the dense versions. For the dense versions *4xFourier-Net+* has the highest Dice scores, while *Fourier-Net* hast the highest SSIM value and lowest MSE value. Band-limited *4xFourier-Net+* has the highest Dice score with the background with band-limited *Fourier-Net* has the highest Dice score without background, SSIM and lowest MSE. The percentage of non-positive Jacobian determinants for the band-limited *Fourier-Net* is equal to the percentage of the dense version, while band-limited *Fourier-Net+* and *4xFourier-Net+* again show almost no folding. Band-limited *Fourier-Net* is faster with 19.9 ms, although the time for dense network is probably an outlier while the band-limited version is fastest for the band-limited networks. *Fourier-Net+* and *4xFourier-Net+* are slower compared to their dense versions with 0.13 ms and 9.58 ms. For the dense networks, *Fourier-Net+* is the fastest network.

4.2 Registration Performance on Subsampled Data

In section 4.1.4, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* were already compared to *VoxelMorph*, however, these comparisons were only done on fully sampled, not accelerated data. To further add to the comparison, *NiftyReg* was used as a traditional registration algorithm. The unaligned test image pairs were treated as a baseline for the lower bound of registration performance. The results can be seen in Table 4.8 for fully sampled ($R = 0$) and subsampled ($R = 4$, $R = 8$, $R = 10$) *ACDC* test data. Values which are worse than the baseline are marked in red, while the best results for each acceleration level and metric is highlighted with blue. All times are computed on CPU due to *NiftyReg* only working on the CPU as the GPU capable versions were deprecated. Additionally, Figure 4.1 shows the performance of the methods for each segmentation label (excluding the background) while Figure 4.2 shows example images and segmentations warped by the different methods for a visual comparison.
For $R = 0$, *Fourier-Net* has the highest Dice score (both with and without the background label), closely followed by *4xFourier-Net+*, while *NiftyReg* performs worse than the baseline. *VoxelMorph* has the highest SSIM, followed by *Fourier-Net* and *NiftyReg*, as well as the lowest MSE where *NiftyReg* again performs worse than the baseline.

Table 4.5: Results for *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* with both dense and band-limited displacement fields on the $R = 4$ *ACDC* test data.

Metrics	Dense Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	**77.37 ± 13.92**	76.90 ± 14.12	77.35 ± 14.04
% Dice* ↑	**68.98 ± 19.10**	68.39 ± 19.30	68.76 ± 19.32
% SSIM ↑	**84.75 ± 7.01**	79.79 ± 9.76	80.32 ± 9.37
MSE (m) ↓	**0.08 ± 0.05**	0.15 ± 0.12	0.13 ± 0.10
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.16 ± 0.19	0.11 ± 0.21	**0.03 ± 0.07**
Time [ms] ↓	8.49	**8.26**	26.33
Metrics	Band-limited Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	76.55 ± 13.89	76.20 ± 13.57	**77.23 ± 13.73**
% Dice* ↑	**67.90 ± 19.44**	66.30 ± 19.01	67.67 ± 18.97
% SSIM ↑	**82.93 ± 8.12**	77.74 ± 10.69	77.96 ± 10.57
MSE (m) ↓	**0.10 ± 0.06**	0.19 ± 0.14	0.18 ± 0.13
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.35 ± 0.41	0.01 ± 0.05	**0.00 ± 0.02**
Time [ms] ↓	6.89	**6.50**	31.26

NiftyReg and *4xFourier-Net+* have the lowest percentage of non-positive Jacobian determinants while *VoxelMorph* has the highest value with 1%, thus indicating some folding. *Fourier-Net* is the fastest method with under $0.1s$ per image pair, while *NiftyReg* needs over $100s$ on the fully sampled data, making it the slowest by a large margin (all other methods were under $1s$).

For $R = 4$, the registration performance decreases for all methods with *NiftyReg* again performing worse than the baseline, while *4xFourier-Net+* has the highest Dice score with the background label, *Fourier-Net* has the highest Dice score without the background. *VoxelMorph* has the highest SSIM and lowest MSE values, followed by *Fourier-Net* and *NiftyReg*, with none of methods performing worse than the baseline. *4xFourier-Net+* again has the lowest percentage of non-positive Jacobian determinants, closely followed by *Fourier-Net+*, while *VoxelMorph*, similar to the fully sampled data, has the highest value. In terms of time, *NiftyReg* is again the slowest method with about $80s$, while all other methods need around $0.1s$ with *Fourier-Net+* being the fastest.

For $R = 8$, *4xFourier-Net+* has the highest Dice score with the background label, while *Fourier-Net* has the highest Dice score without the background label. *NiftyReg* has a lower Dice than the baseline (both with and without the background label). *NiftyReg* again performs best for SSIM and MSE followed by *Fourier-Net* and *NiftyReg*. *4xFourier-Net+* has the lowest percentage of non-positive Jacobian determinants followed by *Fourier-Net+*, while *VoxelMorph* again has the highest value. *NiftyReg* is the slowest method, as before, with over $80s$, similar to $R = 4$, while *Fourier-Net+* is

Table 4.6: Results for *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* with both dense and band-limited displacement fields on the $R = 8$ *ACDC* test data.

Metrics	Dense Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	77.48 ± 14.00	77.30 ± 13.72	$\mathbf{77.80 \pm 14.02}$
% Dice* ↑	68.86 ± 19.34	68.39 ± 19.15	$\mathbf{69.18 \pm 19.37}$
% SSIM ↑	$\mathbf{90.08 \pm 3.05}$	87.60 ± 3.78	87.91 ± 3.64
MSE (m) ↓	$\mathbf{0.05 \pm 0.04}$	0.10 ± 0.10	0.09 ± 0.09
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.11 ± 0.15	0.06 ± 0.13	$\mathbf{0.02 \pm 0.06}$
Time [ms] ↓	11.46	**10.79**	38.55
Metrics	Band-limited Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	75.52 ± 13.69	75.76 ± 13.43	$\mathbf{76.56 \pm 13.47}$
% Dice* ↑	$\mathbf{66.86 \pm 18.97}$	65.45 ± 18.63	66.81 ± 18.77
% SSIM ↑	$\mathbf{89.51 \pm 3.34}$	86.30 ± 4.18	86.29 ± 4.23
MSE (m) ↓	$\mathbf{0.06 \pm 0.04}$	0.13 ± 0.11	0.13 ± 0.11
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.27 ± 0.29	0.03 ± 0.11	$\mathbf{0.00 \pm 0.02}$
Time [ms] ↓	8.97	**7.71**	46.58

again the fastest despite being slightly slower than before.
For $R = 10$, *4xFourier-Net+* has the highest Dice score with the background label, while *Fourier-Net* has the highest Dice score without the background. *NiftyReg* repeatedly has a lower Dice score than the baseline (both with and without background label), while *VoxelMorph* is the best method in terms of SSIM and MSE, followed by *Fourier-Net* and *NiftyReg* with no method being worse than the baseline for these metrics. *4xFourier-Net+* has the lowest percentage of non-positive Jacobian determinants closely followed by *Fourier-Net+*, while *VoxelMorph* has the highest value like before. *Fourier-Net+* is the fastest method with under $0.01s$, while *NiftyReg* is again the slowest despite its best time yet with about $47s$.

Cardiac Labels

To further evaluate the difference between the methods, one can look at the Dice scores of the three individual cardiac labels. These can be seen as boxplots in Figure 4.1a for the left ventricle (LV), Figure 4.1b for the myocardium and Figure 4.1c for the right ventricle (RV). The Dice scores for each label vary widely but the performance of all methods is best for the RV and worst for the myocardium. This seems to be an underlying principle of the data as the same behavior can be seen in the unaligned baseline.
NiftyReg has high values for the LV, even surpassing *VoxelMorph*, while it has the lowest values for all methods on the myocardium (slightly higher than the baseline)

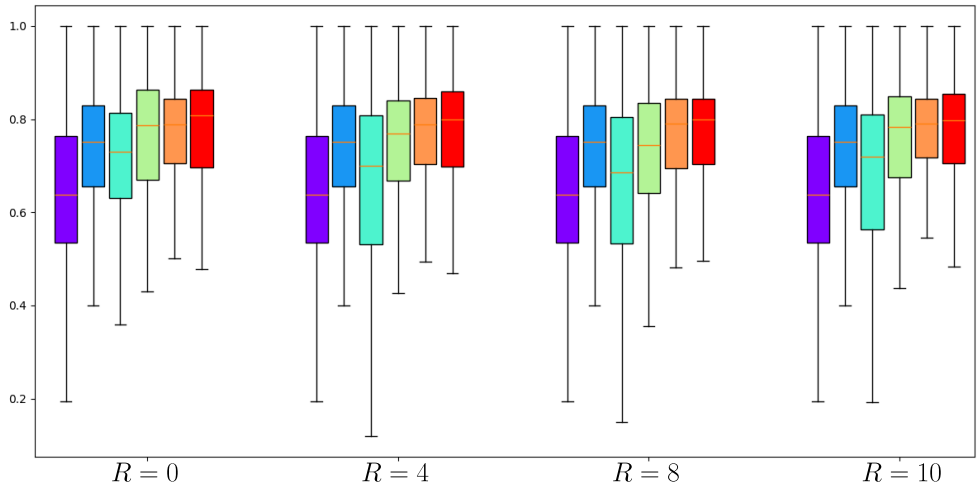
Table 4.7: Results for *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* with both dense and band-limited displacement fields on the $R = 10$ *ACDC* test data.

