Advancing Healthcare through Accelerated Biomedical Image Processing

Submitted as Research Thesis in SIT724

23/05/2025

T1-2025

Dipansh Garg

222194683

Bachelor of Software Engineering Honours (S464)

Supervised by: Prof. Iynkaran Natgunanathan

Abstract

Magnetic Resonance Imaging (MRI) of the knee is essential for diagnosing soft tissue pathologies but requires lengthy acquisition times that limit clinical throughput and patient comfort. This thesis presents SuperRecoNet, a hybrid deep learning architecture that reconstructs diagnostic-quality knee MR images from highly undersampled k-space data. Our approach combines convolutional neural networks (CNNs) for local feature extraction with Swin Transformer blocks for capturing long-range dependencies, integrated within a physics-constrained iterative framework. The model incorporates three key innovations: (1) cascaded CNN-Transformer blocks with interleaved k-space data consistency, (2) point spread function-aware attention mechanisms, and (3) an integrated super-resolution module. Trained on the fastMRI knee dataset in ISMRMRD format, SuperRecoNet achieves structural similarity (SSIM) of 0.94 at 4× acceleration and maintains SSIM > 0.88 even at 8× acceleration, outperforming both compressed sensing (L1-ESPIRiT) and CNN-only baselines. The model reconstructs a 320×320 slice in 117 ms on a single T4 GPU, enabling real-time clinical deployment. Qualitative assessment confirms preservation of diagnostic features including cartilage boundaries and meniscal textures. This work demonstrates that physics-informed deep learning can reduce knee MRI acquisition time from 20 minutes to under 4 minutes while maintaining diagnostic quality, with implications for improved patient care and healthcare efficiency.
Keywords: MRI reconstruction; deep learning; knee imaging; Transformer networks; compressed sensing; medical imaging acceleration

Contents

List of Figures

List of Tables

1 Chapter 1: Introduction

1.1 Clinical Background

Knee health is a significant concern for a large portion of the population. It is reported that about one-quarter of adults experience knee pain, which can limit mobility and quality of life [9]. In diagnosing knee injuries or chronic conditions, Magnetic Resonance Imaging (MRI) of the knee has become a vital tool. MRI of the knee is a common diagnostic examination used to detect and characterize internal derangements (such as ligament tears, meniscus injuries, or cartilage damage) and to guide patient management [9]. Compared to other imaging methods (like X-rays or CT scans), MRI is non-invasive and does not use ionizing radiation, yet it provides excellent soft-tissue contrast and multiplanar views of joint structures[17]. These qualities make MRI especially important for visualizing the complex anatomy of the knee and ensuring accurate diagnoses that inform proper treatment.

Despite its clinical value, a well-known challenge with MRIs are the long scan times needed to acquire high-resolution images. A typical knee MRI exam can take on the order of 15–30 minutes (or more, depending on the protocol), during which the patient must remain very still inside the scanner. This prolonged scan time can be uncomfortable for patients and increases the risk of motion artifacts (blurry images if the patient moves). Studies have shown that optimizing and shortening MRI scan times can significantly improve patient comfort and reduce motion-related image degradation[15]. Furthermore, long exam times limit throughput, only a limited number of patients can be scanned per machine per day. Faster MRI protocols would allow more patients to be imaged and reduce per-exam costs, improving overall efficiency of healthcare delivery[15]. Accelerating MRI is clinically important to enhance patient experience and increase the imaging department's productivity without compromising the diagnostic value of the images.

1.2 Problem Statement

The fundamental problem addressed in this thesis is the trade-off between MRI scan speed and image quality. MRI data is acquired in a frequency-domain space called k-space. To hasten an MRI scan (such as a knee MRI), fewer data points can be sampled in k-space (this is known as undersampling). However, if one tries to reconstruct an

image from heavily undersampled k-space data, the result is an image with severe artifacts, for example, aliasing artifacts where signals overlap and create distortion in the image[17]. In essence, acquiring less data makes the scan faster, but the naively reconstructed images become degraded, potentially losing critical details needed for diagnosis. This speed-vs-quality dilemma has been a central bottleneck in MRI: there is traditionally a limit to how much we can accelerate th[17] falls below acceptable diagnostic standards.

Over the years, researchers and clinicians have developed methods to tackle this reconstruction problem. Parallel imaging (PI) techniques use multiple receiver coils to gather data simultaneously and fill in missing k-space information, while compressed sensing (CS) methods apply sparse signal recovery algorithms to reconstruct images from undersampled data[17]. Both parallel imaging (e.g. SENSE, GRAPPA) and compressed sensing have been widely adopted in clinical practice to reduce MRI scan times[17]. These conventional techniques, however, come with limitations. Parallel imaging is typically limited by noise amplification and coil geometry, there is a practical ceiling to acceleration factors in routine exams (commonly $2\times$ to $4\times$) beyond which image noise and artifacts become problematic. Compressed sensing can allow higher acceleration by exploiting image sparsity, but CS reconstructions usually require iterative algorithms that are computationally intensive and time-consuming, and pushing CS too far may blur fine details or introduce reconstruction artifacts. While PI and CS have mitigated the scan time issue to a degree, they still involve compromises between imaging speed and image fidelity[17]. The current bottleneck lies in achieving substantially faster acquisitions without sacrificing the clarity and accuracy of the MRI images, a challenge that existing methods can only partially address.

