MRI Raw Data Reimagined

Complex-Valued Deep Learning for Direct k-Space Processing and Applications

MRI Raw Data Reimagined

Complex-Valued Deep Learning for Direct k-Space Processing and Applications

Dissertation Thesis

Moritz Rempe

September 16, 2025

Submitted in partial fulfillment of the requirements
for the degree of Doctor rerum naturalium (Dr. rer. nat.)

to the

Faculty of Physics
at Technische Universität Dortmund

1st Reviewer	Prof. Dr. Dr. Jens Kleesiek
2nd Reviewer	Prof. Dr. Kevin Kröninger
3rd Reviewer	Prof. Dr. Harald Quick

technische universität
dortmund

University Medicine Essen
Institute for Artificial Intelligence in Medicine

IMPRINT

MRI Raw Data Reimagined
Complex-Valued Deep Learning for
Direct k-Space Processing and Applications

Published by the Technische Universität Dortmund, Dortmund, Germany.

COLOPHON

This thesis was typeset using LaTeX and the `memoir` documentclass. It is based on Aaron Turon's thesis *Understanding and expressing scalable concurrency*[1] and the LaTeX implementation by Friedrich Wiemer[2].

The bibliography was processed by Biblatex. All graphics and plots are made with PGF/Ti*k*Z.

The body text is set 11/14pt (long primer) on a 26pc measure. The margin text is set 8/9pt (brevier) on a 12pc measure. Matthew Carter's Charter acts as both the text and display typeface. Monospaced text uses Jim Lyles's `Bitstream Vera Mono` ("Bera Mono").

[1] `http://aturon.github.io/academic/turon-thesis.pdf`

[2] `https://github.com/pfasante/phd_thesis`

The task is not so much to see what no one has yet seen;
but to think what nobody has yet thought,
about that which everybody sees.
—Arthur Schopenhauer

Abstract

Magnetic Resonance Imaging (MRI) is a well-established imaging modality in modern medicine, yet the common practice of discarding raw k-Space data after image reconstruction represents significant underutilization of valuable information. This k-Space, the Fourier-domain representation of the MR signal, inherently contains richer information than the final magnitude image alone, including crucial phase data. The focus on image-domain processing means this additional information is frequently lost. We hypothesize, that with this loss of information, the potential for more efficient or deeper insights into diagnostic tasks is often not realized.

This thesis confronts this challenge by proposing and investigating different hypotheses centered around the untapped potential of the k-Space. First, we hypothesize that it is not only feasible but advantageous to perform clinical tasks directly on complex-valued k-Space data, leveraging the richer information it contains. These clinical tasks include segmentations as well as classifications. Second, we hypothesize that the critical bottleneck of data scarcity in raw MRI deep learning can be addressed through the development of generative models that synthesize realistic MRI raw data.

To test these hypotheses, this thesis presents a cycle of interconnected projects. The first hypothesis is validated through "k-Strip", a novel complex-valued U-Net for direct k-Space segmentation. Applied to skull-stripping, it achieves competitive performance while enabling direct anonymization on non-human-readable data, while preserving valuable phase information. This is extended to clinical classification, where a model leveraging k-Space data for prostate cancer likelihood estimation demonstrates improved prediction accuracy, particularly under accelerated acquisition scenarios, promising faster and more robust diagnostic pathways.

To overcome the second hypothesis regarding data scarcity, which is slowing innovation in raw MRI deep learning, a novel complex-valued generative diffusion model named "PhaseGen" is developed. PhaseGen successfully synthesizes realistic MRI raw data, specifically generating phase information conditioned on readily available magnitude images, thereby enabling the creation of extensive synthetic datasets for pretraining. Its efficacy is validated through the improved generalization of k-Strip on real-world raw MRI data and by demonstrating that MRI reconstruction models can achieve performance comparable to full real-data training when augmented with PhaseGen-synthesized data, even with severely limited real-data availability (10-15%).

Finally, these methodologies are synergistically integrated into a publicly available de-identification tool. This software uniquely extends anonymization protocols to raw MRI data, encompassing both

metadata cleansing and k-Space domain feature removal using the PhaseGen-trained k-Strip model, thus facilitating safer data sharing and collaborative research.

Collectively, by confirming these hypotheses, this work provides a foundational framework and practical tools for harnessing the full potential of MRI k-Space through complex-valued deep learning, paving the way for more efficient and insightful medical imaging.

Kurzfassung

Die Magnetresonanztomographie (MRT) ist ein etabliertes bildgebendes Verfahren in der modernen Medizin, jedoch führt die gängige Praxis, Rohdaten aus dem k-Raum nach der Bildrekonstruktion zu verwerfen, zu einer erheblichen Unterausnutzung wertvoller Informationen. Dieser k-Raum, die Darstellung des MR-Signals im Fourier-Raum, enthält von Natur aus reichhaltigere Informationen als das reine Magnitudenbild, einschließlich entscheidender Phasendaten. Durch den Fokus auf die Verarbeitung in der Bilddomäne gehen diese zusätzlichen Informationen häufig verloren. Wir stellen die Hypothese auf, dass durch diesen Informationsverlust das Potenzial für effizientere oder tiefgreifendere diagnostische Einblicke oft nicht realisiert wird.

Diese Dissertation stellt sich dieser Herausforderung, indem sie verschiedene Hypothesen zum ungenutzten Potenzial des k-Raums aufstellt und untersucht. Erstens postulieren wir, dass es nicht nur machbar, sondern auch vorteilhaft ist, klinische Aufgaben wie Segmentierungen und Klassifikationen direkt an komplexwertigen k-Raum-Daten durchzuführen, um deren reichhaltigeren Informationsgehalt zu nutzen. Zweitens wird die Hypothese aufgestellt, dass der kritische Engpass der Datenknappheit im Bereich des Deep Learning mit MRT-Rohdaten durch die Entwicklung generativer Modelle zur Synthese realistischer MRT-Rohdaten überwunden werden kann.

Zur Überprüfung dieser Hypothesen präsentiert diese Arbeit einen Zyklus von miteinander verbundenen Projekten. Die erste Hypothese wird durch k-Strip validiert, einer neuartigen komplexwertigen U-Net-Architektur für die direkte Segmentierung im k-Raum. Angewendet auf das Skull-Stripping, erreicht diese Architektur eine mit etablierten Methoden in der Bilddomäne vergleichbare Leistung, bewahrt dabei wertvolle Phasendaten und ermöglicht durch die Verarbeitung nicht-menschenlesbarer Daten eine direkte Anonymisierung. Die Untersuchung wird auf die klinische Klassifikation ausgeweitet, bei der ein Modell, das k-Raum-Daten zur Schätzung der Wahrscheinlichkeit von Prostatakrebs nutzt, eine verbesserte Vorhersagegenauigkeit zeigt, insbesondere unter beschleunigten Akquisitionsszenarien, was schnellere und robustere diagnostische Wege verspricht.

Um die zweite Hypothese bezüglich der Datenknappheit anzugehen, die Innovationen im Bereich des Deep Learning mit MRT-Rohdaten verlangsamt, wird ein neuartiges komplexwertiges generatives Diffusionsmodell namens PhaseGen entwickelt. PhaseGen synthetisiert erfolgreich realistische MRT-Rohdaten, indem es spezifisch Phaseninformationen basierend auf leicht verfügbaren Magnitudenbildern generiert und so die Erstellung umfangreicher synthetischer Datensätze für das Vortraining ermöglicht. Seine Wirksamkeit wird durch die verbesserte Generalisierung von k-Strip auf realen MRT-Rohdaten validiert und es wird gezeigt, dass MRT-Rekonstruktionsmodelle eine

mit dem Training auf vollständigen realen Daten vergleichbare Leistung erzielen können, wenn sie mit PhaseGen-synthetisierten Daten angereichert werden, selbst bei stark begrenzter Verfügbarkeit realer Daten (10-15%).

Schließlich werden diese Methoden synergistisch in ein öffentlich verfügbares De-Identifizierungswerkzeug integriert. Diese Software erweitert Anonymisierungsprotokolle einzigartig auf MRT-Rohdaten und umfasst sowohl die Bereinigung von Metadaten als auch die Entfernung von Merkmalen im k-Raum mithilfe des mit PhaseGen trainierten k-Strip-Modells, was einen sichereren Datenaustausch und eine kollaborative Forschung erleichtert.

Zusammenfassend liefert diese Arbeit durch die Bestätigung der aufgestellten Hypothesen ein grundlegendes Rahmenwerk und praktische Werkzeuge, um das volle Potenzial des MRT-k-Raums mittels komplexwertigem Deep Learning auszuschöpfen und so den Weg für eine effizientere und erkenntnisreichere medizinische Bildgebung zu ebnen.

Acknowledgements

I would like to express my profound gratitude to the many individuals who have supported me throughout this PhD journey. First and foremost, I am immensely grateful to my supervisor, Prof. Dr. Dr. Jens Kleesiek, for his invaluable guidance, unwavering support, and insightful feedback. His expertise and encouragement have been instrumental in shaping my research and academic growth. I also extend my sincere thanks to Prof. Dr. Kevin Kröninger not only for his supervision of my earlier theses but also for his excellent lectures at the very beginning of my studies, sparking my interest in this field and providing a solid foundation for my research. Thank you to Prof. Dr. Dr. Jan Egger for his insightful feedback and encouragement throughout my research process.

To the whole research group at the Institute for AI in Medicine, I am grateful for the collaborative and stimulating environment that has fostered my development as a researcher. I appreciate the valuable discussions, feedback, and camaraderie that have enriched my experience and contributed to my growth.

Special thanks to Lukas, Fabian and Helmut, who not only supported me as colleagues but also became good friends. Our shared experiences, both in and out of the lab, have made this journey all the more enjoyable and memorable.

To Paul and Jonas, who have been my close study companions throughout university and have always been there to share ideas, resources, and encouragement, I am deeply thankful.

My heartfelt thanks to Irem, for her unwavering support, love, and understanding throughout this journey. Her encouragement has been a constant source of motivation for me. To my parents, Martina and Michael, thank you for your endless love and support, which have been my anchor throughout this process.

Finally I would like to thank the Cancer Research Center Cologne Essen (CCCE) and Bruno & Helene Jöster Foundation for their generous financial support, which has been instrumental in enabling my research.

Publications and other Contributions

The following publications are directly related to the work presented in this thesis. The publications are listed in the order of their publication:

Publication 1. **Rempe, M.**, Mentzel, F., Pomykala, K. L., Haubold, J., Nensa, F., Kröninger, K., Egger, J. & Kleesiek, J. (2024). k-strip: A novel segmentation algorithm in k-space for the application of skull stripping. Computer Methods and Programs in Biomedicine, 243, 107912.

Publication 2. **Rempe, M.**, Hörst, F., Seibold, C., Hadaschik, B., Schlimbach, M., Egger, J., Kröninger, K., Breuer, F., Blaimer, M. & Kleesiek, J. (2025). Tumor likelihood estimation on MRI prostate data by utilizing k-Space information. Proc. Intl. Soc. Mag. Reson. Med. 33.

Publication 3. **Rempe, M.**, Heine, L., Seibold, C., Hörst, F., & Kleesiek, J. (2025). De-identification of medical imaging data: a comprehensive tool for ensuring patient privacy. European Radiology, 1-10.

Publication 4. **Rempe, M.**, Hörst, F., Becker, H., Schlimbach, M., Rotkopf, L., Kröninger, K., & Kleesiek, J. (2025). PhaseGen: A Diffusion-Based Approach for Complex-Valued MRI Data Generation. arXiv preprint arXiv:2504.07560.

The following publications have not been included in this thesis, but have been co-authored or supervised during the doctoral research. The publications are listed in the order of their publication:

1. Hörst, F., **Rempe, M.**, Heine, L., Seibold, C., Keyl, J., Baldini, G., ... & Kleesiek, J. (2024). Cellvit: Vision transformers for precise cell segmentation and classification. Medical Image Analysis, 94, 103143.

2. Schlimbach, J. M., Kleesiek, J., Kröninger, K., **Rempe, M.**, & Weingarten, J. (2024). PS01. 16 Complex-valued Deep Neural Networks for Brain Tumor Characterization in MRI k-Space Data. Physica Medica, 125, 104018.

3. Hörst, F., Rempe, M., Becker, H., Heine, L., Keyl, J., & Kleesiek, J. (2025). CellViT++: Energy-Efficient and Adaptive Cell Segmentation and Classification Using Foundation Models. arXiv preprint arXiv:2501.05269.

4. Li, J., Zhou, Z., Yang, J., Pepe, A., Gsaxner, C., Luijten, G., ..., **Rempe, M.**, ... & Reyes, M. (2025). Medshapenet a large-scale dataset of 3d medical shapes for computer vision. Biomedical Engineering/Biomedizinische Technik, 70(1), 71-90.

5. Shahamiri, N., **Rempe, M.**, Heine, L., Kleesiek, J., & Hörst, F. (2025). Cracking the PUMA Challenge in 24 Hours with CellViT++ and nnU-Net. arXiv preprint arXiv:2503.12269.

PERSONAL CONTRIBUTION STATEMENT

Publication 1. Moritz Rempe was responsible for the conceptualization, data curation, formal analysis, methodology, software development, writing, reviewing, and visualization. Jan Egger and Jens Kleesiek supervised the publication.

Publication 2. Moritz Rempe was responsible for the conceptualization, data curation, formal analysis, methodology, software development, writing, reviewing, and visualization. Jens Kleesiek supervised the publication.

Publication 3. Moritz Rempe and Lukas Heine share first-authorship and were equally responsible for the conceptualization, data curation, formal analysis, methodology, software development, writing, reviewing, and visualization. Moritz Rempe focused on the implementation of the skull-stripping and defacing algorithms, as well as the MRI raw data anonymization. Jens Kleesiek supervised the publication.

Publication 4. Moritz Rempe was responsible for the conceptualization, data curation, formal analysis, methodology, software development, writing, reviewing, and visualization. Jens Kleesiek supervised the publication.

Contents

List of Figures

List of Tables

1

Introduction

Noninvasive imaging techniques have been a milestone in the field of medicine. By enabling the visualization of the body's internal structures without surgery, these techniques revolutionized the way diseases are diagnosed and treated. With the discovery of X-rays, or also known as Röntgen radiation, in 1895 by Wilhelm Conrad Röntgen[1], the field of medical imaging was born [1]. By taking advantage of the varying absorption of different anatomies, two-dimensional images could be produced by detecting X-rays behind a subject. In 1967, computed tomography (CT) was introduced by Godfrey Hounsfield and Allan Cormack[2], which allowed for the first time to visualize the internal structures of the human body in three dimensions [2].

[1] Nobel Prize in Physics, 1901

[2] Nobel Prize in Physiology or Medicine, 1979

Parallel to the advancements of CT, magnetic resonance imaging (MRI) was developed in the 1970s. Different to CT, MRI is based on the magnetic properties of the human body. Felix Bloch and Edward Purcell both formulated the concept of nuclear magnetic resonance (NMR) separately in 1946[3] [3]. Based on this discovery, Paul Lauterbur released the first NMR image in 1973, which was a two-dimensional image of a test tube filled with water[4] [4]. In 1977, Raymond Damadian published the first MRI image of a human body [5]. MRI has since then become a widely used imaging technique in the field of medicine. While MRI scans take significantly longer than CT scans, they provide a higher soft-tissue contrast and do not expose the patient to ionizing radiation.

[3] Nobel Prize in Physics, 1952

[4] Nobel Prize in Physiology or Medicine, 2001

With the increasing demand for MRI scans, the limited availability of MRI scanners and clinical personnel has created a significant bottleneck [6, 7, 8], with waiting times of up to several months, depending

on the country. While the radiology workforce in the UK only grew by 6.4% in 2023, the rate of conducted CT and MRI scans grew by 11% [9], showing that this situation is only going to get worse. Long waiting times can lead to delayed diagnosis and treatment, as well as increased costs for the healthcare system [10]. In order to cope with the increasing demand, the field of medical imaging is undergoing a transformation. The introduction of artificial intelligence (AI) and deep learning (DL) techniques is revolutionizing the field of medical imaging. These techniques are being used to automate and enhance many tasks in the field of radiology, such as image segmentation, classification, and reconstruction [11]. However, even with the introduction of these techniques, the demand for MRI scans still vastly exceeds the available capacity. The field of medical imaging is still in need of new techniques to improve the efficiency and effectiveness of MRI scans.

Tasks such as image segmentation or classification rely on the quality and amount of information of the underlying data. While the final reconstructed MRI is an image that humans can interpret, the raw data - commonly referred to as "k-Space" - contains more information than the final image. This information is commonly lost during the image reconstruction process. This data can potentially be used to improve the quality of downstream tasks and diagnosis. Nevertheless, despite the potential benefits, the raw data is rarely used in practice. A look at the current literature at the point of writing this thesis emphasizes the current lack of research on this topic. Only a few, but promising works can be found besides the ones presented in this thesis [12, 13, 14, 15][5].

[5] We will take a closer look at these related works in the corresponding chapters.

In this thesis, we will rethink the way we use MRI data. Do we need to rely solely on the final image, reconstructed for humans, or can machines interpret the raw data in a more efficient way? We will show that the raw data can be used for other downstream tasks than just image reconstruction. We will present a variety of methods to utilize the raw data for different tasks. Different applications and methods to utilize the raw data of MRI scans will be presented. This work shows the feasibility and potential of combining complex-valued deep learning with raw MRI data to fully leverage the information therein. The thesis is structured as follows: To begin our journey, the basic concepts of MRI and the raw data will be discussed in chapter 2. Afterwards each individual project will be presented, including a background section, the methods used, the results and a discussion.

The order of projects follows the order of their development. The first project, "k-Strip", in chapter 3 presents a framework for the task of skullstripping directly in the k-Space. The next clinical task leveraged with k-Space information is the prostate cancer classification in chapter 4. Chapter 5 presents a framework for the generation of artificial raw data, called *PhaseGen*, followed by an anonymization tool that can be used to anonymize MRI raw data in chapter 6. Finally, this thesis will be concluded with a summary of the results and an outlook on future work in chapter 7. The intuition behind the projects

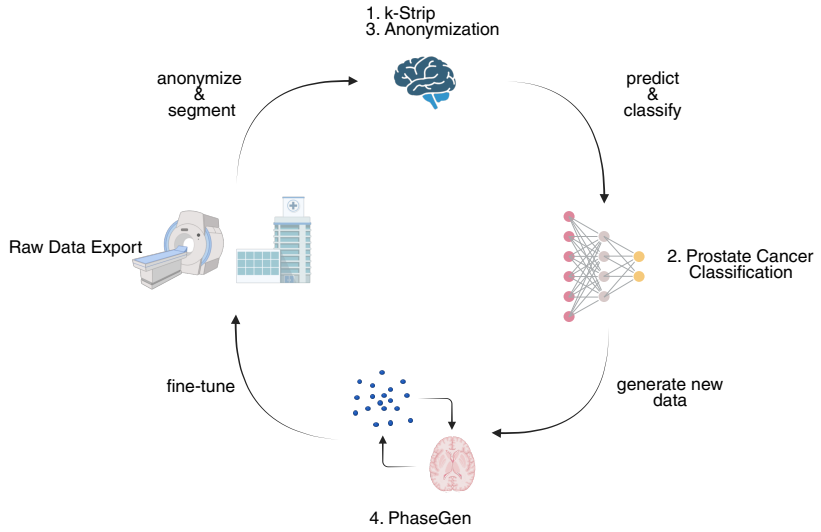

FIGURE 1.1: The project cycle of the thesis. The projects are numbered in the order of their development.

will be presented in each chapter, but the overall project cycle is shown in Fig. 1.1. We can see that the projects are not solely connected by the common theme of using raw data. For example, we realized after the two first projects that there is even less available public data to train deep learning models of MRI raw data than we expected. This led to the development of *PhaseGen*, a framework to generate artificial raw data, with which we are able to solve that issue. With this tool at hand we could then revisit the first project and improve the model, which we then were able to implement in a clinical setting, the anonymization tool.

With this let us start our journey into the world of MRI raw data. Let us explore the possibilities of using this valuable and yet underexplored data.

2

Background

In the following sections we will build the theoretical foundation to understand the concepts and techniques, as well as potential application of the presented works. We will start by introducing the physical concepts of magnetic resonance imaging (MRI), followed by an overview of machine learning algorithms relevant for this work. Specific theoretical backgrounds needed for the individual projects will be introduced before the respective chapters.

2.1 MAGNETIC RESONANCE IMAGING

MRI allows a non-invasive high quality imaging without the radiation exposure of X-ray imaging. In comparison to CT, soft tissue can be better distinguished in MRI. This imaging technique is based on the nuclear magnetic resonance (NMR) phenomenon discovered by Felix Bloch and Edward Purcell in 1946. NMR occurs when atomic nuclei are placed in a magnetic field and exposed to a second oscillating magnetic field. The nuclei absorb energy from the oscillating field and emit it again when the field is turned off. The emitted energy can be detected and used to create an image of the body.

2.1.1 *Nuclear Magnetic Resonance*

NMR is based on the magnetic properties of atomic nuclei[1]. When introduced to an external magnetic field, $\vec{B_0}$, atomic nuclei align with this field. In the case of hydrogen, the nucleus is a single proton. The magnetic dipole moment, $\vec{\mu}$, of a nucleus is proportional to its angular momentum, $\vec{L}$, (i.e., $\mu = \gamma L$), where γ is the gyromagnetic ratio. When aligned with B_0 the dynamic magnetic moment $\vec{m}$ is

[1] In MRI we usually use hydrogen nuclei, which constitute approximately 64% of all atoms in water.

given as

$$\frac{d\vec{m}}{dt} = \gamma \frac{d\vec{L}}{dt} = \gamma(\vec{m} \times \vec{B}_0). \tag{2.1}$$

The atoms will precess around the magnetic field with the Larmor frequency $\omega_0 = -\gamma B_0$, as visualized in Fig. 2.1. This movement can happen in two different states, "spin-up" or "spin-down", describing if the magnetic moment is aligned with or against the magnetic field [2]. The energy difference between these two states is given by the Zeeman effect [16] as

$$\Delta E = \gamma \hbar B_0, \tag{2.2}$$

where $\hbar$ is the reduced Planck constant. The distribution of states can be described with the Boltzmann distribution

$$\frac{N_\uparrow}{N_\downarrow} \approx 1 + \frac{\gamma \hbar B_0}{kT}, \tag{2.3}$$

with $N_\uparrow$ and $N_\downarrow$ being the number of nuclei in the two states, k the Boltzmann constant and T the temperature. In a magnetic field of 1.5 T and a temperature of 300 K, this ratio is about 10 parts per million (ppm) [17]. The net magnetization M_0 along the field $\vec{B}_0$ (z-direction) can then be calculated as

$$M_0 = \frac{1}{2}\gamma\hbar(N_\uparrow - N_\downarrow)e_z = \frac{\gamma^2\hbar^2 B_0 N_s}{4k_B T} e_z. \tag{2.4}$$

If a second magnetic field $\vec{B}_1$, perpendicular to $\vec{B}_0$ is applied with a frequency ω, the protons will absorb energy (the ΔE) and flip their spin. The frequency ω is chosen to be the Larmor frequency ω_0 of the nuclei. The excitation of the nuclei is done with a pulse sequence, which is the basis of MRI. The flip-angle θ_F of the nuclei induced by the *RF*-pulse can be controlled by the strength of B_1 and the duration of the pulse t_{RF}, described by the equation

$$\theta_F = \gamma B_1 t_{RF}. \tag{2.5}$$

This flip leads to a transverse magnetization M_{xy}, reaching its maximum when the flip angle is 90° ($M_{xy} = M_z$). To detect a signal, the RF-pulse is now turned off and the nuclei relax back into their equilibrium state. Described by the Faraday law, changing magnetic fields inside a receiver coil induces a voltage. The change of the transverse magnetization M_{xy} induces a signal in the receiver coil, following a sinusoidal shape, the Free Induction Decay (FID). The nuclei are realligning with the magnetic field $\vec{B}_0$, until the longitudinal magnetization M_z is restored. The time constant of this process

[2] which is why these two states are also called "parallel" and "anti-parallel".

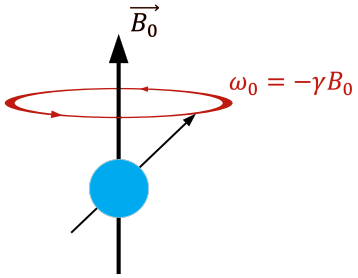

FIGURE 2.1: Precession and Larmor frequency ω_0 of a nucleus around the magentic field $\vec{B}_0$.

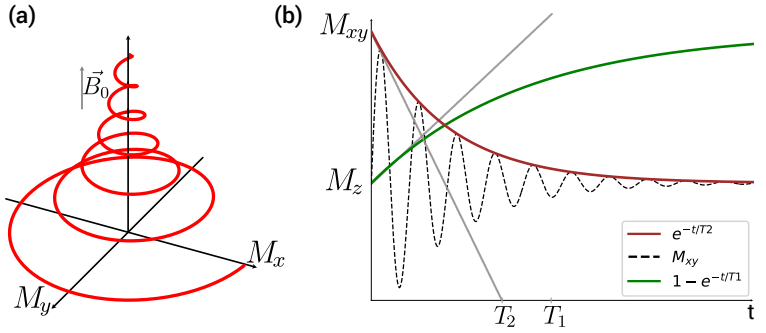

FIGURE 2.2: Visualization of the relaxation processes T_1 and T_2. (a) The longitudinal magnetization M_z relaxes back to its equilibrium state. (b) The FID signal decays over time, based on the T_1 and T_2.

is called the spin-lattice relaxation time T_1. It describes the time it takes the M_z magnetization to reach 67% of its initial value. Small phase incoherences lead to differencies of precession frequencies of the nuclei, which results in a decay of M_{xy}, described by the spin-spin relaxation time T_2. The time constant T_2 is defined as the time until M_{xy} reaches 37% of its initial value right after the RF-pulse [3]. These relaxations are visualized in Fig. 2.2. The depicted process can be approximated by the Bloch equations [19]:

$$\frac{d}{dt}\begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} 0 & \gamma B_z & -\gamma B_y \\ -\gamma B_y & 0 & \gamma B_x \\ \gamma B_y & -\gamma B_x & 0 \end{pmatrix} \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} - \begin{pmatrix} \frac{M_x}{T_2} \\ \frac{M_y}{T_2} \\ \frac{M_z - M_0}{T_1} \end{pmatrix} \tag{2.6}$$

[3] In general, T_1 is always longer than or equal T_2 [18].

2.1.2 *Image Formation*

To now utilize these magnetic effects to create an image, different spatial encodings have to be applied to localize the signal: slice selection, frequency and phase encoding, defining z, x and y location respectively. These different steps will be explained at the example of the gradient-echo-sequence, one of the simplest MRI sequences. This sequence is based on the principle of cross-sectional imaging. The volumetric data results from multiple 2D images acquired sequentially. The sequence is shown in Fig. 2.3(a).

The slice selection is done by applying a gradient in the z-direction (G_z), which leads to a linear change of the Larmor frequency in the z-axis. This gradient leads to different rotational frequencies along the plane. The RF-pulse is then applied with a specific Larmor frequency, only exciting the corresponding nuclei in the slice region[4]. The resulting flip of this excitation can be between 10° and 80°. The thickness of the slice can be influenced by the gradient strength and the duration of the RF-pulse. The stronger the gradient, the steeper

[4] In a perfect world, this excitation would look like a step function. In reality, neighbouring nuclei will also be excited.

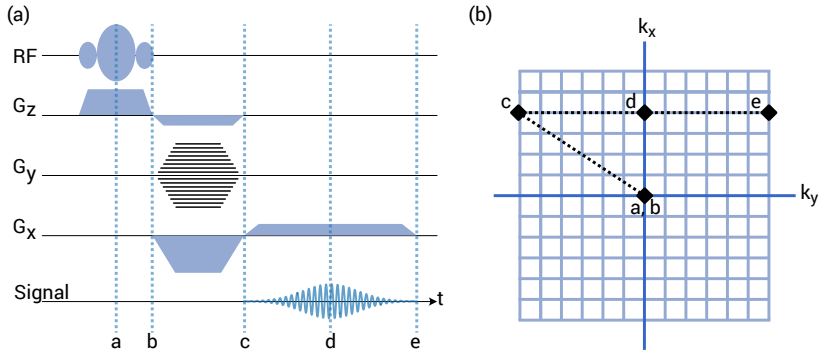

FIGURE 2.3: (a) Gradient echo sequence. The RF-pulse is applied, followed by the slice selection gradient G_z. The phase encoding gradient G_y is then applied, followed by the frequency encoding gradient G_x. The signal is then measured. (b) The k-Space data is filled by repeating the sequence.

the frequency slope, the thinner the slice. The spins now start to dephase due to the gradient-induced frequency differences. To refocus the spins, the gradient G_z is inverted, leading to a rephasing of the spins. This can also be done by applying a second RF-pulse in the opposite direction, flipping the spins by 180°[5]. During the rephasing process, the gradient G_y is applied, also called the phase encoding gradient. This gradient leads to a linear change of the precession frequency in the y-direction. The spins are now precessing with different frequencies in the y-direction, leading to a phase difference between the spins. During the time a gradient is applied, the spins will change their phase. The phase of a spin depending on the time t in a magnetic field gradient $G(t)$ is given by the equation

$$\phi(t) = \gamma B_0 t + \gamma \int_0^t G(t') \cdot x(t') dt', \tag{2.7}$$

with $x(t)$ being the position of the spin. The phase encoding gradient is then turned off and the frequency encoding gradient G_x is applied, which leads to a linear change of the precession frequency in the x-direction. The spins are now precessing with different frequencies in the x-direction, leading to a frequency difference between the spins.

Both gradients G_x and G_y are visualized in Fig. 2.4. By inverting the gradient G_x, the spins are refocusing again, inducing a signal in the receiver coil. This complex-valued signal is stored in the so-called "k-Space". This complex-valued space is explored in the following chapter.

[5] This sequence is then called a "spin-echo".

2.1.3 *Complex Numbers*

Before we can continue with the main topic of this work, the k-Space, we need to introduce the concept of complex numbers. A complex

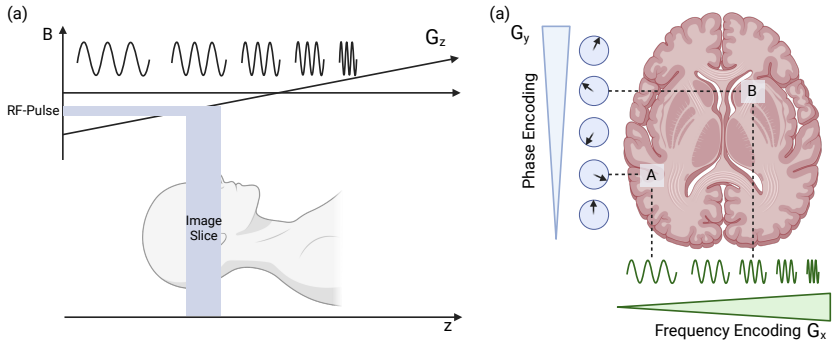

FIGURE 2.4: (a) Slice selection. G_z is applied to select the slice. The RF-pulse is then applied to excite the nuclei with the correct Larmor frequency in the slice. (b) Frequency and phase encoding. G_x is applied to create a linear change of the Larmor frequency in the x-direction. G_y is applied to create a linear change of the phase of the protons.

number z is defined as $z = a + ib$, with a and b being the real and imaginary part respectively ($\Re(z)$, $\Im(z)$). The need for complex numbers arises as soon as polynomial functions cannot be solved with real numbers, an example being the simple equation $x^2+3 = -1$. With the property $i^2 = -1$, the solution is easily found as $x = \sqrt{-4} = 2i$. The pair of real numbers (a, b) can be represented as a point in a two-dimensional complex plane[6] as shown in Fig. 2.5(a). The magnitude r of the complex number describes the distance from the origin to the point (a, b), while the phase φ, or argument, describes the angle between the $\Re$-axis and the line connecting the origin with the point (a, b), see Fig. 2.5(b). Magnitude and phase can be calculated as

$$r = |z| = \sqrt{a^2 + b^2}, \qquad \varphi = \arg(z) = \arctan2\left(\frac{b}{a}\right). \tag{2.8}$$

[6]The *Argand diagram*.

The phase φ is usually given in radians and is defined in the range $(-\pi, \pi]$, while the magnitude r can take on any positive value.

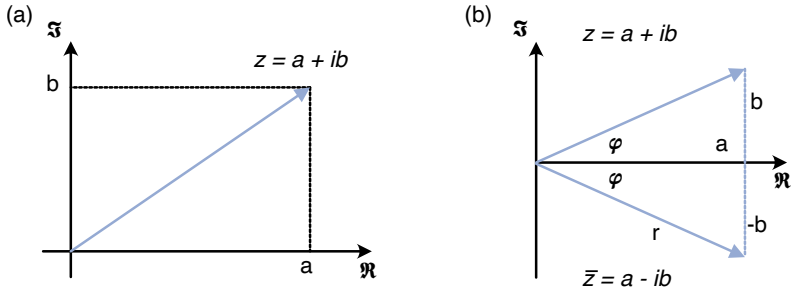

FIGURE 2.5: (a) Argand diagram of a complex number $z = a + ib$. (b) z and its complex conjugate $\bar{z} = a - ib$ of z, with magnitude r and phase φ.