Metrics	Dense Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	77.39 ± 13.89	77.31 ± 13.82	$\mathbf{77.43 \pm 13.99}$
% Dice* ↑	68.86 ± 19.27	68.69 ± 19.07	$\mathbf{68.90 \pm 19.43}$
% SSIM ↑	$\mathbf{92.92 \pm 2.68}$	91.09 ± 3.42	91.33 ± 3.29
MSE (m) ↓	$\mathbf{0.05 \pm 0.04}$	0.09 ± 0.09	0.08 ± 0.09
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.11 ± 0.15	0.06 ± 0.11	$\mathbf{0.04 \pm 0.11}$
Time [ms] ↓	27.96	**9.49**	18.31
Metrics	Band-limited Displacement		
	Fourier-Net	Fourier-Net+	4xFourier-Net+
% Dice ↑	76.87 ± 13.97	76.10 ± 13.83	$\mathbf{76.98 \pm 13.57}$
% Dice* ↑	$\mathbf{68.24 \pm 19.27}$	66.27 ± 19.20	67.40 ± 18.90
% SSIM ↑	$\mathbf{92.77 \pm 2.74}$	90.32 ± 3.29	90.42 ± 3.33
MSE (m) ↓	$\mathbf{0.05 \pm 0.04}$	0.12 ± 0.12	0.11 ± 0.11
$\%\,\lvert J_\phi\rvert \leq 0$ ↓	0.11 ± 0.15	0.01 ± 0.05	$\mathbf{0.00 \pm 0.03}$
Time [ms] ↓	**8.06**	9.62	27.89

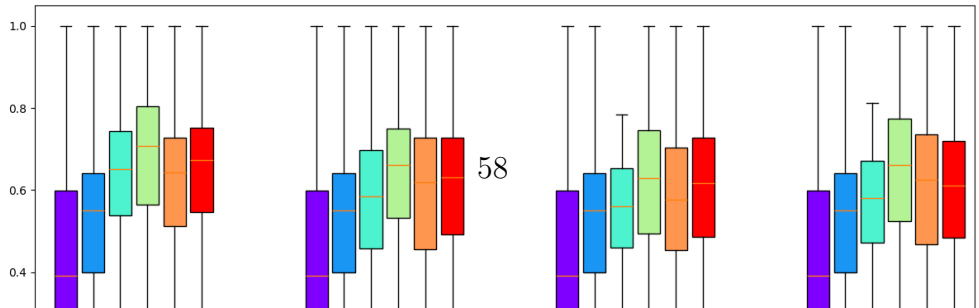
and the RV (far lower than the baseline).
VoxelMorph has the highest values for the RV compared to the other labels being close the *Fourier-Net* variant for $R = 0$, but falling off for the subsampled data, even performing worse than the baseline. It is surpasses the baseline, *NiftyReg* and even *Fourier-Net+* on $R = 0$ for the myocardium, but again loses performance for the subsampled data. This trend also continues for the LV were *VoxelMorph* has values similar to the baseline, but higher than *NiftyReg*, however ends up falling behind the baseline.
Fourier-Net consistently has the highest values for the challenging myocardium, however it is only about as good as *Fourier-Net+* and *4xFourier-Net+* for the LV and RV (even having lower values than the baseline for $R = 8$ on the latter).
Fourier-Net+ performs well for the LV for all reduction factors, although it is surpassed by *Fourier-Net* on $R = 0$ and by *4xFourier-Net+* for $R = 4$ and $R = 10$. For the challenging myocardium *Fourier-Net+* again has good values, but is surpassed by *Fourier-Net* for all reduction factors and only manages to surpass *4xFourier-Net+* for $R = 10$. For the RV *Fourier-Net+* is among the best methods, having higher values than *Fourier-Net* and *4xFourier-Net+* for $R = 8$.
4xFourier-Net+ consistently has the best median value for the LV and is close to having the highest value for the RV. Only on the myocardium it has consistently lower values than *Fourier-Net*.

Table 4.8: Test results for *NiftiReg* (NR), *VoxelMorph* (VM), *Fourier-Net* (F-Net), *Fourier-Net+* (F-Net+) and *4xFourier-Net+* (4xF-Net+) on the *ACDC* test data from fully sampled ($R = 0$) to $R = 10$ with an unaligned baseline for comparison. The best results for each metric and subsampling are highlighted in blue, while values worse than the unaligned baseline are marked with red.

	Methods	Metrics					
		% DICE ↑	% DICE* ↑	% SSIM ↑	MSE (m) ↓	$\%\,\lvert J_\phi\rvert \leq 0$ ↓	Time [s] ↓
$R = 0$	Baseline	70.85 ± 18.27	60.35 ± 25.24	86.39 ± 4.08	0.33 ± 0.23	-	-
	NR	70.74 ± 13.77	57.56 ± 18.86	91.26 ± 3.18	1.94 ± 1.61	0.00 ± 0.02	122.52
	VM	75.84 ± 13.46	68.25 ± 18.36	93.53 ± 3.30	0.06 ± 0.04	1.00 ± 0.57	0.1845
	F-Net	78.31 ± 13.96	71.17 ± 18.99	91.53 ± 3.49	0.09 ± 0.07	0.24 ± 0.25	0.1918
	F-Net+	76.88 ± 13.86	67.86 ± 19.06	88.67 ± 3.88	0.20 ± 0.15	0.02 ± 0.08	0.0893
	4xF-Net+	77.90 ± 13.92	69.21 ± 19.02	88.89 ± 4.05	0.19 ± 0.14	0.00 ± 0.02	0.3262
$R = 4$	Baseline	70.85 ± 18.27	60.35 ± 25.24	76.80 ± 11.02	0.28 ± 0.19	-	-
	NR	69.89 ± 13.73	56.47 ± 17.86	86.03 ± 6.41	0.15 ± 0.14	0.12 ± 0.15	80.08
	VM	71.78 ± 14.57	63.15 ± 18.98	90.73 ± 4.72	0.04 ± 0.03	1.00 ± 1.10	0.1264
	F-Net	76.55 ± 13.89	67.90 ± 19.44	82.93 ± 8.12	0.10 ± 0.06	0.19 ± 0.23	0.1006
	F-Net+	76.20 ± 13.57	66.30 ± 19.01	77.74 ± 10.69	0.19 ± 0.14	0.01 ± 0.03	0.0294
	4xF-Net+	77.23 ± 13.73	67.67 ± 18.97	77.96 ± 10.57	0.18 ± 0.13	0.00 ± 0.02	0.1131
$R = 8$	Baseline	70.85 ± 18.27	60.35 ± 25.24	85.35 ± 4.43	0.22 ± 0.17	-	-
	NR	70.04 ± 13.42	56.37 ± 17.63	91.07 ± 2.80	0.12 ± 0.13	0.08 ± 0.10	88.36
	VM	71.51 ± 14.22	62.17 ± 18.80	94.17 ± 2.80	0.03 ± 0.02	1.00 ± 0.96	0.1973
	F-Net	75.52 ± 13.69	66.86 ± 18.97	89.51 ± 3.34	0.06 ± 0.04	0.27 ± 0.29	0.2404
	F-Net+	75.76 ± 13.43	65.45 ± 18.63	86.30 ± 4.18	0.13 ± 0.11	0.03 ± 0.11	0.1482
	4xF-Net+	76.56 ± 13.47	66.81 ± 18.77	86.29 ± 4.23	0.13 ± 0.11	0.00 ± 0.02	0.5283
$R = 10$	Baseline	70.85 ± 18.27	60.35 ± 25.24	89.17 ± 3.58	0.20 ± 0.17	-	-
	NR	70.40 ± 13.34	56.61 ± 17.71	93.47 ± 2.37	0.11 ± 0.13	0.06 ± 0.08	47.44
	VM	71.89 ± 13.94	62.22 ± 18.58	95.85 ± 2.58	0.02 ± 0.02	1.00 ± 0.96	0.0577
	F-Net	76.87 ± 13.97	68.24 ± 19.27	92.77 ± 2.74	0.05 ± 0.04	0.11 ± 0.15	0.0296
	F-Net+	76.10 ± 13.83	66.27 ± 19.20	90.32 ± 3.29	0.12 ± 0.12	0.01 ± 0.05	0.0059
	4xF-Net+	76.98 ± 13.57	67.40 ± 18.90	90.42 ± 3.33	0.11 ± 0.11	0.00 ± 0.03	0.0275



(a) Boxplot of the Dice scores for the LV cavity.



Qualitative Results

In order to explain the differences observed between the methods previously, a visual examination can be used. There, one can look at the warped images and segmentations as well as the generated displacement fields visualized in Figure 4.2. For all acceleration factors, an example moving and fixed image can be compared to the warped images by the different methods. For further analysis, the corresponding segmentations (with corresponding Dice scores) and displacement fields generated by the methods are given.
For $R = 0$, *Fourier-Net+* and *4xFourier-Net+* have very localized displacements, centered on the cardiac region, leading to a very smooth warped segmentation compared to *Fourier-Net* and *VoxelMorph* where the cardiac labels mix in some regions. Despite this, *Fourier-Net* still achieves the highest Dice score with an 16% increase compared to the baseline. While *NiftyReg* does produce a very smooth warped segmentation the actual displacements are very small and not localized on the cardiac region leading to the lowest Dice score of all methods.
For $R = 4$, the image artifacts due to the subsampling are very apparent. The segmentations for the moving and fixed image obviously remain the same. *Fourier-Net+* and *4xFourier-Net+* again have very smooth segmentations, however the displacements reflect the difficulties of adapting to subsampled data in becoming slightly less local. Surprisingly, *NiftyReg* actually has a more localized displacement compared to the fully sampled data, however the segmentation looks less smooth and the Dice score is slightly lower. Somewhat similar, *VoxelMorph* also has a more localized displacement leading to a smoother segmentation and a higher Dice score. The results of *Fourier-Net* look overall very similar to the fully sampled case, perhaps with a bit more localized displacement leading to a smoother segmentation, although this does not lead to a better Dice Score.
For $R = 8$, the displacements for all methods become more global due to the strong presence of image artifacts, however the Dice scores are higher than before for all methods. *NiftyReg* even outperforms *VoxelMorph* for the first time. This trend does not continue for $R = 10$ however, were *NiftyReg* is again far behind all other methods in terms of Dice with a very much more global displacement. *VoxelMorph* seems to compensate more for the rippling artifacts present in the fixed image than for the change in the cardiac region as seen in the displacement. *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* do not share this behavior although their displacements also become more global and less focused on the cardiac region. The latter network has the highest Dice score for the first time managing an increase in Dice of about 16% compared to the baseline for this specific case despite the heavy subsampling showing the robustness of the network.

