Karlsruhe Institute of Technology

Artificial Intelligence in Magnetic Resonance: A Focus on Enhanced NMR Shimming and RASER MRI Artefact Correction

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)
angenommene

DISSERTATION

von

M. Sc. Moritz Becker

Tag der mündlichen Prüfung:	24.04.2024
Erstgutachter:	Prof. Dr. Jan G. Korvink
Zweitgutachter:	Prof. Dr. Pascal Friederich
Betreuender Mitarbeiter:	Dr. Mazin Jouda

–

Karlsruher Institut für Technologie
Institut für Mikrostrukturtechnik
Hermann-von-Helmholtz Platz 1
76344 Eggenstein-Leopoldshafen

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Änderungen entnommen wurde.

Karlsruhe, 24.04.2024

. . . .

(M. Sc. Moritz Becker)

Kurzfassung

In dieser Dissertation werden Methoden der künstlichen Intelligenz (KI), wie z. B. Deep Learning (DL), für zwei Anwendungen in der Kernspinresonanzspektroskopie (NMR) und der Magnetresonanztomographie (MRT) untersucht. Konkret wurde ein KI-gesteuerter Ansatz zur Verbesserung des mühsamen und zeitaufwändigen Shimming-Prozesses in der NMR-Spektroskopie initiiert, und die Bildqualität einer neuen MRT-Technik, basierend auf dem RASER (Radio-frequency Amplification by Stimulated emission of Radiation), wurde durch DL-basierte Artefaktentfernung verbessert.

Ein hochgradig homogenes Magnetfeld, bis hin zu Teilen pro Milliarde (ppb), ist entscheidend für genaue NMR/MRT-Ergebnisse. Dies kann durch die Verwendung von "Shim-Spulen" erreicht werden, um Magnetfeldgradienten auszugleichen und das Feld so einheitlich wie möglich zu gestalten, was jedoch ein zeitaufwändiger und mühsamer Prozess ist. Daher ist ein angemessenes und schnelles Shimming von entscheidender Bedeutung, das durch die Nicht-Bijektivität zwischen Feldverzerrungen im Volumen der Probe und eindimensionalen NMR-Signalen erschwert wird. In vier Studien, die den Hauptbeitrag dieser Arbeit darstellen, wurde ein KI-gesteuertes Shimming entwickelt, um das Shimming durch die Einbeziehung von Deep Learning (DL) zu beschleunigen, wobei die gesamte Pipeline mit Datenerfassung, Vorverarbeitung, Architekturdesign, Training und Einsatz abgedeckt wurde.

In der ersten Studie wurde gezeigt, wie ein Ensemble von neuronalen Faltungsnetzen (CNN) schnell drei lineare Shim-Ströme aus einer Menge von vier gemessenen Spektren vorhersagen konnte, wodurch die spektrale Qualität bei verschiedenen Substanzen erheblich verbessert wurde. Dieser Ansatz vereinfachte und beschleunigte das Shimming, entweder als eigenständige Methode oder in Verbindung mit herkömmlichen Ansätzen. In der zweiten Studie wurde das KI-gesteuerte Shimming weiter verbessert. Diese Studie nutzte eine zeitliche Historie durch rekurrente Verbindungen, und kombinierte Spektren und vergangene Shim-Aktionen für ein schnelles, quasi-iteratives Shimming auf vier Shims gleichzeitig. Der Ansatz führte auch eine effiziente Datenerfassung durch zufällige Datensatzerfassung ein, was eine Skalierbarkeit und eine erhebliche Verbesserung sowohl der Geschwindigkeit als auch der Leistung der Shimming-Algorithmen ermöglichte. Im Durchschnitt reduzierte diese Methode die Linienbreiten von 4 auf 0.72 Hz, was ihre Effizienz und Effektivität bei der Vermeidung lokaler Minima herkömmlicher Methoden unter Beweis stellt. Eine weitere Studie befasste sich mit den Herausforderungen der parallelen Detektion für einen hohen Durchsatz in der NMR-Spektroskopie, was eine erhebliche Chance für die Erweiterung der Dektoren innerhalb der Magnetbohrung bietet. Bei diesem Ansatz wurde ein maßgearbeiteter Messkopf entwickelt, welcher parallele Spektroskopie ermöglicht, und ein separates Shim-System über jedem Detektor zentriert. Deep Learning wurde eingesetzt, um deren sich überschneidenden, nicht orthogonalen Shim-Felder zu verarbeiten, was eine schnelle Kalibrierung der parallelen Detektoren ermöglichte. Schließlich wurde Deep Reinforcement Learning (DRL), ein Teilbereich der KI für strategische Entscheidungsfindung in Kombination mit DL, untersucht, um das Shimming zu verbessern, da es die Notwendigkeit einer Datensatzerfassung überflüssig macht und DRL direkt für ein Kriterium optimieren kann. Die ersten Experimente in Simulationen und auf realer Hardware haben gezeigt, dass DRL für Shimming funktionieren kann, aber weitere Herausforderungen mit sich bringt. Zusammenfassend lässt sich sagen, dass die künstliche Intelligenz ein leistungsfähiges Werkzeug zur Überwindung der traditionellen Herausforderungen beim Erreichen homogener Magnetfelder darstellt.

Ergänzend zu diesen Fortschritten in der NMR, wurde eine neue MRT-Technik, nämlich RASER, in Angriff genommen. RASER-MRT-Bilder werden (derzeit) entlang von Projektionen erfasst, aber Artefakte entstehen durch nichtlineare Wechselwirkungen zwischen den Bildprojektionen und verhindern qualitative Aussagen in medizinischen Anwendungen. Diese Artefakte wurden mit Hilfe eines DL-Ansatzes entfernt, der verzerrte Projektionen mit einem CNN rekonstruiert und die zusammengesetzten 2D-RASER-MRT-Bilder mit der U-Net-Architektur weiter entrauscht. Die Netzwerke wurden erfolgreich auf rein synthetischen und zufälligen Bilddaten trainiert und getestet, die einer RASER-Simulation unterzogen wurden. Die Modelle zeigten auch eine Generalisierung auf reale Messungen.

Zusammenfassend lässt sich sagen, dass die Integration von KI und DL in NMR-Shimming und RASER-MRT die verbesserten Möglichkeiten dieser Technologien demonstriert und den Weg für zukünftige Innovationen im Bereich der magnetischen Resonanz ebnet.

Abstract

This thesis explores artificial intelligence (AI) methods, such as deep learning (DL), for two applications in nuclear magnetic resonance (NMR) spectroscopy and magnetic resonance imaging (MRI). Specifically, an AI-driven approach for enhancing the tedious and time-consuming shimming process in NMR spectroscopy was initiated. Additionally, the image quality of a new MRI technique, namely RASER (Radio-frequency Amplification by Stimulated emission of Radiation), was improved through DL-based artefact removal.

A highly homogeneous magnetic field, up to parts-per-billion (ppb), is crucial to achieve accurate NMR/MRI results. This can be achieved by using "shim coils" to compensate for magnetic field gradients and make the field as uniform as possible, which is a time-consuming and tedious process. Thus, proper and fast shimming is critical, which is hampered by the non-bijectivity between field distortions in the sample volume and one-dimensional NMR signals. As the primary contribution of this thesis, four studies have been developed to accelerate shimming by incorporating deep learning (DL). These AI-driven shimming studies cover the full DL pipeline from data acquisition, preprocessing, architecture design, training, and deployment.

The first study demonstrated how an ensemble of convolutional neural network (CNN) could rapidly predict three linear shim currents from a batch of four measured spectra, significantly improving spectral quality across various substances. This approach simplified and accelerated shimming, either as a standalone method or in conjunction with traditional methods. The second study further enhanced AI-driven shimming. It employed a temporal history through recurrent connections, combining spectra and past shim actions for fast, quasi-iterative shimming on four shims simultaneously. The approach also introduced efficient data collection through randomized dataset acquisition, allowing scalability and significant enhancements in both the speed and performance of shimming algorithms. On average, this method reduced linewidths from 4 to 0.72 Hz, demonstrating its efficiency and effectiveness in avoiding the local minima of traditional methods. A further study addressed the challenges in parallel detection for high-throughput NMR, presenting a substantial opportunity in upscaling detection sites within the magnet bore. This approach introduced a custom probehead design allowing for parallel spectroscopy, and centred a separate shim system over each detector. Deep learning was employed to handle their overlapping, non-orthogonal shimming fields, which enabled rapid calibration of parallel detectors. Finally, deep reinforcement learning (DRL), a subfield in AI for strategic decision-making combined with DL, was studied to improve shimming. Using DRL can eliminate the need for the acquisition of a dataset, and can optimize for a criterion directly. The first experiments in simulation and on real hardware revealed that DRL can work for shimming, but inherit further challenges. In summary, AI introduces a powerful tool to overcome traditional challenges in achieving homogeneous magnetic fields.

Complementing these advancements in NMR, a novel MRI technique, namely RASER, was tackled. RASER MRI images are (currently) acquired along projections, but artefacts arise from non-linear interactions among the image projections and prevent qualitative statements in medical applications. These artefacts were removed by leveraging a DL approach dedicated to reconstructing distorted projections with a CNN, and further denoising the assembled 2D RASER-MRI images with the U-Net architecture. The networks were successfully trained and tested on pure synthetic and random image data that underwent a RASER simulation. The models also demonstrated generalization to real measurements.

In conclusion, integrating AI and DL into NMR shimming and RASER MRI demonstrates these technologies' enhanced capabilities and paves the way for future innovations in the field of magnetic resonance.

Contents

Nomenclature

Abbreviations

Abbreviation	Meaning
ADC	Analog-to-Digital Converter
FID	Free-Induction Decay
FFT	Fast Fourier Transform
FWHM	Full Width at Half Maximum
LNA	Low-Noise Amplifier
MRI	Magnetic Resonance Imaging
NMR	Nuclear Magnetic Resonance
ppm	parts per million
ppb	parts per billion
RMS	Root-Mean-Square
RF	Radiofrequency
ROI	Region of Interest
SH	Spherical Harmonics
SNR	Signal-to-Noise Ratio
TE	Echo Time
TR	Repetition Time
RASER	Radio-frequency Amplification by Stimulated Emission of Radiation
RT	Radon Transform
TPI	Total Population Inversion
AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
FC	Fully Connected
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NN	Neural Network
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
AC	Actor-Critic
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
MDP	Markov Decision Process
PER	Prioritized Experience Replay
POMDP	Partially Observable MDP
RL	Reinforcement Learning
TD	Temporal Difference

Notations

Symbol	Meaning
a_t	Action (or shim offset) at time step t
A	Action space for reinforcement learning
B_0	Static magnetic field
$\mathcal{D}$	Dataset
$\mathrm{F}_\theta(\cdot)$	Neural network with weights θ
Γ	Pumping rate of parahydrogen
h_t	Hidden state of an LSTM
$\mathcal{L}$	Loss function
m	Number of parallel NC (NMR cells)
n	Number of shims
$\mathcal{N}(\mu, \sigma)$	Gaussian noise with mean μ and standard deviation σ
o_t	Observation at time step t
r	Random steps
p	Predictive steps
R	Reference spectrum
$\texttt{Ref}$	Reference shim values
R	Shim value range
s	systematic shim offset
S	State space for reinforcement learning
S_i	Shim field pattern
$\mathbf{S}$	Shim distortions ($\equiv \hat{\mathbf{y}}$)
t	Time step
T_1	spin-lattice relaxation time
T_2	spin-spin relaxation time
u	Unshimmed spectrum
$u(\cdot)$	Unshimmed spectrum as a function of shim offsets
W	Shim weightings
X, Y, Z	Shims (named after their orientation)
$\mathbf{x}$	Input to neural network
γ	Gyromagnetic ratio
$\mathbf{y}$	Target or label for input $\mathbf{x}$
$\hat{\mathbf{y}}$	Prediction of neural network ($\equiv \mathrm{S}$)
θ	Neural network weights or parameters
ω_L	Larmor frequency

Constants

Constant	Symbol	Value	Unit
Gyromagnetic ratio of hydrogen	γ_H	42.577×10^6	HzT^{-1}
Boltzmann's constant	k_B	1.381×10^{-23}	$\mathrm{J\,K}^{-1}$
Planck's constant	h	6.626×10^{-34}	J s
Speed of Light	c	3.00×10^8	$\mathrm{m\,s}^{-1}$

1. Introduction

1.1. Motivation

Nuclear magnetic resonance (NMR) spectroscopy and magnetic resonance imaging (MRI) are crucial techniques for non-destructive and non-invasive study of in vitro and in vivo samples.

NMR can effectively provide information about a sample's chemical composition or concentration, making it an indispensable method in chemistry, biology and medicine. Almost every chemistry lab or institute worldwide uses low- or high-field NMR magnets to study food quality or drug discovery, e.g. for COVID-19 vaccinations. However, NMR magnets are generally expensive and have complex hardware. Moreover, a magnetic field with extreme homogeneities up to parts per billion (ppb) is required for good measurements. The resonance frequency of the sample's spins is directly proportional to the magnetic field over the sample, so even the slightest changes in the magnetic field homogeneity result in various distortions of the NMR spectrum or the MRI image. The most common solution to this problem is to use additional "shim coils", which make the magnetic field homogeneous over the sample to the highest possible degree. Currents passing through these shim coils superimpose correction fields and help to homogenize the magnetic field by compensating its gradients. One of the challenges is correct and fast shimming of the magnet, i.e., finding the best set of shim currents. The algorithms developed to automate (especially) the signal-based procedure are "non-intelligent", slow and may get stuck in local optima. Local optima in the optimization landscape arise due to inter-dependencies between shim field patterns, and through the lack of bijectivity in mapping a 3D magnetic field profile to a 1D NMR signal. This can potentially render the whole process painstaking and time-consuming, i.e. shimming is still a problem in NMR.

Similarly, almost every clinic has an MRI magnet for diagnosing and identifying diseases in the human body. Immediate results are crucial both for early disease detection and patient convenience. One way to boost MRI measurements is through hyperpolarization, and especially a new MRI technique called RASER (Radio-frequency Amplification by Stimulated emission of Radiation) can be applied as a molecular contrast agent. However, RASER images contain artefacts. Thus, artefacts in RASER images (so far) prevent their use for medical purposes.

The challenges mentioned above can be addressed using well-known scientific paradigms, namely empirical, theoretical, and computational approaches. A fourth paradigm has emerged in recent years: data-intensive scientific discovery (Hey et al., 2009). It is driven by the availability of large datasets and the growing computing power needed to process them. This paradigm includes artificial intelligence approaches that are increasingly being applied in various domains, such as ChatGPT, utilizing state-of-the-art algorithms.

This thesis aims to tackle two large problems in NMR and MRI with the help of artificial intelligence methods, namely the shimming problem in NMR, and artefacts of RASER MRI images. Particularly, there is great potential for accelerating and improving the shimming procedure with the help of AI algorithms. For this purpose, different machine learning methods were investigated, linked and applied to the shimming problem, focusing on simpler ML methods according to Occam's razor theorem. By interpreting NMR spectra as 1D images, state-of-the-art deep learning (DL) approaches for computer vision can be adopted. Furthermore, deep learning has shown great success in artefact removal and denoising in computer vision tasks, and should be able to handle the non-linear behaviour of RASER signals.

These challenges have not been studied before, and thus, it was necessary to develop the whole DL pipeline from scratch.

In summary, this thesis has the following objectives:

1. **Development of an AI-driven shimming approach.** Data-wise, this includes evolving a suitable data sampling strategy and performing acquisition and preprocessing. On the AI side, a proper model needs to be designed, i.e. conceiving DL architectures depending on the inputs' shape, which needs to be trained on the collected dataset. Finally, tests on real hardware need to prove the advantages of AI for shimming compared to traditional methods. The process should be repeated to eliminate the limitations of each preceding approach, while scaling to broader applicability.

2. **Artifact removal for RASER MRI images.** A simulation of the RASER effect must be created and used to generate a dataset of images with RASER-specific distortions and artefacts. Proper deep learning architectures should be carefully selected and tested to assess their ability to improve the resolution of 2D RASER images.

1.2. Thesis structure

The thesis is structured in 10 chapters, and consists of two main parts. After the introduction in chapter one, chapter two describes some fundamentals of NMR and AI. Part I of this thesis covers chapters 3 to 9 and focuses on AI-driven shimming, while describing general considerations in chapter 3. Chapters 4, 5, 6, and 7 describe several use cases and improvements. Chapter 8 summarizes several ablation studies, and chapter 9 discusses AI-driven shimming in general. Part II of this thesis includes chapter 10 and introduces deep learning for predicting RASER MRI profiles. Finally, the thesis concludes in chapter 11 with a brief summary and an outlook for future work.

1.3. Main contributions

The research conducted within this PhD project has led to three publications as leading first author, one with shared first authorship, and one with co-authorship.

- **Deep regression with ensembles enables fast, first-order shimming in low-field NMR**
  Published in *Journal of Magnetic Resonance*, 2022 (Becker et al., 2022a).
  Authorship: Lead author.
  Own contributions: This study showcases the first proof-of-concept study of accelerating the shimming process in nuclear magnetic resonance (NMR) through deep learning (DL). The paper demonstrates that DL can efficiently predict shim currents of three linear shims by correlating measured spectral shapes with shim current specifications, based on just four input spectra. It introduces a database for developing shimming algorithms, which helps understand the impact of shim offsets on ^{1}H NMR signals.

- **Acquisitions with random shim values enhance AI-driven NMR shimming**
  Published in *Journal of Magnetic Resonance*, 2022 (Becker et al., 2022b).
  Authorship: Lead author.
  Own contributions: This paper enhances AI-driven NMR shimming by using acquisitions with random shim values. The approach significantly improves the speed and performance of shimming algorithms, reducing linewidths in a low-field benchtop magnet from around 4 Hz to below 1 Hz in less than 10 NMR acquisitions in 87 out of 100 random distortions. By combining AI-driven with traditional shimming approaches, AI-assisted shimming helps in 96% of the cases to avoid getting stuck in local minima in shim space.

- **Artificial intelligence-driven shimming for parallel high field nuclear magnetic resonance**
  Published in *Scientific Reports*, 2023 (Becker et al., 2023).
  Authorship: The first authorship was shared with Yen-Tse Cheng.
  Own contributions: This research addresses the challenge of achieving parallel NMR spectroscopy with custom hardware, and shimming such a setup. It employs and modifies AI-driven shimming, enabling rapid calibration of parallel detectors with local shim coils in high-throughput screening technologies like drug development. The method outperforms the theoretical requirements of conventional shimming methods by predicting close-to-optimal shim values for each channel with a few random acquisitions.

- **Deep learning corrects artefacts in RASER MRI profiles**
  In preparation.
  Authorship: Lead author.
  Own contributions: This study focuses on AI-driven correction of artefacts in RASER MRI images, which arise due to non-linear interactions among image projections. A convolutional neural network is used to correct 1D image slices, and a U-Net model denoises the corresponding reconstructed image. After AI correction, previously distorted and unrecognizable RASER images can be interpreted and demonstrated both in simulation and on real-world measurements.

- **Dynamic dielectrophoretic cell manipulation is enabled by an innovative electronics platform**
  Published in the journal *Biosensors and Bioelectronics: X*, 2022 (Julius et al., 2023).
  Authorship: Coauthor.
  Own contributions: This study reports a portable manipulation platform to position and immobilize cells with an array of symmetrically arranged electrodes. A deep learning model is developed to map the electrodes' voltage phases to trap the positions of the cells.

2. Fundamentals and related work

Within this chapter, the fundamental concepts for the presented thesis are introduced. Section 2.1 will introduce the principles of nuclear magnetism, followed by nuclear magnetic resonance (NMR) and magnetic resonance imaging (MRI), and concludes with the essential hardware components of such devices. Shimming for NMR spectroscopy, and algorithms used for it, are introduced and described in section 2.2. In section 2.3, an overview of artificial intelligence (AI) methodologies in the area of NMR and MRI will be given, followed by the basics of deep learning (DL) and reinforcement learning (RL).

The topics in this chapter are mostly derived from the following excellent textbooks, and specific sources are given where appropriate:

- **Magnetic resonance**
  - Spin Dynamics by Malcom Levitt (Levitt, 2008).
  - Understanding NMR Spectroscopy by James Keeler (Keeler James, 2006).
  - Shimming for High-Resolution NMR Spectroscopy by Markus Weiger (Weiger & Speck, 2011).
- **Artificial intelligence**
  - Deep Learning by Ian Goodfellow et al. (Goodfellow et al., 2016).
  - Reinforcement Learning: An Introduction by Richard Sutton and Andrew G. Barto (Sutton & Barto, 2018).

2.1. Nuclear magnetic resonance

The nuclear magnetic resonance principle is the basis for highly sensitive, versatile, and non-destructive techniques such as nuclear magnetic resonance spectroscopy and magnetic resonance imaging, widely used in various fields such as chemistry and medicine. It can provide valuable insights into the structure and function of complex biological systems, including proteins, enzymes, small molecules, or even human body parts in the case of MRI.

2.1.1. Nuclear magnetism

All nucleons in matter have the following intrinsic properties: mass, charge, spin angular momentum ($\vec{S}$), and magnetic moment ($\vec{\mu}$). While the mass affects physical properties, and the electrical charge affects the chemical properties of the nucleus, the spin angular momentum and magnetic moment are interconnected but less evident. Magnetism and spin barely affect chemical and physical properties, but measuring them provides insight into the microscopic structure of samples.

Each sub-atomic particle (electron, proton, neutron) exhibits (net)[1] spin 1/2, and based on their combination, each isotope has a certain total **spin number** (I). The **spin angular momentum** vector $\vec{S}$ describes the total angular momentum of particles:

$$|\vec{S}| = \hbar\sqrt{I(I+1)}, \tag{2.1}$$

where $\hbar$ is the Planck constant $h/2\pi$.

As a consequence of spin, a **magnetic moment** is generated:

$$\mu = \gamma\vec{S}, \tag{2.2}$$

where the gyromagnetic ratio γ is unique for each nucleus.

Because NMR spectroscopy predominantly measures samples with spin 1/2, the remaining thesis will refer exclusively to spins as spin-1/2 particles, such as hydrogen ^{1}H or carbon ^{13}C (which only has a natural abundance of 1.1%), which have two possible spin states following the formula[2] $2I + 1$. In an ensemble of spins and the absence of a magnetic field, the spin vectors are randomly distributed and cancel each other out, leaving no net magnetization (Figure 2.1a).

When placed in a magnetic field, all spins tend to align parallel (α) or anti-parallel (β) with the magnetic field to minimize their magnetic energy (Figure 2.1b,c). This effect is called **Zeeman splitting**, and the total energy difference ΔE between the potential energies of particles in each state (E_α, E_β) depends on the strength of the static magnetic field B_0:

$$\Delta E = E_\beta - E_\alpha = \Big(-(-\frac{1}{2})\gamma\hbar B_0\Big) - \Big(-(\frac{1}{2})\gamma\hbar B_0\Big) = \gamma\hbar B_0 \tag{2.3}$$

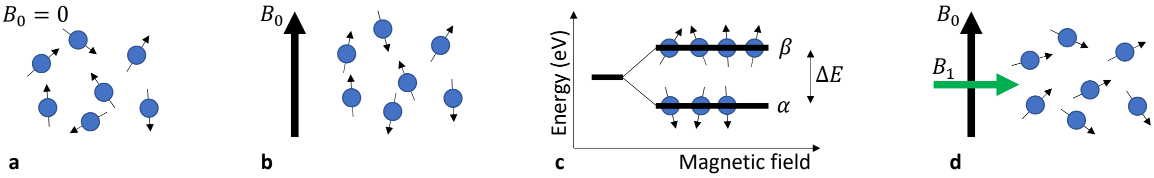


Figure 2.1.: (a) Nuclear spins in the absence of a magnetic field. (b) The spins align with the magnetic field and (c) split into two energy levels: the α and β state. At thermal equilibrium, more spins occupy the lower energy state α. (d) Spins during RF excitation.

In equilibrium, the ratio of splitting, i.e. spins oriented up N_{high} vs. oriented down N_{low}, is roughly 10 ppm (parts per million) for a temperature of $T = 300$ K, as given by the **Boltzmann distribution**:

[1] Given by total spins of quarks and charges.

[2] Note that as a general rule of thumb, even-mass isotopes are NMR silent.

$$\frac{N_{\text{high}}}{N_{\text{low}}} = e^{\frac{-\Delta E}{k_B T}} = e^{\frac{-\hbar\gamma B_0}{k_B T}} \approx 1 - \frac{\hbar\gamma B_0}{k_B T}\,, \tag{2.4}$$

where $k_B = 1.380649 \times 10^{-23}\,\mathrm{m}^2\,\mathrm{kg}\,\mathrm{s}^{-2}\,\mathrm{K}^{-1}$ is the Boltzmann constant. This splitting results in a net magnetization M_0 of the ensemble along the magnetic field axis (usually in the z-direction), also called longitudinal magnetization.

Hyperpolarization methods increase the polarization difference of nuclear spins, enhancing the NMR signal intensity. It is achieved by transferring the polarization from highly polarized states (e.g. para-hydrogen states) to the target molecules. The highly polarized states can be produced using techniques such as spin exchange, optical pumping, or parahydrogen-induced polarization (PHIP).

In analogy to Zeeman splitting of the magnetic moments, the spin vectors precess around the static magnetic field's axis. The frequency of precession, known as the **Larmor frequency**, is defined as:

$$\omega_L \left(= \frac{\Delta E}{\hbar}\right) = \gamma B_0 \tag{2.5}$$

The Larmor frequency is directly proportional to the field strength B_0 for each spin, and is probably the most crucial property for NMR, as the locally experienced magnetic field is distorted depending on the measured molecule's chemical shielding properties.

2.1.2. The path to a signal: Larmor precession

To get a magnetic resonance signal, spins (and their net magnetization M_0) are distorted from equilibrium by excitation with a **radiofrequency (RF) pulse** (Figure 2.1d).

Applying an RF-pulse that matches the frequency ω_L of the observed nuclei, the net magnetization M_0, still in equilibrium and thus full longitudinal magnetization ($M_0 = M_z$), will get flipped by angle α_{RF} towards the xy-plane, and yield some transversal magnetization M_{xy}. The flip angle is dependant on both B_1 and the pulse duration τ:

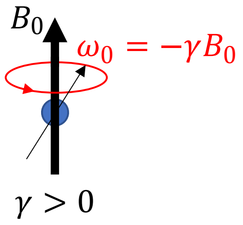
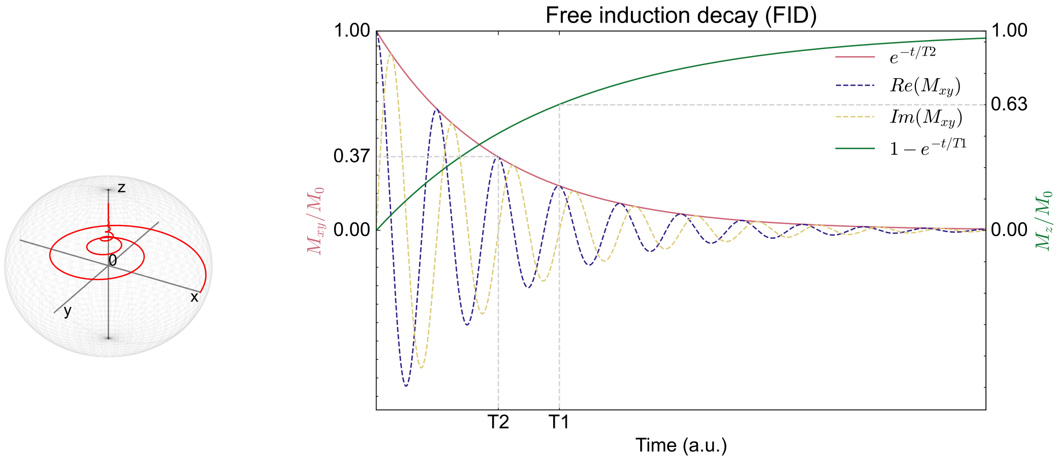


Figure 2.2.: Precession and Larmor frequency.

$$\alpha_{RF} = \gamma B_1 \tau\,, \tag{2.6}$$

where B_1 is the oscillating magnetic field with ω_L. For a 90° flip angle, M_z reaches zero and M_{xy} will be maximum ($M_{xy} = M_0$).

Turning off the RF-pulse, the spin ensemble (and thus net magnetization M_0) will "tilt", i.e., the planar components of M_0 start decreasing, while the longitudinal magnetization recovers back to align with the main magnetic field B_0 (Figure 2.3a). This effect is called



(a) Relaxation.

(b) A decaying FID signal and its connection to T_1 and T_2 times.

Figure 2.3.: Visualization of relaxation and Bloch's equations.

spin-lattice relaxation, measured as the time M_z takes to reach 63% of M_0, i.e. the spin-lattice relaxation time T_1. Meanwhile, slight variations in B_0 due to molecular structure cause the magnetic moments of each spin to accumulate slight frequency differences and phase incoherences during spin precession. This dephasing, in turn, leads to a transverse magnetization decay, characterized by the **spin-spin relaxation** time T_2 that represents the time M_{xy} takes to decay below $37\% M_0$.

According to Faraday's law, placing a coil close to the sample and perpendicular to B_0, the precessing nuclear transverse magnetization M_{xy} induces a voltage in the coil. The measured voltage will be exponentially decaying according to the decay of M_{xy} and will follow a sinusoidal curve with ω_L. This signal is also known as the **free-induction decay** (FID) signal (Figure 2.3b).

Bloch's equations approximate the evolution of net magnetization during and after the pulse:

$$\frac{d}{dt}\begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} 0 & \gamma B_z & -\gamma B_y \\ -\gamma B_y & 0 & \gamma B_x \\ \gamma B_y & -\gamma B_x & 0 \end{pmatrix} \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} - \begin{pmatrix} \frac{M_x}{T_2} \\ \frac{M_y}{T_2} \\ \frac{M_z - 1}{T_1} \end{pmatrix} \tag{2.7}$$

The FID signal is mathematically defined as complex-valued, where the real and imaginary parts conveniently contain M_x and M_y, respectively. Fourier-transforming the FID signal gives a complex-valued NMR spectrum of all frequencies in the sample. The line shapes of the real and imaginary Lorentzian peaks are called absorption and dispersion lines, respectively. Usually, phase correction is applied such that the imaginary or dispersion part of frequency space can be discarded, and the spectrum is purely absorption (see Figure 2.4). For both perfect hardware and a single isotope sample, the real part of the FID's FFT would yield a Lorentzian peak at ω_L with a linewidth measured as the "full width at half maximum" (FWHM):

$$\Delta v = \frac{1}{\pi T_2^*}, \tag{2.8}$$

where Δv is measured in Hz and T_2^* is T_2 including faster decay due to B_0 inhomogeneities.

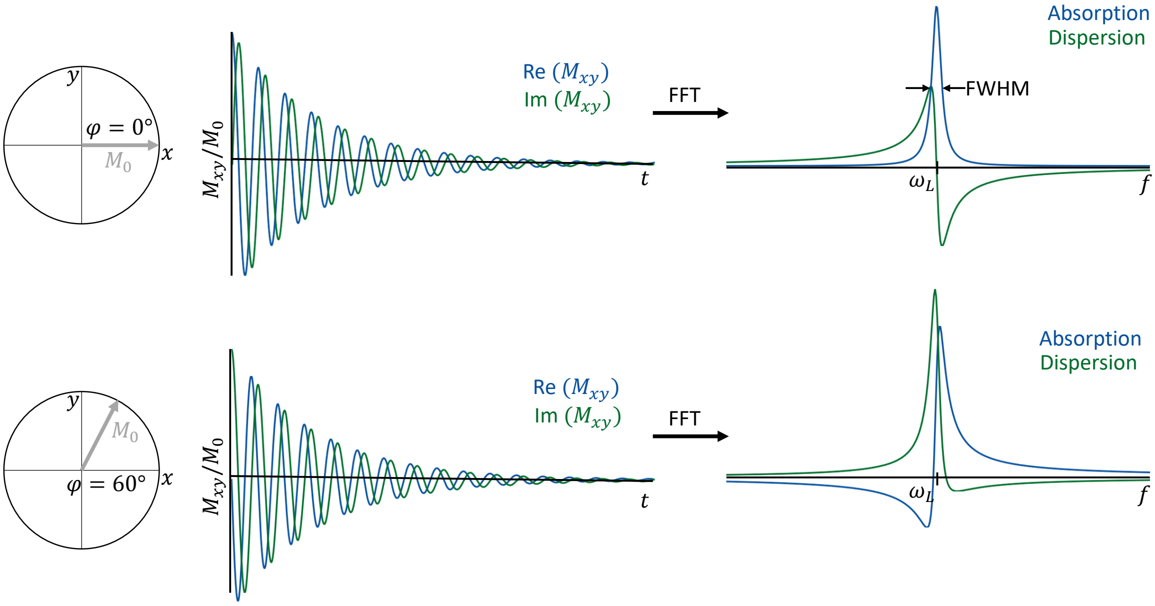


Figure 2.4.: **Influence of initial phase shifts on frequency domain line-shapes.** With no initial phase shift (top), the magnetization vector $M(0)$ yields an absorption line that does not require phase correction. Initial phase shifts (bottom) lead to peaks that are out of phase and necessitate phase correction.

2.1.3. Nuclear magnetic resonance (NMR) spectroscopy

Several non-idealities contribute to the measured NMR signal, causing the frequencies of the spectrum to differ from ω_L. These deviations can be separated into internal spin interactions, which contain the useful NMR information (fingerprint) due to spin couplings and local shieldings σ_i inside the measured molecule, and (unwanted) external influences. Internal spin interactions provide a unique fingerprint of the measured molecule, and can be seen as different frequencies and splittings in the NMR spectrum:

$$B_i = B_0(1 - \sigma_i) \tag{2.9}$$

Intra-molecular magnetic field differences are caused by:

Chemical shift Chemical shifts arise through local electron density differences between nuclei in a molecule, affected by factors such as electronegativity, bond lengths or neighbouring atoms. The electron distribution around a nucleus shields it from the external magnetic field, and thus affects its precession frequency. The chemical shift is measured in parts per

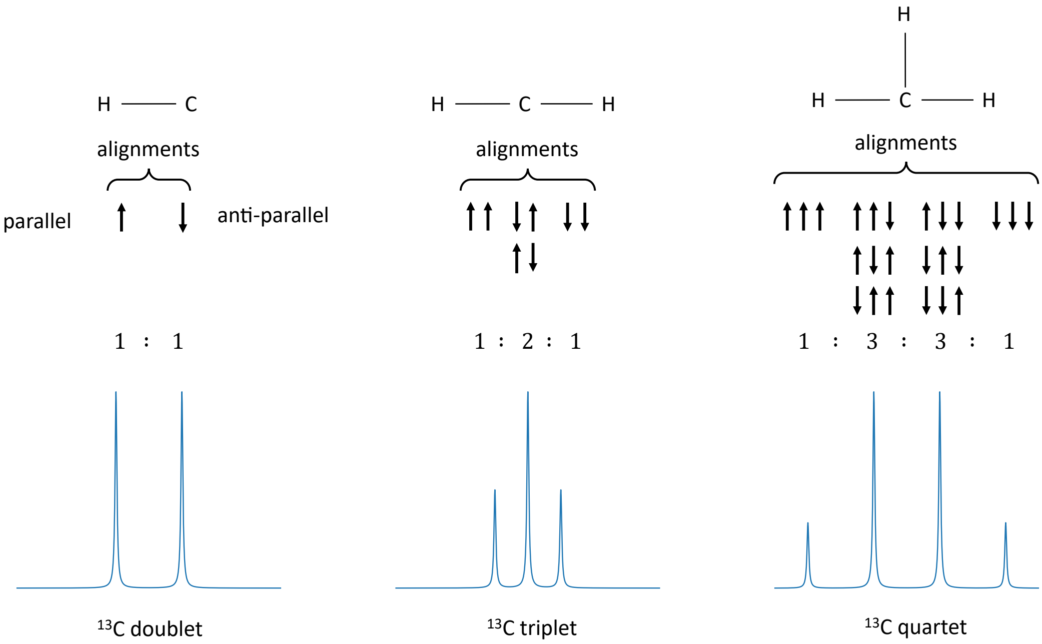


Figure 2.5.: **Multiplet patterns** in 13C NMR spectra arising from J-coupling, illustrating a doublet, triplet and quartet as interactions with one, two, or three ^{1}H protons.

million (ppm) relative to the resonance frequency $\nu_{\text{reference}}$ of a reference compound (e.g. Tetramethylsilane, TMS):

$$\delta = \frac{\nu_{\text{sample}} - \nu_{\text{reference}}}{\nu_{\text{reference}}} \tag{2.10}$$

The chemical shift strongly depends on the atom's electronegativity, where typical values for ^{1}H are usually between 0-10 ppm. The chemical shift allows to identify the types of atoms in a molecule.

Dipole-dipole couplings Indirect dipole-dipole couplings or J-couplings (or spin-spin couplings) describe indirect interactions between two nuclear spins through local electrons. J-couplings appear as a splitting of the resonance signal into multiple peaks (see Figure 2.5), with the spacing between the peaks corresponding to the J-coupling constant, where typical ^{1}H-^{1}H couplings are between 1-10 Hz (B_0 independent). This effect contributes to the distinctive NMR fingerprint of a molecule. (Direct) dipole-dipole coupling appears as the direct influence between neighbouring spins that are not directly bonded to each other. However, this coupling is effectively suppressed by the tumbling of molecules in an isotropic liquid and is thus negligible for the scope of this thesis.

External deviations mostly originate in hardware imperfections and include:

Noise The overall signal in NMR only comes from the difference in polarization (see Equation 2.4), thus, the signal intensity itself is already very low. However, additional noise sources, such as the thermal noise of the coils or the sample, contribute to the finally measured signal. Therefore, signal-to-noise ratio (SNR) is an important quality factor of every NMR spectrometer.

B1 distortions The RF-pulse exciting the spins for an NMR measurement is designed to be as uniform as possible. However, hardware imperfections induce non-uniform flip angles at different positions.

B0 inhomogeneities Fluctuations of the static magnetic field B_0 lead to every voxel of the sample experiencing a different field $\omega = \omega_L \cdot B_0(x, y, z)$ depending on its position in 3D space. If B_0 is inhomogeneous, individual voxel frequencies vary, which results in spectral line broadening and distortions. See section 2.2 for a more detailed description.

Generally, external deviations are stronger than internal effects, exceeding typical chemical shifts and j-coupling values. Therefore, shimming is required to enable the identification of those values.

2.1.4. Magnetic resonance imaging (MRI)

Magnetic resonance imaging (MRI) uses the principles of nuclear magnetism to produce monochromatic images of the internal structures of the human body without ionizing radiation. MRI can reveal information about the anatomy, physiology, metabolism, and pathology of various organs and tissues, as well as the brain's blood flow and oxygen consumption. MR imaging is fundamentally different from other imaging techniques that usually use "scattering" to obtain images (Callaghan, 1993). In MRI, however, the data comes from the chemical environments of each atom's spin. Thus, this section introduces the two main concepts to achieve MRI images based on magnetic resonance: the technique of spatial encoding for volumetric data, and the contrast mechanism of the monochromous images.

Spatial encoding

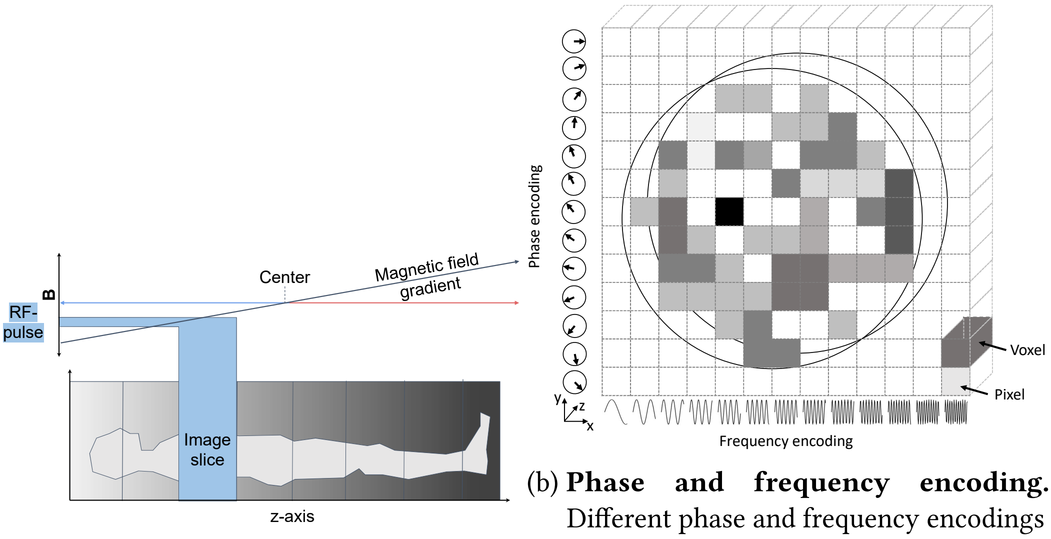
To encode three-dimensional MRI images, three distinguishable encodings must be applied.

Slice selection Linear gradient along (x, y, z)-directions allows for location-specific frequencies:

$$\omega_i = \gamma B_0 + G_x x + G_y y + G_z z\,, \tag{2.11}$$

where the proton frequency ω_i at position (x, y, z) varies with the gradient G.

Applying a linear z-gradient along the human body, and using a bandwidth-limited RF excitation pulse, only specific image slices along z are excited and lead to the selection of an imaging slice (Figure 2.6a). The stronger the gradient, the smaller the slice thickness.



(a) **Slice selection.** Figure modified from Arvidsson & Bertilson (2023).

(b) **Phase and frequency encoding.** Different phase and frequency encodings per voxel along two orthogonal directions allow distinguishable signals.

Figure 2.6.: Spatial encoding in MRI.

Frequency and phase encoding Frequency and phase modulations are used to encode the position in an image slice. Frequency encoding is done by applying an additional gradient along one image dimension (x-axis in Figure 2.6b), and the other dimension is modulated by different phases via a third phase-encoding gradient (PEG). PEG is applied step-wise so that in each repetition of the sequence, a different amplitude of the PEG gradient is used, and a single line of the k-space is acquired. This allows acquiring a two-dimensional image with pixels encoded by frequency and phase for a selected slice.

MRI contrast

The contrast in (proton) MRI mainly comes from the different chemical shifts of hydrogen bound to water or lipid molecules (Schild, 1990). The electronegative oxygen of water pulls the protective electron clouds covering the hydrogen nuclei, causing deshielding and a higher resonant frequency. Therefore, water has a larger spin-lattice relaxation T_1 time than fat.

The contrast of MR images can be adapted based on T_1 or T_2 weighting. T_1-weighting uses short echo and repetition times, leading tissues with shorter T_1 relaxation times (like fat) to appear brighter. T_2-weighted images use longer TE and TR values, and tissues with longer T_2 relaxation times (like water) appear brighter here.

The path to an image: Acquisition sequences and k-space

A typical imaging sequence consists of an RF excitation and three different encoding gradients to generate an MR image. The order of slice selection, phase encoding and frequency encoding can vary, leading to two common MRI sequences: spin-echo (SE) and gradient-echo (GRE) sequences. The SE sequence is generated by two successive RF pulses, one 90° pulse to excite the spins, followed by a 180° pulse, which rephases the spins and leads to an echo signal (see Figure 2.7). The duration between the excitation pulse's midpoint and the echo signal's peak is called the echo time (TE). Contrarily, the repetition time (TR) is the interval between successive excitation pulses used in different phase encoding steps. GRE sequences, on the other hand, use an excitation pulse and a bipolar (de- and rephasing) gradient instead of a 180° pulse to force an echo. GRE usually can have much shorter TE and TR and, thus, can produce images faster.

Repeated imaging sequences are used to fill the k-space, which defines an object in terms of spatial frequencies and is a complex matrix of size $M \times N$, where each row is the MR signal from one measurement. The transformation from k-space with (k_x, k_y) to image space (x, y) is carried out through the inverse Fourier Transform:

$$\text{Image}(x,y) = \mathcal{F}^{-1}[\text{k-space}(k_x,k_y)] = \iint \text{k-space}(k_x,k_y)e^{i2\pi(k_x x + k_y y)}\,dk_x\,dk_y \qquad (2.12)$$

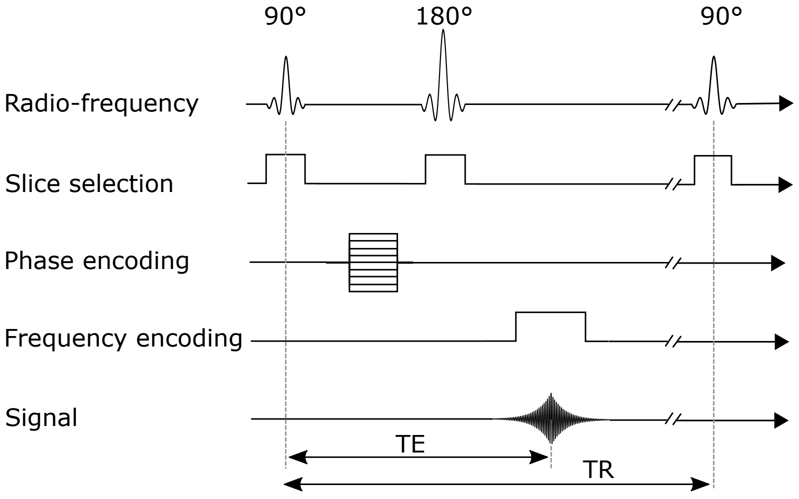


Figure 2.7.: Pulse diagram for a standard spin-echo imaging sequence.

2.1.5. New MRI method: RASER MRI

Recently, Radio-frequency Amplification by Stimulated emission of Radiation (RASER) has emerged to improve NMR and MRI resolution (Suefke et al., 2017; Lehmkuhl et al., 2022). It works by coupling a population inversion created through Signal Amplification by Reversible Exchange (SABRE) to a high-quality-factor resonator. Hyperpolarization is achieved by pumping parahydrogen, which exchanges its polarization with the target molecules, leading to a large total population inversion (TPI) of the spins, i.e. a higher Zeeman splitting. These hyperpolarized molecules then couple to the resonator, producing a sustained RASER signal derived from the scalar couplings of nuclei within the molecule, resulting in NMR spectra with sub-millihertz resolution. This means that RASER-MRI signals spontaneously emerge due to spin interactions without the need for a radiofrequency pulse (RF), which enhances the safety of the process. Since there is no RF pulse, the surrounding molecules do not contribute any signal to the images, which results in images without any background signal. Importantly, RASERs only emerge if a large enough initial total population inversion (TPI) ($d(0) = N_{\text{high}} - N_{\text{low}}$) is given, which should be above the RASER threshold

$$d_{\text{th}} = 4 \cdot \frac{V_s}{\mu_0 \hbar \gamma^2 T_2^* Q}, \tag{2.13}$$

where V_s is the sample volume, μ_0 is the vacuum permeability, and Q is the resonator's quality factor. The evolution of the measured signal is still governed by Bloch's equation, and extensively derived by Lehmkuhl et al. (2022).

One way to measure RASER MRI images is by means of projection reconstruction, which is widely used for computed tomography (CT), where 1D projections of an object are measured from different angles, and then Radon-transformed into an image. For RASER MRI, each 1D projection represents the TPI spread over the image domain.

However, RASER-MRI images frequently exhibit significant image artefacts due to the nonlinear behaviour among image slices.

2.1.6. Hardware

Typical NMR (and MRI) hardware's main components are the magnet, the RF coil, and the shim coils, depicted in Figure 2.8. For MRI, additional gradient coils are required.

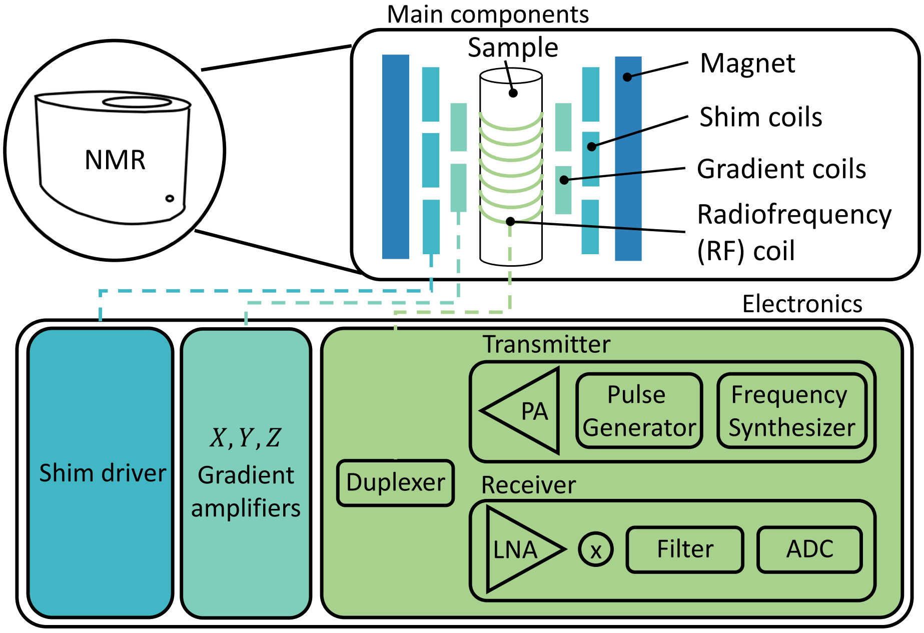


Figure 2.8.: **Magnetic resonance hardware**, divided into the main components and corresponding electronics.

Magnet

The magnet generates a strong and uniform magnetic field B_0 that aligns the nuclear spins of the sample, and increases Zeeman splitting depending on its strength. The magnets can either be permanent or superconducting. Permanent magnets usually produce low field ($< 2\,\mathrm{T}$), whereas superconducting magnets can achieve static magnetic fields up to 28 T, but they require liquid nitrogen or helium for cooling.