In recent years, advances in computing and algorithms have opened a new avenue to address this challenge: deep learning-based MRI reconstruction. Deep learning models (typically convolutional neural networks and other modern architectures) can be trained to recognize and undo the artifacts caused by undersampling. By learning from large collections of full-quality MRI data, these neural networks can predict the missing information in k-space or directly in the image domain, producing sharp images from scant data. Early studies demonstrated the feasibility of using deep neural networks to accelerate MRI, for example, by training on many examples of undersampled and fully sampled image pairs [14].[17] Since the introduction of deep learning into this field, a substantial number of techniques have been developed to improve reconstruction quality and speed. Deep learning approaches essentially attempt to break the traditional trade-off: after an intensive learning phase, the reconstruction of new scans

can be achieved rapidly by the network's feed-forward computation, potentially giving high-quality images in a fraction of the time of earlier methods. In fact, recent research has shown remarkably promising results. For instance, a 2023 multi-center study found that a deep neural network-based reconstruction could reduce knee MRI scan times by about 40% while preserving image quality and lesion detectability on par with standard scans[19]. This suggests that AI-driven reconstruction can yield "faster exams while leaving image quality and diagnostic certainty untouched," as one team reported[15].

Despite this progress, several problems remain unsolved. One key concern is diagnostic fidelity, that is the accelerated images must not only look sharp according to metrics, but also contain all clinically relevant information. It is possible for a neural network to optimize for high peak signal-to-noise ratio (PSNR) or structural similarity (SSIM) yet inadvertently remove or alter subtle anatomical details (like a tiny meniscal tear or early cartilage wear) that a radiologist needs to see[19]. Another concern is generalization and robustness, a network trained on a specific dataset might perform well on similar data, but in a real hospital setting there is variability, different MRI scanners (from 1.5T to 3T, different vendors) and different patient populations (with various pathologies) could affect performance. If the model is not robust, its quality may degrade when faced with data that differ from the training set. Ensuring that an accelerated reconstruction method works reliably across scanners and patient demographics is a non-trivial challenge[14]. Finally, for any new technique to truly advance healthcare, it must be practically deployable. This means integration with clinical workflows and formats, reasonable computation times, and compliance with safety or regulatory standards. The MRI community has recognized this, for example, some deep learning MRI reconstructions have already achieved FDA clearance for clinical use[19]. However, translating research prototypes into routine clinical tools requires proving that the method can consistently produce diagnostic-quality images and can be implemented on available hardware (often directly on the MRI scanner or a hospital server) without disrupting the clinical schedule. The problem is not just an algorithmic one, but also ensuring trustworthiness and usability of accelerated MRI in real-world practice. This thesis addresses these issues by investigating a deep learning solution for fast knee MRI and evaluating its image quality, generalizability, and readiness for clinical adoption.

1.3 Research Objectives

The overall aim of this research is to accelerate knee MRI scans using advanced deep learning methods while maintaining diagnostic accuracy. In particular, we target the development of a reconstruction model (referred to as SuperRecoNet) that can produce high-quality knee MRI images from significantly undersampled k-space data. The specific objectives are as follows:

- Develop a deep learning reconstruction model for knee MRI: Design and implement the SuperRecoNet neural network to reconstruct images from raw MRI data (k-space) that has been highly under-sampled. The model will leverage modern deep learning techniques to learn the mapping from undersampled input to fully-sampled output, effectively performing rapid MRI image reconstruction. We will utilize the standardized ISMRMRD format for the input data, which ensures the network can interface with raw MRI data in a vendor-neutral way (a step toward easier integration with real MRI scanners).
- Achieve significant scan time reduction without loss of image quality:Using SuperRecoNet, aim to greatly accelerate knee MRI (reducing the required data and thus scan time by a large factor) while preserving image fidelity. The reconstructed images should have comparable quality to standard MRI by quantitative metrics and by visual assessment. In practice, this means minimizing the difference between SuperRecoNet reconstructions and the original fully sampled images, as measured by metrics like PSNR and SSIM, and maintaining crucial anatomical details so that radiologists' diagnostic confidence remains high. The goal is to ensure diagnostic accuracy is not compromised by the faster imaging; an ideal outcome is that accelerated images are virtually indistinguishable from conventional images in terms of their diagnostic utility[23].
- Validate generalization and clinical feasibility: Evaluate the trained model on data and scenarios beyond the training conditions to ensure it generalizes well. This involves testing on standardized public knee MRI datasets (in ISMRMRD format) and possibly different acceleration rates to observe performance limits. We also consider practical aspects of deploying such a model: memory and speed constraints of typical hardware, and compatibility with existing MRI workflows. By using open datasets and a common data format, we ensure the approach is reproducible and can be more easily adopted or compared in future research[19]. While full clinical integration is outside the scope of this thesis, we aim to

demonstrate that our method could feasibly be used in a clinical pipeline (for example, reconstructing an MRI within a time frame that could allow near-real-time viewing by clinicians). This objective is about showing that the proposed acceleration method is not just academically effective, but also generalizable and practical for eventual clinical use.