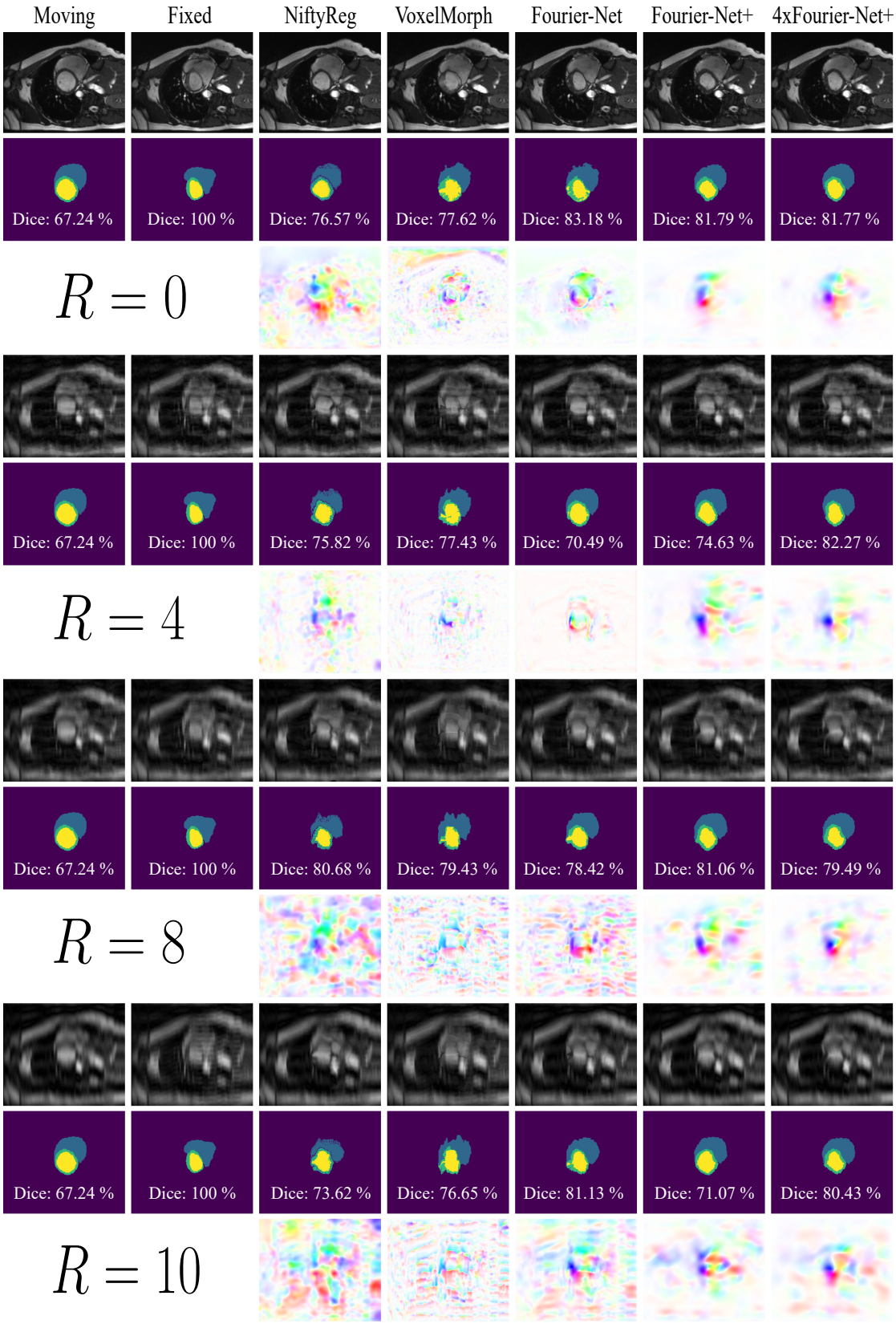


Figure 4.2: Examples of warped images, segmentations and flow fields for *NiftiReg*, *VoxelMorph*, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* together with the original image pair from the fully sampled ($R = 0$) and subsampled ($R = 4$, $R = 8$, $R = 10$) *ACDC* test data.

4.3 Integration into a Motion-Compensated Reconstruction Pipeline

After assessing the registration performance of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+*, a final down-stream task to gauge the applicability of these networks was conducted in the form of an motion-compensated reconstruction pipeline where the networks are used to correct the moment between frames.

4.3.1 K-Space Line Swapping

As discussed in section 3.3.3, the reconstruction pipeline was first tested with motion-corrupted data using k-space line swapping described in section 3.2.3. The results are shown in Table 4.9 with blue marking the best results per metric (not present if all methods have the same performance) and red marking worse results than the baseline. For $z = 16$, performance decreases with higher acceleration. For $R = 4$, *Fourier-Net* has the highest HaarPSI and SSIM values, while *VoxelMorph* has the highest PSNR. *Fourier-Net+* on the other hand, has the lowest values for these metrics. All methods as well as the baseline have the same mean MSE value. Only for the third decimal point in the standard deviation a small difference between the different methods is visible (note that the MSE values are already multiplied by a factor of 100). For $R = 8$, *Fourier-Net* again has the highest value for the HaarPSI, followed by *Fourier-Net+*, as well as the highest SSIM value, followed by *4xFourier-Net+* which also has the highest PSNR. While the MSE values are again very close, *4xFourier-Net+* has a slightly lower mean value than the other methods. For $R = 10$, *4xFourier-Net+* performs best for all metrics with *Fourier-Net+* having the same mean MSE value. *VoxelMorph* and *Fourier-Net* perform worse than baseline for the HaarPSI with the latter also having lower PSNR and SSIM values than the baseline.
For $z = 32$ and $R = 4$, *Fourier-Net* performs best for all metrics, while *4xFourier-Net+* performs worse than baseline for PSNR, SSIM and MSE. For $R = 8$, *Fourier-Net+* has the highest HaarPSI, PSNR and SSIM values followed by *Fourier-Net*. The mean MSE value is again the same for all methods and slightly lower than the baseline. For $R = 10$, *Fourier-Net+* again performs best for all metrics. Only *Fourier-Net* has a lower SSIM than the baseline.

Table 4.9: Reconstruction results *VoxelMorph*, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* on the *CMRxRecon* test data for $R = 4$, $R = 8$ and $R = 10$ as well as an baseline without motion-correction. The best results for each metric and subsampling are highlighted in blue, while values worse than the unaligned baseline are marked with red.

	Methods	Motion-correction for $z = 16$ swapped k-space lines			
		% HaarPSI ↑	PSNR [dB] ↑	% SSIM ↑	MSE (m) ↓
$R = 4$	Baseline	56.884 ± 8.129	28.703 ± 2.388	78.079 ± 5.887	0.160 ± 0.125
	VoxelMorph	56.909 ± 8.118	28.709 ± 2.404	78.090 ± 5.903	0.160 ± 0.126
	Fourier-Net	56.918 ± 8.160	28.706 ± 2.401	78.123 ± 5.893	0.160 ± 0.126
	Fourier-Net+	56.854 ± 8.158	28.699 ± 2.399	78.071 ± 5.951	0.160 ± 0.126
	4xFourier-Net+	56.903 ± 8.109	28.702 ± 2.380	78.085 ± 5.877	0.160 ± 0.125
$R = 8$	Baseline	53.318 ± 7.641	28.104 ± 2.383	77.311 ± 5.934	0.183 ± 0.139
	VoxelMorph	53.342 ± 7.687	28.110 ± 2.405	77.341 ± 5.900	0.183 ± 0.139
	Fourier-Net	53.394 ± 7.681	28.128 ± 2.398	77.416 ± 5.927	0.183 ± 0.138
	Fourier-Net+	53.388 ± 7.664	28.118 ± 2.404	77.398 ± 5.906	0.183 ± 0.138
	4xFourier-Net+	53.376 ± 7.688	28.133 ± 2.398	77.402 ± 5.936	0.182 ± 0.138
$R = 10$	Baseline	52.212 ± 7.388	27.906 ± 2.364	77.175 ± 5.886	0.191 ± 0.141
	VoxelMorph	52.211 ± 7.393	27.907 ± 2.368	77.194 ± 5.863	0.191 ± 0.140
	Fourier-Net	52.192 ± 7.361	27.895 ± 2.362	77.160 ± 5.815	0.191 ± 0.139
	Fourier-Net+	52.240 ± 7.379	27.924 ± 2.357	77.244 ± 5.866	0.189 ± 0.139
	4xFourier-Net+	52.272 ± 7.330	27.931 ± 2.349	77.262 ± 5.816	0.189 ± 0.137
		Motion-correction for $z = 32$ swapped k-space lines			
$R = 4$	Baseline	52.711 ± 7.673	27.506 ± 2.180	74.379 ± 5.869	0.203 ± 0.130
	VoxelMorph	52.747 ± 7.658	27.512 ± 2.188	74.349 ± 5.900	0.203 ± 0.129
	Fourier-Net	52.802 ± 7.708	27.536 ± 2.197	74.421 ± 5.937	0.202 ± 0.128
	Fourier-Net+	52.725 ± 7.724	27.522 ± 2.198	74.366 ± 5.892	0.203 ± 0.129
	4xFourier-Net+	52.711 ± 7.718	27.500 ± 2.205	74.353 ± 5.986	0.204 ± 0.132
$R = 8$	Baseline	50.071 ± 7.259	27.153 ± 2.199	74.033 ± 5.844	0.221 ± 0.138
	VoxelMorph	50.091 ± 7.301	27.164 ± 2.198	74.060 ± 5.890	0.220 ± 0.138
	Fourier-Net	50.152 ± 7.325	27.178 ± 2.202	74.094 ± 5.826	0.220 ± 0.140
	Fourier-Net+	50.159 ± 7.362	27.196 ± 2.221	74.119 ± 5.853	0.220 ± 0.141
	4xFourier-Net+	50.128 ± 7.276	27.165 ± 2.201	74.054 ± 5.842	0.220 ± 0.137
$R = 10$	Baseline	49.193 ± 7.022	27.013 ± 2.170	73.971 ± 5.788	0.227 ± 0.143
	VoxelMorph	49.241 ± 7.047	27.026 ± 2.179	73.976 ± 5.795	0.227 ± 0.142
	Fourier-Net	49.255 ± 7.011	27.024 ± 2.168	73.952 ± 5.793	0.227 ± 0.141
	Fourier-Net+	49.296 ± 7.001	27.045 ± 2.172	74.028 ± 5.784	0.226 ± 0.140
	4xFourier-Net+	49.287 ± 7.066	27.032 ± 2.176	74.022 ± 5.820	0.227 ± 0.143

4.3.2 Non-Linear Lung Transformations

As discussed in section 3.3.3 a second test with simulated non-linear lung motion was conducted. Results are shown in Table 4.10 with blue marking the best results per metric. For $R = 4$, *Fourier-Net* has the highest HaarPSI, PSNR and SSIM value followed by *Fourier-Net+* and *4xFourier-Net+*. *Fourier-Net* also has the lowest MSE value again followed by *Fourier-Net+* and *4xFourier-Net+*. *VoxelMorph* performs better than the baseline, but stays behind *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+*. For $R = 8$, *Fourier-Net+* overtakes *Fourier-Net* for HaarPSI, PSNR, SSIM and MSE followed by *4xFourier-Net+*. *VoxelMorph* is again the weakest method. Overall, the metrics are very much affected by the increased subsampling as the HaarPSI is about 4% lower for the methods compared to $R = 4$. The PSNR, SSIM and MSE are not effected quite as much, but are also lower in the case of PSNR, SSIM by about 1% and the MSE is higher by about $0.03 \cdot 10^{-3}$. The results for $R = 10$ are very similar to $R = 8$. *Fourier-Net+* again has the highest HaarPSI, PSNR and SSIM followed by *4xFourier-Net+* and *Fourier-Net*. This time the decrease in metrics from $R = 10$ compared to $R = 8$ is not quite as large (about 0.5%, 0.2% and 0.1% respectively). *Fourier-Net+* also has the lowest MSE again followed by *4xFourier-Net+* similar to $R = 8$ as the metrics barely changed.