The main difficulty of building such magnets is their required homogeneity of parts per billion because the resolution of the acquired spectra strongly depends on a homogenized magnetic field. Traditionally, the magnet was controlled by placing small, local magnetic fields inside the bore to eliminate global field inhomogeneities. These local fields were produced by permanent magnets mostly made from iron, in the shape of thin wedges, so-called "shims"[3]. Nowadays, local magnetic fields can be generated by altering the current in electrical coils, the "shim coils", and the term "shimming" was adopted for modern technology from its iron ancestors.

Radiofrequency electronics

The radiofrequency (RF) electronics include a transmitter and receiver part (see Figure 2.8). The RF transmitter serves as the part to excite the spins with a temporary magnetic field B_1 perpendicular to B_0 such that the spin ensemble leaves its equilibrium state, and the RF receiver picks up the spins' relaxation back to B_0 to yield an FID.

The transmitter starts with an RF synthesizer, which produces an oscillating signal with ω_{Ref}, the spectrometer's reference frequency. The pulse generator is a fast switching gate to allow time slices of RF wave to pass through. Finally, the RF signal is amplified and sent to the RF coil.

The duplexer switches between transmitting and receiving and ensures (diverts) that high-power RF does not interfere with low-power read-out signals.

The receiver starts with a preamplifier of signal from the duplexer as picked up from the RF coil. Then, the quadrature receiver converts a high-oscillating signal around the Larmor frequency to a relative Larmor frequency $\Omega_L = \omega_L - \omega_{\mathrm{Ref}}$. This process is called frequency demodulation and filters out the high-frequency carrier wave, allowing the following analogue-to-digital (ADC) converter to work in a lower regime.

The precession of the signal ensemble along the x-y plane is a 2D movement, and thus leads to two RF output signals. This x-y position is then encoded into a complex-valued signal.

Gradient coils

Dedicated gradient coils are not available in every NMR spectrometer; however, they are crucial for MRI devices. Gradient coils produce high linear gradients G along the x, y and z axes to allow encoding of the signal's location.

For example, applying a gradient G_z during acquisition will lead to the resonance frequency varying linearly with the coordinate z and give a 1D projection image of the sample:

$$B(z) = B_0 + G_z \cdot z \tag{2.14}$$

[3] The English noun "shim" originally describes a thin piece of material that is inserted between two objects to level them.

Shim coils

Shim coils are electrical coils that are used to generate small, localized magnetic fields to adjust or "shim" inhomogeneities in the main magnetic field B_0.

Shim coil electronics They are controlled with a (high-resolution) digital-to-analogue (DAC) converter, followed by a stable and low-drift current source, which is required to produce precise shim field patterns that are time-invariant over the whole measurement time.

Shim coil design Usually, the coils are designed to produce/superimpose magnetic fields with a spatial variation governed by spherical harmonic (SH) functions. SH can represent 3D field deviations over a spherical volume, which aligns well with the 3D spatial nature of the magnetic field inhomogeneities encountered in MRI and NMR systems. Furthermore, SH is a basis set with orthogonal functions that can describe the magnetic field B_0 as an expansion of SH (Golay, 1958):

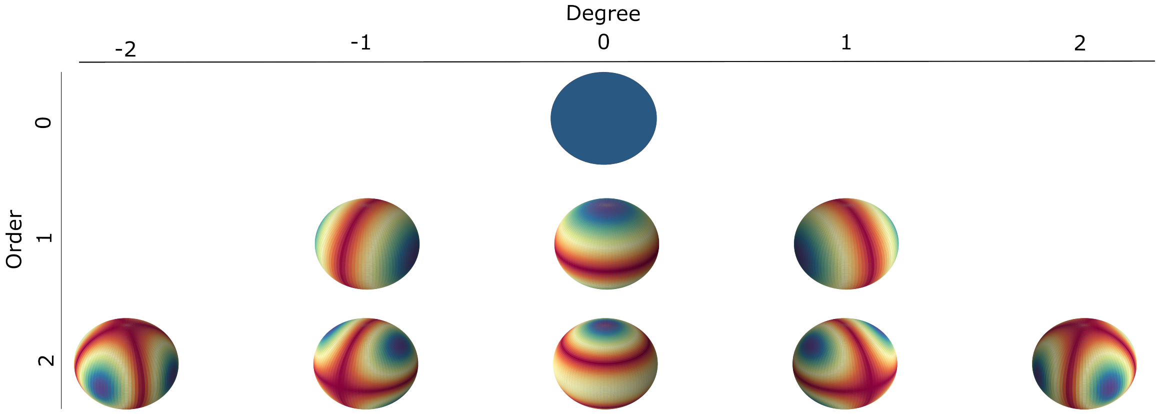


Figure 2.9.: Visualization of spherical harmonics, up to second order.

$$B_0 = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_{lm} Y_{lm}(\psi, \phi)\,, \tag{2.15}$$

where a_{lm} are the coefficients of the expansion, $Y_{lm}(\psi, \phi)$ are the spherical harmonic functions of degree l and order m, and ψ (polar) and ϕ (azimuthal) being the angular coordinates. In turn, the SH functions are defined as:

$$Y_{lm}(\psi, \phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} \cdot P_l^m(\cos(\psi)) \cdot e^{im\phi}\,, \tag{2.16}$$

where P_l^m is the associated Legendre polynomial.

Assuming that B_0 can be accurately described by SH functions, shim coils are designed to produce spherical harmonic correction fields. The design of shim coils is intricately based on Biot-Savart's Law:

$$B(r) = \frac{\mu_0}{4\pi} \int \frac{\mathbf{I} \times \hat{\mathbf{r}}}{r^2} \, d\mathbf{l}, \tag{2.17}$$

which dictates that the magnetic field $B(r)$ with position vector $\mathbf{r}$ depends on the current $\mathbf{I}$ and permeability μ_0. Through this principle, specific winding geometries of the shim coils are determined. For instance, Figure 2.10 displays saddle coils designed for different SH correction fields. Furthermore, Table 2.1 identifies the SH fields in Cartesian coordinates and their corresponding shim names.

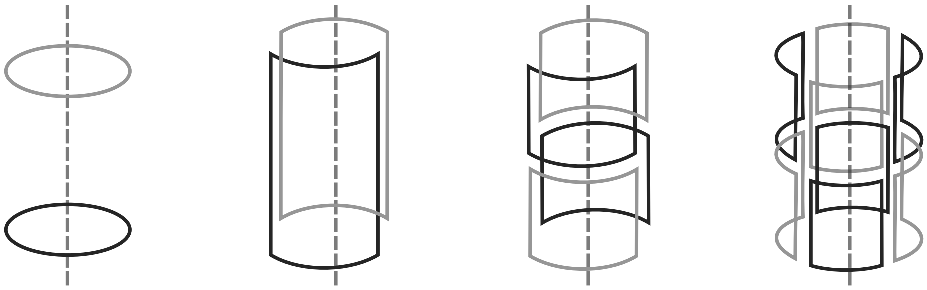


Figure 2.10.: **Saddle coil designs** to produce specific gradient fields along the (a) z-axis, (b), xz-axis, (c) (x or y)-axis, (d) (x^2-y^2)-axis.

Using shim coils is an essential step in preparing the main magnetic field for imaging and spectroscopy, and it allows for a more homogeneous magnetic field and a higher signal-to-noise ratio. However, the general assumption that the produced SH are orthogonal and independent usually does not hold in practice. Furthermore, the manufacturing of the shim coils introduces some shim impurities. Thus, the shimming process requires a lot of time and experience from the person performing the NMR measurement, and it could be tedious and painstaking (Chmurny & Hoult, 1990).

2.2. Shimming for NMR spectroscopy

Order	Degree	Common Name
0	0	Z^0
1	0	Z
1	1	X
1	1	Y
2	0	Z^2
2	1	XZ
2	1	YZ
2	2	$X^2 - Y^2$
2	2	$2XY$
3	0	Z^3

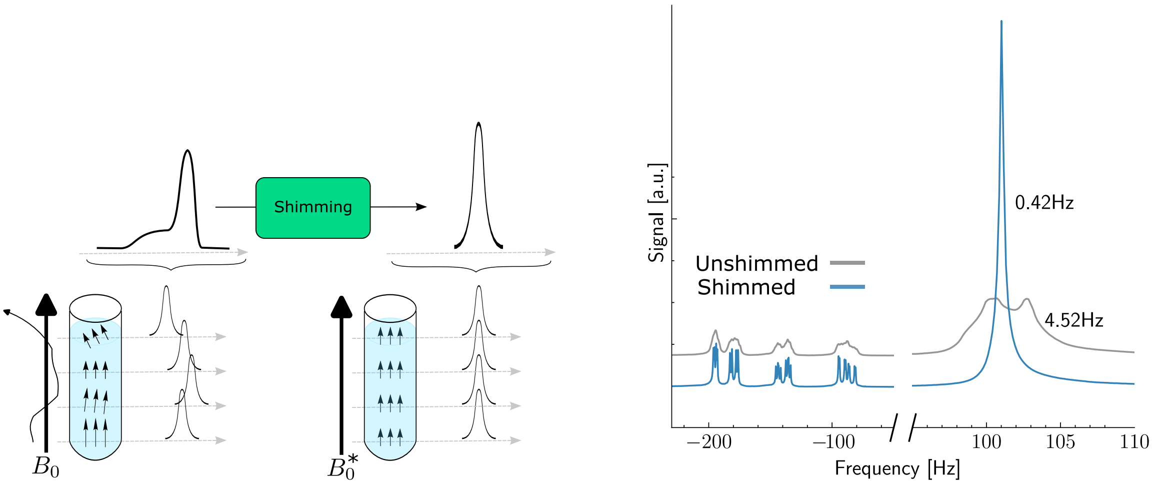
Table 2.1.: Common shim coil functions.

Hardware non-idealities due to manufacturing tolerances in wiring or material of the coil introduce inhomogeneities to the main magnetic field B_0. Any material placed in a (passively) optimized magnet will further distort the homogeneity, e.g. the coils, the probe, or the sample itself. The field can be homogenized by superimposing correction fields as produced by a finite set of shim coils, i.e. "shimmed".

To understand the effects of B_0 inhomogeneities, the three-dimensional inside of the bore (which holds the sample) can conceptually be broken down into small sub-volumes (voxels). With the number of sub-volumes approaching infinite, each voxel in three-dimensional space contains a single spin described by its Larmor frequency. The Larmor frequency directly depends on the local magnetic field: $\omega_L = \gamma B_0$. For a perfectly homogeneous magnetic field $B_0 = \text{const}$ at each point in space, each voxel has the same Larmor frequency, which results in a spectrum containing a single Lorentzian line with natural width centred at ω_L. The amplitude of this Lorentzian line is given by the sum of voxels contributing to the spectrum.

In an inhomogeneous magnetic field B_0, the sub-volumes experience different local fields, thus possessing frequencies shifted around a central Larmor frequency. The acquired spectrum can be described as a convolution or superposition of the Lorentzian lines of all voxels (see Figure 2.11a). This means inhomogeneities broaden the spectrum's lineshapes, decreasing their amplitude and consequently decreasing the signal-to-noise ratio (SNR).

For example, a practical drawback of inhomogeneities w.r.t. NMR spectroscopy can be described as follows: At 14.1 T, the proton scalar coupling lies between $4 - 7$ Hz. With a field disturbance of only 10 ppb (resulting in a linewidth broadening of 5 Hz), this coupling constant would be obfuscated and lost. In Figure 2.11b, it can be observed that neighbouring resonances cannot be distinguished if the linewidth becomes too large.[4]



(a) **Conceptual peak broadening** under an inhomogeneous field.

(b) **Empirical peak broadening.** Nicotinamide (vitamin B3) in HDO in a 1.8T benchtop NMR magnet measured in a homogenous B_0 (shimmed) and inhomogeneous B_0 (unshimmed).

Figure 2.11.: Effects of B_0 field inhomogeneity.

In a theoretical spherical volume, spatial mutually-orthogonal correction fields are generated by adjusting the currents w_i in a set of shim coils, producing the field profiles $\vec{S}_i$, $i \leq n \in \mathbb{N}$, to cancel the field inhomogeneities $\Delta B_0(\vec{r})$. In other words, a shimming algorithm must find the scalar weights $(w_1, w_2, ..., w_n)$ for the n shim currents that render

[4] The strict homogeneity requirements in the ppm range can be interpreted as a deviation of less than 1cm in a 1km road.

B_0^* as uniform as possible:

$$\Delta B_0(\vec{r}) = B_0^*(\vec{r}) - B_0(\vec{r}) \approx \sum_{i=1}^{n} w_i \vec{S}_i(\vec{r}). \tag{2.18}$$

2.2.1. Quality criteria

To evaluate the convergence of shimming algorithms, a quantitative or qualitative criterion is needed to judge the goodness of shimming. Criteria can mainly be differentiated based on the signal source they are measured on, namely either a field map or the FID signal (or spectrum via FFT of the FID).

Field map-based shimming is performed in a least-squared sense, making the need for another criterion redundant. Nevertheless, the homogeneity can be described qualitatively. One can distinguish between peak-to-peak homogeneity, measuring the maximum deviation within a field, and Volume-root-mean-square (VRMS) homogeneity, an "integral" specification and industry-wide standard for the specification of an empty magnet.

Spectral quality can be judged using several metrics:

- Spectral peak width, e.g., full width at half maximum (FWHM).
- Spectral peak height.
- Decay of FID (relaxation time T_2^*).
- Area Under Curve (AUC) of the FID.
- Shape and symmetry of spectral peaks.
- Envelope (Weiger et al., 2006) of spectral peaks.
- Shim quality quotient (Schlenke et al., 2013) by virtual peaks. (Suitable for multiple peaks.)
- Second moment of the spectrum (Ernst, 1968).

Furthermore, the tolerance of shimming depends on the targeted experiments. For example, some applications do not require very high field homogeneity, as measured by the linewidth at different percentages of the peak height. An exemplary overview is given in Table 2.2.

% of peak	Important for
50%	SNR (e.g. exchange & lifetime), Splittings
0.55%	Integral, Shape
0.11%	Water suppression, Selective excitation

Table 2.2.: Linewidth requirements for different applications.

2.2.2. Signal-based shimming

Without access to the shim field profiles S_i or the magnetic field B_0 to solve Equation 2.18, shimming must be performed using only a surrogate function $f(B_0(\vec{r}))$ that maps the 3D space onto a 1D signal, i.e. signal-based shimming on either FIDs or spectra. This imposes several challenges, including spectral line broadening effects and ambiguities between shims and their influence on the signal. For example, the signal caused by an inhomogeneity in the x-direction cannot be distinguished from an inhomogeneity in the y-direction, as the spectrum is just a histogram of all frequencies encountered in the sample volume.

The non-bijective mapping from 3D to 1D space without direct correlation to the shim values prevents using robust methods with gradient calculations w.r.t. an objective function.

Generally, shimming is an iterative process that optimizes a criterion of choice by adapting all shim currents until a satisfactory quality is achieved (see Figure 2.12a). Figure 2.12b and Figure 2.12c visualize the iterative shimming process for a simulated Lorentzian peak that is distorted by Z, Z^2, Z^4 shim functions. Gradually, higher order terms are corrected, which leads to improved lineshapes. The remanent inhomogeneities are plotted underneath.

1-D signal-based shimming

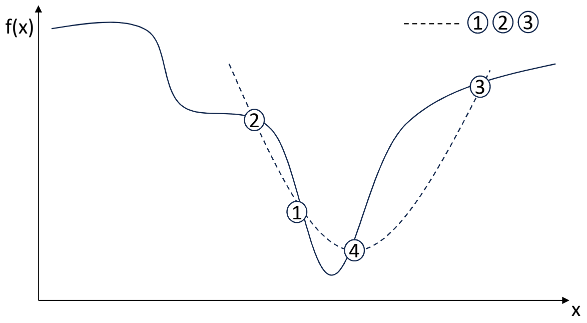
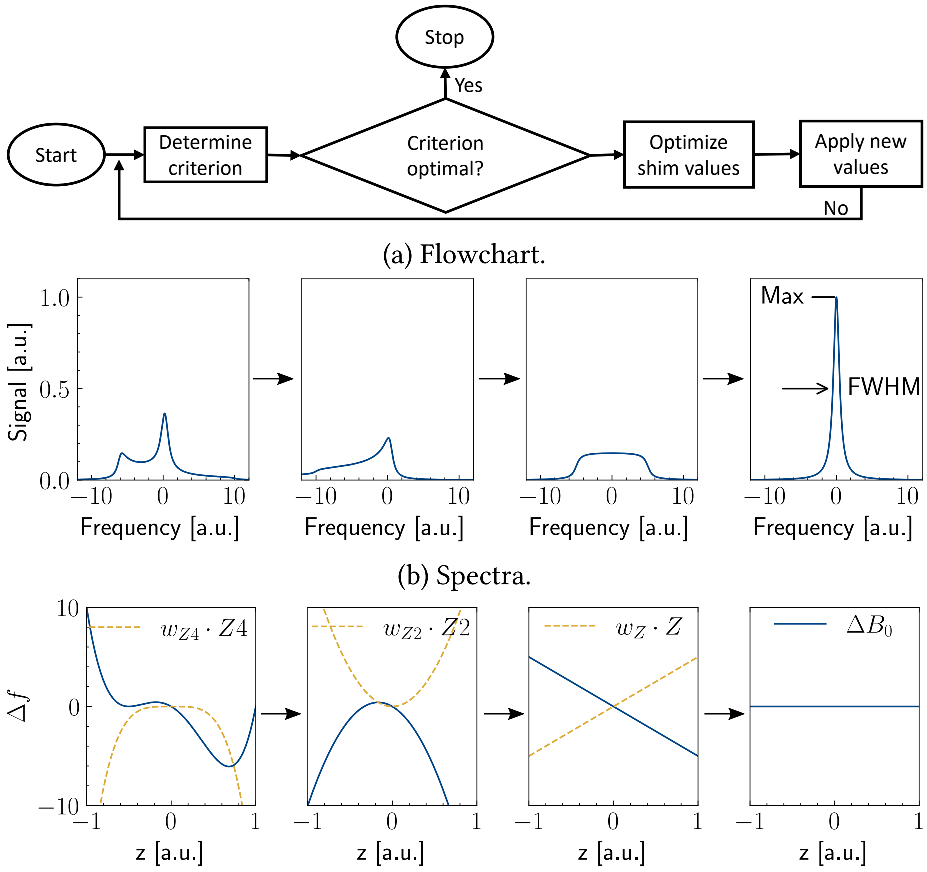


Figure 2.13.: **Parabola shimming.** x represents one shim current, and $f(x)$ is an inhomogeneity measure.

1-D optimization algorithms optimize a single variable at a time, and do not incorporate interdependences among variables. 1D algorithms for shimming include the Tuning (Hull, 2003), Coggins (Holz et al., 1988) or the general parabolic interpolation (Press, 2014) algorithms, which repeatedly compare three NMR acquisitions until the minimum quality criterion of choice can be approximated by fitting a parameterized parabolic curve. These methods are simple and fast, but often need to be iterated as the shim coils have parasitic terms of other coils.

The methods all share similarities to other well-known optimization algorithms, e.g. Coggins is a combination of Powell's method and Davies-Swann method, as stated by Ghani & Barnes (1972), and Brent's method (Press, 2014) also uses parabola interpolation.

The shimming method used on Spinsolve spectrometers (Magritek GmbH, Aachen Germany, `www.magritek.com`) is called QuickShim or parabola shimming. In fact, it shares the main characteristics of Hull (2003) and Holz et al. (1988): Finding the minimum of a Gaussian or parabola curve fitted to 3 points. The only difference is that it does not "hill climb" towards the minimum but adjusts the non-central shim values until the criterion



(a) Flowchart.

(b) Spectra.

(c) Remanent inhomogeneity.

Figure 2.12.: Flowchart and an example of iterative shimming. (Adapted from Becker et al. (2022a).)

differs more than 5% from the centre value. However, if the values are monotonously in- or decreasing, the algorithm moves sideways until a valley is found (i.e. same as in Tuning and Coggins). The optimized objective is the root-mean-square (RMS) of the FID.

Finding an optimal value for a single shim can be described with the following example:

1. Start at an initial point.

2. Increase step size until the function is more than 5% different from the centre value.

3. If all values are monotonously in- or decreasing, move sideways towards the valley.

4. Fit parabola to the last three smallest values (1, 2, 3).

5. Take parabola minimum (4) as optimum.

Figure 2.13 visualizes a parabolic fit through three points. The parabola has a superlinear convergence rate of 1.325, but is prone to getting stuck in local optima.

N-D signal-based shimming

Updating variables one at a time and sequentially leads to long iteration processes or not finding a solution at all. Thus, simultaneous adaption of the variables (i.e. shim currents) should be performed to compensate for the mentioned drawbacks. Optimization methods that optimize n variables at a time are called n-dimensional.

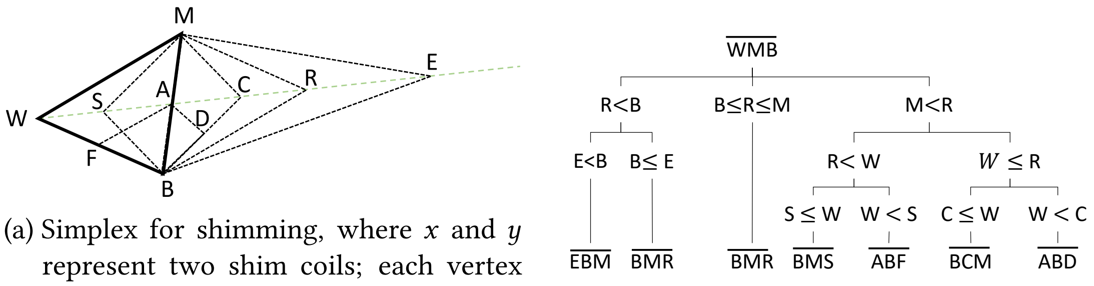
Downhill simplex method and modifications thereof The widely used method downhill simplex method (Nelder & Mead, 1965), has also been applied for NMR shimming (Ernst, 1968). The simplex method optimizes n shims by evolving a geometrical polytope (a "simplex") of $n + 1$ vertexes through the shim space by geometrical operations such as reflection, expansion, and contraction. Each vertex represents the quality criterion corresponding to specific shim settings, sorted based on the worst, average and best quality criterion to decide for the next evolution until a local minimum is reached.

For convenience, assume a two-dimensional scenario with two shims to adapt. The solution to the shimming problem lies in 2D space (i.e. coordinate system) and can be visualized (see Figure 2.14a). After defining an initial simplex[5] in the solution space, the quality criterion is evaluated at the vertices and ordered from best to worst (worst W, middle M, best B). The simplex then evolves in space by contracting, reflecting and expanding towards a solution, represented by the simplex's area being as small as possible. Extraction and expansion are done by inverting the worst point W w.r.t. the centre A of $B\bar{M}$, and the new simplex is calculated according to the scheme in Figure 2.14b. The default scaling factors for reflection (α), expansion (β), contraction (γ) and shrinkage (δ) can be defined as $[\alpha, \beta, \gamma, \delta] = 1, 2, 0.5, -0.5$ and yield the points R, E, C and S, respectively.

Ernst (1968) stated that the advantages of the Simplex algorithm are robustness in terms of convergence speed depending on the initial simplex definition, and efficiency (especially in comparison to the gradient method mentioned below). Nevertheless, the runtime is exponential in worst-case (Klee & Minty, 1972), and optimization can dwell in a local optimum depending on the starting point. Furthermore, Hull (2004) stated that the major limitation of the downhill simplex method is its slow convergence speed.

During the last few years, the Nelder-Mead Simplex method has experienced different refinements to improve either its convergence speed or accuracy. For instance, quasi-gradient methods are used instead of simple reflection (Pham et al., 2011), the shrinking coefficients are adapted depending on the dimensionality (Gao & Han, 2012) or a perturbed centroid (Fajfar et al., 2019) can be used instead of the midpoint between the best and worst criterion (B and W, resp.). Yao et al. (2021) combine all these improvements for an automatic shimming method and further include compensation of magnetic susceptibilities to strengthen their method.

[5] By definition, a simplex is the "most elementary geometric figure in Euclidean space of a given dimension"[https://www.collinsdictionary.com/de/worterbuch/englisch/simplex], e.g. in 2D represented by a triangle.



(a) Simplex for shimming, where x and y represent two shim coils; each vertex measures homogeneity.

(b) Simplex evolution scheme (Ernst, 1968).

Figure 2.14.: Nelder-Mead downhill simplex method.

Other approaches

Michal (2007) introduced a method to orthogonalize the shim coil gradients such that shimming can be reduced to one-dimensional optimization. This is done by calculating truly orthogonal "composite shims" through the Gram-Schmidt orthogonalization procedure. However, the current supply, heating, and other hardware components introduce non-linear behaviour, affecting the composite shims' symmetry. Thus, the composite shims themself are non-orthogonal again and require n-dimensional optimization or iterated 1D optimization.

Webb & Macovski (1991) have developed an FID-based shimming method for in-vivo MRS, which the authors themselves define as a "rapid, modified simplex"[6]. However, the inhomogeneities must be realizable by pure (and first-order only) gradients. With T_2^* being much shorter than T_2, it is guaranteed that the faster decay due to inhomogeneities can be observed. For each of the three linear shims, two additional measurements with offset $\pm\Delta G$ to the gradient field G are obtained to get a relation level of shim current changes.

The modified steepest descent method has been taken from Kiefer & Wolfowitz (1952); Sacks (1958) and adapted for magnet shimming by Ernst (1968). The procedure is as follows (Ernst, 1968): For each axis k (i.e. each shim), two auxiliary points X_n^{+k}, X_n^{-k} are placed symmetrically around a point X_n at the n-th iteration. Their corresponding quality criteria are denoted as Y_n^{+k}, Y_n^{-k} and span a vector of measured differences $\Delta Y_n = [Y_n^{+1}, Y_n^{-1}, ..., Y_n^{+k}, Y_n^{-k}]$. The next point X_{n+1} is then given by

$$X_{n+1} = X_n - \alpha_n(\Delta Y_n / 2\Delta_n), \tag{2.19}$$

where $2\Delta_n$ is the sequence of spacings of the auxiliary points and α_n is the sequence of positive step sizes. However, the simplex method seemed more efficient than this gradient method (Ernst, 1968).

The most recent publications to advance shimming propose using a genetic algorithm (Jang et al., 2023) or Bayesian optimization (Walter et al., 2023). Jang et al. utilize a genetic algorithm shimming in NMR/MRI superconducting magnets, which mimics the process of natural selection. This algorithm starts with a population of randomly generated solutions (shim settings), and in each iteration, evaluates these solutions based on the FWHM. The best-performing solutions are then selected, combined, and mutated to create

[6] In the author's humble opinion, the method shares few similarities with the downhill Simplex method.

a new generation of solutions. Over successive generations, the algorithm converges on an optimal or near-optimal set of shim settings that achieve the desired uniformity in the magnetic field. However, this method starts with an initial population of 50, and evolves roughly 9 generations, each requiring many NMR acquisitions to test the objective function.

Walter et al. discuss the use of Bayesian optimization for shimming in magnetic resonance-based dark matter search. Bayesian optimization (BO) sequentially samples an unknown target function to construct and improve a surrogate of it by placing a prior probability distribution over the function and then updating the prior with the data gathered through sampling to calculate the posterior probability distribution. The authors claim that BO could potentially be faster than the simplex method.

2.2.3. Gradient shimming

Gradient shimming is especially used for in-vivo MRI and requires strong gradient coils to map the magnetic field.

Conventional automated shim methods based on a measured signal (FID, spectrum) adapt to the iterative way a human operator would proceed. Prammer et al. (1988) identified the main drawbacks: First, due to the ambiguity of the 3D-to-1D mapping from sample to spectrum or FID), no spatial information about the field distortion can be inferred. Second, finding a global optimum is not guaranteed and third, the shim sets are, in reality, not ideally orthogonal and thus influence each other, requiring longer algorithm iterations until convergence. Consequently, Prammer et al. (1988) developed a non-iterative method for in-vivo MRI, which solved the shim value problem in a single step using spatial magnetic field maps, and the technique has been widely adapted and improved. The main procedure contains the following steps:

1. **Field mapping**: Acquire a mapping $B_0(x, y, z)$ of the sample, either as an image or projection using principles of MRI (gradient & phase encoding).

2. **Shim system calibration**: (Optional) Measure the true field shapes S_i of the basis functions for all n shims due to non-orthogonalities, e.g. measured on a default uniform phantom.

3. **Calculation of correction coefficients**: Calculate an optimal combination of the basis functions (correction fields) S_i to null the inhomogeneities of the acquired field map with an algorithm, i.e. minimizing a given criterion generally in a least-square sense.

4. **Application**: Apply shim values in a single step (or iterate). This includes the conversion of the predicted shim strengths to an actual shim coil current.

Despite the advantage of fully image-based techniques, the required time to get a field map of high resolution is still an obstacle. Gruetter & Boesch (1992) thus developed a method solving the three-dimensional, **in-vivo** shimming problem using only six exemplary, orthogonal and linear 1D-projections through the sample's centre. Since the introduction of this fast automatic shimming technique by mapping along projections (FASTMAP), several improvements (e.g. FAST(ER)MAP (Shen et al., 1997), FAST(EST)MAP (Gruetter & Tkáč, 2000)) have been proposed. Weiger et al. (2006) state that the main disadvantage of gradient shimming is that the goodness of the shimming outcome is only evaluated in the spatial domain, although a good quality of the frequency domain (spectrum) is the target of desire[7]. Therefore, they introduce a method that performs gradient shimming by **simulation** upon an initially acquired field map B_0 while simultaneously optimizing for an optimal quality criterion relying on a **calculated** spectrum. Finally, the best-simulated shim setting is realized, and the process is repeated at will. An experts' consensus on in-vivo shimming is given by Juchem et al. (2021) in 2020.

Like the previous automated shimming techniques, gradient shimming also involves limitations and pitfalls. The main disadvantages can be summarized as:

- Trade-off between the time required to obtain a well-resolved field map and the mapping resolution is limited by the volume of interest (VOI) and the gradient strengths. Therefore, B_0 maps may experience large steps among adjacent voxels.
- For large ΔTE or ΔB_0, phase correction must be carried out, such that the phase is in the range $[-\pi, \ldots, +\pi]$.
- Optimizing the mean squared error (MSE) of the field delta is no guarantee for a spectrum of good quality.
- An a priori basis set selection may miss field variations. The risk can be reduced with the calibration of shim functions.
- Thermal effects.
- 3D mapping requires a probehead with a three-axis gradient system and appropriate three-channel gradient current amplifiers (Weiger et al., 2006) that can be switched rapidly. (In principle, shim coils should be possible but must be pulsed rapidly. Stable power supplies of modern systems are not designed for pulsed operation. A possible solution could be ramping the coils.)

This thesis focussed on signal-based shimming, mainly due to the abovementioned reasons, especially the unavailability of gradient systems on low-field benchtop magnets.

2.2.4. General limitations of shimming

Shimming, whether done manually, signal-based or with field maps, has several limitations and pitfalls.

Hardware-wise, a prerequisite for successful shimming is a temporally stable environment, i.e. there should be no changes during the acquisition and correction of the magnetic field. This can be controlled by tracking the field lock and temperature of the bore. Furthermore, no ideal manufacturing of the coils and their material can be guaranteed,

[7] Still, gradient shimming solves the interaction among the shims to some extend and converge faster than solely iterative methods.

leading to, e.g. non-orthogonality of the shim functions. As a result, each shim coil produces parasitic fields of other coils. Additionally, the available current range depends on the power source and can be insufficient if the inhomogeneities are too large. The number of coils also restricts the order to which the magnetic field can be corrected; e.g. nine shims coils can cover only first- and second-order distortions.

On the software side, the shims' independence is often assumed, leading to inaccurate calculations/optimization. Most severely, the abovementioned methods have the possibility to get stuck in local minima.

However, some of these limitations are opportunities for an artificial intelligence or deep learning algorithm.

2.3. Artificial intelligence (AI), deep learning (DL) and reinforcement learning (RL)

This chapter describes the basics of **Deep Learning (DL)** and **Reinforcement Learning (RL)**, showcasing their distinct roles within the broader domain of **Artificial Intelligence (AI)**. While AI is an umbrella term for smart computational functionalities, **Machine Learning (ML)** narrows down to data-driven algorithmic learning. DL further specializes in deep neural networks, while RL is a different problem-solving paradigm focusing on maximizing rewards through interactions with an environment.

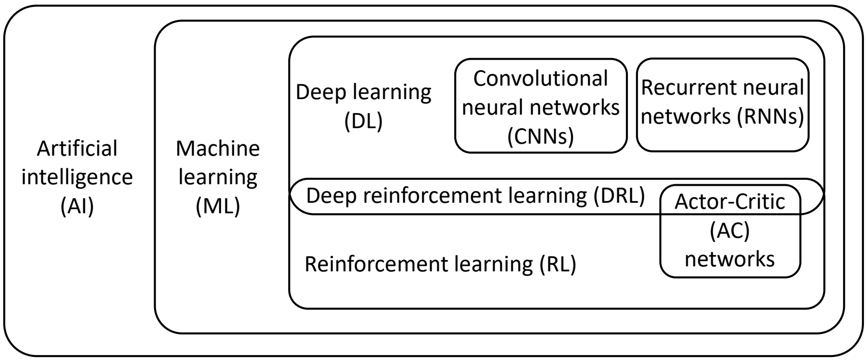

Artificial intelligence (AI)
Machine learning (ML)
Deep learning (DL)
Convolutional neural networks (CNNs)
Recurrent neural networks (RNNs)
Deep reinforcement learning (DRL)
Actor-Critic (AC) networks
Reinforcement learning (RL)

Figure 2.15.: Map of artificial intelligence, focussed on approaches used within this thesis.

A widely accepted definition of machine learning, which also holds for DL and RL, is given by Mitchel (1997):

> A **computer program** is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E.

Experience E is usually provided as a collection of data (usually a dataset), ideally independent and identically distributed (i.i.d. assumption). Regarding RL, the experience is collected through interactions with an environment (see Subsection 2.3.5). The amount of data samples, their quality, biases (Torralba & Efros, 2011), and the domain (Farahani et al.,

2021) of the data are important aspects that influence the success of machine learning models.

Machine learning **tasks T** can be differentiated (or split into two groups) depending on the learning nature and availability of labelled data. **Supervised** learning tasks include classification ($\mathbb{R}^n \rightarrow 1, ..., k$) and regression ($\mathbb{R}^n \rightarrow \mathbb{R}$), and require a label (or target value) for each data sample (usually provided by field experts). Examples of **unsupervised** learning tasks include clustering and dimensionality reduction, and handcrafted labels are not required. Furthermore, **discriminative** and **generative** algorithms can be differentiated, where the former learns decision boundaries or conditional probability $p(y|x)$, and the latter learns the data distribution $p(x, y)$, from which it can generate new data points. Finally, reinforcement learning focuses on making decisions by interacting with the environment to achieve a goal or maximize a reward.

Depending on the task, the machine learning model's **performance P** can be evaluated by metrics such as accuracy and precision for classification, L-norms such as the mean absolute error (MAE) or mean squared error (MSE) for regression problems, or task-specific metrics, e.g. rewards for RL approaches.

The last aspect of Mitchell's definition, namely the choice of the **computer program**, decides the affiliation to subgroups within AI, e.g. if a (deep) neural network is used to solve the task, one commonly refers to deep learning.

2.3.1. Benefits of AI

Why bother implementing complex AI systems instead of using standard methods? Several reasons show the advantages of such systems.

AI algorithms, especially deep learning models, can reconstruct high-quality images from under-sampled data, reducing scan times while maintaining or even improving image quality. Furthermore, AI can automate complex data analysis, identifying patterns and features that human observers might miss, consequently assisting in the diagnosis of diseases, which can lead to earlier and more precise treatments. Finally, AI can optimize scanning parameters, improving measurement efficiency.

Several approaches have been developed to enhance specific areas of NMR and medical imaging by incorporating AI algorithms, and examples are given below.

AI in (Proton) NMR

The utilization of AI in NMR has been a progressive field that aims to augment the existing capabilities of NMR. Key areas where AI has shown promising results are:

- **Designing NMR experiments** with AI can overcome the technical challenges of NMR spectrometers to enhance data acquisition. For example, an evolutionary algorithm designs novel radio frequency (RF) pulses for 2D biomolecular NMR experiments (Manu et al., 2023). Shimming has also been studied by incorporating genetic algorithms (Jang et al., 2023) or Bayesian optimization (Walter et al., 2023).

- **NMR processing** includes methods for the reconstruction of non-uniformly sampled (NUS) spectra (Hansen, 2019; Jahangiri et al., 2023) and truncated free induction decays (FIDs) (Lee et al., 2020). Enhancing the overall visual quality of NMR spectra can be achieved by peak picking (Li et al., 2021), deconvolution (Schmid et al., 2023) or restoration (Xiao et al., 2023). Extrapolation to higher spinning frequencies of magic angle spinning has been studied by Cordova et al. (2023).

- **Data analaysis** of NMR spectra covers chemical shift prediction (Liu et al., 2019; Williams & Jonas, 2023), or full structure prediction (Jonas, 2019) via imitation learning. Further approaches study spectra quantification (Rizzo et al., 2023) or tumour classification (Zhao et al., 2022) in MRS.

Chen et al. (2020); van de Sande et al. (2023); Shukla et al. (2023) summarize AI applications for solving NMR challenges. However, most of these approaches are applied at the post-processing stages after the NMR measurement has taken place, taking the hardware setup for the NMR measurement as granted. In contrast, section I of this thesis proposes to optimize the preparation preceding an NMR acquisition with deep learning methods.

These diverse applications of AI in NMR reflect a growing trend towards leveraging the 4th scientific paradigm of data-driven discovery to address complex challenges in NMR spectroscopy, from data analysis to experiment design, thereby expanding the possibilities of NMR technology.

AI for medical imaging

AI applications in clinical MRI (Magnetic Resonance Imaging) and CT (Computed Tomography) are revolutionizing medical diagnostics and treatment planning. These technologies, leveraging the latest AI advancements, are making significant strides in enhancing image quality, diagnostic accuracy, and patient care. A selection of key contributions includes:

- **Experimental design** can be improved for automatic slice alignment and cardiac shimming in clinical cardiac magnetic resonance imaging (MRI) (Edalati et al., 2022), or by ultrafast design of multidimensional RF pulses, intended for real-time pulse updates (Vinding et al., 2019).

- **Data processing** mainly cover denoising and segmentation of magnetic resonance images (Jiang et al., 2018; Akkus et al., 2017; Ronneberger et al., 2015). In computed tomography imaging, Shen et al. (2019) predict 3D patient data from one projection slice only, and Dong et al. (2019) reconstruct CT data from incomplete sinograms.

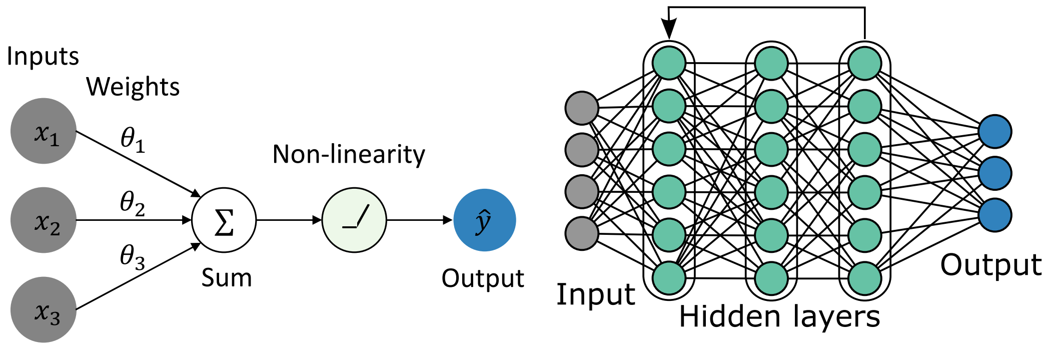
- **Data analysis**, for example, clinical diagnosis such as the diagnosis and prognosis of cancer (Huang et al., 2020).

Extensive and more detailed overviews are provided for MRI by Lundervold & Lundervold (2019); Bogner et al. (2021), and for CT, focusing on reconstruction, by Willemink & Noël (2019); Wang et al. (2020).

2.3.2. The basis of DL: Artificial neurons and multi-layer perceptrons

An **artificial neuron** is a simple processing unit and the fundamental building block of artificial neural networks. It applies a (non-linear) activation function to the sum of all its inputs[8]. This idea has been introduced as the McCulloch-Pitts neuron (McCulloch & Pitts, 1943), and refined as the perception (Rosenblatt, 1958), describing how neurons and neural networks remember and predict.

Neural networks incorporate multiple artificial neurons, usually organized into layers, including an input layer that takes in the data, hidden layers that extract features, and an output layer that produces a corresponding response (see Figure 2.16b). The standard feedforward network is a **multi-layer perceptron** (MLP), also called a **fully connected (FC) neural network**, because all neurons of one layer are connected to all neurons of the next layer with nonlinear activation functions.



(a) A basic artificial neuron.

(b) General deep neural network structure.

Figure 2.16.: The basis of artificial neural networks.

2.3.3. Training a deep neural network

A neural network "learns" by adapting its internal parameters or **weights** during training on a given dataset. The training process is guided through external variables, the so-called **hyperparameters**, determined outside the learning algorithm.

The learning principle

Given the (training) dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_i\}_{i=1}^{|\mathcal{D}|}$, where $(\mathbf{x}, \mathbf{y})_i$ is an input-output pair, a typical feedforward neural network F_θ is trained by (Figure 2.17):

1. **Forward pass**

$$\hat{\mathbf{y}} = \mathrm{F}_\theta(\mathbf{x}) \tag{2.20}$$

The inputs $\mathbf{x}$ are fed into the neural networks, and the activations of each layer are computed until the final output $\hat{\mathbf{y}}$.

[8] For a single neuron, and no activation function, the artificial neuron is just a simple linear regression function.

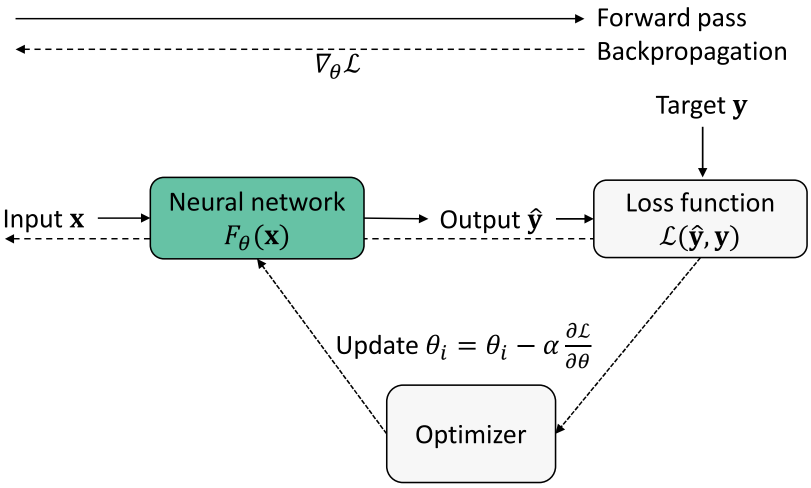

Forward pass
Backpropagation
$\nabla_\theta \mathcal{L}$
Target $\mathbf{y}$
Input $\mathbf{x}$
Neural network
$F_\theta(\mathbf{x})$
Output $\hat{\mathbf{y}}$
Loss function
$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$
Update $\theta_i = \theta_i - \alpha \frac{\partial \mathcal{L}}{\partial \theta}$
Optimizer

Figure 2.17.: **The learning principle of a neural network**, including forward pass, loss calculation, backpropagation and gradient descent.

2. **Loss calculation**

$$\mathcal{L}(\hat{\mathrm{y}}, \mathrm{y}) \tag{2.21}$$

The output $\hat{\mathrm{y}}$ is compared to the true label y of the input, and the loss $\mathcal{L}$ is calculated, which measures the difference between y and $\hat{\mathrm{y}}$ as a scalar error of the network.

3. **Backpropagation**

$$\nabla_\theta \mathcal{L} \tag{2.22}$$

The gradients of the loss w.r.t. the network's weights are computed using the chain rule of calculus through the network.

4. **(Stochastic) gradient descent**

$$\theta_i = \theta_i - \alpha \frac{\partial \mathcal{L}}{\partial \theta_i} \tag{2.23}$$

The weights are updated using the learning rate α in the direction that minimizes the loss function. Usually, momentum (Sutskever et al., 2013) or other optimizers (Kingma & Ba, 2014; Dozat, 2016) are used to speed up convergence (on mini-batches of the dataset).

5. **Iteration** Steps (1-4) are repeated until the network converges, i.e. the (non-convex) loss function is minimized on the training data to achieve the best function approximator.

Because neural networks possess nonconvex loss functions, training the network's parameters should avoid local minima and thus require regularization.

Regularization techniques

The **generalization** performance of a neural network is judged by the prediction performance on new, previously unseen (out-of-sample) data, also known as the **test set**. Therefore, **regularization** techniques are used to prevent the model from overfitting, i.e. fitting or memorizing the training data too closely, and improve its generalization performance.

To mitigate and avoid overfitting, the most commonly used techniques include:

- **Dropout**[9] is a technique that randomly sets some of the neuron outputs to zero during training. This prevents the neurons from co-adapting and forces them to learn more robust features that generalize better to new data.

- **Data Augmentation** pretends the availability of more data by applying random transformations such as rotations, translations, and scaling (depending on the data format) during training.

- **Early stopping** prevents overfitting by stopping the training process when the performance on the validation set starts to decay.

- **Batch and Layer Normalization** stabilize the training process and reduce the model's sensitivity to the initial weights and learning rate. Batch normalization normalizes the inputs across the entire batch, while layer normalization normalizes the inputs across each feature map or channel of the layer.

- **L1 and L2 regularization** add a penalty term to the loss function that forces the weights to be small. L1 regularization adds the absolute values of the weights, while L2 regularization adds the squared values of the weights.

It is important to note that proper regularization does not release from the necessity to tackle biases of the domain, data, application or method.

Activation functions

The activation function of an artificial neuron is applied to the output of each layer. It introduces nonlinearity into the model, which allows neural networks to learn complex and nonlinear relationships between the inputs and outputs.

Commonly used activation functions include:

- **Sigmoid** maps any input to $[0, 1]$ via:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.24}$$

.

[9] Dropout can be seen as a form of "cheap" bagging ensembles.

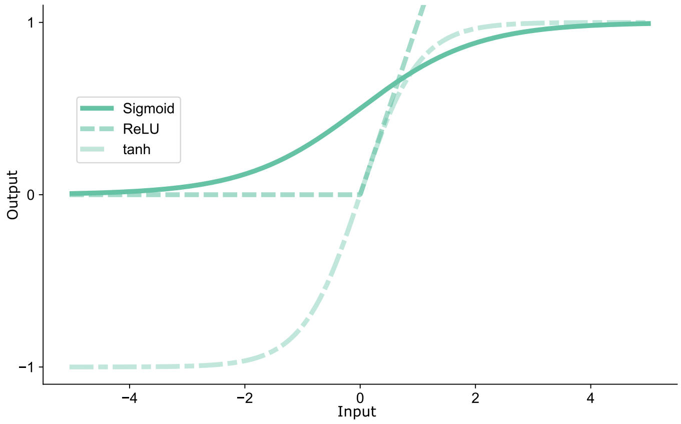


Figure 2.18.: Activation functions' output behaviour, visualized for ReLU, tanh and sigmoid.

- **ReLU** (Rectified Linear Unit) is defined as:

$$f(x) = max(0, x) \tag{2.25}$$

.

- **Tanh** (Hyperbolic Tangent) maps any input to $[-1, 1]$, and is defined as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.26}$$

.

- **Softmax** is commonly used as an activation function in the output layer of classification networks. It maps the output to a probability distribution over the classes via:

$$f(x_i) = \frac{e^{x_i}}{\sum e^{x_j}} \tag{2.27}$$

.

See Figure 2.18 for a visualization. The choice of the correct activation function is not straightforward and depends on the range of input data, and the target.

Hyperparameter optimization

Usually, DL training, and regularization methods used within, include additional tunable parameters (e.g. dropout rate or the extent of data augmentation), which all fall into the set of hyperparameters that must be optimized to train a well-performing neural network. Because it is not appropriate to optimize hyperparameters on the training set, or even worse on the test set, an additional **validation set** is needed to measure the influence of different hyperparameters. Thus, a proper set of hyperparameters prevents the model from overfitting, and helps it to generalize well on the test scenario it was trained for.

2.3.4. Neural network architectures

Architecture refers to the overall structure of the neural network, and allows the handling of different data structures, e.g. images or time series.

MLP and FC neural networks (see Subsection 2.3.2) are the most basic architecture without any assumption about the data format spatial or temporal dependencies. Given enough capacity, they are able to approximate any imaginable function.

Convolutional neural networks

Convolutional neural networks (CNNs) are a type of neural network that is specialized for processing data with fixed dimensionality (e.g., spectra, images and videos). CNNs use (multiple stacked) convolutional layers and pooling layers to extract features from the input and reduce the dimensionality of the data, and fully connected layers to perform a classification or regression task (see Figure 2.19b).

The idea behind convolutional layers is to use trainable filter kernels that sweep over a grid-like input to generate representations for the next layer, inspired by the visual cortex. This is in contrast to direct links used in fully connected layers. CNNs incorporate parameter sharing and sparse connectivity to decrease memory requirements and allow predictions independent of the features' locations (Goodfellow et al., 2016).

Local patterns, so-called feature maps, of the input are captured by moving multiple different filter kernels over the input and computing the dot product of the filter weights and the corresponding pixels (or values) of the input (see Figure 2.19a). Each filter typically covers a small receptive field, but the overall receptive field of the network is increased by stacking convolutional layers (and pooling and activation layers).