For later calculations, we will now discuss some basic operations with complex numbers. The addition of two complex numbers $z_1 = a_1 + ib_1$ and $z_2 = a_2 + ib_2$ is given by

$$z_1 + z_2 = (a_1 + a_2) + i(b_1 + b_2), \tag{2.9}$$

while a subtraction is given by

$$z_1 - z_2 = (a_1 - a_2) + i(b_1 - b_2). \tag{2.10}$$

The multiplication of two complex numbers z_1 and z_2 is given by

$$z_1 \cdot z_2 = (a_1 + ib_1)(a_2 + ib_2) = (a_1a_2 - b_1b_2) + i(a_1b_2 + a_2b_1). \tag{2.11}$$

We can calculate the complex conjugate of a complex number $z = a + ib$ as $\bar{z} = a - ib$.

Especially when working with sinusoidal signals, the exponential form of complex numbers can be useful. Via the Euler formula, we can write the complex number z in its exponential form as

$$z = r(\cos(\varphi) + i\sin(\varphi)) = re^{i\varphi}. \tag{2.12}$$

We will mostly see the form of Eq. 2.12 in the following chapters.

Now that we have introduced the concept of complex numbers, we can continue the discussion of MRI signals. In the following, we will discuss the k-Space, the central object of this work.

2.1.4 *k-Space: The final frontier*

The resulting signals are used to fill the *k-Space*, or raw data domain. By applying the inverse Fourier transformation

$$s(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S(k_x,k_y)\mathrm{e}^{i2\pi(k_x x + k_y y)}dk_x dk_y, \tag{2.13}$$

the data can be translated into the image domain. This process is fully reversible by applying the Fourier transformation to bring the image domain data back into the k-Space:

$$S(k_x,k_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(x,y)\mathrm{e}^{-i2\pi(k_x x + k_y y)}dxdy. \tag{2.14}$$

The k-Space is a matrix with the phase and frequency information of the nuclei, with the center of the matrix containing the low frequency information and the edges the high frequency information. This acquisition process in the k-Space at the example of the gradient-echo explained above is visualized in Fig. 2.3(b). After the RF-pulse is applied, the slice selection gradient G_z is turned on, followed by the phase encoding gradient G_y, selecting the y position in the acquired k-Space. The frequency encoding gradient G_x is then applied, selecting the x position in the k-Space. The signal is measured and the process is repeated until the k-Space is filled.

The name "k-Space" can be derived from the wavenumber $k = 1/\lambda$, λ being the wavelength. k thus represents the number of cycles per unit distance. The center of the now filled k-Space contains the low frequency information (contrasts in the image), while the peripheral regions[7] contain high frequency information (fine details) of the image. The resolution of the image can be influenced by the strength of gradients or the gradient duration T, filling datapoints further in the periphery of the k-Space. Sampling a finer grid in the k-Space leads to a bigger field of view of the final image. The k-Space locations can be calculated by the equations

[7] Center: Low wavenumber→less cycles per unit distance.
Periphery: High wavenumber→more cycles per unit distance.

$$k_x = m\gamma G_x \Delta T_x \qquad k_y = n\gamma \Delta G_y T_y \ . \tag{2.15}$$

The acquired raw data is complex-valued, consisting of magnitude and phase or real (in-phase) and imaginary (out-of-phase) parts. The k-Space inherits a conjugate symmetry, meaning that the data is symmetric around the center of the k-Space. Thus, the signal at two opposite locations can be described as

$$S(k_x, k_y) = S^*(k_x, -k_y), \tag{2.16}$$

where S^* is the complex conjugate of S. This symmetry can be used to reduce the amount of data needed to reconstruct the image. The described conjugate symmetry of the k-Space can be used to speed up the acquisition process. By scanning either only half of the phase encoding lines or only half of the frequency encoding lines, the acquisition time can be reduced by half. Due to phase errors in the data, this approach can lead to artifacts in the image, which is why in practice at least 60% of the k-Space is filled. This is called the *partial Fourier* technique [20].

By scanning the k-Space more densely, the field of view (FOV) in the image domain can be increased ($\Delta k = 1/\text{FOV}$), while a wider k-Space increases the resolution of the image ($\Delta w = 1/k_{\text{FOV}}$, w being the pixel width). This relationship also explains one of the most common artifacts in MRI, the *aliasing* artifact. If the k-Space is not sampled dense enough, we call it undersampled, a wrap-around effect will occur in the image domain. The edges of the image will wrap around to the other side of the image.

In clinical practice, solely the magnitude image is used for most applications. The phase information can be used for additional information, like susceptibility imaging, flow quantification or for the detection of thrombosis. The k-Space data, however is until today only

used for the reconstruction of the final image domain data, mainly because of the non-interpretable nature of this data for humans.

The k-Space data is usually acquired in a Cartesian grid, but can also be acquired in a non-Cartesian grid, like radial or spiral sampling. While these alternative sampling techniques can lead to faster acquisition and are less prone to motion artifacts, they are also more complex to reconstruct. In this work, we will focus on the Cartesian sampling, as this is the most common sampling technique in clinical practice.

2.1.5 *Weighted Imaging and Other Contrast Mechanisms*

Due to varying physical and biological properties, different tissues will have different relaxation times T_1 and T_2. This leads to different contrasts in the image. Tissues with more free water (e.g. cerebrospinal fluid, CSF) will have a longer T_1 and T_2 time, leading to a brighter image, while tissues with tightly bound water (e.g. fat) will have a shorter T_1 and T_2 time. An overview of some important relaxation time values of different tissues is shown in Tab. 2.1.

Tissue	T_1 [ms]	T_2 [ms]
Fat	200-300	50-100
Muscle	800-1200	30-50
Water	3000-4000	200-300
Blood	1500-2000	200-300
CSF	4000-5000	200-300
Cartilage	2000-3000	50-100

TABLE 2.1: Overview of approximate T_1 and T_2 values of different tissues at 1.5 T [21].

The contrast in the image can be influenced by the choice of the sequence parameters, like the repetition time T_R and echo time T_E. The repetition time T_R is the time between two RF-pulses, while the echo time T_E is the time between the RF-pulse and the signal acquisition. During the T_R, the longitudinal magnetization M_z is allowed to recover, while during the T_E, the transverse magnetization M_{xy} is allowed to decay. Depending on the tissue properties, these decays will be different. If a tissue did not relax completely, the signal will be weaker, leading to a darker image. This can be used to create different contrasts in the image. The most common contrasts are T_1-weighted and T_2-weighted images.

To create a T_1-weighted image, the repetition times T_R and T_E are chosen to be short. This leads to tissues with short T_1 times, such as fat, to have a stronger signal and appear brighter in the image. Tissues with long T_1 times, such as water, will have a weaker signal

and appear darker. T_2-weighted images are created by choosing a long T_E and T_R. Only tissues with long T_2 will give a signal, as the transverse magnetization M_{xy} will decay more rapidly for short T_2 times. Water and fat appear bright in T_2-weighted images, while muscle and cartilage appear darker.

Another weighted imaging technique is the *Diffusion Weighted Imaging* (DWI). This technique is used to visualize the diffusion of water molecules in the tissue, which is dependent on the microenvironment of the tissue, possibly indicating a variety of pathologies or lesions, such as acute ischemic strokes [22, 23]. As shown in Eq. 2.7, the location of a spin can be calculated over time, based on the applied gradient. In case of no motion in the tissue, the spins will precess with a constant frequency and thus giving no contrast in the resulting image. In case of diffusion, meaning the spins are moving inside the tissue, random phase shifts will occur. These phase shifts will lead to a signal decay, which can be measured. The resulting reduction in the signal can be described by the Stejskal-Tanner equation [24]:

$$S = S_0 \mathrm{e}^{-bD}, \tag{2.17}$$

where S is the measured signal, S_0 is the signal without diffusion weighting, b is the b-value and D is the diffusion coefficient. The b-value describes the strength and duration of the applied gradient:

$$b = \gamma^2 G^2 \delta^2 \left(\Delta - \frac{\delta^3}{3} \right), \tag{2.18}$$

where G is the gradient strength, δ is the duration of the gradient and Δ is the time between the two gradients. The b-value is measured in $\mathrm{s\,mm^{-2}}$. By varying the b-value, different images can be acquired. Higher b-values lead to stronger diffusion weighting, but also introduce more noise in the image. Commonly used in clinical practice are values between 0 and 1000. The b-value b_0 is the reference value, meaning no diffusion weighting is applied. Based on different acquired b-value measurements, the apparent diffusion coefficient (ADC) can be calculated. The resulting ADC values are then used to create a map of the diffusion in the tissue, the so called ADC map.

2.1.6 *Accelerated MRI*

The acquisition of the k-Space is a time-consuming process, especially for high resolution images. To reduce acquisition time, different techniques have been proposed over time. The most common technique

is *parallel imaging*, which uses multiple receiver coils to acquire different parts of the k-Space simultaneously. The receiver coils have different sensitivities, which result in different signal strengths across the image. The acquired data is then combined to create a final image using different reconstruction algorithms, which will be explained in a later chapter. Common are 16 to 64 receiver coils.

Another technique, often combined with the parallel imaging approach, is to reduce acquisition time is to scan only a part of the k-Space, which is called *undersampling*[8]. By skipping every n-th line of the k-Space during acquisition, the acquisition time can be reduced by a factor of n. The resulting k-Space will show the described wrap-around effect. After acqusition, the k-Space is then filled with the help of an reconstruction algorithm, such as *SENSE* [25] or *GRAPPA* [26][9].

[8] Remember the not densely enough sampled k-Space we called "undersampled".

[9] We will take a closer look at *GRAPPA* in chapter 4.

One issue of accelerated MRI is a decreased signal-to-noise ratio (SNR) of the image. The SNR is defined as the ratio of the signal power P_{Signal} to the noise power P_{Noise}. The SNR can be calculated as

$$SNR = \frac{P_{\text{Signal}}}{P_{\text{Noise}}} \quad . \tag{2.19}$$

Noise is not only an issue in accelerated MRI, but also in the standard acquisition process. The noise in MRI is mainly caused by thermal noise in the receiver coils and the preamplifier. Nevertheless, the issue of low SNR becomes more pronounced in accelerated MRI, as the number of acquired data points is reduced. One way to increase the SNR is to increase the number of averages. This means, that the same slice is acquired multiple times and the resulting images are averaged[10]. The SNR increases with the square root of the number of averages n:

$$SNR = \sqrt{n} \cdot SNR_0, \tag{2.20}$$

where SNR_0 is the SNR of a single acquisition.

[10] Also called number of excitations (NEX) or number of acquisitions (NSA).

2.1.7 *MRI Datatypes*

Medical data is saved in a variety of data formats, depending on the manufacturer and the type of data. While the most common image format for radiological images is the Digital Imaging and Communications in Medicine (DICOM) format [27], the k-Space data is usually saved in unique vendor specific datatype. For Siemens MRI systems, which were used for all self-acquired raw data in this work, the raw data format is called *Twix*.

Most datatypes contain the image data, as well as metadata, which can give further information about the acquisition as well as the subject. Depending on the datatype the information is structured in different ways and needs to be loaded with different libraries. For example we will use the *nibabel* library [28] for Neuroimaging Informatics Technology Initiative (NIfTI) data [29], while the *h5py* library [30] is used for the Hierarchical Data Format (HDF5) format [31].

In this work we will work with the PyTorch library [32], which is a popular open-source machine learning library based on Python. The datatype used in PyTorch is the *torch.Tensor*[11]. We will transform all MRI datatypes into torch.Tensors, which can then be used for further processing with PyTorch.

[11] We will refer to it as *Tensor*.

This concludes the introduction to the physical principles of MRI. Necessary details for the individual projects will be introduced in the respective chapters. In the following section, the theoretical background of machine learning and complex-valued neural networks will be introduced.

2.2 ARTIFICIAL INTELLIGENCE & DEEP LEARNING

The term *artificial intelligence* (AI) was first introduced by John McCarthy in 1956 [33]. It describes a field of computer science that focuses on the development of algorithms, which can perform tasks that would normally require human intelligence. One of the most important subfields of AI is *machine learning* (ML), which focuses on the development of algorithms that can learn from given data and improve over time. After going through multiple "AI winters", a period of stagnation in the field, the last decade has seen a rapid development of AI and ML algorithms. This is mainly due to the availability of large amounts of data, as well as the possibility to process this data with better hardware, such as Graphics Processing Units (GPUs).

> "*[AI is] the science and engineering of making intelligent machines, especially intelligent computer programs*"
> —John McCarthy [34]

2.2.1 *ML & Computer Vision Basics*

Computer vision (CV) is a subfield of AI and ML, that involves tasks such as image classification, object detection and segmentation. Algorithms used in ML are called *Neural Networks*. These networks consist of artificial neurons n, which are connected over hierarchical layers. These connections, called *weights* ω (omega), allow the network to learn. Because of the increasing complexity of the networks,

the term *Deep Learning* (DL) was introduced. The ultimate goal of these networks is to learn a function F that maps the input data x to the output data y. In order to train the network, a loss function L is defined, which measures the difference between the predicted output y_{pred} and the true output y. Based on a loss function $L(y_{pred}, y)$, the weights of the network are then updated using an optimization algorithm, such as the stochastic gradient descent (SGD) [35]. The updated neuron ω_{n+1} can be calculated by

$$\omega_{n+1} = \omega_n - \beta \frac{\partial L}{\partial \omega_n}, \tag{2.21}$$

with β being the step size, or *learning rate*. This process is repeated until the loss function converges, based on the available data, at a minimum. Each repetition over all data points is called an *epoch*.

To be able to solve non-linear problems, the output of each layer is processed through an activation function $g(x)$, introducing non-linearity to the model. A common activation function is the Rectified Linear Unit: (ReLU) $g(x) = \max(0, x)$. Another common activation functions, which is mainly used in the output layer, is the *sigmoid* function $g(x) = 1/(1 + e^{-x})$, which can be used for classification problems[12] by mapping the output to a value between 0 and 1. An artificial neuron and a basic multilayer perceptron (MLP) are shown in Fig. 2.6 (a) and (b) respectively.

[12]Another common activation function for classification tasks is the *softmax* activation.

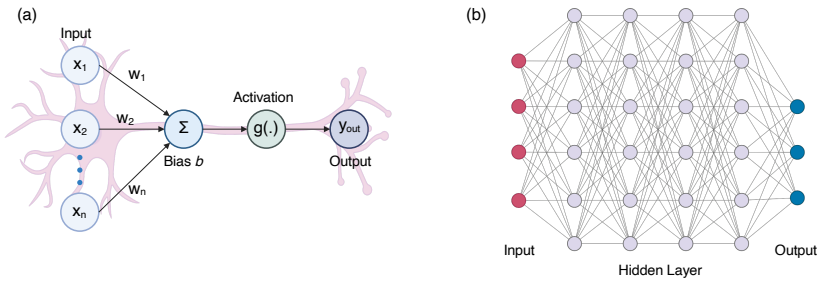

FIGURE 2.6: (a) An artificial neuron. The input data x is multiplied with the weights ω and summed up. The result is then passed through an activation function $g(x)$. (b) A basic multilayer perceptron (MLP) with an input layer, four hidden layers and an output layer.

A major problem in ML is the overfitting of the model. This means, that the model learns the training data too well and is not able to generalize to unseen data. There are multiple approaches to reduce overfitting, such as dropout [36], early stopping and data augmentation. When applying a dropout layer in a neural network, a certain percentage of neurons are randomly set to zero, one could say "turned off", during training. The model is now forced to learn a more robust representation of the data. The concept of early stopping is to

stop the training process, as soon as the loss function on a validation dataset does not improve over a certain number of epochs. Data augmentation is a technique to artificially increase the amount of training data by applying a variety of transformations to the input data. In the case of image data, this can include rotations, translations, scaling and flipping, as well as manipulations of the image contrast, color and brightness.

The above described networks, called *Feedforward Neural Networks* consist of multiple layers of neurons in combination with activation functions and regularization techniques. When working with image data, these kind of networks struggle with the high dimensionality of data. This not only has a negative impact on the performance, but also on the necessary computational resources. While the idea of convolutional neural networks (CNNs) was introduced earlier, LeCun et al. were the first to successfully apply these networks to image data [37]. The core concept of CNNs is the convolutional layer, which convolves its input, reducing the dimensionality of the data. This is done by applying a kernel K to the input data x, which is a small matrix of trainable weights. These kernels are moved over the whole image, creating localized feature maps. Common sizes for these kernels are 3×3 or 5×5. The output size $\dim_{\text{out}}$ of a convolutional layer can be calculated by the equation

$$\dim_{\text{out}} = \frac{\dim_{\text{in}} - K + 2 \cdot P}{S} + 1, \tag{2.22}$$

with K being the kernel size, P the padding size and S the stride. Padding is the number of pixels added to the input image to adjust for the kernel shape. Stride is the number of pixels the kernel is moved over the image for each convolution step. Following pooling layer reduce the spatial dimensionality of the data, by applying a function, such as the max or average function[13]. One of the most famous CNN architectures which follow a U-formed structure is the *U-Net* [38]. This architecture is widely used for image segmentation tasks, as it is able to learn both local and global features of the data. Today, extensions of this architecture are used in a variety of applications, such as image classification or object detection [39].

[13] In this thesis, we will explore another pooling function, called *Spectral Pooling*.

2.2.2 *Complex-Valued Neural Networks*

The concepts described above can be used for real-valued data, such as standard images. However, a lot of data in signal processing is complex-valued, including satellite, radar and also MRI data [40, 41,

42]. To process this data, a two channel approach is often used, in which the real and imaginary parts of the data are processed separately. This approach has some drawbacks, the most obvious one being the neglect of the natural interlink between the real and imaginary parts of complex-valued data. Real and imaginary parts are not independent of each other, but rather define a signal's magnitude and phase at a specific point of time. If this interlink is broken by separating real and imaginary parts, the model has to learn the relationship between the two channels on its own, which can be difficult and inefficient. This can lead to a loss of information and a decrease in performance. Additionally, the two channel approach increases the number of parameters in the model due to the redundancy of the real and imaginary channels, leading to higher computational cost [43]. A more natural approach is to use fully complex-valued neural networks (CVNNs), which are able to process complex-valued data directly. These networks use complex-valued weights and activation functions, allowing them to learn the interlink between the real and imaginary parts of the data. One challenge of this approach is the implementation of these complex-valued operations. As the field of CVNNs is still relatively new, there are only little resources available for the implementation of these networks.

To understand how complex-valued data can be processed by a neural network, we will first introduce the *Wirtinger calculus* [44]. A complex-valued function $f(z) = u(x,y) + iv(x,y)$ can only be differentiated with respect to the complex variable $z = x + iy$ if the Cauchy-Riemann equations are satisfied:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \tag{2.23}$$

That is only the case if the function is holomorphic[14]. In a CVNN, loss function and most of the complex-valued activation functions are not holomorphic. While there are some approaches of holomorphic activation functions, the output of our loss function is real-valued and thus not holomorphic [45]. To be able to calculate the gradient of the loss function, the Wirtinger calculus now comes into play. Wirtinger defined the *Wirtinger derivatives*[15] as

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial u}{\partial x} - i\frac{\partial v}{\partial x}\right) \quad \text{and} \quad \frac{\partial f}{\partial \bar{z}} = \frac{1}{2}\left(\frac{\partial u}{\partial x} + i\frac{\partial v}{\partial x}\right). \tag{2.24}$$

The Cauchy Riemann equations hold for these derivatives for the function $f(z) = f(z, \bar{z})$, as $\frac{\partial f}{\partial \bar{z}} = 0$ [46]. From the Wirtinger derivatives,

[14] A function $f(z)$ is holomorphic, if it is completely differentiable in a set of all neighbouring points.

[15] Also known as *Wirtinger operators* or $\mathbb{R}$-derivatives.

we can formulate some basic rules useful for the implementation of CVNNs. The product rule states that

$$\begin{aligned}\frac{\partial(f\cdot g)}{\partial z} &= \frac{\partial f}{\partial z}\cdot g + f\cdot\frac{\partial g}{\partial z}\\ \frac{\partial(f\cdot g)}{\partial \bar{z}} &= \frac{\partial f}{\partial \bar{z}}\cdot g + f\cdot\frac{\partial g}{\partial \bar{z}} \quad ,\end{aligned} \tag{2.25}$$

while the chain rule states that

$$\begin{aligned}\frac{\partial(f\circ g)}{\partial z} &= \left(\frac{\partial f}{\partial z}\circ g\right)\frac{\partial g}{\partial z} + \left(\frac{\partial f}{\partial \bar{z}}\circ g\right)\frac{\partial \bar{g}}{\partial z}\\ \frac{\partial(f\circ g)}{\partial \bar{z}} &= \left(\frac{\partial f}{\partial z}\circ g\right)\frac{\partial g}{\partial \bar{z}} + \left(\frac{\partial f}{\partial \bar{z}}\circ g\right)\frac{\partial \bar{g}}{\partial \bar{z}} \quad .\end{aligned} \tag{2.26}$$

As stated above, the Wirtinger calculus allows us to calculate the derivatives of non-holomorphic functions, as holomorphic functions are now only a special case of the Wirtinger derivatives [47]. With the help of the Wirtinger calculus, we can now define the complex-valued gradient descent [48] for a complex-valued weight $\omega_{z,n} = x_n + iy_n$ based on Eq. 2.21 as

$$\begin{aligned}\omega_{z,n+1} &= x_n - \beta\cdot\frac{\partial L}{\partial x_n} + i\left(y_n - \beta\cdot\frac{\partial L}{\partial y_n}\right)\\ &= \omega_{z,n} - \beta\cdot\left(\frac{\partial L}{\partial x_n} + i\frac{\partial L}{\partial y_n}\right)\\ &= \omega_{z,n} - 2\beta\cdot\frac{\partial L}{\partial \bar{z}} \quad .\end{aligned} \tag{2.27}$$

With this complex-valued gradient descent, complex-valued neurons can be trained. The benefits of using these special kinds of neurons are still an ongoing research topic. When comparing real-valued neurons with complex-valued neurons, the unique properties inherited by complex numbers become obvious. To mimic a complex-valued neuron, a real-valued neuron with two channels can be used. This approach is frequently used, as it is easier to implement with current frameworks. A real-valued neuron[16] with two channels can be described as

[16] For simplicity we look at a neuron without activation function or bias.

$$\begin{pmatrix} x_{1,out}\\ x_{2,out}\end{pmatrix} = \begin{pmatrix} x_{1,in}\\ x_{2,in}\end{pmatrix}\cdot\begin{pmatrix} W_{1,1} & W_{1,2}\\ W_{2,1} & W_{2,2}\end{pmatrix}, \tag{2.28}$$

where $x_{1,in}$ and $x_{2,in}$ are the input values and W is the weight matrix. In comparison, a complex-valued neuron can be described in polar form as

$$\begin{pmatrix} a_{out}\\ b_{out}\end{pmatrix} = \begin{pmatrix} a_{in}\\ b_{in}\end{pmatrix}\cdot\begin{pmatrix} r\cos\varphi & -r\sin\varphi\\ r\sin\varphi & r\cos\varphi,\end{pmatrix}, \tag{2.29}$$

with a_{in} and b_{in} being the real and imaginary parts of the input z_{in} and r and φ being the magnitude and phase of the complex-valued

neuron. While the real-valued weight matrix W consists of four individual values, the complex-valued weight matrix consists of only two values, the magnitude r and the phase φ. This reduction in degrees of freedom not only reduces the number of parameters in the model, but also allows the model to learn more robust representations. The magnitude r represents an amplitude scaling, while the phase φ represents a rotation. This means that the complex-valued neuron can learn to scale and rotate the input data, while the real-valued neuron can only learn to scale the input data. This is a major advantage of complex-valued neurons, especially when handling sinusoidal data, such as MRI data [49].

Details of the implementation of complex-valued neural networks will be discussed in the respective chapters, including the mathematical formulations.

3

k-Strip: Rethinking k-Space

There's no such thing as the unknown -
only things temporarily hidden,
temporarily not understood.
—Capt. Kirk

Question 1. Is it possible to perform downstream tasks, such as segmentations, directly in the k-Space?

This chapter will introduce the first work of this thesis, which is a novel approach showing the feasibility of utilizing k-Space data for downstream tasks, in this case, image segmentation.

This chapter presents the first approach to perform a segmentation task directly in the k-Space domain. Previous research has primarily focused on the use of k-Space data for image reconstruction, with limited exploration of its application to downstream tasks such as segmentation. More recently, additional studies have demonstrated the feasibility of using k-Space data for cardiac MRI segmentation tasks [14, 50].

Major parts of the presented results in this chapter have been published in the paper "k-strip: A novel segmentation algorithm in k-Space for the application of skull stripping" [51]. A detailed transparency statement can be found in Appendix A.1. The proposed architecture has been improved throughout the course of this thesis, which is why some results may differ from the published paper. The source code of the k-Strip model is publicly available on GitHub[1].

[1] https://github.com/TIO-IKIM/k-Strip

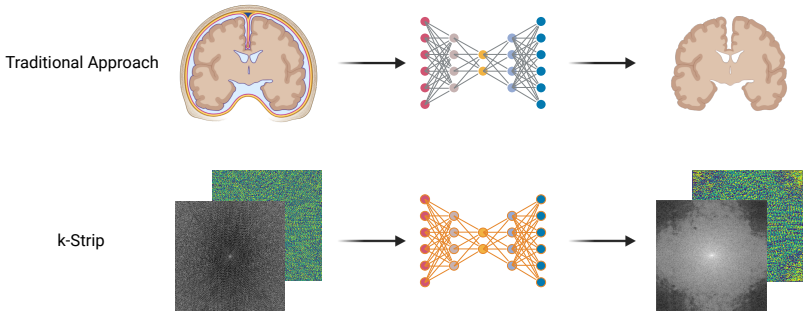

FIGURE 3.1: Schematic comparison between traditional skull stripping algorithms and k-Strip. In comparison to traditional deep learning based approaches, k-Strip operates directly in the complex-valued k-Space domain. The model takes in magnitude and phase information and outputs a k-Space without skull and non-brain tissue. The resulting k-Space can be transformed back into the image domain, where the skull stripped image can be used for further downstream tasks which can utilize the additional phase information preserved by k-Strip.

3.1 INTRODUCTION

The task of skull stripping[2] is chosen as a proof of concept, as it is a well-studied problem in the field of medical image analysis and a common postprocessing step in many medical imaging pipelines [52]. Performing skull stripping directly in the k-Space also provides a direct anonymization of the data, without the need of first transforming it into the image domain. This mitigates any risk of re-identification of patients, as the raw data is not stored in a human-readable format. Last but not least, this approach preserves valuable phase information, usable for further downstream tasks. The resulting model, namely “k-Strip", lays the foundation for the subsequent works of this thesis, as it not only demonstrates the possibility of working directly in the raw data domain of MRI, but also provides a variety of building blocks for future architectures used in the chapters 4 and 5. A schematic overview of the idea behind k-Strip is shown in Fig. 3.1.

[2]Also commonly referred to as “brain segmentation" or “brain extraction".

First, the task of skull stripping is introduced, followed by the challenges and possible benefits of performing segmentation tasks in the k-Space domain. The proposed architecture, including the implemented complex-valued building blocks are presented. The results of the k-Strip model are evaluated in Section 3.3.3. This chapter concludes with a discussion of the results and a summary of the findings which motivated the upcoming projects.

3.2 METHODS

3.2.1 *Skull Stripping and how to do it in k-Space*

Skull stripping, or brain segmentation, is a common postprocessing step in many medical imaging pipelines. The goal of this segmentation task is to separate the brain from the surrounding skull and other non-brain tissues in an MRI image. This step has been shown to be beneficial for many downstream tasks, such as image registration [53] or tumor detection [54]. Additionally, skull stripping is an important processing step when anonymizing medical images, as it removes identifiable features such as facial structures [55].

The task of skull stripping is usually performed in the image domain, where a variety of methods have been proposed, ranging from thresholding techniques [56] and atlas-based methods [57], such as the optiBET algorithm [58], to deep learning-based approaches such as “SynthStrip" by Hoopes et al. [59], HD-BET [60] by Isensee et al., or the 3D-approach by Kleesiek et al. [61].

The major challenge of performing segmentation tasks such as skull stripping in the k-Space is its non-locality. One pixel in the image domain is represented by a superposition of all frequencies in the k-Space. This means that a local operation such as a convolution in the k-Space does not correspond to a local operation in the image domain. Additionally, binary segmentation masks can not be represented in the k-Space, as they are not continuous functions. Zhang et al.[50] utilize a real-valued transformer-based model to perform a cardiac MRI segmentation directly on the magnitude of the k-Space data and achieve promising results at higher undersampling rates. However, we want to keep the complex-valued nature of the data intact, which means that a direct application of traditional deep learning architectures, such as U-Net, is not possible. The task to be solved changes from a binary segmentation task to a complex-valued regression task, we can call it an “image-to-image" task. While the fundamental architecture of k-Strip was already discussed in my Master’s thesis, the proposed architecture has been improved and extended greatly in this work. In the following, the proposed architecture is presented, including the complex-valued building blocks and their intuition.

3.2.2 *The k-Strip Architecture*

The underlying architecture k-Strip is based on the U-Net [38], which has shown to be a powerful architecture for many image-to-image

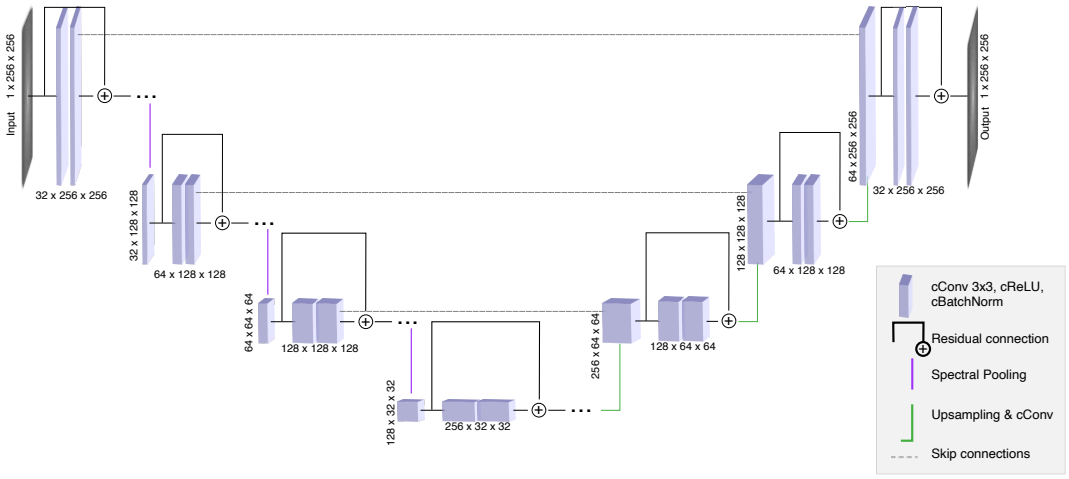

FIGURE 3.2: Overview of the k-Strip architecture. The model consists of an encoder and decoder path. The model is based on the U-Net architecture, but with modifications to account for the complex-valued data and the non-locality of the k-Space. The input and output are complex-valued k-Space data. Adapted from [51].

tasks.

The U-Net consists of an encoder and decoder path, where the encoder path is used to extract features from the input data and the decoder path is used to reconstruct the output data. The k-Strip architecture is similar to the U-Net, but with modifications to account for the complex-valued data and the non-locality of the k-Space. Additionally, we implemented residual connections to improve the flow of information through the network [62]. An overview of the architecture is shown in Fig. 3.2.

$$
\begin{aligned}
&\text{Complex convolution matrix:}\\
&W = X + iY\\
&\text{Complex data:}\\
&d = a + ib\\
&\text{Complex convolution:}\\
&W * d = (X * a - Y * b) + i(X * b - Y * a)
\end{aligned}
\tag{3.1}
$$

[3]We will always use a kernel size of 3×3.

The complex convolution matrix W[3] is defined as a complex-valued matrix, where X and Y are real and imaginary parts respectively. The custom implementation was later replaced by the native PyTorch implementation in favor of computation time. We additionally implemented a custom complex-valued batch normalization layer, The implementation is based on the work of Trabelsi et al. [63] and adapted from the implementation presented in [64]. The complex-

valued batch normalization is defined as shown in Eq. 3.2.

$$
\begin{aligned}
&\text{Complex batch normalization of } x\text{:}\\
&\tilde{x} = (V)^{-\frac{1}{2}}(x - \mathrm{E}[x])\\
&\text{Covariance matrix } V\text{:}\\
&V = \begin{pmatrix} V_{rr} & V_{ri} \\ V_{ir} & V_{ii} \end{pmatrix}\\
&= \begin{pmatrix} \mathrm{Cov}(Re\{x\}, Re\{x\}) & \mathrm{Cov}(Re\{x\}, Im\{x\}) \\ \mathrm{Cov}(Im\{x\}, Re\{x\}) & \mathrm{Cov}(Im\{x\}, Im\{x\}) \end{pmatrix}
\end{aligned}
\tag{3.2}
$$

Batch normalization is a common technique to stabilize the training process of deep learning models. It normalizes the input data to have a mean of 0 and a standard deviation of 1. The covariance matrix V is calculated based on the real and imaginary parts of the input data. The batch normalization layer is applied to each channel of the input data separately. In all experiments a momentum of 0.1 is used. The momentum defines how much the previous batch is taken into account when calculating the running mean and variance, with lower values leading to slower updates.