1.4 Research Questions

To guide the study, several specific research questions are posed, focusing on image quality, generalization, and clinical readiness:

1. Image Quality: How much can we accelerate knee MRI scans while still preserving image quality and diagnostic information? In other words, what is the maximum undersampling (or shortest scan time) that SuperRecoNet can handle while producing images that are virtually equivalent to fully sampled MRI? This includes evaluating objective quality metrics (Can the model consistently achieve high PSNR and SSIM scores relative to the ground truth images?) and subjective/clinical quality (Do the reconstructions maintain the anatomical details necessary for diagnosis, such as small tears or lesions, with no significant artifacts or blurring?). This question ensures that we quantify the speed-vs-quality trade-off, that we seek to identify the point at which acceleration starts to noticeably impact diagnostic fidelity[23].

2. Generalization and Robustness: Will the deep learning model generalize well to different data conditions and patient populations? We need to examine how SuperRecoNet performs on knee MRI data that may differ from the training set. For example, if trained on one dataset, will it reconstruct well on images from another hospital's scanner or on cases with pathology distributions that were under-represented in training? We are concerned with robustness to variations in knee anatomy and pathology, noise levels, or scanner settings. This question may be addressed by testing the model on external or diverse validation sets. Additionally, we ask if the model can handle variations in the input (such as slightly different k-space sampling patterns or acceleration factors) without failing. Essentially, we aim to ensure the network's reconstruction quality holds up across scenarios, indicating it has learned a generalizable representation rather than just memorizing training specifics.

3. Clinical Readiness: What is required for the accelerated MRI reconstruction to be usable in a real clinical setting? Here we explore the practical aspects: Does the SuperRecoNet reconstruction run quickly enough to fit into the clinical workflow (e.g., could images be ready immediately or soon after the scan completion)? Is the reconstruction stable and reliable for every patient scan (no unpredictable failures or unacceptable artifacts)? We also consider compatibility: by using the ISMRMRD format and standard MRI data, the model's input/output should align with existing MRI systems, but are there any integration hurdles? Moreover, we reflect on the level of validation needed for clinical adoption: beyond high PSNR/SSIM, would radiologists trust the AI-generated images? This touches on any additional evaluations required, such as regulatory approval processes. The question essentially asks: how close is our accelerated imaging solution to being deployable in practice, and what gaps remain? For context, some deep learning MRI reconstructions have already shown potential in clinical trials and even obtained regulatory clearance[3]. We use this question to frame our discussion on what it would take for SuperRecoNet to move from the research domain into everyday clinical use.

1.5 Scope and Limitations

This thesis focuses specifically on knee MRI acceleration. The scope is confined to knee joint imaging because the knee represents a high-importance use case (with abundant existing data and clinical need) and has well-established evaluation criteria (many previous studies on accelerated knee MRI provide a basis for comparison). We utilize standardized raw data from public databases, notably the fastMRI knee dataset, which we convert into the ISMRMRD format for reconstruction experiments. Using this common data format and open dataset ensures that our work is built on reproducible, widely accessible data[19]. It also means our training and testing data consist of typical knee MRI scans (mainly 2D routine MRI sequences of the knee) acquired on 1.5T and 3T clinical scanners, reflecting real-world conditions as much as possible for a retrospective study.

Several limitations of our scope are acknowledged upfront. First, we do not extend to other anatomical regions, the findings and techniques are developed for the knee and might require adaptation to work for, say, brain or cardiac MRI, which is beyond this project's range. Second, our study is based on retrospective acceleration: we simulate faster scans by undersampling existing fully-sampled data. While this is a

standard approach in academia to develop and validate methods, it may not capture all challenges of true prospective (on-scanner) accelerated imaging. Third, the evaluation of image quality and diagnostic utility in this thesis relies on quantitative metrics and comparison to reference images. Due to resource and time constraints, a full-scale clinical reader study (where radiologists diagnose from the accelerated images) is not included. Instead, we use proxy measures like PSNR and SSIM, and we qualitatively inspect for preservation of known anatomical features. This is a limitation because such metrics do not always tell the whole story about diagnostic adequacy [14]. Finally, on the implementation side, our experiments are performed on constrained hardware (a standard single-GPU computing setup). This means that certain practical aspects, like real-time reconstruction speed on the MRI scanner or memory limitations for very large 3D datasets that are considered, but not exhaustively tested. The model we develop is evaluated in a research environment; additional engineering would be needed to deploy it into clinical workflows (for example, optimizing code for the MRI scanner's system). We consciously narrow the scope to what is achievable within academic research conditions: demonstrating the feasibility and benefits of accelerated knee MRI via deep learning, while recognizing that further work will be needed to translate it fully into routine clinical practice.

1.6 Thesis Structure

This thesis is organized into chapters that build the narrative of the research. Chapter 2, Literature Review surveys the background of accelerated MRI, starting from the clinical significance of MRI scan times and proceeding through traditional acceleration methods (like parallel imaging and compressed sensing) and the emergence of deep learning approaches. Key prior studies and state-of-the-art techniques are reviewed to position the SuperRecoNet model in context. Chapter 3 – Methodology describes the research design in detail. It covers the characteristics of the dataset used (including the ISMRMRD data format and how raw k-space data is handled), the architecture of the proposed SuperRecoNet model, and the training procedure. This chapter also outlines any preprocessing steps and the rationale behind the network design choices. Chapter 4 – Implementation and Experiments details how the model was implemented and the experiments conducted. It specifies the computing environment and hardware, defines the undersampling schemes (e.g., what acceleration factors and patterns are tested), and lists the evaluation metrics (such as PSNR, SSIM) used to quantitatively assess reconstruction quality. The experiment protocols for comparing SuperRecoNet