The output dimension of the feature maps depends on the following parameters, which need to be optimized depending on the task at hand:

$$\mathrm{dim}_{\mathrm{out}} = \frac{\mathrm{dim}_{\mathrm{in}} + 2 \cdot P - K}{S} + 1\,, \tag{2.28}$$

where P is the padding, S is the stride and K is the filter kernel size.

Pooling layers downsample the feature maps produced by the convolutional layers, reducing their spatial dimensionality.

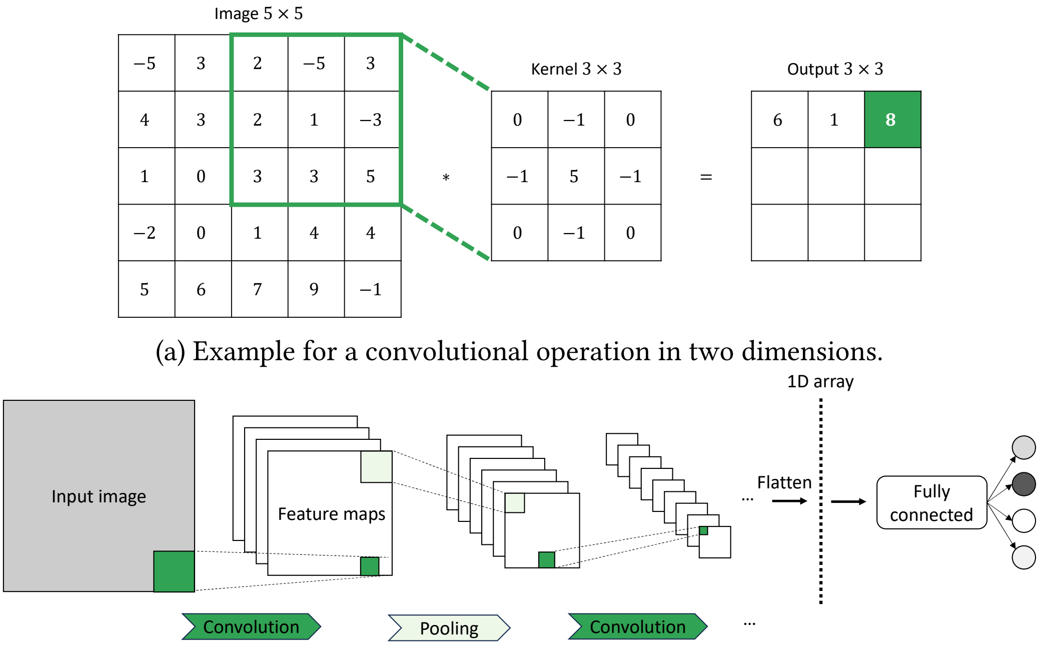
NMR spectra in the frequency domain can be interpreted as 1D-images (Qu et al., 2020), to apply CNNs and developments from computer vision.

Recurrent neural networks

In contrast to standard feedforward neural networks (such as FCNs and CNNs), which take an input $\mathbf{x}$, and provide an output $\hat{\mathbf{y}} = \mathrm{NN}(\mathbf{x})$, **recurrent neural networks (RNN)** sequentially maps a pair of input $\mathbf{x}_t$ and hidden state h_t to an output and a new hidden state:

$$\hat{\mathbf{y}}, h_{t+1} = \mathrm{RNN}(\mathbf{x}_t, h_t) \tag{2.29}$$

Here, the hidden state encodes the previously processed input, and acts as a cyclic memory to retain explicit information over several steps. Internally, RNNs contain three function



(a) Example for a convolutional operation in two dimensions.

(b) **Convolutional neural networks** with convolutional backbone and fully connected head.

Figure 2.19.: Deep learning with convolutions.

approximators V, U and W: U maps the inputs to features, V maps those features to the output, and W the features to the next hidden state: $h_t = \tanh(U \cdot \mathbf{x}_t + W \cdot h_{t-1})$ and $\hat{\mathbf{y}}_t = V \cdot h_t$. Standard forward and backward propagation algorithms can be utilized to learn the network's parameters by unfolding the repeated network function in time (Figure 2.20). RNNs are not restricted to fixed input or output dimensions and, thus, are more flexible than e.g. CNNs.

Long short-term memory (LSTM) networks are RNNs with special gate functions to forget unimportant information in the (input and hidden) sequences. Different flavours of LSTM gates, connections, and activations are discussed in Greff et al. (2017).

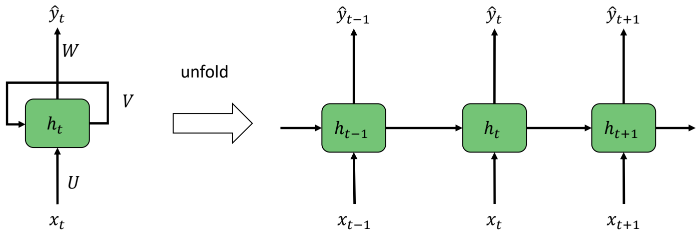


Figure 2.20.: **Recurrent neural network**, unfolded in time.

Encoder-decoder networks

Another special network architecture is the **encoder-decoder network**, which consists of an encoder that takes an input and encodes it into a useful representation, while the decoder takes this representation and decodes it into a transformed output.

An example of encoder-decoder networks includes the **autoencoder**. Autoencoders (Hinton & Salakhutdinov, 2006) (see Figure 2.21) rely on the general assumption that data shares a low-dimensional non-linear manifold, which can be exploited for dimensionality reduction. Thus, autoencoders primarily focus on input reconstruction and effectively learn to preserve essential features by using an encoder E to map the input $\mathbf{x}$ into a (usually lower-dimensional) representation, i.e. the latent space $h = E(\mathbf{x})$, and a decoder D to reconstruct $\hat{\mathbf{x}} = D(h)$. If the decoder and encoder of an autoencoder are linear functions, it is nearly principal component analysis (PCA), a standard ML method. Generally, the encoder and decoder may use different network layers, such as recurrent or convolutional layers. Encoder-decoder networks are often developed for supervised learning tasks to decode a target value $\mathbf{y}$, compared to autoencoders that copy their input $\mathbf{x}$. The U-Net architecture (Ronneberger et al., 2015), for example, was originally developed for semantic segmentation of biomedical images, and utilizes convolutional layers during encoding, deconvolutional (or transposed convolutional) layers during decoding, and skip connections between the encoder and the decoder to preserve spatial information and improve the localization accuracy of the segmentation. Transformers, described in the next section, can be configured in encoder-decoder structures.

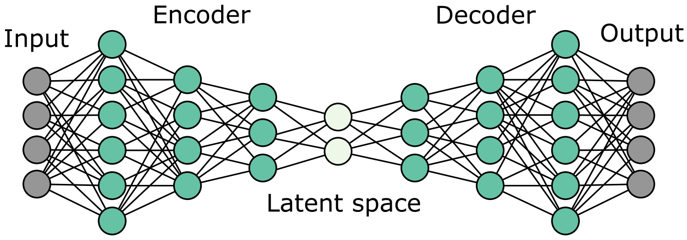


Figure 2.21.: Example of a fully-connected autoencoder architecture.

Attention layers and transformers

One of the major breakthroughs in recent years was the attention mechanism and transformers (Vaswani et al., 2017), which are basically just attention layers without recurrent connections. Transformer models are the basis of large language models such as ChatGPT. Unlike RNNs, Transformers simultaneously look at a full input sequence using the (self and multi-head) attention mechanism, instead of token by token.

Attention mechanisms bias the selection over values (sensory inputs). They achieve this by employing attention pooling, a technique that utilizes both intentional and unintentional cues in the form of queries and keys, respectively. The process involves mapping a query and a set of key-value pairs to an output, all represented as vectors. The output is then computed as a weighted sum of the values, where each weight is determined by a compatibility function of the query with the corresponding key (Vaswani et al., 2017).

Ensemble methods

Ensemble methods in machine learning combine multiple models to construct a more powerful model to achieve higher accuracy or lower variance in predictions (Breiman, 1996; Dietterich, 2000). In general, ensembles consist of two levels: multiple weak learners (level-0), and a combination of their predictions (level-1), often represented by a meta-model. Several forms, like bagging, boosting, or stacking (Dietterich, 1990), can be distinguished, and they differ in data handling or training of the different levels' models. Bagging (bootstrap aggregation) divides the training set into bootstrap replicates (subsets) of the original set, such that each weak learner is trained on such a subset (Breiman, 1996). Here, the level-0 models generally have the same structure. In comparison, stacking uses the same training set for heterogeneous weak learners, and the fusion strategy usually is not defined *a priori*, i.e., a meta-model is fitted (Dietterich, 1990). Boosting is used for sequentially improving a model on samples that the previous version got wrong (Schwenk & Bengio, 2000).

Overall, the variance of an ensemble method is reduced as follows. Assuming n independent models with a variance of σ^2, each predicting the same target variable, represented as random variables $X_1, X_2, ..., X_n$. Combining predictions by averaging, gives $Y = \frac{1}{n}\sum_{i=1}^{n} X_i 0$. The variance of the ensemble prediction $\text{Var}(Y)$ can be calculated using the properties of variance for independent random variables: $\text{Var}(Y) = \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}(X_i)$. Which, for X_i with the same variance σ^2, simplifies to: $\text{Var}(Y) = \frac{\sigma^2}{n}$. Thus, the variance of the ensemble's prediction should decrease inversely with the number of models in the ensemble.

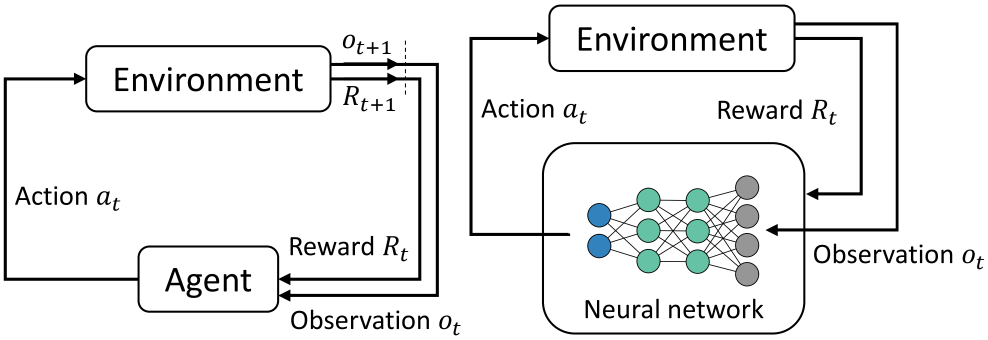
Ensembles can also be seen as some sort of regularization or used to quantify uncertainty (Lakshminarayanan et al., 2017).

2.3.5. (Deep) reinforcement learning

Reinforcement learning (RL) is based on the idea that agents learn to take actions in an environment to maximize a cumulative reward from their experience and feedback, rather than from explicit supervision or prior knowledge given by a fixed dataset. RL agents observe the consequences of their actions as states and rewards. RL aims to find a policy, i.e., a rule which maps states to actions, that maximizes the sum of discounted rewards over time.

One of RL's challenges is balancing **exploration** (trying new actions for potential future benefits) and **exploitation** (opting for the best-known action). Another challenge is dealing with large and complex state and action spaces, which make it difficult for the agent to learn. To overcome this challenge, **deep reinforcement learning (DRL)** combines RL with deep neural networks, which are powerful function approximators that can learn from high-dimensional and nonlinear data.

DRL has achieved remarkable results in various domains, such as playing Atari games, mastering the game of Go, and controlling robots.



(a) Reinforcement learning (RL). (b) Deep reinforcement learning (DRL).

Figure 2.22.: Differences between RL and DRL frameworks. DRL agents use deep neural networks as function approximators.

A standardized reinforcement learning description

In a standard RL setup, the environment provides the agent with a state s_t at each discrete time step $t = 0, 1, 2, ...$ and the agent responds by taking action a_t, causing the next reward r_{t+1} and next state s_{t+1} from the environment until the agent reaches a terminal state, i.e. the end of an episode. Whenever the environment is only partially observable, i.e. partially observable Markov decision processes (POMDP), a history of observation-action pairs $s_t = (o_1, a_1, ..., a_{t-1}, o_t)$ may be necessary to describe the state.

Mathematically, reinforcement learning may be characterized by a **Markov decision process (MDP)** with the state space S, action space A, the transition dynamics $P(s_{t+1}|s, a)$ and a reward function $R(s, a)$. The agent's action selection is given by a policy $a_t = \pi(o_t)$ defining the probability to perform an action for each observation, which ideally matches the real transition dynamics $\pi : S \longrightarrow P(A)$. The goal in RL is that the agent learns a policy which maximizes the sum of discounted future rewards (= total return)

$$G_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)\,, \tag{2.30}$$

where $\gamma \in [0, 1]$ is the discount factor which enables long-term correlations during one episode. The policy may be learned directly or via a proxy, e.g. the action-value $q_\pi(s, a)$ or state-action value function $v_\pi(s)$, which describes the expected return given state (and action). Many RL approaches make use of the Bellman equation to determine state-action values (q-values), and Bellman optimality is defined as:

$$q_*(s, a) = \sum_{s_{t+1}, r} p(s_{t+1}, r|s, a) \left(r + \gamma \cdot \max_{a_{t+1}} q_*(s_{t+1}, a_{t+1}) \right) \tag{2.31}$$

Under the premise of Q-learning (off-policy) this rewrites as $R_{t+1} + \gamma Q(S_{t+1}, a_{t+1})$.

The main problem in RL is to find an optimal policy π^* that maximizes the expected reward. Generally, RL algorithms can be trained with temporal difference (TD) learning from experience. TD combines Monte Carlo methods with dynamic programming, using bootstrapping to update its value function.

$$\underbrace{Q(s_t, a_t)}_{\text{new value}} \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \max_a \underbrace{Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s, a)}_{\text{old value}} \bigg)}^{\text{temporal difference}} \tag{2.32}$$

Taxonomy of RL

Classifying RL algorithm types strongly depends on the objective, and several grouping options are available, which, however, are non-exclusive. A helpful overview is given by Zhang & Yu (2020).

One can differentiate between RL agents with continuous or discrete action; also, the agent's training can learn either on- or off-policy, where off-policy learning learns an optimal policy π from suboptimal behaviour from another policy (i.e. exploration and exploitation are separated), whereas on-policy learning improves π by following π; or the agent can be trained online by interacting with the environment directly or offline by using a pre-sampled dataset (or experience).

Furthermore, one can differentiate between model-free and model-based RL. Model-free RL learns (1) a policy, (2) the value, or (3) both (see Paragraph 2.3.5) directly, whereas model-based RL learns a model of the environment and then uses planning to select actions. Value-based methods focus on estimating the value of states or state-action pairs, and popular algorithms include Q-learning and Deep Q-Networks (DQN). Policy-based methods aim to directly learn the optimal policy, which defines the agent's actions. Algorithms like REINFORCE and Trust Region Policy Optimization (TRPO) fall into this category. Actor-critic methods combine elements of both value-based (critic) and policy-based (actor) methods. This category includes algorithms like Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG).

However, there is not a single RL algorithm that can be considered "the one". Instead, several types and flavours of RL algorithms are designed to address different challenges and requirements. Therefore, it is important to carefully choose the appropriate algorithm for a particular RL task, and this thesis focuses on model-free actor-critic methods with continuous action space (of shim values), which are trained in an off-policy fashion.

Deep reinforcement learning

For large state spaces S or action spaces A, tracking all possible combinations of expected returns becomes impossible. DRL can handle high-dimensional and complex state and action spaces by using neural-network-based function approximators to model various agent components, such as policies π or values q, and learn from raw sensory inputs such as images or sounds.

Various flavours of DRL models can be differentiated based on the state and action dimensionality and/or continuity.

DQN A **Deep Q-Network (DQN)** is a deep learning model that learns control policies directly from high-dimensional sensory input using reinforcement learning (Mnih et al., 2013). A DQN is trained with a variant of Q-learning, where the input to a convolutional neural network are raw pixels of an image, and the output is a value function that estimates future rewards. DQN has been primarily limited to discrete action tasks, such as playing Atari games. Several improvements have been proposed and combined into one "rainbow" method (Hessel et al., 2017). Main modifications include prioritized experience replay (see below), multisteps, distributional Q-learning, noisy networks and duelling networks.

AC networks **Actor-critic models** combine the advantages of policy-based and value-based methods in reinforcement learning, and they consist of two components: an actor-network that learns the policy (the action selection) and a critic network that learns the value function (the expected future rewards). The actor-network takes the current state as input and predicts a probability distribution over the actions. On the other hand, the critic network estimates the value function, representing the expected future rewards given the current state and the predicted action from the actor.

DDPG **Deep Deterministic Policy Gradient** is a specific actor-critic algorithm designed for continuous action spaces (Lillicrap et al., 2015). DDPG uses an actor-critic architecture with two tricks to improve stability: a replay buffer to store and reuse past experiences (see the next paragraph), and target networks to avoid divergence during training. Target networks are copies of the original actor and critic networks updated sparsely with "soft" updates.

The pseudocode for internal DDPG updates is given below.

Several modifications and improvements have been made to the DDPG method/technique, e.g., delayed updates to twin networks for both actor and critic (TD3) (Fujimoto et al., 2018), or stochastic policies with entropy are used instead of deterministic ones in soft actor-critics (Haarnoja et al., 2018).

Other advances in DRL convergence and performance As the agent's interactions with the environment are generally expensive, observed transitions $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ (or o_t instead of s_t for POMDPs) can be stored in memory. Updating the policy might then also use previously observed transitions from a (static) dataset or memory $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1}, a_{t+1})_i\}_{i=1}^{|\mathcal{D}|}$. **Prioritized experience replay** (Schaul et al., 2015) is a technique that improves the efficiency of reinforcement learning agents by sampling replay transitions from such a memory with high expected learning progress more frequently. Not all experiences are equally useful for learning, e.g. similar random exploratory steps are more frequent in early training episodes. Prioritized experience replay assigns a priority value based on the magnitude of the temporal difference error to measure the usefulness of each experience. The higher the priority, the more likely the experience will be sampled for learning. As prioritization can lead to a loss of diversity, using stochastic sampling interpolates between pure greedy prioritization and uniform random sampling during experience replay.

Algorithm 1: Deep Deterministic Policy Gradient. Modified from Achiam (2018) and Lillicrap et al. (2015).

```
Randomly initialize actor $\mu_\theta$, and critic network $Q_\phi$ with parameters $\theta$, $\phi$
Set target parameters equal to main parameters $\theta_{\text{targ}} \longleftarrow \theta$, $\phi_{\text{targ}} \longleftarrow \phi$
Initialize replay buffer $\mathcal{D}$
for episode = 1, M do
    Initialize random noise process $\mathcal{N}$ for exploration
    Get initial observation $s_1$
    for t=1,T do
        Observe state $s$ and select action $a = \mu_\theta(s) + \mathcal{N}$
        Execute $a$, observe next state $s_{t+1}$, reward $r$, and done signal $d$
        Store $(s, a, r, s_{t+1}, d)$ in replay buffer $\mathcal{D}$
        if $s_{t+1}$ is terminal then
            Reset environment state
        if update interval reached then
            for updates do
                Randomly sample a batch of transitions $B = \{(s, a, r, s_{t+1}, d)\}$ from $\mathcal{D}$
                Compute targets $y(r, s_{t+1}, d) = r + \gamma(1 - d)Q_{\phi_{\text{targ}}}(s_{t+1}, \mu_{\theta_{\text{targ}}}(s_{t+1}))$
                Update critic by one step of gradient descent using
                  $\nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s_{t+1},d) \in B} \left(Q_\phi(s, a) - y(r, s_{t+1}, d)\right)^2$
                Update actor by one step of gradient ascent using
                  $\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\phi(s, \mu_\theta(s))$
                Update target networks with $\phi_{\text{targ}} \longleftarrow \tau\phi_{\text{targ}} + (1 - \tau)\phi$
                $\theta_{\text{targ}} \longleftarrow \tau\theta_{\text{targ}} + (1 - \tau)\theta$
```

Part I.

Artificial Intelligence for NMR shimming

Shimming is a cumbersome and time-consuming procedure preceding most NMR experiments. This part of the thesis describes several approaches that speed up shimming by incorporating artificial intelligence aspects by learning and leveraging features from distorted spectra during the shimming process.

First, the general problems of shimming are revisited in chapter 3, and why AI could learn from 1D spectra to speed up the process. Also, the general requirements to enable such a framework are described. Chapter 4 introduces the first feasibility study that used DL to enable fast, first-order shimming. In chapter 5, most limitations of this first proof-of-concept study were tackled, which led to an improved methodology that allows DL to shim four shims based on acquisitions from random shim offsets. This method was further utilized to tackle another problem when using parallel NMR channels for high throughput NMR: an increased number of interconnected shim coils. Chapter 6 describes the first study for AI-driven shimming of a custom-built parallel probehead with two NMR channels, each equipped with six local shim coils. Finally, chapter 7 introduces another paradigm of AI, namely reinforcement learning (RL), which does not require pre-labelled datasets and interacts with an environment to explore it. The foundations of using this approach for shimming were explored. Finally, chapter 8 describes several ablation studies that prove design choices made during the development of all the methods mentioned above. Part I of this thesis concludes with a discussion about AI-driven and AI-assisted shimming.

An overview of corresponding publications per chapter is given in Table 2.3. All published articles generally include a repository with the published code and the data (collected at mobecks/ShimDB) used, allowing reproducibility or easy improvements. Unfortunately, most publications in NMR research lack this standard.

Chapter	Method name	Publication	Github repository
4	Deep Regression with Ensembles (DRE)	Becker et al. (2022a)	mobecks/dre-nmr-shim
5	enhanced Deep Regression (eDR)	Becker et al. (2022b)	mobecks/random-ai-shim
6	Parallel eDR (PeDR)	Becker et al. (2023)	mobecks/parallel-ai-shim
7	RL-based	(In preparation.)	-

Table 2.3.: Overview of the developed methods for AI-driven shimming. All datasets for the corresponding publications are also published.

3. Why and what can DL learn from 1D spectra for shimming?

AI has just started gaining momentum in NMR (Chen et al., 2020; van de Sande et al., 2023; Shukla et al., 2023). However, most of these approaches take the hardware setup for granted and focus on post-processing methods, which, in turn, require shimming before the experiments to allow for a minimum signal quality.

In part I of the thesis, AI is introduced to speed up and improve shimming based on 1D spectra. The assumption that this is possible is based on two main concepts: (as theoretically proven and known) specific peak distortions help to identify the shim order, and (not exploited previously) hardware non-idealities introduce additional distinguishable features between shims. These effects are described in section 3.2 and section 3.3, followed by requirements to exploit those features with DL or RL methods in section 3.5.

Iterative signal-based shimming is the main focus of this thesis. The optimization process is based on a scalar quality criterion using signals such as the FID, lock channel, or spectral lineshapes. The approach allows to neglect the theoretical limitations of spatial functions and instead directly optimizes the shim currents.

3.1. Problems of shimming (revisited)

(Signal- based) shimming generally suffers from multiple problems that leave the process tedious and time-consuming. The main challenges summarize as:

- **Non-orthogonality** of the spherical harmonic shim system, mainly due to manufacturing inaccuracies of the coils.
- **Line broadening effects** due to susceptibility differences and B_0 inhomogeneities.
- **Large number of shim coils**.
- **Ambiguity problem**, originating from non-bijectivity between 3D sample volume and 1D signal.

Existing shimming algorithms automate the shimming procedure to increase the spectral quality. However, the following issues remain. Gradient shimming requires rapidly switchable gradient coils and can lead to distorted B_0 field maps (Bao et al., 2016), which makes the process iterative. Signal-based shimming methods, which are iterative by design, also suffer from local optima, especially due to the non-orthogonality of shims. Robust against orthogonality issues are n-dimensional optimization algorithms such as the simplex method (Ernst, 1968; Yao et al., 2021); however, they converge slowly and leave shimming to be time-consuming.

When considering new shimming approaches, all available (visible and hidden) features of the unshimmed NMR spectra should be exploited, instead of only a single optimization scalar, to extract more knowledge during the shimming process. This starts with disambiguating the shims, incorporating their non-orthogonality, and consequently, speeding up shimming.

Chmurny & Hoult (1990) already gave guidelines for manual shimming in 1990, and called it an "art" to interpret the distortion effects of shims on the FID and spectrum. However, a human expert operator must incorporate this knowledge for shimming.

3.2. Line shapes reveal shims' order

The first step in solving the ambiguity between shims is to differentiate the order of shim distortion from the peak distortions of a single peak in ^{1}H spectra.

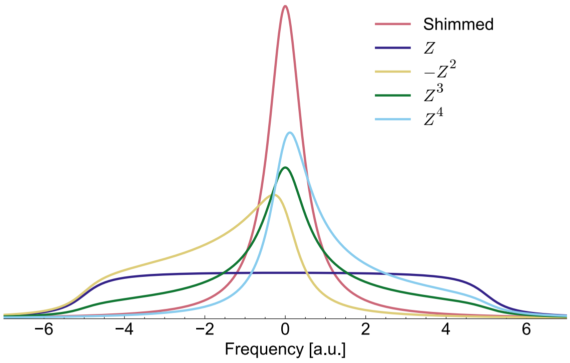


Figure 3.1.: Shim order revealed by peak's distortion shape.

Fortunately, different distortions to the spectral line shape can reveal the shims' order, particularly if they are designed as spherical harmonics (Weiger & Speck, 2011; Pearson, 1993). Distortions caused by odd-ordered gradients tend to be symmetrical, while those induced by higher-ordered shims predominantly affect the lower parts of the spectral peaks. Figure 3.1 visualizes these distinguishable broadening effects of Z, Z^2, Z^3 shims, simulated with the SHIMpanzee shimming environment (see Subsection 3.5.1).

However, this effect only reveals the shims' order and is ambiguous between shims, i.e., X^2 and Y^2 would yield the same distortion.

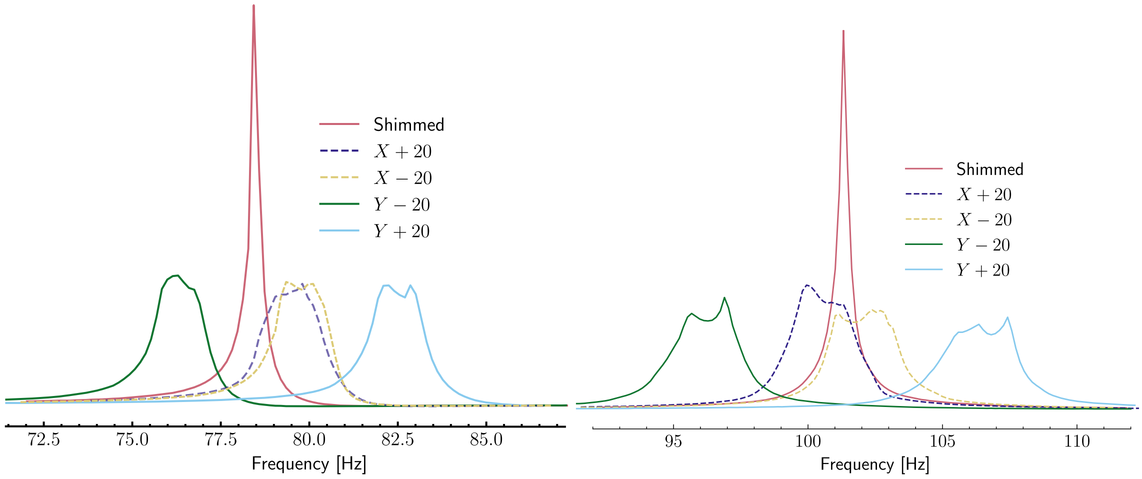
3.3. Hardware non-idealities reveal shim‘s sign and allow disambiguity

Besides the theoretically known distinguishable effects among higher-order shims, shims of the same order should ideally be fabricated orthogonal; thus, they should have the same broadening effect. Empirical evidence, regardless, demonstrates that shims of the same order also show distinct features when measured on real hardware. This effect was observed on different spectrometers, and is presented for two Magritek benchtop NMR magnets described in Subsection 3.5.1.

Both negative and positive shim offsets, as well as the shims themselves, can be differentiated and lead to unique peak distortions and frequency shifts (see Figure 3.2a and Figure 3.2b). Unique spectral features can be distinguished between coils, such as the right spike for the Y shim on 80 MHz, as visualized in Figure 3.2b. This spike is similar for positive and negative offsets, and does not appear for X shim offsets. Furthermore, positive and negative offsets can be differentiated as a peak position shift. This effect, however, was not unique to the Magritek 60 Ultra for the X shim (see Figure 3.2a).

All these effects could be caused by hardware non-idealities of the spectrometer, such as differences between the shim coils wiring. Furthermore, off-centre positioning or non-ideal coil wirings could lead to a Z^0 shift. This effect is expected to be even more severe for manufactured custom hardware.

A DL algorithm is expected to discover even more such features that are not visible to the human eye, as they are minor and hardware-dependent.



(a) Measured on 60 MHz.

(b) Measured on 80 MHz.

Figure 3.2.: Non-idealities of different machines allow to disambiguate between shims but are unique per device. (Subfigure b adopted from Becker et al. (2022a).)

3.4. Summary - Why learning is needed (and the solution)

In summary, the above-mentioned features of spectral peak distortions can be exploited to infer and distinguish shim currents. This, however, required a more sophisticated approach that can process full spectra and correlate them to shim correction terms.

This gap was filled using AI algorithms to guide the shimming process. It is hypothesized that an AI algorithm, either DL or RL, can learn shim values given 1D ^{1}H spectra. The main objective of the approaches in this thesis was to enhance and accelerate the shimming process.

It was anticipated that developing an AI-driven shimming approach would lead to the realization of the following benefits:

1. An AI algorithm, either DL or RL, can learn shim values given 1D ^{1}H spectra.
2. AI-driven shimming can speed up and improve shimming.
3. AI extracts additional hidden features in sequences of multiple spectra.
4. Using AI relieves the necessity to define priors or mathematical formulations of the spatial shim functions, as required for gradient shimming
5. AI-driven shimming is supposed to be a knowledge-based approach, simply learning from data.

3.5. Requirements for a DL-based shimming approach

Setting up a machine learning framework for a new problem brings many hidden technical costs, i.e., the ML code itself is comparably simple to implement compared to the surrounding infrastructure that, for example, includes configuration, data collection, or machine interfaces. For an AI-driven shimming approach to function, several requirements are given (see Figure 3.3), namely a target system to be shimmed, an interface to this, and some form of data(set) from this system, which then can be used to train a formerly selected neural network architecture. Finally, a reproducible evaluation scenario should be created, both "offline", i.e. on a hold-out test set, and "online", i.e. on real hardware.

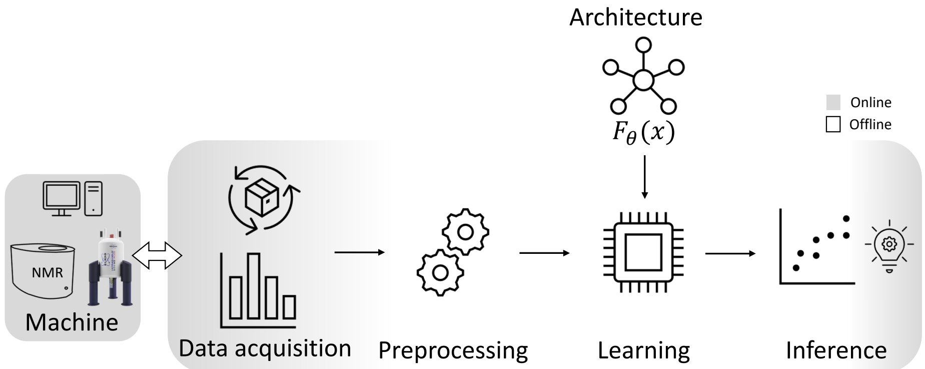


Figure 3.3.: Working sites of a deep learning pipeline, where online (on real hardware) and offline are differentiated.

3.5.1. System to shim

Generally, deep neural networks can be trained in either simulation or reality. Simulated environments are cheap and easy to compute and do not require real hardware, which is expensive to buy and operate. However, simulated environments often do not represent the real world, and models trained in one domain, usually do not generalize well to new domains (Farahani et al., 2021; Pan & Yang, 2010). This is also described as the reality gap. This is particularly notable for the hardware non-idealities for NMR spectrometers, as seen in section 3.3.

This makes it unavoidable to use real hardware for testing purposes. However, simulations can be utilized for prototyping and to reduce the search space of hyperparameters.

The following systems are considered within this thesis:

SHIMpanzee – A shimming simulation The SHIMpanzee simulator is an open-source project published by van Meerten and Franssen that mimics the shimming problem for shims up to third order (van Meerten & Franssen, 2017). The simulation allows inferring changes to a single spectral peak by applying spherical harmonic distortions (the shims' usual spatial representation (Golay, 1958)) to a cylindrical volume. The spectrum is obtained as the histogram over all voxels of the distorted volume, which follows the general shimming idea (section 2.2).

Due to orthogonality among most shims in an ideal simulation, any changes to similar shim pattern currents (e.g. X and Y) result in the same line broadening effects. Thus, Z, Z^2, Z^3 shims must be used to get distinguishable broadening effects for DL training.

For the purpose of this thesis, the SHIMpanzee simulator has been modified, namely, normalization of shim values to $[-1, 1]$ and spectrum scaling to avoid vanishing gradients during DL training. Further, the simulation has been wrapped into a gym environment (Brockman et al., 2016) to allow easy interfacing with common RL algorithms.

Magritek benchtop NMR spectrometers Low-field benchtop magnets are relatively low cost (<100k€), and their compactness allows usage within small laboratories. Furthermore, they do not require maintenance, such as filling liquid nitrogen or helium for cooling. However, they show lower resolution than high-field systems and usually do not have any gradient coils.

Three spectrometer systems from Magritek GmbH (Aachen Germany, 202 (2021)) were used with standard 5 mm sample tubes:

- Magritek Spinsolve 80 Carbon with a ^{1}H frequency of 80.24 MHz.
- Magritek Spinsolve 60 Ultra Carbon with a ^{1}H frequency of 61.92 MHz.
- Magritek Spinsolve 60 Ultra Multi-X with a ^{1}H frequency of 61.95 MHz.

The systems are equipped with first and second-order shim coils, and come with the Spinsolve-Expert software that allows script editing.

Preclinical Bruker MRI magnet A preclinical 15.2 T ultra high field magnet (Bruker, Ettlingen, Germany) was used with a commercial probe equipped with a 35 mm diameter birdcage coil tuned to 650 MHz for ^{1}H nuclei. The system was controlled with the Paravision software.

3.5.2. System interface

An interface between the DL framework and the actual shim controller and corresponding NMR signal outputs is required to allow flexible control of the systems mentioned above. In terms of the simulator, a simple Python wrapper in the form of a gym environment (Brockman et al., 2016) was implemented.

For real hardware, special scripts are required:

Magritek systems An interface between Python and the python-like programming language `Prospa`, the basis of Magritek's Spinsolve Expert software, was implemented. The interface allows to utilize open-source python libraries for NMR data processing (`nmrglue` Helmus & Jaroniec (2013)) and deep learning frameworks (`PyTorch` (Paszke et al., 2019), and `ray tune` (Liaw et al., 2018)).

Bruker systems It is challenging to modify deployed software applications to access specific hidden parameters, such as the hardware shim components in Bruker's Paravision software. Also, this magnet was used with a custom-built parallel probehead (see chapter 6), which requires an external shim control source (and software). Conveniently, a Python click bot was utilized to connect the Bruker software for automated spectra acquisition, the shim current source's interface, and the file system for signal output.

3.5.3. Data collection and preprocessing

Deep learning algorithms strongly rely on the availability of datasets. Ideally, the datasets should be as large as possible and represent the shimming task accurately and in an unbiased way. Indeed, biases are often overlooked issues in datasets (Torralba & Efros, 2011) but are sometimes unavoidable due to the desired application. Therefore, each AI-driven shimming approach requires the acquisition of a new dataset tailored for the dedicated shimming use case and the number of shims, e.g., shimming a probe from scratch, regular fine-tuning, or shimming parallel channels.

Proper preprocessing steps such as data cleaning, normalization, and feature extraction are crucial after data acquisition to ensure sufficient and representative data that is digestible[1] by a DL model.

A sharing platform has been created to make all acquired datasets publicly available for other researchers. The shimming database (ShimDB), available via `https://github.com/mobecks/ShimDB`, collects proton NMR signals recorded under shim coil field distortions.

3.5.4. Input data design and architecture selection

In contrast to computer vision problems, where a bijective mapping between inputs and outputs is usually given at the input's pixel level, NMR problems are non-bijective. Shimming an NMR spectrometer, for example, originates in an inhomogeneous 3D volume, yielding a 1D NMR signal. Thus, there is no straightforward link from a single 1D signal

[1] For example, data that is not normalized will lead to exploding gradients during training.

to all proper shim values. For example, even for a Z^2 distortion that will give asymmetric line distortion, from which the sign of the shim correction can be inferred, it is unclear what values the other shim coils must take.

DL, however, is well known to find hidden patterns in data (sequences). Therefore, the input to the neural network should consist of a smart batch or sequence of 1D NMR signals under different known shim perturbations. This can be compared to RGB images, where each image channel represents a differently coloured "viewpoint" of the actual target.

Once a proper input shape has been selected, a fitting neural network architecture must be chosen to process these (high-dimensional) inputs and give proper (low-dimensional) predictions. For example, an NMR spectrum will benefit from convolutional processing, where spatial features of the spectrum can be detected by convolutional feature maps. On the other hand, Recurrent Neural Networks (RNNs) are more appropriate for sequential data of varying lengths, as they work well in handling temporal dynamics and dependencies in the data. Notably, the architecture's output layer must be carefully tailored to align with the specific objectives of shimming.

The choice of architecture, therefore, is not just a technical decision but a strategic one, directly influencing the success of the shimming process. Incorporating automated neural architecture search (NAS) algorithms offers a systematic and efficient method for exploring the vast landscape of possible architectures, identifying the most suitable tasks like shimming. NAS algorithms automate the design process of neural network architectures, iterating through numerous configurations to find the optimal structure that balances performance with computational efficiency.

3.5.5. Training and optimization

The DL model must be trained on the collected data using an appropriate optimization algorithm, such as stochastic gradient descent (SGD) or Adam (Kingma & Ba, 2014). This involves defining a suitable loss function and determining the optimal hyperparameters for the approach, such as the learning rate, batch size or the number of epochs, as they significantly influence the model's learning process and performance.

Automated Hyperparameter Optimization (HPO) algorithms can be employed to simplify the hyperparameter tuning process. HPO basically acts as an optimizer for the DL optimizer. It automatically trains multiple models with different hyperparameter settings and optimizes these parameters to minimize an error function. This method significantly enhances the efficiency of the model's training, leading to more accurate DL models, while reducing manual interactions.

3.5.6. Evaluation and validation

Evaluating and validating a trained DL model are critical steps to assess the model's performance and applicability. Firstly, testing the model on a separate test set, unseen during training, is essential to objectively evaluate its performance and generalization. Analyzing the error on the test set ensures that the DL model is statistically sound. Finally, and most importantly, the model's real-world applicability must be assessed, which, in the context of shimming, means evaluating the model's performance in an "online" setting. These in situ evaluation tests ensure they are practically effective in their intended application.

4. Feasibility study for AI-driven shimming of linear shims

The proposed method of this chapter was published in an article titled "Deep regression with ensembles enables fast, first-order shimming in low-field NMR" in the "Journal of Magnetic Resonance" in 2022. The work was also presented at the Experimental Nuclear Magnetic Resonance Conference (ENC) 2022 and the European Conference on Magnetic Resonance (EUROMAR) 2022. This chapter's text and figures were reproduced and adapted from the author's published work.

4.1. Overview

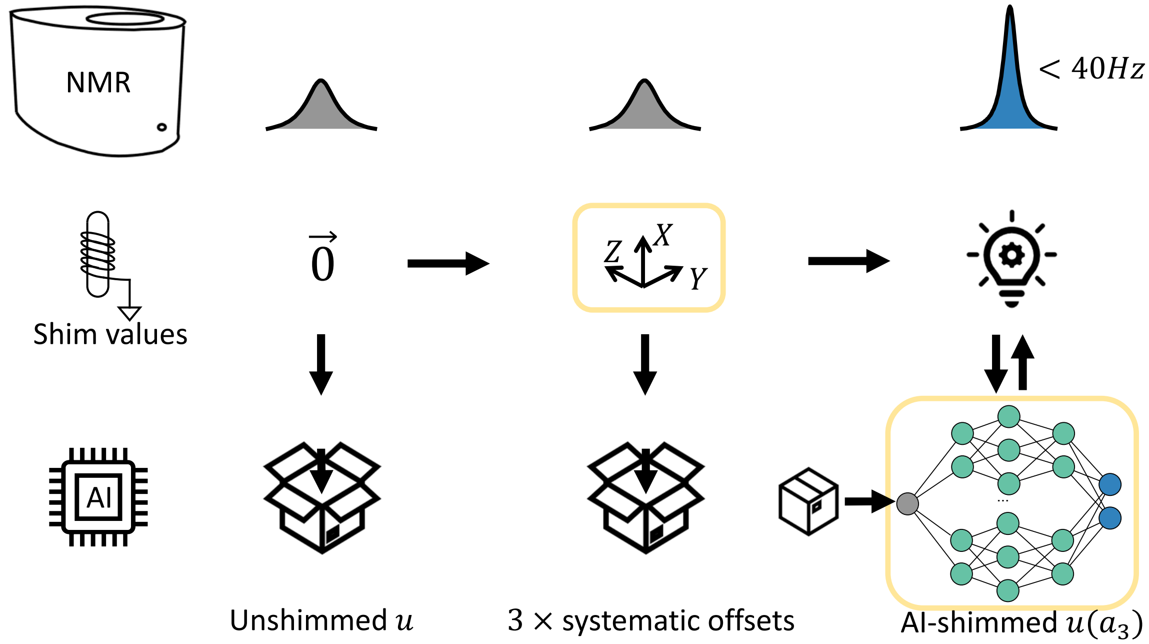


Figure 4.1.: **Graphical abstract of deep regression with ensembles (DRE)**. Differences from the other approaches within this thesis are highlighted.

Signal-based shimming is still a tedious and time-consuming process, mainly due to the non-orthogonalities of the shim coils and ambiguities between field distortions in the sample volume and one-dimensional NMR signals. Even though it was known that NMR signals reveal information about inhomogeneity patterns (see chapter 3), general optimization algorithms were used to shim the magnetic field.

This chapter describes a unique proof-of-concept for utilizing AI for shimming, i.e. AI-driven shimming, and hypothesized that a deep neural network could learn shim values given a batch of one-dimensional ^{1}H NMR spectra (see Figure 4.1). Accordingly, a method for rapid shimming was established based on a supervised deep regression approach with ensembles. Deep regression approaches are supposed to automatically detect non-linear relations between high-dimensional input data and numerical targets. Furthermore, ensembles of multiple weak learners are usually combined to reduce the prediction variance. The proposed approach was called **deep regression with ensembles (DRE)**, which focused on a non-iterative scenario for initial shimming to reach a state near the global optimum rapidly. The use-case represented shimming a probe from scratch with first-order shims only, which could be beneficial for the acceleration or improvement of existing automated shimming methods, focusing on high-throughput NMR alone or in conjunction with miniaturized hardware (Li et al., 1999; Korvink et al., 2019), without the use of gradients or even a lock channel.

Due to the lack of publicly available data and machine-specific non-idealities, a dataset for first-order NMR shimming that allowed inference of spectral changes depending on shim offsets was acquired. The dataset was then used to train multiple deep regression models (so-called weak learners) that could simultaneously predict three first-order shim currents given four distinct NMR measurements: the current unshimmed spectrum, and three spectra with individually modified shim values. An ensemble then combined the predictions of its weak learners via a *meta*-model to increase prediction stability. The performance of the proposed method was evaluated on real hardware. Furthermore, a limited comparison with regular shimming based on the downhill simplex method was conducted.

The main contributions of this chapter are summarized as follows:

- First feasibility study of AI-driven shimming,
- by applying a deep convolutional neural network to regress shim correction values,
- utilizing the first dataset of NMR spectra for shimming.

4.2. Method - DL part

This section focused on the deep learning part of AI-driven shimming, starting with a formal problem definition, followed by a description of the neural network architectures used for this chapter's approach, and completed with the concept of how the DL approach was applied to shimming.

4.2.1. Problem definition

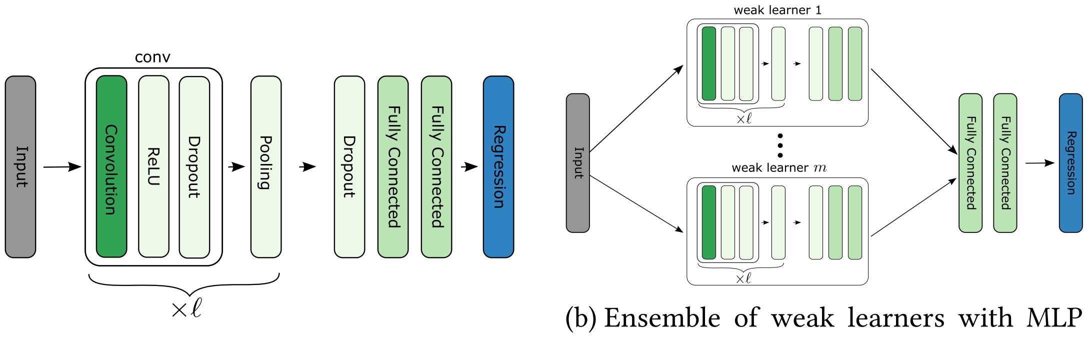
Let $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_i\}_{i=1}^{|\mathcal{D}|}$ be a database, where $(\mathbf{x}, \mathbf{y})_i$ is a pair including the input $\mathbf{x} \in \mathbb{R}^{W \times 4}$ with dimensions $W \times 4$, where W is the input spectra's width, and the associated target $\mathbf{y} = (y_1, y_2, ..., y_n) \in \mathbb{R}^n$ is defined as a real-valued vector of n separate shim coils. Also, consider the regression model $\mathrm{F}_\theta(\cdot)$, represented by a deep convolutional neural network with parameters θ. The network parameters θ are learned in a supervised manner using the database $\mathcal{D}$ to minimize the mean squared error (MSE) between the prediction $\hat{\mathbf{y}} = \mathrm{F}_\theta(\mathbf{x})$ and the target $\mathbf{y}$.

Regarding shimming, each shim value w_i is represented by a prediction $\hat{\mathbf{y}}_i$. The inputs $\mathbf{x}$ are defined as $\mathbf{x} = [u(0), u(s_X), u(s_Y), u(s_Z)]$, where the unshimmed spectrum u changes as a function of the systematic shim offsets s_X, s_Y, s_Z to the X, Y, Z shim coils. The DL model predicts the shim correction terms $\mathrm{F}_\theta(\mathbf{x}) = (\hat{y}_X, \hat{y}_Y, \hat{y}_Z)$, such that $\mathbf{y}_i - \hat{\mathbf{y}}_i \approx 0$. Note that neither the shim field patterns S_i, nor the magnetic field B_0 is know.

The model F is represented by an ensemble of weak learners combined with different meta-models, e.g. a multi-layer perceptron (MLP).

4.2.2. DL architecture

CNNs are a great candidate for high-dimensional input data with sparse information or spatially distributed features across multiple and high-dimensional input dimensions (see Subsection 2.3.4). Thus, a one-dimensional convolutional neural network regressed shim values given (multiple) NMR spectra. Furthermore, an ensemble of numerous weak learners was supposed to reduce prediction variance and uncertainty.



(a) Weak learner.

(b) Ensemble of weak learners with MLP meta-model.

Figure 4.2.: DL architectures for DRE. Given a batch of four spectra, the convolutional neural network (and ensemble thereof) predicts three shim correction values. (Edited from Becker et al. (2022a).)

Level-0 weak learners Each level-0 model (see Figure 4.2a) consisted of a convolutional block that extracted features from the four input spectra, followed by two fully connected layers with 32 nodes that predicted the shim corrections. In detail, the convolutional block consisted of $3-5$ layers, each represented by a sequence of one-dimensional convolutions with 32 filters and varying kernel sizes, ReLU activation, and dropout. The last layer used linear activation for the regression of the targets.

Level-1 meta models The *meta*-model combined features of its m weak learners, representing a mixture of stacking and boosting (see Subsection 2.3.4). Heterogeneous weak learners were chosen to allow for higher variance in predictions useful for the ensemble model. The following modifications were considered and evaluated against each other:

- Simple averaging of the weak learner predictions.
- A single fully connected layer combining the weak learners' regression layer ($\in \mathbb{R}^{m \times n}$).
- A two-layer multi-layer perceptron (MLP) based on the second-to-last fully-connected layer of the level-0 models. The MLP used $m \times 32$ nodes in its first and 32 in its second layer (see Figure 4.2b).

4.2.3. Concept

Conceptually, shimming with DRE is a non-iterative process. Starting with the unshimmed initial spectrum, three offsets were applied to X, Y and Z shims. The corresponding batch of four spectra was fed into the DL model to predict the shim corrections (Figure 4.3).