To introduce non-linearity to our model, we have to implement a non-linear activation function. The most common activation function in deep learning is the ReLU activation function and its variants, such as leaky ReLU. Because ReLU-based activation functions are not defined for complex numbers, due to the lack of natural ordering of complex numbers. Thus the question arises which activation function to use for complex-valued data. While there is a variety of specifically crafted complex-valued activation functions proposed in the literature, such as the “Complex Swish" [65] or “Cardiod" activation function [66], we decide to use a split approach for the activation function, as first presented in [67]. By applying the ReLU function to both real and imaginary part of the input and combining both outputs back to a complex number, we can achieve a similar effect as the ReLU activation function. We will call this activation function “cReLU". A visualization of this activation function is shown in Fig. 3.3. The cReLU activation function is defined as

$$
\mathrm{cReLU}(x) = \mathrm{ReLU}(Re\{x\}) + i \cdot \mathrm{ReLU}(Im\{x\}) \quad . \tag{3.3}
$$

The cReLU activation function is applied to the real and imaginary parts of the input data separately. This allows us to use the ReLU activation function, which is a common choice in deep learning ar-

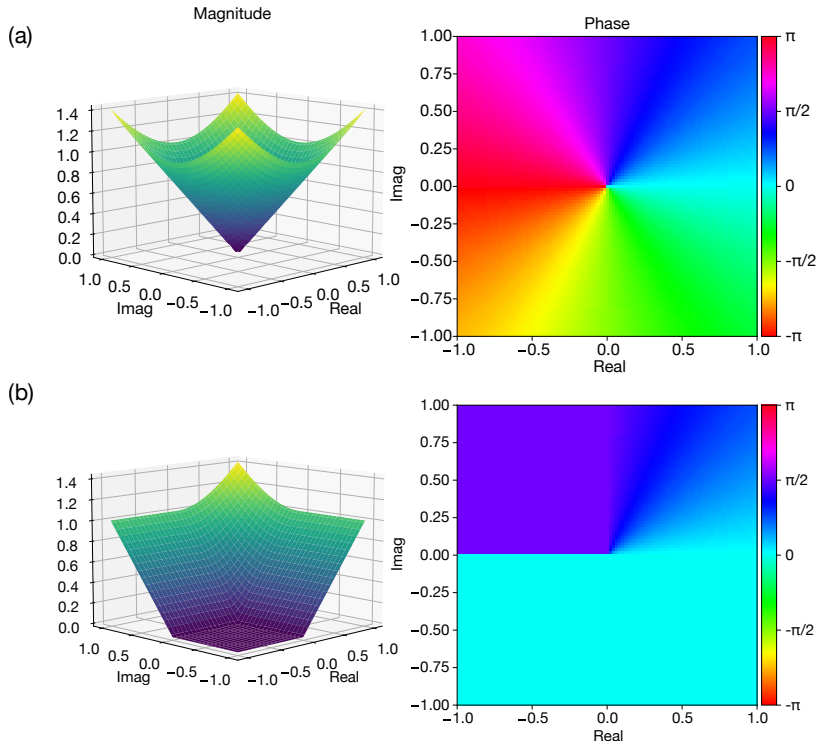

FIGURE 3.3: Visualization of the cReLU activation function. Left: Magnitude; Right: Phase. (a) Complex-valued input z. (b) Output of the cReLU activation function. The real and imaginary parts of the input data are passed through the ReLU activation function separately and then combined back to a complex number. Adapted from [68].

chitectures, while still preserving the complex-valued nature of the data.

Together the complex-valued convolution layer, the cReLU activation function and the complex-valued batch normalization forms one *cConv* block.

The next important building block is the pooling layer. Pooling is a technique to reduce the input size in consecutive downsampling layers. This is commonly done by applying a max pooling operation, which takes the maximum value of a local region in the input data, or average pooling, which takes the average value of a local region [69]. Both operations can be formulated as

$$\text{MaxPool}(X)_{i,j,k} = \max_{m,n} X_{i\cdot s_x+m, j\cdot s_y+n, k} \tag{3.4}$$

$$\text{AvgPool}(X)_{i,j,k} = \frac{1}{f_x \cdot f_y} \sum_{m,n} X_{i\cdot s_x+m, j\cdot s_y+n, k} \quad , \tag{3.5}$$

with X being the input data, i, j the indices of the output data, k the channel index, s_x, s_y the stride in x and y direction, f_x, f_y the filter size in x and y direction, and m, n the indices of the local region. The pooling operation is applied to each channel of the input data separately.

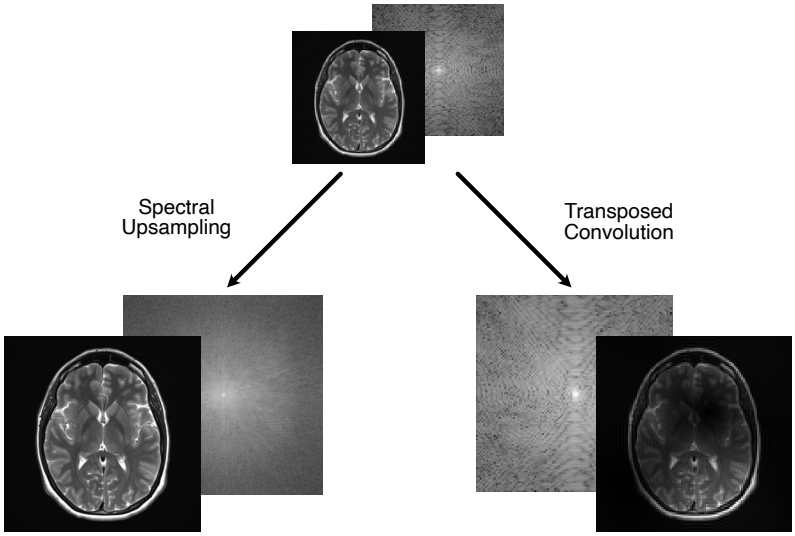

FIGURE 3.4: Comparison of the transposed convolution layer and the spectral upsampling layer applied to pooled k-Space data. The transposed convolution layer applies an elementwise multiplication on the frequency data, which translates to a convolution in the image domain. The spectral upsampling layer utilizes the residuals of the pooling operation to preserve the high-frequency information.

While average pooling could be used for complex-valued data, max pooling is not defined for complex numbers[4]. Applied to image data, average pooling leads to a more unsharp image, as it takes the average over multiple pixels to compress the size of the image. Problems occur when this technique is used on k-Space data. If we now average multiple pixels in the k-Space, we are averaging multiple frequencies, which leads to artifacts in the image domain. These will be wrapping artifacts, as we artificially decreased the distance Δk between sampling points in the k-Space. Thus, *Fourier pooling*[5] is applied to the k-Space data, cutting off rings of frequencies in the periphery of the k-Space [71]. The existing Fourier pooling layer is extended by saving the pooled frequencies and can be formulated as

[4] One would need to apply max pooling on the absolute values.

[5] Or *Spectral pooling* [70].

$$\text{FourierPool}(X)_{i,j,k} = X_{i+\frac{H-\kappa H}{2},j+\frac{W-\kappa W}{2},k} \tag{3.6}$$

$$r_k = X_k - \text{FourierPool}(X)_k \quad , \tag{3.7}$$

with H and W being the height and width of the input data, κ the pooling factor and r the residual of the pooling operation. This allows us to introduce a novel upsampling layer for k-Space data, which replaces the common transposed convolution layer [72]. The transposed convolution layer, also known as deconvolution or fractionally strided convolution, is used to upsample feature maps in the decoder path of the U-Net. By sliding a kernel over the input data and applying element-wise multiplications[6] in combination with summation, the

[6] The *Hadamard* product.

transposed convolution upsamples the data. Again, this approach becomes problematic when applied to k-Space data, as the convolution theorem holds[7]:

$$u * v = \mathcal{F}^{-1}\{\mathcal{F}\{u\} \cdot \mathcal{F}\{v\}\} \tag{3.8}$$

$$u \cdot v = \mathcal{F}^{-1}\{\mathcal{F}\{u\} * \mathcal{F}\{v\}\} \quad , \tag{3.9}$$

with f and g being two functions, $\mathcal{F}^{-1}$ and $\mathcal{F}$ the (inverse) Fourier transform, "$\cdot$" the Hadamard product and "$*$" the convolution operator. This means that a pointwise multiplication in the frequency domain corresponds to a convolution in the image domain. The resulting effect on the input data is shown in Fig. 3.4 in comparison to the novel spectral upsampling layer. The spectral upsampling layer utilizes the saved residuals of the spectral pooling by attaching them back onto the input, preserving the high-frequency information. The spectral upsampling layer can be formulated as

$$\text{SpectralUpsample}(\tilde{X})_k = \tilde{X} + r_k \quad . \tag{3.10}$$

$\tilde{X}$ is the pooled input data and r_k the residual of the pooling operation. To introduce learnable parameters, a complex-valued convolution is applied to the upsampled output.

To improve the flow of information in our model, we implement residual connections in each cConv block. The residual connections transfer the initial input information of a block to the end of the block and combine it with its output. The residual connection can be formulated as

$$\text{Residual}(X) = X + \text{cConv}(X) \quad . \tag{3.11}$$

The resulting output is then passed through another activation function. We insert multiple cConv blocks on each downsampling layer to increase the number of learnable parameters and the expressiveness of the model. The number of cConv blocks is defined as n_{blocks} and is set to 2 in all experiments.

To stabilize training and increase the models ability to generalize, we implement a complex-valued version of the dropout layer. The dropout layer is a common technique to prevent overfitting in deep learning models. It randomly sets a fraction of the input units to 0 during training, which forces the model to learn more robust features. Our dropout implementation randomly sets a fraction of the inputs' real parts to zero. Neurons that have a zeroed real part are then also assigned a zero imaginary part.

[7] Vice versa it holds as

$$\mathcal{F}\{u * v\} = \mathcal{F}\{u\} \cdot \mathcal{F}\{v\}$$
$$\mathcal{F}\{u \cdot v\} = \mathcal{F}\{u\} * \mathcal{F}\{v\}$$

3.3 EXPERIMENTS & RESULTS

To show the feasibility of performing segmentation tasks in the k-Space, one has to compare the results of k-Strip to the results of traditional skull stripping algorithms and if possible on public datasets. Due to the lack of publicly available k-Space datasets which include segmentation masks, we used two publicly available image domain datasets and one in-house susceptibility weighted imaging (SWI) dataset from the University Hospital Essen, which includes phase data. Additionally, a image domain dataset from the University Hospital Essen which includes pathologies such as glioblastoma multiforme (GBM) in some of the brain scans.

NFBS The Neurofeedback skullstripped (NFBS) repository [73] consists of 24 000 fully sampled brain MRI scans from a total of 125 patients. All scans were acquired with a T_1 weighted sequence and a resolution of 1 mm^3. The ground truth, in this case the skull stripped brain images, are manually refined based on automatic segmentations performed with the BEaST algorithm [74].

CC-359 The Calgary-Campinas-359 (CC-359) dataset [75] consists of 359 patients. The sequences and resolutions are again T_1 weighted and 1 mm^3. In this dataset, the ground truth is generated via the “Simultaneous Truth and Performance Level Estimation" (STAPLE) algorithm, which combines the results of multiple skull stripping algorithms to create a consensus mask [76]. For testing purposes, the dataset also comes with 12 additional subjects, which were segmented manually. These volumes will be used as part of the test dataset. The CC-359 dataset is a multi-vendor dataset, which includes scans from different vendors, namely Siemens (Germany), Philips (Netherlands), GE (USA).

SWI The SWI dataset is an in-house dataset of the University Hospital Essen and consists of 83 000 slices from a total of 433 patients. SWI data commonly includes phase information, which we combined with the T_1-weighted magnitude images with a resolution of 2 mm^3. The phase data is scaled into the range of $(0; 2\pi]$ and then combined with the magnitudes via Eq. 2.12. The ground-truth is generated by applying the HD-BET algorithm on the magnitude data in the image domain.

GBM The GBM dataset contains 21 822 slices from a total of 150 patients. The scans were acquired with 1.5 T and 3 T T_1 sequences. The majority of scans in this dataset include pathologies. This enables

us to evaluate the model on a dataset with pathologies, which is a common use case in medical imaging. The ground truth is generated by applying the HD-BET algorithm on the magnitude data in the image domain.

An overview of all datasets used in this work is shown in Appendix Table B.1.

Because all the used datasets are in the image domain, the data is transformed into the k-Space via the Fourier transform in Eq. 2.14. Prior to transformation in k-Space, each slice is resized to a size of 256×256 pixels. To train our model in the k-Space, the ground truth needs to be transformed accordingly. First, we multiply the binary segmentation masks in the image domain with the initial images. The resulting images of the brain tissue are then transformed into the k-Space, equal to the input data.

Each dataset is split into training, validation and test set. The training set consists of 80 % of the data, the validation set of 10 % and the test set of 10 %. All splits are performed patient-wise to prevent data leakage.

3.3.1 *Periphery Augmentation*

For data augmentation, we apply a novel augmentation technique, which we will call *periphery augmentation*. The k-Space data is randomly multiplied with a factor in the range of $[0.7; 1.3]$ in the periphery of the k-Space. The periphery range is randomly chosen in the range of $[5; 40]$ pixels distance from the center. These ranges have been determined heuristically based on the GBM dataset. Let

$$\alpha \sim \text{Uniform}(0.7, 1.3)$$

be the augmentation factor and

$$r_{\text{min}}, r_{\text{max}} \sim \text{Uniform}(5, 40)$$

the minimum and maximum radius of the periphery as a distance from the center, then the augmentation k' of the k-Space data k can be formulated as

$$k'(x,y) = \begin{cases} \alpha \cdot k(x,y), & \text{if } r_{\text{min}} \leq d(x,y) \leq r_{\text{max}} \\ k(x,y), & \text{otherwise} \end{cases} \tag{3.12}$$

with $d(x,y)$ being the distance from the center of the k-Space. This augmentation enables us to train the model on varying scales of k-Space data while introducing additional noise in the periphery of the

k-Space. Additionally, we do not need to transform k-Space data back into the image domain for augmentation.

3.3.2 *Model Hyperparameters*

The k-Strip model underwent an extensive grid-search to find the optimal hyperparameters based on the GBM dataset. During a grid-search, the model is trained with different hyperparameters and the best performing model is selected based on the validation loss. The initial learning rate of the model is set to 1×10^{-3} , which is reduced by 50% every 50th epoch. The Adam optimizer [77] has a beta coefficient of 0.99 and epsilon of 1×10^{-8} We apply a complex-valued dropout after every convolution layer of 5% and a batch normalization layer with a momentum of 0.1 The model is trained for 150 epochs with a batch size of 64 and a least absolute error (LAE) loss function. We trained the model for three days on an *NVIDIA A100* GPU with 80GB of graphics memory.

3.3.3 *Experiments*

To evaluate our model, we employed the two most common metrics for image segmentation tasks, the Dice coefficient and the Hausdorff distance [78, 79]. The Dice coefficient (DSC) is a measure of similarity between two sets and is defined as

$$\mathrm{DSC}(A,B) = \frac{2|A \cap B|}{|A| + |B|} \quad , \tag{3.13}$$

with A and B being the two sets. In the context of segmentation, A is the ground truth and B the predicted segmentation mask. The Dice coefficient ranges from 0 to 1, with 1 being a perfect match between the two sets. The Hausdorff distance (HD) is a measure of the distance between two sets and is defined as

$$\mathrm{HD}(A,B) = \max\left(\max_{a \in A} \min_{b \in B} d(a,b), \max_{b \in B} \min_{a \in A} d(b,a) \right) \quad , \tag{3.14}$$

with $d(a,b)$ being the distance between the two points a and b. The Hausdorff distance ranges from 0 to ∞, with 0 being a perfect match between the two sets. In the case of image segmentation the distance is given in pixels. Additionally we report the accuracy, sensitivity and specificity of the model in regards to the predicted segmentations.

To calculate the above mentioned metrics, the output of k-Strip is being transformed into the image domain via the inverse Fourier transformation. Afterwards, a threshold is applied on the image to

TABLE 3.1: Results of the k-Strip model on the different test sets. The results are reported as mean and standard deviation over all test subjects. The model is compared with the SOTA methods BSE [80], OptiBET [58] and HD-BET [60]. Additionally, the k-Strip model was trained with 70 000 and 25 000 training samples respectively to see the impact of the training set size on the results.

Model	DSC (%) ↑	DHD (pixels) ↓	Acc (%) ↑	Sens (%) ↑	Spec (%) ↑
SWI					
BSE [80]	93.3 ± 2.5	4.04 ± 0.37	96.6 ± 0.21	95.5 ± 0.23	97.0 ± 0.19
OptiBET [58]	92.5 ± 2.2	4.26 ± 0.56	95.9 ± 0.22	96.0 ± 0.22	96.6 ± 0.21
k-Strip (83 000)	97.4 ± 0.5	3.42 ± 0.45	98.9 ± 0.13	97.5 ± 0.18	99.5 ± 0.04
k-Strip (25 000)	96.9 ± 1.6	3.35 ± 0.53	98.9 ± 0.13	97.2 ± 0.19	99.5 ± 0.04
GBM					
BSE [80]	94.6 ± 2.1	5.03 ± 0.57	97.0 ± 0.19	98.5 ± 0.19	94.2 ± 0.26
OptiBET [58]	94.8 ± 2.1	4.53 ± 0.66	97.4 ± 0.18	98.7 ± 0.13	96.0 ± 0.22
k-Strip	94.1 ± 2.5	4.06 ± 0.41	97.4 ± 0.18	94.5 ± 0.26	98.8 ± 0.12
NFBS [73]					
BSE [80]	95.8 ± 2.8	3.55 ± 0.57	98.5 ± 0.14	98.3 ± 0.15	98.5 ± 0.14
HD-BET [60]	96.3 ± 2.0	3.35 ± 0.52	98.7 ± 0.13	98.4 ± 0.14	99.7 ± 0.03
OptiBET [58]	95.2 ± 1.3	3.89 ± 0.41	98.4 ± 0.14	97.0 ± 0.20	98.7 ± 0.13
k-Strip	95.7 ± 1.2	3.94 ± 0.64	98.5 ± 0.14	97.1 ± 0.19	99.1 ± 0.09
CC-359 [75]					
BSE [80]	94.2 ± 3.5	3.81 ± 0.43	97.6 ± 0.17	98.7 ± 0.16	97.1 ± 0.19
HD-BET [60]	96.4 ± 1.8	3.25 ± 0.36	97.6 ± 0.17	98.7 ± 0.16	97.1 ± 0.19
OptiBET [58]	95.3 ± 2.1	3.98 ± 0.49	98.1 ± 0.15	95.6 ± 0.23	98.6 ± 0.13
k-Strip	94.4 ± 2.3	4.15 ± 0.43	98.2 ± 0.14	96.1 ± 0.21	98.8 ± 0.12

create a binary segmentation mask. The threshold is set to 1.7 times the mean value of the predicted image. This threshold was chosen based on quantitative evaluations. The resulting binary mask is then used to compute the metrics.

The results of the k-Strip model are compared with three state-of-the-art (SOTA) methods: BrainSuite algorithm (BSE) [80] and OptiBET, as well as HD-BET.

We only consider slices with at least 5000 pixels of brain tissue as relevant for evaluation. With an initial input image size of 256×256 pixels, this corresponds to a minimum of roughly 7.6% of brain tissue in the image. We justify this threshold by the fact that all slices with less brain tissue will not have any relevant phase information and are thus not relevant for our use case. Slices affected by this threshold are the lower parts of the brain stem and the top most brain slices.

The results of the k-Strip model on each test set in comparison to the SOTA methods are shown in Tab.3.1. The results are reported as mean and standard deviation over all test subjects. It can be seen that the k-Strip model achieves comparable results to the SOTA methods. In case of the SWI dataset, which includes actual phase data, k-Strip surpasses both BSE and OptiBET with a DSC of 97.4 % and a DHD of 3.42 pixels in comparison to 93.3 % and 4.04 pixels for BSE and

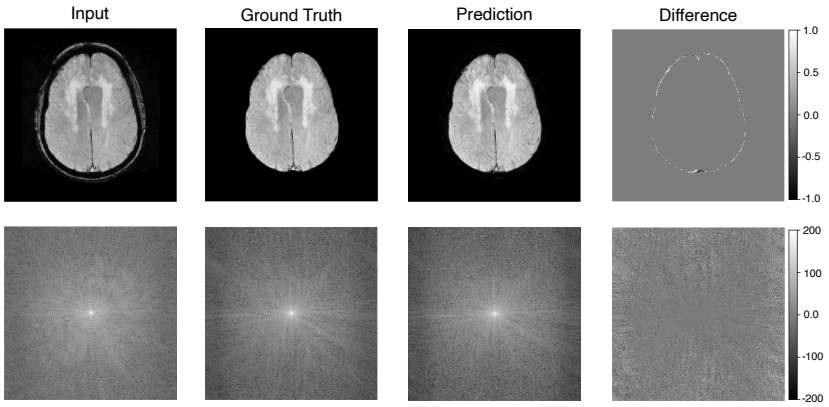

FIGURE 3.5: Exemplary output of k-Strip on the SWI test dataset. The first column shows the input data, the second column the output of k-Strip and the third column the ground truth. The last column shows the difference between the output and the ground truth. All images show the magnitude data in image domain and k-Space respectively. The k-Space data is shown logarithmically. Adapted from [51].

92.5 % and 4.26 pixels for OptiBET. To further investigate the results on the SWI dataset, we conducted the same experiment with a reduced training set size of 25 000 samples. The results show that the model is still able to achieve a DSC of 96.9 % and a DHD of 3.35 pixels, which surpasses both compared SOTA methods. These results indicate, that the k-Strip model utilize the additional information of the phase information in the k-Space data and thus achieves better results than the SOTA methods. In comparison, when only transforming the magnitude data into the k-Space, the model achieves similar results as the compared methods.

Exemplary results of the k-Strip model on the SWI dataset are shown in Fig. 3.5. In the difference image, we can see that the model is able to accurately remove the skull and non-brain tissue from the input data. The more interesting difference image is in the bottom row, which shows the difference in the k-Space domain between ground truth and prediction. While the model us able to accurately predict the center of the k-Space, it struggles with the periphery of the k-Space. This is most likely due to the fact, that the values in the periphery of the k-Space are smaller by magnitudes than the values in the center. The model thus is penalized less when predicting the periphery wrong.

It showed that in regions above the eyes, the model is able to predict the skull and non-brain tissue better than in the lower regions of the brain stem with DSC values of $98.8 \pm 0.8\%$ and $95.1 \pm 1.8\%$ respectively. Exemplary outputs of regions below the eyes can be seen in Fig. 3.6. This is most likely due to the nature of the frequency domain, as the non-locality favors the prediction of symmetric structures

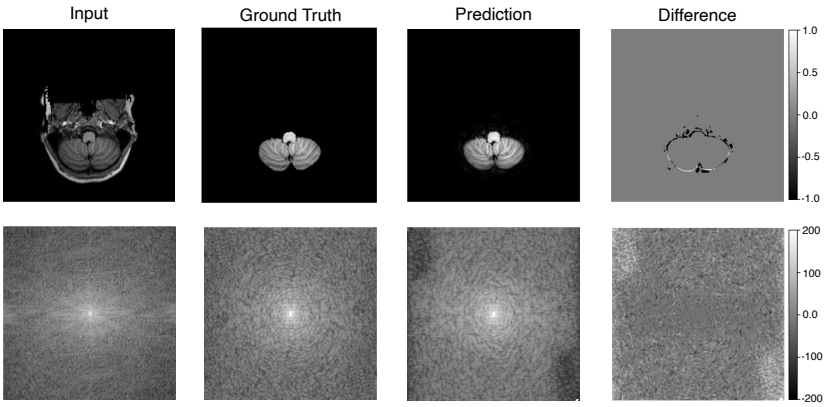

FIGURE 3.6: Exemplary output of k-Strip on the NFBS test dataset at the brain region below the eyes. The first column shows the input data, the second column the output of k-Strip and the third column the ground truth. The last column shows the difference between the output and the ground truth. All images show the magnitude data in image domain and k-Space respectively. The k-Space data is shown logarithmically. Adapted from [51].

in all quadrants of the image.

The CC-359 dataset includes scans from different vendors. 3D renderings of all three tested vendors can be seen in the Fig. 3.7. k-Strip achieves consistent results on all three vendors, with an average DSC of 94.4 % and a DHD of 4.15 pixels.

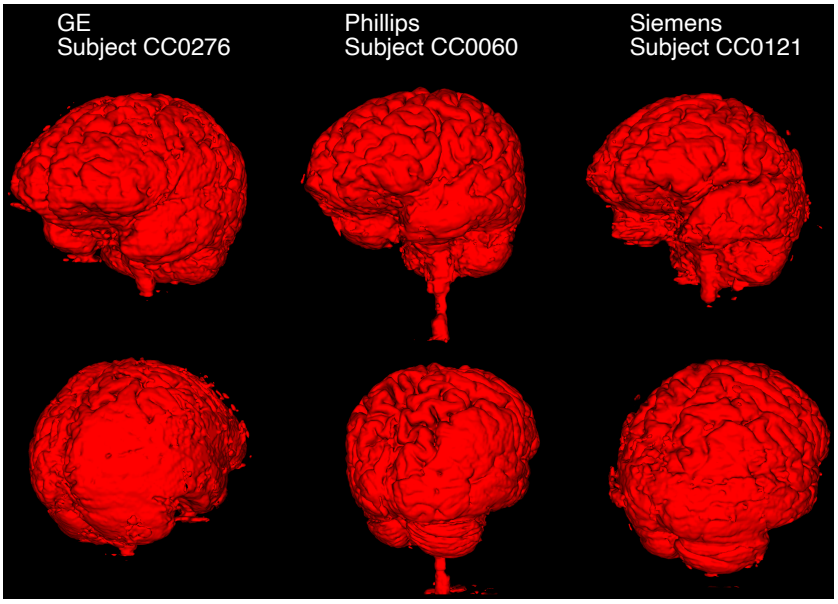

FIGURE 3.7: Exemplary 3D visualization of predicted brain segmentations of data from three different vendors (GE, Philips & Siemens). The segmentation was performed slice-wise in the k-Space domain. For visualization purposes, each slice was transformed into the image domain, stacked and visualized with the ITK-Snap software [81]. Adapted from [51].

3.4 DISCUSSION & CONCLUSION

In this chapter, we presented k-Strip, a novel approach to perform segmentation tasks in the k-Space domain. This first work showed

the feasibility of utilizing k-Space data for downstream tasks, in this case skull stripping. The proposed architecture is based on the U-Net and includes complex-valued building blocks to account for the non-locality and complex-valued nature of the k-Space. k-Strip achieves comparable results to the SOTA methods and outperforms them when using additional phase information. The model is able to accurately predict the skull and non-brain tissue in the k-Space domain, while preserving the phase information. The results show that k-Strip is able to achieve a DSC of 97.4 % and a DHD of 3.42 pixels on the SWI dataset, which surpasses both BSE and OptiBET.

While the results show that k-Strip achieves comparable results to the SOTA methods and outperforms them when using with additional phase information, this study comes with some drawbacks and limitations. We do not use real k-Space data, due to the fact that there is no data publicly available which includes ground truth segmentation masks. The SWI data comes closest to a real-world dataset but is still not fully representative of real k-Space data. Another limitation is the usage of a fixed threshold to create the binary segmentation mask. This threshold was chosen based on quantitative evaluations, but may not be optimal for all datasets. A more sophisticated approach would be to use a learnable threshold, which could be trained together with the model. Only the CC-359 dataset includes manual segmentations for testing purposes. The other datasets only include the results of the SOTA methods as ground truth. This makes it difficult to compare the results of k-Strip to the SOTA methods, as we do not know how accurate the ground truth is.

These drawbacks have to be taken into account for further research, but do not detract from the fact that k-Strip was the first algorithm of its kind to show the feasibility of performing segmentation tasks in the k-Space domain. The importance of this work lies in the fact that it not only demonstrates the possible benefits of utilizing k-Space data, but also prepares a fundamental architecture for future projects.

k-Strip also allows for anonymization of MRI data, even before it is transformed into the image domain. This is a major advantage, as it allows for the anonymization of data before it is interpretable by humans, greatly reducing the risk of privacy breaches.

In chapter 5 we will revisit k-Strip and train it with artificially generated k-Space data to show how to enable the model to learn to generalize on real-world data without the need for actual k-Space

data. We will also utilize k-Strip again in chapter 6 where we will implement it into a publicly available anonymization tool which allows the anonymization of a wide range of medical data, including MRI raw data.

KEY CONTRIBUTIONS

- **Novel Segmentation in k-Space:** Introduced k-Strip, the first deep learning architecture designed to perform segmentation tasks directly in the complex-valued k-Space domain of MRI data.
- **Complex-Valued Deep Learning Model:** Developed a complex-valued U-Net architecture, incorporating novel complex-valued building blocks, such as spectral upsampling and periphery augmentation, to effectively handle the unique properties of k-Space data.
- **Feasibility Demonstrated:** Demonstrated the feasibility of utilizing k-Space data for downstream tasks, achieving comparable results to state-of-the-art methods in skull stripping. This approach allows for the preservation of phase information, which is often lost in traditional image domain processing.
- **Anonymization Potential:** Enable anonymization of MRI data before it is transformed into the image domain, significantly reducing the risk of privacy breaches and enhancing data security in medical imaging.

4

Classification in k-Space: A Case Study

Question 2. Can k-Space data be used to enhance classification results?

Question 3. Can we use k-Space data to implement a faster downstream task pipeline?

The following chapter presents a case study for the classification of prostate cancer likelihood utilizing k-Space data as well as a faster preprocessing method for MRI diffusion data. The presented work titled *Tumor likelihood estimation on MRI prostate data by utilizing k-Space information* was published and presented at the International Society for Magnetic Resonance in Medicine (ISMRM) Conference 2025 as an oral presentation and published in full as a preprint on arXiv [82]. A detailed transparency statement can be found in Appendix A.2. The source code of this project is publicly available on GitHub[1].

[1] https://github.com/TIO-IKIM/Tumor-likelihood-estimation-on-MRI-prostate-data-by-utilizing-k-Space-information

4.1 INTRODUCTION

Worldwide, the availability of MRI scanners as well as required personnel is limited, leading to long waiting times for patients [6]. Long MRI acquisition times are a common problem in MRI, not only leading to a burden for the patients, but also to a high cost for the healthcare system, including valuable resources in form of clinical staff and MRI machines [83]. Thus the need for faster MRI acquisition methods is high. Furthermore, this work is motivated by the principle that MRI

data should be used to its full potential, which is why we propose to use the k-Space data directly for classification tasks.

After we showed that segmentations are possible directly in the k-Space in chapter 3, the next step is to evaluate the benefits of using k-Space data for classification tasks. As a use case we chose the prostate cancer likelihood estimation, firstly because of its importance in the clinical field and secondly because of the availability of a raw MRI dataset with corresponding labels. In 2023, prostate cancer was the most common malignancy among men [84]. The most common method to evaluate the likelihood of prostate cancer is the Prostate-Specific Antigen (PSA) test, digital rectal exams or trans-rectal ultrasound scan (TRUS) [85]. However, these methods introduce a high level of discomfort for the patient, which is why MRI is becoming the preferred method for prostate cancer diagnosis. The resulting MRI scans are commonly categorized into five categories according to the Prostate Imaging Reporting and Data System (PI-RADS) [86]. The PI-RADS system is a standardized reporting system for prostate MRI that helps radiologists communicate findings and assess the likelihood of clinically significant prostate cancer. There have been multiple works presenting deep-learning based approaches for prostate cancer classification [87, 88], however most of them rely only on image domain information and existing segmentations of the prostate, as well as fully reconstructed images.