Table 4.10: Reconstruction results *VoxelMorph*, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* on the *CMRxRecon* test data for $R = 4$, $R = 8$ and $R = 10$ as well as an baseline without motion-compensation. The best results for each metric and subsampling are highlighted in blue, while values worse than the unaligned baseline are marked with red.

	Methods	Metrics			
		% HaarPSI ↑	PSNR [dB] ↑	% SSIM ↑	MSE (m) ↓
$R = 4$	Baseline	28.882 ± 3.802	23.291 ± 1.654	70.375 ± 4.467	0.504 ± 0.197
	VoxelMorph	54.177 ± 7.495	28.347 ± 2.369	79.873 ± 4.878	0.174 ± 0.138
	Fourier-Net	59.886 ± 8.876	29.399 ± 2.766	82.731 ± 5.296	0.141 ± 0.127
	Fourier-Net+	59.868 ± 8.824	29.303 ± 2.761	82.572 ± 5.037	0.146 ± 0.142
	4xFourier-Net+	59.884 ± 8.875	29.365 ± 2.764	82.704 ± 5.293	0.144 ± 0.141
$R = 8$	Baseline	28.572 ± 3.773	23.173 ± 1.686	70.603 ± 4.533	0.519 ± 0.207
	VoxelMorph	50.537 ± 7.213	27.596 ± 2.460	78.848 ± 5.121	0.209 ± 0.161
	Fourier-Net	55.326 ± 8.415	28.465 ± 2.846	81.383 ± 5.489	0.191 ± 0.129
	Fourier-Net+	55.406 ± 8.432	28.519 ± 2.799	81.447 ± 5.478	0.178 ± 0.162
	4xFourier-Net+	55.402 ± 8.463	28.508 ± 2.916	81.436 ± 5.329	0.181 ± 0.153
$R = 10$	Baseline	28.486 ± 3.766	23.132 ± 1.695	70.742 ± 4.436	0.524 ± 0.211
	VoxelMorph	49.072 ± 6.854	27.331 ± 2.417	78.756 ± 4.962	0.220 ± 0.163
	Fourier-Net	54.012 ± 8.046	28.217 ± 2.758	81.224 ± 5.323	0.193 ± 0.162
	Fourier-Net+	54.025 ± 8.053	28.228 ± 2.745	81.264 ± 5.331	0.189 ± 0.164
	4xFourier-Net+	54.016 ± 8.064	28.223 ± 2.691	81.236 ± 5.237	0.191 ± 0.148

Qualitative Results

Similar to the registration performance on the *ACDC* dataset in section 4.2, visual examples can give insights and reasons for trends observed in the metrics. These examples can be seen for *VoxelMorph*, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* in Figure 4.3. There, the reconstruction results using the different networks for motion-compensation and the motion-corrupted baseline are compared to the fully sampled ground truth image using error maps for all three subsampling factors $R = 4$, $R = 8$ and $R = 10$.
In the top row, the images for the ground truth, the subsampled and motion-corrupted baseline, as well as the reconstructions using *VoxelMorph*, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* are shown. In the top right corner of the images, the SSIM calculated in comparison to the ground truth is shown. It should be noted that all images were croped around the cardiac region for better visualization of the details in the images. Lastly, in the bottom row, the error maps show the differences between the images and the ground truth. The color bar on the left indicates the severity of the deviation going from blue (no/small difference) to yellow (large difference).
For $R = 4$, as seen before in Table 4.10, *Fourier-Net* has the highest SSIM. But what about the visual look? Starting with the motion-corrupted baseline, it is clear that the image is affected by artifacts. The cardiac region looks blurred and distorted with a big tear being visible. This is most like the result of the motion-corruption as the subsampling only leads to blurring and aliasing artifacts, which are also visible. Continuing with *VoxelMorph*, it is clear that the reconstruction has not been able to restore the image as parts of the cardiac region are still blurred and a small tear is also visible. This can also be seen in the error map where subsampling artifacts can still be observed. Despite the superior performance in terms of metrics, there are still some subsampling artifacts visible for *Fourier-Net*. The cardiac region, however, looks far better compared to *VoxelMorph*, though still slightly blurred when compared to the ground truth. Next up is *Fourier-Net+*, where more artifacts are again present. The cardiac region is again strongly blurred which can also be seen in the error map. *4xFourier-Net+* also shows some blurring but not as much as *Fourier-Net+*. Despite this the SSIM for the image is slightly lower.
For $R = 8$, *Fourier-Net+* has the highest SSIM followed by *4xFourier-Net+*. *VoxelMorph* again shows a lot of artifacts with a small tear ans strong blurring in the cardiac region similar to the baseline. *Fourier-Net* on the other hand has far less artifacts though the whole image still appears quite blurred. The image for *Fourier-Net+* looks less blurred however artifacts from the subsampling are still visible. *4xFourier-Net+* again looks more like *Fourier-Net* before with stronger blurring and failing to remove some of the motion artifacts. However, it still has a slightly better SSIM.
For $R = 10$, *Fourier-Net+* has the highest SSIM, although the value is lower than for $R = 8$. *VoxelMorph* shows artifacts similar to before, though without a tear in the cardiac region this time. *Fourier-Net* also has a lot of artifacts distorting the image similar to *VoxelMorph*. *Fourier-Net+* is mostly free of artifacts except for some blurring. *4xFourier-Net+* similar to *Fourier-Net* has large blurring artifacts in the cardiac region and the contrast seems to be slightly different.

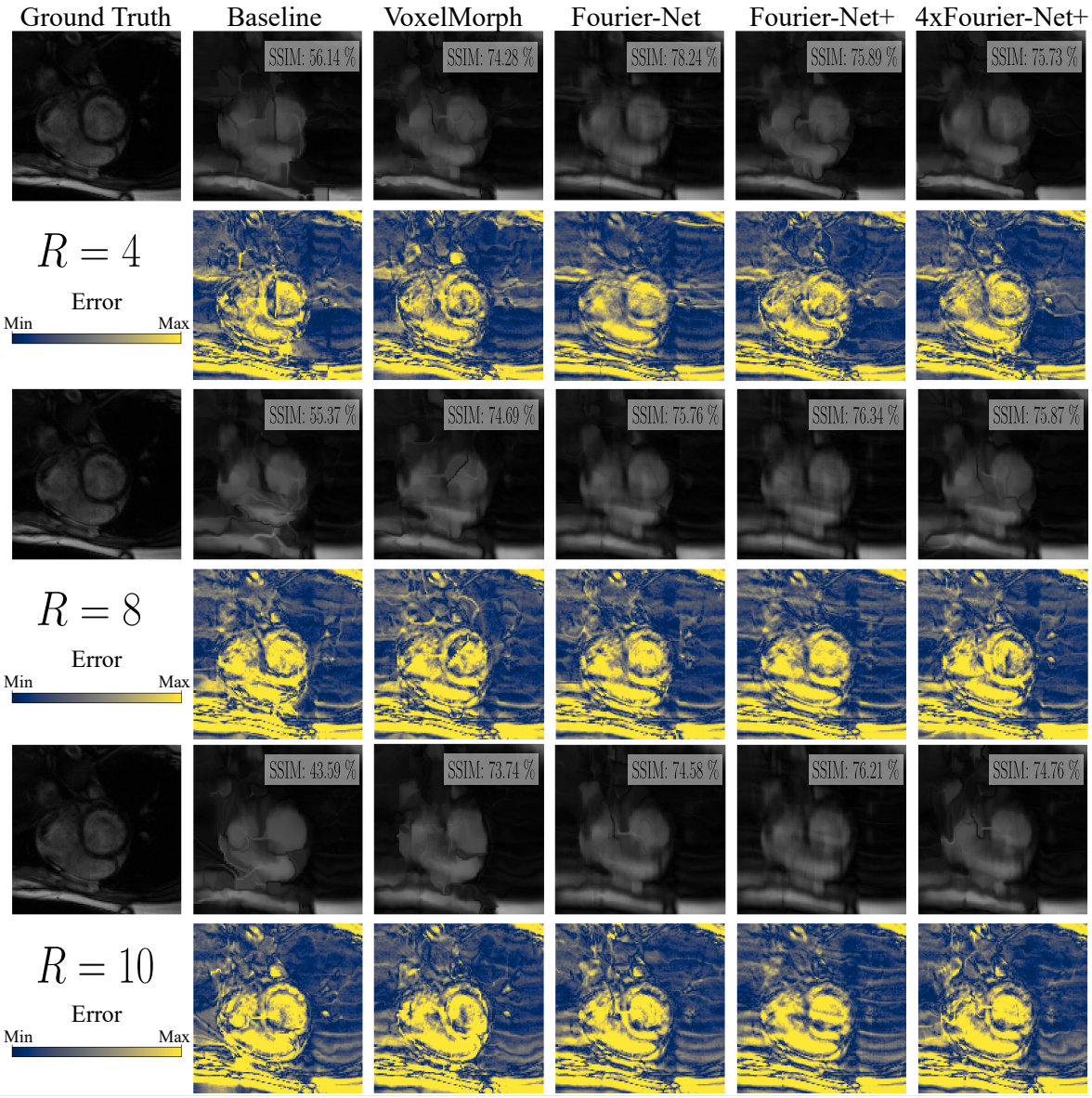


Figure 4.3: Visual results of the reconstruction pipeline for *VoxelMorph*, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* as well as the ground truth image and the motion-corrupted baseline on the *CMRxRecon* test data for $R = 4$, $R = 8$ and $R = 10$.

5

Discussion

After presenting the results of the experiments described in section 3.3, the findings and limitations are discussed in this chapter. First, the ablation studies, parameter tests and registration performance tests conducted on the *ACDC* dataset are discussed before assessing the evaluation of the reconstruction pipeline on the *CMRxRecon* dataset.

5.1 Ablation Studies and Parameter Tests

For the first tests, different ablation studies and parameter tests were conducted to find the best parameters for *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* on the *ACDC* dataset. We will discuss the impact of these tests and its effects on the parameter choices in subsequent experiments.