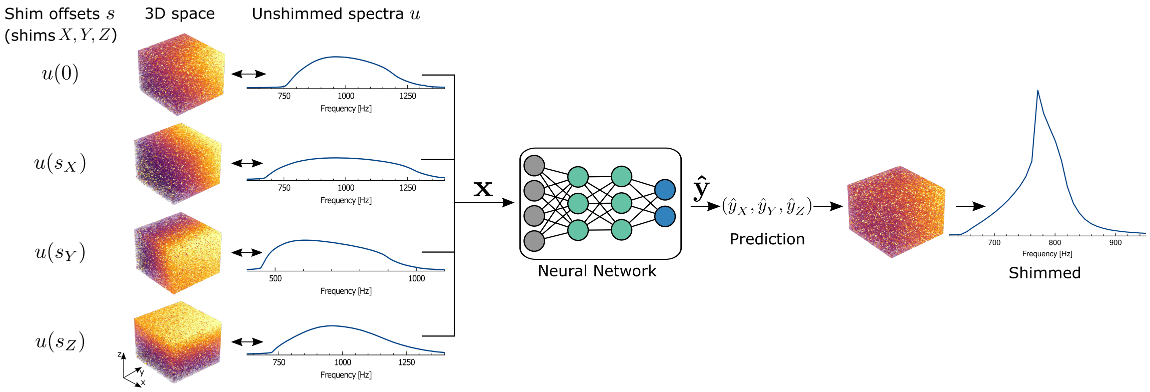


Figure 4.3.: **Concept of deep regression with ensembles (DRE).** Three-dimensional distortions (illustrated as inhomogeneity cubes) of the sample volume collapse to a one-dimensional signal. A batch of distinguishable spectra is obtained by systematic offsets of the available shim currents, which serve as input to a deep neural network. The prediction contains the shim values to achieve a more homogeneous field and, thus, a higher-quality spectrum. (Adapted from Becker et al. (2022a).)

4.3. Method - Hardware setup and dataset acquisition

4.3.1. Spectrometer hardware and sample

A low-field benchtop magnet (Magritek 80MHz) without gradient coils was chosen for the feasibility study in this chapter (see chapter 3 for a detailed description).

The measured sample consisted of distilled water mixed with copper sulfate to reduce spin-lattice or longitudinal relaxation time T_1, allowing for faster data acquisition. In analogy to the study (Kjær et al., 1987), the 50 mL H_2O was mixed with 0.062 g $CuSO_4$+5 H_2O (CAS No. 7758-99-8), resulting in a concentration of 5 mmol/L $CuSO_4$. Inversion recovery experiments revealed that $T_1 \approx 290$ ms.

4.3.2. Dataset acquisition

A small-scale dataset containing over 9000 NMR signals has been recorded under the application of linear shim offsets. The dataset is called LinearShimDB and is part of the ShimDB database. LinearShimDB allows inference of changes to the NMR spectrum or free induction decay (FID) depending on X, Y and Z shim offsets.

The dataset's reference spectrum R and reference shim values `Ref` have been acquired after shimming the magnet with the manufacturer's automated shimming technique, based on the downhill simplex method (Press, 2014). All data samples were then obtained by relative, systematic offsets s from the reference shim values R in a range R with step sizes α, β, γ, which change in a grid-like manner and define the target $\mathbf{y} = \mathbf{y}_{\mathrm{R}} + (\alpha s_X, \beta s_Y, \gamma s_Z)$. R is chosen large enough to mimic shimming a probe from scratch.

Each data instance includes the following information: A binary file containing the raw ^{1}H-FID with dimensions 1×32768; the shim values $\in [-2^{15}, 2^{15}]$ for n shims; the acquisition parameters; and the processing parameters. We pretend that the spectrometer only has $n = 3$ first-order shims. Thereby, only X, Y, and Z shim values are non-zero.

All experimental parameters and the dataset's characteristics are summarized in Table 4.1.

Characteristics	Nr. spectra	9261
	Shim range R	±10000
	Step size s	1000
	Shims	X, Y, Z
Acquisition parameters	Nucleus	^{1}H
	Bandwidth	20 kHz
	Points	32768
	Repetition time	2000 ms
	Filtering	-
	Phase correction	ϕ_0

Table 4.1.: Characteristics and acquisition parameters of the first-order shimming dataset (LinearShimDB).

Transfer dataset Acquisition in a grid-like manner exhibits several disadvantages. One is a temporal discrepancy between samples acquired at the beginning versus at the end of dataset collection. During inference, offsets close to the starting values are sampled, compared to row and column-like acquisition in row and columns during grid-like dataset collection. Therefore, an additional transfer database $\mathcal{T}$ with $|\mathcal{T}| = 100$ has been acquired to differentiate from the systematic nature of data collection. $\mathcal{T}$ was obtained under the same conditions as $\mathcal{D}$, but each of the four spectra in a batch $\mathbf{x}$ was jointly acquired and used to fine-tune the weak learner or the *meta*-model.

4.4. Results - Deep Learning

4.4.1. Data preprocessing

Dataset creation The input batches $\mathbf{x}$ were constructed from the dataset by mining unique target values $\hat{\mathbf{y}}$ from the dataset $\mathcal{D}$, and three spectra corresponding to offsets s to X, Y, and Z shims, respectively. All labels were defined as relative shim offsets to the reference spectrum of $\mathcal{D}$ to prevent the model from memorising absolute shim values for spectra. The spectra were generated by fast-Fourier-transformation and phase-correction of the raw FIDs using `nmrglue` (Helmus & Jaroniec, 2013) and the phase correction values given by the system's auto-phase method. Then, the real part of the spectrum was extracted.

The dataset was split into training, validation, and test of size 6400/800/801, respectively.

Normalization To achieve faster convergence, each spectrum was normalized by a constant normalization factor of 10^5 (the maximum intensity for perfect shims) such that the spectral values lie within the range $[0, 1]$. All spectra were downsampled from 32768 to 2048 data points as the distorted peaks span almost the full bandwidth of the spectrum. The `int-16` regression targets were divided by 2^{15} to avoid exploding gradients and multiplied by 100 to avoid vanishing gradients during training.

Augmentation Data augmentation included shifting the spectra, i.e. the peak position changes and imitates a drifting magnet. Furthermore, label noise was used to increase the prediction robustness.

4.4.2. DL training of a convolutional neural network for shimming

As mentioned in section 2.3, tuning the hyperparameters during DL training and designing an optimal neural network architecture for the task at hand is as cumbersome and error-prone as shimming a magnet by hand.

Here, Hyperparameter Optimization (HPO) and Neural Architecture Search (NAS) are handy to significantly accelerate the model development process and enhance performance while reducing the need for manual intervention. Both methods can be utilised by frameworks such as ray tune (Liaw et al., 2018), where one can define search spaces across different hyperparameters and architecture design choices.

Weak learners training The NAS search space of a single model (or weak learner) consisted of a variable number of layers ℓ, kernel sizes, and other design choices.

More precisely, the search space for convolutional layers covers 3, 4 or 5 layers, kernel sizes $\in \{11, 21, 31, 41, 51, 71\}$, stride $\in \{1, 2, 4\}$, 32 or 64 feature channels and pooling of 1 or 2. Dropout was used for convolutional and fully connected layers with probabilities 0.2 and 0.5, respectively.

Ensemble training Ensembles of neural networks are trained by first training several weak learners, e.g. as described in the previous chapter, and their weights are "frozen", i.e.

the trained weights of the weak learners are not allowed to be updated. The prediction of several weak learners was combined by a meta-model, which needs to be trained to shift importance between or combine the level-0 predictions.

The two trainable meta-model types introduced in Subsection 4.2.2, namely FC and MLP, were trained with hyperparameter optimization (HPO) using random search and early stopping. The search space of HPO for meta-model training is described in Table 4.2. For both meta-models, 500 networks with different hyperparameters were trained, and the best models were selected w.r.t. their validation loss.

Target	Parameter	Options
Architecture	Nr. weak learners	$\{10, 50\}$
	Meta type	{FC, MLP}
	Dropout	$\{0, 0.1, 0.2, 0.5, 0.8\}$
Augmentation	Shift type	{individual,batch}
	Shift values	$\in \{5, 10, 100\}$
	Label noise	$\{0, 0.1, 0.5, 1\}$
Training details	Learning rate	$[1 \times 10^{-4}, 1 \times 10^{-6}]$
	Epochs	{50,100}
	Batch sizes	$[8, 16, 32, 64, 128]$
	Weight decay	$\{0, 10^{-4}, 10^{-2}, 0.1, 0.5, 1, 2, 10\}$
	Optimizer	{SGD, Adam}
	Training set	$\{\mathcal{D}, \mathcal{T}\}$

Table 4.2.: HPO and NAS search space for ensemble training.

The shift type, given in array indices, describes whether all spectra in the batch were shifted differently (individual), or the whole batch was shifted with the same value (batch). The SGD optimizer used a momentum of 0.9, and the Adam optimizer $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to minimize the MSE loss.

Hardware requirements The training was performed using an AMD Ryzen 5900X with 64 GB RAM, and a graphics processing unit NVIDIA GeForce RTX 3090. The LinearShimDB dataset roughly requires 3.4 GB of disc space. Each weak learner allocated between 0.5 and 2.5 MB of disc space, and the *meta*-models roughly 200 KB. Time-wise, inference with the most resource-intense MLP-based ensemble model required 190 ms on average for prediction, using an Intel Core i5-8500 and 200 MB of RAM. In comparison, a typical NMR acquisition can take more than 4 s, and the raw FID takes more than 350 kB of disc space. Thus, AI-driven shimming does not put an additional burden on resource requirements.

4.4.3. DL training results

Before this thesis, whether DL could learn shim correction terms from a batch of NMR spectra was unknown. Offline training on the dataset $\mathcal{D}$ already showed that a DL model can predict three shim corrections given four 1D-signals while there is no apparent correlation between the input and output. A single model achieved an MAE of 596 ± 769 (mean± standard deviation) on the test set. The sampling resolution of the offset steps was $s = 1000$; thus, the prediction performance was close to the optimum near 1/2 of sampling.

Detailed results are given in Table 4.3. The entire network with ensembles was tested *in situ* only.

Weak learner results The top-50 architectures among 300 runs were selected w.r.t. validation error, and an exemplary selection of the top 10 is given in Table 4.3, which shows that the models have heterogeneous architectures with varying test errors. This, in turn, is beneficial for the diversity of the ensemble as it can ignore predictions of "bad" level-0 models by masking the weights with ReLU.

Additionally to NAS on level-0 weak learners, HPO was investigated, leading to worse *in situ* results. Thus, HPO was neglected for ensemble training.

Rank	MSE (↓) val	Channels	Kernel	Layers	Pooling	Stride	MAE (↓) test
1	10.1	64	51	3	1	2	596 ± 769
2	10.2	64	41	3	1	2	666 ± 806
3	11.5	64	31	3	1	2	702 ± 919
4	12.2	64	31	5	1	2	573 ± 622
5	12.4	32	51	4	1	2	575 ± 744
6	13.5	32	71	4	1	2	662 ± 758
7	18.3	32	71	3	1	4	717 ± 859
8	19.3	32	31	3	1	2	814 ± 954
9	20.2	64	51	5	1	1	813 ± 954
10	24.4	32	11	5	1	2	844 ± 891

Table 4.3.: **Exemplary training results for the top-10 weak learners.** Reported parameters are the mean squared error (MSE) on the validation set, the number of filter channels, kernel sizes and stride, the number of layers, the pooling factor, and the mean absolute error (MAE) on the test set.

Ensemble results The best parameters w.r.t. the validation error for both FC- and MLP-based meta-models shared 50 weak learners utilizing dataset $\mathcal{D}$, augmentation with full batch shifts of 10, weight decay 10^{-4}, 100 epochs and dropout of 0. The best FC-based ensemble achieved an $\text{MSE}_{val} = 22.0$ and was trained with batch size 32 and the SGD optimizer and a learning rate of 3.5×10^{-5}. The best MLP-based ensemble achieved an even better $\text{MSE}_{val} = 3.5$, while being trained with a batch size of 16, label noise of 0.1 (w.r.t. step size s) and the Adam optimizer with a learning rate of 1.5×10^{-5}.

4.5. Results - Experiments

4.5.1. Performance metrics and evaluation protocol

Several performance metrics were used to judge the improvement of spectral quality through shimming.

Metrics A custom criterion c has been introduced that combines linewidth and peak height into a single scalar value, and c describes whether a spectrum g is better than a reference spectrum r as:

$$c(g,r) = \frac{1}{2}\left(\lambda_1 \cdot \frac{\text{FWHM}(r)}{\text{FWHM}(g)} + \lambda_2 \cdot \frac{\max(g)}{\max(r)}\right), \tag{4.1}$$

where $max(\cdot)$ is the maximum peak height and FWHM$(\cdot)$ is the full width at half maximum of that peak. The criterion c indicates whether spectrum g is worse $(0 < c < 1)$ or better $(c > 1)$ than the reference r. Furthermore, λ_i is a weighting term between linewidth and peak height.

Based on this custom criterion, the success rate was introduced, which describes whether shimming was successful and is defined as:

$$\text{SR} = \begin{cases} 1, & \text{if } c_{sh} > [c_{init}, c_X, c_Y, c_Z] \\ 0, & \text{otherwise}, \end{cases} \tag{4.2}$$

where $[c_{init}, c_X, c_Y, c_Z]$ are the quality values for the input batch, calculated on the initial spectrum and the spectra with shim offsets to the X, Y, Z shims, and c_{sh} is the criterion after shimming.

Furthermore, the direction ratio DiR was introduced to indicate whether the signs of prediction and distortion match for all n shims. This indicates whether the method pointed towards the global minimum, and it is defined as:

$$\text{DiR} = \frac{1}{n}\sum_{i=1}^{n}\left(\text{sgn}(\hat{y}_i) == \text{sgn}(y_i)\right), \tag{4.3}$$

where sgn$(\cdot)$ is the sign function, $\hat{y}_i$ is the model's prediction and y_i is the true distortion.

Finally, the mean absolute error (MAE) between predicted and true shim distortions was measured.

Evaluation protocol To evaluate in-situ functionality and generalizability, in total 5 different models, i.e. DR and DRE with different meta-models, were tested on 100 random distortions $y_X, y_Y, y_Z \in [-10000, 10000]$ of X, Y, Z shims, drawn from a uniform distribution. The investigated samples include pure water with $CuSO_4$, ethanol (CAS No. 67-66-3) dissolved in water with a molar fraction of $\chi = [0.1, 0.5]$, and isopropanol (CAS No. 67-63-0) in water with $\chi = 0.5$.

4.5.2. In-situ experiments

Average values for all models, metrics, and the selected samples are reported in Table 4.4. The results indicate that an AI-driven shimming method works in practice, and it can predict shim correction terms given only a batch of four unshimmed NMR spectra with systematic offsets to the three linear shim currents.

The following describes detailed comparisons between the different model types and generalization tests to other samples.

Different model architectures Both the single model (weak learner) and MLP-based DRE achieve SR of 93% and SR 94% on water, respectively.

Simple averaging of the best 50 weak learners shows worse performance, and confirms that a meta-model needs to be trained. Here, a two-layer MLP with non-linear connections of the second-to-last features offers an advantage in improving all metrics compared to a non-linear combination of the weak learners' last layer.

The major difference between single and MLP-based DRE is the variance in criterion improvement and error. The results indicate a slightly more robust prediction behaviour for ensembles, especially in its variance. A qualitative evaluation is also given in section 8.5. The single model can, on average, achieve narrower linewidths but shows higher variance. This can also be seen in Figure 4.4, showing the distribution of linewidth change compared to the first spectrum, where the MLP was overall more stable.

Even a single model, i.e. weak learner, achieved an SR of 93% and a large improvement (mean of +435%) on the spectral quality for water.

For the most promising models, namely a single model and MLP-based DRE, graphical shimming results are reported in Table 4.5 for H_2O, ethanol and isopropanol.

Generalization to different samples Generalization of AI-driven shimming based on DR(E) was tested on samples with more than one spectral peak, namely ethanol with $\chi = [0.1, 0.5]$ and isopropanol with $\chi = 0.5$. As the training datasets $\mathcal{D}$ and $\mathcal{T}$ only consist of one sample with a single peak (water with CuSO4), general DL findings would assume that the prediction performance was biased towards this sample. Surprisingly, shimming performance yielded success rates above 91% for ethanol and isopropanol despite the risk of confusion among the peaks. One reason could be that the initial distortions of the peaks are so broad, that the peaks overlap and smear into one broad peak.

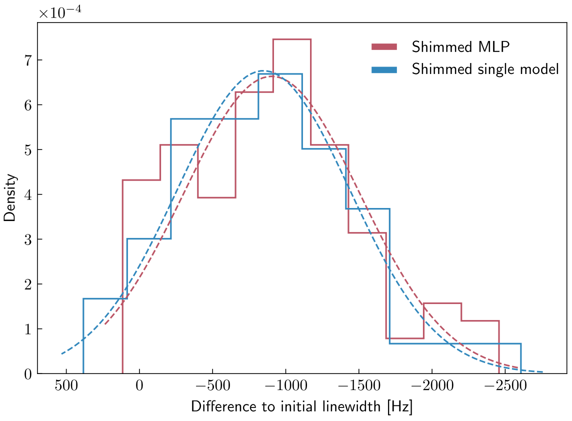


Figure 4.4.: Probability density function of linewidth changes, compared between a single model and an MLP ensemble. The dashed line indicates a fitted Gaussian. (Edited from Becker et al. (2022a).)

		Single model		Ensemble		
		Untuned	Tuned to $\mathcal{T}$	Average	FC	**MLP**
H_2O	Success Rate SR ↑	0.93	0.91	0.61	0.76	**0.94**
	Direction Ratio DiR ↑	**0.90**	0.89	0.77	0.78	0.89
	Mean *c* boost ↑	**+435%**$_{\pm 559}$	+394%$_{\pm 471}$	+125%$_{\pm 244}$	+127%$_{\pm 124}$	+399%$_{\pm 560}$
	Averaged MAE ↓	1878$_{\pm 1271}$	1703$_{\pm 969}$	3989$_{\pm 2529}$	3191$_{\pm 1459}$	**1756**$_{\pm 1016}$
Ethanol (0.5)	SR	0.92	0.89	0.55	0.73	0.93
	DiR	0.86	0.86	0.79	0.78	0.85
	Mean *c* boost	157%$_{\pm 130}$	136%$_{\pm 99}$	28%$_{\pm 14}$	73%$_{\pm 63}$	123%$_{\pm 34}$
	Averaged MAE	2431$_{\pm 1287}$	2492$_{\pm 1303}$	4510$_{\pm 1999}$	3441$_{\pm 1602}$	2460$_{\pm 1217}$
Ethanol (0.1)	SR	0.91	0.90	0.68	0.71	0.89
	DiR	0.87	0.85	0.77	0.78	0.85
	Mean *c* boost	+265%$_{\pm 214}$	+205%$_{\pm 158}$	+77%$_{\pm 38}$	+91%$_{\pm 66}$	+257%$_{\pm 314}$
	Averaged MAE	2170$_{\pm 1292}$	2478$_{\pm 1334}$	3995$_{\pm 2344}$	3462$_{\pm 1701}$	2250$_{\pm 1217}$
Isopropanol (0.5)	SR	0.91	0.95	0.65	0.80	0.90
	DiR	0.89	0.90	0.77	0.82	0.87
	Mean *c* boost	+299%$_{\pm 320}$	+257%$_{\pm 278}$	+74%$_{\pm 154}$	+91%$_{\pm 107}$	+201%$_{\pm 156}$
	Averaged MAE	1967$_{\pm 1177}$	2102$_{\pm 1229}$	4199$_{\pm 2533}$	3393$_{\pm 1561}$	2097$_{\pm 956}$

Table 4.4.: ***In situ*** **results of AI-driven shimming using the DR(E) approach** with a single model and different ensemble types. Values are reported as mean ± standard deviation over 100 random distortions drawn from a uniform distribution for different samples with molar fraction (χ). The best values are marked in bold. Abbreviations: *FC* = fully-connected, *MLP* = multi-layer perceptron, $\mathcal{T}$ = transfer database, SR = success rate, DiR = direction ratio, MAE = mean absolute error, *c* = criterion.

4.6. Results - Comparison

The shimming performance of the best models in this chapter, namely a single model (DR) and DRE with MLP as the *meta*-model, were compared to Magritek's built-in implementation of the downhill simplex method, which in turn is based on the algorithm in Press (2014).

The following scenarios were compared:

1. Theoretical minimum number of acquisition
2. Required number of acquisitions of AI-driven vs traditional shimming
3. Criterion improvement by combining AI and traditional methods, called "AI-assisted" shimming

Theoretical minimum The simplex method requires at least $n+1$ measurements for n shims to initialize its simplex structure, and one to four (average of two) function evaluations per iteration (Ernst, 1968). The proposed DRE method consistently needs four spectra for one iteration, and one spectrum to check the results, allowing for faster shimming in a fixed amount of time, given that the method succeeds.

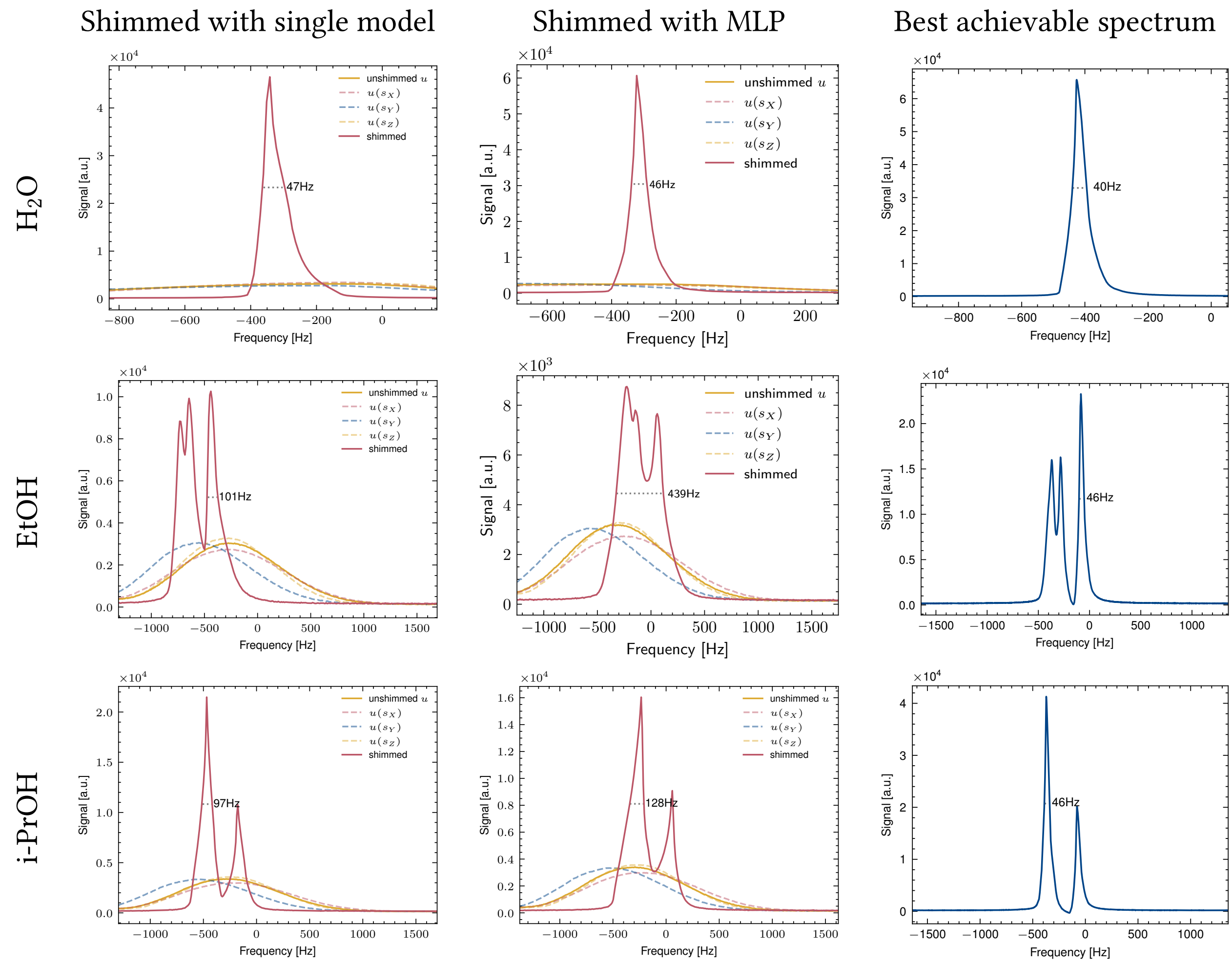


Table 4.5.: **Exemplary selected results of shimmed spectra** for water, ethanol (χ = 0.5), and isopropanol. A single model's performance was compared to ensembles with an MLP-based *meta*-model. Additionally, the optimal spectrum was reported by applying the simplex method only to the first-order shims. Abbreviations: EtOH = ethanol, i-PrOH = isopropanol. (Adopted from Becker et al. (2022a).)

Method	$\text{FWHM}_{initial}$	$\text{FWHM}_{shimmed}$ (↓)	fe (↓)
DR	$1432_{\pm 532}$	$516_{\pm 330}$	**4+1**
simplex	"	$552_{\pm 411}$	$16.3_{\pm 11.1}$
DRE	"	$486_{\pm 342}$	**4+1**
Simplex	"	$\mathbf{473}_{\pm 414}$	$21.1_{\pm 16.6}$

Table 4.6.: Comparison of necessary NMR acquisitions of the simplex method to achieve a similar criterion as obtained with the DR(E) method. The best values are marked in bold. Abbreviations: fe = function evaluations, FWHM = full width at half maximum, DR = Deep Regression, DRE = DR with ensembles.

Convergence speed of AI-driven vs traditional shimming The convergence speed, i.e. the number of necessary acquisitions, of AI-driven and traditional shimming was compared: The DR and DRE methods proposed in this chapter predicted shim corrections for 100 random distortions drawn from a uniform distribution. Then, the simplex was started from the same distortion with step size $s = 1000$, and it was stopped if it reached a linewidth equivalent to the one achieved with either DR or DRE, or it was stopped if it could not find an equivalent criterion within 50 iterations.

The resulting number of acquisitions is reported in Table 4.6 and proves that DR and DRE can accelerate the shimming process, even though the simplex tends to be robust against the initial simplex' size and shape (Ernst, 1968), and is known to produce rapid drops of initial values (Lagarias et al., 1998).

Criterion improvement of AI-assisted shimming The second comparison covers "AI-assisted" shimming, i.e. traditional methods were initialized with the prediction of the DL-based approach proposed in this chapter.

The results in criterion improvement were reported after i iterations of the simplex method with step size $s = 1000$, and after $i - 4$ simplex iterations while initialized with DR or DRE. Iterations were reduced by four because one iteration of DRE needs four spectra for its prediction.

Table 4.7 demonstrated the advantage of AI-assisted shimming: By initialising a traditional optimization method, which guarantees convergences, with a powerful but unstable DL method, the criterion could be increased roughly by a factor of 3, and this for the same number of NMR acquisitions. Thus, AI-assisted shimming is a promising candidate to achieve fast and reliable shimming results.

Note that deep regression (DR) performs similarly to deep regression with ensembles (DRE) in quality improvement. Still, ensembles demonstrate an advantage in the number of necessary NMR acquisitions compared to the simplex method. Compared to the simplex method's iterative procedure, which can take up to one hour, this is a major step in reducing time requirements (and can thus save money).

Method	Iterations i (↓)	Criterion after DR/DRE (↑)	Criterion after simplex (↑)
simplex	10	-	×5.2 ± 3.96
DR + simplex	1 + 6	×4.6 ± 4.9	×15.4 ± 11.1
DRE + simplex	1 + 6	×**5.0** ± 6.3	×**15.7** ± 11.7

Table 4.7.: Comparison of criterion improvement w.r.t. initial spectrum between the default downhill simplex and simplex initialized with the DRE method. Default simplex was run for i iterations and DR/DRE+simplex for $i - 4$ iterations because one iteration of DR/DRE requires four measurements. Here $i = 10$. Values are reported as the initial criterion $c_{init}\times$ improvement ± std. Best values are marked in bold. Abbreviations: DR = Deep Regression, DRE = DR with ensembles.

4.7. Discussion/Limitations

The first proof-of-concept for AI-driven shimming to accelerate the shimming process focused on a scenario with a deep convolutional neural network and only three linear shims, which generally was insufficient to achieve spectroscopy resolution. Nevertheless, the approach demonstrated that very broad initial lineshapes could be improved, nearly reaching the optimum achievable with linear shims only, and this, despite the non-orthogonality of shims and the bijectivity (or ambiguity) problem between field distortions and NMR signals.

However, DRE is a one-shot method compared to traditional iterative methods. Therefore, convergence is not guaranteed. Nevertheless, all DL-based approaches cannot guarantee convergence due to their stochasticity inside the network but can only provide uncertainty with their predictions. After all, the main drawback of DRE is its grid-like data acquisition, which does not scale well to more shims.

The next chapter introduces solutions, namely by addressing a different data sampling strategy, more shims, and allowing for an iterative AI-driven shimming method.

5. Doubly-deep learning for shimming with random shim values

The method proposed in this chapter has been previously published in an article titled "Acquisitions with random shim values enhance AI-driven NMR shimming" in the special issue on "Artificial Intelligence in NMR, EPR, and MRI" in the "Journal of Magnetic Resonance" in 2022. Additionally, the work has been presented at the Experimental Nuclear Magnetic Resonance Conference (ENC) 2023 and the Helmholtz AI conference 2023. The text and figures in this chapter have been reproduced and adapted from the author's published work.

5.1. Overview

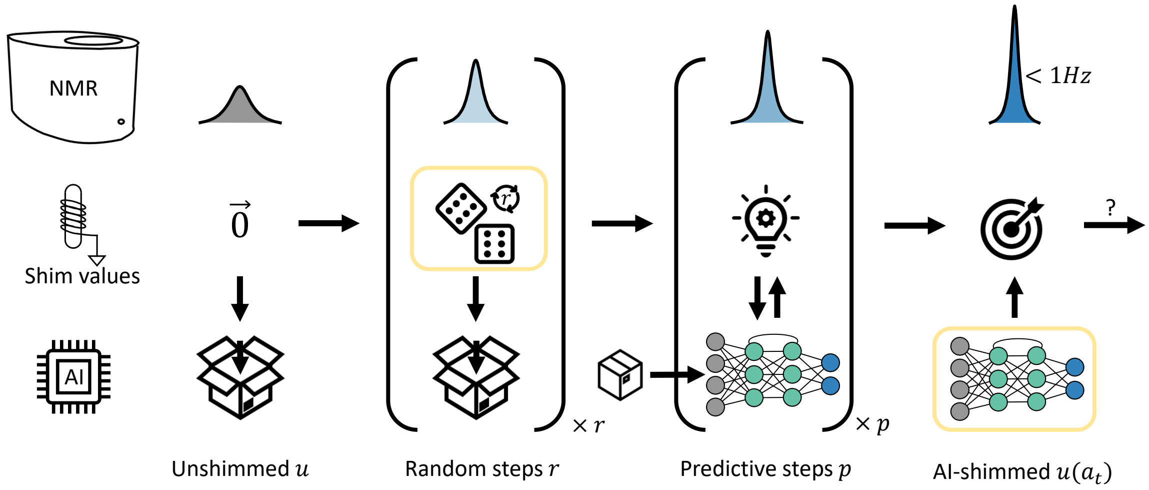


Figure 5.1.: **Graphical abstract of enhanced deep regression (eDR)**. Differences from the other approaches within this thesis are highlighted. (Edited from Becker et al. (2022b).)

In chapter 4, the advantage of using DL to speed up shimming has been proven, and it was shown that a batch of NMR spectra contain information to correct broad peak distortions of linear shims, i.e., the ambiguity between same-order shims can be solved. However, the DRE method was limited to providing one-shot predictions without iterations, making it susceptible to failure. A significant drawback was data collection in a grid-like manner, which exponentially[1] increased time consumption depending on the number of shims.

The approach described in this chapter was therefore developed to include more efficient data collection and more sophisticated DL algorithms, and the following enhancements (see Figure 5.1) have been implemented compared to chapter 4:

1. Incorporating an additional higher order shim, namely, Z^2,
2. Setting other available shim coils to a reference optimum value[2],
3. Experimental dataset collection with random sampling, including shim influence weighting,
4. Custom neural network architecture that combines convolutional layers with recurrent "action" information over temporal sequences,
5. Below 1 Hz linewidths on a reference sample peak.

The approach in this chapter was titled **enhanced deep regression (eDR), for fast, signal-based, and AI-driven shimming** and focused on machine-specific shimming of reference sample peaks for compact NMR.

5.2. Method - DL part

This section focuses on the deep learning part of AI-driven shimming, starting with a formal problem definition, followed by a description of the neural network architectures used for this chapter's approach, and completed with the concept of how the DL approach was applied for shimming.

5.2.1. Problem definition

Let $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_i\}_{i=1}^{|\mathcal{D}|}$ be a static dataset, where $(\mathbf{x}, \mathbf{y})_i$ is an input-output pair. With n being the number of separate shim coils, the input sequences $\mathbf{x} \in \mathbb{R}^{t \times (L,n)}$ of t entities are defined as $\mathbf{x} = \Big[\big(u(\vec{0}), \vec{0}\big), \big(u(a_1), a_1\big), ..., \big(u(a_t), a_t\big)\Big]$, where the unshimmed spectrum u of length L changes as a function of (random) shim offsets (or actions) $a \in \mathbb{R}^n$. Each associated target $\mathbf{y} = (y_1, y_2, ..., y_n) \in \mathbb{R}^n$ represents the distortion from the optimum and is defined as a real-valued vector of n elements. The regression model $\mathrm{F}_\theta(\cdot)$, represented by a custom deep neural network with parameters θ, predicts the shim correction terms $\mathrm{F}_\theta(\mathbf{x}) = (\hat{y}_X, \hat{y}_Y, \hat{y}_Z, \hat{y}_{Z^2})$, such that $\mathbf{y}_i - \hat{\mathbf{y}}_i \approx 0$. The network parameters θ are learned in a supervised manner using the dataset $\mathcal{D}$ to minimize a loss term $\mathcal{L}$ (usually the mean squared error) between the prediction $\hat{\mathbf{y}} = \mathrm{F}_\theta(\mathbf{x})$ and the target $\mathbf{y}$. Note the difference of the shim distortions S w.r.t. the reference spectrum (see Subsection 5.2.3), that simultaneously represent the labels $\mathbf{y}$, and the shim offsets (or actions) a_t w.r.t. the first, unshimmed spectrum in a shimming sequence.

[1] It is important to note that this issue could be resolved by ensuring each batch is sampled together.

[2] This imposes different requirements on the DL models due to significantly different features.

5.2.2. Architecture

A custom, "doubly-deep" neural network architecture was designed (see Figure 5.2), where each component has a specific advantage:

1. A convolutional part (CNN) extracts features from the individual spectra,
2. Past actions relative to the initial unshimmed spectrum are infused into the model, and
3. An LSTM cell allows to learn temporal dependencies in shimming sequences of flexible lengths.

The model's input-to-features section (`i2f`) extracts features from the spectrum via l convolutional blocks and infuses them with the current action a_t in a fully-connected layer. Taking the features and the last hidden state h_{t-1} as inputs, the LSTM cell then generates two outputs: a prediction of the shim distortions $\hat{\mathbf{y}}$ via a small fully-connected head (`f2o`), and the new hidden state h (via `f2h`). Different regularization techniques such as dropout (Srivastava et al., 2014) and layer normalization (Ba et al., 2016) are implemented throughout the architecture.

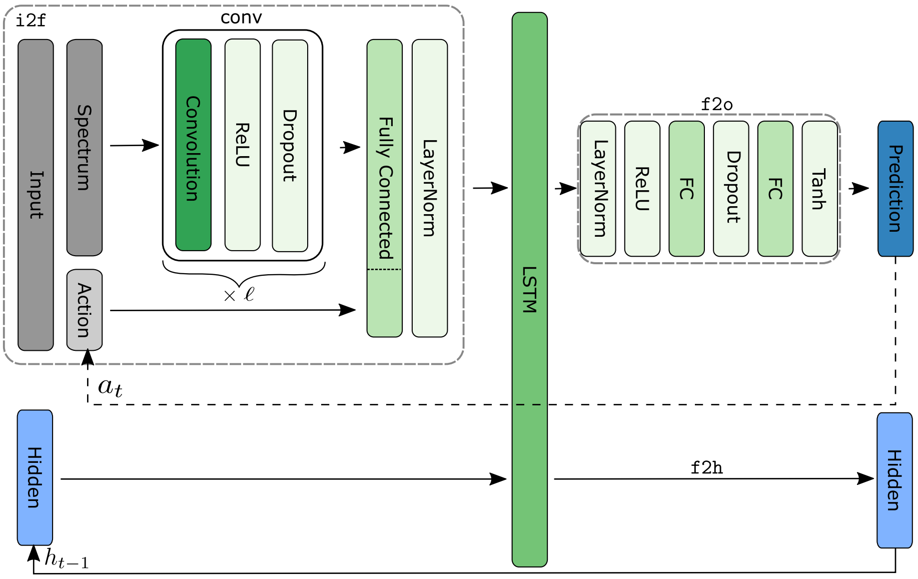


Figure 5.2.: **Convolutional LSTM architecture of enhanced deep regression (eDR).** Convolutional layers in the `i2f` block extract features from spectra, combined with the past action information. An LSTM cell maps these features and the last hidden state to the output prediction via `f2o`, and to the next hidden state via `f2h`. (Edited from Becker et al. (2022b).)

The input at time step t consists of a 1D spectrum, and the current shim offsets a_t. Action infusion helps the model to locate itself in shim space by learning from a sequence of past spectra and corresponding actions relative to the initial, unshimmed spectrum.

The value of action a_t depends on the time step:

$$a_t = \begin{cases} \vec{0}, & \text{for} \quad t = 0 \\ \mathcal{N}, & \text{for} \quad t = [1, ..., r] \\ \hat{\mathbf{y}}_{t-1}, & \text{for} \quad t = [r+1, ..., r+p+1], \end{cases} \tag{5.1}$$

where either Gaussian noise $\mathcal{N}$ or the last prediction $\hat{\mathbf{y}}_{t-1}$ was used to generate the next spectrum.

section 8.3 gives a detailed description of the architecture design process.

5.2.3. Concept

The AI-driven shimming concept (see Figure 5.3) in this chapter was based on a fixed number of r random shim offset steps applied to an initial, unshimmed spectrum u. Correspondingly, acquired spectra and shim offsets are fed into a convolutional recurrent neural network. These random steps serve as an exploration phase to resolve ambiguities between shims and generate a model-internal "shimming history". This helps the model locate itself in shim space while its predictions are ignored. During the following p predictive steps, the model's output was applied to correct the shim distortions, which gives an ideally shimmed spectrum $u(a_t)$ after $t = r + p + 1$ steps with actions a_t.

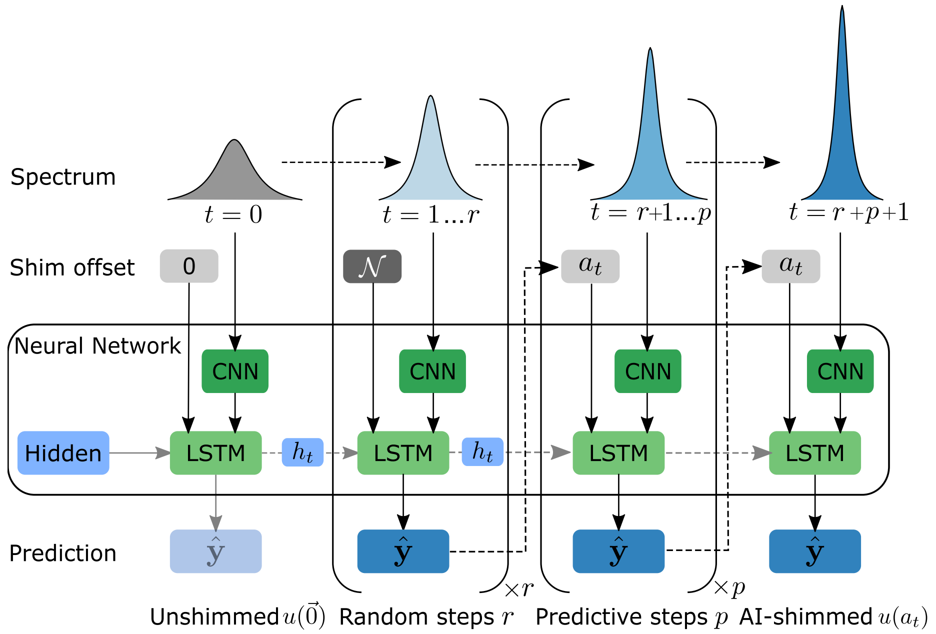
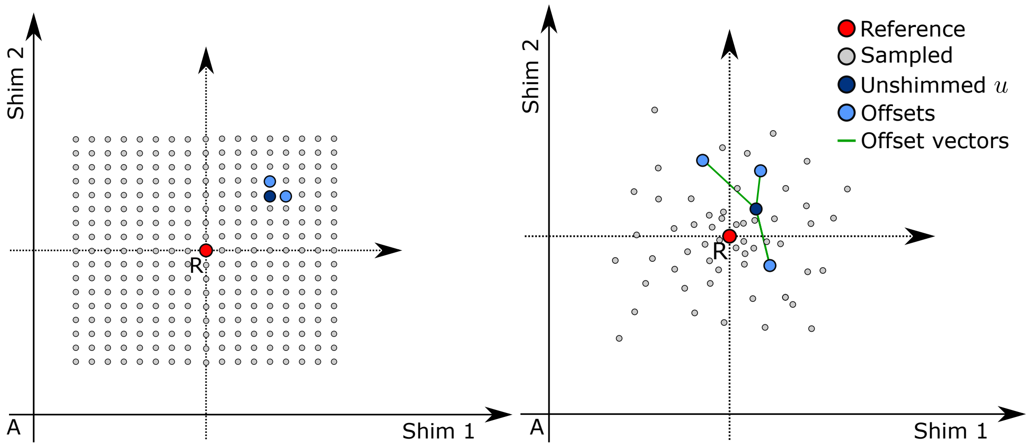


Figure 5.3.: **Concept of enhanced deep regression (eDR) for AI-driven shimming.** The model receives the current spectrum and shim offset values at each time step t to predict the shim corrections $\hat{\mathbf{y}}$. For r random steps, the shim offsets are represented by Gaussian noise $a_{t=[1,...,r]} = \mathcal{N}$, and for p predictive steps the predicted shim corrections serve as the next action $a_{t=[r+1,...,r+p+1]} = \hat{\mathbf{y}}$ (shim offset). (Edited from Becker et al. (2022b).)

As random exploration steps help the model locate itself, the dataset needs to be acquired correspondingly, and a static dataset of random shim offsets was used. Thus, the model was trained on only random steps r during training. The randomness avoids the limitations of grid-like sampling (see Figure 5.4a) for high dimensions: The number of acquisitions A in a grid-like sampling with m steps would increase exponentially with the number of dimensions n: $A = m^n$. Thus, randomized data sampling was used, independent of the problem's dimensions (see Figure 5.4b). Randomized sampling only requires adding the information of the current offset vector to the model's input.



(a) Grid-like acquisition.

(b) Random acquisition.

Figure 5.4.: **Comparison between grid-like and random data sampling** for a scenario of two shim coils. Abbreviations: A = absolute shim value frame, R = relative reference frame. (Adapted from Becker et al. (2022b).)

5.3. Method - Hardware setup and dataset

5.3.1. Spectrometer hardware and sample

For this study, a Magritek 80 MHz low-field benchtop magnet was selected, which did not have any gradient coils.

The measured samples consisted of (I) distilled water mixed with copper sulfate (5 mmol/L $CuSO_4$; CAS No. 7758-99-8) and (II) the magnet's reference sample, a mixture of 5vol% H_2O in D_2O.

5.3.2. Shim weighting

Shims coils, due to their different wiring patterns or specific sample's geometrical a-symmetry, have different influences on the NMR signal. This means the same step size for all shim coils would introduce a strong bias towards the most prominent shim coil. Therefore, the shim effects are normalized by extracting weighting factors based on the shim's influence on the root mean square (RMS) of the FID signal.

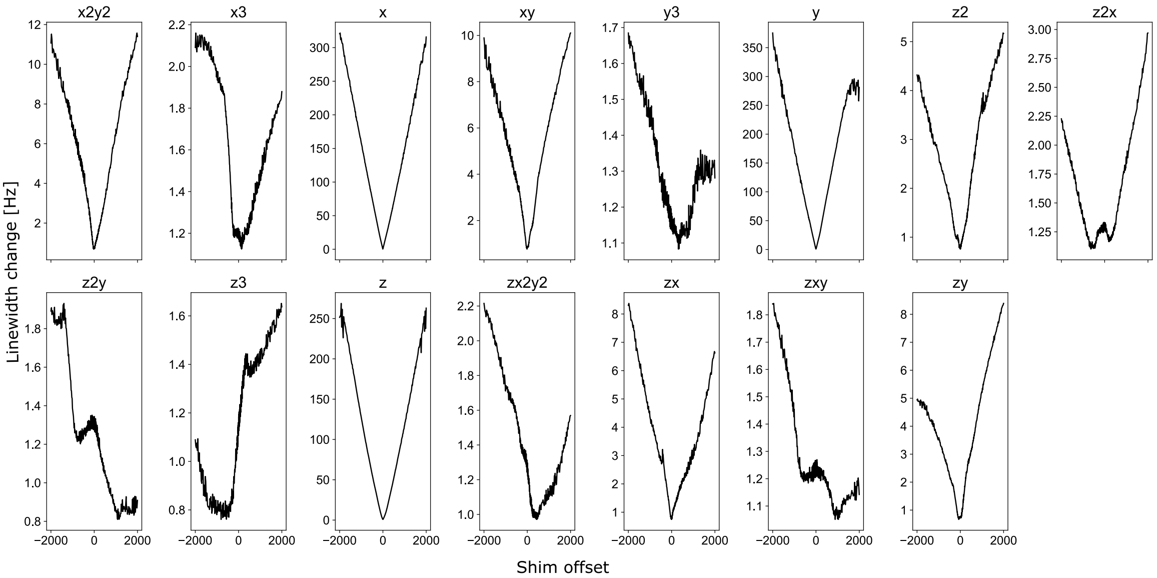


Figure 5.5.: Linewidth changes of a single peak with the same offsets to all shims. First-order shims have a stronger impact on the resolution.

Starting from a perfectly shimmed FID of a 5vol% H_2O in D_2O sample, each shim was changed until the FID's RMS decreased below 50% from its maximum. The number of steps needed for each shim with values $\in [-2^{15}, 2^{15}]$ to reach 50% degradation are used to define their influence. On the Magritek Spinsolve 80, the Y-shim had the most substantial impact, and thus, all other shims were re-normalize w.r.t. the Y shim, resulting in a shim range of ±50 for Y and weights $W(X, Y, Z, Z^2) = [1.2, 1.0, 2.0, 18.0]$. Figure 5.5 visualizes the impact of each shim on the linewidth of a single peak depending on the same shim offsets. The linewidth is affected differently by various shims. For example, the Y shim has a more significant impact than the Z shim.

5.3.3. Dataset acquisition

The shimming dataset RandShimDB consists of 5k and 10k spectra for samples I and II, respectively.

The reference spectrum R of optimal quality was obtained by the manufacturer's automated shimming techniques. The data samples were then acquired random distortions $\mathrm{S} \in \mathbb{R}^n$ from the reference shim values $\mathbf{y}_R$, following a Gaussian distribution: $\mathrm{S} = \mathcal{N}(0, \frac{1}{3}^2) \cdot R \cdot W$, where the weights W represent the shim influence and R is the range. Each instance of RandomShimDB includes the following information: a binary file containing the raw ^{1}H-FID with dimensions 1×32768, the shim values for n shims, the acquisition and processing parameters. By intent, only the X, Y, Z, and Z^2 shim values were varied, and the other shims were kept at their reference values to avoid the large distortions resulting from setting them to zero, as revealed in chapter 4.

All experimental parameters and the dataset's characteristics are summarized in Table 5.1.

Dataset parameters	Shims	X, Y, Z, Z^2
	Weightings W	[1.2, 1.0, 2.0, 18.0]
	Shim range R	±50
	Sample I	H_2O+$CuSO_4$
	Sample II	5vol% H_2O in D_2O
	Nr. spectra	{5000, 10000}
Acquisition parameters	Nucleus	^1H
	Bandwidth	5 kHz
	Points	32768
	Repetition time	2000 ms
	Phase correction	ϕ_0

Table 5.1.: Parameters of the randomized shimming dataset (RandomShimDB). W normalizes the influence between shims, and R is the discrete range for possible shim distortions, given as `int-16`.

5.4. Results - Deep Learning

This section describes critical pre-processing details to achieve faster convergence and generalization, which includes normalization and augmentation, as well as deep learning training details, and the results achieved offline on the dataset.

5.4.1. Data preprocessing

Dataset creation Input to the DL model was a sequence $\mathbf{x}$ of real-valued spectra and corresponding shim actions. The first spectrum in the sequence is uniformly random sampled from the dataset $\mathcal{D}$ with input-target pairs $(\mathbf{x}, \mathbf{y})_i$ (see Subsection 5.3.3), where the target value to this sequence is $\hat{\mathbf{y}}$ ($\equiv \mathbf{S}$). The following t spectra and their relative offsets a_t w.r.t. the first spectrum were added to the sequence by random extraction from the training set of $\mathcal{D}$. Thus, each sampling from the dataset during training yielded a sequence of changing spectra-actions pairs. Splitting ratios of train, validation, and test set for DL training were 80/10/10 % of $\mathcal{D}$, respectively.

Normalization The neural network's training convergence strongly relies on normalized input and output data to avoid vanishing or exploding gradients. Thus, the first spectrum of a sequence $\mathbf{x}$ was normalized to $[0, 1]$, and the following spectra according to the first's maximum. A region of interest (ROI) with 2048 points was defined around the HDO (hydrodeuterium oxide) peak, resulting in a sampling resolution of 0.15 Hz. The regression targets were normalized to $[-1, 1]$.

Augmentation Data augmentation can artificially increase the amount of data samples for scenarios where data acquisition is expensive, for example, NMR measurements. Augmentation included uniformly random Z^0 shift $\in [-4, 4]$ (array indices), uniform label noise of 0.1 w.r.t. the step size of each shim (derived from R and W), uniform shim interaction noise of 0.1, first-order phase distortions of $[-0.5, 0.5]$, additive white Gaussian noise (AWGN) with signal-to-noise ratio (SNR) of 30.