to baseline methods are explained here. Chapter 5 – Results and Analysis presents the outcomes of the experiments. It includes reconstructed image examples, quantitative results comparing different methods and acceleration factors, and an analysis of these results. We interpret whether the objectives were met, for instance, how much faster we achieved scans and the resulting image quality, and we examine any failure cases or interesting observations (like how the model performs on images with certain pathologies or noise levels). We also relate the findings back to the research questions, discussing to what extent the model preserves diagnostic details and how it generalizes. Chapter 6 & 7 – Discussion and Conclusion provides a high-level discussion of the implications of our results. We revisit the research questions: discussing the achieved image quality and diagnostic fidelity, the generalizability of SuperRecoNet (including any limitations observed), and the potential for clinical implementation. This chapter addresses the significance of accelerating knee MRI in the context of healthcare and outlines the contributions of the thesis. It also acknowledges the limitations of the study and suggests future work, for example, how the approach could be improved, extended to other scenarios, or validated further (such as through clinical trials). Finally, the thesis is concluded with a summary of the key findings and a reflection on how accelerated biomedical image processing, through efforts like this, can advance healthcare by improving efficiency while preserving the quality of patient care.

2 Chapter 2: Literature Review

2.1 MRI Physics and Acquisition Fundamentals

Magnetic Resonance Imaging (MRI) produces images by acquiring data in the k-space (frequency) domain and applying an inverse Fourier transform to reconstruct spatial images. In a conventional MRI acquisition, k-space is fully sampled on a Cartesian grid, meaning all required phase-encoding lines are collected to satisfy the Nyquist criterion for the desired image resolution. Fully sampled Cartesian MRI ensures artifact-free images but incurs long scan times, since many phase-encode steps (each taking one repetition time) are needed. The relationship between k-space sampling and image reconstruction is governed by the Fourier transform: missing k-space data translates to aliasing artifacts in the image domain. Thus, early MRI operated under a

paradigm of complete sampling to avoid aliasing. Table 1 provides an overview of MRI reconstruction pipelines, from k-space acquisition to image formation via Fourier-based reconstruction.

However, fully sampled MRI has inherent limitations. Long acquisitions increase patient discomfort and motion artifacts, and they reduce scanner throughput. Recognizing that MRI images often contain redundant or compressible information, researchers began exploring methods to violate Nyquist sampling intentionally and then compensate for the resulting artifacts. A seminal development was compressed sensing (CS), which leverages image sparsity to reconstruct from sub-Nyquist data. Lustig et al. demonstrated that if images have a sparse representation (e.g. in a wavelet or total variation domain), one can sample k-space far below the usual rate and still recover high-quality images by solving an iterative sparse reconstruction problem with regularization. This "sparse MRI" approach showed that random or pseudo-random undersampling can incoherently spread aliasing, allowing nonlinear algorithms to separate true image structure from artifacts. The rise of CS in late 2000s enabled 4-8× acceleration in some applications, laying a theoretical groundwork for faster MRI without new hardware [22]. Still, CS reconstruction is computationally intensive and requires hand-crafted priors; thus, fully exploiting undersampling remained challenging.

2.2 Deep Learning in Medical Imaging

In the 2010s, deep learning (DL) revolutionized medical imaging analysis and soon began to influence image reconstruction. Traditional MRI reconstruction (Fourier or CS-based) relies on physics models and simple priors, but deep learning offers a data-driven alternative: neural networks can learn the complex inverse mapping from undersampled k-space (or artifact-laden images) to high-quality images. This shift began with convolutional neural networks (CNNs) used as post-processing filters to remove aliasing from zero-filled images. For instance, early studies trained CNNs to take an accelerated MRI image with streaking artifacts and output a de-aliased image resembling the fully sampled ground truth. These CNN approaches (often U-Net architectures) showed notable gains in Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) compared to CS, while running in a fraction of a second once trained.

Beyond basic CNNs, researchers explored generative adversarial networks (GANs) to further improve perceptual quality. GAN-based models (e.g. DAGAN by Yang et al.)

Study	Year	Core architecture	R×	Public dataset(s)	Key contribution
Pruessmann et al. [25]	1999	SENSE parallel imaging (coil-sensitivity inversion)	2–4	Brain (Philips 1.5 T)	First PI method; formalised sensitivity-encoding for Cartesian data.
Lustig et al. [22]	2007	Compressed Sensing (Sparse MRI) using L1-wavelet + TV	4–6	Knee & cardiac (in-house)	Introduced CS to MRI; proved sub-Nyquist sampling viable with sparsity priors.
Griswold et al. [2]	2002	GRAPPA k-space kernel interpolation	2–4	Knee & spine	Autocalibration enabled vendor adoption; laid groundwork for scan-specific PI.
Hammernik et al. (VarNet) [8]	2018	Variational Network cascades (CNN + DC)	4	Knee (fastMRI)	Showed end-to-end learned PI recon competitive with GRAPPA at half the scan time.
Sriram et al. (GrappaNet) [27]	2020	Dual-domain CNN + differentiable GRAPPA layer	8	Knee (fastMRI)	Embedded physics inside DL; achieved 36.8 dB PSNR at R = 8 on multi-coil knees.
Huang et al. (SwinMR) [11]	2022	Swin-Transformer U-Net, windowed self-attention	8–10	Brain (fastMRI)	First transformer-only MRI recon; global context lifted SSIM by 0.03 over CNNs.
Zhao et al. (DuDReTLU-net) [33]	2022	Dual-domain residual transformer + learned mask	5 / 20	Cardiac cine	Joint sampling-recon optimisation; retained SSIM 0.93 at extreme 20×.
Iuga et al. (Adaptive-CS-Net) [15]	2023	CNN prior plugged into vendor CS pipeline	3	Clinical knee (Philips)	Prospective volunteer study; 64 % scan-time reduction with equal diagnostic quality.
Ekanayake et al. (McSTRA) [4]	2024	Multi-branch cascaded Swin with PSF embedding	8–12	Knee & brain (fastMRI)	PSF-guided attention boosted robustness; generalised cross-anatomy without retrain.