Fourier-Net versus Fourier-Net+

In a first comparative study, *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* were compared in terms of performance in section 4.1.1 together with their respective memory consumption. For the dense displacement variants, the diffeomorphic transform increases the performance slightly for *Fourier-Net* and *4xFourier-Net*, but decreases for *Fourier-Net+*. The times between the baseline and diffeomorphic versions are comparable. The results for the band-limited versions are very similar with *Fourier-Net* and *Fourier-Net+* performing slightly worse with the diffeomorphism, while *4xFourier-Net* performs better. These changes are not consistent with the dense displacement versions, however, the band-limited *4xFourier-Net* version might be an outlier, as all of the other models are not affected as much by the diffeomorphic transform. The percentage of non-positive Jacobian determinants decreases for all diffeomorphic variants, which was to be expected.
Overall, the dense displacement versions perform better in terms of Dice (with and without the background label), SSIM and MSE for all models. This is to be expected as the dense displacement versions can use all of the available k-space data for the registration task, while the band-limited versions have only a limited amount of k-space data, directly impacting and limiting their performance. The dense displacement are only slightly better in the SSIM and MSE metrics, however the difference is not quite as large as with the Dice scores. As the differences between the frames is rather small, except for the cardiac region, the weaker registration performance does not impact this

metric as much as the Dice score, which is focused on the moving cardiac region.
Only in terms of the percentage of non-positive Jacobian determinants does the band-limited displacement actually perform better as the band-limiting leads to a smoother deformation. However, the difference is still very small, especially for the diffeomorphic versions, and it could be argued that a better registration performance at the cost of some image folding is an acceptable trade-off.
As opposed to our intuition, the dense displacements variants are faster in terms of inference speed despite having the larger encoder, however all of the model are very fast (7 ms up to 33.5 ms). Lastly, the memory consumption needs to be addressed. The dense displacement variants have a larger encoder as discussed before and thus have more parameters which is very noticeable for the otherwise very efficient *Fourier-Net+* and *4xFourier-Net* (about 5 times more). The number of Mult-Adds and the amount of total memory are more than doubled (almost tripled for *4xFourier-Net*) for the dense displacement versions compared to those with a band-limited displacement across all models. This shows that a larger network produces better results, however for certain applications where efficiency is needed a smaller, but slightly less performant, network might be more beneficial.
To summarize, *Fourier-Net* performs best for the band-limited displacement, while *4xFourier-Net+* works better with a dense displacement. While the dense displacements overall provide the better performance, the memory cost is enormous. Utilizing a diffeomorphic transform does not generally lead to a better performance.

Starting Channel Size

One of the most important parameters for the networks is the channel size. A larger channel size means more features, but also a larger network. The results from section 4.1.2 show that memory efficiency and registration performance for *Fourier-Net+* and *4xFourier-Net+* cannot be optimized at the same time. There is a clear trade-off between the two. For both *Fourier-Net+* and *4xFourier-Net+* a clear increase in performance with larger channel sizes can be observed as more features are utilized. The percentage of non-positive Jacobian determinants also decreases as the registration performance increases, likely due to it having more features enabling the prediction of better displacements. The inference time is not impacted by channel size. The memory however increases exponentially when doubling the channel sizes as all layers are effected, not just the starting one as the name might suggest. Thus, the number of parameters drastically increases with starting size leading to an increase in Mult-Add and thus overall memory.
To summarize, as the channel sizes increase, making the network larger, the performance also increases. While this effect is quite large at the beginning, it plateaus rather quickly. This trade-off and the diminishing returns imply that there is a sweet spot where the network is still efficient while maintaining a good performance. The starting channel size 16 was thus chosen for the following experiments.

Fourier-Transform Crop Size

The FT crop is specific to *Fourier-Net+* and *4xFourier-Net+*, however, the size of the crop has a mayor impact on the input images as a smaller crop leads to more compressed images, but also better memory usage. Similarly to channel size before, the results in section 4.1.3 indicate that there is no sweet-spot to be found that maximizes both memory efficiency and registration performance. The latter decreases drastically with a smaller crop size. This is to be expected as the smaller FT drop compresses the images more, making an accurate registration more difficult. Another factor that negatively effects images for smaller crop sizes are potential over-folding artifacts due to the properties of the k-space. As a consequence, a reduction in crop size does not only lead to a smaller spatial resolution, but also a reduction of the FOV. This can cause the image to fold back into itself upon applying the iFFT leading to over-folding artifacts which can also severely hinder registration.
Interestingly, the percentage of non-positive Jacobian determinants actually decreases for smaller crop sizes, perhaps because the displacements on the compressed images are not as extreme and thus more smooth. Inference time also seems to decrease slightly with a smaller crop, although the time measurements are not quite consistent. The number of parameters does not change for different crop sizes as the networks themselves do not change, however the number of Mult-Adds and the total memory still change with the image size. Both decrease with a larger crop size as the image gets smaller. This effect is again not linear as a reduction in image size yields a larger reduction in memory for large crop sizes before flattening out for smaller ones. Overall, there is a trade-off between memory efficiency and registration performance similar to the channel size.

Comparison with VoxelMorph

As a comparison with a very common unsupervised registration network, the registration performance and memory imprint of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* was compared in section 4.1.4 to *VoxelMorph*. *Fourier-Net* has the best registration performance as indicated by the Dice score, followed by *4xFourier-Net+* as expected. *VoxelMorph*, however, does perform best in terms of SSIM and MSE followed by *Fourier-Net*, so it seems that the dense network aligns the overall image well, but struggles to compensate the strong changes in the cardiac region between frames. *VoxelMorph* has the highest percentage of non-positive Jacobian determinants compared to the other models. This is perhaps caused by the dense displacement which causes a better overall alignment (as indicated by the good SSIM and MSE values) at the cost of some minor folding occurring. While *VoxelMorph* has the least amount of parameters and Mult-Adds it needs almost 40 MB of memory, perhaps due to the dense displacement. *Fourier-Net* is the largest model with a total memory of 90 MB. Overall, *4xFourier-Net+* is the only model that is truly more efficient and has a better performance in terms of Dice than *VoxelMorph*. *Fourier-Net+* is more efficient, but not necessarily better in terms of Dice score, while *Fourier-Net* is far superior in reg-

istration performance (as measured by the Dice score), but less efficient in terms of memory consumption.

Dense Displacements

As a final ablation study, the difference between a band-limited and a dense displacement in *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* was tested in section 4.1.5. The results for the band-limited networks are consistently worse (or in the best case just as good) compared to their dense versions across all acceleration factors for Dice, SSIM and MSE. The band-limiting of the displacement helps with avoiding folding and in most cases decrease inference time, however at the expense of registration performance. The hypothesis that the advantage of the dense displacement compared to the band-limited displacement disappears for accelerated data does not seem to hold as the dense networks outperform the band-limited ones, though the difference is small in some cases. In the end, the larger dense networks perform better, but also require more memory. Thus, similar to the previous experiments, a balance between performance and memory efficiency has to be found.

5.2 Registration Performance on Subsampled Data

In section 4.2 the registration performance of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* was compared to an unaligned baseline as well as *NiftyReg* and *VoxelMorph* for four different reduction factors. To summarize the results, *Fourier-Net+* is the fastest method for all acceleration levels while *NiftyReg* is far behind all other methods in execution time as it is not a deep learning based method. *VoxelMorph* performs best in terms of SSIM and MSE for all acceleration factors, however it also has the highest percentage of non-positive Jacobian determinants indicating potential folding artifacts. *4xFourier-Net+* performs best in terms of Dice with the background label except for $R = 0$ while consistently having the lowest percentage of non-positive Jacobian determinants with a mean value under 0.004% for all acceleration factors. *Fourier-Net+* performs well in terms of Dice (with and without the background label), however it does not manage to surpass the other *Fourier-Net* variants despite having the second lowest percentage of non-positive Jacobian determinants across the acceleration factors. *Fourier-Net* performs best in terms of Dice without the background label for all acceleration level as well as in Dice with the background for $R = 0$. *NiftyReg* consistently performs worse than the baseline in terms of Dice (with and without the background) across all acceleration factors while having decent SSIM and MSE values. The performance, in general, decreased for all methods for higher acceleration factors as seen in the Dice metric, however, perhaps counter-intuitively, the SSIM is highest and the MSE the lowest for the $R = 10$ baseline with only $R = 4$ breaking the trend. This shows once again that image similarity metrics are not completely objective when it comes to evaluating registration performance across different acceleration factors.

Cardiac Labels

With the examination of the Dice scores it becomes apparent that these vary widely for each cardiac label. The overall performance of the methods is best for the right ventricle and worst for the myocardium which seems to be an underlying principle of the data as the unaligned baseline exhibits the same behavior. Overall, *NiftyReg* is not heavily affected by the artifacts present in the subsampled data, but is also not able to perform better than the other methods consistently. *VoxelMorph* is able to outperform the baseline, *NiftyReg* and even *Fourier-Net+* on one occasion, but is severely hampered by the artifacts caused by the subsampling of the k-space thus limiting its usability for accelerated MRI applications. While *Fourier-Net* is effected by the subsampling the effect is far less severe compared to *VoxelMorph*, and *Fourier-Net* reaches a far better performance, especially for the challenging myocardium. The same is true for *Fourier-Net+*, which performs better on the right ventricle, and *4xFourier-Net+*, which performs better for the left ventricle.

Qualitative Results

Lastly, the visual examination needs to be discussed. *Fourier-Net+* and *4xFourier-Net+* overall have very localized displacements, which centered on the cardiac region. This leads to very smooth warped segmentations even for highly accelerated images. *Fourier-Net*, *VoxelMorph* and *NiftyReg* on the contrary have more diffuse displacement leading to cardiac labels mixing in some cases. Thus those networks have learned a less localized transformation, which becomes more pronounced for higher reduction factors. While *Fourier-Net+* and *4xFourier-Net+* also experience this effect to some extend, they are far more robust than the other methods. In the worst case, *VoxelMorph* and *NiftyReg* try to align rippling and ringing artifacts in the images that differ between moving and fixed images. This is of course not desirable as the movement between the frames happens mostly in the cardiac region with the rest of the frames remaining mostly stationary. It should be noted however that a smoother displacement does not necessarily lead to a better Dice score.

5.3 Integration into a Motion-Compensated Reconstruction Pipeline

As a final downstream task, the usage of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* as part of an motion-compensated reconstruction pipeline was evaluated.