The last point brings us to the second part of this chapter. As already introduced in the background chapter, a common technique to reduce acquisition times is to apply undersampling in combination with parallel imaging, only filling every n-th k-Space line with multiple different receiver coils. The resulting undersampled k-Space data is then combined and reconstructed using algorithms such as GRAPPA. This reconstruction process can become time-consuming in itself, which is why we propose a faster preprocessing method for the k-Space data, tailored for our classification task which utilizes the raw MRI data. We show that a simple principal component analysis (PCA) can be used to combine the coils, while still preserving the necessary information for the classification task. Additionally, commonly used reconstruction methods such as GRAPPA and ESPIRiT are not suitable for very high undersampling factors, which is why we propose a classification pipeline that is not dependent on an exact reconstruction of the k-Space data. Our results show that this method is significantly outperforming common reconstruction and classification methods at

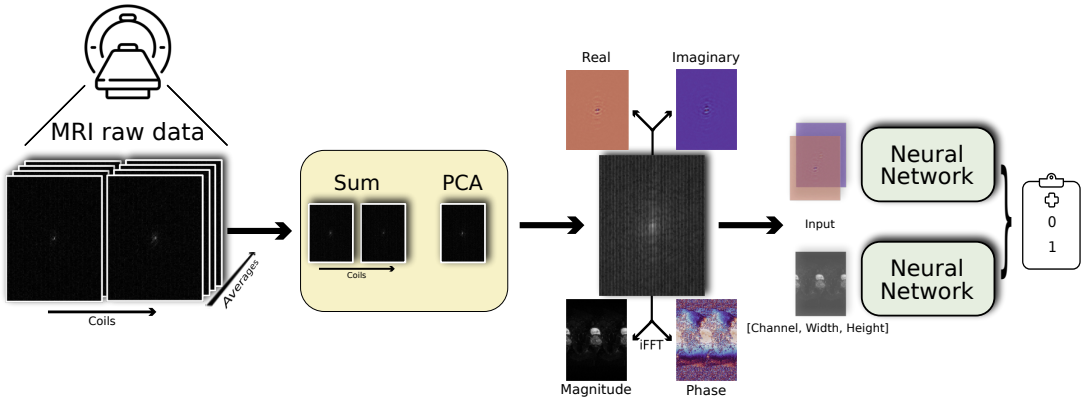

FIGURE 4.1: Methodical overview of the classification pipeline via PCA. The complex-valued MRI raw data 2D slices are summed over the averages and then fed into the PCA. The first PCA component is then split into its image domain magnitude and phase, as well as its k-Space real and imaginary part.

high undersampling factors, enabling fast pre-diagnoses of prostate cancer likelihood. An overview of the proposed method is shown in Fig. 4.1 and will be discussed in the Method section 4.2.

By now there have been some works which show the benefit of using k-Space data for classification tasks; however, they all rely on an adaptive sampling approach, which is currently not feasible in clinical practice. Du et al. [15] propose a framework to use active sampling, which is a method to select the most informative k-Space lines for each individual task, in combination with a classification model which works directly on the k-Space magnitude data. In a similar manner Yen et al. [13] and Singhal et al. [12] propose their *Adaptive Sampling for MR* (ASMR) framework to select the most informative k-Space lines for the classification task. These works show that the k-Space data can be used to improve the classification performance at higher undersampling factors.

4.2 METHODS

In the following chapter, we will discuss the dataset, the novel preprocessing method as well as the classification approach and compare it to existing methods.

4.2.1 *Dataset*

The dataset used in this study is the publicly available FastMRI prostate dataset [89]. We introduce this dataset first, because the further processing steps are based on the datatype of this dataset. The FastMRI Prostate dataset is a one of its kind, as there are no other comparable raw MRI datasets which include clinically validated labels. The

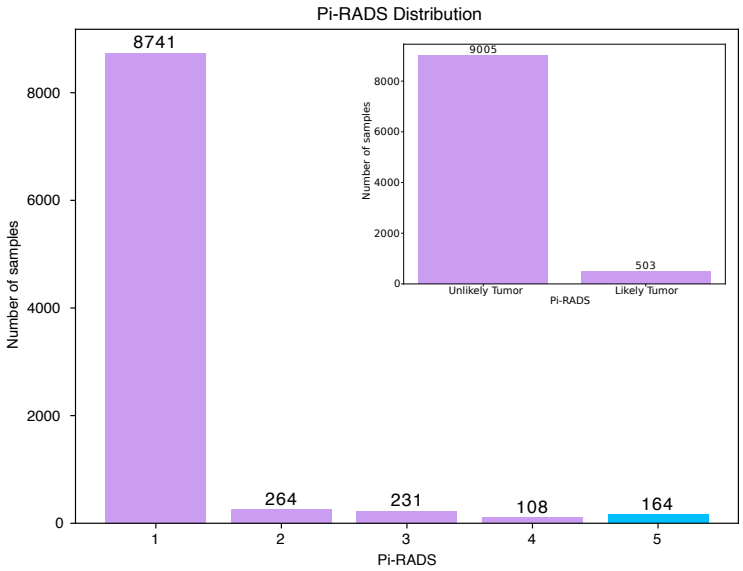

FIGURE 4.2: Distribution of the Pi-RADS scores in the FastMRI prostate dataset. The dataset is highly unbalanced, with a total of 8741 slices with a Pi-RADS score of 1 and only 767 slices with a Pi-RADS score of 2, 3, 4 or 5.

dataset consists of T_2-weighted turbo spin echo (TSE) and echo planar imaging (EPI)-DWI raw data which was acquired with 14 to 30 receiver coils on a 3 T device. Both fundamental MRI modalities were explained in chapter 2. The T_2 data was acquired with three averages, while the DWI data was acquired with 4 and 12 averages.

All DWI scans have an in-plane resolution of 2.0 mm and a matrix size of 100 pixels × 100 pixels. The FOV is then 200 mm × 200 mm. The data is undersampled by a factor of two with an equidistant cartesian sampling mask[2]. The diffusion based MRI data consists of multiple b-values in x, y and z direction, including four averages for B_{50}, 12 averages for B_{1000} and two averages for ${B_0}^3$. 312 subjects where scanned leading to a total of 9508 slices. The corresponding Pi-RADS scores were validated by radiologists on the fully reconstructed image domain data. The Pi-RADS score is a 5-point scale, where 1 indicates a very low likelihood of clinically significant prostate cancer and 5 indicates a very high likelihood. One challenge of this dataset is the highly unbalanced distribution of the Pi-RADS scores. The dataset contains 8741 slices with a Pi-RADS score of 1, while the scores 2, 3, 4 and 5 only contain a total of 767 slices. This leads to a highly unbalanced dataset, which is a common problem in medical imaging datasets. For that reason, we follow the authors of the dataset, as well as the Pi-RADS guidelines, and split the data into two classes, namely "Unlikely Tumor" for classes 1 and 2 and "Likely Tumor" for classes 3, 4 and 5. An overview of the data distribution is shown in Fig. 4.2.

[2]Thus, only every second k-Space line is acquired.

[3]B_0 being the non-diffusion-weighted (T_2-weighted contrast) data.

The datasets main purpose is the task of reconstruction of the undersampled k-Space data, which is why the authors provide a reconstruction pipeline for the data, including the fully reconstructed data. In the following section, we will discuss the reconstruction pipeline and how we can accelerate is for our classification task.

4.2.2 *Reconstruction Pipeline*

The reconstruction pipeline provided by the authors of the FastMRI prostate dataset is based on two very common algorithms in the MRI reconstruction field, namely the “Eigenvalue-based Self-consistent Parallel Imaging Reconstruction" algorithm “ESPIRiT" [90] and the “GeneRalized Autocalibrating Partially Parallel Acquisition” algorithm “GRAPPA" [26].

GRAPPA

GRAPPA is a parallel imaging technique that estimates the missing k-Space data of multicoil acquisitions, by using information of neighbouring k-Space lines. During acquisition, the k-Space data is undersampled by a factor of R. A partition of the center of the k-Space is usually fully sampled. This fully sampled region is called the autocalibration signal region (ACS). The ACS region, is used to calculate the GRAPPA kernel g_{ri}, a kernel function for each coil. These kernels estimate the projected data as well as artifacts of each coil. In combination with the received images from each coil, the GRAPPA kernel is then used to reconstruct the missing k-Space data. The estimation of a k-Space sample x at the position r in the k-Space can be formulated as follows:

$$x_i(r) = (P_r R_r y)^T g_{ri} \quad , \tag{4.1}$$

with P_r being the sampling pattern in the neighbourhood of $x_i(r)$, R_r being the reconstruction operator defining the region of interest and y being the acquired zero-filled k-Space data. The GRAPPA kernel g_{ri} is then calculated based on the fully sampled ACS region by solving Eq. 4.1 for g_{ri}. Finally, the coils are combined using a sum of squares method. While GRAPPA is a common method for undersampled data[4], it is only applicable for relatively small undersampling factors of $R \leq 4$. For larger undersampling factors, the resulting noise in the k-Space data is too high, leading to a poor reconstruction quality [91].

[4] It is also a relatively fast method, if calculated on a GPU. On a CPU the processing time becomes significantly longer.

ESPIRiT

In comparison to GRAPPA, ESPIRiT uses the autocalibration region to estimate the coil sensitivity maps, which are then used to reconstruct the missing k-Space data. This is done by performing a singular value decomposition (SVD) on the fully sampled k-Space data. The resulting projection matrix is then transformed into the image domain, where the coil sensitivity maps are calculated via an eigenvalue decomposition (EVD)[5]. With the coil sensitivity maps at hand, the missing k-Space can then be reconstructed iteratively.

[5]The eigenvalues≈1 are chosen as sensitivity maps. In fully-sampled regions, the calibration data should be consistent with the initial signal.

Coil combination via PCA

Instead of using root squared sum (RSS) or adaptive coil-combination methods in the image domain [92], we propose to use a simple principal component analysis (PCA) to combine the coils directly in k-Space. The PCA is a common method to reduce the dimensionality of data, while preserving the most important information [93]. The core idea of PCA is to find the directions, or principal components, in which the data varies the most. These directions are then used to transform the data into a new coordinate system, where the first principal component corresponds to the direction of maximum variance, the second principal component corresponds to the direction of second maximum variance and so on.

In our work, we apply PCA on the complex-valued k-Space data and choose the first principal component as the combined coil data. In comparison to methods like RSS, this allows us to preserve the phase information of the data to use it for the classification task. By just using the first principal component, we can reduce the dimensionality of the data, but do loose some information which is present in the other principal components. However, we conducted experiments with more than one principal component and found that more than one principal component does not improve the classification performance.

Proposed processing pipeline

The two reconstruction methods presented above are commonly used for the reconstruction of undersampled k-Space data. However, they are both time-consuming and require a lot of computational resources. Additionally, they are not suitable for very high undersampling factors. The overall reconstruction pipeline is shown in Fig. 4.1. We will now go through this pipeline step by step.

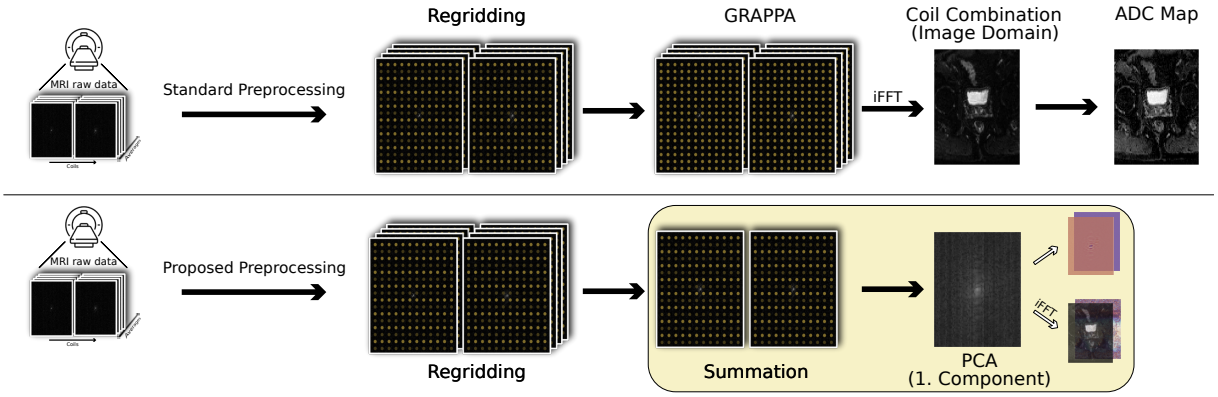

FIGURE 4.3: Comparison of the standard preprocessing pipeline and the proposed preprocessing pipeline. The standard pipeline includes regridding, GRAPPA reconstruction, followed by coil combination in the image domain and ADC map calculation; the phase information is lost. The proposed pipeline first performs regridding, followed by summation and coil compression via PCA. This approach is less computationally intensive and preserves the complex-valued k-Space data. Adapted from [82].

1. Regridding
2. Summing Averages
3. Coil combination via PCA
4. Classification

The DWI data is in a initial format of 5D complex-valued data: Averages, Coils, Depth, Width, Height. The first step is to perform regridding, which is a common method to transform the k-Space data into a Cartesian grid, by interpolating the k-Space data via trapezoidal regridding. Afterwards the averages of the data are summed to reduce the dimensionality of the data and to reduce noise. To simplify the pipeline, in this work we average over all averages instead of splitting the data into the different b-values[6]. The resulting data, now 4D complex-valued data, now has to be combined over the coils. We do not perform any coil sensitivity estimation or calibration, as well as extensive reconstruction such as GRAPPA. Instead we apply the PCA on the complex-valued k-Space data and choose the first principal component as the combined coil data and use it directly for the classification task. This leaves us with a 2D complex-valued data, which is still undersampled by the initial acceleration factor of n.

A comparison between the proposed preprocessing pipeline and a common reconstruction pipeline is shown in Fig. 4.3. The proposed pipeline is significantly faster and less computationally expensive. Additionally, in combination with the classification task, the whole workflow is not limited by the acceleration factor, as we will show in the results section.

[6]The benefit of different b-values is that they can provide complementary information about the tissue microstructure.

To make the results comparable, we adapt the GRAPPA algorithm such that we receive the intermediate complex-valued data after the coil combination step, which we can transform back into the k-Space domain. This allows us to compare the classification results of the proposed pipeline with the GRAPPA reconstruction pipeline.

4.2.3 *Classification Pipeline & Training*

The classification task at hand is a binary problem. The output of the model is a single value between 0 and 1, which indicates the likelihood of prostate cancer. With a threshold of 0.5 we can classify the data into two classes, namely “Unlikely Tumor" and “Likely Tumor".

After the preprocessing steps described above, the resulting data is a 2D complex-valued tensor. While we used a fully complex-valued neural network in the previous work, we will now use a splitting approach to separate the complex-valued tensor into different real valued parts. This decision is based on prior experiments with different networks and had an impact on the further course of this thesis, as we will see in the last work presented in this thesis in chapter 6.

The most popular neural networks are commonly pretrained on vast amounts of data, such as the ImageNet dataset, which includes over 14 million images separated into 22 000 classes [94]. This allows the networks to learn a initial feature representation of the data, which can then be used to transfer knowledge to other tasks[7]. Experiments demonstrate that, in transfer learning scenarios, the data for the final target task do not need to closely resemble the pretraining data from the source domain to yield a positive effect on classification performance [95]. Pretraining becomes even more important, when the dataset is small or unbalanced, as it is the case in the FastMRI prostate dataset. There are currently no pretrained complex-valued neural networks and there is no dataset comparable to the ImageNet dataset for complex-valued data.

[7] One can say, that the model gets a “head-start" for the final task.

In preliminary experiments, we first tried to use the encoder part of the k-Strip architecture for the classification task. A classification head is added to the encoder, which is a simple multilayer perceptron (MLP) with two hidden layers and a final sigmoid activation function. The input data is the 2D complex-valued tensor. This experiments showed that the model is not able to learn the classification task.

To understand if the issue is either the complex-valued neural network or the lack of pretraining, we decided to use a ConvNeXt without pretraining in the next experiment. ConvNeXt is a convolu-

tional neural network architecture, which is a modernized form of the ResNet architecture [96]. The model is implemented with the *timm* library [97]. We split the 2D complex-valued tensor into its real and imaginary part, as well as its magnitude and phase. The experiment is conducted with the real and imaginary part, as well as the magnitude and phase separately and the magnitude alone. The results show that, no matter the input data, the model is not able to learn the classification task

This experiment shows that the issue at hand is the lack of pretraining. Based on this we chose the following approach of using two pretrained ConvNext models, one taking the magnitude in the image domain and one taking the real and imaginary part in the k-Space domain as input. This approach is shown in Fig. 4.1. The output of both models final sigmoid activation function is then averaged and used as the final output of the model.

Undersampling Augmentation

To improve the performance of our model we apply a new augmentation technique we call *Undersampling Augmentation*. This augmentation takes advantage of the uniqueness of k-Space data by artificially undersampling the data with an equidistant cartesian sampling mask with a random acceleration factor of n. An example of the resulting data is shown in Fig. 4.4. Throughout the experiments we will use an undersampling factor of $R \leq 8$ for the augmentation. We investigate the effectiveness of the proposed augmentation technique by comparing the results of both the PCA and GRAPPA pipeline with and without the augmentation. Additionally we apply the augmentation to a off-domain dataset, in this case the CIFAR-10 dataset [98]. The results can be seen in Appendix B.1. The results show that the augmentation can increase the performance for all three datasets.

Hyperparameters

For each of the following experiments the models were trained with the Adam optimizer, a beta coefficient of 0.99, an epsilon of 1×10^{-8} and an initial learning rate of 1×10^{-4} A cosine annealing learning rate scheduler is used to reduce the learning rate over time in a cosine manner. The batch size is set to 128. Early stopping is applied with a patience of 10 epochs to stop the training as soon as the loss does not decrease any more. Due to the high unbalance of the dataset

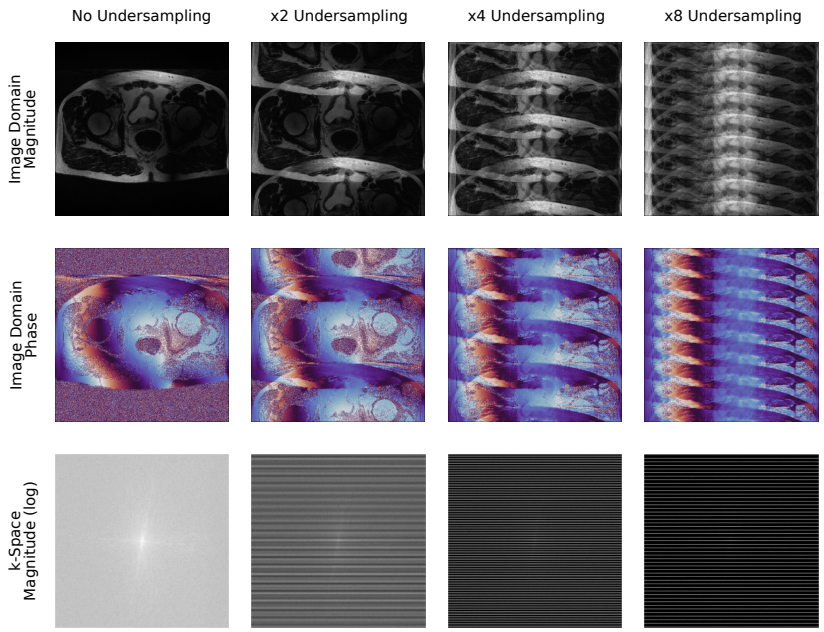

FIGURE 4.4: Example of Undersampling in image domain from a fully sampled sample of the FastMRI+ Dataset. Undersampling refers to omitting lines in the k-Space. While reducing the imaging time, this leads to artifacts in the image domain, as seen in the top rows. The image wraps in itself, depending on the undersampling factor. Published as is in [82].

we decided to use a Cross-Entropy loss with a weight of 17 to 1. In addition to the undersampling augmentation, the input is randomly flipped horizontally. The input is normalized and standardize batch-wise to have a mean of 0 and a standard deviation of 1, applied to each channel individually. Normalization and standardization is applied to the k-Space data in this work, even if it prevents obtaining the original, uncalibrated image domain data. In comparison to k-Strip, this transformation is not necessary in this work, as the model is supposed to learn the classification task directly in k-Space and does not need to predict data which can be transformed in the image domain.

Evaluation Metrics

For evaluating the performance of the models, we use the area under the receiver operating characteristic curve (AUROC), as well as the area under the precision recall curve (AUPRC). The AUROC is a common metric for binary classification tasks and is defined as the area under the curve of the true positive rate (TPR) against the false positive rate (FPR). The AUPRC is defined as the area under the curve of the precision against the recall.

The precision is defined as the number of true positives (TP) divided by the sum of true positives and false positives (FP). The

recall is defined as the number of true positives divided by the sum of true positives and false negatives (FN).

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \end{aligned} \tag{4.2}$$

For the calculation of confidence intervals we apply a bootstrapping method with 1000 iterations [99]. The AUROC and AUPRC are calculated for each iteration and the mean and standard deviation are calculated from the resulting values. The confidence intervals are then calculated as the mean value plus/minus the standard deviation.

4.3 EXPERIMENTS & RESULTS

In this section the conducted experiments and their results will be presented. We will often speak of *additional k-Space information*. This should be understood as the k-Space data which is always gathered during the MRI acquisition, but commonly discarded for any further downstream task.

Experiment 1: Effect of additional k-Space information

In the first experiment we investigate the impact of additional k-Space information on the classification task. For this experiment we use the adapted GRAPPA reconstruction pipeline as well as the PCA pipeline. The experiment is conducted for not undersampled data and an undersampling factor of two. We compare the results for three different data combinations, each with and without the additional k-Space data:

- ADC map, Trace_{50} and Trace_{1000} with GRAPPA pipeline
- Trace_{50} and Trace_{1000} with GRAPPA pipeline
- b_{50} with PCA pipeline

The ADC map is only calculated for the image domain data (magnitude), as it is not possible to calculate the ADC map in k-Space. The resulting AUROC values are shown in Fig. 4.5. Additional AUPRC values can be seen in Tab. 4.2.

The results show that the additional k-Space information leads to an increase of the AUROC and AUPRC values in all cases, except when using all the available data (ADC, Trace_{50} and Trace_{1000}) with

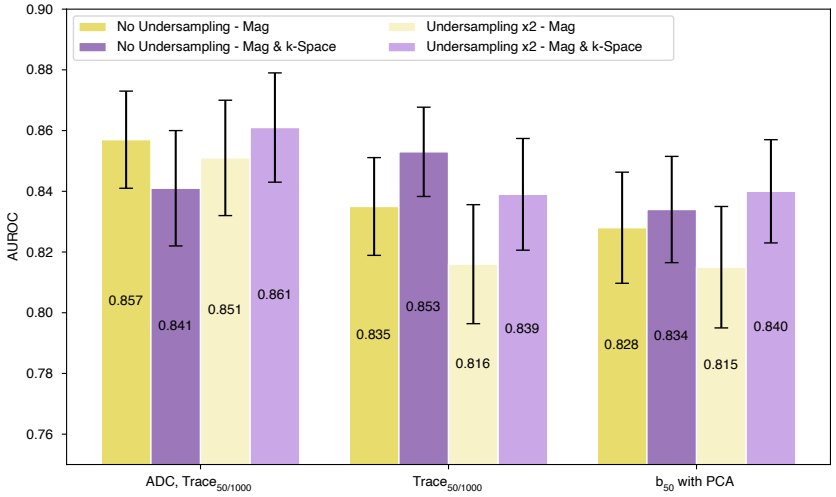

FIGURE 4.5: Results on the effect of additional k-Space information on the classification results of prostate cancer. The AUROC is compared for no undersampling and an undersampling factor of two. Different combinations of data are compared: ADC map, Trace_{50} and Trace_{1000}; Trace_{50} and Trace_{1000}; b_{50}. Adapted from [82].

the GRAPPA pipeline and no undersampling. When applying a undersampling of factor two, the model trained with additional k-Space data already excels the model trained with only the image domain data, with an AUROC of 86.1% and an AUPRC of 64.3%. The benefit of k-Space data is even more pronounced when not using the ADC map[8], where the model trained with additional k-Space data achieves an AUROC of 85.3% and an AUPRC of 62.2%. The model trained with only the image domain data achieves an AUROC of 83.5% and an AUPRC of 59.5%.

[8]This has the benefit of reducing the amount of acquisitions even further.

This first experiment already shows that the additional k-Space information can be beneficial for classification tasks such as prostate tumor classification. The results also suggest that the benefit becomes more pronounced with increasing undersampling factors. Thus, we conducted the next experiment to investigate the impact of the undersampling factor on the classification task.

Experiment 2: Classification at high undersampling rates

High undersampling rates of $R = 8$, $R = 16$ or even higher allow for faster acquisitions, lowering the stress and enabling a higher throughput of patients. Very high undersampling rates in combination with faster downstream pipelines can enable clinicians to perform real-time pre-diagnoses, with follow-up scans only if necessary. To investigate the impact of additional k-Space data at high undersampling rates we remove k-Space lines retrospectively, depending on the simulated acceleration factor. For this the undersampling augmentation described

TABLE 4.1: Results on the effect of additional k-Space information on the classification results of prostate cancer. The AUROC and AUPRC are compared for no undersampling and an undersampling factor of two. Different combinations of data are compared: ADC map, $Trace_{50}$ and $Trace_{1000}$; $Trace_{50}$ and $Trace_{1000}$; b_{50}.

Method & Undersampling	AUROC (%) ↑	AUPRC (%) ↑
Image Domain (ADC, $Trace_{50}$ & $Trace_{1000}$)		
x0	**85.7±1.6**	**64.6±2.3**
x2	85.1±1.9	63.4±2.2
Image Domain & k-Space		
x0	84.0±1.9	63.5±2.2
x2	**86.1±1.8**	**64.3±2.3**
Image Domain ($Trace_{50}$ & $Trace_{1000}$)		
x0	83.5±1.6	59.5±1.5
x2	81.6±1.9	59.5±1.7
Image Domain & k-Space		
x0	**85.3±1.5**	**62.2±2.0**
x2	**83.9±1.8**	**60.1±1.7**
k-Space (b_{50})		
x0	82.8±1.8	60.5±1.9
x2	81.5±2.0	58.8±1.5
Image Domain & k-Space		
x0	**83.4±1.7**	**63.4±2.3**
x2	**84.0±1.7**	**62.3±2.2**

before is used at an acceleration factor between 2 and 8. Compared are the results of the PCA pipeline with the GRAPPA pipeline. Both approaches are compared with and without additional k-Space information and phase data:

- PCA / GRAPPA pipeline with magnitude data in image domain
- PCA / GRAPPA pipeline with magnitude and phase data in image domain
- PCA / GRAPPA pipeline with magnitude and additional k-Space data

The results are shown in Fig. 4.6. Both pipelines benefit from additional k-Space information, especially when applying higher undersampling rates. At undersampling rates of $R \geq 8$, the classification results of PCA pipeline with only the magnitude data starts to decrease significantly, while the additional k-Space data is able to stabilize the classification results. At an undersampling factor of $R = 24$ the PCA pipeline with magnitude data achieves an AUROC of $67.2\% \pm 3.1\%$, while the same pipeline with additional k-Space information achieves

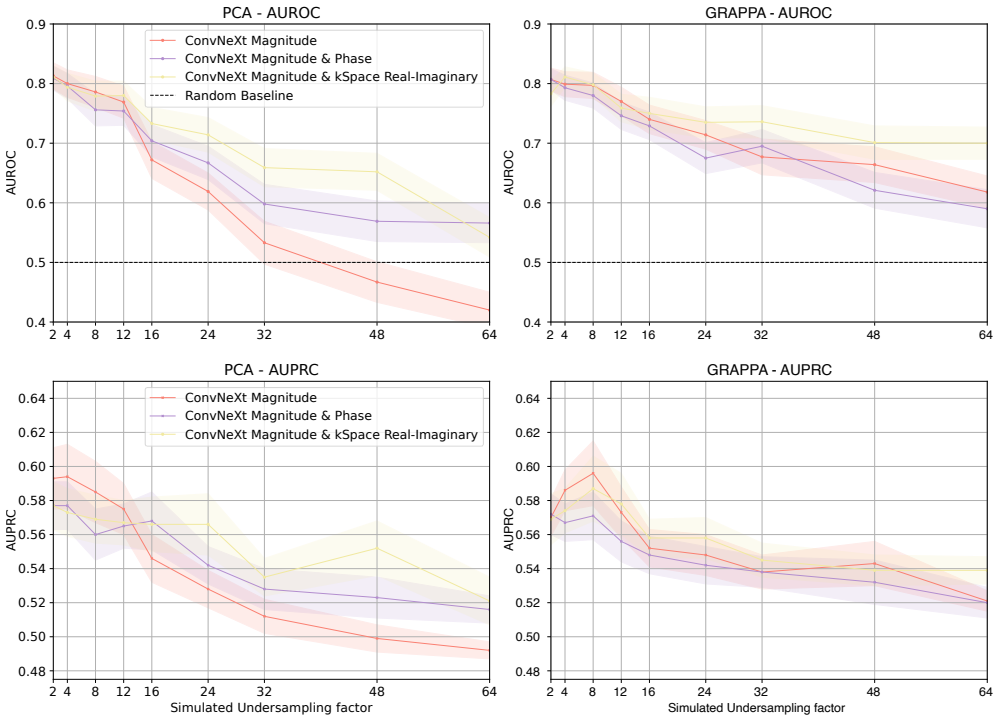

FIGURE 4.6: PCA (left) and GRAPPA (right) results (top: AUROC; bottom: AUPRC) for the PCA test dataset. Compared are the inputs magnitude in image domain (i) only, magnitude and phase in image domain, as well as magnitude in image domain real and imaginary part in k-Space (k). The AUPRC baseline is 0.051. The model has been trained with the undersampling augmentation of a factor up to x8. Adapted from [82].

an AUROC of 73.3% ± 2.7%. With a baseline AUROC of 50%, which would indicate a random guess, the PCA pipeline with magnitude data falls under said baseline at a undersampling factor of $R = 40$, while the PCA pipeline with additional k-Space information achieves an AUROC of 65.9% ± 3.2%. The model trained with additional phase data, but no k-Space data outperforms the magnitude-only model, but does not reach the performance of the model trained with additional k-Space data. The same results can be seen for the GRAPPA pipeline, even though the difference between the models trained with different data types is not as pronounced as in the PCA pipeline.

The proposed pipeline is computationally less expensive that the GRAPPA pipeline. On a CPU, the combined acquisition and reconstruction time of the GRAPPA pipeline is roughly 36 minutes, while the proposed PCA pipeline takes only 27.9 minutes. At a undersampling rate of $R = 16$, the GRAPPA pipeline takes 12 minutes, while the PCA pipeline takes only 3.9 minutes, a reduction of 32.1 minutes in comparison to the fully sampled GRAPPA pipeline. A detailed overview of the time consumption of the different pipelines per volume can be seen in Tab.B.2.

The lower AUROC and AUPRC in comparison to the first experi-

TABLE 4.2: Results on the effect of additional k-Space information on the classification results of prostate cancer. The AUROC and AUPRC are compared for different undersampling factors and for both the GRAPPA and PCA pipeline. Different combinations of data are compared: Magnitude (i) only, Magnitude (i) and Phase (i), Magnitude (i) and Real & Imaginary (k). (i): Image domain; (k): k-Space. Adapted from [82].

	GRAPPA		PCA	
	AUROC (%) ↑	AUPRC (%) ↑	AUROC (%) ↑	AUPRC (%) ↑
x2 Undersampling				
Magnitude (i)	80.7±1.9	57.0±1.1	81.3±2.2	59.3±1.8
+ Phase (i)	80.7±1.8	57.2±1.2	80.9±2.1	57.7±1.4
+ Real & Imaginary (k)	78.3±2.1	56.9±1.5	81.1±1.9	57.7±1.4
Simulated x4 Undersampling				
Magnitude (i)	79.9±2.1	58.6±1.2	80.0±2.3	59.4±1.9
+ Phase (i)	79.3±2.1	56.7±1.1	79.6±2.1	57.7±1.4
+ Real & Imaginary (k)	**81.1±1.7**	57.4±1.3	79.4±2.3	57.3±1.4
Simulated x8 Undersampling				
Magnitude (i)	79.7±2.2	59.6±1.9	78.6±2.6	58.5±1.8
+ Phase (i)	78.0±2.1	57.1±1.4	75.6±2.7	56.0±1.5
+ Real & Imaginary (k)	**79.8±2.0**	58.7±1.9	77.9±2.5	57.5±1.4
Simulated x16 Undersampling				
Magnitude (i)	74.9±2.4	52.2±1.1	67.2±3.1	54.6±1.4
+ Phase (i)	72.9±2.3	54.8±1.1	70.4±2.7	**56.8±1.7**
+ Real & Imaginary (k)	**75.0±2.7**	55.8±1.1	**73.3±2.7**	**56.6±1.6**
Simulated x24 Undersampling				
Magnitude (i)	71.4±2.4	54.8±1.2	61.9±3.1	52.8±1.1
+ Phase (i)	67.5±2.6	54.2±1.1	66.7±2.8	54.2±1.1
+ Real & Imaginary (k)	**73.5±2.6**	**55.8±1.2**	**71.4±2.9**	**56.6±1.8**
Simulated x32 Undersampling				
Magnitude (i)	67.7±3.0	53.8±1.0	53.4±3.6	51.2±1.0
+ Phase (i)	69.5±2.8	53.8±0.9	59.8±3.3	52.8±1.2
+ Real & Imaginary (k)	**73.6±2.7**	**54.5±1.0**	**65.9±.3.2**	**53.5±1.1**
Simulated x48 Undersampling				
Magnitude (i)	66.4±3.0	54.3±1.0	46.7±3.4	49.9±0.8
+ Phase (i)	62.1±3.0	53.2±1.3	56.9±3.4	52.3±1.2
+ Real & Imaginary (k)	**70.1±2.8**	**53.9±0.9**	**65.2±3.1**	**55.2±1.6**
Simulated x64 Undersampling				
Magnitude (i)	61.8±2.7	52.1±0.6	42.0±3.0	49.2±0.5
+ Phase (i)	59.0±3.2	52.0±0.9	**56.6±3.3**	51.6±0.8
+ Real & Imaginary (k)	**70.0±2.7**	**53.9±0.8**	54.2±3.3	**52.1±1.4**

ment can be explained by the fact that the model is trained with the undersampling augmentation, which leads to better results at higher undersampling factors, but the results are worse at lower undersampling factors.