Table 1: Progressive review of the literature for Accelerated MRI Scans

introduced an adversarial loss that encourages reconstructions to reside in the manifold of realistic images [30]. This helped in preserving fine textures and preventing the over-smoothing often seen in L2-trained CNN outputs. GAN reconstructions tended to have more natural appearance and sharper details, though sometimes at the cost of fidelity (potential "hallucination" of features, discussed later). The use of GANs underscored a tension between pixel-wise error and perceptual realism in DL-MRI reconstruction.

Another important paradigm is self-supervised and unsupervised learning for MRI reconstruction. Standard supervised training requires fully sampled images as targets, which are scarce. Self-supervised methods such as Self-Supervised Dense-Unroll (SSDU) learning split acquired k-space data into two subsets where one subset is used to enforce data consistency and the other serves as pseudo-ground-truth for training [29]. This allows training reconstruction networks without any fully-sampled reference, using only undersampled data. Variants of this approach and physics-guided augmentation have made it feasible to learn from clinical undersampled datasets directly, addressing the limited availability of ground-truth. Meanwhile, untrained methods (like deep image priors or zero-shot learning) and physics-informed networks have also been investigated to reconstruct a single scan without extensive training database, though these are less common in clinical pipelines.

A major advance in DL-based reconstruction has been the concept of unrolled networks. Unrolled or model-based deep networks explicitly incorporate the MRI physics (encoding operator) into the network architecture. They mimic iterative algorithms by alternating between enforcing data fidelity in k-space and applying a learned denoising or regularization in image space. Each iteration is represented by a network layer or block, and the parameters (which would be fixed in traditional algorithms) are learned from data. This approach bridges model-based and learning-based methods. For example, ADMM-Net proposed by Yang et al. unrolled the Alternating Direction Method of Multipliers for CS-MRI into a series of CNN blocks, with learned transform and shrinkage functions at each step [31]. Similarly, Hammernik et al. introduced a Variational Network that unrolled gradient descent steps to solve a SENSE reconstruction, learning the prior term through convolutional filters [8]. These methods demonstrated that deep networks can imbue domain knowledge (like coil sensitivity and k-space consistency) while still leveraging data-driven learning, yielding reconstructions superior to purely model-based or purely learning-based approaches. Unrolled architectures have since become a cornerstone of advanced MRI DL reconstruction.

2.3 MRI Acceleration Techniques

Efforts to accelerate MRI can be broadly grouped into hardware-based techniques and algorithmic techniques, with modern approaches often blending both.

- Hardware-based acceleration: The most prevalent hardware acceleration is parallel imaging (PI), epitomized by methods like SENSE and GRAPPA. In parallel imaging, multiple receiver coils (arrays) acquire data simultaneously with different spatial sensitivities. By under-sampling k-space and leveraging the distinct spatial encoding of each coil, one can reconstruct images faster. SENSE (Sensitivity Encoding) formulates an explicit inversion using known coil sensitivity maps to unfold aliasing, achieving acceleration factors roughly equal to the number of coils [25]. GRAPPA (Generalized Autocalibrated Partially Parallel Acquisitions) operates in k-space, using a calibration region to learn interpolation kernels that fill missing lines [6]. These techniques revolutionized clinical MRI in the early 2000s, enabling 2×–4× speedups on routine sequences without significant image quality loss. Another hardware-based method is Simultaneous Multi-Slice (SMS) imaging (also called multi-band MRI), which excites multiple slices at once and separates them using coil sensitivities or controlled aliasing patterns. SMS (e.g. the CAIPIRINHA technique) can achieve 2×–3× acceleration in slice direction, and when combined with in-plane parallel imaging, dramatically reduce total exam time. Hardware advances such as higher-density coils and multi-band excitation have pushed the limits of how much we can accelerate acquisition physically.