K-Space Line Swapping

The reconstruction pipeline was tested using two different methods for simulating motion. The first was motion/mistriggering simulated by swapping lines in the k-space. As expected, the results from section 4.3.1 show that $z = 16$ is easier to compensate than $z = 32$ as the latter contains more artifacts. In both cases, performance also

decreases for higher acceleration as expected. Overall, the results for $z = 32$ are far easier to interpret. *Fourier-Net* performs best for $R = 4$ with *4xFourier-Net+* underperforming, while *Fourier-Net+* performs best for $R = 8$ and $R = 10$. For $z = 16$, the results are far less cohesive. *Fourier-Net* performs best in HaarPSI and SSIM for $R = 4$ and $R = 8$, but underperforms for $R = 10$ while *4xFourier-Net+* is best in PSNR and MSR for $R = 8$ and performs best across all metrics for $R = 10$. *Fourier-Net+* underperforms for $R = 4$, but is decent for the other reduction factors. The MSE metrics overall is barely usable as the differences between the images are not captured by the metric (differences in mean values often in the sixth decimal point). However, even the other metrics are very similar for each reduction factor which implies that the reconstruction does not have a large impact on image quality. This is perhaps due to motion artifacts mostly blurring the images and still being too tame. Since the k-space line swapping occurs in the high-frequencies, the first idea was to register them in the band-limited space as it could be corruption agnostic (only low frequencies). But the evaluation is not easy so, as a second experiment, we choose another approach for simulating motion. These potential problem were addressed in the next test as extra frames with simulated lung movement were used.

Non-Linear Lung Transformations

As a second test, non-linear lung movement was simulated by deforming additional lung frames. The results can be seen in section 4.3.2. For $R = 4$, *Fourier-Net* performs best followed by *Fourier-Net+* and *4xFourier-Net+* while for $R = 8$ and $R = 10$ *Fourier-Net+* overtakes *Fourier-Net*. It is surprising that *Fourier-Net+* outperforms *4xFourier-Net+* in this experiment as the latter had previously shown better performance at the expense of memory efficiency. *VoxelMorph* is the weakest method though still quite a lot better than the baseline without any motion-compensation. Interestingly, the metrics, while being very much affected by the increased subsampling going $R = 8$ to $R = 4$, do not decrease quite as much going from $R = 8$ to $R = 10$. This is probably due to the reduction factor only increasing by 2 compared to the 4 for the step before.

Qualitative Results

In the visual examination is becomes clear that *VoxelMorph* is not able to combat the severe artifacts caused by the motion-corruption. Despite the superior performance in terms of SSIM, *Fourier-Net* still struggles to remove all the artifacts leading to blurring being present in the images caused by the subsampling. *Fourier-Net+* and *4xFourier-Net+* have similar problems, however it is surprising that these problems are not even more prevalent for increased subsampling. Perhaps the pipeline can removed most of the large artifacts caused by high subsampling factor, but struggles with finer image details. Interestingly, the SSIM is in some cases slightly higher for an image which does look slightly worse. This is the case for *4xFourier-Net+* compared to *Fourier-Net* on $R = 8$ and $R = 10$, where the latter looks better but has a slightly worse SSIM. This shows that the SSIM is also not a perfect metric, however, for example the MSE, which would provide a more direct comparison to the ground truth, barely changes for

the different reconstructed images as they are still very similar despite the artifacts as these mostly blur the image contents. It is of course important to keep in mind that this is just one of 21096 frames, which were evaluated in Table 4.10, so slight deviation are to be expected.

5.4 Efficiency and Performance Trade-offs

A reoccurring phenomenon of a trade-off between memory efficiency and performance. In both the starting channel results of section 4.1.2 and the FT crop size results in section 4.1.3 this trend is visible. Even further, in the comparison on subsampled data *Fourier-Net* and *4xFourier-Net+* often showed the best results in section 4.2. All of these points support the idea of a trade-off as the most efficient network variant, *Fourier-Net+*, often lacks behind it predecessor and its cascaded version. This makes sense as it contains an image crop that reduces the information the network receives as input. The whole reasoning behind *4xFourier-Net+* is to recover some of the lost performance by cascading multiple instances of *Fourier-Net+* with independent weights similar to a random forest or boosting approach.
This, however, is only possible because *Fourier-Net+* is so much more efficient than e.g. *Fourier-Net* and *VoxelMorph*. In the results from section 4.1.4, *Fourier-Net+* only needs about 5%, 27% and 12% of the memory that *Fourier-Net*, *4xFourier-Net+* and *VoxelMorph* need. *Fourier-Net+* is extremely efficient which can be imperative for certain applications like small portable devices. While this is definitely not the case for big MRI scanners, this aspect of *Fourier-Net+* should not be overlooked. Even a 4x cascaded Version in *4xFourier-Net+* only needs about 19% of the memory that *Fourier-Net* requires and still less than half compared to *VoxelMorph*. Despite this, it frequently outperforms the latter in e.g. registration performance and competes with *Fourier-Net* on many occasions.
It should also be noted that *Fourier-Net* does not always perform best despite its large memory footprint. In some instances, like the motion-compensated reconstruction, the added registration performance does not seem to be as much of a factor (especially for higher accelerations) and the usefulness of the memory efficiency that *Fourier-Net* provides becomes even more pronounced.

5.5 Limitations and Outlook

Now to the limitations of the experiments and results. There are multiple potential points for improvements of both the performance as well as the experimental setup. Lastly, future work is discussed in the outlook.

Extending the Scope of the Experiments and Network Training

While the registration performance of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* was very good on the *ACDC* dataset, beating traditional algorithms and other neural

networks, further comparisons with other more advanced networks such as LAPNet [27] on another cardiac dataset would be interesting. While three different subsampling factors were used for most of the experiment, even higher accelerations could be an challenging test for the networks adaptability to even more severe aliasing artifacts. In general, all network versions were specific to their acceleration. A test for the generalization of the learned properties might be informative as the networks could be applied to another acceleration as they were trained on. In the best case a network could be trained on all of the available data with different subsamplings to hopefully generalize the results for all accelerations. This would make an application to real world problems far easier as a single network could be used for a range of different data. Changes and additions to the network architecture and training of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* might also be interesting to further optimize the performance. However, a first step should be to extend the network and experiments to 3D as all previous experiments were conducted on only 2D slices although MRI data is naturally acquired as 3D volumes and this should ideally be reflected in the registration networks. This would not just improve the ease of use as the data needs less processing for usage, but it might also highlight the efficiency of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* as another spatial dimension makes the networks even larger leading to a bigger advantage for efficient methods.

Challenges of Evaluation

A problem with evaluating the reconstruction pipeline are the different networks training performance. As a new network is trained for each reduction factor, the performance might not just be impacted by the acceleration of the image, but also by the networks training. As the latter is non-deterministic in nature it can be hard in some cases to separate whether the network simply underperformed compared to its usual potential or whether the task is truly much more challenging. As an example, *Fourier-Net+* in some cases outperformed its cascaded version *4xFourier-Net+* which is unexpected. In most cases the smaller, more efficient *Fourier-Net+* would show a decrease in performance compared to the larger *Fourier-Net* and *4xFourier-Net+*, however, in some cases due to fluctuation in the training, this assumption would not hold. This further complicates accurately interpreting the results obtained in the different experiments on top on evaluation limitations inherent to the reconstruction pipeline due to the simulated motion. The k-space line swapping was difficult to evaluate as it is more of a motion-correction problem for a single image, rather than motion-compensation for multiple images. The latter is more suitable for the usage of image registration and was explored with the simulated non-linear lung motion. This however is also a very artificial test as the data is very synthetic and not quite realistic for actual lung movement. In the future, this test should repeated with real lung motion data preferably in 3D to be as close to the real-world application as possible.

6

Conclusion

In this thesis, the usage of *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+* for efficient and fast registration of subsampled MRI data was explored. This research question was approached from multiple different angles examining registration both directly, comparing the networks to traditional algorithms and other neural networks, as well as indirectly as part of a motion-compensated reconstruction pipeline.
After training the network on subsampled MRI data, multiple experiments were described to examine the research question. These included multiple ablation studies and parameter tests on the *ACDC* dataset followed by downstream tests on the *CMRxRecon* dataset. While the latter was for testing the applicability of the networks, the parameter tests were used to find optimal model parameters for *Fourier-Net*, *Fourier-Net+* and *4xFourier-Net+*. Then their registration performance was compared to *NiftyReg*, a traditional registration algorithm, and *VoxelMorph*, a state-of-the-art neural network.
For the second part of experiments, the three networks were used as part of an motion-compensated reconstruction pipeline. For this, the *CMRxRecon* dataset was used as it contained the needed k-space data and subsampling masks. We simulated motion for a downstream test using a motion-compensated reconstruction pipeline. Two different approaches were used to simulate mis-triggering and non-linear lung motion between frames. These were then reconstructed using a pipeline with neural networks for motion-compensation. The three networks were again compared to *VoxelMorph* in this downstream test.

Bibliography

[1] Robert W. Brown et al. *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. Wiley, Apr. 2014. ISBN: 9781118633953. DOI: 10.1002/9781118633953.

[2] Suraj D. Serai. "Basics of magnetic resonance imaging and quantitative parameters T1, T2, T2*, T1rho and diffusion-weighted imaging". In: *Pediatric Radiology* 52.2 (Apr. 2021), pp. 217–227. ISSN: 1432-1998. DOI: 10.1007/s00247-021-05042-7.

[3] Dilbag Singh et al. "Emerging Trends in Fast MRI Using Deep-Learning Reconstruction on Undersampled k-Space Data: A Systematic Review". In: *Bioengineering* 10.9 (Aug. 2023), p. 1012. ISSN: 2306-5354. DOI: 10.3390/bioengineering10091012.

[4] Yutong Chen et al. "AI-Based Reconstruction for Fast MRI—A Systematic Review and Meta-Analysis". In: *Proceedings of the IEEE* 110.2 (Feb. 2022), pp. 224–245. ISSN: 1558-2256. DOI: 10.1109/jproc.2022.3141367.

[5] Xiang Chen et al. "Deep learning in medical image registration". In: *Progress in Biomedical Engineering* (Dec. 2020). ISSN: 2516-1091. DOI: 10.1088/2516-1091/abd37c.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241. ISBN: 9783319245744. DOI: 10.1007/978-3-319-24574-4_28.

[7] Thomas Küstner et al. "Self-Supervised Motion-Corrected Image Reconstruction Network for 4D Magnetic Resonance Imaging of the Body Trunk". In: *APSIPA Transactions on Signal and Information Processing* 11.1 (2022). ISSN: 2048-7703. DOI: 10.1561/116.00000039.