4.4 DISCUSSION & CONCLUSION

In this chapter, we investigated the impact of additional phase or k-Space information on classification results such as the likelihood estimation of prostate cancer. Additionally, we proposed a new preprocessing pipeline which is based on a simple PCA and is less computationally expensive and faster than the standard pipeline based

on GRAPPA. Due to the lack of pretrained complex-valued neural networks, we used the split-approach and used two pretrained ConvNeXt networks, one for the image domain data and one for the k-Space data. The results show that the additional k-Space information is beneficial for classification tasks such as prostate tumor classification. The results also suggest that the benefit becomes more pronounced with increasing undersampling factors. The PCA based pipeline in combination with higher undersampling rates can reduce the acquisition time significantly, while still achieving meaningful classification results. This can enable real-time pre-diagnoses of prostate cancer, with follow-up scans applied only if necessary. This is a promising step towards faster and more efficient MRI acquisitions, which might lead to a higher throughput of patients and a better patient experience. Again, we have to emphasize that the k-Space data utilized in this work is always acquired and does not need to lead to longer acquisition times. We simply use the data which is already available, instead of discarding it after image reconstruction, as commonly done.

In this project we did have certain limitations. The undersampling applied retrospectively is not the same as the undersampling applied during acquisition. The retrospective undersampling can lead to different artifacts in the data, which can have an impact on the classification results. By measuring less k-Space lines, the signal-to-noise ratio (SNR) is reduced [100], which is not reflected in the retrospective undersampling. Motion artifacts can also be reduced by the shorter acquisition times [101]. In case of DWI acquisitions, higher undersampling rates during EPI can lead to less $T2^*$ wighting and thus a change in image contrast [102]. Implementing these physical changes to the augmentation is a promising step to improve the generalization of the model.

In addition, the GRAPPA reconstruction would be applied after the undersampling. While this could lead to better results at low undersampling factors, as steeper performance drop-off at higher undersampling rates with the GRAPPA method is to be expected, due to said artifacts and higher noise amplification.

We stated that the first PCA component is used for the classification task, as the other components do not improve the classification performance. However, this does not mean that the other components are not useful for other tasks or acquisitions. The applicability of the PCA pipeline needs to be validated for other datasets and tasks.

Another factor which likely has an impact on the classification

results is the lack of cropping in the presented pipeline. Usually, the data is cropped to a fixed size such that only the prostate is visible. This is not the case in the presented pipeline, as the data is not cropped and the whole image is used for the classification task. Manipulating k-Space data, such as by cropping, requires careful consideration. For instance, removing high-frequency components (cropping the periphery of k-Space) primarily reduces spatial resolution, leading to image blurring. This is distinct from changing k-Space sampling density, which affects the Field of View. Techniques of k-Space cropping have to be investigated in future works.

With this work we again showed the possible benefits of using all the available data gathered in an MRI acquisition. After already showing the feasibility of performing segmentation tasks and reaching state-of-the-art performance in chapter 3, we now showed that additional k-Space data can excel the results of SOTA algorithms for tasks such as prostate likelihood estimation. With the proposed approach the acquisition time can be reduced by up to 90% while still achieving meaningful results, enabling clinicians to perform real-time pre-diagnoses of prostate cancer.

KEY CONTRIBUTIONS

- **Validation of k-Space Processing for Diagnostic Classification:** Successfully applied k-Space-based deep learning to a clinical classification task (prostate cancer likelihood estimation), demonstrating that the benefits observed in segmentation extend to diagnostic prediction and improve accuracy.
- **Novel, Computationally Efficient Preprocessing Pipeline:** introduced a new pipeline using Principal Component Analysis (PCA) for direct k-Space coil combination. This approach bypasses the need for full, time-consuming image reconstruction, offering a significantly faster and more direct path from raw data to clinical insight.
- **Enhanced Performance in Accelerated Imaging Scenarios:** Quantified the advantage of including k-Space features in undersampled acquisitions, showing that the proposed method maintains higher accuracy and robustness at acceleration factors where traditional image-domain models begin to fail.

5

PhaseGen: Generative AI meets k-Space

Question 4. Can we generate synthetic MRI raw data to train our models?

In the previous chapter we used a split-approach including two pretrained ConvNeXt networks. The pretraining is performed on image domain data, which is not comparable to the complex-valued k-Space data. The question arises if a complex-valued neural network pretrained on complex-valued raw MRI data could outperform the split-approach. Could we also improve the performance of CVNNs used for segmentation or reconstruction tasks? While intriguing, this line of thought immediately confronts a major challenge: the scarcity of large-scale, publicly available raw MRI datasets required for such pretraining. We will now investigate a novel generative AI model to tackle exactly this challenge. We introduce *PhaseGen*, a diffusion-based complex-valued generative model, which is able to generate complex-valued MRI data, enabling pretraining of complex-valued neural networks.

Most of the results in the following chapter have been published in our paper "*PhaseGen: A Diffusion-Based Approach for Complex-Valued MRI Data Generation*" [103]. A detailed transparency statement can be found in Appendix A.3. The source code of this project is publicly available on GitHub[1].

[1] https://github.com/TIO-IKIM/PhaseGen

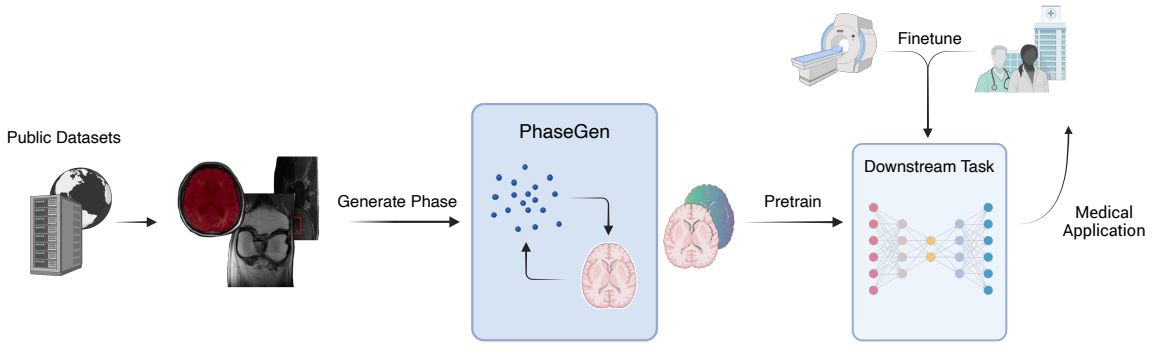

FIGURE 5.1: Overview of the proposed approach. Publicly available magnitude image-domain data can be used to generate a large amount of synthetic raw MRI data by using the proposed generative AI model *PhaseGen*. The generated data can then be used to pretrain neural networks for downstream tasks. The pretrained model can be fine-tuned on the limited amount of available real-world MRI raw data. Figure published as is in [103].

5.1 INTRODUCTION

The disparity in publicly available MRI data is stark: for every subject with raw MRI data, there are approximately 168 subjects for whom only magnitude image-domain data is available[2] [89, 104, 105, 106, 107]. This staggering difference is probably one reason for the lack of research in the field of MRI raw data and complex-valued neural networks in this field. This lack of available data also prevents existing models from fully reaching their potential due to the lack of pretraining. There are two possible solutions for this problem: (1) Gathering more data in the form of raw MRI data in the clinics or (2) generating synthetic MRI raw data, which resembles real MRI data as closely as possible. While (1) would be the ideal solution, it is expensive, time consuming and not likely to be achieved in the near future. Therefore, we propose to generate synthetic MRI raw data using generative AI. The vast amount of data which can be generated can be used for pretraining complex-valued neural networks. The limited amount of available real-world MRI raw data can then be used to fine-tune the pretrained model. This approach allows us to utilize the data available as efficient as possible. An overview of the suggested approach is shown in Fig. 5.1.

[2]In total there are roughly 1,68 million magnitude image-domain subjects [104] and 10 000 raw MRI subjects publicly available.

The presented complex-valued diffusion model does not only present a novelty in the field of k-Space research by enabling the generation of artificial raw MRI data. It is also the first complex-valued diffusion-based model and thus presents an important step towards the application of generative AI in the field of complex-valued data.

5.2 METHODS

5.2.1 *Generative AI*

"The term generative AI refers to computational techniques that are capable of generating seemingly new, meaningful content such as text, images, or audio from training data." [108]

Generating new data is not only a fascinating task, but can also be used to improve existing techniques. Data augmentation is a well-known technique to artificially increase the amount of training data by applying transformations to existing data. Approaches such as shifting, rotating or flipping images are widely used in the field of computer vision and can have a great impact on the model performance [109]. It has to be noted that these transformations only alter the existing data, while preserving the underlying information. With the help of generative models new data can be created, which is not only a transformation of existing data. The application of generative AI as a tool for data augmentation showed promising results in recent works, surpassing classical augmentation techniques [110, 111, 112].

In recent years, denoising diffusion probabilistic models (DDPMs)[3] [114] have gained significant attention due to their impressive performance in generating high-quality images. Diffusion-based models are the latest stage in the field of generative AI. Their most prominent predecessors are Variational Autoencoders (VAE) [115], which are based on the idea of encoding data into a lower-dimensional latent space and then decoding it back to the original space. By introducing a probability distribution in the latent space, VAEs are able to generate new samples by sampling from this distribution. However, VAEs often struggle with generating high-quality images due to the limitations of the latent space representation. Generative Adversarial Networks (GANs) [116] were introduced to overcome these limitations by using a two-player game between a generator and a discriminator. The generator tries to create realistic samples while the discriminator tries to distinguish between real and generated samples. While GANs can produce high-quality images, they are often difficult to train and can suffer from mode collapse [117].

Diffusion models, on the other hand, are based on a different approach. During training, they learn to reverse a diffusion process that gradually adds noise to the data. Diffusion models have been proven to generate higher quality samples compared to GANs and VAEs, closer resembling real samples and thus leading to better results

[3] There are deviations from the original DDPM method, such as latent diffusion models (LDM), which generate new data in the latent space which is then decoded [113]. In this work, when we talk about "diffusion models", DDPMs are meant.

when used for training models on downstream tasks [118, 119].

5.2.2 *(Complex-valued) Diffusion Models*

The objective of a diffusion-based model is to predict the added noise at each step[4]. We will first explain the general approach of diffusion models and then extend the definition to complex-valued diffusion models. The *forward diffusion process* is defined as a Markov chain, which gradually adds Gaussian[5] noise to the data. The noising step of a data point x_t at a time step t during the diffusion process with the noise ϵ sampled from a Gaussian distribution $\mathcal{N}(0, I)$ can be defined as

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t I), \tag{5.1}$$

where β_t is a hyperparameter controlling the amount of noise added at each step and I is the identity matrix. The scheduler β_t ranges between 0 and 1, depending on the applied scheduler and the current time step. The forward process can be defined as a sequence of T steps, where the initial data point x_0 is gradually transformed into a noise distribution x_T. The model is now trained by performing the *reverse diffusion process*, which aims to predict the added noise at each step. The reverse process is defined as

$$\begin{aligned} p_\theta(x_{t-1}|x_t) &= \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \\ \text{with} \quad \mu &= \sqrt{1-\beta_t} \cdot x_{t-1} \quad . \end{aligned} \tag{5.2}$$

θ defines the parametrization of the model, while μ_θ and Σ_θ are the mean and covariance of the predicted noise. The model is trained to minimize the difference between the predicted noise and the actual noise added during the forward process. In the original work, this loss function is chosen to be the KullbackLeibler (KL) divergence [121], which is a measure of how one probability distribution diverges from a second expected probability distribution. Due to better implementations of the loss functions, by now it is more common to use other loss functions like the mean-squared error (MSE). To sample new data during the inference, the model starts with a random noise sample x_T and iteratively applies the reverse diffusion process, depending on the learned parametrization. This random noise sample can be accompanied by a conditioning variable to control the generated data. The conditioning variable can be a text prompt, an image or any other type of data. The model is trained to learn the joint distribution of the data and the conditioning variable, allowing it to generate new samples that are consistent with the conditioning variable.

[4] Here is a fine but important difference to the other approaches. The model is not learning to predict the original image, but the noise which was added to the image.

[5] Other types of noise can be applied, but Gaussian noise proved to be more stable during training [120].

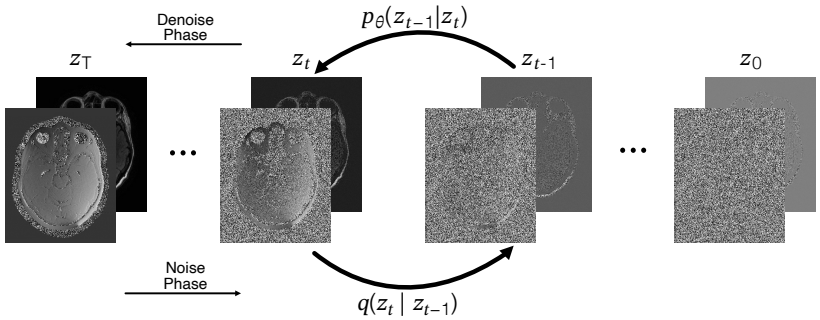

FIGURE 5.2: Forward and backward diffusion process. The forward diffusion process is defined as a Markov chain, which gradually adds complex valued noise to the data. The backward diffusion process aims to predict the added noise at each step. The proposed complex-valued diffusion process aims to mainly superimpose noise onto the phase. Figure published as is in [103].

The described approach is well defined for real-valued data. However, there have been no works to translate this idea to complex-valued data. The main challenge is the definition of the noise distribution and the reverse process. In the following, we will show how to extend the diffusion model to complex-valued data. Our approach is tailored for the usecase of generating complex-valued MRI data based on a conditioning variable in form of magnitude image-domain data. We want our model to focus on the generation of the phase and its complex-valued connection to the magnitude in the image domain. The magnitude should be preserved. Thus the noise needs to be adapted such that it mainly influences the phase of the complex-valued input data z_t. The proposed complex-valued noise distribution $\mathcal{N}_z$ is defined as

$$\mathcal{N}_z = 1 \cdot e^{i\epsilon}; \;\; \epsilon \backsim \mathcal{U}(-\pi, \pi) \quad ,$$

with $\mathcal{U}$ being the uniform distribution in the range of $(-\pi, \pi]$ to ensure that the noise is periodic.

We formulate the forward diffusion process as

$$z_t = (|z_{t-1}|\sqrt{\alpha_t} + |\epsilon|\sqrt{1-\alpha_t}) \cdot \exp(\angle z_{t-1} + \angle\epsilon\sqrt{1-\alpha_t}) \quad , \quad (5.3)$$

with $\alpha_t := 1 - \beta_t$. $|.|$ and $\angle$ denote the magnitude and phase respectively. In Eq. 5.3 we can see, that the noised magnitude and phase are calculated with the same scheduler and combined via the complex exponential function. This approach does not fully preserve the magnitude, but allows us to minimize the effect on the magnitude while focusing the diffusion steps on the phase information. The resulting forward and backward diffusion process is show in Fig. 5.2 and presented as algorithmic form in Alg. 5.1.

Algorithm 1 Training

repeat
  $z_0 \backsim q(z_0)$
  $t \backsim \text{Uniform}(\{1, ..., T\})$
  $\epsilon \backsim \mathcal{N}_z$
  Take gradient descent step on
    $\bigtriangledown_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}} z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2$
until convergence

Algorithm 2 Sampling

$z_T \backsim \mathcal{N}_z$
for $t = T, ..., 1$ **do**
  $\eta \backsim \mathcal{N}(0, \text{I})$, if t > 0, else $\eta = 0$
  $z_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t \eta$
end for
return z_0

ALGORITHM 5.1: Algorithms for training and sampling in the complex-valued diffusion model. Adapted from [114] and published as is in [103].

As a loss function we use the mean absolute error (L1 loss) between the predicted noise ϵ_θ and the actual noise ϵ added during the forward diffusion process. The model architecture we use for our complex-valued diffusion model is an adapted version of the *k-Strip* algorithm presented in chapter 3. The model is extended with a positional encoding to encode the timestep t. This is done by adding a sinusoidal encoding. In our model we use two sinusoidal encodings, one for the real and one for the imaginary part, which are then concatenated into a two-channel tensor.

Let:

- $t \in \mathbb{R}^{B \times 1}$ be a tensor of timesteps (broadcasted),
- C be the total number of channels (even),
- $\omega_i = \frac{1}{10000^{\frac{2i}{C}}} \quad \text{for } i = 0, 1, \ldots, \frac{C}{2} - 1,$
- $\text{inv\_freq} = [\omega_0, \omega_1, \ldots, \omega_{\frac{C}{2}-1}] \in \mathbb{R}^{\frac{C}{2}}.$

Then the positional encoding is given by:

$$\begin{aligned} \text{pos\_enc}_a &= \sin(t \cdot \text{inv\_freq}) \\ \text{pos\_enc}_b &= \cos(t \cdot \text{inv\_freq}) \\ \text{pos\_enc} &= \left[\text{pos\_enc}_a \,\|\, \text{pos\_enc}_b\right] \end{aligned}$$

where || denotes concatenation along the channel dimension.

The inference is then performed by sampling a random complex-valued noise sample z_T. An image-domain magnitude image is used as a conditioning variable c in a second channel. The model is then applied to the complex-valued noise sample and the conditioning variable. The model predicts the noise ϵ_θ at each step, which is then used to iteratively apply the reverse diffusion process. In each step c is added into the magnitude of the complex-valued tensor to further guide the diffusion and preserve the image domain information. The final output is a complex-valued tensor z_0 with the initial magnitude c and a phase generated by the diffusion model.

5.3 EXPERIMENTS & RESULTS

5.3.1 *Evaluation Metrics*

To validate generative machine learning algorithms, suitable evaluation metrics have to be defined. There have been several works which proposed metrics to evaluate the performance of generative models. The most common metrics to evaluate the perceptual quality are the Fréchet Inception Distance (FID) [122, 123] and the Inception Score (IS) [124]. Both of these metrics evaluate the quality of the generated images by analyzing the distribution of the generated images. The FID compares the distribution of the generated images to the distribution of the real images, while the IS only evaluates the distribution of the generated images. Both of these metrics are defined for real-valued images, as they utilize a pretrained *Inception network* [125] to extract features from the images. Common implementations of the FID and IS metrics are not able to handle complex-valued data. Therefore, we use a different approach which is commonly used to evaluate the generated data of diffusion models: The evaluation on downstream tasks such as classifications [126, 127].

In our case, we will evaluate the generated data on the task of image reconstruction and brain segmentation. For the task of brain segmentation we revisit the work presented in chapter 3 and use the same model architecture. For the task of image reconstruction we use a publicly available dataset which is commonly used for the training of MRI reconstruction algorithms. For both tasks we compare the performance of the model trained with three different sets of training data:

- **No Phase**: The model is trained on complex-valued data without any phase information.
- **Naive Phase**: The model is trained on complex-valued data with phase data generated via a naive approach based on a sinusoidal function.
- **PhaseGen**: The model is trained on complex-valued data with phase data generated via the proposed generative model *PhaseGen*.
- In case of the reconstruction task we also train with the available real-phase data.

The *No Phase* data can be defined as

$$z = |x| \cdot e^{i \cdot 0} = |x| \quad ,$$

where x is the magnitude image-domain data. To simulate the *Naive Phase* data, we define a function which superimposes sinusoidal functions along both axis of the image, based on the magnitude data. To simulate the phase more closely to real MRI raw data, small fluctuations σ of noise are added to the sinusoidal function. We define this novel approach as

$$\phi(x,y) = [\sin\left(\frac{2\pi x}{N}\right) + \cos\left(\frac{2\pi y}{N}\right)] \cdot \hat{M}(x,y) + \eta(x,y) \quad , \quad (5.4)$$

with x and y being the pixel coordinates, N being the size of the image, $\hat{M}$ being the magnitude image-domain data and η being a Gaussian noise with $\sigma = 0.05$.

For the evaluation of the skullstripping task we will use the already introduced metrics DSC and HD. For the task of MRI reconstruction, we will use the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [128]. The PSNR is defined as

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right) \quad , \quad (5.5)$$

with MAX being the maximum possible pixel value of the image and MSE being the mean squared error between the ground truth and the generated image. The PSNR is a common metric to evaluate the quality of reconstructed images by giving a measure of the difference between the maximum power of a signal and the power of noise in the signal.

The SSIM is defined as

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad , \quad (5.6)$$

with x and y being the two images, μ_x and μ_y being the mean of the two images, σ_x^2 and σ_y^2 being the variance of the two images and σ_{xy} being the covariance of the two images. The constants c_1 and c_2 are used for numerical stability [129]. The SSIM takes on values between -1 and 1, with 1 indicating a perfect match between the two images. The SSIM is a perceptual metric which takes into account the luminance, contrast and structure of the images.

Additionally we use the *Mean Squared Error* (MSE) and the *Normalized Root Mean Squared Error* (NRMSE) as additional metrics. The MSE is defined as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \quad , \tag{5.7}$$

with N being the number of pixels in the image and x_i and y_i being the pixel values of the two images. The NRMSE is defined as

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2}}{\max(x) - \min(x)} \quad , \tag{5.8}$$

with $\max(x)$ and $\min(x)$ being the maximum and minimum pixel value of the image. The NRMSE is a normalized version of the MSE, which takes into account the range of the pixel values.

For both downstream-tasks, the data is transformed into the k-Space. If the proposed generative model manages to generate realistic MRI raw data, the models trained with the generated data should be able to outperform the models trained with the naive phase data and no phase data. We do not expect the generated data to excel the results on real data, as this would indicate flaws in the experimental setup.

In the following, we will go into more detail about the two downstream tasks, the training procedure and the used datasets. Additionally, we present our in-house MRI raw dataset utilized for training PhaseGen.

5.3.2 *Datasets*

In-House MRI Raw Data

To be able to generate a wide variety of MRI raw data, we want to use a heterogeneous dataset, including different MRI sequences and anatomies. In the course of this work, we collected a dataset of roughly 1500 MRI raw data subjects in the University Hospital Essen [24-11872-BO]. The dataset includes different MRI sequences such

as T_1, T_2 and FLAIR. A detailed list of the included sequences can be found in Appendix Tab. B.4. It also includes different anatomies such as brain, abdomen and knee. All subjects were acquired on a 1.5 T or 3 T MRI scanner (Siemens Magnetom Aera or Siemens Magnetom Skyra). The data was acquired in a clinical setting and thus includes a wide variety of subjects, including pathological data. The initial raw data is saved in the *Twix* format and stored on the MRI devices for two weeks before being deleted. During that time period we manually export each scan to our in-house storage. Before further processing all data is anonymized by removing all patient-related metadata using an anonymization tool developed during this thesis. This anonymization tool will be presented in chapter 6.

With the help of a reconstruction pipeline based on GRAPPA and the Python tool *Twixtools* [130], the Twix raw data is converted into a three-dimensional tensor with the shape $[N, H, W]$, with N being the number of slices, H and W being the height and width of the image. The resulting data is split into individual slices and stored as 2D PyTorch tensor files.

For training of the PhaseGen diffusion model, we utilize 12 071 2D raw data slices from 390 subjects, including different anatomies and resolutions. We do not utilize the whole dataset, as not all sequences can be reconstructed with our pipeline at the time of the experiments. Each slice is transformed into the image-domain and reshaped into the shape of 256×256 pixels. The dataset is split into a training and validation set with a ratio of 80:20.

Skullstripping Dataset

For the task of skullstripping, which is our first validation experiment, we use the in-house *GBM* dataset already used in chapter 3. The dataset includes 21 822 2D slices from 150 subjects. The magnitude image-domain brain MRI data was acquired on 1.5 T and 3 T MRI scanners and partially includes pathologies. The corresponding phase data for input and ground truth is generated with the methods listed above and transformed into the k-Space. The trained models are evaluated on 14 real-world MRI raw data volumes scanned on the same machines as the In-House raw dataset. We generate the corresponding ground truth by applying the two skullstripping algorithms *HD-BET* and *Synthstrip*, as well as our de-identification tool which will be presented in chapter 6. The generated segmentation masks of the three algorithms are then combined into a single mask using the

STAPLE algorithm. The data is initially stored as NIfTI and converted into 2D PyTorch tensors.

By testing on real-phase data, we can test the ability of generalizing to real-world data without including them in the training data. If the model is able to generalize to real-world data, it indicates that the generated data is realistic and can be used for pretraining of complex-valued neural networks.

Reconstruction Dataset

For the reconstruction downstream-task we utilize the FastMRI single coil knee dataset [105]. The dataset includes 41 877 2D slices scanned on 1.5 T (Siemens Magnetom Skyra, Prisma and Biograph mMR) and 3 T (Siemens Magnetom Aera) MRI devices. The data was acquired with a 15 channel knee coil array and a 2D TSE acquisition with Cartesian sampling. All scans were acquired at the NYU School of Medicine during clinical routine. The initial data is store in ISMRMRD h5 format [131] with a in-plane resolution of 0.5mm × 0.5mm and a FOV of 320x320 pixels. To simulate single-coil data based on the multi-coil data, the authors of the FastMRI dataset employ "an emulated single-coil (ESC) methodology [132]"[6]. The ground truth of this data is the fully sampled k-Space data, which is then reconstructed using the root-sum-of-squares method on the multi-coil data [133].

[6] "ESC computes a complex-valued linear combination of the responses from multiple coils, with the linear combination fitted to the ground-truth root-sum-of-squares reconstruction in the least-squares sense." [105]

We load the data using the *h5py* python library and convert the data into 2D PyTorch tensors [30].

The authors propose a custom dataset split with a training-set size of 34 742 slices (973 volumes), a validation-set size of 7135 slices (199 volumes) and a test-set size of 3903 slices (108 volumes). We use the same split for our experiments. The data is already in the k-Space and includes real-phase data. To compare the performance of the models trained with generated data, we transform the k-Space data into the image domain and remove the phase information. The phase is then genrated by applying the different methods described above. The generated data is then transformed back into the k-Space.

5.3.3 *PhaseGen Training & Inference*

We train PhaseGen with 1000 timesteps and an ϵ-value of 0.008 in the diffusion process. The model has 30.4 million parameters, which leads to an inference time of roughly 10 s per slice of the shape of 256 × 256 pixels. The model is trained with the Adam optimizer with a initial learning rate of 1×10^{-4} which is reduced exponentially at a

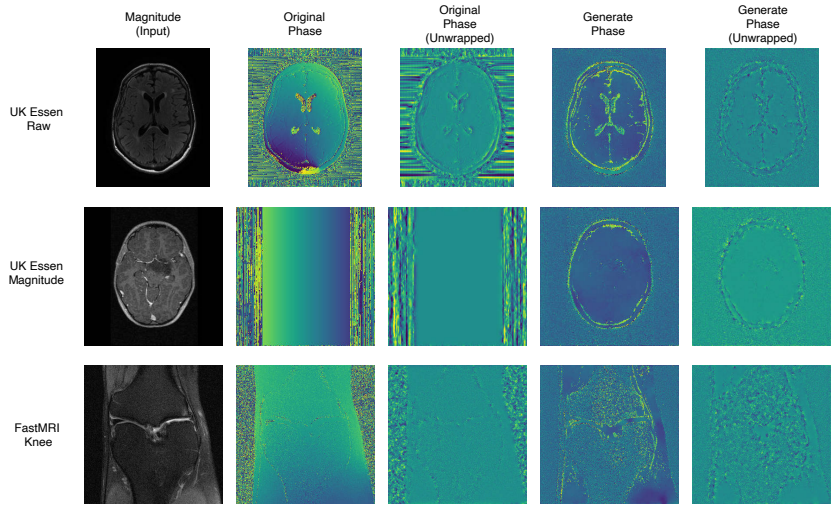

FIG. 5.3: Exemplary inference outputs of the proposed generative model PhaseGen. The first row shows exemplary data from the University Hospital Essen (UK Essen) dataset, the second row from the UK Essen Magnitude dataset which we use for the skullstripping task and the last row shows exemplary results from the FastMRI Knee dataset. For each example the original phase and the generated phase of the data is shown, as well as its unwrapped form. Figure published as is in [103].

gamma factor of 0.995 The beta and epsilon coefficient of Adam are 0.99 and 1×10^{-8} respectively. We train the model for 200 epochs with a batch size of 128. The model is trained on a single NVIDIA A100 GPU with 80GB of graphics memory. The training time is roughly 10 hours.

Exemplary inference outputs of PhaseGen can be seen in Fig. 5.3. The examples are from the University Hospital Essen (UK Essen) raw dataset, the UK Essen Magnitude dataset used for skullstripping and the FastMRI Knee dataset.

Shown are the original magnitude, the original phase (in case of the UK Essen Magnitude data there is no phase available), the generated phase as well as the unwrapping of the phase.

Examplary results of the naive phase generation method can be found in Appendix B.2.

Phase maps often show so-called *fringelines* and *cutlines*, which are visible as phase jumps in the unwrapped phase [134, 135]. These artifacts are caused due to the border at $-\pi$ and $+\pi$ or by phase rotations of more than π at two neighbouring pixels. These wrapped phase maps can make the interpretation of the phase difficult. In our case, these artifacts can lead to the perception of larger differences between the real and generated phase maps than there actually are. The removal of these artifacts is called *phase-unwrapping*. There are several different approaches to this problem [134, 136, 137], which we will not further discuss in this work. We decided to use the

Laplacian-based Phase Unwrapping in Python tool [138], which can be integrated seamlessly into our workflow. To use the unwrapping tool, we first need to convert the complex-valued pytorch tensors into complex-valued NIfTI files.

While the generated data does show some differences to the real phase maps, especially in regions of low brightness, the overall structure and the nature of phase maps, such as phase jumps, are preserved. It has to be noted, that the generation of phase maps is a ill defined problem, due to the lack of information available to the generative model, such as coil sensitivites and introduced (thermal) noise [139]. Most importantly, the generated phase is consistent with the magnitude.

5.3.4 *Downstream Task 1: Skullstripping*

In the first evaluation experiment we revisit the work presented in chapter 3. Skullstripping is the task of removing non-brain tissue from brain MRI data. In later experiments with the trained k-Strip architecture, we realized that the model trained only on magnitude data[7] is not able to generalize on real-phase data. While the initial k-Strip project was able to show the feasibility of performing segmentation tasks in the k-Space, this lack of generalization prevents a adoption in clinical routine. The goal of this experiment is to show that by training the model with generated raw MRI data from our proposed method, k-Strip is able to perform the segmentation task on real-phase data. This experiment is also testing k-Strips ability to perform zero-shot segmentation[8], as the real-phase data is not present in the training or validation dataset.

For each of the phase-generation[9] methods we conducted an extensive grid-search with the help of the *Weights & Biases* tool [140]. The grid-search included the following hyperparameters: Activation function, Dropout, Batch size, Learning rate, Optimizer, Number of Residual Blocks per Layer.

An overview of the grid search can be found in the appendix. The best performing model for each phase-generation method is then trained until the validation loss converges and does not decrease for 30 consecutive epochs.

[7] The data was transformed into the k-Space without an actual phase in the image domain.

[8] Zero-shot segmentation or classification refers to the ability of a model to perform a task on data from a distribution it has not seen during training. In this context, the model is evaluated on real-phase data after being trained only on synthetic-phase data.

[9] No-phase, Naive phase, PhaseGen

In the case of the model trained with the generated data by PhaseGen, we use the following hyperparameters:

- Activation function: PReLU
- Dropout: 0.2
- Batch size: 128
- Learning rate: 1e-4
- Optimizer: Adam
- Number of Residual Blocks per layer: 2

As a loss function we use the *L1*-loss, same as in the original k-Strip work. The model is trained on a single NVIDIA A100 GPU with 80GB of graphics memory.

The results can be seen in Fig. 5.4 as well as in Tab. 5.1. All results are reported as mean and standard deviation over the 14 real-phase data test subjects. The mean and standard deviation are calculated for each slice. For the evaluation we use DSC and HD as metrics.