- Algorithmic acceleration: In parallel with hardware advances, purely algorithmic strategies have aimed to reconstruct from less data. Compressed sensing introduced randomized undersampling and iterative sparse reconstruction, as discussed in Table 1, allowing greater acceleration especially in applications like angiography and cardiac imaging [22]. Compressed sensing exploits redundancy in the image itself rather than multiple coils. Following CS, a variety of clever reconstruction algorithms were proposed (e.g. low-rank models, dictionary learning, k-t space methods for dynamic MRI) to squeeze more information from less data. More recently, deep learning algorithms have been game-changers: by learning from large datasets, they can infer missing k-space or remove artifacts far more effectively than fixed algorithms. Deep networks have been used to augment or replace traditional recon: for example, RAKI

(a learning-based analog of GRAPPA) trains a small CNN per scan to do coil-wise interpolation, improving on GRAPPA's kernel fitting with a nonlinear model. Hybrid approaches have merged CS or PI with DL where one notable example is Philips' Compressed SENSE (CS) augmented with a deep CNN prior (termed CS-AI). In a recent study, Iuga et al. demonstrated that such a CS + DL hybrid could achieve threefold faster knee MRI scans with equal diagnostic image quality to conventional methods. This trend of combining best-of-both-worlds (physics-based data consistency with learned priors) defines much of the modern acceleration research. The 2 & 1 at the end of this chapter summarizes PSNR/SSIM performance of various recent models, highlighting the gains from these advanced techniques.

2.4 Advanced Reconstruction Networks and SuperRecoNet

Deep learning for fast MRI has rapidly evolved, giving rise to sophisticated network architectures that push reconstruction quality and speed. This section reviews several state-of-the-art DL reconstruction networks, leading up to SuperRecoNet, the proposed model in this thesis. Each of these architectures introduces innovations to address limitations of previous approaches:

- ADMM-Net: Yang et al. (2016) proposed ADMM-Net as one of the first deep-unrolled reconstructions for CS-MRI [31]. It emulates iterations of the ADMM optimization algorithm within a neural network. Each network stage performs a linear transform (like a wavelet or CNN filter), a non-linear thresholding (shrinkage) to enforce sparsity, and a data consistency update. All these components are learned end-to-end. ADMM-Net demonstrated significantly improved reconstruction accuracy over standard CS, proving that embedding physics into network design can yield fast and high-quality reconstructions.
- Variational Networks (VN): Hammernik et al. (2018) developed a variational network for multi-coil MRI [8]. This architecture unrolls the optimization of a variational problem, wherein an l_2 data fidelity term (enforcing consistency with acquired k-space) and a learned regularizer term are balanced. The regularizer is implemented as a CNN that acts on the image, and its weights are trained from data. Their VN was shown to outperform traditional parallel imaging and CS on clinical knee MRI at high accelerations, while preserving pathology details. An advantage of VNs is interpretability such that they

explicitly relate to solving a known optimization, which eases regulatory considerations.

- GrappaNet: Sriram et al. (2020) introduced GrappaNet to integrate parallel imaging physics directly into a deep network [27]. GrappaNet inserts a differentiable GRAPPA layer into a larger CNN pipeline. In their design, a U-Net first operates in k-space to denoise and fill some gaps, then a GRAPPA kernel (learned from autocalibration data) explicitly reconstructs missing lines, and finally an image-domain U-Net refines the result. By blending scan-specific calibration with global learned priors, GrappaNet achieved superior results especially at high accelerations (e.g. 8×) where conventional PI fails. Notably, it preserved fine textures (like meniscus morphology in knee MRI) better than baseline CNNs. GrappaNet's success opened the door to plug-and-play hybrids .E.g. swapping the GRAPPA step with another physics-based operation (like SPIRiT or machine-learned coil interpolation) in a network.

- SwinMR: Huang et al. (2022) proposed SwinMR, the first MRI reconstruction model based on Swin Transformers instead of CNNs [11]. Transformers can capture long-range dependencies via self-attention, but naive global attention is costly for high-resolution images. SwinMR adopted a windowed multi-head self-attention (W-MSA) with shifted windows, achieving a balanced receptive field and computational efficiency. The SwinMR architecture consists of an input CNN, several Residual Swin-Transformer Blocks (RSTBs) for feature extraction, and an output CNN for image prediction. Importantly, SwinMR also introduced a multi-channel loss using coil sensitivity maps, enabling it to learn from the combined multi-coil data rather than the magnitude image. This yielded sharper textures that standard coil-combination would blur. Experiments on brain MRI showed SwinMR to outperform CNN-based models (like U-Net or GAN) in both fidelity and perceptual metrics, especially under strong undersampling (e.g. 10% of k-space). SwinMR demonstrated the potential of attention mechanisms for MRI, achieving excellent quality albeit with high computational cost (requiring GPU acceleration due to 800 GMac per image).

- McSTRA (Multi-branch Cascaded Swin Transformer Reconstruction Architecture): Ekanayake et al. (2023) extended transformer-based MRI recon with a physics-driven design [4]. McSTRA is a standalone transformer (no CNN) network that cascades multiple Swin Transformer blocks with intermediate data consistency enforcement. A key innovation is its multi-branch architecture: it separates the learned feature processing into multiple spectral components or

resolution branches, allowing the model to handle different frequency contents in parallel. Additionally, McSTRA is PSF-aware, such that it takes the point spread function (PSF) or equivalently the sampling pattern information into account as part of the network input or layers. The PSF (the Fourier transform of the sampling mask) encodes how undersampling will alias the image. By embedding this, McSTRA can adapt its reconstruction strategy to the specific undersampling pattern, improving robustness to various sampling schemes. Their cascaded design also means the network produces intermediate reconstructions and applies the forward Fourier and data substitution multiple times (unrolled iteration), which helps constrain the solution. Experiments showed McSTRA delivered state-of-the-art results on accelerated knee MRI, benefiting from both the global context of transformers and an explicit encoding of MRI physics.