5.4.2. DL training of a ConvLSTM for shimming

An automated search algorithm using `ray tune` (Liaw et al., 2018) was utilized to find the best combination of architecture choices and hyperparameters. 150 models with varying architecture and hyperparameter search spaces were trained, and the best model was selected based on the validation error. All models during HPO were trained with sequence lengths $t = 10$, a learning rate of 5×10^{-4}, batch size of 32, and the Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.999$ for a maximum of 100 epochs to minimize the Huber loss while utilizing early stopping. The search space included between 3 and 5 convolutional layers with different kernel sizes and strides, and hidden sizes for the LSTM block.

The best parameters were selected as 5 convolutional layers with 64 channels, kernel size 19 and stride 2; a 2-layer LSTM cell with 1024 hidden units; dropout probabilities of 0.2 after both convolutional and fully-connected layers. The final training was conducted step-wise on varying sequence lengths, where every 25th epoch, the length of the input sequence was increased by 2 (up to 10).

5.4.3. DL training results

Convergence of the model during offline training, i.e. training on the static dataset $\mathcal{D}$, confirmed that a DL model can learn shim correction terms for first- and higher-order shims, given a sequence of random offsets and corresponding spectra.

The model achieved a normalized MAE on the test set of 0.023 ± 0.019 (mean ± standard deviation), where one discrete sampling step in shim space equals 0.02. The per-shim error of $[0.028, 0.023, 0.025, 0.017]$ for $[X, Y, Z, Z^2]$ indicated that the error was influenced by the shims' order. This can be explained as the sign of distortion of even-ordered shims can be inferred from a single spectrum as an asymmetric behaviour on the spectral line shapes.

5.5. Results - Experimental

This section captures experimental results on real hardware, including the corresponding performance metrics and evaluation protocol.

5.5.1. Performance metrics and evaluation protocol

Metrics To evaluate in-situ shimming performance, the linewidths at FWHM (50%) and 0.55% of the peak's maximum were reported. Furthermore, the mean absolute error (MAE), direction ratio DiR, and modified success rate SR (compared to Subsection 4.5.1) were reported. DiR indicates whether the predictions' and targets' signs match, and the modified success rate SR $\in [0, 1]$ for N experiments is defined as

$$\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N} \max\big(u_i(\hat{\mathbf{y}}_t)\big) > \max\big(u_i(\vec{0})\big), \tag{5.2}$$

where $u(\vec{0})$ is the initial spectrum, and $u(\hat{\mathbf{y}}_t)$ the corrected spectrum with prediction $\hat{y}_t$ at time step t.

Evaluation protocol The eDR method was tested *in situ* on different samples and a set of 100 random distortions $\mathbf{y}_X, \mathbf{y}_Y, \mathbf{y}_Z, \mathbf{y}_{Z^2} = R \cdot W \cdot \mathcal{N}(0, \frac{1}{3}^2)$ of X, Y, Z, Z^2 shims. The generalization ability of the approach was tested on multiple samples, i.e., 5vol% and 10vol% H_2O in D_2O, respectively, 100% distilled water, and Nicotinamide (CAS No. 98-92-0, 400 mg) in 1 mL D_2O (CAS No. 7789-20-0).

The following samples were used during dataset acquisition and evaluation experiments:

Nr	Content	Dataset	Evaluation
I	H_2O+$CuSO_4$	✓	
II	5vol% H_2O in D_2O	✓	✓
III	10vol% H_2O in D_2O		✓
IV	100% distilled water		✓
V	Nicotinamide (CAS No. 98-92-0, 400 mg) in 1 ml D_2O (CAS No. 7789-20-0)		✓

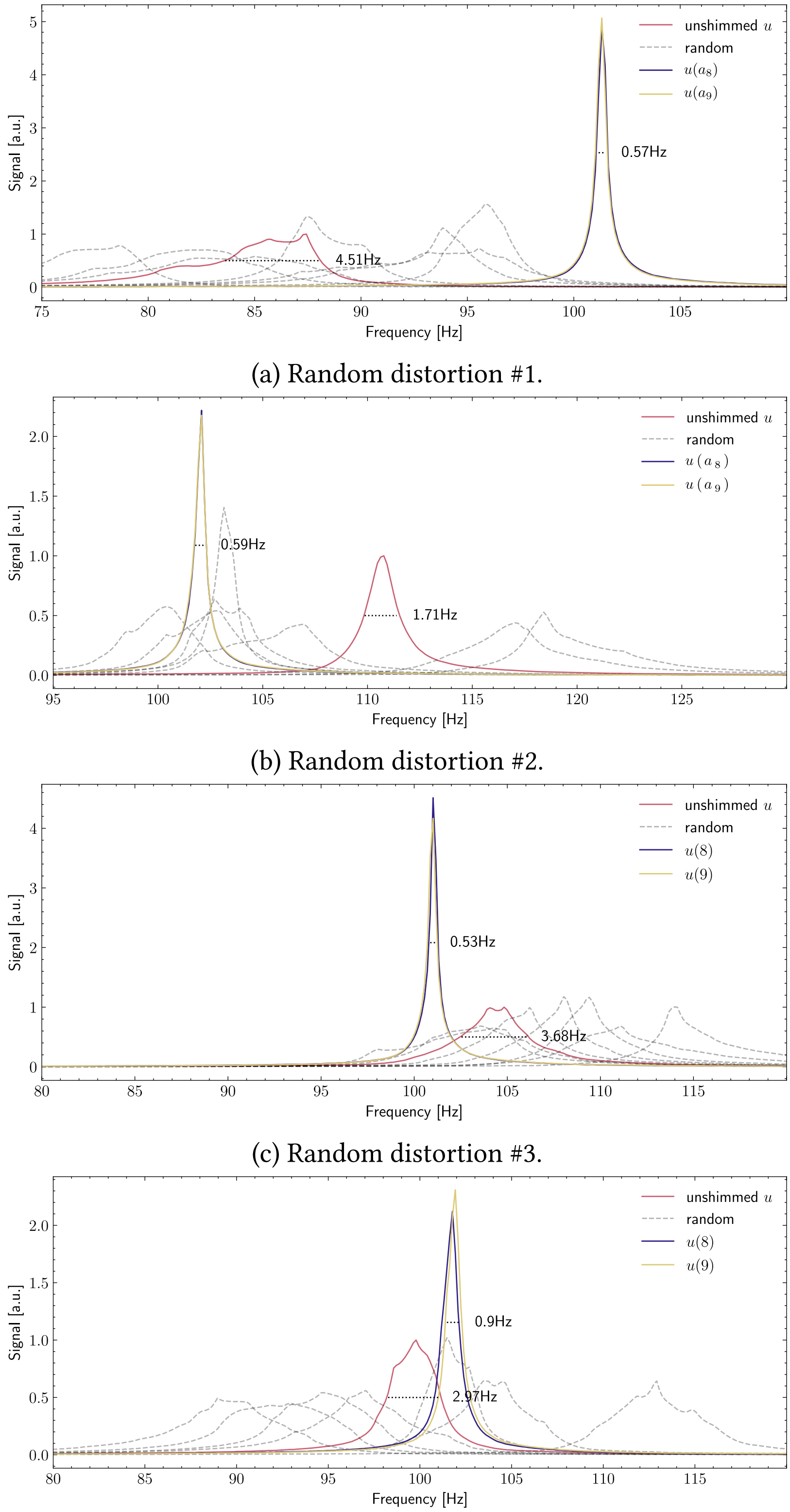
Table 5.2.: Samples used during dataset acquisition and evaluation experiments.

5.5.2. In-situ experiments

H_2O [%]			lw50 [Hz]				lw0.55 [Hz]		
in D_2O	SR	DiR	Best	Initial	Shimmed	+x%	Initial	Shimmed	MAE
5%	0.99	0.93	0.34	3.9 ± 1.6	**0.72 ± 0.28**	+507%	59.7 ± 13	25.4 ± 4	0.041
10%	1.00	0.92	0.38	3.8 ± 1.6	0.76 ± 0.25	+444%	54.9 ± 12	24.2 ± 4	0.049
100%	0.99	0.89	1.63	4.4 ± 1.5	2.25 ± 0.56	+104%	50.8 ± 10	34.0 ± 4	0.084

Table 5.3.: *In situ* results of AI-driven shimming with the eDR method for 7 random steps and 2 predictive steps. Values are reported as mean ± standard deviation over 100 random distortions drawn from a uniform distribution for different H_2O concentration. The best values are marked in bold. Abbreviations: SR = success rate, DiR = direction ratio, lw{50,0.55} = linewidth at {50,0.55}% of the maximum, MAE = mean absolute error.

The results reported in Figure 5.6 show that AI-driven shimming with sequences of random shim offsets yields overall promising results. Distorted spectra were shimmed to



(a) Random distortion #1.

(b) Random distortion #2.

(c) Random distortion #3.

(d) Random distortion #4.

Figure 5.6.: **Visualization of selected shimming results of the eDR method** with $r = 7$ random and $p = 2$ predictive steps on the reference sample. The unshimmed spectrum $u(a_t)$ changes with the actions a_t over $t = \{1, ..., r + p + 1\}$ steps. (Subfigures a and b adopted from Becker et al. (2022b).)

well-defined peaks using four shims simultaneously. Random offsets from the unshimmed spectrum u show similarly bad lineshapes but help the model to predict shim corrections close to the global optimum. Table 5.3 gives a qualitative analysis showing that the spectral quality could be improved in 99% of the cases. Within only 9 NMR acquisitions, eDR could reduce the linewidth from ~4 Hz to 0.72 Hz on average, and in 87% of the cases even to below 1 Hz FWHM. Compared to the true distortions, the model's predictions had an MAE error of 0.041, corresponding to 4% discrepancy to the reference shim values for 5% H_2O.

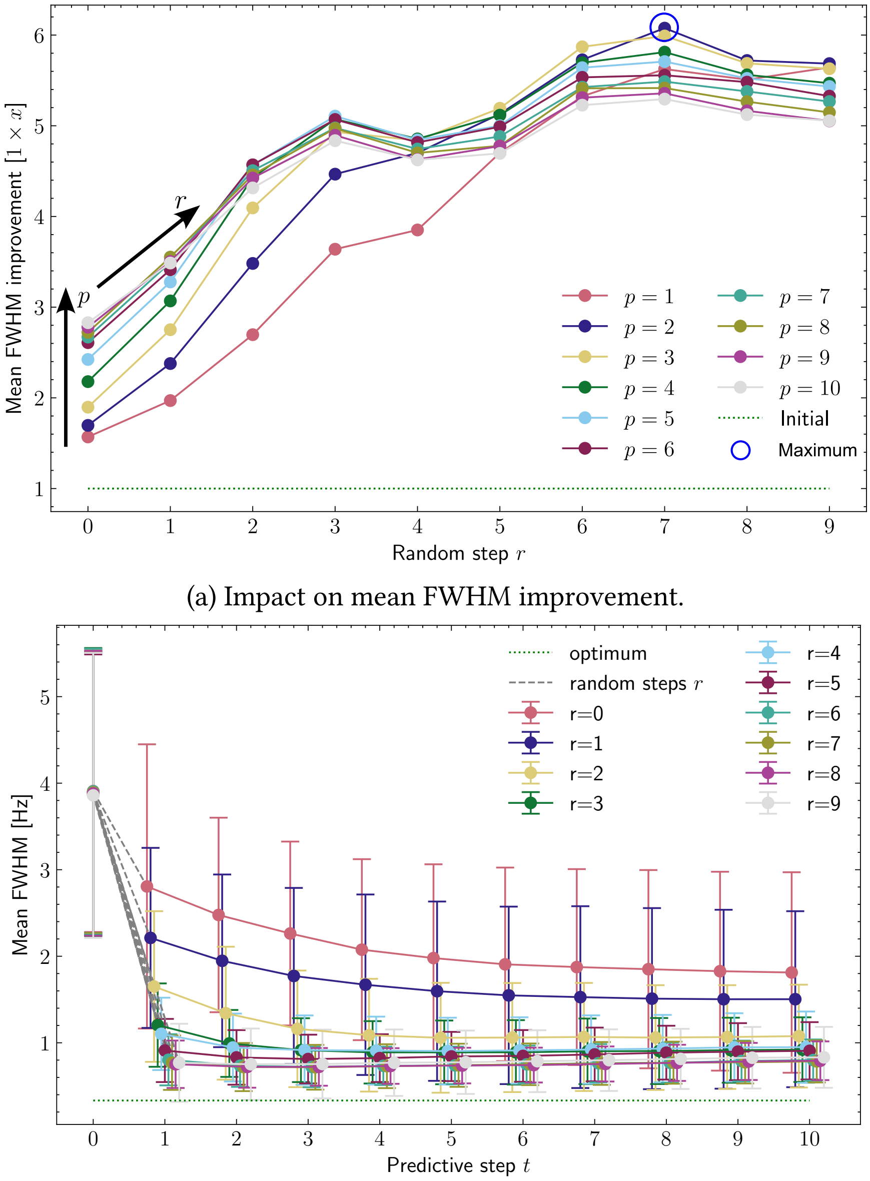
Impact of sequence length As the model was designed to handle variable sequence lengths with both flexible random and predictive steps, their impact was investigated. Overall, the evaluation protocol of Subsection 5.5.1 was repeated for $r = [0, 10]$ random and $p = 10$ predictive steps. Correspondingly, the full width at half maximum (FWHM) was reported in Figure 5.7, averaged over all 100 runs. Figure 5.7a shows the mean FHWM improvement in dependence on r random and p predictive steps. On the other hand, Figure 5.7b visualizes the impact on the absolute FWHM directly. The results showed that $r = 7$ random and $p = 2$ predictive steps yield the best FWHM improvement, a good compromise between predictive performance and the number of total steps. Furthermore, increasing the number of random steps improves the FWHM significantly more than the predictive steps.

In general, this experiment showed that the number of random steps was more critical than the number of predictive steps. This could be explained by the fact the model was trained on random steps only during offline training, and was not prepared to handle predictive steps.

Furthermore, it seemed that the model's predictive performance saturates around the global optimum after $8 - 10$ random steps, where the spectral features are not visible anymore, and the model misinterprets noise in already shimmed peaks.

Shimming performance on other samples The successful shimming results of the method on various samples with different water concentrations indicated that the technique is generally applicable. For instance, the method could shim nicotinamide (vitamin B3) dissolved in D2O, as illustrated in Figure 5.8. It should be noted that the lw50 optima in Table 5.3 differ due to radiation-damping effects in higher water concentrations.

Hardware requirements DL training was conducted on an AMD Ryzen Threadripper 3970X with 256 GB RAM and two NVIDIA GeForce RTX A5000 graphics processing units. The training process took approximately 47 minutes, and the RandomShimDB requires about 5.5 GB of disk space. The trained model has ~ 26M parameters allocating 102 MB of disc space. Inference required 15-50 ms using an Intel Core i5-8500. Compared to acquisition times for the NMR measurement, and storage space available on recent computers, these are negligible requirements. Loading the model and the first step of eDR takes 19 s, and consecutive steps roughly 10 s. Standard proton experiments without time-saving effort were acquired, i.e. full acquisition time 6.5 s for 32k points and a dwell time of 200 µs, which are not necessarily necessary for shimming, and require more time.



(a) Impact on mean FWHM improvement.

(b) Impact on the FWHM directly.

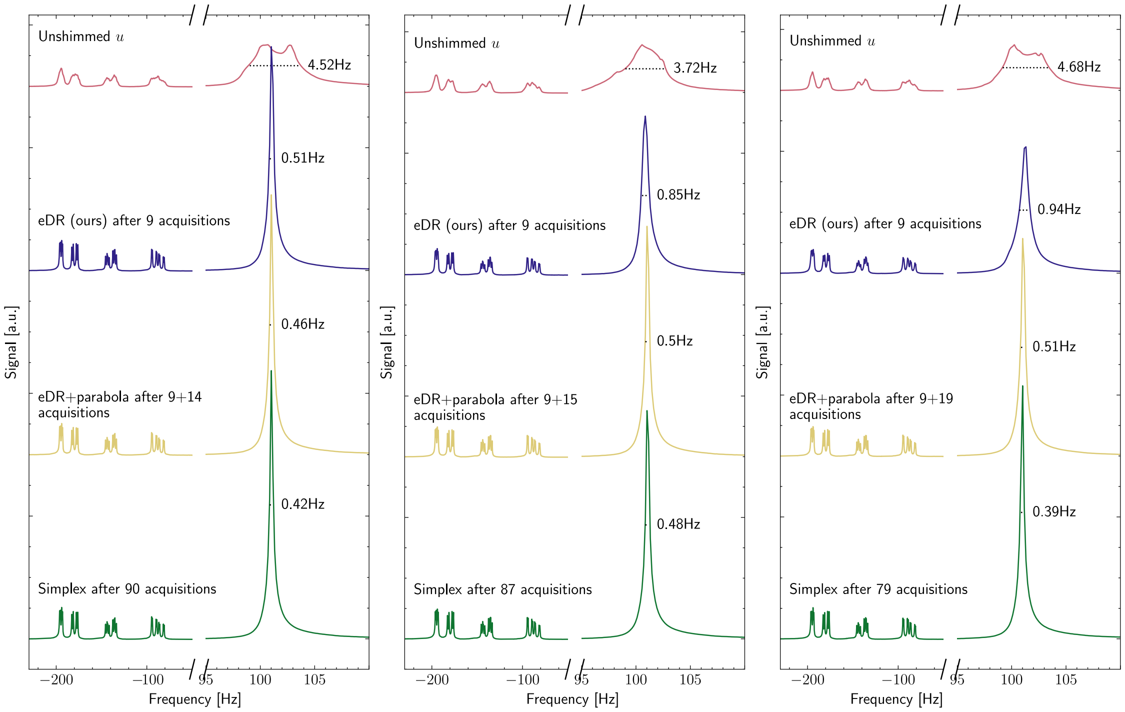
Figure 5.7.: **Influence of random steps r and predictive steps p** of the model's input sequence on the shimmed FWHM. Average values are visualized for 100 random distortions. The green dotted line represents the initial spectrum without any improvement. (Adopted from Becker et al. (2022b).)

5.6. Results - Comparison

To prove that AI-driven shimming was superior, the developed method of this chapter was compared to the default shimming methods available on Magritek devices. This includes the parabola method (see Subsection 2.2.2), and the simplex method (see Subsection 2.2.2).

The following three scenarios were considered. First, the theoretical number of minimal required NMR acquisitions was compared. Then, AI-driven shimming was directly compared to traditional methods, and finally, an AI-assisted approach was considered.

Note that the optimum line width was only achieved after multiple hours of standby shimming, and no method (AI-driven or traditional) could achieve it within its first iterations.



(a) Random distortion #1.

(b) Random distortion #2.

(c) Random distortion #3.

Figure 5.8.: **Shimming comparison** of AI-driven shimming (eDR), a combination of traditional and AI-driven shimming, and the simplex method. eDR shims on the HDO peak of nicotinamide (vitamin B_3) in D_2O, and all methods are run until convergence was achieved. (a) demonstrates an example of AI-driven shimming that does not require fine-tuning, while (b) and (c) show examples for which traditional shimming may be beneficial afterwards. (Subfigures a and b adopted from Becker et al. (2022b).)

Theoretical comparison Both signal-based methods require a minimum number of steps for initialization: The initial simplex shape of the Nelder-Mead method requires $n + 1$ spectra, and the parabola needs at least $3n$ spectra to fit a parabola for each of the n shims. On the other hand, shimming via DL can have a flexible number of acquisitions, demonstrating linewidth improvement after even 1 step and decent performance with below 10 measurements (see Figure 5.7).

AI-driven vs traditional shimming It was compared how many NMR acquisitions the related methods require to reach the linewidth that the eDR method achieved after ($r =$

$7, p = 2$) steps (Table 5.4). The direct comparison (Table 5.4) between eDR and traditional shimming methods shows a significant speed-up of AI-driven shimming. The parabola method needs ~ 6.1 ± 6 more spectra, and the simplex method even needs ~ 3.7× more acquisitions. Both methods fail to reach the linewidth achieved by eDR in some cases, and the parabola even fails in over 65% of the cases. Note that the maximum number of convergence steps for the simplex method was cut off after 150 steps.

Method	NMR acquisitions	Failure
eDR (ours)	**9**	(1%)
Parabola	15.1 ± 6.0	65%
Simplex	33.4 ± 16.7	26%

Table 5.4.: Comparison of necessary NMR acquisitions to reach eDR's shimmed linewidth. Average values ± standard deviation are reported over 100 random distortions if the method can reach eDR's linewidth; otherwise, it was counted as a failure. Failure of eDR itself is 1 − SR. Lower is better, and the best values are marked in bold. Abbreviations: eDR = enhanced Deep Regression.

AI-assisted shimming AI-assisted shimming (Table 5.5) is a promising approach that combines traditional methods with AI-driven shimming: By leveraging eDR to forecast an initial point near the global optimum of shim values, the initialization of the parabola method was notably optimized. This integration not only lowers performance variance but also enables the parabola method to circumvent local minima in 96% of cases. Consequently, AI-assisted shimming achieves linewidths nearly equivalent to the optimal full width at half maximum (FWHM) with just 30% of the steps needed by the simplex method.

Method	NMR acquisitions	FWHM [Hz]
Unshimmed		3.9 ± 1.6
eDR(ours)	**9**	0.72 ± 0.28
Parabola	26.7	0.82 ± 0.29
+ eDR	15.9+9	**0.44 ± 0.05**
Simplex	87.7	0.41 ± 0.04
+ eDR	70.8 + 9	0.41 ± 0.05
Optimum		0.34

Table 5.5.: Combination of traditional and AI-driven shimming methods ("AI-assisted shimming"). Traditional shimming methods are initialized with the eDR method, and then iterated until converging on their own. Average values for necessary acquisitions and FWHM are reported over 100 random distortions. Lower is better, and important values are marked in bold. Abbreviations: FWHM = full width at half maximum.

A graphical comparison between AI-driven, AI-assisted, and traditional shimming on nicotinamide dissolved in D_2O is given in Figure 5.8. The water peak of the (99.9 atom %) D_2O solvent was used for shimming.

The method in this chapter only tackles the most important and, consequently, a relatively small number of shims. Thus, it was not compared to Yao et al. (2021), as they report equal performance between their improved and the standard simplex implementation if a low number of shims is considered.

Comparison to DRE Contrasting with the approach outlined in chapter 4, which involves employing four systematic shim offsets and their corresponding spectra as network input, the model, in this case, achieves only a 1.7-fold improvement in Full Width at Half Maximum (FWHM). This outcome supports the notion that incorporating random steps enhances the model's ability to navigate the shim space more effectively, leading to improved predictions.

5.7. Discussion/Limitations

The modifications proposed in this chapter did effectively enhance AI-driven shimming. Substantial contributions employed a shimming history enabled through recurrent connections in the neural network's architecture, and random sampling for fast and scalable data acquisition.

The experiments have shown that AI-assisted shimming, i.e., initializing traditional 1D optimization methods with an AI-driven approach, avoids local minima and can significantly accelerate shimming.

Still, the scalability to more shims was questionable, as for high-field magnets, sometimes over 28 shim coils are used. Furthermore, the supervised learning nature of this DL approach requires an a priori known reference optimum, and a labelled dataset with machine-specific spectral features.

6. AI-driven shimming of a custom parallel probehead for high-field NMR

This chapter's proposed method was published together with Yen-Tse Cheng in an article titled "Artificial intelligence-driven shimming for parallel high field nuclear magnetic resonance" in the "Scientific Reports" journal in 2023. This chapter's text and figures were reproduced and adapted from the authors' published work.

6.1. Overview

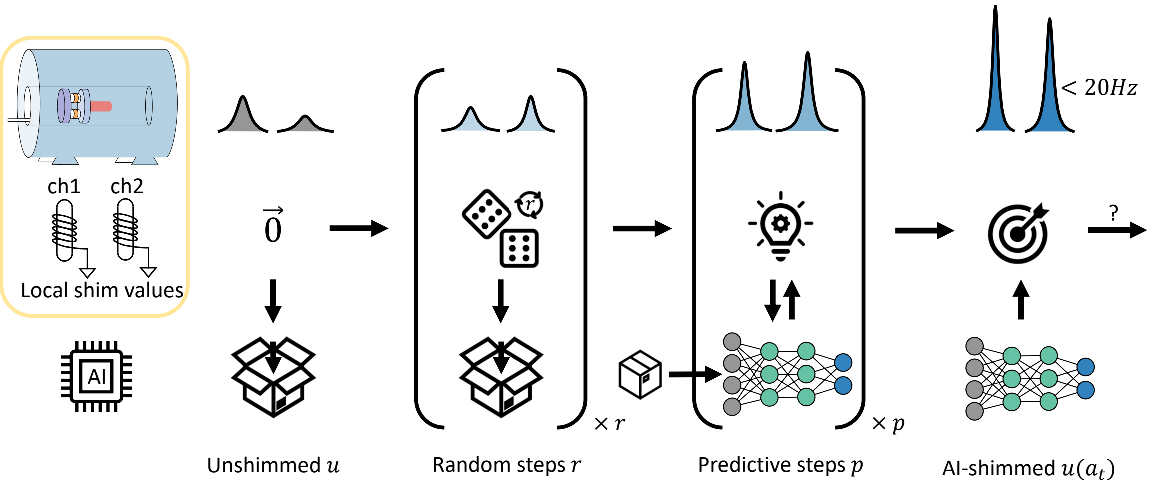


Figure 6.1.: **Graphical abstract of parallel enhanced deep regression (PeDR).** Differences from the other approaches within this thesis are highlighted.

Traditional NMR spectroscopy with a single channel suffers from low throughput due to time-consuming steps such as sample loading or coil tuning. Even though automated flow systems for liquid samples (Nassar et al., 2021) can increase sample loading speeds, the experiments are still limited by their sequential measurement style. Parallel spectroscopy aims to drastically increase throughput by enabling simultaneous experiments on multiple samples with additional channels (Cheng et al., 2022). However, shimming becomes a challenging task when multiple channels are used to acquire signals from different samples in one coherent magnetic field. Unlike single-channel spectroscopy, where a single set of orthogonal shim coils can correct the field inhomogeneity, parallel shimming requires a more sophisticated approach considering shim interactions and RF couplings between the channels. Additionally, custom hardware may exhibit non-idealities due to manual assembly, resulting in non-orthogonal shim fields that complicate the optimization process. Even if the shim coils are orthogonal for each channel taken separately, they may not necessarily be for two or more coils, making classical algorithms such as the simplex method prone to performing many redundant actions.

This chapter summarizes a study on parallel shimming supported by artificial intelligence, i.e., AI-driven shimming of a custom parallel probehead (see Figure 6.1), and the approach is titled **Parallel enhanced Deep Regression (PeDR)**. AI has shown the ability to manage shimming, which has high-dimensional and non-linear complexities. Furthermore, it should take note of the complicated cross-sensitivity among parallel channels. This chapter showed that AI can adjust itself to non-orthogonal shim fields and identify adequate shim settings for every channel. Overall, the following contributions were made regarding AI-driven shimming, and the differences to previous approaches are highlighted in Table 6.1.

	DRE & eDR	**PeDR**
Channels	1	2
Magnet	Permanent	Super-conducting
Vendor	Magritek	Bruker
Probehead	Built-in	Custom
Local RF and shim coils	-	✓

Table 6.1.: PeDR contributions: Differences to previous approaches (DRE and eDR).

6.2. Method - DL part

This section focuses on the deep learning part of parallel AI-driven shimming, starting with a formal problem definition, followed by a description of the modifications to the neural network architecture used for this chapter's approach, and completed with the concept of how the DL approach was applied for shimming.

6.2.1. Problem definition

Let $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_i\}_{i=1}^{|\mathcal{D}|}$ be a static dataset, where $(\mathbf{x}, \mathbf{y})_i$ is an input-output pair. With m being the number of parallel channels, and n the number of separate shim coils, the input sequences $\mathbf{x} \in \mathbb{R}^{m \times t \times (L,n)}$ of t entities are defined as

$$\mathbf{x}_m = \left[\left(u_m(\vec{0}), \vec{0}\right), \left(u_m(a_1), a_1\right), ..., \left(u_m(a_t), a_t\right) \right] ,$$

where the unshimmed spectrum u_m of length L for each channel m changes as a function of (random) shim offsets (or actions) $a \in \mathbb{R}^n$. Each associated target $\mathbf{y} = (y_1, y_2, ..., y_{m \cdot n}) \in \mathbb{R}^{m \cdot n}$ represents the distortion from the reference shim values and is defined as a real-valued vector of $m \cdot n$ elements. The regression model $\mathrm{F}_\theta(\cdot)$, represented by a custom deep neural network with parameters θ, predicts the shim correction terms $\hat{\mathbf{y}} = \mathrm{F}_\theta(\mathbf{x})$, such that $\mathbf{y}_i - \hat{\mathbf{y}}_i \approx 0$. The network parameters θ are learned in a supervised manner using the

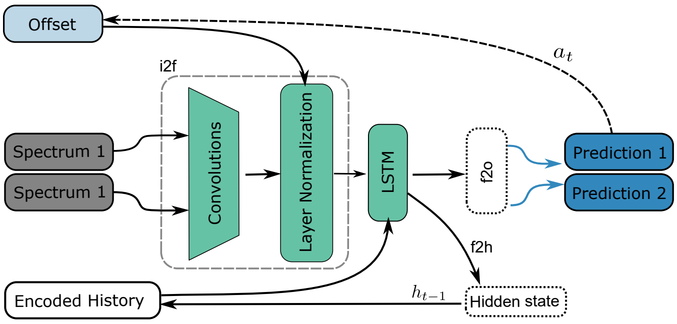


Figure 6.2.: Simplified neural network architecture for PeDR. Given two spectra, processed by convolutional and recurrent layers, the model predicts shim correction values. (Edited from Becker et al. (2023).)

dataset $\mathcal{D}$ to minimize the loss term $\mathcal{L}$ between the prediction $\hat{\mathbf{y}}$ and the target $\mathbf{y}$. Note that the shim distortions S w.r.t. the reference spectrum serve as the labels $\mathbf{y}$, whereas the shim offsets (or actions) w.r.t. the first, unshimmed spectrum are denoted as a_t.

6.2.2. Architecture

AI-driven shimming for parallel spectroscopy followed the same architecture as single-channel shimming with the eDR method (see chapter 5), although two input channels were used for the model. A minimum schematic overview is given in Figure 6.2.

The model consists of (1) a convolutional part that extracts features from the two input spectra at each time step t, (2) layer normalization and concatenation with the past actions a_t w.r.t. the initial spectrum, (3) a long short-term memory (LSTM) that allows recurrency of the model and learning of temporal dependencies in sequences, and (4) an output head (`f2o`) that predicts the shim corrections for the two channels. The convolutional part consists of 3 blocks with convolutions (64 filters, kernel size 41, stride 2), ReLU activation, dropout, and pooling. The normalization was done on the concatenated convolutional feature maps and last actions. The normalized features were fed into the LSTM cell with the last hidden state h_{t-1} to generate a new output and the next hidden state. The `f2o` part consists of a first layer normalization layer, followed by one FC layer with dropout and ReLU, and a final FC layer with tanh activation, which predicts the shim correction values.

6.2.3. Concept

The AI-driven shimming process from chapter 5 was adopted, which consists of an initialization step, two phases with varying steps, and a final wrap-up step (see Figure 6.1). This process begins with acquiring an initial, unshimmed spectrum denoted as u. Afterwards, r random shim offset steps are applied, forming an internal shimming history within the model and assisting the DL model in navigating the shim space. Following this, p predictive steps are undertaken, where the model's output dictates the subsequent shim action a_t. After $t = r + p + 1$ steps, a shimmed spectrum $u(a_t)$ was obtained.

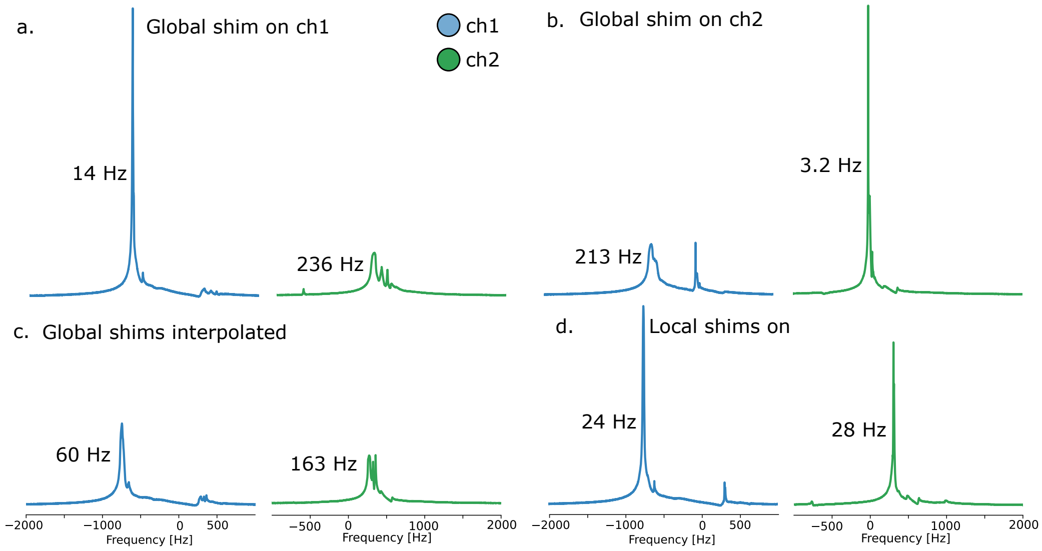


Figure 6.3.: **Necessity of local shim coils.** Global shimming (a,b) leaves the respective other channel with bad linewidths. Only with local shim (d), both channels show decent linewidths. (Edited from Becker et al. (2023).)

During the random step phase, the action a_t was characterized as Gaussian noise $\mathcal{N}$. In the predictive phase, the model uses the previous prediction ($a_t = \hat{\mathbf{y}}_{t-1}$) to generate the next spectrum.

6.3. Method - Hardware setup and dataset

Commercially available shim set systems can largely homogenize the field of standard superconducting magnets, provided that the sample is in a 5 mm or 10 mm tube and is placed at the isocenter of the shim coils. Small sample volumes require precise placement of the shim system at the centre of the sample.

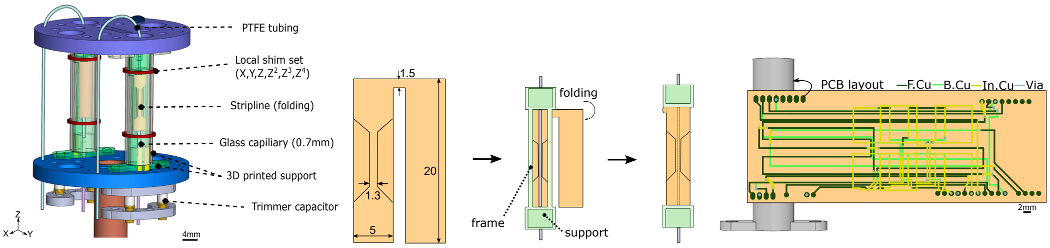
Thus, a parallel scenario requires more sophisticated modifications to the hardware. A custom parallel probehead has been developed based on the concept of an NMR cell (NC) (Cheng et al., 2022).

6.3.1. Custom parallel probehead

A custom probehead is essential, as a conventional single-coil probe cannot accommodate parallel detection on different samples. Furthermore, shimming of such a system cannot be achieved with the global shim system, and thus, local shim coils were required per channel. Figure 6.3 depicts this issue, showing that with global shims (built-in shim set), only a single channel could be shimmed for spectroscopy. Local shim coils allow the shimming of parallel channels independently.

The design and manufacturing of the probehead were performed by Yen-Tse Cheng, the first author of the corresponding publication. A brief overview of essential parts will be given below, but a detailed description in Becker et al. (2023) or Yen-Tse Cheng's PhD Thesis is recommended.

The parallel probehead (see Figure 6.4a) was designed for a 15.2 T preclinical MRI magnet (see Figure 6.5) from Bruker (Bruker, Ettlingen, Germany), and consists of two parallel channels, each equipped with a folded-up stripline RF coil, shim coils, and flow channels for sample loading.



(a) Probehead.

(b) RF coils.

(c) Shim coils.

Figure 6.4.: Hardware of the custom parallel probehead. (Courtesy of Yen-Tse Cheng from Becker et al. (2023).)

Folded-up stripline RF coil The stripline's dimensions are 25 mm in length and 5 mm in width, and it has a sensitive conductor section of 8 mm × 1.3 mm. The coil has a sensitive volume of interest that equals 1.57 µL and a self-resonance frequency of 1.75 GHz. Both RF coils were placed in the probehead, with their B_1 fields arranged perpendicular to the static magnetic field B_0.

Shim coils Each RF coil or channel has a local shim set to correct local field distortions (see Figure 6.4c). To allow for easy fabrication on a single 3-layer flexible PCB, (X, Y, Z, Z^2, Z^3, and Z^4) shim coils were chosen as they possess sufficient shimming capabilities. The shim coils' design was based on finite element simulations.

Flow channels A fluidic tubing was used for each channel's sample loading, allowing continuous high-throughput screening measurements (Nassar et al., 2021).

Custom shim current source Additionally to the local shim coils, an extremely-low drift current source was developed by Achim Voigt (IMT, KIT). The current source featured 28 channels with ±300 mA, from which 12 channels were used for the local shim coils. For a detailed description of the electronic design, see Becker et al. (2023).

6.3.2. Dataset collection

As shown in chapter 3, AI can learn features for shimming due to machine-specific hardware non-idealities. Thus, a real dataset with 8799 samples was acquired with the custom-built parallel probehead. Each dataset sample contains two ^{1}H spectra as a result of random shim distortions S to the reference shim values `Ref` of the local shim coils, following a Gaussian distribution with $\sigma = \frac{1}{3} \times \texttt{Ref}$. The manufacturer's automated shimming algorithm was used to achieve optimal global shim settings for each channel individually. The global linear shim values were then obtained by interpolating between the two NC (see Figure 6.3c), i.e., each cell was shimmed with the vendor's automated shimming routine and the isocenter was linearly interpolated between their optimized shim values. The local reference shim values `Ref` were achieved by manually shimming for several hours. In total, the dataset consists of four subsets with varying numbers of samples acquired within two consecutive days. The measured sample for dataset collection was 50 vol% H_2O in D_2O for both channels. Detailed parameters are reported in Table 6.2.

Dataset parameters	NC channels	2
	Local shim coils	6
	Reference values `Ref`	[0, 90, 140, 14, 0, 60, 8, 80, 20, 22, 0, 0]
	Shim range R	$\pm 2 \times \texttt{Ref}$
	Sample 1 and 2	50 vol% H_2O in D_2O
	Number of spectra per subset	$\{524, 1536, 2738, 4001\} \times 2$
Acquisition parameters	Nucleus	^1H
	Excitation bandwidth	5 kHz
	Recorded FID points	19736
	Repetition time	1500 ms
	Postprocessing	Bruker default

Table 6.2.: Parameters of the parallel randomized shimming dataset (PaRandShimDB).

6.4. Results - Deep Learning

Preprocessing the data and training the DL model followed the same procedure described in chapter 5. Differences are highlighted below.

6.4.1. Data preprocessing

Dataset creation For the offline deep learning training, the splitting ratios of the train, validation, and test set were 80/10/10 %, respectively. As was done in chapter 5, sequences were constructed on the fly during training. All steps were assumed to be random offsets from the initial unshimmed spectrum $u(0)$.

Normalization Normalization of input spectra of sequence $\mathbf{x}$ and regression target y followed the same scheme as chapter 5, i.e., normalization to $[0, 1]$ for input and $[-1, 1]$ for output. The only difference was that all spectra were cut to a region of interest (ROI) of size 4096 and then downsampled by a factor of 2 to yield size 2048.

Augmentation The input spectra were shifted randomly with a uniform distribution between −4 and 4 array indices. To simulate real-world situations, different types of noise were added, including uniform label noise of 0.1, uniform shim interaction noise of 0.1, first-order phase distortions of ±0.5, and additive white Gaussian noise (AWGN) with a signal-to-noise ratio (SNR) of 30. This process was similar to the one described in chapter 5.

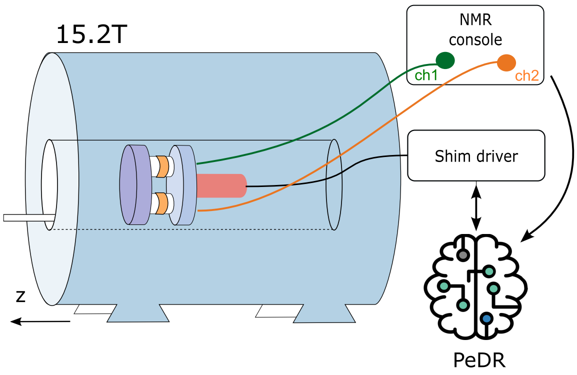


Figure 6.5.: Schematic setup of parallel spectroscopy and AI-driven shimming. (Adopted from Becker et al. (2023), created by Yen-Tse Cheng.)

6.4.2. Training details of a ConvLSTM for parallel shimming

The model was trained for 100 epochs with a learning rate of 10^{-4} (reduced on plateau by an automated scheduler), a batch size of 256, and the `Adam` optimizer to minimize the Huber loss. The sequence (or shim trajectory) length during training was increased by 2 for every 25 epochs, ranging from 4 to 10.

6.4.3. DL training results

The model achieved a normalized MAE ($\in [0, 1]$) of 0.033 on the test set of the static dataset $\mathcal{D}$.

Hardware requirements DL training was conducted on an AMD Ryzen Threadripper 3970X with 256 GB RAM and two NVIDIA GeForce RTX A5000 GPUs. The PaRandShimDB requires about 5.1 GB of disk space, and the trained model allocates 331 MB of disc space.

6.5. Results - Experimental

6.5.1. In-situ spectroscopy experiments

Spectroscopic capabilities of the parallel probehead were tested on homonuclear and heteronuclear scenarios, namely

1. homonuclear 1D ^{1}H measurement, i.e. a 0.4 M Niacinamide solution in channel 1 and 17.4 M Acetic acid in channel 2.
2. homonuclear 1D ^{1}H measurement, with a 0.16 M D-(+)-maltose solution in channel 1 and a 0.3 M Tris(hydroxymethyl)aminomethane acetate (Trizma) solution in channel 2.
3. heteronuclear 1D ^{1}H & ^{19}F measurement, where samples were water in channel 1 and FC-770 ($C_{10}F_{22}$) in channel 2.

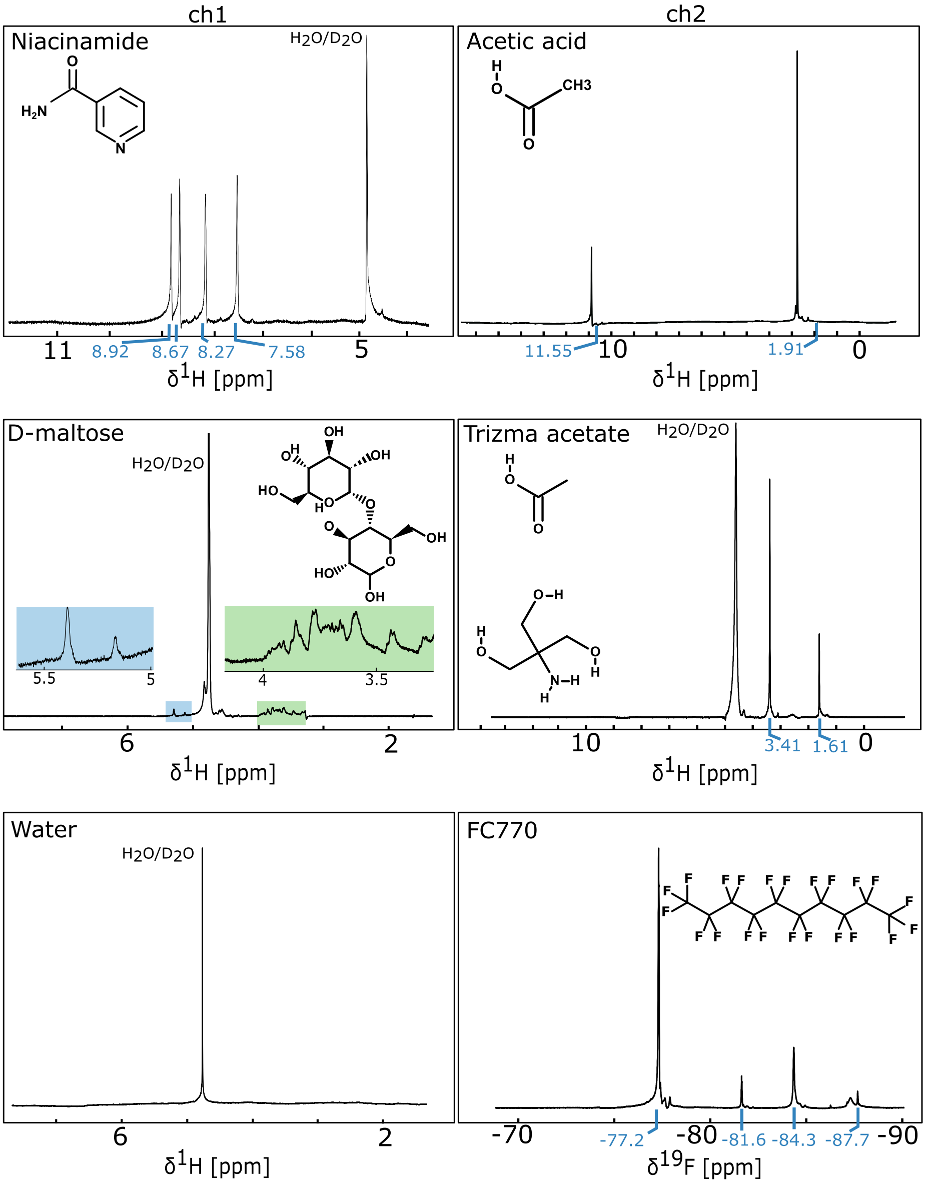


Figure 6.6.: Spectroscopic experiments on a custom parallel probehead in 15.2 T. (Adopted from Becker et al. (2023), created by Yen-Tse Cheng.)

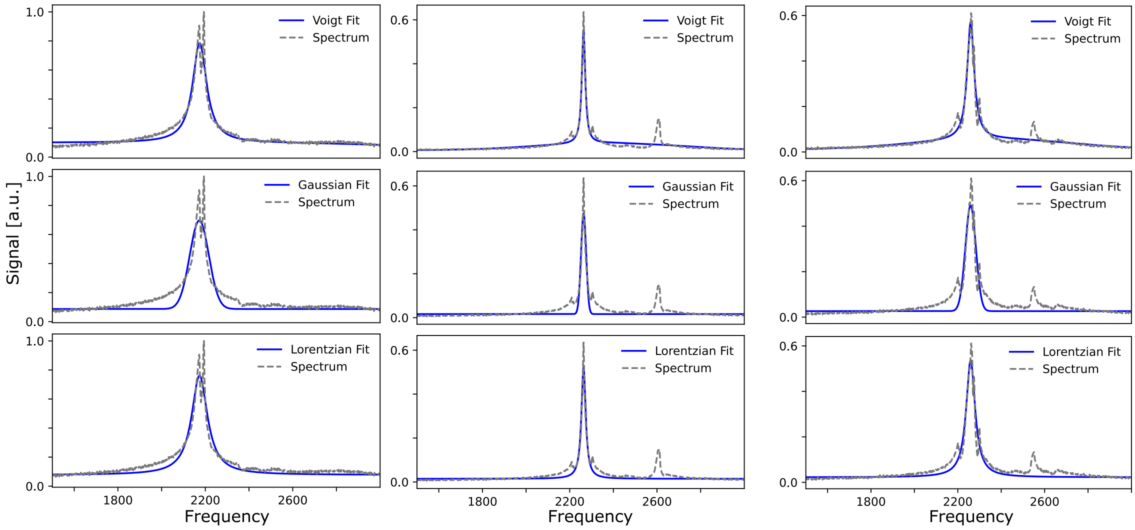
During the experiments, 256 scans were performed, and the resulting spectra are averaged and displayed in Figure 6.6. However, due to the RF coupling between the two RF channels, which mainly occurred due to the RF coils and coaxial cables used, the raw spectra had to be post-processed by signal subtraction. Despite this, all the necessary main peaks were resolved and detected in the measurement results.

It is worth noting that homonuclear parallel NMR can cause signal coupling between the two channels due to simultaneous excitation and signal reception at the same frequency. However, true simultaneous parallel spectroscopy without coupling signals between the striplines can be achieved in heteronuclear mode by tuning them to different frequencies, such as ^{1}H and ^{19}F.

6.5.2. DL performance metric and experimental evaluation protocol

Metrics Similar to chapter 4 and chapter 5, the linewidth at 50% (FWHM), the mean absolute error (MAE), direction ratio (DiR) and success rate (SR) were reported.

Due to severe B_0 field inhomogeneities around the parallel channels, the spectral peak could experience splitting, and thus, the linewidth was measured on a Voigt line fit (Marshall et al., 1997) for split peaks. For a visualization of the Voigt fit on split peaks, refer to Figure 6.7. As a reminder, the custom metric direction ratio DiR $\in [0, 1]$ indicates whether the predictions' and targets' signs match, and the success rate SR $\in [0, 1]$ equals to 1 when the spectral peak intensity increased for both channels.



(a) Example 1. (b) Example 2. (c) Example 3.

Figure 6.7.: Examples of a Voigt fit (top) to spectra with split peaks. Gaussian (middle) and Lorentzian (bottom) fits are visualized for comparison.

Evaluation protocol Parallel shimming performance of the PeDR method was evaluated over 50 random distortions $\in 2\times$ the reference values obtained by manual shimming. The samples in channels 1 and 2 were 50 vol% H_2O in D_2O. Furthermore, generalization was tested with shimming on acetic acid in channel one (ch1) and water in channel two (ch2) to mimic a high-throughput scenario with different solvents per channel.