The k-Strip model trained with the generated data by PhaseGen outperforms both of the other phase-generation methods. With a DSC of 80.1 % and a HD of 1.534 pixels the model is able to generalize

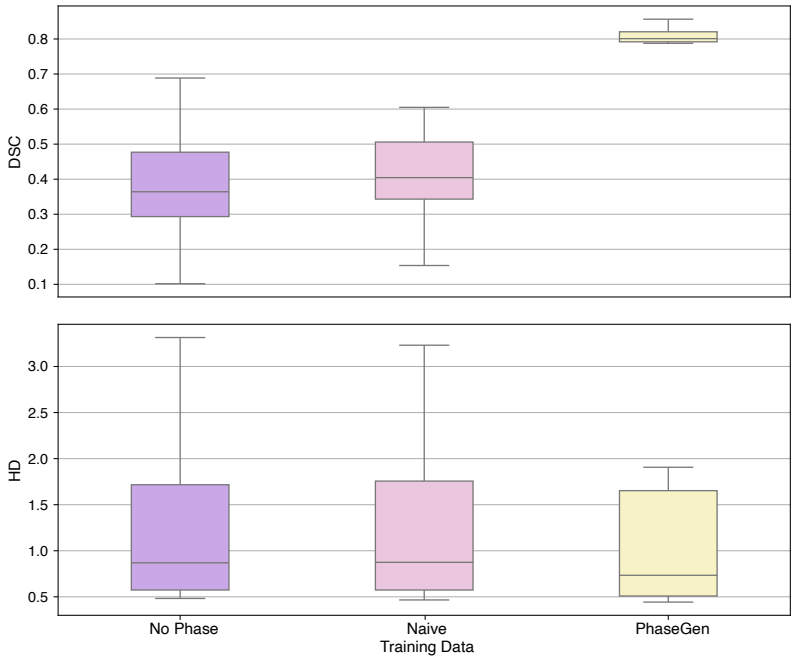

FIG. 5.4: Results on the skullstripping evaluation task test dataset consisting of 14 real-phase MRI subjects. Compared are the three training data generation methods: No Phase, Naive Phase, PhaseGen. Top: Dice Similarity Score, Bottom: Hausdorff Distance. Figure published as is in [103].

TABLE 5.1: Results of the skullstripping task. Compared are the three training data generation methods.

Training Data	DSC (%) ↑	HD (pixel) ↓
No Phase	40.1 ±17.2	1.577 ±1.799
Naive	41.1 ±13.5	1.634 ±1.946
Diffusion (Proposed)	**80.1 ±3.2**	**1.534 ±1.923**

on real-phase data. The model trained with the naive phase data shows a DSC of 41.1 % and a HD of 1.634 pixels. The model trained with no phase data shows a DSC of 40.1 % and a HD of 1.577 pixels. The no-phase and naive-phase model do not produce meaningful segmentation leading to these low DSC values.

Further finetuning with real-phase data could lead to even better results, but was not performed in this downstream-task due to the lack of available real-phase data. Nevertheless, this first evaluation task proves the feasibility of the proposed generative model PhaseGen. The generated data is able to improve the performance of the k-Strip model on real-phase data, while the other generative approaches do not lead to meaningful results.

5.3.5 *Downstream Task 2: MRI Reconstruction*

In the second evaluation experiment we evaluate the performance of a deep neural network trained with the different phase generated data on the task of MRI reconstruction. The task of MRI reconstruction is to fill missing k-Space lines which have not been acquired to speed up the acquisition times. This experiment allows us to evaluate again the performance of the generative model in generating realistic MRI raw data. Additionally we can now test the combination of real-phase data and generated phase data. We argue that the approach of combining available real-phase MRI raw data with bigger amount of generated MRI raw data will have a significant impact on the research community by allowing the training of deep neural networks on tasks which by now do not have enough available real-phase MRI data.

Algorithms such as GRAPPA have already been introduced in an earlier chapter. In recent years more MRI reconstruction approaches utilizing deep neural networks have been proposed [141, 142]. We use a complex-valued residual UNet structure similar to the k-Strip architecture to perform the reconstruction task. We extended k-Strip with data-consistency layers due to issues with lack of consistency in the final output. In preliminary experiments without these data-

consistency layers we observed that the model lost or altered the information of existing k-Space lines. By preserving these existing information, the model can learn more efficiently and we can guarantee that no already existing information is lost in the data [143, 144].

Our implementation of data-consistency can be formulated as following:

Let:

- x_{pred}: Predicted k-Space data
- x_{in}: Measured k-Space data
- M: Binary sampling mask, $M \in \{0,1\}^{N \times N}$
- ε: Small constant for numerical stability

Then the data consistency is applied as in Alg. 3.

Algorithm 3 Data Consistency Layer

Precondition: x_{pred} (predicted k-Space), x_{in} (measured k-Space)
▷ Get sampling mask
$M \leftarrow 1_{|x_{\text{in}}|>0}$
▷ Keep only sampled k-Space values
$x_{\text{in}} \leftarrow M \cdot x_{\text{in}}$
▷ Keep only predicted values for unsampled locations
$x_{\text{pred}} \leftarrow (1-M) \cdot x_{\text{pred}}$
$\text{scale\_factor} \leftarrow \dfrac{\mathbb{E}[|x_{\text{pred}}|] + \varepsilon}{\mathbb{E}[|x_{\text{in}}|] + \varepsilon}$
▷ Normalize measured values
$x_{\text{in}} \leftarrow \text{scale\_factor} \cdot x_{\text{in}}$
▷ Combine predictions and measurements
$x_{\text{out}} \leftarrow x_{\text{in}} + x_{\text{pred}}$
return x_{out}

The presented data-consistency algorithm calculates a sampling mask by creating a binary mask based on the magnitude values of the measured k-Space data with a threshold of zero. The mask is then used to remove all values from the predicted k-Space data which are already present in the measured k-Space data. The predicted k-Space data is then scaled based on the mean of the predicted and measured k-Space data. This scaling is done to ensure that the predicted k-Space data is in the same range as the measured k-Space data. This step also introduces a regularization into the model. The final output is then calculated by adding the predicted k-Space data to the measured k-Space data.

The data-consistency layer is applied after each residual block in the downsample path of our model. It is not used in the upsampling layer, as the skip connections already preserve the information of the input data.

We compare the performance of the model trained with the different phase generation methods as well as with the available real-phase data. We again conducted a grid search to find the best hyperparameters for each model. A full overview of the resulting hyperparameters can be found in Appendix B.6. The hyperparameters for the model trained with the generated data by PhaseGen are:

- Activation function: ReLU
- Dropout: 0.4
- Batch size: 128
- Learning rate: 1e-4
- Optimizer: AdamW [145]
- Noise Scheduler: Exponential

For each data type we train a small model with 209 000 parameters (200k model) and a bigger model with 3.3 million parameters (3M model). Following the implementation of the FastMRI challenge in [105], we apply Cartesian undersampling masks with a fully sampled center region of 8% for an undersampling factor of four and a fully sampled center region of 4% for an undersampling rate of eight.

The results of the different phase generation methods on the reconstruction task at an undersampling factor of four are shown in Fig. 5.5. A full overview of the results at undersampling factors of four and eight are shown in Tab. 5.2.

We show the results on the original data seperated from the generative approaches, as the original data is supposed to be the gold standard and baseline. A perfectly generated phase data would lead to the same results as the original data. A surpassing of the original data would indicate flaws in the experimental setup.

Again we can see that the model trained with data generated by PhaseGen surpasses both other generative methods, with both the 200k and the 3M model. At a undersampling rate of four, the 3M model trained with PhaseGen-data achieves a SSIM of 63.16 % and a PSNR of 23.95 dB, while the model trained with the naive phase data achieves a SSIM of 56.31 % and a PSNR of 21.81 dB. The model

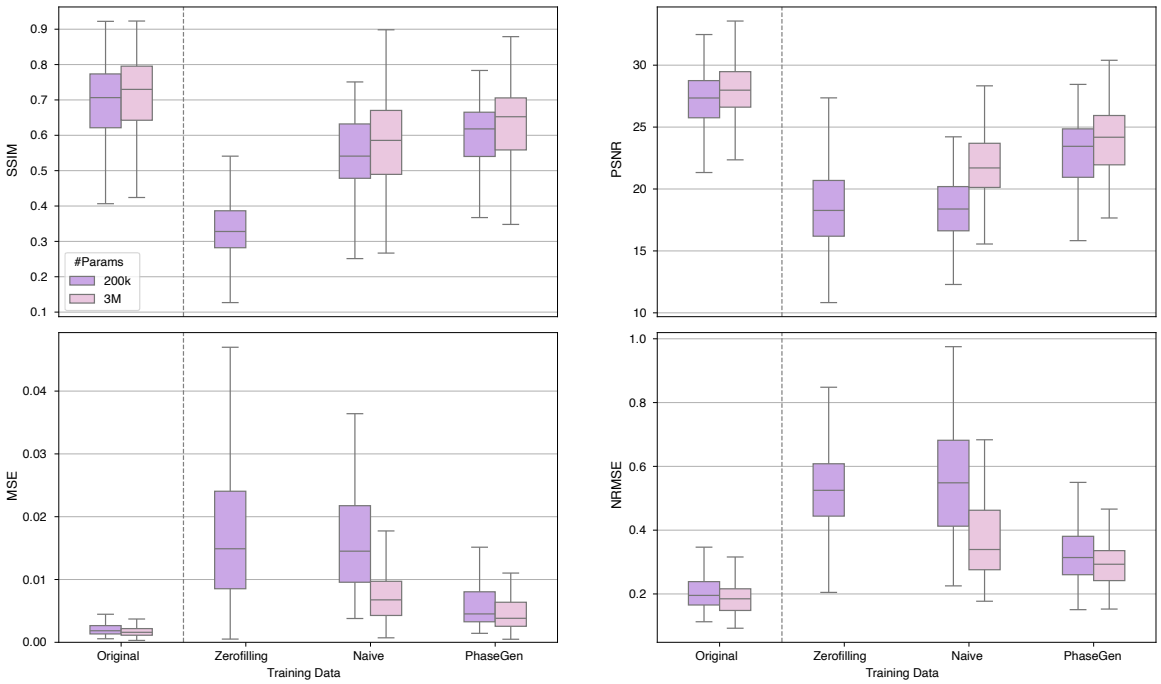

FIG. 5.5: Results of the second evaluation experiment, the MRI reconstruction task, at an undersampling factor of four. Compared are the 200k and 3M model trained with three different training data generation methods, as well as the original real-phase data (left). Shown are the metrics SSIM, PSNR, MSE, NRMSE. Figure adapted from [103].

trained with no phase data achieves a SSIM of 59.73 % and a PSNR of 22.76 dB. The smaller 200k model in comparison achieves a SSIM of 59.73 % and a PSNR of 22.76 dB, while the model trained with naive phase data achieves a SSIM of 54.32 % and a PSNR of 18.31 dB. The model trained with no phase data achieves a SSIM of 63.16 % and a PSNR of 23.95 dB.

The results at an undersampling factor of eight show the same trend. The model trained with the generated data by PhaseGen outperforms both other generative methods, with both the 200k and the 3M model. At a undersampling rate of eight, the 3M model trained with PhaseGen-data achieves a SSIM of 55.39 % and a PSNR of 21.22 dB, while the model trained with the naive phase data achieves a SSIM of 47.52 % and a PSNR of 20.32 dB. The model trained with no phase data achieves a SSIM of 54.57 % and a PSNR of 20.61 dB. The smaller 200k model in comparison achieves a SSIM of 54.57 % and a PSNR of 20.61 dB, while the model trained with naive phase data achieves a SSIM of 55.74 % and a PSNR of 20.41 dB. The model trained with no phase data achieves a SSIM of 55.39 % and a PSNR of 21.22 dB.

As expected, none of the generative methods reaches the performance of the model trained on real-phase data. Overall this experiment shows that the proposed generative model PhaseGen is able to

TABLE 5.2: Results of the MRI reconstruction task at an undersampling factor of four and eight. Compared are the 200k and 3M model trained with three different training data generation methods, as well as the original real-phase data (italic). Evaluated are the four metrics SSIM, PSNR, MSE and NRMSE.

Training Data	SSIM (%) ↑	PSNR (dB) ↑	MSE ↓	NRMSE ↓
Undersampling x4				
Original [200k]	*69.10 ±11.44*	*27.19 ±2.22*	*0.002 ±0.001*	*0.207 ±0.061*
Original [3M]	*71.28 ±11.76*	*27.98 ±2.58*	*0.002 ±0.001*	*0.190 ±0.066*
Zerofilling	33.88 ±7.89	18.62 ±3.32	0.017 ±0.011	0.553 ±0.153
Naive [200k]	54.32 ±10.64	18.31 ±2.53	0.018 ±0.011	0.606 ±0.289
Naive [3M]	56.31 ±14.44	21.81 ±3.01	0.008 ±0.007	0.403 ±0.192
PhaseGen [200k]	**59.73 ±9.93**	**22.76 ±2.94**	**0.007 ±0.007**	**0.351 ±0.134**
PhaseGen [3M]	**63.16 ±10.87**	**23.95 ±2.91**	**0.005 ±0.004**	**0.301 ±0.098**
Undersampling x8				
Original [200k]	*64.74 ±11.38*	*25.58 ±2.46*	*0.003 ±0.002*	*0.247 ±0.069*
Original [3M]	*67.01 ±11.97*	*26.26 ±2.61*	*0.003 ±0.002*	*0.231 ±0.071*
Zerofilling	31.15 ±8.47	17.12 ±3.58	0.025 ±0.016	0.673 ±0.248
Naive [200k]	55.74 ±11.11	20.41 ±3.26	0.012 ±0.008	0.465 ±0.199
Naive [3M]	47.52 ±13.41	20.32 ±2.57	0.011 ±0.007	0.454 ±0.139
PhaseGen [200k]	**54.57 ±11.53**	**20.61 ±3.28**	**0.011 ±0.009**	**0.434 ±0.096**
PhaseGen [3M]	**55.39 ±12.11**	**21.22 ±3.22**	**0.010 ±0.010**	**0.405 ±0.096**

generate more realistic MRI raw data than the compared generative methods.

In many cases, we do have some real-phase data[10], but not enough to train a deep neural network. The combination of the available real-phase data with a larger amount of generated data could lead to better results than training with only the real-phase data. To test this hypothesis, we now combine a varying amount of real-phase data with the generated data by PhaseGen. For this we remove different amounts of the real-phase data from the training and validation set and replace it with generated data. In the end, the magnitudes in the different datasets will stay the same, but the ratio between real-phase and generated data will vary.

[10] or perhaps are able to acquire a small amount of raw data in a clinical setting

We test the following ratios of real-phase data to generated data: 0.1%, 1%, 5%, 10%, 15%, 20%, 100%. We again train both the 200k and the 3M model with these different ratios. The results for the undersampling rate of four can be seen in Fig. 5.6 and in Tab. 5.3. All models are trained with the same hyperparameters as in the previous experiment. The results for the undersampling rate of eight can be found in the appendix.

The results show that the model trained with only 10%-15% of the real-phase data in combination with synthetic raw MRI data generated with PhaseGen is able to reach comparable results to the model trained with 100% real-phase data. The 3M model trained with 10% of the

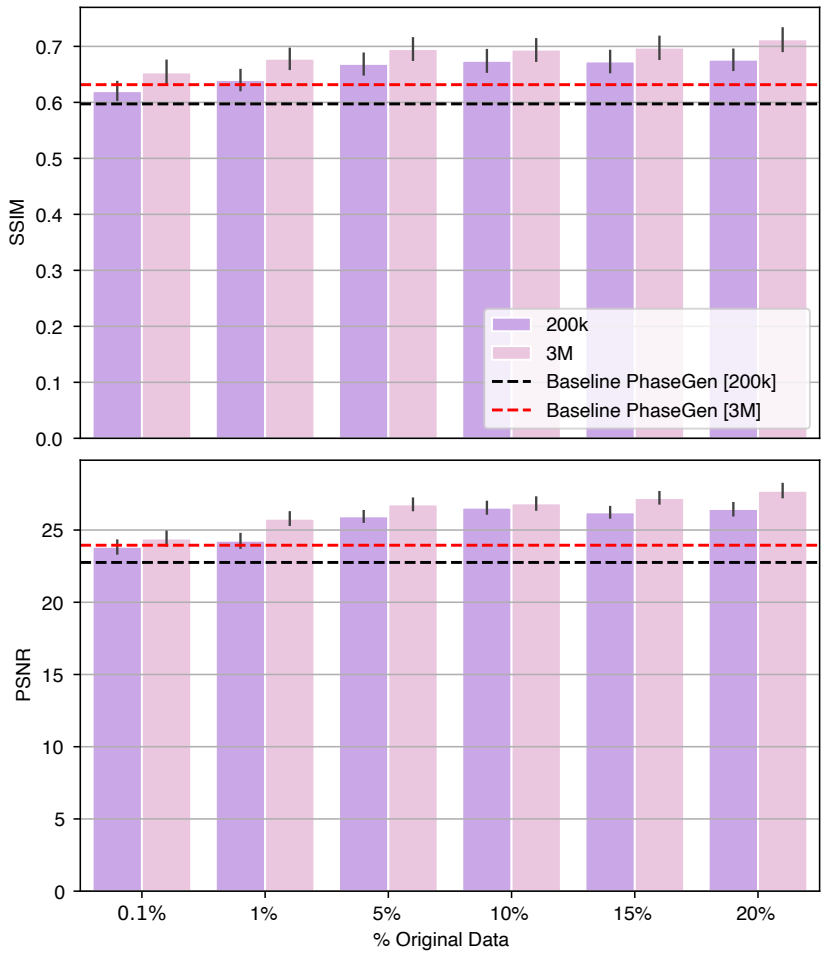

FIG. 5.6: Results of the MRI reconstruction task with different ratios of real-phase data to synthetic-phase data by PhaseGen. Shown are the results for the following amounts of real-phase data: 0.1%, 1%, 5%, 10%, 15%, 20%. The experiment is conducted for the 200k and the 3M model at an undersampling rate of four. Figure published as is in [103].

real-phase data achieves a SSIM of 69.65 % and a PSNR of 26.90 dB, while the model trained with 100% real-phase data achieves a SSIM of 71.28 % and a PSNR of 27.98 dB. The 200k model trained with 10% of the real-phase data achieves a SSIM of 67.42 % and a PSNR of 26.55 dB, while the model trained with 100% real-phase data achieves a SSIM of 69.10 % and a PSNR of 27.19 dB. Even when using only 1% of the real-phase data, the 3M model is able to achieve a SSIM of 67.80 % and a PSNR of 25.78 dB, which is an increase by 4.7 percentage points in SSIM and 1.83 dB in PSNR compared to the model trained with only the generated data.

The question arises if the improved performance is due to the combination of the real-phase data with the generated data or if the model is only learning from the small amount of available real-phase data. To investigate this, we train the same model with the same amount of real-phase data, but without the generated data.

TABLE 5.3: Results of the MRI reconstruction task with different ratios of real-phase data to synthetic-phase data by PhaseGen. Shown are the results for the following amounts of real-phase data: 0.1%, 1%, 5%, 10%, 15%, 20%. The experiment is conducted for the 200k and the 3M model at an undersampling rate of four.

% of real world data	SSIM (%) ↑	PSNR (dB) ↑	MSE ↓	NRMSE ↓
200k model				
0.1%	62.00 ±9.92	23.84 ±2.76	0.005 ±0.004	0.310 ±0.131
1%	63.98 ±10.37	24.26 ±27.15	0.005 ±0.004	0.301 ±0.143
5%	66.87 ±10.90	25.95 ±2.21	0.003 ±0.002	0.240 ±0.083
10%	67.42 ±11.36	26.55 ±2.47	0.003 ±0.002	0.223±0.071
15%	67.31 ±11.07	26.22 ±2.13	0.003 ±0.001	0.231 ±0.073
20%	67.62 ±11.12	26.46 ±2.61	0.003 ±0.002	0.223 ±0.058
100%	69.10 ±11.44	27.19 ±2.22	0.002 ±0.001	0.207 ±0.061
3M model				
0.1%	65.34 ±11.76	24.42 ±3.02	0.005 ±0.004	0.295 ±0.135
1%	67.80 ±10.78	25.78 ±2.67	0.003 ±0.003	0.249 ±0.100
5%	69.52 ±11.74	26.77 ±2.55	0.003 ±0.002	0.219 ±0.007
10%	69.65 ±11.66	26.90 ±2.44	0.002 ±0.002	0.218 ±0.082
15%	69.78 ±11.67	27.22 ±2.43	0.002 ±0.002	0.207 ±0.070
20%	71.26 ±12.06	27.72 ±2.65	0.002 ±0.001	0.198 ±0.071
100%	71.28 ±11.76	27.98 ±2.58	0.002 ±0.001	0.190 ±0.066

With 10% of real-phase data and no additional synthetic data the 200k model achieves a SSIM of 65.73 %±10.13 % and a PSNR of 25.86 db±2.56 dB. In comparison, the 200k model with additional synthetic raw data reaches a SSIM of 67.42 %±11.36 % and a PSNR of 26.55 dB±2.47 dB.

To summarize the second evaluation experiment, we can say that the proposed generative model PhaseGen is able to generate realistic MRI raw data which can be used to train deep neural networks on tasks which do not have enough available real-phase data. The combination of real-phase data with generated data leads to better results than training with only the real-phase data. The results show that the model trained with only 10%-15% of the real-phase data in combination with synthetic raw MRI data generated with PhaseGen is able to reach comparable results to the model trained with 100% real-phase data.

5.4 DISCUSSION & CONCLUSION

In this work, we developed PhaseGen, a novel complex-valued, diffusion-based generative model for synthesizing raw MRI data. The model is conditioned on a magnitude MRI image to generate a corresponding and realistic phase component. We validated the generated raw MRI data with two different downstream tasks, skullstripping directly in the k-Space and MRI reconstruction. In both experiments the models trained with the synthetic raw MRI data generated by PhaseGen managed to reach significant performance improvements compared to the

compared methods, demonstrating the effectiveness of our approach in enhancing MRI data processing capabilities. In case of the skull-stripping task, the model trained with the generated data was able to generalize on real-phase data, while the other generative approaches did not lead to meaningful results. In the MRI reconstruction task, the model trained with the generated data was able to reach comparable results to the model trained with 100% real-phase data, while only using 10%-15% of the real-phase data in combination with a larger amount of generated data.

In future works the generative model needs to be extended to work on multicoil data. Currently the only way to generate multicoil data is to generate each coil separately. This approach has the drawback of ignoring the coherence between the different coils. Additionally, the inference time of roughly 10 s per slice with a shape of 256×256 pixels can be limiting when working with larger datasets. In a current project we retrained our PhaseGen with only 100 diffusion steps instead of 1000 diffusion steps, reducing the inference time to 1 second per slice, which significantly improves the usability of the model in practical applications. The impact on the quality of the generated data is still under investigation.

To summarize, this work presents a tool to utilize the limited amount of real-phase MRI raw data currently available to expand the research and usability of deep neural networks for the application to k-Space data. We are now able to pretrain models on synthetic data which is close to the target domain of MRI raw data. The limited amount of real-phase data can then be used to finetune the model on the target domain.

In the following chapter we will present the last project of this work which is a tool used to anonymize medical image data, such as Computer Tomography (CT), histological and MRI data, including MRI raw data. To realize the anonymization of MRI raw data we combined the power of the generative PhaseGen model and the k-Strip work for skullstripping.

KEY CONTRIBUTIONS

- **A Generative Model for Synthetic Raw MRI Data:** Developed PhaseGen, the first complex-valued diffusion-based model capable of generating realistic, complex-valued raw MRI data by synthesizing a phase component conditioned on a magnitude-only image.
- **Overcoming the Raw Data Bottleneck:** Proved that PhaseGen can effectively turn the vast public repositories of magnitude-only images into a source of training data for MRI raw data-based deep learning, unlocking new research possibilities.
- **Demonstrated Utility for Downstream Task Performance:** Validated the quality of the synthetic data by showing its benefits in two distinct tasks: It enabled the k-Strip segmentation model to generalize to real-world raw data, a task it previously failed. It allowed an MRI reconstruction model to achieve performance comparable to full real-data training using only 10-15% real data, drastically reducing the need for scarce raw datasets.

6

A Comprehensive Tool for Medical De-identification including Raw MRI Data

Question 5. How can we de-identify MRI raw data?

Before working with data gathered in clinical routine, patient related information have to be de-identified to protect the privacy of the patients. This is a legal requirement in many countries and is also a good practice to follow in any research involving human subjects. The de-identification process involves removing or modifying personal identifiers from the data, such as names, addresses, dates of birth, and other information that could be used to identify individuals. In case of imaging data, this also involves the removal of image features which can be used to identify individuals. This is most obvious in case of imaging data in the region of the head. While the k-Space is not interpretable by humans, the image information which can be used to identify the subject is just an inverse fourier transformation away. Thus, the de-identification of k-Space data is a non-trivial task which is crucial when working with or sharing MRI raw data. We will use the term *De-identification* in this chapter, as it is a more general term than *Anonymization*. De-identification aims to remove or modify only direct indentifiers, while anonymization aims to remove all information which could be used to identify individuals. Another term which is often used in this context is Pseudonymization, which refers to te process of replacing personal identifiers with pseudonyms, which are not directly identifiable but can be linked back to the original

data if necessary.

In this chapter we will introduce a deep learning tool which is able to de-identify a wide range of medical imaging data, including MRI raw data. An overview of the tool is shown in Fig. 6.1. We will focus on the de-identification of MRI raw data, as it is essential for ensuring patient privacy and compliance with legal standards when it comes to the data concerning this thesis.

An overview of the proposed tool is shown in Fig. 6.1.

The importance of this project in the overall picture of this thesis lies in the de-identification of the MRI raw data. For the previous project *PhaseGen* we utilized MRI raw data gathered in clinical routine. This data first had to be de-identified. Due to the lack of publicly available tools for the de-identification of MRI raw data, we had to implement our own tool. This will become especially important as soon as we want to share our data with other research groups or the public.

The following chapter will focus on the parts of the tool most relevant in the picture of this thesis. The full implementation details and evaluation results can be found in the corresponding publication [146].

The results of this project have been published in the paper "De-Identification of Medical Imaging Data: A Comprehensive Tool for Ensuring Patient Privacy" [146]. A detailed transparency statement can be found in Appendix A.4. The tool is publicly available on GitHub[1].

[1] https://github.com/TIO-IKIM/medical_image_deidentification

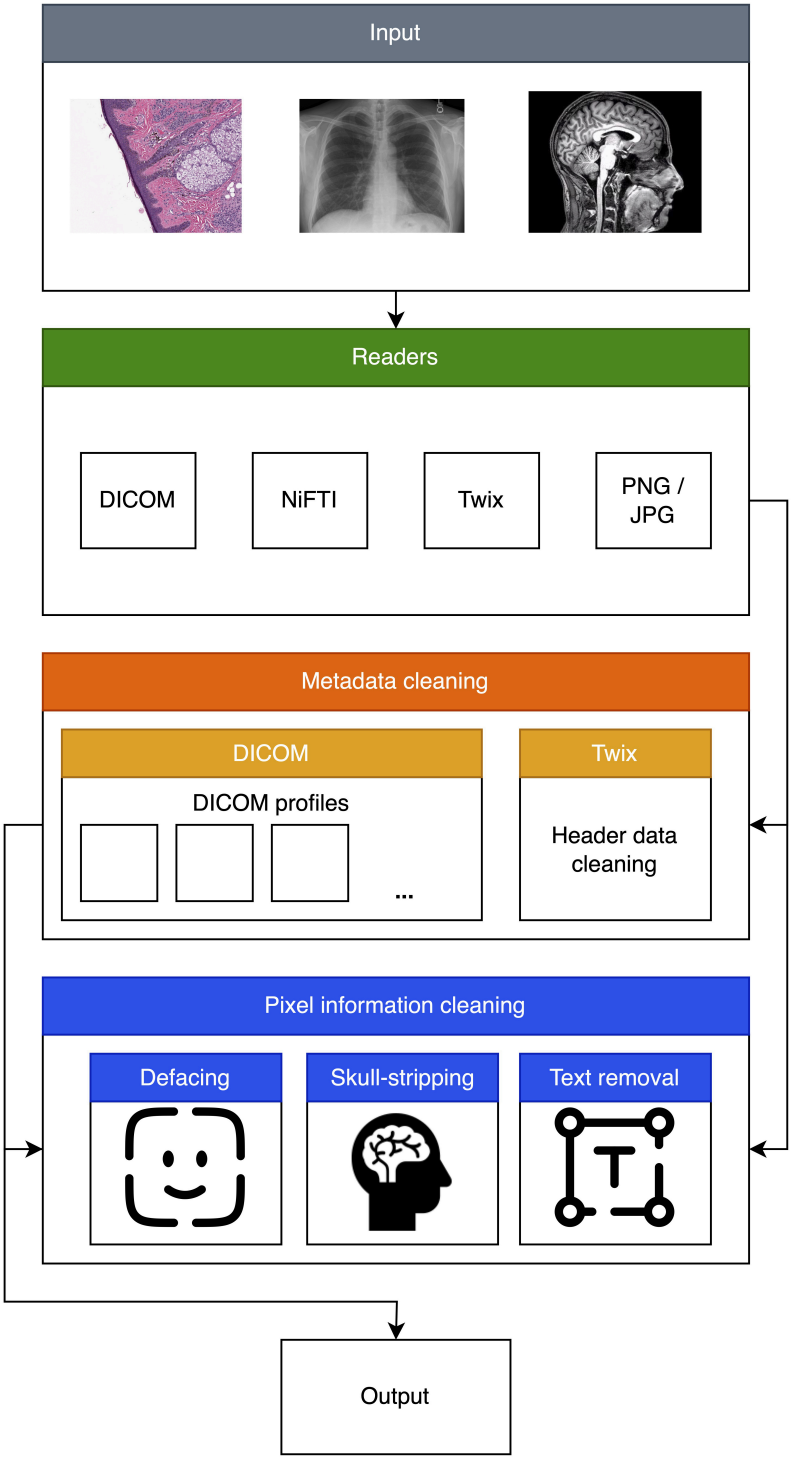

FIG. 6.1: Overview of the de-identification tool. The tool is able to de-identify a wide range of medical imaging data, including MRI raw data. Figure published as is in [146].

6.1 INTRODUCTION

Working with medical imaging data requires a high level of data protection and privacy. Information such as patient name, age, gender and even image features can be used to identify individuals [147]. The latter is especially true for imaging data in the region of the head, where facial features can be used to identify individuals [148].

With the wide range of imaging modalities and different data formats, the de-identification of medical imaging data is a non-trivial task. The most common data formats are DICOM, NIfTI, as well as JP(E)G or PNG, each of which has its own set of metadata and image features. Today there are a number of tools available for anonymizing medical imaging data, each tailored for a specific data format or imaging modality. For example, the tool *PyDicom* is a python library which allows the de-identification of DICOM metadata [149].

While the de-identification of metadata is a standardized process, the of image features is still an open research problem. In case of facial features, two approaches are common: the first approach is to remove the skull and facial features from the image, a process known as skull stripping. There is a variety of tools available for this task, such as *SynthStrip* [59] or *HD-BET* [60]. The second approach is called defacing. The idea is to only remove the facial features from the image, while keeping the skull intact. This has the benefit that the image has a higher interpretability, but can result in a lower de-identification quality. Available tools for this task include *mrideface* [150] or *pydeface* [151]. Due to a lack of GPU acceleration, some of these tools are not suitable for large datasets or high-resolution images. Additionally, most of these tools are only able to de-identify a specific data format or imaging modality. This is a major drawback, as many researchers work with a wide range of imaging data and need a tool which is able to de-identify all of them.

When it comes to the de-identification of MRI raw data, there are no tools publicly available which can de-identify the k-Space data.

6.2 PROPOSED DE-IDENTIFICATION TOOL

To address the need for a unified de-identification solution, we developed a comprehensive, open-source tool that combines multiple deep learning models and data handling techniques. For the de-identification of DICOM and NIfTI metadata, we implement the DI-

COM guidelines for metadata-handling [152], as well as a conversion tool for NIfTI metadata into DICOM metadata.

The de-identification of image features is done by training two deep learning models for skull stripping and defacing respectively on a wide range of imaging data, including MRI and CT data. As a underlying model, we use the pretrained 3D MedNeXt architecture [153].

We additionally provide a tool for the removal of text in ultrasound images, which is a common problem in ultrasound imaging. We utilize *tesseract* [154] in an iterative fashion for this task.

The de-identification task most relevant for this thesis is the de-identification of MRI raw data. This task includes two unique challenges: first, the de-identification of the metadata, which is not defined by the DICOM standard, and second, the de-identification of the k-Space image data while preserving the complex valued nature of the data. In the following, we will go into detail on how we implemented both of these de-identification tasks for the case of Siemens *Twix* raw data format.

The software is designed with a modular architecture, allowing users to apply a complete de-identification pipeline or select specific modules as needed. The core workflow, as illustrated in Fig. 6.1, proceeds through the following stages:

1. Input and Reader Selection: The tool accepts a variety of data formats, including DICOM, NIfTI, PNG/JPG, and raw Siemens *Twix* files. It automatically selects the appropriate reader for the given input data.

2. Metadata Cleaning: The first processing step is the sanitization of all metadata. This is a crucial step for all data types but is particularly complex for raw MRI data, as detailed in 6.3.1.

3. Pixel Information Cleaning: The tool then removes identifiable features from the image data itself. Users can choose between three primary modes:

    a) Skull-stripping: Removes the entire skull and facial features. For raw MRI data, this is performed directly in k-Space using the k-Strip architecture described in chapter 3 of this thesis.

b) Defacing: Removes only the facial features while preserving the skull, which can be useful for certain clinical applications [155].

c) Text Removal: Specifically designed for ultrasound images, this mode identifies and removes burned-in text annotations.