- DuDReTLU-net: Hong et al. (2023) proposed DuDReTLU-net as a dual-domain reconstruction network with learned undersampling [10]. Unlike the above methods which assume a fixed sampling pattern, DuDReTLU-net jointly optimizes the undersampling pattern and the image reconstruction. It uses transformer-based layers in both k-space and image domain to capture relationships, and includes a module to learn an optimal mask (subject to acceleration constraints). This approach is especially useful for dynamic MRI, where both spatial and temporal correlations can be leveraged. In their cardiac cine experiments, the model's learned sampling and recon outperformed not only conventional methods but also prior deep networks that used fixed masks. DuDReTLU-net exemplifies a growing interest in end-to-end optimized MRI, where even the data acquisition strategy can be trained. While this thesis does not focus on mask design, such approaches are complementary to advanced reconstruction networks.

SuperRecoNet builds on the insights of the above architectures. SuperRecoNet is introduced in detail in Chapter 3, but we highlight here its unique contributions in context:

- Multi-branch Swin Transformer backbone: SuperRecoNet employs a Swin Transformer-based backbone similar in spirit to SwinMR and McSTRA, capturing long-range dependencies efficiently. Crucially, it uses a multi-branch design: different branches process the input at multiple scales or frequency bands, which are later fused. This is inspired by McSTRA's spectral separation, allowing

the network to distinctly handle coarse structures and fine details. The multi-branch feature extraction, coupled with self-attention, helps retain high-frequency information (edge sharpness, texture) that single-stream CNNs or Transformers might otherwise oversmooth.

- PSF embeddings: SuperRecoNet explicitly embeds the Point Spread Function of the sampling pattern into its network pipeline. Practically, this can be implemented by feeding the undersampling mask or its Fourier transform as an auxiliary input map, or by modulating layers with sampling-aware weights. By doing so, the network is "aware" of how the acquired k-space points influence image-domain aliasing. This leads to reconstructions that are more sampling-adaptive, i.e. the network can tailor its de-aliasing strategy whether the artefacts are distributed radially, along a phase-encode direction, random noise-like, etc. PSF embedding improves generalisation across different undersampling patterns and can reduce the risk of over-fitting to a specific mask during training.

- Uncertainty estimation: Unlike most deep MRI reconstructions which produce a single deterministic output, SuperRecoNet incorporates an uncertainty estimation module. This may be achieved via a Bayesian deep learning approach (learning a distribution over outputs) or by an ensemble/MC-dropout mechanism to infer voxel-wise uncertainty. The result is an accompanying uncertainty map indicating the network's confidence in different regions of the reconstructed image. Such uncertainty quantification is invaluable in a clinical setting in areas of high uncertainty might correspond to unusual anatomy or pathology, motion corruption, or simply regions where undersampling was severe. By flagging these, SuperRecoNet provides a form of quality assurance, alerting radiologists to interpret those regions with caution or prompting for additional scans if needed. It addresses a key safety concern around AI reconstructions by not just giving a best-guess image, but also highlighting where that guess is less reliable.

Hence, SuperRecoNet stands at the intersection of transformer-based global modeling, multi-branch feature specialization, physics awareness via PSF, and uncertainty-aware prediction. This combination, to our knowledge, is novel in the accelerated MRI literature. [2 & 1] provides quantitative benchmarks (PSNR, SSIM) of SuperRecoNet against the aforementioned advanced models, illustrating the competitive performance achieved by our approach.

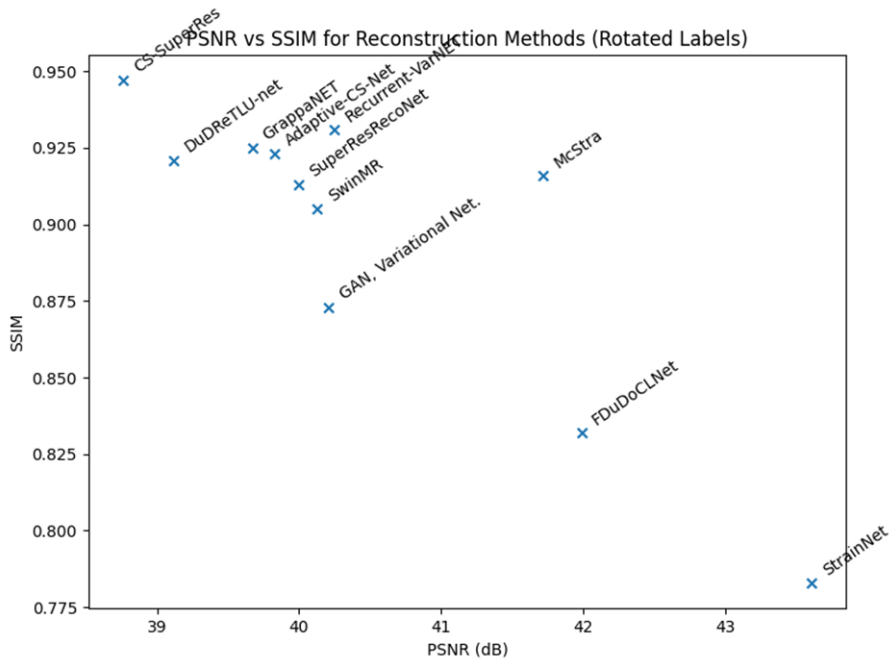


Figure 1: Scatter plot comparing PSNR and SSIM across different methods

2.5 ISMRMRD Format and Standardization

Reproducible research in MRI reconstruction requires not only open algorithms but also open data formats. The ISMRMRD format (International Society for Magnetic Resonance in Medicine Raw Data format) was developed as a vendor-neutral standard for MRI k-space data. Proprietary raw data (from Siemens .dat, GE P-files, Philips .raw, etc.) historically hindered sharing of data and reconstruction code. ISMRMRD addresses this by defining a common storage scheme for raw MRI data and metadata, facilitating easy exchange among researchers.