[8] Aya Ghoul et al. "Attention-aware non-rigid image registration for accelerated MR imaging". In: *IEEE Transactions on Medical Imaging* (2024), pp. 1–1. ISSN: 1558-254X. DOI: 10.1109/tmi.2024.3385024.

[9] Jiazhen Pan et al. "Motion-Compensated MR CINE Reconstruction With Reconstruction-Driven Motion Estimation". In: *IEEE Transactions on Medical Imaging* 43.7 (July 2024), pp. 2420–2433. ISSN: 1558-254X. DOI: 10.1109/tmi.2024.3364504.

[10] Grant Haskins, Uwe Kruger, and Pingkun Yan. "Deep learning in medical image registration: a survey". In: *Machine Vision and Applications* 31.1–2 (Jan. 2020). ISSN: 1432-1769. DOI: 10.1007/s00138-020-01060-x.

[11] Jesse Hamilton, Dominique Franson, and Nicole Seiberlich. "Recent advances in parallel imaging for MRI". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 101 (Aug. 2017), pp. 71–95. ISSN: 0079-6565. DOI: 10.1016/j.pnmrs.2017.04.002.

[12] Qi Liu et al. "Comparison of high-resolution MRI with CT angiography and digital subtraction angiography for the evaluation of middle cerebral artery atherosclerotic steno-occlusive disease". In: *The International Journal of Cardiovascular Imaging* 29.7 (May 2013), pp. 1491–1498. ISSN: 1573-0743. DOI: 10.1007/s10554-013-0237-3.

[13] Wajiha Bano. *Model-based reconstruction of accelerated quantitative magnetic resonance imaging (MRI)*. en. 2019. DOI: 10.7488/ERA/61.

[14] J. Hennig. "K-space sampling strategies". In: *European Radiology* 9.6 (July 1999), pp. 1020–1031. ISSN: 1432-1084. DOI: 10.1007/s003300050788.

[15] George Yiasemis et al. *On Retrospective k-space Subsampling schemes For Deep MRI Reconstruction*. 2023. DOI: 10.48550/ARXIV.2301.08365.

[16] C.E. Shannon. "Communication in the Presence of Noise". In: *Proceedings of the IRE* 37.1 (Jan. 1949), pp. 10–21. ISSN: 0096-8390. DOI: 10.1109/jrproc.1949.232969.

[17] Maxim Zaitsev, Julian Maclaren, and Michael Herbst. "Motion artifacts in MRI: A complex problem with many partial solutions". In: *Journal of Magnetic Resonance Imaging* 42.4 (Jan. 2015), pp. 887–901. ISSN: 1522-2586. DOI: 10.1002/jmri.24850.

[18] Lu Lin et al. "Free-breathing cardiac cine MRI with compressed sensing real-time imaging and retrospective motion correction: clinical feasibility and validation". In: *European Radiology* 33.4 (Nov. 2022), pp. 2289–2300. ISSN: 1432-1084. DOI: 10.1007/s00330-022-09210-7.

[19] Junyu Chen et al. *A survey on deep learning in medical image registration: new technologies, uncertainty, evaluation metrics, and beyond*. 2023. DOI: 10.48550/ARXIV.2307.15615.

[20] Freddy Odille et al. "Generalized Reconstruction by Inversion of Coupled Systems (GRICS) applied to free-breathing MRI". In: *Magnetic Resonance in Medicine* 60.1 (June 2008), pp. 146–157. ISSN: 1522-2594. DOI: 10.1002/mrm.21623.

[21] John Ashburner. "A fast diffeomorphic image registration algorithm". In: *NeuroImage* 38.1 (Oct. 2007), pp. 95–113. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2007.07.007.

[22] Tom Vercauteren et al. "Diffeomorphic demons: Efficient non-parametric image registration". In: *NeuroImage* 45.1 (Mar. 2009), S61–S72. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2008.10.040.

[23] Marc Modat et al. "Fast free-form deformation using graphics processing units". In: *Computer Methods and Programs in Biomedicine* 98.3 (June 2010), pp. 278–284. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2009.09.002.

[24] Christopher Gilliam, Thomas Kustner, and Thierry Blu. "3D motion flow estimation using local all-pass filters". In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE, Apr. 2016. DOI: 10.1109/isbi.2016.7493264.

[25] Miaomiao Zhang and P. Thomas Fletcher. "Fast Diffeomorphic Image Registration via Fourier-Approximated Lie Algebras". In: *International Journal of Computer Vision* 127.1 (May 2018), pp. 61–73. ISSN: 1573-1405. DOI: 10.1007/s11263-018-1099-x.

[26] Yabo Fu et al. "Deep learning in medical image registration: a review". In: *Physics in Medicine & Biology* 65.20 (Oct. 2020), 20TR01. ISSN: 1361-6560. DOI: 10.1088/1361-6560/ab843e.

[27] Thomas Küstner et al. "LAPNet: Non-Rigid Registration Derived in k-Space for Magnetic Resonance Imaging". In: *IEEE Transactions on Medical Imaging* 40.12 (Dec. 2021), pp. 3686–3697. ISSN: 1558-254X. DOI: 10.1109/tmi.2021.3096131.

[28] Jian Wang and Miaomiao Zhang. *DeepFLASH: An Efficient Network for Learning-based Medical Image Registration*. 2020. DOI: 10.48550/ARXIV.2004.02097.

[29] Jing Zou et al. "A review of deep learning-based deformable medical image registration". In: *Frontiers in Oncology* 12 (Dec. 2022). ISSN: 2234-943X. DOI: 10.3389/fonc.2022.1047215.

[30] Guha Balakrishnan et al. "VoxelMorph: A Learning Framework for Deformable Medical Image Registration". In: *IEEE Transactions on Medical Imaging* 38.8 (Aug. 2019), pp. 1788–1800. ISSN: 1558-254X. DOI: 10.1109/tmi.2019.2897538.

[31] Jun Zhang. *Inverse-Consistent Deep Networks for Unsupervised Deformable Image Registration*. 2018. DOI: 10.48550/ARXIV.1809.03443.

[32] Tony C.W. Mok and Albert C.S. Chung. "Fast Symmetric Diffeomorphic Image Registration with Convolutional Neural Networks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. DOI: 10.1109/cvpr42600.2020.00470.

[33] Xi Jia et al. “Fourier-Net: Fast Image Registration with Band-Limited Deformation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 1. 2023, pp. 1015–1023.

[34] Xi Jia et al. *Fourier-Net+: Leveraging Band-Limited Representation for Efficient 3D Medical Image Registration*. 2023. DOI: 10.48550/ARXIV.2307.02997.

[35] M. 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.

[36] Anagha Deshmane et al. “Parallel MR imaging”. In: *Journal of Magnetic Resonance Imaging* 36.1 (June 2012), pp. 55–72. ISSN: 1522-2586. DOI: 10.1002/jmri.23639.

[37] Mark A. Griswold et al. “Generalized autocalibrating partially parallel acquisitions (GRAPPA)”. In: *Magnetic Resonance in Medicine* 47.6 (June 2002), pp. 1202–1210. ISSN: 1522-2594. DOI: 10.1002/mrm.10171.

[38] George Yiasemis et al. “Deep MRI reconstruction with radial subsampling”. In: *Medical Imaging 2022: Physics of Medical Imaging*. Ed. by Wei Zhao and Lifeng Yu. SPIE, Apr. 2022. DOI: 10.1117/12.2609876.

[39] Ilkay Oksuz et al. “Deep Learning-Based Detection and Correction of Cardiac MR Motion Artefacts During Reconstruction for High-Quality Segmentation”. In: *IEEE Transactions on Medical Imaging* 39.12 (Dec. 2020), pp. 4001–4010. ISSN: 1558-254X. DOI: 10.1109/tmi.2020.3008930.

[40] Qing Zou. “Motion-resolved 3D Pulmonary MRI Reconstruction using Sinusoidal Representation Networks”. In: *Current Medical Imaging Reviews* 20 (Jan. 2024). ISSN: 1573-4056. DOI: 10.2174/0115734056270732231204061123.

[41] Chiheb Trabelsi et al. *Deep Complex Networks*. 2017. DOI: 10.48550/ARXIV.1705.09792.

[42] Aya Ghoul et al. *Highly efficient non-rigid registration in k-space with application to cardiac Magnetic Resonance Imaging*. 2024. DOI: 10.48550/ARXIV.2410.18834.

[43] Thomas Küstner et al. “CINENet: deep learning-based 3D cardiac CINE MRI reconstruction with multi-coil complex-valued 4D spatio-temporal convolutions”. In: *Scientific Reports* 10.1 (Aug. 2020). ISSN: 2045-2322. DOI: 10.1038/s41598-020-70551-8.

[44] Christopher M. Kramer et al. “Standardized cardiovascular magnetic resonance imaging (CMR) protocols: 2020 update”. In: *Journal of Cardiovascular Magnetic Resonance* 22.1 (Jan. 2020), p. 17. ISSN: 1097-6647. DOI: 10.1186/s12968-020-00607-1.

[45] John P Ridgway. “Cardiovascular magnetic resonance physics for clinicians: part I”. In: *Journal of Cardiovascular Magnetic Resonance* 12.1 (Oct. 2010), p. 71. ISSN: 1097-6647. DOI: 10.1186/1532-429x-12-71.

[46] John D Biglands, Aleksandra Radjenovic, and John P Ridgway. “Cardiovascular magnetic resonance physics for clinicians: part II”. In: *Journal of Cardiovascular Magnetic Resonance* 14.1 (Jan. 2012), p. 78. ISSN: 1097-6647. DOI: 10.1186/1532-429x-14-66.

[47] P Lanzer et al. “ECG-synchronized cardiac MR imaging: method and evaluation.” In: *Radiology* 155.3 (June 1985), pp. 681–686. ISSN: 1527-1315. DOI: 10.1148/radiology.155.3.4001369.

[48] Pei G. Chew et al. “Feasibility and reproducibility of a cardiovascular magnetic resonance free-breathing, multi-shot, navigated image acquisition technique for ventricular volume quantification during continuous exercise”. In: *Quantitative Imaging in Medicine and Surgery* 10.9 (Sept. 2020), pp. 1837–1851. ISSN: 2223-4306. DOI: 10.21037/qims-20-117.

[49] Keyan Wang et al. “Prognosis in patients with coronary heart disease and breath-holding limitations: a free-breathing cardiac magnetic resonance protocol at 3.0 T”. In: *BMC Cardiovascular Disorders* 21.1 (Dec. 2021). ISSN: 1471-2261. DOI: 10.1186/s12872-021-02402-x.