4. Output: The tool saves the fully de-identified data in the same format as the input, with cleaned metadata and pixel/k-Space data.

6.3 MRI RAW DATA DE-IDENTIFICATION

6.3.1 *Metadata de-identification*

The metadata of Siemens *Twix* raw data does not follow the DICOM standard, as it contain much more information and is not structured in a way which is compatible with the DICOM standard. However, the *Twix* metadata does also contain a DICOM header[2], which can be anonymized following the DICOM guidelines. When only anonymizing the DICOM header, the underlying *Twix* metadata[3] will update the DICOM header as soon as the data is loaded again. This second metadata-header, as its label already imply, is in a string format. We perform a regex search [156] for all relevant identifiers[4], including patient name, date of birth or even the MRI device serial number. According to the type of metadata identifier, we replace the information with a random string or number of the same length. Fig. 6.2 shows an excerpt of an exemplary anonymized *Twix* metadata header. We can see that the patient name is replaced by a string of 'x', while identifiers such as data of birth or gender are replaced by the number '0.0'.

The anonymized metadata can optionally be saved separately from the anonymized raw data to allow for a more flexible data handling.

Metadata we currently keep intact are MRI field strength, sequence and other acquisition parameters, as these information are crucial for the correct interpretation of the MRI data. Additionally, parameters such as undersampling and oversampling rates are kept intact, as they are important for the correct reconstruction of the k-Space data.

[2] The DICOM header inside the *Twix*-file is labeled as "hdr".

[3] This second and much bigger metadata is labeled as "hdr_string".

[4] The loading of the data is adapted from the implementation by the authors of https://github.com/openmrslab/suspect and will not be discussed further as it is beyond the scope of this chapter.

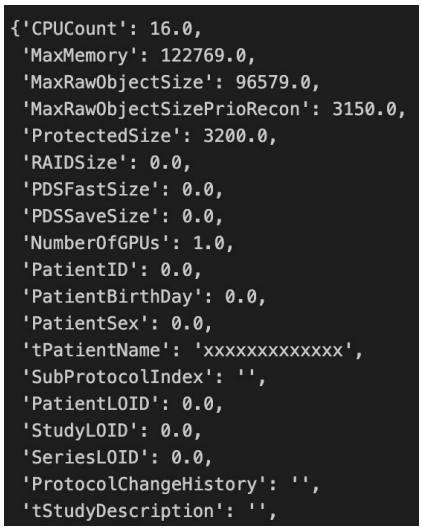

FIG. 6.2: Excerpt of an anonymized *Twix* metadata header. The following identifiers are anonymized: PatientID, PatientBirthday, PatientSex, tPatientName, PatientLOID, StudyLOID, SeriesLOID. The de-identification is either done by replacing the identifier with a string of 'x' or the number '0.0'. Figure adapted from [146].

6.3.2 *Skullstripping*

The second challenge of anonymizing the image data by performing skullstripping directly on the k-Space data is solved by implementing the *k-Strip* algorithm presented in chapter 3. To train this model we utilize our generative model *PhaseGen* presented in chapter 5 to turn two publicly available datasets of image domain, magnitude-only MRI datasets into synthetic MRI raw data. These two datasets are the Neurofeedback Skull-stripped (NFBS) repository [74] and the Calgary-Campinas-359 (CC-359) dataset [75], with a total of 479 MRI volumes, including segmentation masks. Each 2D slice is processed by our generative model *PhaseGen* and transformed into k-Space to create 479 synthetic MRI raw data volumes. Our k-Strip model is then trained on these synthetic MRI raw data volumes and the corresponding segmentation masks. The performance of said model was already evaluated in chapter 3 and chapter 5 and thus is not discussed further here.

The resulting model is implemented in the de-identification tool and can be used via a simple command line interface (cli)-tool.

6.4 DISCUSSION & CONCLUSION

In this chapter we presented a deep learning based tool for the de-identification of a variety of medical imaging data. The tool is able to de-identify medical images via metadata de-identification, skullstripping or defacing for the de-identification of facial features, as well as

text removal in ultrasound images. Additionally the tool allows for the de-identification of MRI raw data in the *Twix* format. We implement this feature by utilizing the works from the earlier chapters of this thesis, including *k-Strip* and *PhaseGen*. The current state of the tool when it comes to the de-identification of MRI raw data is currently limited to Siemens *Twix* data. Future work will include the implementation for other raw data formats. Additionally, currently we can only perform skullstripping on the k-Space data. The tool will be extended to also include a defacing algorithm for the de-identification of facial features directly in the k-Space. For achieving this goal, more labeled data needs to be acquired.

KEY CONTRIBUTIONS

- **A Comprehensive, Publicly Released De-Identification Tool:** Developed and released a robust, open-source software tool that provides a unified solution for anonymizing a wide range of medical imaging data, including DICOM, NIfTI, and raw MRI data.
- **Novel End-to-End Workflow for Raw MRI Anonymization:** Created a workflow that handles both the complex, non-standard metadata of raw data files and the removal of identifying features directly in k-Space by utilizing k-Strip.
- **Synergistic Application of Generative and Segmentation Models:** Demonstrated the thesis's project cycle by using the PhaseGen-trained k-Strip model as the model for k-Space anonymization. This showcases a tangible, real-world application built directly upon the generative and segmentation frameworks developed earlier.

7

Thesis Discussion, Conclusion & Outlook

7.1 DISCUSSION & FUTURE WORK

MRI is a widely used imaging modality in clinical practice. High waiting times for patients are common due to the technique's long acquisition times and high costs. Nevertheless, the amount of data generated during an MRI acquisition is underutilized. The proposed approach in this thesis aims to address this issue by utilizing all available data, including the k-Space data, to improve the efficiency and accuracy of MRI-based downstream tasks. This has the potential to significantly improve patient outcomes and reduce healthcare costs. The research presented in this thesis is not merely a collection of algorithmic improvements but an argument to fundamentally rethink how we approach MRI data processing. With multiple projects we tried to confirm the hypothesis that the current "reconstruct-then-analyze" paradigm is not the most efficient way to utilize the available data.

This thesis documents a systematic effort to challenge the conventional MRI pipeline. The core argument of ours is that by discrading raw k-Space data, the medical imaging community is missing out on valuable diagnostic information. Each project in this thesis is building upon the last, creating a cohesive narrative, from initial feasibility to direct clinical application.

We started with a feasibility study of the k-Space data for the task of skull-stripping in chapter 3, a preprocessing step that is crucial for many downstream tasks. We showed that the k-Space data can be preserved in this task by using a complex-valued neural network. The

presented algorithm *k-Strip* is able to perform the segmentation task directly in the k-Space domain, which allows for a more efficient processing of the data while ensuring patient privacy. The results of this feasibility study were comparable to state-of-the-art image-domain methods, demonstrating the potential of direct k-Space processing for downstream tasks beyond conventional MRI reconstruction. However, the k-Strip model showed a significant performance drop when tested on real-world data, which was not used for initial training. While this problem was mitigated by the development of PhaseGen, this issue highlights the need for more publicly available k-Space data for training and evaluating k-Space based algorithms.

Moving from feasibility to practical advantage, we presented a method to utilize the k-Space data acquired during DWI acquisition for prostate cancer likelihood estimation in chapter 4. By utilizing the available k-Space data, we were able to improve the performance of the predictive model, especially at high undersampling rates. The proposed pipeline allows for fast and efficient prostate cancer likelihood estimation, which can be used as a first step in the diagnostic process, potentially leading to a higher throughput of patients.

However, the first two projects revealed a critical bottleneck in the scalability of any direct k-Space paradigm: the scarcity of publicly available MRI raw data. This limitation led to the development of a novel complex-valued diffusion model in chapter 5, called *PhaseGen*, that is able to generate realistic phase information corresponding to input magnitude images. This generative AI model is able to harvest the potential of millions of publicly available MRI images to generate large pre-training datasets for k-Space based algorithms. We showed that the generated raw MRI data can be used to train k-Space based models. We tested the generated data on the k-Strip algorithm and on a MRI reconstruction task. The k-Strip model was able to generalize on real-world data with only being trained on synthetic k-Space data. The reconstruction model only needed 15% of real-world data in combination with the generated data to achieve the same performance as a model trained on 100% real-world data. This shows the potential of the generative AI model to create large pre-training datasets for k-Space based algorithms.

The last part of the thesis presented a method to anonymize k-Space data. The task of de-identification of MRI raw data is a non-trivial task, as no standardized method exists for this datatype. For this task we combine works earlier presented in this thesis. We use the gen-

erative diffusion model *PhaseGen* to generate synthetic MRI raw data to train the skull-stripping algorithm *k-Strip*. The de-identification pipeline includes the removal of patient-related metadata, as well as the removal of sensitive information from the k-Space data by performing skullstripping in the k-Space domain. This de-identification pipeline is able to preserve the k-Space data which can then be used for further downstream tasks.

The technical core of this thesis resides in the usage of complex-valued neural networks. This choice is based on the physical nature of the k-Space data, which is inherently complex-valued. Real-valued neural networks, which are commonly used in the field of medical imaging, are not fully suited for processing this complex-valued data, as they have to learn the intrinsic relationship between magnitude and phase from scratch. Complex-valued neural networks in contrast can natively represent the rotational and scaling properties of signals in the frequency domain.

In this thesis we laid the groundwork for the utilization of k-Space data for various downstream tasks. By developing different architectures and methods which can harvest the potential of complex-valued data, we provide a foundation for future research in this area. This thesis does not only present a novelty in the field of MRI, but also provides novel methods in the field of complex-valued deep learning. For example, the presented *PhaseGen* model is the first diffusion-based model which is able to generate complex-valued data by using a combination of a complex-valued UNet and complex-valued noise insertion.

Because this thesis covered a wide range of topics of a highly under-researched field, there are many open questions and directions for future research. In the following we will discuss some of them.

7.1.1 *Standardized k-Space Export*

The data used in this work was either publicly available or acquired on two different MRI scanners used in the clinical routine. The data acquired consists of different sequences, different field strengths and different anatomies. The data was exported manually onto a local storage unit, followed by a anonymization via the presented anonymization tool. Future work should focus on the development of a standardized export pipeline for k-Space data. As soon as data is acquired, it should automatically be de-identified and exported to a local storage unit. This would allow for a more efficient processing of the data, saving

time and resources.

At the time of writing this thesis, there is a lack of large annotated datasets for k-Space data. That is why we are constantly gathering raw data from clinical routine. The goal is to create a large dataset of k-Space data, which can be used for training and evaluating k-Space based algorithms on a variety of downstream tasks. Details about the current state of this dataset can be found in the appendix, including the current size, number of different sequences and anatomy distributions. The University Hospital Essen supports the *FHIR* standard [157] for the exchange of medical data. This standard allows us to gather further medical data connected to the MRI data, such as clinical reports or other imaging modalities saved in the clinical database. To gather this data, we are currently working on a pipeline to automatically export condition diagnosed shortly after the MRI acquisition. This data is saved in the *International Statistical Classification of Diseases and Related Health Problems* (ICD) format [158]. To extract that data we utilize the python-based *sFHAN*-toolkit, developed by Rostalski et al. at the Institute for AI in medicine in Essen. Future work will focus on the refinement of the dataset. This includes consultations with radiologists to verify the exported conditions. Additionally, we will gather additional data, such as laboratory data or full clinical reports.

7.1.2 *k-Space Preprocessing*

Another open question is the most suitable preprocessing of the k-Space data. While there are common preprocessing steps for image domain data, such as histogram normalization, standardization and augmentations, there is no standardized preprocessing for k-Space data. In the works presented in this thesis, we experimented with different preprocessing steps, such as complex-valued standardization and undersampling augmentations. However, extensive experiments are needed to find the most suitable preprocessing steps for k-Space data. One possible preprocessing step might be a weighting of the k-Space data, such that the periphery of the k-Space data, which is magnitudes smaller than the center, has higher influence on the training.

7.1.3 *Novel complex-valued Architectures*

The presented architectures in this thesis are all based on convolutional neural networks. The concept of convolutional neural networks is well established in the field of image processing due to their ability

to learn spatial hierachies of features. While the results presented in this thesis achieved with said architecture are promising, we hypothesize that other architectures, such as transformers, might be more suitable for working in the k-Space domain. Due to the non-local nature of the k-Space data, the local convolutional operations might not be well fitted for this task. Transformers in comparison are able to learn long-range dependecies due to their patch-based approach in the image domain. This global approach sounds more promising for the non-local k-Space data. In a future project, we will explore the use of radial transformer patches, creating a physical-informed vision transformer, which utilizes the nature of the k-Space data. The transformer will, in comparison to the equally distributed squared patches of traditional vision transformers, use radial patches which are centered around the k-Space center.

Another possible architecture could be convolutional neural networks with *Fourier Convolutional* layers [159, 160]. These layers, which are based on the convolution theorem explained earlier, can replace the standard convolutional layers. Earlier works showed that these layers become more efficient with increasing kernel size. Applied in the k-Space domain, these layers could be used to perform the convolutional operations.

The search for novel activation functions is another open question. In this work we adapted existing activation functions tailored for real-valued neural networks to the complex domain. There have already been some works on special complex-valued activation functions, such as the *complex cardioid* activation function [161] or the *modReLU* activation function [63]. However, these activation functions are not optimized and extensively tested for the k-Space domain. With activation functions tailored for the non-local nature of the k-Space and its differences between center and periphery, our models might be able to learn more complex features.

7.1.4 *Adaptive Sampling*

Adaptive Sampling is a technique to gather more data in regions of interest, while reducing the amount of data gathered in less important regions. Previous works have shown that adaptive sampling can be used to improve the performance of MRI reconstruction algorithms [162, 163] and even in downstream tasks such as pathology detection [13, 15]. Building on these works, future research should focus on the utilization of adaptive sampling in the k-Space domain.

We can combine the architectures presented in this thesis with the reinforcement-based algorithms presented in the works mentioned in [13, 15, 162, 163]. These adaptive approaches can also be used to better understand the k-Space data and its features used by the neural networks. Even when fully sampled k-Space data is available, retrospective adaptive sampling methods can be applied to highlight regions of interest.

7.1.5 *k-Radiomics: Radiogenomics rethought*

[1]Funded by the Bruno & Helene Jöster Foundation.

This thesis is part of the k-Radiomics project[1], which has the overarching goal of utilizing k-Space data in combination with novel radiomic features to improve the diagnostic process. This thesis lays the groundwork for this project by showing the feasibility of using k-Space data for various downstream tasks, as well as providing the necessary tools and architectures to do so. The next step in this project is now to combine the k-Space data with novel radiomic and genomic features. For this we will utilize the currently gathered clinical data. With the help of the Smart Hospital Information Platform (SHIP) infrastructure, we will be able to extract all necessary information from the clinical database. This includes the extraction of genomic data, such as mutations or other genetic features. There have been first works in the k-Radiomics project which explored novel radiomic features in the k-Space domain, such as peak power spectrum or spectral moments [164]. Further work is needed to explore the potential of these features in combination with the predictive models presented in this thesis. With the power of deep learning, we will be able to learn complex features from the k-Space data, which might hold more information than commonly used radiomic features.

But for an acceptance in the clinical routine, another important step is the explainability of these features and models. Currently we are working with so-called *black-box* models, which are not well interpretable. This is a major issue in the field of medical imaging, as the models need to be interpretable for the radiologists. There is the need to understand what the model is “seeing” in the k-Space data. One approach would be to use heatmap algorithms similar to algorithms such as *Grad-CAM* [165] to visualize the features learned by the model in the k-Space domain. A dual-domain approach could be used to visualize these features in the image domain.

7.2 CONCLUSION

With that we come to the end of this thesis. We have presented novel methods and architectures for the utilization of k-Space data in various downstream tasks. I want to come back to the quote by Erwin Schrödinger and Arthur Schopenhauer at the beginning of this thesis:

> *The task is not so much to see what no one has yet seen;*
> *but to think what nobody has yet thought,*
> *about that which everybody sees*.

The field of MRI research needs to undergo a paradigm shift. Faced with pressing resource limitations, including available MRI scanners and personnel, we still allow ourselves to waste at least 50% of the available data. This thesis aims to catalyze this shift by demonstrating the vast, untapped potential of direct k-Space processing for a variety of downstream tasks. Indeed we were able to show the benefits of this data which is often neglected, even though always available right after the acquisition. We hope that this work will inspire other researchers to utilize the k-Space data for their own tasks. The presented methods and architectures are only a starting point for future research in this area. I am convinced that the k-Space data has a lot of potential and I am looking forward to seeing what the future holds for this field.

A

Transparency Statement

a.1 K-STRIP: RETHINKING K-SPACE

This chapter is based on the publication “k-strip: A novel segmentation algorithm in k-space for the application of skull stripping" [51], published in the journal *Computer Methods and Programs in Biomedicine*. The source code of the k-Strip model is publicly available on GitHub. The following chapter are based on the aforementioned publication:

- **Introduction**: This section is added to put the work into context of the whole thesis.
- **Methods**: The methods section is an extension of the chapter *Methods* in the publication. The methods section is extended with additional information regarding implementation and background.
- **Experiments & Results**: This section is based on the chapters *Experiments* and *Results* in the publication.
- **Discussion & Conclusion**: This section is based on the chapters *Discussion* and *Conclusion* in the publication.

a.2 CLASSIFICATION IN K-SPACE: A CASE STUDY

This chapter is based on the publication “Tumor likelihood estimation on MRI prostate data by utilizing k-Space information” [82], which was presented at the International Society for Magnetic Resonance in Medicine (ISMRM) Conference 2025 as an oral presentation and published in full as a preprint on arXiv. The source code of the project is publicly available on GitHub. The following chapters are based on the aforementioned arXiv publication:

- **Introduction**: This section is an extension of the chapters *Introduction* and *Related Work* in the publication. The introduction is extended with a more detailed background for a better understanding in the context of this thesis.
- **Methods**: This section is an extension of the chapter *Material & Methods*, as well as partially the chapter *Experiments*, in the publication. The section is extended with additional information regarding implementation and background.

- **Experiments & Results**: This section is based on the chapters *Experiments* and *Results* in the publication.
- **Discussion & Conclusion**: This section is based on the chapters *Discussion* and *Conclusion* in the publication.

a.3 PHASEGEN: GENERATIVE AI MEETS K-SPACE

This chapter is based on the publication "PhaseGen: A Diffusion-Based Approach for Complex-Valued MRI Data Generation" [103], published as a preprint on ArXiv. The source code of the project is publicly available on GitHub. The following chapters are based on the aforementioned arXiv publication:

- **Introduction**: This section is an extension of the chapters *Introduction* in the publication.
- **Methods**: This section is an extension of the chapters *Related Work* and *Material & Methods* in the publication. The methods section is extended with additional information regarding the background of generative AI.
- **Experiments & Results**: This section is based on the chapter *Experiments & Results* in the publication. The section is extended with additional information regarding the used data and experiments.
- **Discussion & Conclusion**: This section is based on the chapters *Discussion* and *Conclusion* in the publication.

a.4 A COMPREHENSIVE TOOL FOR MEDICAL DE-IDENTICATION INCLUDING RAW MRI DATA

This chapter is based on the publication "De-identification of medical imaging data: a comprehensive tool for ensuring patient privacy" [146], which was published in the journal *European Radiology*. The source code of the project is publicly available on GitHub. The following chapters are based on the aforementioned publication:

- **Introduction**: This section is based on the chapters *Introduction* and *Related Work* in the publication.
- **Proposed De-Identification Tool**: This section is based on the chapter *Materials and Methods* in the publication. The section was shortened to focus on the key aspects relevant for this thesis.
- **MRI Raw Data De-Identification**: This section was added to provide a detailed overview of the tools functionality relevant for this thesis. It is partially based on the chapters *Materials and Methods* and *Results* in the publication.
- **Discussion & Conclusion**: This section is based on the chapters *Discussion* and *Conclusion* in the publication.

B

Supplementary Material

b.1 K-STRIP: RETHINKING K-SPACE

Tab. B.1 shows the datasets used for trainiing, validation and testing of the k-Strip model for the skullstripping task.

TABLE B.1: Overview of the four datasets used for training, validation and testing of the k-Strip model. The datasets are used to train the model on different modalities and field strengths.

Dataset	Subjects	Modality	Field strength	Vendor	Voxel Size
GBM	207	T1ce	1.5T/3.0T	Siemens	$2mm^3$
SWI	433	SWI (T1&Phase)	1.5T/3.0T	Siemens	$2mm^3$
NFBS [73]	125	T1	1.5T/3.0T	Siemens	$1mm^3$
CC-359 [107]	359	T1	1.5T/3.0T	GE/Philips/Siemens	$1mm^3$

b.2 CLASSIFICATION IN K-SPACE: A CASE STUDY

Figure B.1 shows the results of the augmentation on the PCA pipeline, GRAPPA pipeline, and the off-domain dataset CIFAR-10. The testing is performed without additional augmentation. Training is performed with a undersampling factor of up to $R = 8$. The CIFAR-10 dataset is used as an off-domain dataset to show the applicability of the augmentation method also to datasets not related to MRI. CIFAR-10 consists of 10 classes of images, each containing 6000 images compised of the shape 32×32 pixels. The results indicate that the augmentation can increase the performance for all three datasets

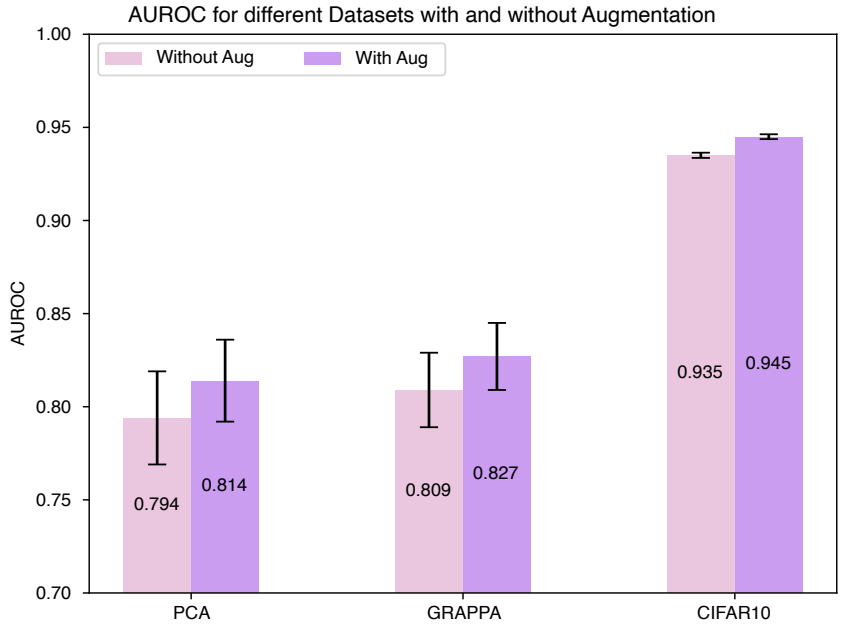

FIG. B.1: Results with and without additional undersampling augmentation on the PCA pipeline, GRAPPA pipeline and the off-domain dataset CIFAR-10. The results show that the undersampling augmentation can increase the performance for all three datasets.

TABLE B.2: Time consumption of the GRAPPA reconstruction pipeline and the proposed PCA pipeline. The times include the time for acquisition at different undersampling rates. The times are reported in minutes per volume. The reconstruction is performed on CPU devices.

Undersampling Factor R	GRAPPA (min / volume)	PCA (min / volume)
No Undersampling	26.0min	27.9min
$R = 2$	23.2min	15.1min
$R = 4$	16.8min	8.7min
$R = 8$	13.6min	5.5min
$R = 12$	12.5min	4.4min
$R = 16$	12.0min	3.9min
$R = 24$	11.5min	3.4min
$R = 32$	11.2min	3.1min
$R = 48$	10.9min	2.8min

b.3 PHASEGEN: GENERATIVE AI MEETS K-SPACE

Fig. B.2 shows examples of the naive phase generation method. This generative method is used to create synthetic phase information from available magnitude images. The resulting data is used to compare another generative method with the proposed PhaseGen model.

Tab. B.3 shows the datasets used for training, validation and testing of the PhaseGen, segmentation and reconstruction model.

Tab. B.4 shows the available sequences in the PhaseGen dataset. The dataset consists of five different sequences, each with a different number of slices. The included sequences are T_2 TSE, T_2 TIRM Dark Fluid, T_1 TSE, T_1 TSE Dark Fluid and T_1 FLAIR. The TSE sequences are common in clinical practice due to its low acquisition time and improved SNR. The TIRM Dark Fluid sequences

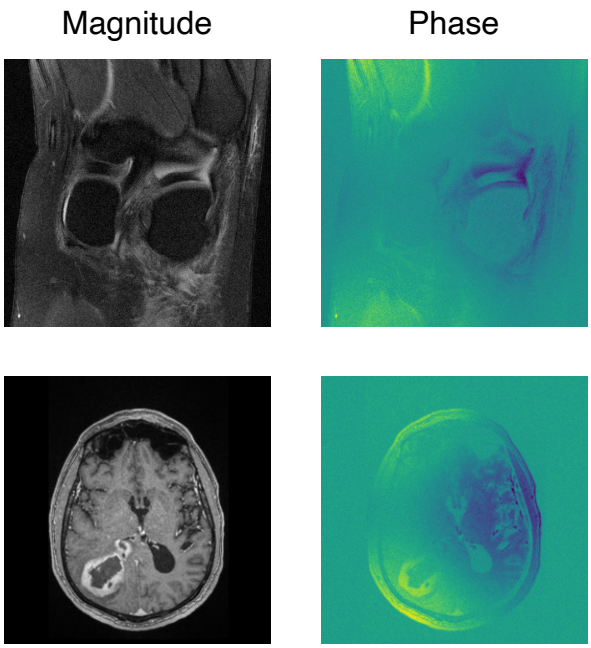

FIG. B.2: Examples of the naive phase generation. Shown are the magnitude as well as the generated phase. Top: Example from the FastMRI Knee dataset. Bottom: Example from the UK Essen Magnitude dataset.

TABLE B.3: Overview of the used datasets in the PhaseGen project. UME: University Hospital Essen.

Dataset	Size	Field strength	Type	Source
PhaseGen Training & Validation	14101 slices	1.5T/3.0T	Raw MRI	UME
Skullstrip Training & Validation	21822 slices	1.5T/3.0T	Image Domain	UME
Skullstrip Testing	14 volumes	1.5T/3.0T	Raw MRI	UME
Reconstruction Training & Validation	40450 slices	1.5T/3.0T	Raw Knee MRI	FastMRI
Reconstruction Testing	1427 slices	1.5T/3.0T	Raw Knee MRI	FastMRI

are used to suppress the signal from fluids, such as cerebrospinal fluid (CSF), which is useful for visualizing brain structures [166]. The T_1 TSE and T_1 FLAIR sequences are used to visualize different tissue contrasts, with FLAIR being particularly useful for detecting lesions in the brain [167]. Each of these sequences includes different orientations.

TABLE B.4: Overview of the available sequences in the PhaseGen dataset. TSE: Turbo Spin Echo, TIRM: Turbo Inversion Recovery Magnitude, FLAIR: Fluid Attenuated Inversion Recovery.

Sequence	Number of Slices
T_2 TSE	11462
T_2 TIRM Dark Fluid	1491
T_1 TSE	795
T_1 TSE Dark Fluid	284
T_1 FLAIR	69

Tab. B.5 and Tab. B.6 show the hyperparameters used for the segmentation and reconstruction task, respectively.

TABLE B.5: Hyperparameters for the skullstripping task.

Hyperparameter	PhaseGen
Learning Rate	1e-4
Batch Size	128
Dropout	0.2
Epochs	200
Noise Scheduler	Exponential
Activation function	PReLU
Optimizer	Adam
# Parameters	33.5M

TABLE B.6: Hyperparameters for the reconstruction task.

Hyperparameter	No Phase	Naive	PhaseGen
Learning Rate	3.8e-4	3.8e-4	4.4e-4
Batch Size	32	32	128
Dropout	0.1	0.1	0.4
Epochs	180	185	275
Noise Scheduler	Exponential	Exponential	Exponential
Activation function	ReLU	ReLU	ReLU
Optimizer	AdamW	AdamW	AdamW
# Parameters	209k / 3.3M	209k / 3.3M	209k / 3.3M

Bibliography

[1] Wilhelm Conrad Röntgen. *Über Eine Neue Art von Strahlen*. BoD–Books on Demand, 2020. ISBN: 3-7526-2578-3 (cit. on p. 1).

[2] Eyal Bercovich and Marcia C. Javitt. "Medical Imaging: From Roentgen to the Digital Revolution, and Beyond". In: *Rambam Maimonides Medical Journal* 9.4 (Oct. 4, 2018), e0034. ISSN: 2076-9172. DOI: 10.5041/RMMJ.10355. pmid: 30309440 (cit. on p. 1).

[3] Aaron Filler. "The History, Development and Impact of Computed Imaging in Neurological Diagnosis and Neurosurgery: CT, MRI, and DTI". In: *Nature Precedings* (May 2009). ISSN: 1756-0357. DOI: 10.1038/npre.2009.3267.1 (cit. on p. 1).

[4] P. C. Lauterbur. "Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance". In: *Nature* 242.5394 (Mar. 1973), pp. 190–191. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/242190a0 (cit. on p. 1).

[5] R Damadian, M Goldsmith, and L Minkoff. "Nmr in Cancer: Xvi. Fonar Image of the Uve Human Body". In: *Physiological chemistry and physics* 9.1 (1977), pp. 97–100 (cit. on p. 1).

[6] Bjørn Hofmann, Ingrid Øfsti Brandsaeter, and Elin Kjelle. "Variations in Wait Times for Imaging Services: A Register-Based Study of Self-Reported Wait Times for Specific Examinations in Norway". In: *BMC Health Services Research* 23.1 (2023), p. 1287. ISSN: 1472-6963 (cit. on pp. 1, 37).

[7] Julie Nightingale et al. "Establishing the Size and Configuration of the Imaging Support Workforce: A Census of National Workforce Data in England". In: *BJR| Open* 6.1 (2024), tzae026. ISSN: 2513-9878 (cit. on p. 1).

[8] Mansour Almanaa et al. "Evaluating MRI and CT Scan Scheduling Workflows: A Retrospective Analysis". In: *Journal of Radiation Research and Applied Sciences* 17.4 (2024), p. 101201. ISSN: 1687-8507 (cit. on p. 1).

[9] The Royal College of Radiologists. *Clinical Radiology UK Workforce Census 2023 Report*. Tech. rep. The Royal College of Radiologists, June 2024. URL: https://www.rcr.ac.uk/media/5befglss/rcr-census-clinical-radiology-workforce-census-2023.pdf (cit. on p. 2).

[10] Francisco Reyes-Santias et al. "Cost Analysis of Magnetic Resonance Imaging and Computed Tomography in Cardiology: A Case Study of a University Hospital Complex in the Euro Region". In: *Healthcare*. Vol. 11. MDPI, 2023, p. 2084. ISBN: 2227-9032 (cit. on p. 2).

[11] Kicky G. Van Leeuwen et al. "How Does Artificial Intelligence in Radiology Improve Efficiency and Health Outcomes?" In: *Pediatric Radiology* 52.11 (Oct. 2022), pp. 2087–2093. ISSN: 0301-0449, 1432-1998. DOI: 10.1007/s00247-021-05114-8 (cit. on p. 2).

[12] Raghav Singhal et al. "Accelerated MR Screenings with Direct K-Space Classification". In: *Joint Annual Meeting ISMRM-ESMRMB & ISMRT 31st Annual Meeting (London, 7–12 May 2022)*. 2022 (cit. on pp. 2, 39).

[13] Chen-Yu Yen et al. “Adaptive Sampling of K-Space in Magnetic Resonance for Rapid Pathology Prediction”. In: *arXiv preprint arXiv:2406.04318* (2024). arXiv: 2406.04318 (cit. on pp. 2, 39, 91, 92).

[14] Ruochen Li et al. “Classification, Regression and Segmentation Directly from k-Space in Cardiac MRI”. In: *International Workshop on Machine Learning in Medical Imaging*. Springer, 2024, pp. 31–41 (cit. on pp. 2, 21).

[15] Yuning Du et al. “Active Sampling for MRI-based Sequential Decision Making”. In: *arXiv preprint arXiv:2505.04586* (2025). arXiv: 2505.04586 (cit. on pp. 2, 39, 91, 92).

[16] Pieter Zeeman. *Ueber einen Einfluss der Magnetisirung auf die Natur des von einer Substanz emittirten Lichtes*. 1896 (cit. on p. 6).

[17] Joseph P Hornak. “The Basics of MRI”. In: (1996) (cit. on p. 6).

[18] Daniel D Traficante. “Relaxation. Can T2, Be Longer than T1?” In: *Concepts in Magnetic Resonance* 3.3 (1991), pp. 171–177. ISSN: 1043-7347 (cit. on p. 7).