6.5.3. In-situ shimming experiments

The AI-driven shimming method PeDR could, within only 10 NMR acquisitions, successfully and simultaneously shim distorted spectra from two parallel NMR channels with six shims each, to well-defined lineshapes. Evaluated over 50 random distortions (Table 6.3), PeDR was able to shim from 93 ± 142 Hz for channel 1 and 91 ± 102 Hz for channel 2 to 39 ± 19 Hz and 26 ± 20, respectively. This corresponds to a relative linewidth improvement of +139% and +436%, drastically reducing the standard deviation. PeDR achieved an 88%

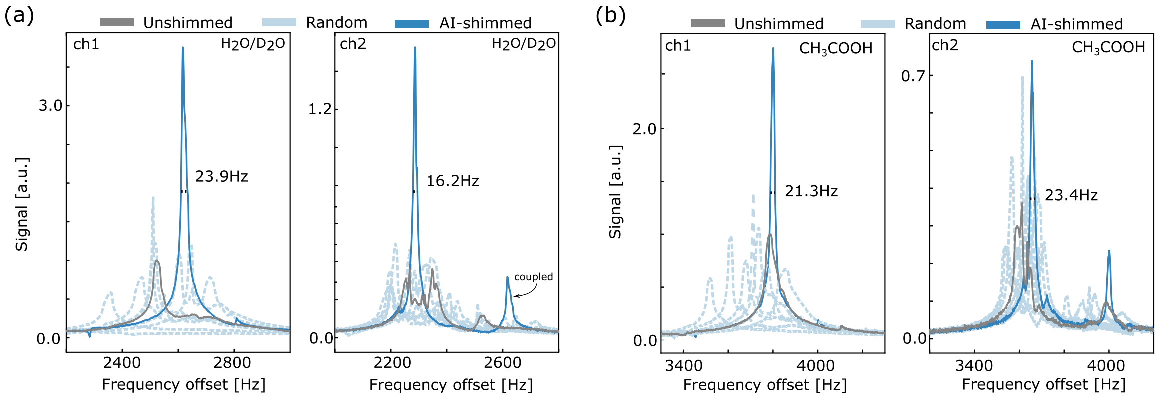


Figure 6.8.: **Exemplary results of parallel AI-driven shimming.** Visualization (a) on HDO in channels one and two (ch1, ch2), and (b) on acetic acid in both channels. (Adapted from Becker et al. (2023).)

success rate, a DiR of 0.9, and a total MAE between real distortions and predictions of 0.089.

The shimming process of PeDR is visualised in Figure 6.8, starting from the unshimmed spectrum (grey). Random shim offsets lead to distorted spectra (light-blue), fed into the DL model to predict a shim correction term. The predictions yielded shimmed spectra (blue) after one predictive step. Similar to the findings of chapter 5, random steps were more crucial for shimming performance than predictive steps.

Generalization of the AI-driven shimming approach was demonstrated by shimming the X-OH peak of acetic acid. Evaluated over the same 50 random distortions as above, a mean success rate of 78%, DiR of 0.85, and an MAE of 0.112 were achieved. The performance was slightly worse than for water in both channels, which was unsurprising as the dataset consists of HDO spectra only, leaving the model uncertain in its predictions on other samples.

section 8.5 discusses the reliability of shimming, and Figure 8.6 shows a clear trend in the distributions between unshimmed and shimmed linewidths for the experiments. Furthermore, if a more diverse dataset was available, the model should generalize well to any single peak in the spectrum, i.e., a reference TSP peak.

6.6. Results - Comparison

Recent chapters have demonstrated that AI-powered shimming can produce well-shimmed spectra with only a few acquisitions. Specifically, four and nine spectra for DRE and eDR, respectively. Traditional signal-based methods, such as simplex (explained in Subsection 2.2.2) and parabola (described in Subsection 2.2.2) methods, are typically slow and require initial acquisitions before optimization can begin. However, implementation of these algorithms was not available for the Bruker device used in this study for parallel spectroscopy, so only a comparison with the minimum number of initialization acquisitions was feasible.

Sample			Channel #1 - lw50 [Hz]			Channel #2 - lw50 [Hz]			
	SR	DiR	Reference	Initial	Shimmed	Reference	Initial	Shimmed	MAE
D_2O	.88	.90	18.1	93 ± 142	**39 ± 19**	7.4	91 ± 102	**26 ± 20**	0.089
AA	.78	.85	16.9	73 ± 51	52 ± 26	8.7	85 ± 91	30 ± 23	0.112

Table 6.3.: *In situ* results of automated shimming of the PeDR method for 7 random steps and 3 predictive steps for D2O in H2O and acetic acid (AA). Values are reported as mean ± standard deviation over 50 random distortions drawn from a uniform distribution. If the peak shows splitting, the linewidth was measured on a Lorentzian/Voigt fit. The best values are marked in bold. Abbreviations: SR = success rate, DiR = direction ratio, lw{50,0.55} = linewidth at {50,0.55}% of the maximum, MAE = mean absolute error.

Nevertheless, a theoretical comparison of the minimum number of acquisitions for both methods was conducted. The parabola and simplex method scale with the number of channels m and the number of shim coils n, and both methods need a minimum number of spectra to initialize. Parabola initialization requires $m \cdot n \cdot 3$ acquisitions. Provided that the first guess does not need iterations, and the best value lies at the parabola's minimum, 1 acquisition per shim coil is required to check the resulting linewidth. For the present case, with two channels and six shims each, this would total 48 acquisitions. To initiate a simplex polytype, $(m \cdot n) + 1$ spectra are required, i.e., 13 acquisitions in the current scenario. Then, each optimization step takes 2.5 acquisitions, on average. Due to the simplex method's slow convergence speed (Lagarias et al., 1998) (up to 90 steps in chapter 4), more than 200 additional acquisitions would be required for shimming.

As mentioned in this chapter, the proposed parallel AI-driven shimming technique demonstrated runtimes below the specified thresholds. Moreover, it was possible to predict shim corrections near the global minimum region by performing only 10 NMR acquisitions.

It's worth noting that manual adjustment of shims to the reference values for dataset collection required approximately two hours. This highlights the impracticality of manual shimming for parallel spectroscopy. The inefficiency of manual shimming is not only due to the time-consuming nature of the process but also because it becomes increasingly unmanageable as the number of parallel sites grows.

6.7. Discussion/Limitations

In conclusion, the proposed work represents a significant advance in parallel NMR spectroscopy, as it combines parallel high-field detection and AI-driven shimming in a compact and scalable probehead design. The system can potentially enable high-throughput compound screening and analysis via NMR, with applications in drug development, metabolomics, and material science.

Scaling up the number of channels and shim coils is recommended to improve the system's performance. However, the study found that using six local shim coils was sufficient for shimming the major field inhomogeneities, while also being easy to manufacture on a 3-layer PCB. Additionally, a combined shimming approach that incorporates both local and all global shim coils could significantly enhance the resolution to the point of achieving spectroscopic linewidths.

The study utilized a water-only dataset comprising 8,000 samples to concentrate solely on evaluating the effectiveness of the AI-driven shimming approach in a controlled environment, without the complexities that could arise from other types of samples. Additionally, the relatively small dataset helped to avoid costly data collection while still permitting exploration of the model's behaviour and limitations. Future research could broaden the dataset to enhance its generalizability.

7. Towards deep reinforcement learning for shimming

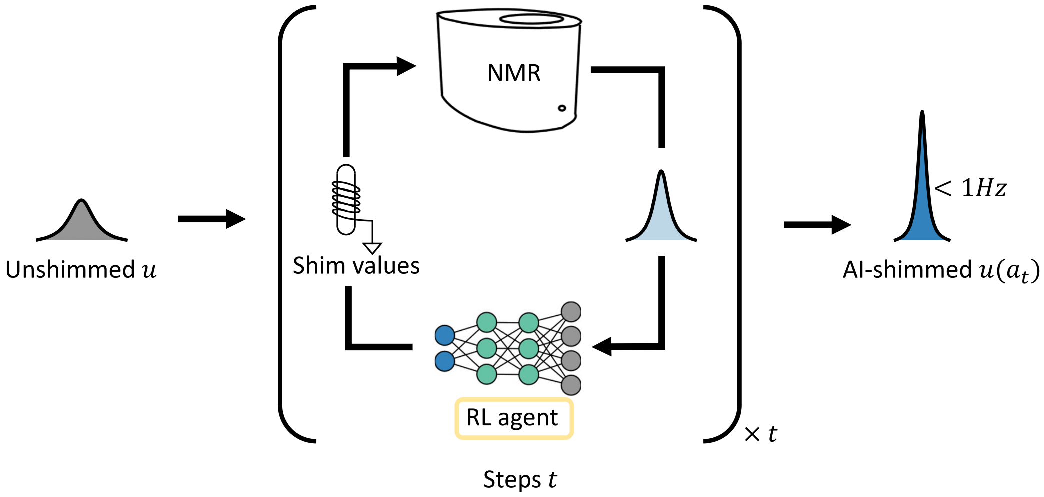


Figure 7.1.: **Graphical abstract of shimming with a deep reinforcement learning (DRL) agent.** The agent interacts with the environment by taking shim actions and receiving corresponding spectra and a reward or punishment until a given tolerance threshold (e.g. linewidth) is reached. Differences from the other approaches within this thesis are highlighted.

Recent studies (chapter 4, chapter 5, chapter 6) have showcased the significant potential of AI-driven shimming using supervised deep learning techniques. However, these approaches necessitate the acquisition of a labeled dataset with labels defined relative to a pre-established optimum. Additionally, they do not inherently optimize for a direct quality criterion, such as spectral quality. Furthermore, even the automatic shimming routines available in commercial systems can fail for applications such as micro samples, which raises a persistent need for time-consuming manual shimming.

In light of these limitations, this chapter explores shimming using deep reinforcement learning (DRL), as outlined in Figure 7.1. The objective is to develop an AI-driven shimming method that autonomously navigates its environment to maximize a predefined reward, explicitly focusing on enhancing spectral quality.

To achieve successful results in shimming, the application of DRL relies on numerous critical factors: the design of a suitable environment that accurately represents the shimming process, the definition of clear and effective action and state representations, the careful shaping of the reward function to align with the spectral quality goals, and the selection of an appropriate RL agent or algorithm capable of efficiently learning and making decisions in this context.

7.1. Environment setup

Essential aspects of custom RL environments include appropriate Markov Decision Process (MDP) (or Partially Observable Markov Decision Process POMDP) properties (see Subsection 2.3.5), and other design choices such as proper reward function shaping, appropriate state representations, initial state distributions, reward, and episode termination procedures (Reda et al., 2020).

RL problems must fulfil the Markov property, i.e., the next state can only depend on the current state and action. In the case of partially observable MDP (POMDPs), a state can be defined as a fixed history of observations, including actions, and still fulfil the Markov property (Sutton & Barto, 2018).

The following environments were designed and considered for RL-driven shimming.

Simulated environment A digital twin of the shimming problem was utilized to rapidly prototype different RL algorithms, namely a gym(nasium) wrapper of the SHIMpanzee simulator (see Subsection 3.5.1). `gymnasium` (Brockman et al., 2016), formerly OpenAI `gym`, is a common playground and API standard for RL research problems, and allows for an easy interface to most state-of-the-art RL methods.

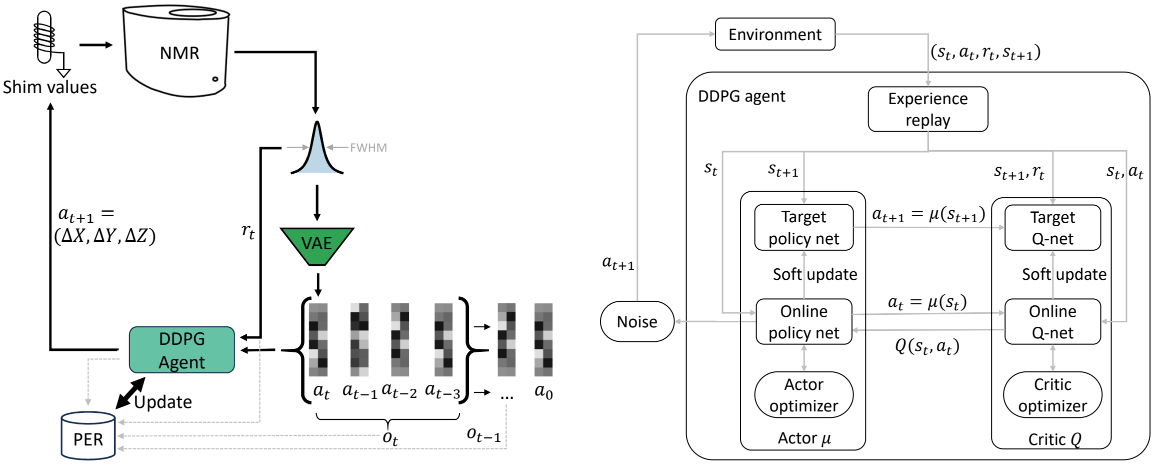
Real hardware environment A dummy `gymnasium` wrapper enabled RL on real hardware, which interfaced a Magritek 80 Carbon spectrometer. The implementation was modified such that the spectrometer's software returned a new state (compressed spectrum) and reward (current linewidth) given an action (shim values). As described below, the `gymnasium` wrapper furthermore internally included a VAE for state compression.

7.1.1. State representation

Four compressed NMR spectra were merged into one observation to extract more features, which was necessary during the previous chapters. The set was organized as a queue, meaning that each new observation was added to the end, shifting the existing content. This allowed for a more comprehensive data analysis, providing a more detailed understanding of the underlying chemical composition.

State space compression Training an RL agent with raw images or high-dimensional input data has additional disadvantages, namely slow learning speed for the model (actor and critic in RL), and high computational loads such as memory footprint and consequentially high bandwidth demands. Kendall et al. (2019) used a variational autoencoder (VAE) to compress images before training their RL agent for autonomous driving, and this idea seemed useful for high-dimensional NMR spectra.

Spectra compression was tested in two different scenarios before being applied to RL. Both studies are described in section 8.1.



(a) Experimental RL setup. Observations are represented by a history of four compressed spectra. The predicted actions are the delta corrections for each shim.

(b) Internal structure and update paths of the DDPG agent. In the shimming scenario, the states s_t are given as observations o_t due to POMDP constraints.

Figure 7.2.: RL setup for real experiments, and internal structure of the DDPG (Deep Deterministic Policy Gradient) agent with Prioritized Experience Replay (PER).

Consequently, instead of feeding NMR spectra to the RL agent, the spectra were compressed to a latent vector with the encoder part of a convolutional VAE and then provided to the agent to support training convergence. By using a pre-trained VAE to extract essential features of spectral peaks, the RL agent's internal actor and critic models were relieved from learning this task, and could directly optimize their networks in correspondence to the expected rewards (see Figure 7.2a). Thus, the observation was represented as:

$$o_t = \Big[\big(\mathrm{E}_\theta(u(a_{t-3})), a_{t-3}\big), \big(\mathrm{E}_\theta(u(a_{t-2})), a_{t-2}\big), \big(\mathrm{E}_\theta(u(a_{t-1})), a_{t-1}\big), \big(\mathrm{E}_\theta(u(a_t)), a_t\big) \Big], \quad (7.1)$$

where $\mathrm{E}_\theta(\cdot)$ was the pre-trained VAE encoder that mapped the unshimmed spectrum $u(\cdot) \in \mathbb{R}^L$ of size $L = 2048$ to a latent vector $\in \mathbb{R}^{16}$, and $a_t \in \mathbb{R}^n$ was the shim actions at step t for n shims.

Initial state distribution The initial observation was represented by the unshimmed spectrum without any shim offset, repeated four times, i.e., $o_t = \Big[\big(\mathrm{E}_\theta(u(\vec{0})), \vec{0}\big) \times 4 \Big]$.

Terminal state An agent was defined to be done when the reward exceeded a given tolerance threshold (e.g. linewidth). The corresponding terminal state was the following spectrum, however, the next Q-value of the target critic must be defined as zero before being saved in the PER memory.

7.1.2. Action representation

The agent's shimming capabilities depend on the representation of the action space, which affects the convergence speed and impacts the solution quality. The action space was set to be continuous to reflect the multi-target regression nature of shimming, i.e. the action at step t was represented as a vector $a_t \in \mathbb{R}^n$ for the n available shim coils.

Furthermore, the action was always defined with respect to the last spectrum $u(a_{t-1})$, in contrast to the DL definition in previous chapters that described it with respect to the first unshimmed spectrum $u(\vec{0})$.

Lower action space During several testing runs, it was observed that the agent tried to reach the terminal state within one step instantly; thus, it did not try to improve the Area Under Curve (AUC) of the FID. This corresponds to strong greedy exploitation behavior, and no exploration occurs. Accordingly, and to force the agent to improve the criterion of choice, the maximum possible step size was reduced with a scaling factor $\lambda_a \cdot a_t$ such that the agent needed to improve the reward step by step.

7.1.3. Reward function

Designing the reward function is one of the most crucial and challenging aspects of an RL algorithm. It requires extensive hand-crafting and testing as it guides the agent's learning, and improper design could lead to the agent exploiting design faults in the reward system.

A reliable way to evaluate the uniformity of a magnetic field is by measuring the Area Under Curve (AUC) of the Free Induction Decay (FID), calculated via $\mathrm{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{L}\mathrm{FID}^2}$ for an FID of length L, and it is the main objective for high-resolution NMR measurements. This is because field inhomogeneity is reflected by low T_2^*, which also directly gives the linewidth $\Delta\nu \approx \frac{1}{\pi T_2^*}$. Each sample's global maximum FID area can further be calculated via its T_2 time, e.g., measured by a nutation experiment.

However, optimizing only for the direct measure of the FID area alone did not guarantee convergence, and the reward was additionally shaped. The following modifications were added:

Log reward Inspired by Kaiser et al. (2022), a logarithmic reward was used to increase the rewards when approaching an acceptable state. The agent, therefore, learned to improve the reward more than just trying to hit the termination state, which had no specific reward assigned. Additionally, the reward was clipped to $[-100, 100]$.

$$r_{t+1} = log_e\left(\frac{|\mathrm{RMS}_t - \mathrm{RMS}_b|}{\mathrm{RMS}_b}\right) - log_e\left(\frac{|\mathrm{RMS}_{t+1} - \mathrm{RMS}_b|}{\mathrm{RMS}_b}\right), \tag{7.2}$$

where RMS_b was the RMS of the optimum FID area, RMS_t the current and RMS_{t+1} the next criterion.

Using traditional metrics, such as the MSE, would yield vanishing rewards near the optimum.

Double negative punishment Negative rewards were multiplied with the bad action scale factor λ^{-}. Whenever the agent took an action that gave a negative reward, this reward was doubled ($r = \lambda^{-} = 2 \cdot r$) to stress the badness of this action.

Out-of-bounds punishment As high shim currents could damage the hardware, predicted shim values outside the allowed range were punished by an OOB factor λ_{OOB}. The punishment intensity depended on the degree of surpassing a_{max}:

$$r = \lambda_{\text{OOB}} \cdot \sqrt{\frac{a - a_{max}}{a_{max}}^{2}} \cdot r \tag{7.3}$$

7.2. Deep reinforcement learning agent

The shimming agent was based on the Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015), and the baseline implementation from Achiam (2018), with prioritized experience replay (PER) (Schaul et al., 2015) memory (see Figure 7.2b). The agent took the observation o_t, a history of four compressed spectra and actions, to predict the following action a_{t+1}. Internally, the actor and critic models were multi-layer perceptrons (MLP) with four fully connected linear layers with 512 hidden neurons, ReLU activations in between, and a tanh activation in case of the actor's last activation. Their weights were initialized with Glorot initialization (Glorot & Bengio, 2010). Both networks had twin target networks, updated using a soft update strategy governed by the parameter τ, which ensured a gradual convergence towards the main networks' weights.

Furthermore, a prioritized experience replay (PER) memory was used, a strategic refinement to the standard experience replay mechanism. By prioritizing each experience based on its temporal difference error, PER ensures a higher replay probability for experiences with greater learning potential. This method improves sample efficiency and mitigates the risk of catastrophic forgetting by frequently revisiting rare but pivotal experiences. For clarification, PER's control variables α and β balance exploring new experiences and exploiting known experiences, where α adjusts the level of prioritization in sampling experiences, and β manages the bias correction to ensure that the learning process remains stable and effective.

7.3. Experiments and results

The experiments as described in Table 7.1 using DRL for shimming were conducted.

7.3.1. RL in simulation

For rapid prototyping and testing different environmental and agent settings, the initial shimming experiments with DRL were executed in simulation. RL in simulation was conducted with the DDPG agent described above, and with the hyperparameters reported in Table 7.2. The overall exploration noise decay was defined via $(\text{explore noise start})^{(\text{episode})}$.

Training	Data type	Shims
Offline	Simulated	Z, Z2, Z3
Online	Real	X, Y, Z

Table 7.1.: Overview of conducted deep reinforcement learning (DRL) experiments for shimming.

Target	Variable	Value
Architecture	Compressed size	16
	Compression training data	Real
	FC layers	4
	FC hidden size	512
	Replay buffer length	100k
Environment	Concatenated observations	4
	PER α	0.9
	PER β	0.9
	Action scale λ_a	0.3
Reward	Objective	FID area
	Target tolerance	0.92
	Discount factor γ	0.55
	Delayed target update τ	10^{-2}
	OOB punishment λ_{OOB}	-10
	Bad action scale λ^-	2
Training	Episodes	4000
	Critic LR	10^{-3}
	Actor LR	10^{-4}
	Batch size	128
	Explore noise start	0.99
	Max steps	20

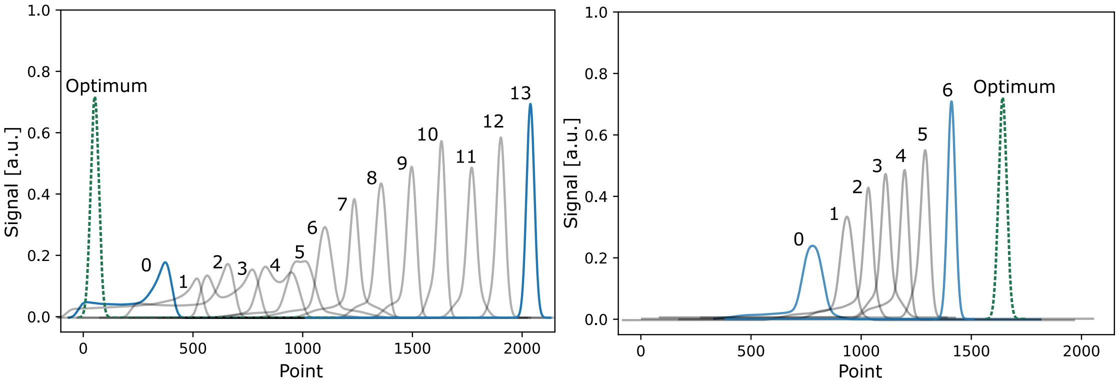
Table 7.2.: RL hyperparameters in simulation.

After training the agent, evaluation was done on 100 random distortions. For two simulated shims, the agent could achieve the tolerance within 3.61 steps, on average, and for three simulated shims, the agent required roughly 8 steps. The results are summarized in Table 7.3. Two exemplary shimming results on 3 shims are displayed in Figure 7.3, where each step is displayed. In Figure 7.3a, the RL agent reached the target region within 10 steps, and in Figure 7.3b, within 6 steps. Furthermore, Figure 7.3 shows that the agent needs more steps to locate itself in shim space (steps 0-5) and corrects a prediction error in step 11.

Hardware requirements Training in simulation (2-shim scenario) takes roughly 4 hours using an AMD Ryzen Threadripper PRO 3955WX with 256 GB RAM and two Nvidia RTX

Shims	Mean nr. steps	Mean reward	MAE
2 (Z, Z^2)	4.5 ± 4.1	2.3 ± 1.5	0.017 ± 0.008
3 (Z, Z^2, Z^3)	8.1 ± 5.5	1.45 ± 0.78	0.035 ± 0.015

Table 7.3.: Simulation results after training an agent for 4k episodes on the SHIMpanzee gym wrapper. The average number of steps, mean reward and the normalized MAE are reported for simulation with 2 and 3 shims.



(a) Example 1.

(b) Example 2.

Figure 7.3.: Two exemplary RL shimming results on Z, Z^2, Z^3 shims in simulation. Peaks are shifted for visualization, and axes are arbitrary units.

A5000 GPUs. The trained actor (without the critic, as this is not required for inference) roughly allocates 2.2 MB of space.

7.3.2. Online RL on real hardware

After confirming the success of RL for shimming in a simulated environment, this approach was applied to an actual spectrometer. All findings during the simulation were incorporated. However, modifications were made to the reward function and the shims. Instead of using the RMS of the FID area (see Equation 7.2), the reward was now defined with respect to the optimal linewidth achievable on the spectrometer, expressed as $1 - \text{lw}$. This implies that linewidths greater than 1 Hz get a negative reward. Additionally, the corresponding tolerance was adjusted to assess the feasibility of RL for shimming under less stringent conditions. The shims of interest were also changed to X, Y, Z, replacing the Z, Z^2, Z^3 configuration, a modification motivated by the optimality of the simulation.

All hyperparameters for training the online RL agent are reported in Table 7.4. Online training of the RL agent required 26′272 NMR acquisitions; thus, the process was time-consuming compared to supervised DL training. The total training time took roughly three days and three hours using an Intel Core i5-8500 with 8 GB RAM interfaced with the Magritek 80 spectrometer. Indeed, RL is known to be sample inefficient, especially when trained online. Nevertheless, the RL agent managed to explore the shim space on its own to exploit it for shimming.

Target	Variable	Value
Architecture	Compressed size	16
	Compression training data	Real
	FC layers	4
	FC hidden size	512
	Replay buffer length	100k
Environment	Concatenated observations	4
	PER α	0.9
	PER β	0.9
	Action scale λ_a	0.3
Reward	Objective	1-lw
	Target tolerance	0.5
	Discount factor γ	0.55
	Delayed target update τ	10^{-2}
	OOB punishment λ_{OOB}	-10
	Bad action scale λ^{-}	2
Training	Episodes	1500
	Critic LR	10^{-3}
	Actor LR	10^{-4}
	Batch size	128
	Explore noise start	0.99
	Max steps	20

Table 7.4.: RL hyperparameters on real hardware.

					lw50 [Hz]		
Stage	Nr. samples	Average steps	SR	DiR	Initial	Shimmed	MAE
1	50	14.1 ± 6.3	0.92	0.86	3.86 ± 1.64	**1.5 ± 2.3**	0.107
1	↪ 24	8.75 ± 5.0	1	0.93	3.96 ± 1.57	**0.59 ± 0.05**	0.044
2	10	18.6 ± 4.2	0.70	0.80	3.93 ± 1.22	**2.47 ± 1.30**	0.160

Table 7.5.: *In situ* results of AI-driven shimming with a DRL agent. Values are reported as mean ± standard deviation over 100 random distortions drawn from a uniform distribution. Stage 1 is tested directly after training, while stage 2 includes reshimming and resetting the environment. Abbreviations: SR = success rate, DiR = direction ratio, lw50 = linewidth at 50% of the maximum, MAE = mean absolute error.

Convergence curves for mean reward, mean number of steps, and actor and critic losses are displayed in Figure 7.4. After training, the agent was directly tested on 50 random distortions and could shim to 1.5 ± 2.3 Hz within 14.1 steps, on average. Testing directly after training, without resetting the environment or shimming, was "stage 1" of evaluation. During this stage 1, 24 runs achieved a "Done" state, i.e. were able to reach the tolerance region, whereas the remaining runs were stopped after a maximum of 20 steps. For those 24 converged runs, shimming performance was superior to any previous results, namely 0.59 ± 0.05 Hz within 8.75 ± 5.0 steps.

Figure 7.5 showcases exemplary shimming results obtained using an RL agent during stage 1 of testing. A variation was evident in the average number of steps required to achieve a linewidth within the accepted tolerance. Specifically, in Examples 1, 2, and 4, the agent achieved linewidths below 1 Hz in just 3, 10, and 6 acquisitions, respectively. Example 3 presents a case where the desired linewidth exceeded the tolerance range. In this instance, the agent oscillates between states close to 1 Hz.

However, in "stage 2" of the evaluation, namely reshimming the magnet and resetting the environment (including calibration and locking), the DRL agent was tested again on 10 random distortions. Here, only one run could achieve a done state, and consequently shim to 0.56 Hz within 6 steps. All other runs took the maximum allowed number of steps, and were then stopped, still achieving some linewidth improvement ($\approx$ 2.5 Hz), but not the target value.

Failure causes / Environment drift The evaluation results of DRL-based shimming in stage 2 showed that the trained actor could not achieve decent shimming results after resetting the environment, shimming, and reloading the network weights. This may be caused due to environmental drifts that the agent explored and exploited. Resetting those drifts leaves the agent in a new domain, thus failing to shim.

Overall, this behaviour causes DRL to be impracticable for real shimming scenarios if this environmental drift cannot be solved. More efforts should be made to investigate that issue.

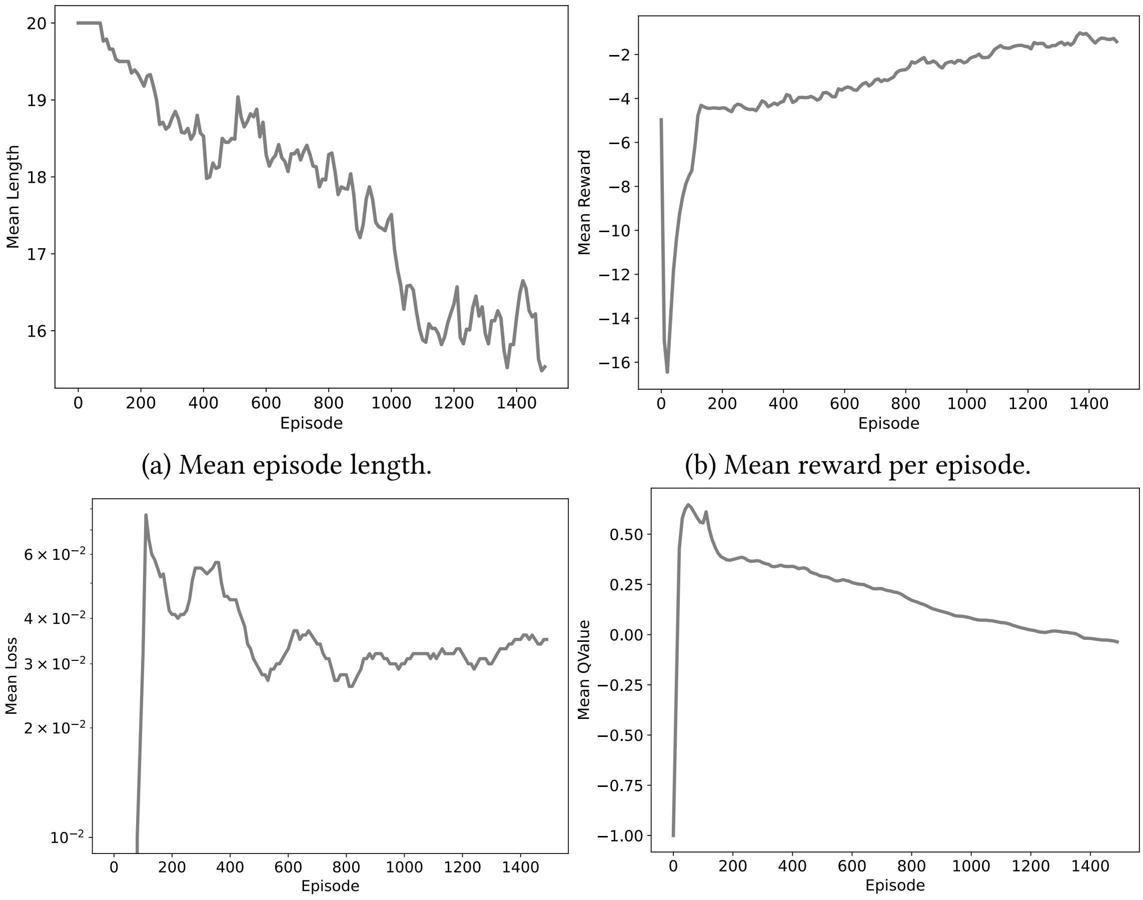
7.4. Limitations of reinforcement learning

DRL enables surpassing several issues of a supervised DL approach for shimming. This includes the acquisition of a labelled dataset, which further needs a known optimum to define the targets. Also, DL-based approaches do not optimize for a quality criterion.

However, RL methods bring their own limitations, and they are

- heavily sample-inefficient, making it expensive for NMR applications,
- strongly dependent on the correct reward function design,
- prone to local optima,
- weakly reproducible, even with the same hyperparameters and seeds.

The major limitation of RL during the completed experiments was the environmental drift that caused the agent to fail after a reset, making it applicable only to a standby shimming scenario.



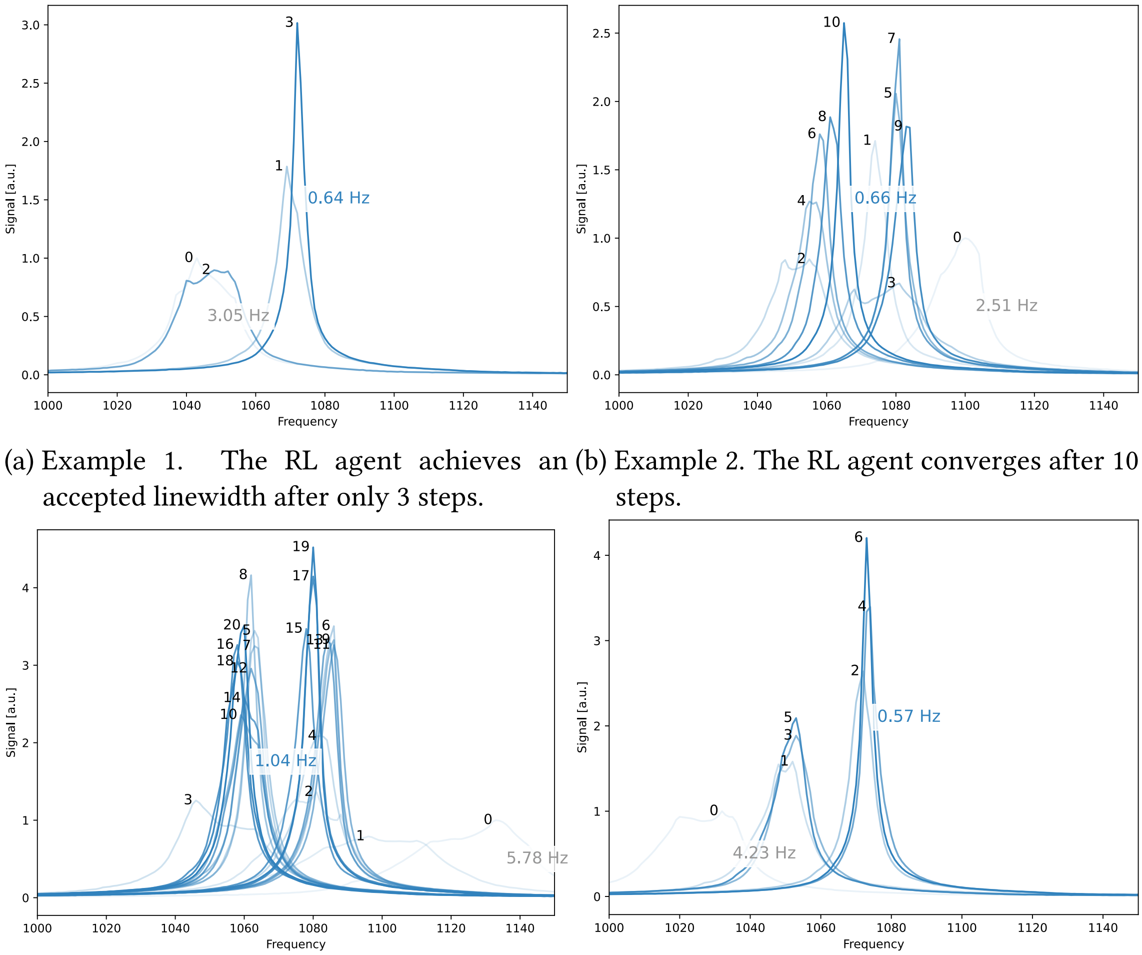
(a) Mean episode length.

(b) Mean reward per episode.

(c) Mean critic loss.

(d) Mean policy loss.

Figure 7.4.: Metric plots of training an RL agent online and on real hardware.



(a) Example 1. The RL agent achieves an accepted linewidth after only 3 steps.

(b) Example 2. The RL agent converges after 10 steps.

(c) Example 3. The agent oscillates between two local optima until stopped.

(d) Example 4. The RL agent converges after 6 steps.

Figure 7.5.: Arbitrarily chosen RL shimming results on a real spectrometer (stage 1), tested directly after training.

8. Ablation studies of AI-driven shimming

AI-driven shimming has proven to be successful, and it can greatly accelerate the cumbersome shimming procedure. This innovative approach, a first in its field, opens a virtually boundless realm of possibilities, given the previously unexplored nature of this research area. Throughout the preceding chapters, several ablation studies were conducted to understand the various components of this AI-driven method, and an overview is given in Table 8.1. This section aims to overview the critical design choices, challenges encountered, and unresolved questions related to scalability and explainability.

Experiment	Device	Method	Data	Tested	Shims
Spectral peak compression	M80	VAE	Real	Offline	-
Compressed eDR	M80	CeDR	Real	Online	4
Dataset size influence	B650	PeDR	Real	Online	2×6
Architecture choice	SHIMpanzee	eDR	Sim	Offline	3
Reliability and robustness	M80	DRE, eDR, PeDR	Real	Online	3-12
Nr. of simultaneous shims	M80	eDR	Real	Online	1-16
Input signal	M80	eDR	Real	Online	6
Explainable AI (XAI)	M80	eDR	Real	On&Off	4
Dataset drift	M80	-	Real	Online	-

Table 8.1.: Overview of all conducted ablation studies. Abbreviations: M80 = Magritek benchtop NMR with ^{1}H frequency of 80 MHz, B650 = Bruker preclinical MRI with 650 MHz, Sim = Simulated data.

8.1. Compression

NMR spectra are usually sparse, i.e., they contain much non-informative data such as noise. Thus, compression of spectra to a lower dimensional representation should be feasible without losing too much information.

First, this section describes the variational autoencoder for spectra (or peak) compression, including its architecture and training. The compression performance of the VAE was evaluated after being trained on real NMR spectra. Furthermore, this section describes an ablation study conducted for AI-driven shimming, especially the supervised DL method eDR, introduced in chapter 5. The new method was referred to as compressed eDR (CeDR), and the CeDR architecture uses spectra that were compressed to size 16 with a trained VAE, instead of feeding full spectra of size 2048, and then predicts shim correction terms.

As described in chapter 7, an RL agent's computation complexity and convergence could be improved with compressed spectra.

8.1.1. Variational autoencoder for spectral peak compression

DL problem definition for peak compression Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ be a dataset comprising only spectral data, where each $\mathbf{x}_i \in \mathbb{R}^{2048}$ is a spectrum of length 2048. A convolutional Variational Autoencoder (VAE) is tasked with compressing these spectra into a latent space $\mathbf{z} \in \mathbb{R}^{16}$ and subsequently reconstructing them. The encoder $\mathrm{E}_\theta(\cdot)$, parameterized by θ, maps the input spectrum $\mathbf{x}$ to the latent representation $\mathbf{z}$. Conversely, the decoder $\mathrm{D}_\psi(\cdot)$, with parameters ψ, reconstructs the spectrum from $\mathbf{z}$. The VAE is trained to minimize a loss function $\mathcal{L}$, which includes a reconstruction loss and a β-weighted Kullback-Leibler divergence term that regularizes the latent space. The loss function can be formulated as follows:

$$\mathcal{L} = \underbrace{\mathrm{MSE}(\mathbf{x}, \mathrm{D}(\mathrm{E}(\mathbf{x})))}_{\text{Reconstruction Loss}} + \underbrace{\beta \cdot KL(q_\theta(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z}))}_{\text{KL Divergence Term}}, \tag{8.1}$$

where MSE measures the mean squared error between the original spectrum $\mathbf{x}$ and its reconstruction, and KL is the Kullback-Leibler divergence between the learned latent distribution $q_\theta(\mathbf{z}|\mathbf{x})$ (as modelled by the encoder) and the prior distribution $p(\mathbf{z})$ (typically a standard Gaussian with mean μ and standard deviation σ).

Variational autoencoder architecture For the purpose of spectra compression, a variational autoencoder (see Figure 8.1) was used that consists of a 3-layer convolutional encoder with kernel size 51 and $[8, 32, 16]$ hidden neurons per layer. Each encoder layer included a convolutional layer, a batch normalization (BN) layer, and a LeakyReLU activation function. The decoder layers include transposed convolutions (or up-convolutional layers), a BN layer, and LeakyRelU, respectively. The last VAE layer was an upconvolutional layer, followed by a linear FC layer, and a sigmoid activation. For reconstruction, only the mean vector of the latent space was taken. The convolutional layers in the encoder and decoder efficiently captured the local structures within the spectral data, which was essential for effective compression and accurate reconstruction.

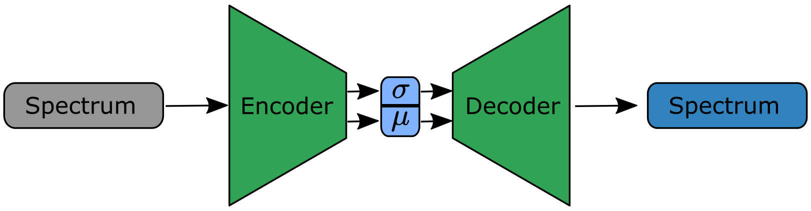


Figure 8.1.: Simplified variational autoencoder (VAE) structure. μ and σ are the latent Gaussian distribution's mean and standard deviation vectors.

Preprocessing The dataset from chapter 5 was used for VAE training. This dataset includes one-dimensional NMR spectra with adjustments to X, Y, Z, Z^2 shims. All spectra were normalized to a range of $[0, 1]$.

Compression training HPO was incorporated to find a balance between hidden dims $2^{[3-7]}, 2^{[3-6]}, 2^{[3-5]}$, learning rate (LR) of $[10^{-3}, 5 \times 10^{-4}]$, L2 weight $[10^{-6}, 10^{-2}]$, KLD weight $[10^{-7}, 10^{-4}]$ to yield the best performing model. The final model was trained for 200 epochs using the Adam optimizer with an LR of 2×10^{-4} and batch size 32. The KLD part of the VAE loss was weighted by 8.5×10^{-7}, and the L2 weight was 1.1×10^{-6}. Augmentation during the training included a normal shift of 4 and phase distortions of 10. A Gaussian kernel of 15 was used to smoothen the predictions during testing.

Compression results The trained VAE could compress spectra of size 2048 to a latent dimension of size 16, and reconstruct the original spectra with an average MSE test loss of 0.00043. Exemplary reconstruction examples are displayed in Figure 8.2. Out-of-distribution samples from additional monitoring datasets had similar errors of 0.00052 and 0.00049.

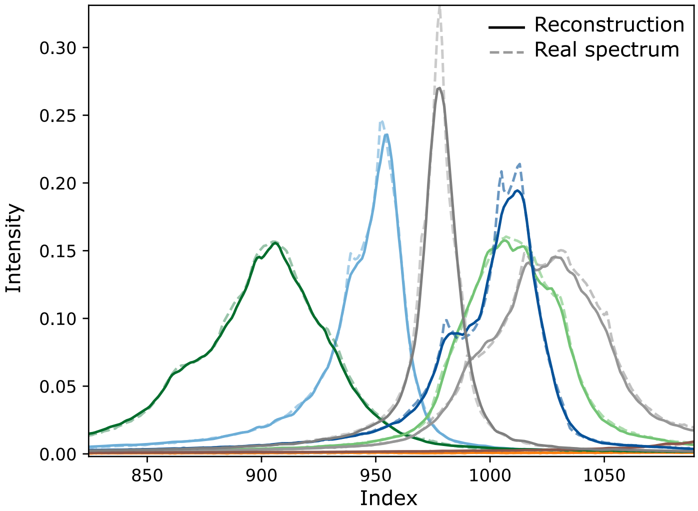


Figure 8.2.: Exemplary results of VAE compression and reconstruction on real spectra. Latent dim = 16.

8.1.2. Compressed enhanced deep regression for DL-based shimming

As described in the previous chapters, DL-driven shimming has succeeded in speeding up the shimming process. So far, the used architectures handle high-dimensional NMR spectra together with shim actions to predict shim corrections. The NMR spectra are internally processed by a convolutional neural network block to extract features. However, these features are over-represented, i.e., they usually have more neurons than the actual input, increasing the memory footprint and computational load of running AI-driven shimming.

Compressed enhanced deep regression (CeDR) was a proxy method to infer whether the latent space of a previously trained VAE captures the essential information content of the input spectra. The CeDR architecture replaces the CNN part of ConvLSTM with a pre-trained compression stage, represented by a VAE. See Figure 8.3 for an illustration.

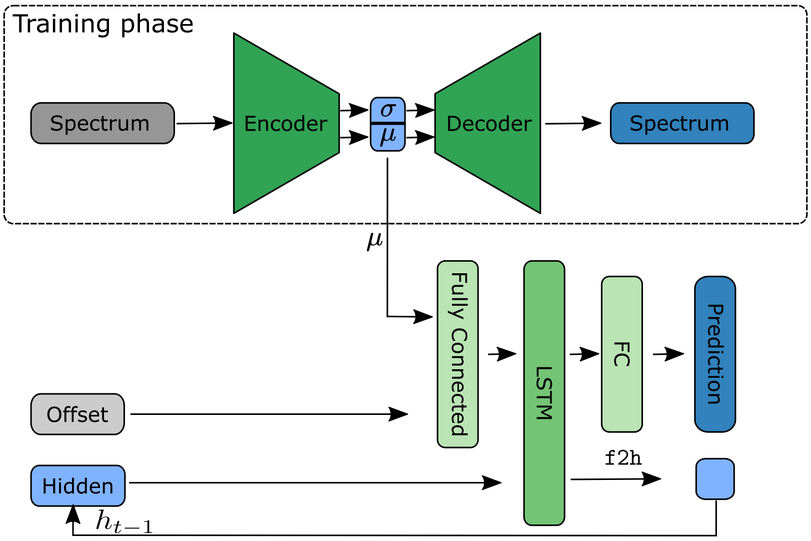


Figure 8.3.: CeDR architecture, including VAE for spectra compression. The compression VAE is trained separately, and CeDR incorporates the encoder with frozen weights.

Training The CeDR model was trained similarly to eDR, and corresponding hyperparameters are summarized in Table 8.2.

The only differences were that only label noise was applied during training, as the data was compressed beforehand for faster training, and different hidden sizes were utilized due to the smaller input shape.

After training on the same datasets of eDR, namely 10k H_2O and 5k H_2O with $CuSO_4$ (see Table 5.1), CeDR achieves an MAE of 0.0427 ± 0.0376 on the test sets, and averages of $[0.0479, 0.0406, 0.0584, 0.0240]$ for $[X, Y, Z, Z^2]$ shim. In comparison, eDR achieved an MAE of 0.023 ± 0.019 on the test set.

Real experiments Experiments on real hardware also validated that spectra compression with a VAE covers the important features of the spectra for shimming. In-situ performance was slightly worse than eDR with full spectrum as an input, but CeDR performed well

Target	Variable	Value
Architecture	Compressed size	16
	Data type	Real
	Input sequence length	10
	FC hidden size	64
	Dropout	0.2
	LSTM layers	4
Augmentation	Label noise	0.1
Training	Epochs	200
	Learning rate	0.0027
	Optimizer	Adam
	Loss	MSE

Table 8.2.: DL hyperparameters of CeDR.

enough and achieves a success rate of 0.98, and it can shim from 3.91 Hz to 1.27 Hz with a mean absolute error of 7.89% (see Table 8.3).

Figure 8.4 furthermore visualizes the performance of CeDR compared to eDR in dependency of the number of random steps during inference. The results of both methods are reported over ten random distortions.

Model	SR	DiR	initial FWHM	shimmed FWHM	MAE
eDR (chapter 5)	.99	0.93	3.9	0.72	0.041
CeDR	.98	.878	3.91	1.27	0.079

Table 8.3.: Results of AI-driven shimming with compressed eDR (CeDR) after 7 random and 2 predictive steps.

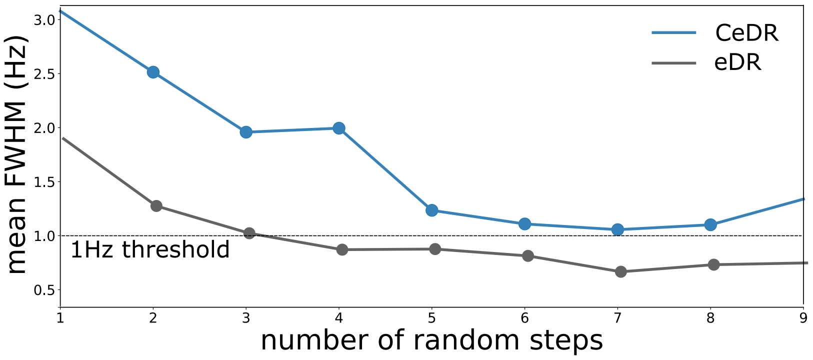


Figure 8.4.: In-situ comparison between shimming results of compressed eDR and eDR over ten random distortions indicating the dependency on the number of random steps. (Gratefully modified from Pierre Labouré.)

Shimming experiments with CeDR have proven that the compressed latent state of a spectrum contains all necessary features for shimming, and thus, the VAE is a fitting candidate to compress the state space for an RL agent from high-dimensional spectra to a latent vector.