[50] Dianna M. E. Bardo and Nicholas Rubert. “Radial sequences and compressed sensing in pediatric body magnetic resonance imaging”. In: *Pediatric Radiology* 52.2 (May 2021), pp. 382–390. ISSN: 1432-1998. DOI: 10.1007/s00247-021-05097-6.

[51] C. B. Ahn, J. H. Kim, and Z. H. Cho. “High-Speed Spiral-Scan Echo Planar NMR Imaging-I”. In: *IEEE Transactions on Medical Imaging* 5.1 (Mar. 1986), pp. 2–7. ISSN: 1558-254X. DOI: 10.1109/tmi.1986.4307732.

[52] Gary H. Glover and Adrian T. Lee. “Motion Artifacts in fMRI: Comparison of 2DFT with PR and Spiral Scan Methods”. In: *Magnetic Resonance in Medicine* 33.5 (May 1995), pp. 624–635. ISSN: 1522-2594. DOI: 10.1002/mrm.1910330507.

[53] Dwight G. Nishimura, Pablo Irarrazabal, and Craig H. Meyer. “A Velocity k-Space Analysis of Flow Effects in Echo-Planar and Spiral Imaging”. In: *Magnetic Resonance in Medicine* 33.4 (Apr. 1995), pp. 549–556. ISSN: 1522-2594. DOI: 10.1002/mrm.1910330414.

[54] P. C. LAUTERBUR. “Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance”. In: *Nature* 242.5394 (Mar. 1973), pp. 190–191. ISSN: 1476-4687. DOI: 10.1038/242190a0.

[55] Klaus Scheffler and Jürgen Hennig. “Frequency resolved single-shot MR imaging using stochastic k-space trajectories”. In: *Magnetic Resonance in Medicine* 35.4 (Apr. 1996), pp. 569–576. ISSN: 1522-2594. DOI: 10.1002/mrm.1910350417.

[56] Mark S. Cohen et al. “Sensory stimulation by time-varying magnetic fields”. In: *Magnetic Resonance in Medicine* 14.2 (May 1990), pp. 409–414. ISSN: 1522-2594. DOI: 10.1002/mrm.1910140226.

[57] C. L. G. Ham et al. “Peripheral nerve stimulation during MRI: Effects of high gradient amplitudes and switching rates”. In: *Journal of Magnetic Resonance Imaging* 7.5 (Sept. 1997), pp. 933–937. ISSN: 1522-2586. DOI: 10.1002/jmri.1880070524.

[58] Kai Xuan et al. “Learning MRI k-Space Subsampling Pattern Using Progressive Weight Pruning”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 178–187. ISBN: 9783030597139. DOI: 10.1007/978-3-030-59713-9_18.

[59] Klaas P. Pruessmann et al. “SENSE: Sensitivity encoding for fast MRI”. In: *Magnetic Resonance in Medicine* 42.5 (Nov. 1999), pp. 952–962. ISSN: 1522-2594. DOI: 10.1002/(sici)1522-2594(199911)42:5<952::aid-mrm16>3.0.co;2-s.

[60] Michael Lustig and John M. Pauly. “SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space”. In: *Magnetic Resonance in Medicine* 64.2 (June 2010), pp. 457–471. ISSN: 1522-2594. DOI: 10.1002/mrm.22428.

[61] Martin Uecker et al. “ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA”. In: *Magnetic Resonance in Medicine* 71.3 (May 2013), pp. 990–1001. ISSN: 1522-2594. DOI: 10.1002/mrm.24751.

[62] Dong Liang et al. “Accelerating SENSE using compressed sensing”. In: *Magnetic Resonance in Medicine* 62.6 (Sept. 2009), pp. 1574–1584. ISSN: 1522-2594. DOI: 10.1002/mrm.22161.

[63] Ricardo Otazo et al. “Combination of compressed sensing and parallel imaging for highly accelerated first-pass cardiac perfusion MRI”. In: *Magnetic Resonance in Medicine* 64.3 (Aug. 2010), pp. 767–776. ISSN: 1522-2594. DOI: 10.1002/mrm.22463.

[64] Ricardo Otazo, Emmanuel Candès, and Daniel K. Sodickson. “Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components: L+S Reconstruction”. In: *Magnetic Resonance in Medicine* 73.3 (Apr. 2014), pp. 1125–1136. ISSN: 0740-3194. DOI: 10.1002/mrm.25240.

[65] Lucilio Cordero-Grande et al. “Sensitivity Encoding for Aligned Multishot Magnetic Resonance Reconstruction”. In: *IEEE Transactions on Computational Imaging* 2.3 (Sept. 2016), pp. 266–280. ISSN: 2334-0118. DOI: 10.1109/tci.2016.2557069.

[66] D. Atkinson et al. “Automatic correction of motion artifacts in magnetic resonance images using an entropy focus criterion”. In: *IEEE Transactions on Medical Imaging* 16.6 (1997), pp. 903–910. ISSN: 0278-0062. DOI: 10.1109/42.650886.

[67] P. G. Batchelor et al. “Matrix description of general motion correction applied to multishot images”. In: *Magnetic Resonance in Medicine* 54.5 (Sept. 2005), pp. 1273–1280. ISSN: 1522-2594. DOI: 10.1002/mrm.20656.

[68] Melissa W. Haskell et al. “Network Accelerated Motion Estimation and Reduction (NAMER): Convolutional neural network guided retrospective motion correction using a separable motion model”. In: *Magnetic Resonance in Medicine* 82.4 (May 2019), pp. 1452–1461. ISSN: 1522-2594. DOI: 10.1002/mrm.27771.

[69] Muhammad Usman et al. “Retrospective Motion Correction in Multishot MRI using Generative Adversarial Network”. In: *Scientific Reports* 10.1 (Mar. 2020). ISSN: 2045-2322. DOI: 10.1038/s41598-020-61705-9.

[70] Anika Strittmatter, Anna Caroli, and Frank G. Zöllner. “A Multistage Rigid-Affine-Deformable Network for Three-Dimensional Multimodal Medical Image Registration”. In: *Applied Sciences* 13.24 (Dec. 2023), p. 13298. ISSN: 2076-3417. DOI: 10.3390/app132413298.

[71] Alda L. Tam et al. “Image-Guided Biopsy in the Era of Personalized Cancer Care: Proceedings from the Society of Interventional Radiology Research Consensus Panel”. In: *Journal of Vascular and Interventional Radiology* 27.1 (Jan. 2016), pp. 8–19. ISSN: 1051-0443. DOI: 10.1016/j.jvir.2015.10.019.

[72] A. M. Chen et al. “MRI-guided radiotherapy for head and neck cancer: initial clinical experience”. In: *Clinical and Translational Oncology* 20.2 (June 2017), pp. 160–168. ISSN: 1699-3055. DOI: 10.1007/s12094-017-1704-4.

[73] Bastien Rigaud et al. “Deformable image registration for radiation therapy: principle, methods, applications and evaluation”. In: *Acta Oncologica* 58.9 (June 2019), pp. 1225–1237. ISSN: 1651-226X. DOI: 10.1080/0284186x.2019.1620331.

[74] Wanni Xu, You-Lei Fu, and Dongmei Zhu. “ResNet and its application to medical image processing: Research progress and challenges”. In: *Computer Methods and Programs in Biomedicine* 240 (Oct. 2023), p. 107660. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2023.107660.

[75] Haonan Xiao et al. “A review of deep learning-based three-dimensional medical image registration methods”. In: *Quantitative Imaging in Medicine and Surgery* 11.12 (Dec. 2021), pp. 4895–4916. ISSN: 2223-4306. DOI: 10.21037/qims-21-175.

[76] Preetjot Kaur and Roopali Garg. “Towards Convolution Neural Networks (CNNs): A Brief Overview of AI and Deep Learning”. In: *Inventive Communication and Computational Technologies*. Springer Singapore, 2020, pp. 399–407. ISBN: 9789811501463. DOI: 10.1007/978-981-15-0146-3_38.

[77] Farhad Mortezapour Shiri et al. *A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU*. 2023. DOI: 10.48550/ARXIV.2305.17473.

[78] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980.

[79] Xi Jia et al. *U-Net vs Transformer: Is U-Net Outdated in Medical Image Registration?* 2022. DOI: 10.48550/ARXIV.2208.04939.

[80] Jingfan Fan et al. “BIRNet: Brain image registration using dual-supervised fully convolutional networks”. In: *Medical Image Analysis* 54 (May 2019), pp. 193–206. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.03.006.

[81] Junyu Chen et al. “TransMorph: Transformer for unsupervised medical image registration”. In: *Medical Image Analysis* 82 (Nov. 2022), p. 102615. ISSN: 1361-8415. DOI: 10.1016/j.media.2022.102615.

[82] Z. Wang et al. “Image Quality Assessment: From Error Visibility to Structural Similarity”. In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/tip.2003.819861.

[83] Adrian V. Dalca et al. “Unsupervised Learning for Fast Probabilistic Diffeomorphic Registration”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2018, pp. 729–738. ISBN: 9783030009281. DOI: 10.1007/978-3-030-00928-1_82.

[84] Vincent Arsigny et al. “A Log-Euclidean Framework for Statistics on Diffeomorphisms”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 924–931. ISBN: 9783540447085. DOI: 10.1007/11866565_113.

[85] Max Jaderberg et al. “Spatial Transformer Networks”. In: *ArXiv* abs/1506.02025 (2015). URL: https://api.semanticscholar.org/CorpusID:6099034.

[86] Olivier Bernard et al. “Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved?” In: *IEEE Transactions on Medical Imaging* 37.11 (Nov. 2018), pp. 2514–2525. ISSN: 1558-254X. DOI: 10.1109/tmi.2018.2837502.

[87] Chengyan Wang et al. *CMRxRecon: An open cardiac MRI dataset for the competition of accelerated image reconstruction*. 2023. DOI: 10.48550/ARXIV.2309.10836.

[88] P. B. Roemer et al. “The NMR phased array”. In: *Magnetic Resonance in Medicine* 16.2 (Nov. 1990), pp. 192–225. ISSN: 1522-2594. DOI: 10.1002/mrm.1910160203.

[89] Adrian V. Dalca et al. “Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces”. In: *Medical Image Analysis* 57 (Oct. 2019), pp. 226–236. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.07.006.

[90] Rafael Reisenhofer et al. “A Haar Wavelet-Based Perceptual Similarity Index for Image Quality Assessment”. In: (2016). DOI: 10.48550/ARXIV.1607.06140.