[19] F. Bloch. “Nuclear Induction”. In: *Physical Review* 70.7-8 (Oct. 1946), pp. 460–474. ISSN: 0031-899X. DOI: 10.1103/PhysRev.70.460 (cit. on p. 7).

[20] G McGibney et al. “Quantitative Evaluation of Several Partial Fourier Reconstruction Algorithms Used in MRI”. In: *Magnetic resonance in medicine* 30.1 (1993), pp. 51–59. ISSN: 0740-3194 (cit. on p. 11).

[21] Paul A Bottomley et al. “A Review of Normal Tissue Hydrogen NMR Relaxation Times and Relaxation Mechanisms from 1–100 MHz: Dependence on Tissue Type, NMR Frequency, Temperature, Species, Excision, and Age”. In: *Medical physics* 11.4 (1984), pp. 425–448. ISSN: 0094-2405 (cit. on p. 12).

[22] Roland Bammer. “Basic Principles of Diffusion-Weighted Imaging”. In: *European journal of radiology* 45.3 (2003), pp. 169–184. ISSN: 0720-048X (cit. on p. 13).

[23] Anna Caroli et al. “Diffusion-Weighted Magnetic Resonance Imaging to Assess Diffuse Renal Pathology: A Systematic Review and Statement Paper”. In: *Nephrology Dialysis Transplantation* 33.suppl_2 (2018), pp. ii29–ii40. ISSN: 0931-0509 (cit. on p. 13).

[24] Edward O Stejskal and John E Tanner. “Spin Diffusion Measurements: Spin Echoes in the Presence of a Time-dependent Field Gradient”. In: *The journal of chemical physics* 42.1 (1965), pp. 288–292. ISSN: 0021-9606 (cit. on p. 13).

[25] Klaas P Pruessmann et al. “SENSE: Sensitivity Encoding for Fast MRI”. In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 42.5 (1999), pp. 952–962. ISSN: 0740-3194 (cit. on p. 14).

[26] Mark A Griswold et al. “Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA)”. In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 47.6 (2002), pp. 1202–1210. ISSN: 0740-3194 (cit. on pp. 14, 41).

[27] Peter Mildenberger, Marco Eichelberg, and Eric Martin. “Introduction to the DICOM Standard”. In: *European radiology* 12 (2002), pp. 920–927. ISSN: 0938-7994 (cit. on p. 14).

[28] M Brett et al. “Nipy/Nibabel: 5.2. 1. Zenodo; 2024”. In: () (cit. on p. 15).

[29] DFW Group. “Neuroimaging Informatics Technology Initiative”. In: (2018) (cit. on p. 15).

[30] Andrew Collette et al. “H5py/H5py: 3.8. 0”. In: (2023) (cit. on pp. 15, 65).

[31] Mike Folk et al. “An Overview of the HDF5 Technology Suite and Its Applications”. In: *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*. 2011, pp. 36–47 (cit. on p. 15).

[32] A Paszke. “Pytorch: An Imperative Style, High-Performance Deep Learning Library”. In: *arXiv preprint arXiv:1912.01703* (2019). arXiv: 1912.01703 (cit. on p. 15).

[33] John McCarthy et al. “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955”. In: *AI magazine* 27.4 (2006), pp. 12–12. ISSN: 2371-9621 (cit. on p. 15).

[34] John McCarthy. “What Is Artificial Intelligence”. In: (2007) (cit. on p. 15).

[35] Shun-ichi Amari. “Backpropagation and Stochastic Gradient Descent Method”. In: *Neurocomputing* 5.4-5 (1993), pp. 185–196. ISSN: 0925-2312 (cit. on p. 16).

[36] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958. ISSN: 1532-4435 (cit. on p. 16).

[37] Yann LeCun et al. “Learning Algorithms for Classification: A Comparison on Handwritten Digit Recognition”. In: *Neural networks: the statistical mechanics perspective* 261.276 (1995), p. 2 (cit. on p. 17).

[38] 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: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241. ISBN: 3-319-24573-2 (cit. on pp. 17, 23).

[39] Fabian Isensee et al. “nnU-Net: A Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation”. In: *Nature methods* 18.2 (2021), pp. 203–211. ISSN: 1548-7105 (cit. on p. 17).

[40] Huilin Mu et al. “CV-GMTINet: GMTI Using a Deep Complex-Valued Convolutional Neural Network for Multichannel SAR-GMTI System”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–15. ISSN: 0196-2892 (cit. on p. 17).

[41] Alexander Fuchs et al. “Complex-Valued Convolutional Neural Networks for Enhanced Radar Signal Denoising and Interference Mitigation”. In: *2021 IEEE Radar Conference (RadarConf21)*. IEEE, 2021, pp. 1–6. ISBN: 1-7281-7609-3 (cit. on p. 17).

[42] ChiYan Lee, Hideyuki Hasegawa, and Shangce Gao. “Complex-Valued Neural Networks: A Comprehensive Survey”. In: *IEEE/CAA Journal of Automatica Sinica* 9.8 (2022), pp. 1406–1426. ISSN: 2329-9266 (cit. on p. 18).

[43] MM Hammad. “Comprehensive Survey of Complex-Valued Neural Networks: Insights into Backpropagation and Activation Functions”. In: *arXiv preprint arXiv:2407.19258* (2024). arXiv: 2407.19258 (cit. on p. 18).

[44] Wilhelm Wirtinger. “Zur Formalen Theorie Der Funktionen von Mehr Komplexen Veränderlichen”. In: *Mathematische Annalen* 97.1 (1927), pp. 357–375. ISSN: 0025-5831 (cit. on p. 18).

[45] Tohru Nitta. “An Extension of the Back-Propagation Algorithm to Complex Numbers”. In: *Neural Networks* 10.8 (1997), pp. 1391–1415. ISSN: 0893-6080 (cit. on p. 18).

[46] Reinhold Remmert. *Theory of Complex Functions*. Vol. 122. Springer Science & Business Media, 1991. ISBN: 0-387-97195-5 (cit. on p. 18).

[47] Jose Agustin Barrachina et al. “Theory and Implementation of Complex-Valued Neural Networks”. In: *arXiv preprint arXiv:2302.08286* (2023). arXiv: 2302.08286 (cit. on p. 19).

[48] Akira Hirose. *Complex-Valued Neural Networks*. Vol. 32. Wiley Online Library, 2006 (cit. on p. 19).

[49] Rayyan Abdalla. "Complex-Valued Neural Networks–Theory and Analysis". In: *arXiv preprint arXiv:2312.06087* (2023). arXiv: 2312.06087 (cit. on p. 20).

[50] Yundi Zhang et al. "Direct Cardiac Segmentation from Undersampled K-Space Using Transformers". In: *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2024, pp. 1–4. ISBN: 979-8-3503-1333-8 (cit. on pp. 21, 23).

[51] Moritz Rempe et al. "K-Strip: A Novel Segmentation Algorithm in k-Space for the Application of Skull Stripping". In: *Computer Methods and Programs in Biomedicine* 243 (2024), p. 107912. ISSN: 0169-2607 (cit. on pp. 21, 24, 33, 34, 95).

[52] Hafiz Zia Ur Rehman, Hyunho Hwang, and Sungon Lee. "Conventional and Deep Learning Methods for Skull Stripping in Brain MRI". In: *Applied Sciences* 10.5 (2020), p. 1773. ISSN: 2076-3417 (cit. on p. 22).

[53] Tony CW Mok and Albert Chung. "Affine Medical Image Registration with Coarse-to-Fine Vision Transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20835–20844 (cit. on p. 23).

[54] K Bhagyalaxmi, B Dwarakanath, and P Vijaya Pal Reddy. "Deep Learning for Multi-Grade Brain Tumor Detection and Classification: A Prospective Survey". In: *Multimedia Tools and Applications* 83.25 (2024), pp. 65889–65911. ISSN: 1573-7721 (cit. on p. 23).

[55] Fina GieSSler et al. "Facial Feature Removal for Anonymization of Neurological Image Data". In: *Current Directions in Biomedical Engineering*. Vol. 7. De Gruyter, 2021, pp. 130–134. ISBN: 2364-5504 (cit. on p. 23).

[56] Mouli Laha, Prasun Chandra Tripathi, and Soumen Bag. "A Skull Stripping from Brain MRI Using Adaptive Iterative Thresholding and Mathematical Morphology". In: *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*. IEEE, 2018, pp. 1–6. ISBN: 1-5386-3039-7 (cit. on p. 23).

[57] Jimit Doshi et al. "Multi-Atlas Skull-Stripping". In: *Academic radiology* 20.12 (2013), pp. 1566–1576. ISSN: 1076-6332 (cit. on p. 23).

[58] Evan S Lutkenhoff et al. "Optimized Brain Extraction for Pathological Brains (optiBET)". In: *PloS one* 9.12 (2014), e115551. ISSN: 1932-6203 (cit. on pp. 23, 32).

[59] Andrew Hoopes et al. "SynthStrip: Skull-Stripping for Any Brain Image". In: *NeuroImage* 260 (2022), p. 119474. ISSN: 1053-8119 (cit. on pp. 23, 82).

[60] Fabian Isensee et al. "Automated Brain Extraction of Multisequence MRI Using Artificial Neural Networks". In: *Human brain mapping* 40.17 (2019), pp. 4952–4964. ISSN: 1065-9471 (cit. on pp. 23, 32, 82).

[61] Jens Kleesiek et al. "Deep MRI Brain Extraction: A 3D Convolutional Neural Network for Skull Stripping". In: *NeuroImage* 129 (2016), pp. 460–469. ISSN: 1053-8119 (cit. on p. 23).

[62] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778 (cit. on p. 24).

[63] Chiheb Trabelsi et al. "Deep Complex Networks". In: *arXiv preprint arXiv:1705.09792* (2017). arXiv: 1705.09792 (cit. on pp. 24, 91).

[64] Maxime W Matthès et al. “Learning and Avoiding Disorder in Multimode Fibers”. In: *Physical Review X* 11.2 (2021), p. 021060. ISSN: 2160-3308 (cit. on p. 24).

[65] Mehmet Çelebi and Murat Ceylan. “The New Activation Function for Complex Valued Neural Networks: Complex Swish Function”. In: *SETSCI-Conference Proceedings*. Vol. 9. SETSCI-Conference Proceedings, 2019, pp. 169–173. ISBN: 2687-5527 (cit. on p. 25).

[66] Patrick M Virtue. *Complex-Valued Deep Learning with Applications to Magnetic Resonance Image Synthesis*. University of California, Berkeley, 2019. ISBN: 1-392-58560-0 (cit. on p. 25).

[67] Simone Scardapane et al. “Complex-Valued Neural Networks with Nonparametric Activation Functions”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 4.2 (2018), pp. 140–150. ISSN: 2471-285X (cit. on p. 25).

[68] Kerstin Hammernik, Thomas Küstner, and Daniel Rueckert. “Machine Learning For MRI Reconstruction”. In: *MRI Reconstruction: Theory, Methods and Applications*. 2021 (in Preparation) (cit. on p. 26).

[69] Afia Zafar et al. “A Comparison of Pooling Methods for Convolutional Neural Networks”. In: *Applied Sciences* 12.17 (2022), p. 8643. ISSN: 2076-3417 (cit. on p. 26).

[70] Oren Rippel, Jasper Snoek, and Ryan P Adams. “Spectral Representations for Convolutional Neural Networks”. In: *Advances in neural information processing systems* 28 (2015) (cit. on p. 27).

[71] Moritz Rempe. “Exploiting MRI k-space for segmentation using complex-valued convolutional neural networks”. MA thesis. Dortmund, Germany: TU Dortmund University, Faculty of Physics, AG Kröninger, 2022 (cit. on p. 27).

[72] Vincent Dumoulin and Francesco Visin. “A Guide to Convolution Arithmetic for Deep Learning”. In: *arXiv preprint arXiv:1603.07285* (2016). arXiv: 1603.07285 (cit. on p. 27).

[73] Benjamin Puccio et al. “The Preprocessed Connectomes Project Repository of Manually Corrected Skull-Stripped T1-weighted Anatomical MRI Data”. In: *Gigascience* 5.1 (2016), s13742–016. ISSN: 2047-217X (cit. on pp. 29, 32, 97).

[74] Simon F Eskildsen et al. “BEaST: Brain Extraction Based on Nonlocal Segmentation Technique”. In: *NeuroImage* 59.3 (2012), pp. 2362–2373. ISSN: 1053-8119 (cit. on pp. 29, 85).

[75] Roberto Souza et al. “An Open, Multi-Vendor, Multi-Field-Strength Brain MR Dataset and Analysis of Publicly Available Skull Stripping Methods Agreement”. In: *NeuroImage* 170 (2018), pp. 482–494. ISSN: 1053-8119 (cit. on pp. 29, 32, 85).

[76] Simon K Warfield, Kelly H Zou, and William M Wells. “Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation”. In: *IEEE transactions on medical imaging* 23.7 (2004), pp. 903–921. ISSN: 0278-0062 (cit. on p. 29).

[77] Diederik P Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv preprint arXiv:1412.6980* (2014). arXiv: 1412.6980 (cit. on p. 31).

[78] Lee R Dice. “Measures of the Amount of Ecologic Association between Species”. In: *Ecology* 26.3 (1945), pp. 297–302. ISSN: 0012-9658 (cit. on p. 31).

[79] Felix Hausdorff. *Grundzüge Der Mengenlehre*. Vol. 7. von Veit, 1914. ISBN: 3-11-098985-9 (cit. on p. 31).

[80] David W Shattuck and Richard M Leahy. “BrainSuite: An Automated Cortical Surface Identification Tool”. In: *Medical image analysis* 6.2 (2002), pp. 129–142. ISSN: 1361-8415 (cit. on p. 32).

[81] Paul A Yushkevich et al. “User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability”. In: *Neuroimage* 31.3 (2006), pp. 1116–1128. ISSN: 1053-8119 (cit. on p. 34).

[82] Moritz Rempe et al. “Tumor Likelihood Estimation on MRI Prostate Data by Utilizing K-Space Information”. In: *arXiv preprint arXiv:2407.06165* (2024). arXiv: 2407.06165 (cit. on pp. 37, 43, 46, 48, 50, 51, 95).

[83] William A Edelstein, Mahadevappa Mahesh, and John A Carrino. “MRI: Time Is Dose—and Money and Versatility”. In: *Journal of the American College of Radiology: JACR* 7.8 (2010), p. 650 (cit. on p. 37).

[84] Rebecca L Siegel et al. “Cancer Statistics, 2023”. In: *CA: a cancer journal for clinicians* 73.1 (2023), pp. 17–48. ISSN: 0007-9235 (cit. on p. 38).

[85] Philip Cornford et al. “EAU-EANM-ESTRO-ESUR-ISUP-SIOG Guidelines on Prostate Cancer—2024 Update. Part I: Screening, Diagnosis, and Local Treatment with Curative Intent”. In: *European urology* (2024). ISSN: 0302-2838 (cit. on p. 38).

[86] Baris Turkbey et al. “Prostate Imaging Reporting and Data System Version 2.1: 2019 Update of Prostate Imaging Reporting and Data System Version 2”. In: *European urology* 76.3 (2019), pp. 340–351. ISSN: 0302-2838 (cit. on p. 38).

[87] Sunghwan Yoo et al. “Prostate Cancer Detection Using Deep Convolutional Neural Networks”. In: *Scientific reports* 9.1 (2019), p. 19518. ISSN: 2045-2322 (cit. on p. 38).

[88] Patrick Schelb et al. “Classification of Cancer at Prostate MRI: Deep Learning versus Clinical PI-RADS Assessment”. In: *Radiology* 293.3 (2019), pp. 607–617. ISSN: 0033-8419 (cit. on p. 38).

[89] Radhika Tibrewala et al. “FastMRI Prostate: A Public, Biparametric MRI Dataset to Advance Machine Learning for Prostate Cancer Imaging”. In: *Scientific Data* 11.1 (2024), p. 404. ISSN: 2052-4463 (cit. on pp. 39, 56).

[90] Martin Uecker et al. “ESPIRiT—an Eigenvalue Approach to Autocalibrating Parallel MRI: Where SENSE Meets GRAPPA”. In: *Magnetic resonance in medicine* 71.3 (2014), pp. 990–1001. ISSN: 0740-3194 (cit. on p. 41).

[91] Nicole Seiberlich et al. “Improved Radial GRAPPA Calibration for Real-time Free-breathing Cardiac Imaging”. In: *Magnetic resonance in medicine* 65.2 (2011), pp. 492–505. ISSN: 0740-3194 (cit. on p. 41).

[92] David O Walsh, Arthur F Gmitro, and Michael W Marcellin. “Adaptive Reconstruction of Phased Array MR Imagery”. In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 43.5 (2000), pp. 682–690. ISSN: 0740-3194 (cit. on p. 42).

[93] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal Component Analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52. ISSN: 0169-7439 (cit. on p. 42).

[94] Jia Deng et al. “Imagenet: A Large-Scale Hierarchical Image Database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255. ISBN: 1-4244-3992-2 (cit. on p. 44).

[95] Frederic Jonske et al. “Why Does My Medical AI Look at Pictures of Birds? Exploring the Efficacy of Transfer Learning across Domain Boundaries”. In: *Computer Methods and Programs in Biomedicine* (2025), p. 108634. ISSN: 0169-2607 (cit. on p. 44).

[96] Zhuang Liu et al. “A Convnet for the 2020s”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11976–11986 (cit. on p. 45).

[97] Wightman, Ross. *Pytorch Image Models*. 2019 (cit. on p. 45).

[98] Alex Krizhevsky and Geoffrey Hinton. “Learning Multiple Layers of Features from Tiny Images”. In: (2009) (cit. on p. 45).

[99] Bradley Efron. “Bootstrap Methods: Another Look at the Jackknife”. In: *Breakthroughs in Statistics: Methodology and Distribution*. Springer, 1992, pp. 569–593 (cit. on p. 47).

[100] Patrick Virtue and Michael Lustig. “The Empirical Effect of Gaussian Noise in Undersampled MRI Reconstruction”. In: *Tomography* 3.4 (2017), p. 211 (cit. on p. 52).

[101] Julie Woodfield and Susan Kealey. “Magnetic Resonance Imaging Acquisition Techniques Intended to Decrease Movement Artefact in Paediatric Brain Imaging: A Systematic Review”. In: *Pediatric radiology* 45 (2015), pp. 1271–1281. ISSN: 0301-0449 (cit. on p. 52).

[102] Nora-Josefin Breutigam et al. “Combined Acquisition of Diffusion and T2*-Weighted Measurements Using Simultaneous Multi-Contrast Magnetic Resonance Imaging”. In: *Magnetic Resonance Materials in Physics, Biology and Medicine* 35.3 (2022), pp. 421–440. ISSN: 1352-8661 (cit. on p. 52).

[103] Moritz Rempe et al. “PhaseGen: A Diffusion-Based Approach for Complex-Valued MRI Data Generation”. In: *arXiv preprint arXiv:2504.07560* (2025). arXiv: 2504.07560 (cit. on pp. 55, 56, 59, 60, 66, 68, 72, 74, 96).

[104] Katharine A Dishner et al. “A Survey of Publicly Available MRI Datasets for Potential Use in Artificial Intelligence Research”. In: *Journal of Magnetic Resonance Imaging* 59.2 (2024), pp. 450–480. ISSN: 1053-1807 (cit. on p. 56).

[105] Jure Zbontar et al. “fastMRI: An Open Dataset and Benchmarks for Accelerated MRI”. In: *arXiv preprint arXiv:1811.08839* (2018). arXiv: 1811.08839 (cit. on pp. 56, 65, 71).

[106] Eddy Solomon et al. “fastMRI Breast: A Publicly Available Radial k-Space Dataset of Breast Dynamic Contrast-Enhanced MRI”. In: *Radiology: Artificial Intelligence* 7.1 (2025), e240345. ISSN: 2638-6100 (cit. on p. 56).

[107] Roberto Souza et al. “Calgary-Campinas Raw k-Space Dataset: A Benchmark for Brain Magnetic Resonance Image Reconstruction”. In: *Joint Annual Meeting ISMRM & SMRT 31st Annual Meeting (Virtual, 08–14 August 2020)* (2020) (cit. on pp. 56, 97).

[108] Stefan Feuerriegel et al. “Generative Ai”. In: *Business & Information Systems Engineering* 66.1 (2024), pp. 111–126. ISSN: 2363-7005 (cit. on p. 57).

[109] Connor Shorten and Taghi M Khoshgoftaar. “A Survey on Image Data Augmentation for Deep Learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48. ISSN: 2196-1115 (cit. on p. 57).

[110] Fazle Rahat et al. “Data Augmentation for Image Classification Using Generative Ai”. In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 4173–4182. ISBN: 979-8-3315-1083-1 (cit. on p. 57).

[111] Kaliprasad Pani and Indu Chawla. “Synthetic MRI in Action: A Novel Framework in Data Augmentation Strategies for Robust Multi-Modal Brain Tumor Segmentation”. In: *Computers in Biology and Medicine* 183 (2024), p. 109273. ISSN: 0010-4825 (cit. on p. 57).

[112] André Ferreira et al. “Enhanced Data Augmentation Using Synthetic Data for Brain Tumour Segmentation”. In: *International Challenge on Cross-Modality Domain Adaptation for Medical Image Segmentation*. Springer, 2023, pp. 79–93 (cit. on p. 57).

[113] Robin Rombach et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695 (cit. on p. 57).

[114] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851 (cit. on pp. 57, 60).

[115] Diederik P Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: (2013) (cit. on p. 57).

[116] Ian Goodfellow et al. “Generative Adversarial Networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144. ISSN: 0001-0782 (cit. on p. 57).

[117] Youssef Kossale, Mohammed Airaj, and Aziz Darouichi. “Mode Collapse in Generative Adversarial Networks: An Overview”. In: *2022 8th International Conference on Optimization and Applications (ICOA)*. IEEE, 2022, pp. 1–6. ISBN: 1-6654-7681-8 (cit. on p. 57).

[118] Prafulla Dhariwal and Alexander Nichol. “Diffusion Models Beat Gans on Image Synthesis”. In: *Advances in neural information processing systems* 34 (2021), pp. 8780–8794 (cit. on p. 58).

[119] Muhammad Usman Akbar et al. “Brain Tumor Segmentation Using Synthetic MR Images-A Comparison of GANs and Diffusion Models”. In: *Scientific Data* 11.1 (2024), p. 259. ISSN: 2052-4463 (cit. on p. 58).

[120] Eliya Nachmani, Robin San Roman, and Lior Wolf. “Non Gaussian Denoising Diffusion Models”. In: *arXiv preprint arXiv:2106.07582* (2021). arXiv: 2106.07582 (cit. on p. 58).

[121] Solomon Kullback and Richard A Leibler. “On Information and Sufficiency”. In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86. ISSN: 0003-4851 (cit. on p. 58).

[122] Martin Heusel et al. “Gans Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 61).

[123] Jonathan Ho and Tim Salimans. “Classifier-Free Diffusion Guidance”. In: *arXiv preprint arXiv:2207.12598* (2022). arXiv: 2207.12598 (cit. on p. 61).

[124] Tim Salimans et al. “Improved Techniques for Training Gans”. In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 61).

[125] Christian Szegedy et al. “Going Deeper with Convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9 (cit. on p. 61).

[126] Brandon Trabucco et al. “Effective Data Augmentation with Diffusion Models”. In: *arXiv preprint arXiv:2302.07944* (2023). arXiv: 2302.07944 (cit. on p. 61).

[127] Kai Packhäuser et al. “Generation of Anonymous Chest Radiographs Using Latent Diffusion Models for Training Thoracic Abnormality Classification Systems”. In: *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2023, pp. 1–5. ISBN: 1-6654-7358-4 (cit. on p. 61).

[128] Alain Hore and Djemel Ziou. “Image Quality Metrics: PSNR vs. SSIM”. In: *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 2366–2369. ISBN: 1-4244-7541-4 (cit. on p. 62).

[129] Zhou Wang et al. “Image Quality Assessment: From Error Visibility to Structural Similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612. ISSN: 1057-7149 (cit. on p. 63).

[130] P Ehses and C Huynh. “Physics, BM Twixtools”. In: *GitHub repository* (2020) (cit. on p. 64).

[131] Souheil J Inati et al. “ISMRM Raw Data Format: A Proposed Standard for MRI Raw Datasets”. In: *Magnetic resonance in medicine* 77.1 (2017), pp. 411–421. ISSN: 0740-3194 (cit. on p. 65).

[132] Mark Tygert and Jure Zbontar. “Simulating Single-Coil MRI from the Responses of Multiple Coils”. In: *Communications in Applied Mathematics and Computational Science* 15.2 (2020), pp. 115–127. ISSN: 2157-5452 (cit. on p. 65).

[133] Peter B Roemer et al. “The NMR Phased Array”. In: *Magnetic resonance in medicine* 16.2 (1990), pp. 192–225. ISSN: 0740-3194 (cit. on p. 65).

[134] R Kramer and Otmar Loffeld. “A Novel Procedure for Cutline Detection”. In: *AEU-ARCHIV FUR ELEKTRONIK UND UBERTRAGUNGSTECHNIK-INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATIONS* 50.2 (1996), pp. 112–116. ISSN: 0001-1096 (cit. on p. 66).

[135] Sofia Chavez, Qing-San Xiang, and Li An. “Understanding Phase Maps in MRI: A New Cutline Phase Unwrapping Method”. In: *IEEE transactions on medical imaging* 21.8 (2002), pp. 966–977. ISSN: 0278-0062 (cit. on p. 66).

[136] Zhi-Pei Liang. “A Model-Based Method for Phase Unwrapping”. In: *IEEE transactions on medical imaging* 15.6 (1996), pp. 893–897. ISSN: 0278-0062 (cit. on p. 66).

[137] Hongyu Zhou et al. “The PHU-NET: A Robust Phase Unwrapping Method for MRI Based on Deep Learning”. In: *Magnetic Resonance in Medicine* 86.6 (2021), pp. 3321–3333. ISSN: 0740-3194 (cit. on p. 66).

[138] Blake E Dewey. “Laplacian-Based Phase Unwrapping in Python (v1. 0)”. In: *Zenodo. https://doi.org/10.5281/zenodo* 7198991 (2022) (cit. on p. 67).

[139] Santiago Aja-Fernández and Gonzalo Vegas-Sánchez-Ferrero. “Statistical Analysis of Noise in MRI”. In: *Switzerland: Springer International Publishing* (2016) (cit. on p. 67).

[140] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: `https://www.wandb.com/` (cit. on p. 67).

[141] Chang Min Hyun et al. “Deep Learning for Undersampled MRI Reconstruction”. In: *Physics in Medicine & Biology* 63.13 (2018), p. 135007. ISSN: 1361-6560 (cit. on p. 69).

[142] Arghya Pal and Yogesh Rathi. “A Review and Experimental Evaluation of Deep Learning Methods for MRI Reconstruction”. In: *The journal of machine learning for biomedical imaging* 1 (2022), p. 001 (cit. on p. 69).

[143] Hao Zheng, Faming Fang, and Guixu Zhang. “Cascaded Dilated Dense Network with Two-Step Data Consistency for MRI Reconstruction”. In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 70).

[144] Jingshuai Liu, Chen Qin, and Mehrdad Yaghoobi. “High-Fidelity MRI Reconstruction Using Adaptive Spatial Attention Selection and Deep Data Consistency Prior”. In: *IEEE Transactions on Computational Imaging* 9 (2023), pp. 298–313. ISSN: 2333-9403 (cit. on p. 70).

[145] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *arXiv preprint arXiv:1711.05101* (2017). arXiv: `1711.05101` (cit. on p. 71).

[146] Moritz Rempe et al. “De-Identification of Medical Imaging Data: A Comprehensive Tool for Ensuring Patient Privacy”. In: *European Radiology* (June 2025). ISSN: 1432-1084. DOI: `10.1007/s00330-025-11695-x` (cit. on pp. 80, 81, 85, 96).

[147] Blake Murdoch. “Privacy and Artificial Intelligence: Challenges for Protecting Health Information in a New Era”. In: *BMC medical ethics* 22 (2021), pp. 1–5 (cit. on p. 82).

[148] Christopher G Schwarz et al. “Identification of Anonymous MRI Research Participants with Face-Recognition Software”. In: *New England Journal of Medicine* 381.17 (2019), pp. 1684–1686. ISSN: 0028-4793 (cit. on p. 82).

[149] Darcy Mason. “SU-E-T-33: Pydicom: An Open Source DICOM Library”. In: *Medical Physics* 38.6Part10 (2011), pp. 3493–3493. ISSN: 0094-2405 (cit. on p. 82).

[150] Amanda Bischoff-Grethe et al. “A Technique for the Deidentification of Structural Brain MR Images”. In: *Human brain mapping* 28.9 (2007), pp. 892–903. ISSN: 1065-9471 (cit. on p. 82).

[151] Omer Faruk Gulban et al. *Poldracklab/Pydeface: PyDeface v2.0.2*. Zenodo. July 2022. DOI: 10.5281/ZENODO.6856482. (Visited on 10/04/2024) (cit. on p. 82).

[152] DICOM Committee. “DICOM PS3.15 2024c - Security and System Management Profiles”. In: *Published online (2024). URL: https://dicom.nema.org/medical/dicom/* () (cit. on p. 83).

[153] Saikat Roy et al. “Mednext: Transformer-Driven Scaling of Convnets for Medical Image Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2023, pp. 405–415 (cit. on p. 83).

[154] Anthony Kay. “Tesseract: An Open-Source Optical Character Recognition Engine”. In: *Linux Journal* 2007.159 (2007), p. 2. ISSN: 1075-3583 (cit. on p. 83).

[155] Athena E Theyers et al. “Multisite Comparison of MRI Defacing Software across Multiple Cohorts”. In: *Frontiers in psychiatry* 12 (2021), p. 617997. ISSN: 1664-0640 (cit. on p. 84).

[156] Alfred V Aho. *Algorithms for finding patterns in strings, Handbook of theoretical computer science (vol. A): algorithms and complexity*. 1991 (cit. on p. 84).

[157] Health Level Seven International. *FHIR Release 5.0.0 - Normative and STU*. https://hl7.org/fhir/. HL7 International. 2023 (cit. on p. 90).

[158] World Health Organization. *International Statistical Classification of Diseases and Related Health Problems: Alphabetical Index*. Vol. 3. World Health Organization, 2004. ISBN: 92-4-154654-9 (cit. on p. 90).

[159] Lu Chi, Borui Jiang, and Yadong Mu. “Fast Fourier Convolution”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4479–4488 (cit. on p. 91).

[160] Harry Pratt et al. “Fcnn: Fourier Convolutional Neural Networks”. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 17*. Springer, 2017, pp. 786–798. ISBN: 3-319-71248-9 (cit. on p. 91).

[161] Patrick Virtue, X Yu Stella, and Michael Lustig. “Better than Real: Complex-valued Neural Nets for MRI Fingerprinting”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3953–3957. ISBN: 1-5090-2175-2 (cit. on p. 91).

[162] Jiechao Wang et al. “Joint Optimization of Cartesian Sampling Patterns and Reconstruction for Single-contrast and Multi-contrast Fast Magnetic Resonance Imaging”. In: *Computer Methods and Programs in Biomedicine* 226 (2022), p. 107150. ISSN: 0169-2607 (cit. on pp. 91, 92).

[163] Tim Bakker et al. “On Learning Adaptive Acquisition Policies for Undersampled Multi-Coil MRI Reconstruction”. In: *International Conference on Medical Imaging with Deep Learning*. PMLR, 2022, pp. 63–85. ISBN: 2640-3498 (cit. on pp. 91, 92).

[164] Dina Abdelhamid. “Extraction and Classification of Radiomics Features in k-Space: A Machine Learning Approach for Tumor Grading”. Master’s thesis. Cairo: Technical University Dortmund, 2023 (cit. on p. 92).

[165] Ramprasaath R Selvaraju et al. “Grad-CAM: Why Did You Say That?” In: *arXiv preprint arXiv:1611.07450* (2016). arXiv: 1611.07450 (cit. on p. 92).

[166] GM Bydder, JV Hajnal, and IR Young. “MRI: Use of the Inversion Recovery Pulse Sequence”. In: *Clinical radiology* 53.3 (1998), pp. 159–176. ISSN: 0009-9260 (cit. on p. 99).

[167] Rohit Bakshi et al. “Fluid-Attenuated Inversion Recovery Magnetic Resonance Imaging Detects Cortical and Juxtacortical Multiple Sclerosis Lesions”. In: *Archives of neurology* 58.5 (2001), pp. 742–748. ISSN: 0003-9942 (cit. on p. 99).

Declaration

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly. The legal binding declaration (Eidesstattliche Versicherung) is appended at the end of this dissertation. All scientific content, ideas, and analyses presented in this thesis are my own original work or properly cited where derived from other sources.

I acknowledge the use of the following writing assisting tools in the preparation of this thesis: For the language editing of this thesis, Chat-GPT (GPT 4/4o) and Gemini 2.5 Pro have been used. These tools were used solely for language editing purposes, specifically for grammar correction, clarity improvement, and text coherence enhancement. I declare that these tools were not used to generate, modify, or enhance any scientific content, analysis, or conclusions.