8.2. Influence of dataset size

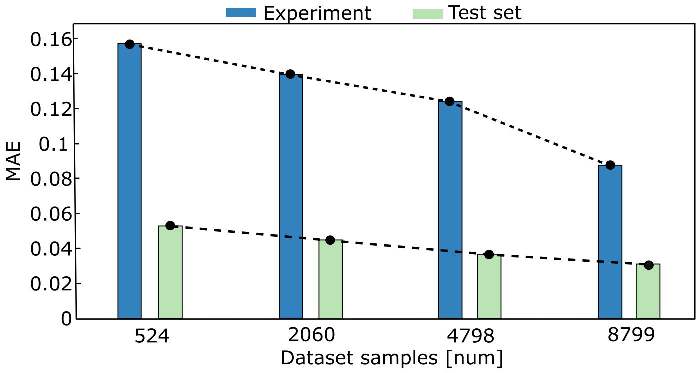


Figure 8.5.: Influence of dataset size, given for PeDR's performance. The model's prediction performance scales with the available training data. (Adopted from Becker et al. (2023).)

Deep learning models generally require large datasets to achieve good performance, while, especially for NMR applications, the exact relationship between dataset size and performance remains unclear. Most AI approaches for NMR use simulated data, which can be generated infinitely, but real data acquisition is expensive, as required for AI-driven shimming.

An ablation study was conducted on subsets of a whole dataset for parallel shimming to test this dependency. As expected, more training data results in a lower error in both the test set and real experiments (see Figure 8.5).

Furthermore, an apparent discrepancy between offline and online evaluations can be observed, probably caused by drifts in the experimental setup and stochasticity of data generation. This issue was indeed present in all AI-driven shimming methods.

8.3. Architecture impact

Neural network architecture design is not straightforward, especially in research fields currently being explored with AI. Thus, it was unclear whether to choose well-explored architectures that have proven to work or state-of-the-art (SOTA) approaches.

The design process for the eDR NN architecture incorporated several decisions, which were validated in simulation, and included:

- systematic versus random data acquisition,
- adding action information to the model's input sequence,

- convolutional neural network (CNN) versus Long Short Term Memory (LSTM) networks versus a combination of CNN and LSTM (ConvLSTM) versus a combination of CNN and Transfomer (ConvTransformer).

8.3.1. Data

The architecture design choices were evaluated using data from the SHIMpanzee simulator (see section 3.5) on Z, Z^2 and Z^3 shims, which give distinguishable peak distortions. The simulation used 10^5 "spins" (or voxels) with 3 shims (Z, Z^2, Z^3) and 2048 spectral points. Depending on the architecture, input type and sampling strategy, i.e. random offsets or action infusion, a dataset with 5000 samples was created.

8.3.2. Architectures

The following modifications to the network architectures were undertaken: If action information was infused, it was done in the first fully connected layer, followed by an additional layer normalization. The stand-alone LSTM architecture consists of a 2-layer LSTM, where the input sequence consists of a concatenated spectrum and action. The convolutional recurrent NN (ConvLSTM) architecture was similar to chapter 5 and 5 layers with 64 channels, kernel size 20, stride 2, no pooling, a 3-layer LSTM with hidden size 1024 was used to cope with the simulated data. A dropout of 0.2 for convolutional and fully-connected layers was applied. The CNN-Transformer combination replaces the LSTM of ConvLSTM with a transformer block, including four transformer encoder layers with seven heads and a dropout of 0.2. No positional encoding or masking was used, as the order inside a shimming sequence does not matter.

Ensemble architectures in chapter 4 have high computational requirements but did not show significant performance advantages and were neglected. Furthermore, AI research has an environmental footprint (Strubell et al., 2020), i.e., without major benefits, large computations should be reconsidered.

8.3.3. Training

All networks were trained with the same hyperparameters. This includes 4 input spectra (batched or as a sequence), uniformly random Z^0 shift $\in [-32, 32]$, label noise of 0.1, and phase distortions of $[-10, 10]$. All models were trained for 100 epochs with a learning rate of 2×10^{-4} and the Adam optimizer to minimize the Huber loss. Gradients were clipped to a maximum norm of 2 during training. Only the transformer-based architecture was trained with a lower learning rate of 10^{-5}.

The computations were performed with an AMD Ryzen Threadripper 3970X equipped with 256 GB RAM, and two graphics processing units NVIDIA GeForce RTX A5000.

8.3.4. Testing and evaluation

The MAE over 500 test samples for each architecture is reported in Table 8.4. The experiments validated that infusing past actions and using temporal information in the neural network architecture allows random offsets during inference and concurrently improves the prediction performance. Using the SOTA transformer did not seem to give improvements, probably due to its demand for massive datasets.

Acquisition	Action info	CNN	LSTM	Transformer	MAE
Systematic	-	✓	-	-	0.039
Random	-	✓	-	-	0.292
Random	-	✓	✓	-	0.114
Random	✓	✓	-	-	0.117
Random	✓	-	✓	-	0.051
Random	✓	✓	✓	-	**0.031**
Random	✓	✓	-	✓	0.036*

Table 8.4.: Comparison of dataset acquisition strategies and architecture choices on the test set's mean absolute error (MAE) in simulation. *: ConvTransformer required lower LR to converge. Abbreviations: CNN = convolutional neural network, LSTM = long short-term memory, MAE = mean absolute error.

8.4. Domain shifts in AI-driven shimming

Domain adaptation is a significant area of research in AI that aims to address a common problem where AI models are unable to generalize in the real world after being trained in a simulated environment. This issue, known as the sim2real gap, results from the differences that emerge when models trained in a controlled, simulated environment are applied to unpredictable real-world scenarios. Even if models are trained with real-world data, there is still the risk of small environmental drifts, where the underlying distribution of the data changes over time, which may lead to a decline in model performance.

These two challenges also arise in AI-driven shimming. While a shimming simulation can never accurately represent real hardware non-idealities and still maintain an acceptable computational load, the studies in the previous chapters have also suffered from environmental drifts, some more than others. For instance, DL approaches only experienced a slight performance decrease compared to the test set. At the same time, an RL agent trained online failed entirely when the environment, i.e., the NMR spectrometer, was recalibrated.

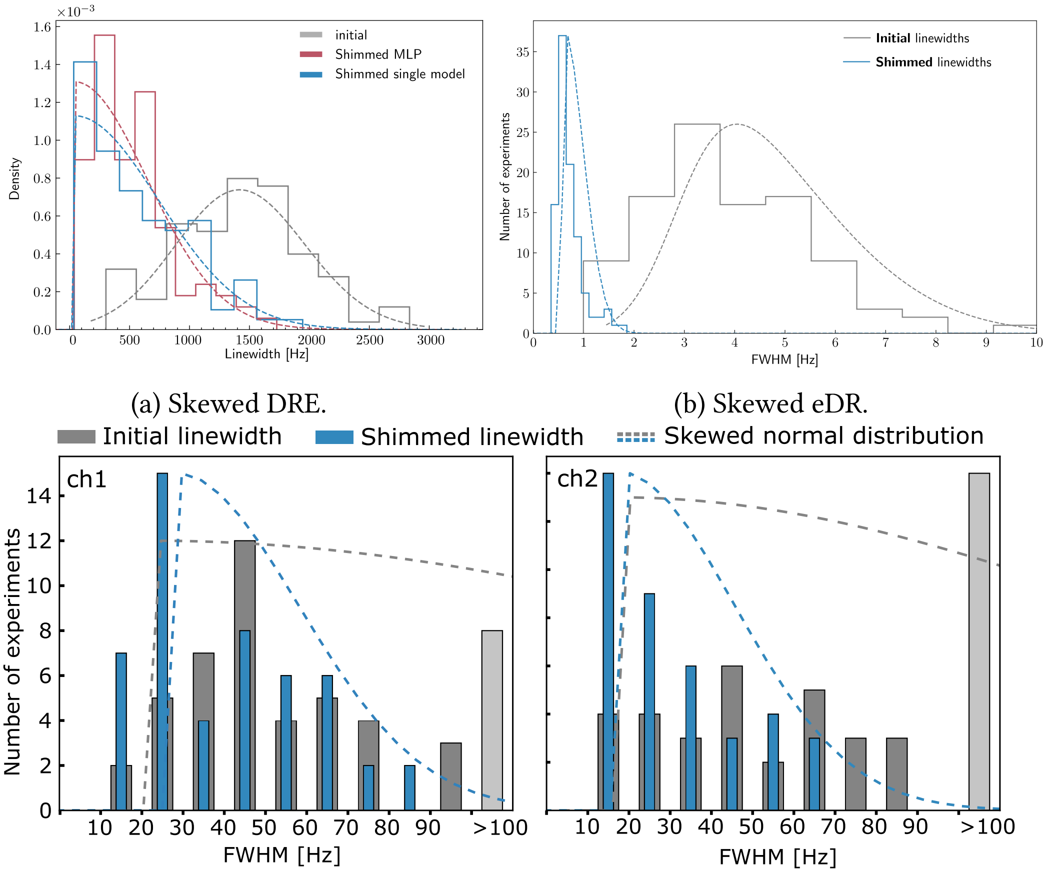
Also, refer to the next section for an example of the discrepancy between offline tests and online performance.

8.5. Reliability of AI-driven shimming

DL methods involve a trade-off between risk and return due to the random nature of their networks. However, the DL-driven shimming methods, including DRE, eDR, and PeDR, demonstrated remarkable stability under experimental conditions. The probability density functions between the initial and shimmed linewidths (FWHM) for all experimental test runs are plotted in Figure 8.6.

The probability density function (PDF) over 100 *in situ* runs for DRE is plotted in 8.6a. eDR exhibited the best performance, with over 99% probability of improving the initial linewidth in all 100 random distortions while keeping the improvement below 1 Hz in 87% of the cases. PeDR's performance was measured on 50 random distortions, showing a clear trend in linewidth improvement for both channels in the parallel setup.

In summary, all methods revealed some degree of variance in their prediction, which may be reduced through ensemble methods, as demonstrated in the case of DRE. However, traditional methods also tend to fail, as shown in Table 5.4, while having less chance for fast shimming.



(a) Skewed DRE.

(b) Skewed eDR.

(c) Skewed PeDR.

Figure 8.6.: **Reliability of DL-based shimming approaches.** Probability density function (PDF) indicating the distribution of initial and shimmed linewidths (FWHM). The dashed lines visualize skewed normal distributions fitted to the histogram (continuous line). (Adapted from Becker et al. (2022a,b, 2023))

8.6. Number of simultaneous shims

AI-driven shimming allows the prediction of shim corrections for multiple shims simultaneously. The first feasibility study in chapter 4 predicted three shim corrections. The enhanced method from chapter 5 already used four shims, and parallel shimming (see chapter 6) was conducted on 2 × 6 shims. However, the limit of simultaneous shims was unknown and strongly depended on their interdependencies and the used hardware. For example, in the parallel scenario, a total of 12 shims were corrected, but they were arranged as 2 × 6 (primarily independent) shims.

The ablation in this section summarizes shimming performances for different numbers of shims, and additional experiments were conducted. Especially, the most promising eDR method was tested on varying numbers of shims. However, this required acquiring a new dataset for each experiment. Note that the eDR model (ConvLSTM) experienced slight adjustments per experiment, e.g., kernel size, sequence length or learning rate, to accommodate higher learning complexity and different features.

Nr. shims	Method	Range	Nr. data samples	Steps (random, predictive)	Initial lw [Hz]	Shimmed lw [Hz]	MAE
3	DRE	10000	9261	4	1438	534	0.176
4	eDR	50	15k	(7,2)	3.9	0.72	0.041
6	eDR	20	30k	(10,1)	1.71	0.69	0.093
2 × 6	PeDR	2 × `Ref`	8799	(7,3)	92|93	39|26	0.089
8	eDR	20	10k	(7,2)	1.74	1.12	0.156
8	eDR	20	60k	(7,2)	1.69	1.21	0.141
16	eDR	20	10k	(17,3)	1.98	1.64	0.098
16	eDR	10	10k	(17,3)	1.17	1.02	0.263

Table 8.5.: Performance of simultaneously shimmed coils. The range was derived from int16 shim values [-32768, 32768]. The number of steps is given in the format "(random, predictive)", where systematic offsets are applied in the case of the DRE method. Linewidths are reported as the mean over in-situ evaluations. MAE is given in percentage to the range.

The results indicate a soft limit of simultaneous shim-able parameters for more than 6 shims. This decision is derived from the shimmed linewidth, which does not go below 1 Hz for more than 8 simultaneous shims. This could have several reasons: Insufficient random and predictive steps, DL architecture not being capable of learning all correlations, and (most probably) overlapping features and interdependencies introducing ambiguities that are not even visible to a DL algorithm. Furthermore, the MAE of predictions increased with the number of shims, indicating that either more data is needed or that the model might not be large enough to capture all underlying patterns.

Note that even though larger datasets were acquired for 6 and 8 shims, the performance did not seem to benefit from more samples. Also, the experiments with 16 shims seem to have an overall lower MAE, but as the number of shims is higher, the distortion from non-ideal values accumulates to worse linewidths.

8.7. Input signal

During shimming experiments with different samples using the eDR method, phase issues were encountered on spectra with multiple peaks. It was hypothesized that this could be resolved by training the model on the spectral magnitude (Becker et al., 2022b), if a higher degree of robustness is needed. Therefore, an ablation study was conducted to test the influence of the spectrum's value, i.e. taking the real part as input to the model was compared to the magnitude and the real plus imaginary part of the spectrum.

The same eDR model was trained three times, once with the real spectrum as the input signal, one with the magnitude of real and complex, and one with complex-valued input via two additional channels to the model. For 50 random distortions and an offset range of 20, the resulting shimming performance is reported in Table 8.6. The best performance was still achieved with the model designed for real spectra only, while a complex-valued network performed on par. However, feeding the real and imaginary did not yield any benefit as one may expect. Furthermore, a model trained with the magnitude of the spectrum performs worse.

In summary, a phase-robust approach would have some performance trade-offs or require further developments.

Input signal	SR	DiR	initial FWHM	shimmed FWHM	MAE
Real	0.96	.86	1.71 ± 0.70	0.69 ± 0.21	0.093 ± 0.042
Absolute	0.88	0.81	1.74 ± 0.64	0.94 ± 0.34	0.127 ± 0.057
Complex	1	0.85	1.66 ± 0.67	0.72 ± 0.19	0.103 ± 0.039

Table 8.6.: Results of AI-driven shimming with eDR using different input signals, on 6 shims for 10 random and 1 predictive step.

8.8. Towards explainable AI (XAI)

Neural networks are usually treated as black boxes. The concept of Explainable Artificial Intelligence (XAI) aims to make the decision-making process of intelligent machines transparent and understandable to humans. Recently, the European Union proposed the AI Act, which seeks to ensure the trustworthiness, fairness, accountability, and ethicality of AI applications, especially in critical domains that impact human lives, rights, or values.

This chapter discusses and conducts experiments introducing XAI in AI-driven shimming. Although XAI is not necessary for AI-driven shimming for NMR spectroscopy, it is beneficial for transparency and understanding of the decision-making process.

Uncertainty of AI-driven shimming One key point of XAI is trustworthiness, which could be measured by the uncertainty of a neural network's predictions. A common approach to measure DL models' uncertainty is using the variance in the ensemble of multiple models (Lakshminarayanan et al., 2017). In chapter 4, ensembles have proven to reduce prediction variance, i.e. increase trustworthiness. However, training a large quantity of models has a substantial environmental footprint (Strubell et al., 2020), and should be avoided if no immediate benefit is present. This is the case for AI-driven shimming with DRE, where the ensemble did not perform much better than a single model, and trustworthiness is not crucial.

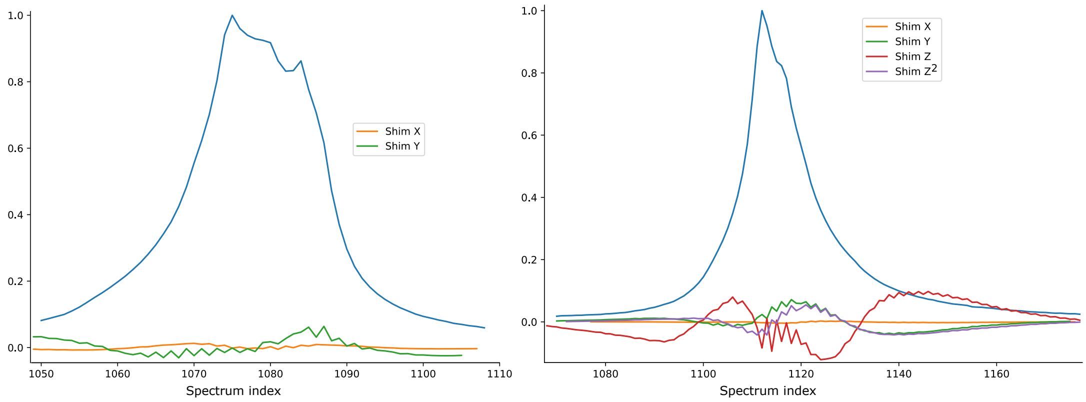
Another approach for ensemble methods that do not require repeatedly training multiple models can be achieved by dropout during inference (Gal & Ghahramani, 2016). This will lead to different predictions if queried with varying rates of dropout. Exemplary experiments were conducted with the best-performing eDR model from chapter 5 for two different dropout probabilities. First, a high dropout of 0.2 for the FC and convolutional layer, and second, a lower probability of 0.02|0.01 for conv|fc, respectively. The predictions were averaged by querying each spectrum of one step 10 times.

However, the resulting performance did not seem to improve. Both experiments show success rates of 1.00, DiR of 0.915 and 0.87, but the linewidths could only be shimmed from 3.9 Hz to 1.2 Hz and 1.23 Hz for dropout 0.2 and 0.02|0.01, respectively. The errors were MAE of 0.086 and 0.07. In comparison, eDR without dropout during inference achieved linewidths of 0.72 Hz, on average.

Opening the black box It is nearly impossible to reliably explain a neural network's predictions for custom DL architectures as used for eDR, which includes convolutional layers, recurrent connections, and fully-connected heads.

Commonly, SHapley Additive exPlanations (SHAP) values are used to explain predictions of neural networks (Lundberg & Lee, 2017). SHAP values are calculated by considering all possible combinations of input features and computing the difference in the model's prediction with and without the feature in question, which only works well for small dimensions. A neat approach to diving into understanding (or unscrambling) fully connected networks has been developed by Amey et al. (2021). Furthermore, the captum framework (Kokhlikyan et al., 2020) allows to interface of Pytorch models and allows for some explainability. Captum especially enables the explanation of predictions for computer vision models, where, for example, heatmaps can be superimposed on the input image to show importance weightings. Integrated Gradients, a feature attribution method in the Captum library, helps explain a neural network's predictions by attributing them to input features through gradient integration. This provides insights into important features and the model's behaviour for specific inputs.

First efforts were made to explain eDR's predictions with integrated gradients (IG) from Captum. One issue related to explaining sequences, and the initial step involved calculating the importance values with IG for a single input spectrum. Figure 8.7 gives two examples for integrated gradients of the trained eDR model on spectra with pure distortions, i.e., offsets of one shim only. Figure 8.7a shows a scenario with a pure distortion of the Y shim. As expected in section 3.3 and visualized in Figure 3.2b, the Y shim leads to lineshape distortion as a spike on the right side. Figure 8.7a also demonstrates that the models gave more importance to this region of the peak for Y while putting less emphasis on the X shim. However, residual importance is given to other regions of the spectrum and the other shims. A pure Z^2 shim distortion of $[0, 0, 0, 20]$ is given in Figure 8.7b. Despite the fact that the model predicted the correction values quite closely ($[-1, 4, -3, 15]$) given only



(a) Feature importance on a pure Y shim distortion.

(b) Feature importance on a pure Z^2 shim distortion.

Figure 8.7.: Towards explainability with integrated gradients. Tested on pure distortions with the trained eDR model.

one (!) input spectrum, the integrated gradient showed importance values to other shims as well. The results indicate that the model can differentiate single shim distortions, but mainly benefits from the sequence of spectra and actions, which were not considered here. Finally, this ablations study does not help to understand the reason behind the NN's predictions.

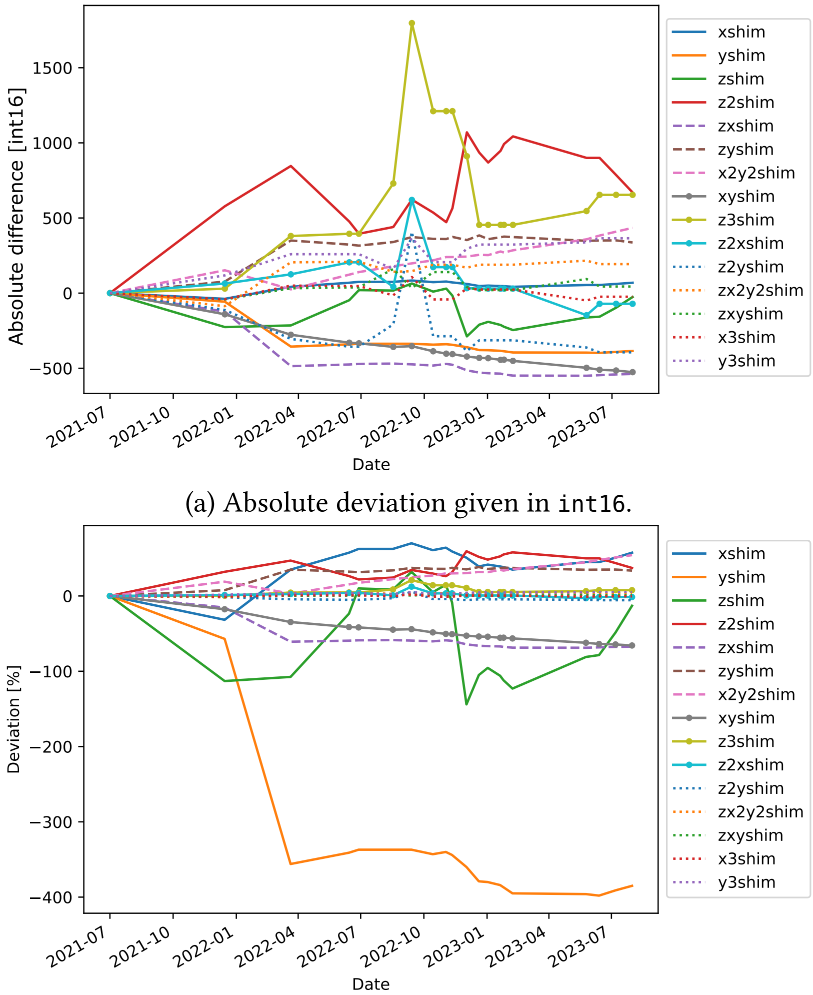
Another approach could be compression, which can enable more explainable predictions. For example, the decoder part of VAE can be designed to force the compression into a set of human-readable variables, such as moments or parameters of a Lorentzian peak. Afterwards, the decisions of an NN model should be explainable.

The central question of XAI approaches, however, is whether they help to explain what is going on inside the network, or (more importantly) really help to understand the reason behind it (Krenn et al., 2022) to lead to new scientific insight.

8.9. Shim optima drift

Why is it not possible to store the best reference shims for each sample, and load them at will?

A drift of the magnet clearly prevents the implementation of this idea, where the drift leaves the reference shim values to vary over time. Temperature, as well as other environmental influences, could also play a role that influences the drift. To show the shim optima drift exemplarily for an 80 MHz benchtop NMR, small monitoring datasets were acquired sparsely distributed among the years 2021 and 2024, and the reference shim values were extracted from all additional datasets with H_2O. Plotting the deviation from the first reference values indicates a drift of the optimum shim values, making it impossible to rely on a saved file of absolute shim values for a sample of choice. Figure 8.8 shows both the absolute deviation of the shim values as stored as `int16` values, and the deviation normalized by the shim influence or weightings, as measured in Subsection 5.3.2.



(a) Absolute deviation given in `int16`.

(b) Deviation normalized by influence in %.

Figure 8.8.: Reference shim value deviation on a benchtop NMR magnet.

9. Discussions on shimming with artificial intelligence

Using AI-driven shimming with DL models or RL agents is a promising approach to streamline the tedious shimming process. These methods could easily be incorporated into commercial systems.

So far, the following challenges remain:

- **Scalability.** Increasing the number of channels towards commercially available shim systems with up to 48 coils introduces overlapping features and will make shimming, even for an intelligent algorithm, harder. Splitting the problem into subgroups, such as what is done for the simplex method, will keep complexity in a controllable state.
- **Related approaches.** This thesis did not consider gradient shimming algorithms as they generally require gradients that are unavailable on most permanent benchtop magnets and on custom miniaturized hardware. Furthermore, the resolution of B_0 maps may be insufficient for parallel microtubes, as used in chapter 6. However, signal-based shimming with random offsets shows similarities to projection-based gradient shimming, where systematic projections with gradients along different orientations are taken to map the field. Intrinsically, random shim offsets similarly produce projections along random directions, but with very weak gradients (produced by the shim coils).
- **Algorithms.** Supervised deep learning can obtain internal feedback by a shimming history, however, this is not correlated to spectral quality, as in the case of RL. Thus, one may think that a DL model, instead of doing "real shimming" by improving spectral quality, correlates spectral peak shapes to shim distortions, which can be corrected in a second step if that correlation is correct.
- **Data.** Maybe the most severe challenges refer to the data necessary for AI-driven shimming.
  - **Dataset.** The features exploitable by a neural network are due to hardware non-idealities and, thus, are machine-specific. This imposes an unavoidable bias in the data distribution. Furthermore, DL models strongly depend on a large amount of available data samples, and RL agents are even more data-hungry (and sample-inefficient). As a consequence, it is inevitable to acquire machine-specific datasets for effective shimming on the machine. Transfer learning from a large database collected from different spectrometers and fine-tuning powerful models could reduce this constraint and reduce dataset biases.

- **Measured sample.** Another data-related challenge is the sample measured. So far, a hydro-deuterium oxide (HDO) peak was selected for all datasets, which may not be the solvent of choice. Thus, shifting to a generally available reference peak, such as tetramethylsilane (TMS) or trimethylsilylpropanoic acid (TSP)[1], would allow broader generalizability beyond water peaks.
- **Reference values.** Finally, DL models require a labelled dataset, which is defined relative to the global optimum (that must be known). An RL agent mitigates this challenge and can optimize for spectral quality directly.

[1] The chemical structure of TMS and TSP is mostly symmetric, and thus, their chemical shift is close to 0 ppm.

Part II.

Artificial Intelligence for RASER MRI

10. AI for RASER MRI

The proposed method of this chapter is being prepared as an article titled "Deep learning corrects artefacts in RASER MRI profiles", to be submitted to a journal in 2024. Additionally, the work in this chapter led to the master thesis "Deep Learning for the prediction of RASER-MRI profiles" (Arvidsson & Bertilson, 2023), conducted by Filip Arvidsson and Jonas Bertilson, which focused on several testing phases toward feasible AI correction of RASER MRI images. Figures were modified from this work.

10.1. Overview

Magnetic resonance imaging (MRI) is a critical diagnostic tool in medical practice but has inherent spatial resolution limitations, which may limit its diagnostic capabilities. Furthermore, standard MRI is limited by the following requirements: Strong gradients[1] are necessary for spatial encoding, an RF excitation pulse needs to be applied to excite the spins from equilibrium, leading to heating of tissues due to the absorption of RF power, and MRI images show background signals as all surrounding molecules in an excited image slice are excited. Recently, Radio-frequency Amplification by Stimulated emission of Radiation (RASER) has emerged to improve MRI resolution by hyperpolarization. RASER-MRI signals spontaneously emerge without the need for a radiofrequency pulse (RF), which enhances the safety of the process. Furthermore, the signal of RASER MRI images is higher due to hyperpolarization, which relatively reduces background noise. Currently, RASER MRI images are acquired along projections, only requiring small gradients to create angles for the image projections. Furthermore, RASER eliminates T_2^* constrictions of pulsed NMR when continuous parahydrogen pumping is used. However, RASER-MRI images frequently exhibit significant image artefacts due to the nonlinear nature of the signals.

This chapter examines the effectiveness of using deep artificial neural networks to eliminate image artefacts in RASER MRI (see Figure 10.1). However, RASER is a very new technique, and barely any RASER images exist. On top of that, hyperpolarized MRI is not very well automated, making the idea of measuring a dataset with tens of thousands of images, which is necessary for effective deep learning, impossible. Fortunately, the RASER equations are known (Lehmkuhl et al., 2022), which allows the simulation of RASER signals, and the neural networks are trained on purely simulated data.

The study is divided into two phases from a DL perspective: correcting the projections themselves (1D AI) and denoising the entire image (2D AI). The main objective was to reconstruct random 2D RASER-MRI images with varying degrees of realism during dataset simulation. The findings revealed that a simple convolutional neural network trained with synthetic data can correct RASER distortions in image projections, and adding a denoising

[1] The resolution is inversely proportional to the gradient's strength.

image-to-image network to the 1D correction network can significantly enhance image quality.

RASER also requires good magnetic field homogeneity; thus, it also benefits from a fast shimming algorithm.

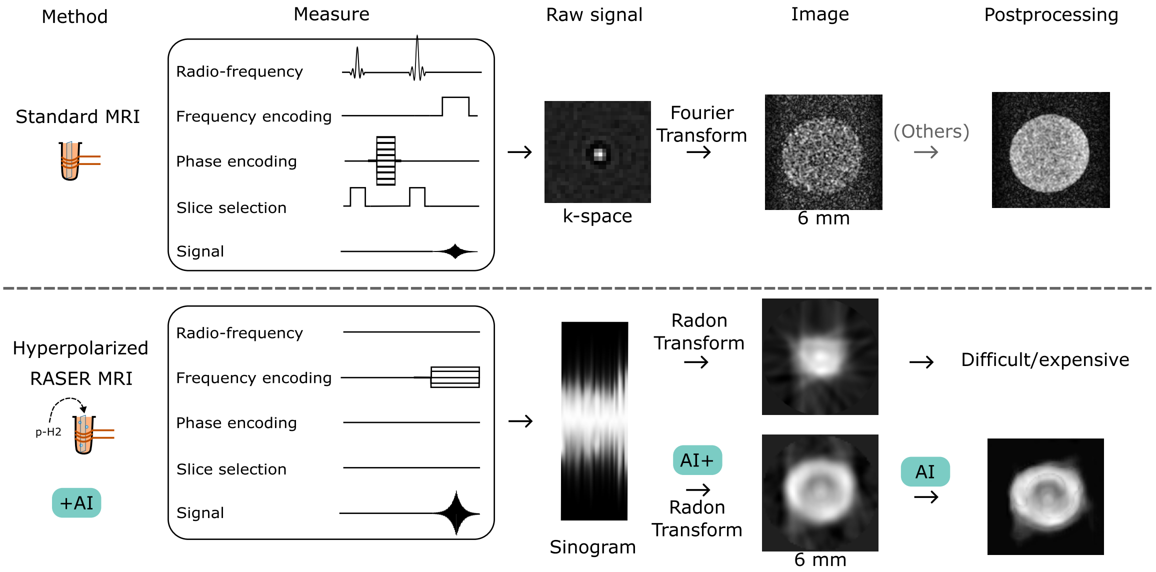


Figure 10.1.: Traditional MRI imaging requires strong gradients, an RF excitation pulse, and shows background noise. On the other hand, RASER MRI does not require an RF pulse, and has lower background noise due to hyperpolarization, however, it shows significant artefacts. AI can correct RASER MRI image slices to yield images of better quality.

10.2. Method - DL part

This section will focus on the deep learning part of RASER image correction using deep learning. It will start with a formal problem definition and describe the neural network architectures used for this chapter's approach. Then, the concept of the DL approach for projection correction and image enhancement is explained, concluding with DL training details.

10.2.1. Formal problem definition

Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ be a static dataset, where $\mathbf{x}$ is the input, and $\mathcal{T}_i = \{\{k_j, p\}_j\}_{j=1}^{N}$ be a subset for every $\mathbf{x}$ with N slices/angles.

In detail, let $\mathbf{x} \in \mathbb{R}^{44\times 44}$ be a randomly generated image, and $p_1, \ldots, p_j \in \mathbb{R}^{1\times 67}$ be slices/projections thereof, with j being the number of projections/angles. Each slice undergoes a RASER simulation R_α with different parameters α (TPI, pumping rate, ...), yielding a RASER signal $r_j = R_\alpha(p_j) \in \mathbb{R}^{1\times 4096}$. The distorted spectra $k_j \in \mathbb{R}^{1\times 200}$ can be obtained by Fourier transformation (FFT) of r_j, and cutout to a region of interest (ROI). By

Radon transformation (RT) of the spectra k_j, the RASER image $\mathbf{z} = \mathrm{RT}(k_1, \ldots, k_N)$ with artifacts can be obtained.

Now a 1D model $\mathrm{F}^A_{\theta_A}(k_j)$ is defined, where F^A is a convolutional neural network with parameters θ_A, that is supposed to correct each distorted spectra k_j (or a number of k_j with different RASER parameters α) to its original slice counterpart $\hat{p}_j = \mathrm{F}^A_{\theta_A}(k_j)$. Ideally, $\hat{\mathbf{y}} = \mathrm{RT}(\hat{p_1}, \ldots, \hat{p_N})$ of all corrected slices should match the random image $\mathbf{x}$, and be of better quality/resolution than the RASER image $\mathbf{z}$.

An additional model $\hat{\mathbf{x}} = \mathrm{F}^B_{\theta_B}(\hat{\mathbf{y}})$, where F^B is an encoder-decoder architecture based on U-Net (Ronneberger et al., 2015) with parameters θ_B, should denoise the 1D-corrected image $\hat{\mathbf{y}}$ to $\hat{\mathbf{x}} = \mathbf{x}$.

Both networks are trained in a supervised manner, F^A using subset $\mathcal{T}$ to minimize the MSE loss L_1 between the predictions $\hat{p}$ and p, and F^B to minimize the MAE loss L_2 between $\hat{\mathbf{x}}$ and $\mathbf{x}$.

10.2.2. Architectures

The architectures used for RASER artefact removal are twofold. First, a 1D-AI model corrects image projections, and then a 2D-AI model enhances the entire 2D image quality.

All final models were carefully selected from a large space/cohort of candidates, all trained with limited neural architecture search (NAS) and hyperparameter optimization (HPO) using raytune.

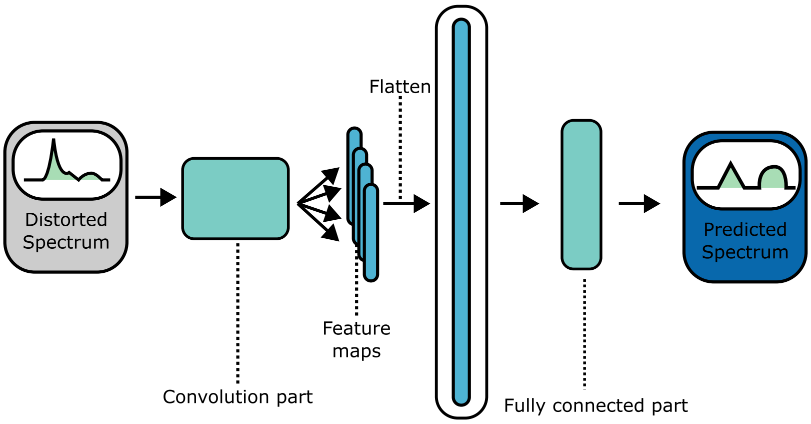


Figure 10.2.: 1D-AI: Convolutional neural network to correct 1D spectra/projections. (Created by Arvidsson & Bertilson.)

1D RASER signal correction architecture One-dimensional distorted RASER slices were corrected with a vanilla convolutional neural network architecture. The architecture (depicted in Figure 10.2) consists of a convolutional and a fully connected part. Three convolutional blocks are used, each consisting of convolution, ReLU, dropout and pooling. Kernel size 8, 64 filters and a dropout of 0.2 are used. The FC part has five fully connected blocks, each represented by an FC layer, activation, batch normalization and dropout of 0.2. The final activation is an Exponential Linear Unit (ELU).

When designing the architecture for the neural network, there were several options to choose from. Batches of input spectra, ranging from 1 to 16, all with varying TPI values, were evaluated, and the corresponding TPI value was fed to the network. Different approaches were tried for batched inputs, including providing each spectrum individually or treating them as "RGB" channels. Additionally, different numbers of filter kernels (< 128), kernel sizes, convolutional layers, and fully connected layers were incorporated into the network. Finally, different learning rates were experimented with. The ablation experiments (summarized by Arvidsson & Bertilson (2023)) revealed that the best-performing network used a single spectrum with the highest possible TPI, without feeding this TPI value to a convolutional neural network. This approach kept the network complexity at a minimum.

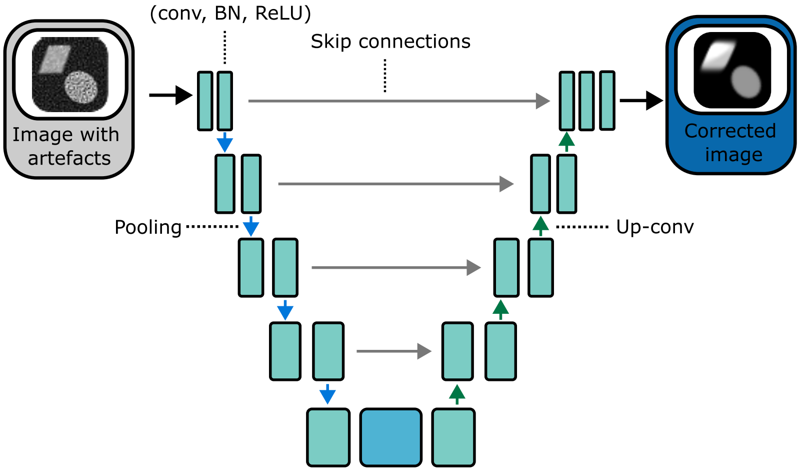


Figure 10.3.: 2D-AI: U-Net encoder-decoder architecture to correct 2D RASER MRI images.

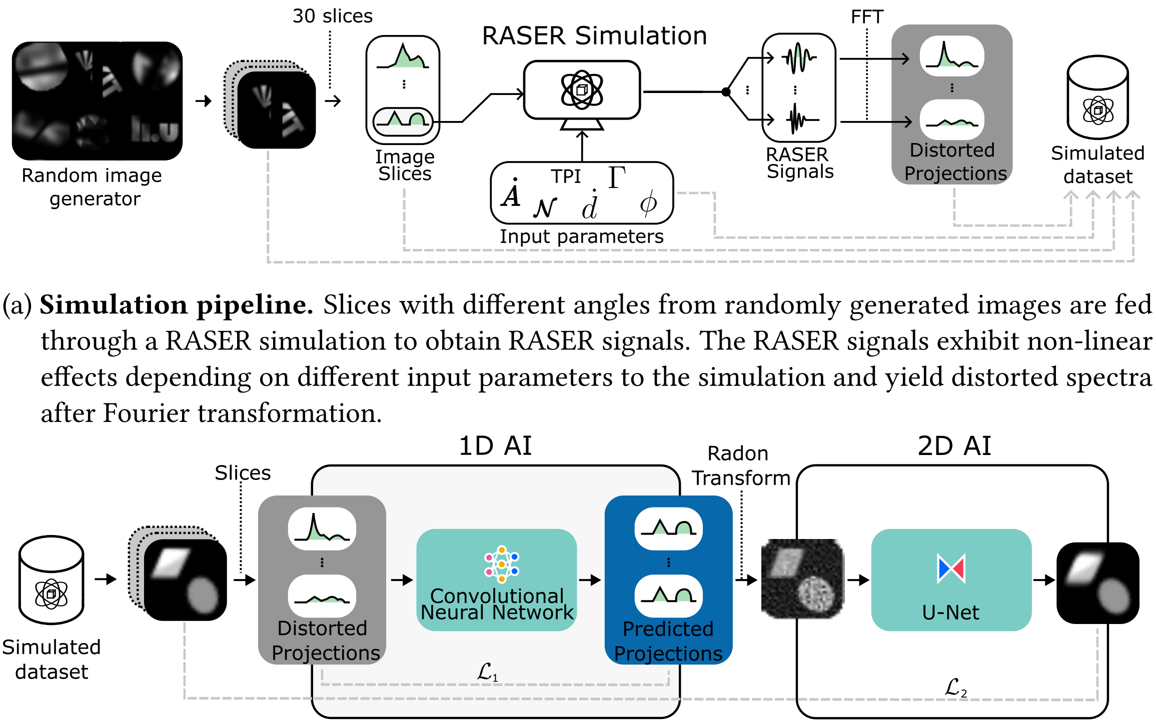
2D RASER image denoising architecture Image-to-image denoising was enabled with a U-Net architecture with reduced feature size, adapted from Persson (2021) and depicted in Figure 10.3.

The input to the architecture was the inverse Radon transform of the 1D-AI-corrected image slices, and the output was an image of size 44×44 pixels. The "down path" of the U-Net architecture was made up of four double convolution layers, each consisting of convolutions, batch normalization, and ReLU activation, with increasing numbers of features $[8, 16, 32, 64]$ and a kernel size of 3. The "up path" of the architecture had four transposed convolutions with kernel size 2 and stride 2, with the number of features being inverted, i.e., $[64, 32, 16, 8]$.

Amongst the cohort of possible architectures were autoencoders and the U-Net with varying input-output combinations. For example, sinogram-to-image, sinogram-to-sinogram and image-to-image modalities were tested. The sinogram-to-image approach was inspired by Zhang et al. (2023), and should avoid losses induced by Radon transformation. However, all tests performed worse, and the best architecture was chosen as an image-to-image U-Net.

10.2.3. Concept

The process for correcting RASER MRI images with AI is illustrated in Figure 10.4. This approach was developed and tested using simulated RASER data. Initially, images $\mathbf{x}$ are generated using a custom random image generator and then sliced into projections p_j. These image slices are then processed through a RASER simulation R_α with varying parameters α, resulting in distorted spectra $k_j = R_\alpha(p_j)$ that mimic the non-linear effects of RASER. When these distorted spectra are subjected to an inverse Radon transformation, the resulting image $\mathbf{z}$ will be distorted or unrecognizable. Therefore, a two-phase deep learning approach was used to correct the images. The first DL model (1D-AI) was trained to correct one-dimensional distorted spectra k_j back to their original projection $\hat{p}_j$. The individually corrected projections were then subjected to a Radon transformation, and the resulting image $\hat{\mathbf{y}}$ was processed through a second DL model (2D-AI) that removes further artefacts to achieve a predicted image $\hat{\mathbf{x}}$ that matches the original image $\mathbf{x}$ before the RASER simulation.



(a) **Simulation pipeline.** Slices with different angles from randomly generated images are fed through a RASER simulation to obtain RASER signals. The RASER signals exhibit non-linear effects depending on different input parameters to the simulation and yield distorted spectra after Fourier transformation.

(b) **DL pipeline.** Random images and corresponding slices from a simulated dataset are fed to DL models. A convolutional neural network (1D AI) corrects one-dimensional distorted spectra, which are reconstructed with the Radon transformation to an image. This image is further enhanced by a U-Net model (2D AI).

Figure 10.4.: **Concept of DL for the correction of RASER MRI images**, separated into simulation and deep learning parts.

10.2.4. DL training

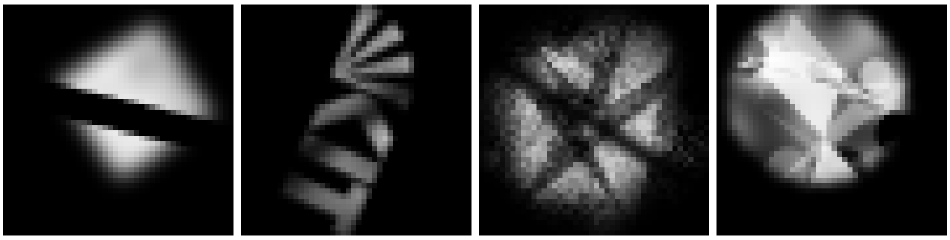
Normalization To improve the resolution for the followed Fourier transformation, all image slices are zero-padded by double the number of points on both sides of the time domain. The absolute value of the resulting spectrum was then cut to a region of interest (ROI) of size 200, which was centred at the maximum value. This ROI was then normalized to $[0, 1]$ with respect to the TPI value. Similarly, the target slices, which are the regression targets, are also normalized to $[0, 1]$.

Training details All models are trained using the Pytorch framework, using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 4.145×10^{-5} and weight decay of 1.012×10^{-6}. The 1D model was trained for 500 epochs, and a batch size of 300 to minimize the MSE loss, and the 2D model was trained for 20 epochs with a batch size of 5 to minimize the MAE loss.

Hardware requiremens The simulations and DL training were performed with an AMD Ryzen 9 5950X equipped with 64 GB RAM, and a graphics processing unit NVIDIA GeForce RTX 3080Ti. The datasets roughly allocate 99.6 GB of disc space. The trained 1D-AI convolutional and U-Net models have $\sim$ 7.2M and $\sim$ 490k parameters allocation 27.9 MB and 1.9 MB and of disc space, respectively.

10.3. RASER simulation

10.3.1. Random image generator



(a) Example 1. (b) Example 2. (c) Example 3. (d) Example 4.

Figure 10.5.: Examples of randomly generated 44×44 pixels images for the RASER MRI simulation. (Created by Arvidsson & Bertilson.)

For a diverse range of images, a random image generator was implemented. The randomness was inspired by domain randomization (Tobin et al., 2017), where models are trained on synthetic data from several domains with arbitrary variations. The real domain should appear to the model as another domain, and it should be able to generalize well. For RASER MRI, this means that a "real" RASER MRI image should appear to the model as one random image, and it should be able to process it.

Each generated image was of size 44×44, combining basic shapes and image transformations. Arbitrarily, 2 to 20 shapes were selected from a circle (80%) or a polygon shape with 4-8 vertices and a random opacity $\in [0, 1]$. Additionally, Gaussian blur ($\sigma = 2$) was applied with 50% chance, and Gaussian white noise with $\sigma^2 = 0.01$ was added with 30% chance. The image was then masked with a circle (56%), square (24%), star (5%), scribble (5%), KIT logo (5%), or liu logo (5%), each scaled randomly between 50-100% of the images' width and the centre location shifted randomly within $\frac{1}{4}$ of the image size, also randomly rotated $\in [0, 360]^\circ$ and Gaussian blurred with $\sigma \in 1, 2, 3$. Finally, a line with random width among $[2, 10]$ pixels, random rotation $\in [0, 360]^\circ$ and 70% Gaussian blur ($\sigma \in 1, 2, 3$) was added to 80% of the images. Examples are given in Figure 10.5. Furthermore, parahydrogen pumping was simulated with a decaying pumping rate at two random positions between the index 24 and 42 of each projection.

In total, 10k images were generated without pumping, 10k images with pumping, and 1k images with high TPI fluctuations (pm 20% TPI) among slices of one image. Each image was sliced into 30 signals, which underwent the RASER simulation to yield input-output pairs for the 1D-AI model.

10.3.2. RASER simulation details

The RASER simulation R_α is governed by non-linearly coupled differential equations (Lehmkuhl et al., 2022) with different parameters α, including the population inversion d, signal amplitude A, phase ϕ, pumping rate Γ of para-hydrogen, and the total population inversion (TPI), which is defined at $d(0)$. During the simulation of RASER signals, written by Sören Lehmkuhl and Peng Wang in MATLAB, the signal amplitude A and phase ϕ were varied at a random value within the scope of typical experimental noise, while ϕ experienced uniform and A normal noise, respectively. Most importantly, different TPI values were chosen for different signals both within one image as well as between images. The TPI is supposed to be above the RASER threshold (Equation 2.13). Thus, varying TPI intrinsically covers different resonator Q factors, varying number of spins n_S, concentration changes V_S of the substrate, i.e., the chemistry's "goodness", and changing relaxation times T_2^*. Additionally, variations of the polarization pumping rate Γ were considered, which can occur, for example, when mixing is not perfect, or there is still parahydrogen diffusion even if bubbling or shaking the sample for parahydrogen dissolution has been halted. A decaying pumping rate Γ was included on two needles at random positions between positions 24 and 42 of each slice.

Each randomly generated image of size 44×44 underwent RT to generate sinograms with 30 angles (evenly distributed between 0° and 180°). Then, each projection was fed to the RASER simulation with varying parameters α, resulting in a RASER signal of 4096 points. After FFT of this signal, a distorted spectrum was obtained and cut to 200 points.

RASER simulation on all 32k images results in over 300k distorted and target spectra pairs.

10.4. Results

10.4.1. Performance metric and evaluation protocol

To evaluate the performance of artefact removal and reconstruction of 2D RASER MRI images, the mean squared error (MSE) and the Structural Similarity Index Measure (SSIM) (Wang et al., 2004), a quality measure based on human perception, are used.

The AI pipeline for correction of RASER MRI images, including 1D and 2D models, was evaluated first on a simulated test set, and then on real RASER images.

A hold-out test set of 100 images was simulated with the random image generator from Subsection 10.3.1. Additionally, the Modified Shepp-Logan CT phantom was generated, and its projections were fed through the RASER simulation.

Finally, the generalization of the AI models from simulation to reality was tested on a real RASER image of a standard 5 mm glass tube.

10.4.2. Simulated RASER experiments

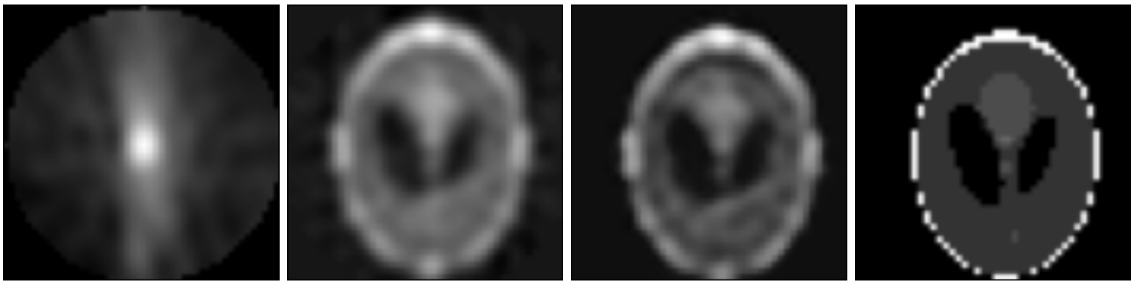
The trained 1D model achieved an MSE of 0.015 ± 0.007 and SSIM of 0.449 ± 0.151, and the 2D model an MSE of 0.002 ± 0.002 and SSIM of 0.906 ± 0.065 on a hold-out set of 100 randomly generated images. Generalization to out-of-distribution samples was demonstrated on the Modified Shepp-Logan CT phantom (see Figure 10.6), which the model has not seen before during its training process, yielding an MSE of 0.056 and SSIM of 0.275 after AI-driven correction of the 1D signals, and MSE of 0.026 and SSIM of 0.599 after the entire correction pipeline (1D+2D). All important structures are visible after correction, as compared to the raw RASER MRI image, even though the resolution was relatively low at 44x44 pixels.

Figure Figure 10.7 showcases exemplary results from the 1D-AI component of this study, which reconstructs projections $\hat{p}_j$ given distorted RASER spectra k_j. The input for this model consists of spectra k_j with a size of 200, while the targets p_j are image projections sized at 67. The model's predictions align well with the general features of the target projections. However, it overlooks the high-frequency variations in the profiles, leading to a smoothed appearance in the resultant images. This issue might stem from the convolutional layers' kernels not effectively capturing these high-frequency features, as the primary improvement in loss reduction was achieved by approximating the overall projection shape. Implementing stricter regularization strategies could potentially enhance the model's ability to predict these high-frequency variations more accurately.

10.4.3. Real RASER experiments

To evaluate the generalizability of AI-based correction of RASER MRI images, real images were measured on a Magritek 60 Ultra Multi-X benchtop magnet. The shim coils of the device were used to produce gradients such that projections of the sample could be measured.

For the measurements, it was crucial to have precisely the same parameters as used during the simulation; otherwise, a capture bias would introduce a domain shift or reality gap, and good predictions are not guaranteed.



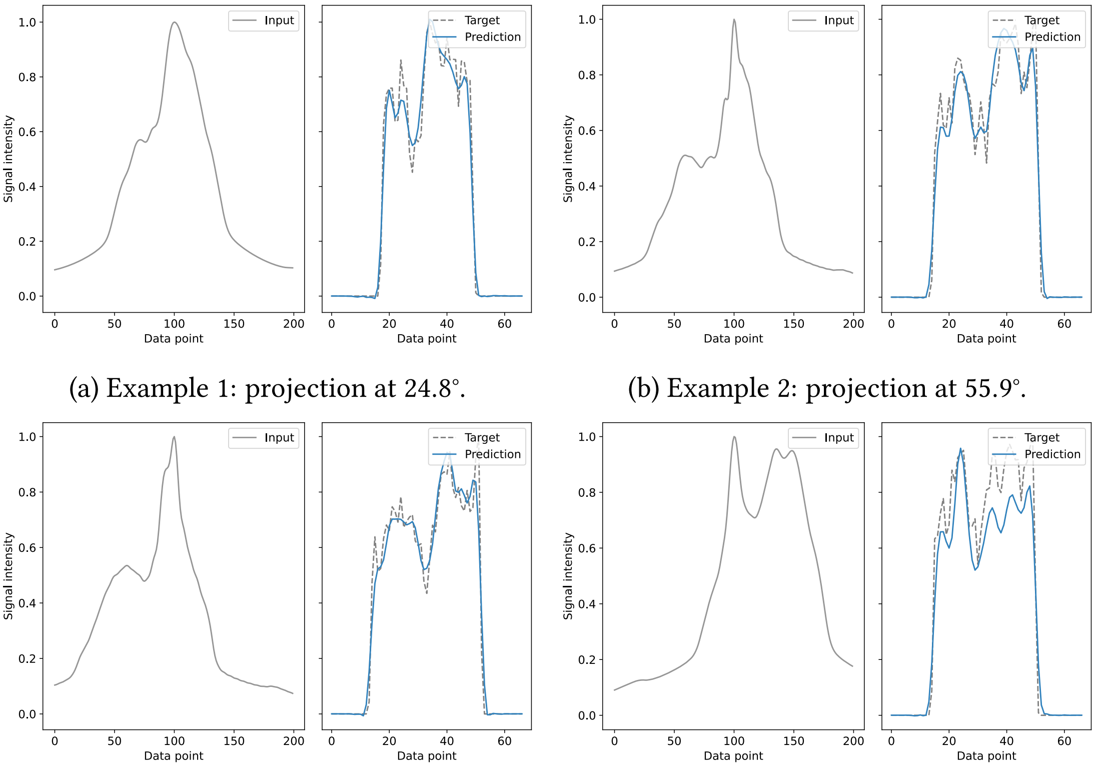
(a) Distorted RASER image $\mathbf{z}$.

(b) 1D-AI-corrected image $\hat{\mathbf{y}} = RT(\hat{p_1}, \ldots, \hat{p_N})$.

(c) 2D-AI-corrected image $\hat{\mathbf{x}}$.

(d) Target image $\mathbf{x}$.

Figure 10.6.: RASER MRI image correction using deep learning on the Shepp-Logan phantom with 44 × 44 pixels. The Structural Similarity Index Measure (SSIM) after 1D and 2D corrections were 0.275 and 0.599, respectively.



(a) Example 1: projection at 24.8°.

(b) Example 2: projection at 55.9°.

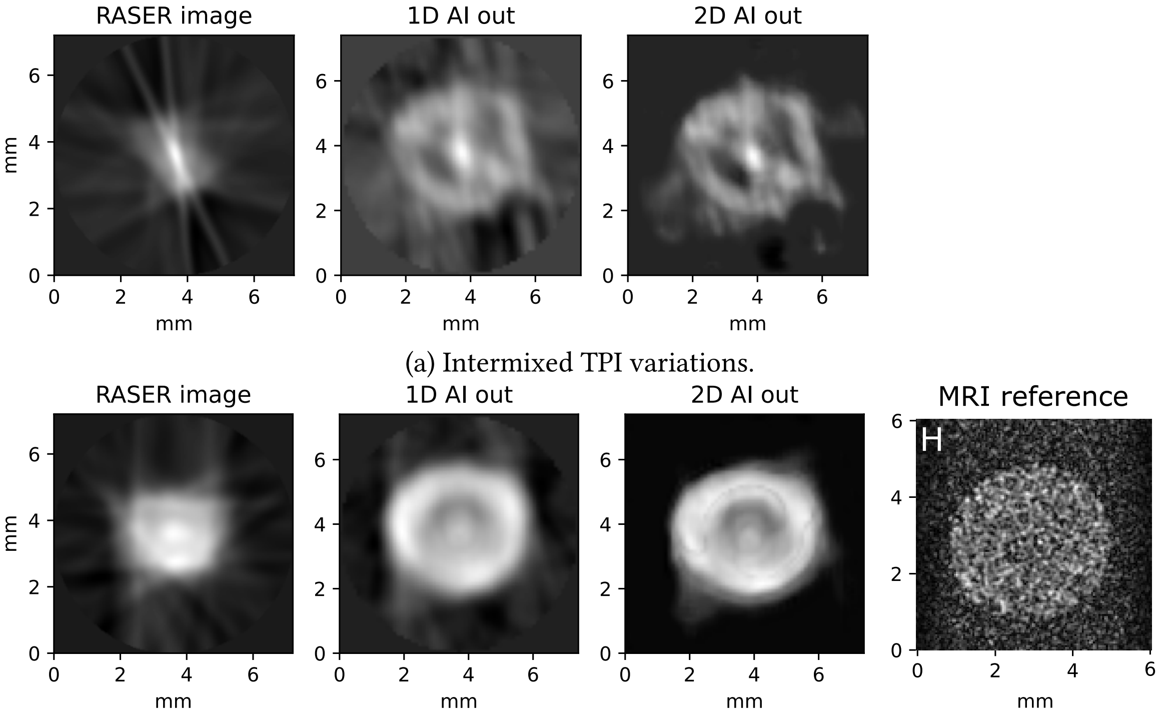
(c) Example 3: projection at 68.3°.

(d) Example 4: projection at 130.4°.

Figure 10.7.: **Correction of RASER MRI image projections with the 1D-AI part.** Examples of the Shepp-Logan phantom from Figure 10.6. Given input signals as cut-outs from the frequency domain (left), the model is supposed to predict the target profile (right, grey). The 1D AI neural network predictions are overlayed (right, blue).

Experimental setup A SABRE experiment with a standard sample (mixture of Ir-IMes catalyst, pyrazine and methanol) ensured hyperpolarization. The Ir-IMes catalyst was made in-house, leading to $C_{19}H_{36}ClIr$ (0.451 mg, 70.3%). The SABRE sample was prepared with 0.6 mL deuterated and degassed methanol, 3 mM catalyst and 60 mM pyrazine.

Parahydrogen gas of 98%, obtained by a helium compressor at 23 K with an Iron (III) oxide hydroxide catalyst, was bubbled for 20 s with a thin capillary through the sample in a 5 mm NMR tube, while placed in a 6.5 mT magnetic field to ensure maximum polarisation transfer. Then, the capillary was removed from the tube, and the sample was transferred to a Spinsolve 60 spectrometer (Magritek). ^{1}H-NMR spectra were recorded in deuterated methanol (Sigma Aldrich) at room temperature with a custom acquisition pulse sequence. The "pulse sequence" itself does not contain an excitation pulse, but acquires the receiver signal with 32768 points, a dwell time of 500 µs, and a bandwidth of 2 kHz.



(a) Intermixed TPI variations.

(b) High TPI regime.

Figure 10.8.: **AI-corrected real RASER MRI images of a tube** with an inner diameter (ID) 4.1 mm. Uncorrected RASER projections would yield a distorted image (left). After correcting the 1D projections with AI, the real dimensions of the tube are visible (middle), and even enhanced with the 2D AI pipeline (right). (a) Real experiments vary in quality. (b) Consistent TPI values increase the reconstruction performance. The image obtained with additional MRI shows low signal-to-noise, and background noise.

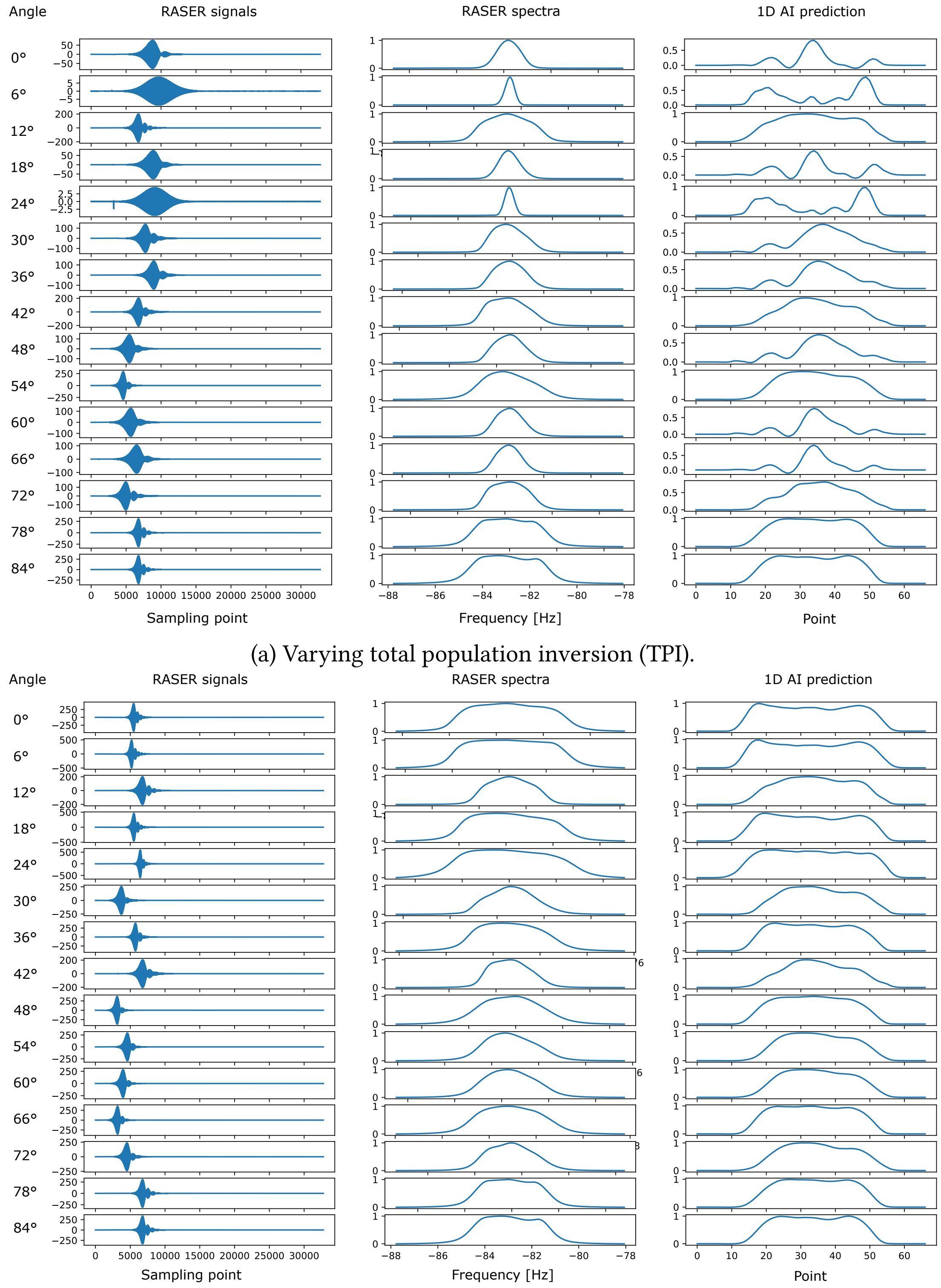
Results on a standard 5 mm **tube** The first subject for imaging was a standard 5 mm tube with an inner diameter (ID) of 4.1 mm. Thus, the measured image should show a clear round circle with that diameter. Overall, 30 projections were measured of the tube at 30 different angles between [0,180]°. The initial reconstruction of the raw RASER projections yielded a distorted image, as seen in Figure 10.8 (left). However, after applying the 1D correction to the image projections and 2D artefact removal, the tube phantom's outline became visible, as shown in Figure 10.8.

Nonetheless, several artefacts remained, predominantly due to signals measured with TPI values close to the RASER threshold d_{th} (see Equation 2.13). This issue is evident in Figure 10.9 (a), where projections at $0, 6, 18, 24^{\circ}$ show signal intensities (which directly correlate to TPI) below 150, resulting in poor predictions.

As the simulation was restricted to high TPI values, the DL model expects signals measured with overall high TPI values. Thus, measurements close to the RASER threshold were repeated. Figure 10.9 (b) displays these signals, now with consistently high TPI values, and demonstrates improved prediction performance for these projections. The overall reconstructed and AI-corrected image is visualized in Figure 10.8 (b), where the tube phantom is clearly visible, approximating its true 4 mm inner diameter.

As a reference, a traditional (2D cross-section) MRI was measured of the 5 mm NMR tube using a standard spin-echo sequence, while the shim coils are exploited to produce (small) gradients[2]. A mock sample of 90%D_{20} and 10%H_{20} is placed in the resonator, encoded using frequency and phase encoding and the recorded k-space is Fourier transformed (FT) to yield the image.

[2] Note that no slice selection was applied.



(a) Varying total population inversion (TPI).

(b) High TPI.

Figure 10.9.: **Real RASER signals with varying TPI values** for the first 15 angles, leading to the images as reported in Figure 10.8. Low TPI values, visible as RASER signal intensities below 150, yield distorted spectra and predictions.

10.5. Discussions

Scalability One limitation of the current setup was that the DL model's input was limited to 67 modes, corresponding to 44 × 44 pixel images after Radon transformation. Scaling the simulation dimensions would be required but would have a massive computational cost. Figure 10.10 shows that increasing the number of modes, i.e. the width of the input spectrum to the RASER simulation scales computational time t^3. For each mode, 30 slices are simulated, corresponding to one image.

To extend the applicability of this approach to high-resolution imaging, the simulation should be expanded to include larger images and the neural network architectures should be adapted accordingly. Consequently, better prediction generalizability should be addressed.

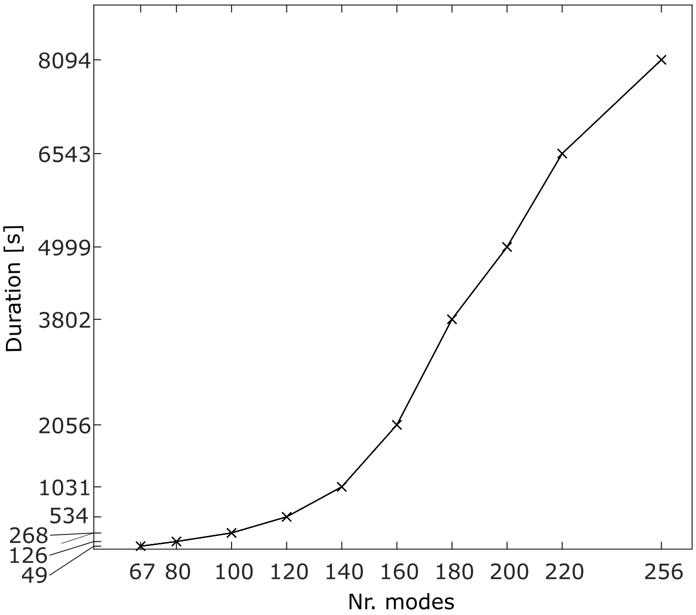


Figure 10.10.: Duration of RASER simulation for one image with 30 angles and different modes. Time demands for simulating more modes roughly scales cubic.

Total Population Inversion (TPI) values During the simulation, strong hyperpolarization, i.e. high TPI values, was assumed. However, the DL model strongly depends on these high TPI values, and it fails for TPI values close to the RASER threshold. Real RASER experiments were conducted to verify this, and the DL model's accuracy was compared with the experimental results. The measurements showed that the predictions are far from accurate for signals with lower TPI, recognizable by a broader "blob" in time and a shorter peak in the spectrum. On the other hand, for high TPIs, the model works perfectly fine. The signal intensity in the time domain scales with the TPI value, and it can be seen that values below 200 yield worse predictions, as the DL model was trained with high TPI values. To better understand the results, Figure 10.9a illustrates the measurements of varying TPI with the time signal, frequency domain, and AI prediction on the spectrum, and Figure 10.9b repeated and high TPI.

Part III.

Conclusion and outlook

11. Conclusion

The rising success of artificial intelligence approaches in all areas of research and science did not stop at the magnetic resonance frontier. This thesis demonstrated success in enhancing two main challenges: The cumbersome shimming process in NMR, previously tackled by slow optimization algorithms, and the presence of artefacts in RASER MRI images, previously unsolved remainders in measured images.

11.0.1. Can AI speed up shimming?

The first objective of this thesis was to speed up the tedious shimming process by incorporating AI approaches. Several methods were developed, each surpassing the limitations of the previous one. Overall, a significant contribution was made in gaining insight into the working sites needed for AI-driven shimming: A proper dataset, its preprocessing steps, a fitting neural network architecture, and finally, evaluation protocols to test generalization on real hardware. DL architectures ranging from vanilla CNNs to custom convolutional LSTM architectures showed great success in handling varying input designs to the neural network, such as sequences of "shim histories". Experiments have demonstrated success for varying numbers of shims, or even for a parallel custom probehead setup. A paradigm shift to DRL built a foundation for further research, where no dataset and no optimum are required, and the spectral quality can be optimized directly. However, a primary challenge still remains, namely environmental drift, which causes discrepancies between offline training and testing, and deployment to the machine.

Taking a look beyond the shimming horizon Several broader questions affecting shimming remain.

First, it's worth asking if shimming with a finite set of electrical coils is the only approach to this problem. Researchers have investigated the usage of multi-coils that are not restricted to a small set of simple (orthogonal) basis functions (Juchem et al., 2011), as well as making (higher-order) shim coils less crucial or even obsolete through methods such as RF shimming or "Shim pulses" (Topgaard et al., 2004).

Ultimately, the question arises: Is shimming the only solution at all to acquire a spectrum of good quality? This leads to the promising approach of reference deconvolution, a post-processing technique applied to imperfect measurements. Although several NMR methods, such as selective excitation of peaks and hyperpolarisation, need decent shimming, this technique can provide an excellent alternative. The idea is already several years old (Metz et al., 2000; Morris et al., 1997), and more recently, attracted DL-based approaches (Schmid et al., 2023).

Combining AI-driven shimming and (AI-driven) deconvolution will undoubtedly provide the most time savings for users of NMR spectrometers who don't want to waste time optimizing the machine.

11.0.2. Can AI correct RASER MRI image artefacts?

RASER is a new hyperpolarized MRI technique that offers better image contrast and resolution than traditional pulsed MRI but can cause unwanted artefacts along the measured image projections. To address this issue, AI has shown promising results in removing the non-linear effects of RASER from image projections and Radon-transformed images thereof. The approach involved using a vanilla convolutional neural network or a standard U-Net architecture, both of which have shown significant reconstruction and denoising performance. While the AI models were trained using synthetic and random image data, they have been found to work on real MRI measurements. Further experiments with more complex phantoms have to prove whether DL stays a robust candidate for artefact correction.

Future directions RASER MRI requires a form of hyperpolarization, and in experimental setups, signal amplification by reversible exchange (SABRE) or parahydrogen-induced polarisation (PHIP) can be used. These approaches, however, require the use of (toxic) catalysts and solvents. Targeting human MRI, current research focuses on biocompatible solutions of hyperpolarization (MacCulloch et al., 2023).

Furthermore, acquiring RASER along projections to measure 2D images requires an extension to map 3D volumes of interest, which should include higher resolutions than the studies conducted within this thesis.

Finally, and for the sake of interpretable and reliable medical diagnosis, mathematical denoising techniques (Gaa et al., 2023) could be considered.

Acknowledgments

It would be a sacrilege to claim that I finished this thesis alone.

The biggest thanks are due to my Doktorvater JGK: without your constant support, your uninterrupted stream of new ideas, and your excellent ability to motivate, this thesis would never have been finished. Your generous gestures to allow me to travel to conferences and spend evening discussions with hoppy drinks completely changed my perspectives and were heavily encouraging. Also, the freedom that you gave for this project made it a real success. I am very grateful that you allowed me to be part of your team.

I also wish to thank my second PhD examiner, Pascal, for his uplifting words and inspiring discussions. Thanks also to the chair Hans-Jörg Bauer at my defence for his neutral eyes.

No less thanks are due to Mazin, the best supervisor you could wish for. Your constant availability and your willingness to discuss things in detail, as well as your friendly and calm manner, have made this work a pleasure. شكراً لك!

At this point, I also need to thank my unofficial second mentor, Sören. I am very grateful that you joined the group, as your feedback boosted every aspect of my work. I am also pleased to call you my friend, with whom I have eaten more Greek food than my whole life. I am also glad we shared a room during ENC 2023; it was really fun.

Special thanks are due to Stefan Kesselheim for his hour-long online discussions about all deep learning aspects of my projects, which resulted in qualitatively appealing results. I really enjoyed these Helmholtz AI-funded talks.

Not to be neglected, the work done by my students Pierre, Filip and Jonas was excellent, and I could not have done it better. I sincerely owe you modesty for parts of my thesis. I hope that you also profited from your time at IMT.

@YTC, thanks for your great work on our shared paper, and I enjoyed our time before and at the ENC! I sincerely thank all my co-authors on our publications, namely Achim, Anastasiya, Ajmal, and Mengjia, for their efforts. You did great work; without you, the publications would have been a painfully long process. This also includes (hopefully) soon-to-be-published papers, so big thanks to Jing, Elene and Peng. And, of course, big thanks to Albina for letting me be part of her research project. My thanks should also be extended to the patience of members of Mazin's group: Mehrdad, Hossein, Mohammad, YTC, Chun, Nourhan, and CJ.

Furthermore, working at IMT and in the labs would not have been possible without the constant work done by the permanent staff. So, thanks a lot for your help!

Some official disclaimers between the lines: I would like to acknowledge the assistance provided by AI tools, such as the language enhancement tool grammarly, which offered valuable time savings for editing through grammar and punctuation corrections. After using this tool/service, I reviewed and edited the content as needed and take full responsibility for the content of the publication.

What makes a working place a pleasant working place? Colleagues. Thus, I want to thank all my colleagues for the friendly talks in the hallway, long coffee breaks, and fun times. This especially applies to Franz, Julia, Emil, Andrea, Leon, Raj, George, Dominique, Jakoba, Shishir, Paco (+Pia). Full of joy, I can claim that some colleagues have become friends.

Talking of friends, I need to thank the friends that I had before starting my PhD for their patience. I will be more available from now on, so call me; the first beer is on me! Fruitful discussions with Daniel and Alfredo guided the outcomes of this thesis. Furthermore, thanks to Florian, Maxi, Tobi and Sebastian for distracting me with gaming; without you, I would have finished faster. Furthermore, I thank Rhythm and Nicolas for their genuine friendship.

Last but most importantly, I want to warmly and heartfully thank all my family members and my girlfriend for their (financial and emotional) support and understanding. Ich hab euch lieb!

In all modesty,
Moritz

List of Figures

List of Tables

Publications

Journal Articles

[1] **M. Becker**, A. Kolchinskaya, M. Jouda, J.G. Korvink (2022). "Deep regression with ensembles enables fast, first-order shimming in low-field NMR". In: *Journal of Magnetic Resonance*, 10.1016/j.jmr.2022.107151.

[2] **M. Becker**, S. Lehmkuhl, S. Kesselheim, J.G. Korvink, M. Jouda (2022). "Acquisitions with random shim values enhances AI-driven NMR shimming". In: *Journal of Magnetic Resonance*, 10.1016/j.jmr.2022.107323.

[3] L.A.N. Julius, H. Scheidt, G. Krishnan, **M. Becker**, O. Nassar, S. Torres-Delgado, D. Mager, V. Badilita, J.G. Korvink (2023). "Dynamic dielectrophoretic cell manipulation is enabled by an innovative electronics platform". In *Biosensors and Bioelectronics: X*, 10.1016/j.biosx.2023.100333.

[4] **M. Becker***, Y. Cheng*, A. Voigt, A. Chenakkara, M. He, S. Lehmkuhl, M. Jouda, J.G. Korvink (2023). "Artificial intelligence-driven shimming for parallel high field nuclear magnetic resonance". In *Scientific Reports*, 10.1038/s41598-023-45021-6. (*These authors contributed equally)

In preparation

[5] **M. Becker**, J. Bertilson, F. Arvidsson, E. Aslanikashvili, J.G. Korvink, M. Jouda, S. Lehmkuhl (2024). "Deep learning for the correction of artefacts in RASER MRI profiles".

Conference contributions

[6] **M. Becker**, A. Kolchinskaya, M. Jouda, J.G. Korvink. "Deep regression with ensembles enables fast, first-order shimming in low-field NMR". **Experimental Nuclear Magnetic Resonance Conference (ENC)**, Orlando, USA, 2022. *Oral Presentation.*

[7] **M. Becker**, A. Kolchinskaya, M. Jouda, J.G. Korvink. "Using deep learning for first-order shimming". **European Conference on Magnetic Resonance (EUROMAR)**, Utrecht, The Netherlands, 2022. *Poster presentation.*

[8] **M. Becker**, S. Lehmkuhl, S. Kesselheim, J.G. Korvink, M. Jouda. "Acquisitions with random shim values enhance AI-driven NMR shimming". **Experimental**

Nuclear Magnetic Resonance Conference (ENC), Pacific Grove, USA, 2023. *Poster presentation.*

[9] **M. Becker**, P. Laboure, S. Lehmkuhl, S. Kesselheim, J.G. Korvink, M. Jouda. "AI in NMR: Applications for shimming". **Helmholtz AI conference**, Hamburg, Germany, 2023. *Poster presentation.*

[10] F. Arvidsson, J. Bertilson, **M. Becker**, S. Lehmkuhl. "Deep Learning for the prediction of RASER-MRI profiles". **Helmholtz AI conference**, Hamburg, Germany, 2023. *Poster presentation.*

Supervised theses

[A] F. Arvidsson, J. Bertilson (2023). "Deep Learning for the prediction of RASER-MRI profiles" [Linköping University]. oai:DiVA.org:liu-199640.

Bibliography

(2021). Magritek GmbH.

Achiam, J. (2018). Spinning Up in Deep Reinforcement Learning.

Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D. L., & Erickson, B. J. (2017). Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. *Journal of Digital Imaging*, 30(4), 449–459.

Amey, J. L., Keeley, J., Choudhury, T., & Kuprov, I. (2021). Neural network interpretation using descrambler groups. *Proceedings of the National Academy of Sciences*, 118(5), 1–7.

Arvidsson, F. & Bertilson, J. (2023). *Deep Learning for the prediction of RASER-MRI profiles.* PhD thesis, Linköping University.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer Normalization. *arXiv preprint.*

Bao, Q., Chen, F., Chen, L., Song, K., Liu, Z., & Liu, C. (2016). A new gradient shimming method based on undistorted field map of B0 inhomogeneity. *Journal of Magnetic Resonance*, 265, 25–32.

Becker, M., Cheng, Y. T., Voigt, A., Chenakkara, A., He, M., Lehmkuhl, S., Jouda, M., & Korvink, J. G. (2023). Artificial intelligence-driven shimming for parallel high field nuclear magnetic resonance. *Scientific Reports*, (pp. 1–11).

Becker, M., Jouda, M., Kolchinskaya, A., & Korvink, J. G. (2022a). Deep regression with ensembles enables fast, first-order shimming in low-field NMR. *Journal of Magnetic Resonance*, 336, 107151.

Becker, M., Lehmkuhl, S., Kesselheim, S., Korvink, J. G., & Jouda, M. (2022b). Acquisitions with random shim values enhance AI-driven NMR shimming. *Journal of Magnetic Resonance*, 345, 107323.

Bogner, W., Otazo, R., & Henning, A. (2021). Accelerated MR spectroscopic imaging—a review of current and emerging techniques. *NMR in Biomedicine*, 34(5).

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym. *arXiv preprint.*

Callaghan, P. (1993). *Principles of Nuclear Magnetic Resonance Microscopy.* Oxford: Clarendon Press.

Chen, D., Wang, Z., Guo, D., Orekhov, V., & Qu, X. (2020). Review and Prospect: Deep Learning in Nuclear Magnetic Resonance Spectroscopy. *Chemistry – A European Journal*, 26(46), 10391–10401.

Cheng, Y.-T., Jouda, M., & Korvink, J. (2022). Sample-centred shimming enables independent parallel NMR detection. *Scientific Reports*, 12(1), 14149.

Chmurny, G. N. & Hoult, D. I. (1990). The Ancient and Honourable Art of Shimming. *Concepts in Magnetic Resonance*, 2(3), 131–149.

Cordova, M., Moutzouri, P., Simões de Almeida, B., Torodii, D., & Emsley, L. (2023). Pure Isotropic Proton NMR Spectra in Solids using Deep Learning. *Angewandte Chemie International Edition*, 62(8), 1–9.

Dietterich, T. G. (1990). Machine Learning Research: Four Current Directions. *AI magazine*, (pp. 1–47).

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1857 LNCS (pp. 1–15). Springer.

Dong, J., Fu, J., & He, Z. (2019). A deep learning reconstruction framework for X-ray computed tomography with incomplete data. *PLOS ONE*, 14(11), e0224426.

Dozat, T. (2016). Incorporating Nesterov Momentum into Adam.

Edalati, M., Zheng, Y., Watkins, M. P., Chen, J., Liu, L., Zhang, S., Song, Y., Soleymani, S., Lenihan, D. J., & Lanza, G. M. (2022). Implementation and prospective clinical validation of AI-based planning and shimming techniques in cardiac MRI. *Medical Physics*, 49(1), 129–143.

Ernst, R. R. (1968). Measurement and Control of Magnetic Field Homogeneity. *Review of Scientific Instruments*, 39(7), 998–1012.

Fajfar, I., Bürmen, A., & Puhan, J. (2019). The Nelder–Mead simplex algorithm with perturbed centroid for high-dimensional function optimization. *Optimization Letters*, 13(5), 1011–1025.

Farahani, A., Voghoei, S., Rasheed, K., & Arabnia, H. R. (2021). A Brief Review of Domain Adaptation. In R. Stahlbock (Ed.), *Advances in Data Science and Information Engineering* (pp. 877–894). Springer International Publishing.

Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods. *35th International Conference on Machine Learning, ICML 2018*, 4, 2587–2601.

Gaa, D., Chizhov, V., Peter, P., Weickert, J., & Adam, R. D. (2023). Gaining Insights into Denoising by Inpainting.

Gal, Y. & Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research* (pp. 1050–1059). New York, New York, USA: PMLR.

Gao, F. & Han, L. (2012). Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1), 259–277.

Ghani, S. N. & Barnes, L. (1972). Parameter optimization for unconstrained object functions - a bibliography. *Computer-Aided Design*, 4(5), 247–259.

Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research* (pp. 249–256). Chia Laguna Resort, Sardinia, Italy: PMLR.

Golay, M. J. E. (1958). Field Homogenizing Coils for Nuclear Spin Resonance Instrumentation. *Review of Scientific Instruments*, 29(4), 313–315.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, Massachusetts: MIT Press.

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232.

Gruetter, R. & Boesch, C. (1992). Fast, noniterative shimming of spatially localized signals. In vivo analysis of the magnetic field along axes. *Journal of Magnetic Resonance (1969)*, 96(2), 323–334.

Gruetter, R. & Tkáč, I. (2000). Field mapping without reference scan using asymmetric echo-planar techniques. *Magnetic Resonance in Medicine*, 43(2), 319–323.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *35th International Conference on Machine Learning, ICML 2018*, 5, 2976–2989.

Hansen, D. F. (2019). Using Deep Neural Networks to Reconstruct Non-uniformly Sampled NMR Spectra. *Journal of Biomolecular NMR*, 73(10-11), 577–585.

Helmus, J. J. & Jaroniec, C. P. (2013). Nmrglue: an open source Python package for the analysis of multidimensional NMR data. *Journal of Biomolecular NMR*, 55(4), 355–367.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2017). Rainbow: Combining Improvements in Deep Reinforcement Learning. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, (pp. 3215–3222).

Hey, T., Tansley, S., Tolle, K., & Gray, J. (2009). *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research.

Hinton, G. E. & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786), 504–507.

Holz, D., Jensen, D., Proksa, R., Tochtrop, M., & Vollmann, W. (1988). Automatic shimming for localized spectroscopy. *Medical Physics*, 15(6), 898–903.

Huang, S., Yang, J., Fong, S., & Zhao, Q. (2020). Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges. *Cancer Letters*, 471, 61–71.

Hull, W. E. (2003). NMR Tips for Shimming, Part I. Computerized Shimming with the Tuning Algorithm. *Bruker Spin Report*, 152/153.

Hull, W. E. (2004). NMR Tips for Shimming, Part II. Computerized shimming with the Simplex algorithm. *Bruker Spin Report*, 154/155.

Jahangiri, A., Han, X., Lesovoy, D., Agback, T., Agback, P., Achour, A., & Orekhov, V. (2023). NMR spectrum reconstruction as a pattern recognition problem. *Journal of Magnetic Resonance*, 346, 107342.

Jang, J. Y., Hwang, Y. J., Kim, J., Song, S., Kim, G., Choi, K., Hahn, S., Ahn, M., & Lee, S. (2023). Experimental Study of a Room-Temperature Shimming Technology Employing Genetic Algorithm for NMR/MRI Superconducting Magnets. *IEEE Transactions on Applied Superconductivity*, 33(5), 1–5.

Jiang, D., Dou, W., Vosters, L., Xu, X., Sun, Y., & Tan, T. (2018). Denoising of 3D magnetic resonance images with multi-channel residual learning of convolutional neural network. *Japanese Journal of Radiology*, 36(9), 566–574.

Jonas, E. (2019). Deep imitation learning for molecular inverse problems. *Advances in Neural Information Processing Systems*, 32(NeurIPS).

Juchem, C., Brown, P. B., Nixon, T. W., McIntyre, S., Rothman, D. L., & de Graaf, R. A. (2011). Multicoil shimming of the mouse brain. *Magnetic Resonance in Medicine*, 66(3), 893–900.

Juchem, C., Cudalbu, C., de Graaf, R. A., Gruetter, R., Henning, A., Hetherington, H. P., & Boer, V. O. (2021). B 0 shimming for in vivo magnetic resonance spectroscopy: Experts' consensus recommendations. *NMR in Biomedicine*, 34(5).

Julius, L. A. N., Scheidt, H., Krishnan, G., Becker, M., Nassar, O., Torres-Delgado, S. M., Mager, D., Badilita, V., & Korvink, J. G. (2023). Dynamic dielectrophoretic cell manipulation is enabled by an innovative electronics platform. *Biosensors and Bioelectronics: X*, 14(March), 100333.

Kaiser, J., Stein, O., & Eichler, A. (2022). Learning-based Optimisation of Particle Accelerators Under Partial Observability Without Real-World Training. *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, (pp. 10575–10585).

Keeler James (2006). Understanding NMR spectroscopy. *Choice Reviews Online*, 43(10), 43–5896.

Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V.-D., Bewley, A., & Shah, A. (2019). Learning to Drive in a Day. In *2019 International Conference on Robotics and Automation (ICRA)*, volume 2019-May (pp. 8248–8254).: IEEE.

Kiefer, J. & Wolfowitz, J. (1952). Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23(3), 462–466.

Kingma, D. P. & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.*

Kjær, L., Thomsen, C., Henriksen, O., Ring, P., Stubgaard, M., & Pedersen, E. J. (1987). Evaluation of Relaxation Time Measurements by Magnetic Resonance Imaging. *Acta Radiologica*, 28(3), 345–351.

Klee, V. & Minty, G. J. (1972). How Good is the Simplex Algorithm. In *Inequalities III* (pp. 159–175).

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., & Reblitz-Richardson, O. (2020). Captum: A unified and generic model interpretability library for PyTorch.

Korvink, J. G., MacKinnon, N., Badilita, V., & Jouda, M. (2019). "Small is beautiful" in NMR. *Journal of Magnetic Resonance*, 306, 112–117.

Krenn, M., Pollice, R., Guo, S. Y., Aldeghi, M., Cervera-Lierta, A., Friederich, P., dos Passos Gomes, G., Häse, F., Jinich, A., Nigam, A., Yao, Z., & Aspuru-Guzik, A. (2022). On scientific understanding with artificial intelligence. *Nature Reviews Physics*, 4(12), 761–769.

Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998). Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, 9(1), 112–147.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Lee, H., Lee, H. H., & Kim, H. (2020). Reconstruction of spectra from truncated free induction decays by deep learning in proton magnetic resonance spectroscopy. *Magnetic Resonance in Medicine*, 84(2), 559–568.

Lehmkuhl, S., Fleischer, S., Lohmann, L., Rosen, M. S., Chekmenev, E. Y., Adams, A., Theis, T., & Appelt, S. (2022). RASER MRI: Magnetic resonance images formed spontaneously exploiting cooperative nonlinear interaction. *Science Advances*, 8(28).

Levitt, M. H. (2008). *Spin dynamics: Basics of Nuclear Magnetic Resonance, Second Edition.* Chichester, UK: John Wiley & Sons, Ltd, second edi edition.

Li, D.-W., Hansen, A. L., Yuan, C., Bruschweiler-Li, L., & Brüschweiler, R. (2021). DEEP picker is a deep neural network for accurate deconvolution of complex two-dimensional NMR spectra. *Nature Communications*, 12(1), 5229.

Li, Y., Wolters, A. M., Malawey, P. V., Sweedler, J. V., & Webb, A. G. (1999). Multiple Solenoidal Microcoil Probes for High-Sensitivity, High-Throughput Nuclear Magnetic Resonance Spectroscopy. *Analytical Chemistry*, 71(21), 4815–4820.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., & Stoica, I. (2018). Tune: A Research Platform for Distributed Model Selection and Training. *ICML*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

Liu, S., Li, J., Bennett, K. C., Ganoe, B., Stauch, T., Head-Gordon, M., Hexemer, A., Ushizima, D., & Head-Gordon, T. (2019). Multiresolution 3D-DenseNet for Chemical Shift Prediction in NMR Crystallography. *The Journal of Physical Chemistry Letters*, 10(16), 4558–4565.

Lundberg, S. M. & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 4765–4774). Curran Associates, Inc.

Lundervold, A. S. & Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, 29(2), 102–127.

MacCulloch, K., Browning, A., Guarin Bedoya, D. O., McBride, S. J., Abdulmojeed, M. B., Dedesma, C., Goodson, B. M., Rosen, M. S., Chekmenev, E. Y., Yen, Y.-F., TomHon, P., & Theis, T. (2023). Facile hyperpolarization chemistry for molecular imaging and metabolic tracking of [1–13C]pyruvate in vivo. *Journal of Magnetic Resonance Open*, 16-17, 100129.

Manu, V. S., Olivieri, C., & Veglia, G. (2023). AI-designed NMR spectroscopy RF pulses for fast acquisition at high and ultra-high magnetic fields. *Nature Communications*, 14(1), 4144.

Marshall, I., Higinbotham, J., Bruce, S., & Freise, A. (1997). Use of voigt lineshape for quantification of in vivo 1 H spectra. *Magnetic Resonance in Medicine*, 37(5), 651–657.

McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133.

Metz, K. R., Lam, M. M., & Webb, A. G. (2000). Reference deconvolution: A simple and effective method for resolution enhancement in nuclear magnetic resonance spectroscopy. *Concepts in Magnetic Resonance*, 12(1), 21–42.

Michal, C. A. (2007). Magnetic field homogeneity: A new approach to orthogonalizing and optimizing shim gradients. *Journal of Magnetic Resonance*, 185(1), 110–117.

Mitchel, T. M. (1997). *Machine Learning*. McGraw-Hill.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *arXiv preprint*.

Morris, G. A., Barjat, H., & Home, T. J. (1997). Reference deconvolution methods. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 31(2-3), 197–257.

Nassar, O., Jouda, M., Rapp, M., Mager, D., Korvink, J. G., & MacKinnon, N. (2021). Integrated impedance sensing of liquid sample plug flow enables automated high throughput NMR spectroscopy. *Microsystems and Nanoengineering*, 7(1).

Nelder, J. A. & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7(4), 308–313.

Pan, S. J. & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32, 8024–8035.

Pearson, G. (1993). Shimming an NMR Magnet.

Persson, A. (2021). Unet Github.

Pham, N., Malinowski, A., & Bartczak, T. (2011). Comparative Study of Derivative Free Optimization Algorithms. *IEEE Transactions on Industrial Informatics*, 7(4), 592–600.

Prammer, M. G., Haselgrove, J. C., Shinnar, M., & Leigh, J. S. (1988). A new approach to automatic shimming. *Journal of Magnetic Resonance (1969)*, 77(1), 40–52.

Press, W. H. (2014). *Numerical Recipes*. Cambridge: Cambridge University Press, 3 edition.

Qu, X., Huang, Y., Lu, H., Qiu, T., Guo, D., Agback, T., Orekhov, V., & Chen, Z. (2020). Accelerated Nuclear Magnetic Resonance Spectroscopy with Deep Learning. *Angewandte Chemie International Edition*, 59(26), 10297–10300.

Reda, D., Tao, T., & van de Panne, M. (2020). Learning to Locomote: Understanding How Environment Design Matters for Deep Reinforcement Learning. In *Motion, Interaction and Games* (pp. 1–10). New York, NY, USA: ACM.

Rizzo, R., Dziadosz, M., Kyathanahally, S. P., Shamaei, A., & Kreis, R. (2023). Quantification of MR spectra by deep learning in an idealized setting: Investigation of forms of input, network architectures, optimization by ensembles of networks, and training bias. *Magnetic Resonance in Medicine*, 89(5), 1707–1727.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, , J. Hornegger, , W. M. Wells, , & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241).: Springer International Publishing.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.

Sacks, J. (1958). Asymptotic Distribution of Stochastic Approximation Procedures. *The Annals of Mathematical Statistics*, 29(2), 373–405.

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized Experience Replay. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, (pp. 1–21).

Schild, H. (1990). *MRI made easy.* Schering AG.

Schlenke, J., Hildebrand, L., Felsinger, S., Reusch, B., & Brecker, L. (2013). Automated Signal Detection as Tool to Evaluate Magnetic Field Homogeneity from Fourier Transformed Proton NMR Spectra. *Applied Magnetic Resonance*, 44(6), 745–758.

Schmid, N., Bruderer, S., Paruzzo, F., Fischetti, G., Toscano, G., Graf, D., Fey, M., Henrici, A., Ziebart, V., Heitmann, B., Grabner, H., Wegner, J., Sigel, R., & Wilhelm, D. (2023). Deconvolution of 1D NMR spectra: A deep learning-based approach. *Journal of Magnetic Resonance*, 347, 107357.

Schwenk, H. & Bengio, Y. (2000). Boosting Neural Networks. *Neural Computation*, 12(8), 1869–1887.

Shen, J., Rycyna, R. E., & Rothman, D. L. (1997). Improvements on an in vivo automatic shimming method (FASTERMAP). *Magnetic Resonance in Medicine*, 38(5), 834–839.

Shen, L., Zhao, W., & Xing, L. (2019). Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning. *Nature Biomedical Engineering*, 3(11), 880–888.

Shukla, V. K., Heller, G. T., & Hansen, D. F. (2023). Biomolecular NMR spectroscopy in the era of artificial intelligence. *Structure*, 31(11), 1360–1374.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958.

Strubell, E., Ganesh, A., & McCallum, A. (2020). Energy and Policy Considerations for Modern Deep Learning Research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09), 13693–13696.

Suefke, M., Lehmkuhl, S., Liebisch, A., Blümich, B., & Appelt, S. (2017). Para-hydrogen raser delivers sub-millihertz resolution in nuclear magnetic resonance. *Nature Physics*, 13(6), 568–572.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *30th International Conference on Machine Learning, ICML 2013*, 28(PART 3), 2176–2184.

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2nd edition.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 23–30).: IEEE.

Topgaard, D., Martin, R. W., Sakellariou, D., Meriles, C. A., & Pines, A. (2004). "Shim pulses" for NMR spectroscopy and imaging. *Proceedings of the National Academy of Sciences*, 101(51), 17576–17581.

Torralba, A. & Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR 2011* (pp. 1521–1528).: IEEE.

van de Sande, D. M. J., Merkofer, J. P., Amirrajab, S., Veta, M., van Sloun, R. J. G., Versluis, M. J., Jansen, J. F. A., van den Brink, J. S., & Breeuwer, M. (2023). A review of machine learning applications for the proton MR spectroscopy workflow. *Magnetic Resonance in Medicine*, 90(4), 1253–1270.

van Meerten, B. & Franssen, W. (2017). SHIMpanzee.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, 2017-Decem, 5999–6009.

Vinding, M. S., Skyum, B., Sangill, R., & Lund, T. E. (2019). Ultrafast (milliseconds), multidimensional RF pulse design with deep learning. *Magnetic Resonance in Medicine*, 82(2), 586–599.

Walter, J., Bekker, H., Blanchard, J., Budker, D., Figueroa, N. L., Wickenbrock, A., Zhang, Y., & Zhou, P. (2023). Fast Shimming Algorithm Based on Bayesian Optimization for Magnetic Resonance Based Dark Matter Search. *Annalen der Physik*.

Wang, G., Ye, J. C., & De Man, B. (2020). Deep learning for tomographic image reconstruction. *Nature Machine Intelligence*, 2(12), 737–748.

Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.

Webb, P. & Macovski, A. (1991). Rapid, fully automatic, arbitrary-volumein vivo shimming. *Magnetic Resonance in Medicine*, 20(1), 113–122.

Weiger, M. & Speck, T. (2011). Shimming for High-Resolution NMR Spectroscopy. In *Encyclopedia of Magnetic Resonance*. Chichester, UK: John Wiley & Sons, Ltd.

Weiger, M., Speck, T., & Fey, M. (2006). Gradient shimming with spectrum optimisation. *Journal of Magnetic Resonance*, 182(1), 38–48.

Willemink, M. J. & Noël, P. B. (2019). The evolution of image reconstruction for CT—from filtered back projection to artificial intelligence. *European Radiology*, 29(5), 2185–2195.

Williams, J. & Jonas, E. (2023). Rapid prediction of full spin systems using uncertainty-aware machine learning. *Chemical Science*, 14(39), 10902–10913.

Xiao, X., Wang, Q., Zhang, X., Jiang, B., & Liu, M. (2023). Restore High-Resolution Nuclear Magnetic Resonance Spectra from Inhomogeneous Magnetic Fields Using a Neural Network. *Analytical Chemistry*.

Yao, K., Liu, M., Zheng, Z., Shih, T., Xie, J., Sun, H., & Chen, Z. (2021). Automatic Shimming Method Using Compensation of Magnetic Susceptibilities and Adaptive Simplex for Low-Field NMR. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–12.

Zhang, H. & Yu, T. (2020). Taxonomy of Reinforcement Learning Algorithms. In *Deep Reinforcement Learning* (pp. 125–133). Singapore: Springer Singapore.

Zhang, Y., Hu, D., Hao, S., Liu, J., Quan, G., Zhang, Y., Ji, X., & Chen, Y. (2023). DREAM-Net: Deep Residual Error Iterative Minimization Network for Sparse-View CT Reconstruction. *IEEE Journal of Biomedical and Health Informatics*, 27(1), 480–491.

Zhao, D., Grist, J. T., Rose, H. E., Davies, N. P., Wilson, M., MacPherson, L., Abernethy, L. J., Avula, S., Pizer, B., Gutierrez, D. R., Jaspan, T., Morgan, P. S., Mitra, D., Bailey, S., Sawlani, V., Arvanitis, T. N., Sun, Y., & Peet, A. C. (2022). Metabolite selection for machine learning in childhood brain tumour classification. *NMR in Biomedicine*, 35(6).