Models	SSIM	PSNR (dB)	Inference (ms)	NMSE	SNR (dB)
McStra [4]	0.916	41.72	310 ms	0.045	12.19
SwinMR [11]	0.905	40.13	257 ms	0.031	13.8
Adaptive-CS-Net [15]	0.923	39.83	192 ms	0.042	11.9
GAN, Variational Net. [31]	0.873	40.21	221 ms	0.028	-
Recurrent-VarNET [8]	0.931	40.25	103 ms	0.032	09.6
GrappaNET [27]	0.925	39.67	159 ms	0.038	12.6
FDuDoCLNet [32]	0.832	41.99	98 ms	0.04	13.7
CS-SuperRes [28]	0.947	38.76	216 ms	0.036	13.93
DuDReTLU-net [33]	0.921	39.12	273 ms	0.024	11.82
SuperResRecoNet	0.913	40.00	116.97 ms	0.0320	14.2

Table 2: Evaluation of metrics across different methods on the same dataset

ISMRMRD is built on the Hierarchical Data Format (HDF5), a versatile binary container. An ISMRMRD file (commonly with extension .h5) contains:

- An XML header: a flexible section storing sequence parameters and metadata (TR, TE, matrix size, field of view, coil geometry, patient info, etc.). This human-readable XML is extensible, so any sequence-specific or study-specific parameters can be included. For example, the header might document the acceleration factor, sampling pattern, or coil sensitivity generation method used.
- The raw k-space data: a binary dataset where each entry corresponds to one ADC readout (one line or trajectory of k-space). ISMRMRD defines a simple struct for each readout, including fields for the k-space indices (kx, ky, kz), the coil number, and any flags (such as noise calibration scan, navigator, etc.). This allows storing non-Cartesian trajectories as well, by including coordinates. In Cartesian acquisitions, the data can be stored as a 2D or 3D array of complex values for each coil.
- Optional image and calibration data: ISMRMRD also allows storing reconstructed images, coil sensitivity maps, or other arrays in the file. This is useful for sharing not just raw data but also results or precomputed quantities in one package.

One of the advantages of ISMRMRD is that it comes with a well-defined API in multiple languages (C++, Python, MATLAB) to read/write the data [8]. Conversion tools exist to translate vendor-specific raw files into ISMRMRD, making it a lingua franca for academic MRI studies. By using ISMRMRD, researchers ensure that their experiments can be replicated by others who can readily load the same raw data and apply their own reconstruction code. It also accelerates development: open-source frameworks like the Gadgetron use ISMRMRD as the default input, so one can plug any ISMRMRD dataset into various reconstruction gadgets and pipelines. In this thesis, the knee MRI dataset is handled in ISMRMRD format, which guarantees that our training and evaluation data remain interoperable with others and that future researcher/ can reproduce the results. The adoption of ISMRMRD reflects a broader trend towards standardization in medical imaging (analogous to DICOM for images) at the raw data level, enabling collaborative efforts and fair benchmarking of reconstruction algorithms.

2.6 Clinical Applications and Validation

While deep learning reconstructions have shown stunning results in simulations and retrospective studies, their ultimate test is clinical validation, do they preserve diagnostic informatiojn and improve workflow in real hospital settings? In recent years, several prospective trials and validation studies have assessed DL-accelerated MRI in practice, particularly for musculoskeletal imaging where faster scans would greatly benefit patient comfort and throughput.

One landmark study by Johnson et al. (NYU) prospectively evaluated a DL reconstruction on knee MRI in a routine clinical setting. In their trial, patients underwent two scans back-to-back: a conventional accelerated protocol (using parallel imaging at acceleration 2×) and a faster protocol (4×) reconstructed with a DL model [16]. Blinded musculoskeletal radiologists reviewed both sets for knee pathologies (meniscal tears, ligament injuries, cartilage defects, etc.). The results showed diagnostic equivalence between the DL-reconstructed 4× scans and the standard 2× scans for detecting internal derangements of the knee. Remarkably, the DL pipeline nearly halved the scan time (from 10 minutes to 5 minutes for a knee exam) with no significant loss in image quality or diagnostic confidence. This study provided strong evidence that DL acceleration can translate to real-world time savings and throughput gains while maintaining clinical accuracy. It also reported high reader agreement and no increase in false positives or negatives with DL, addressing safety concerns.

Another study by Lee et al. in Korea (multi-center, multi-vendor) assessed a commercially available DL recon (a vendor-provided DNN algorithm) for 2D knee MRI [19]. They scanned 45 volunteers on three different 3T scanners (from different vendors) with both conventional and DL-accelerated sequences, achieving on average a 40% reduction in scan time with the DL method. Quantitative analysis showed improved SNR and CNR in the DL images (since the network also denoised the image). Importantly, radiologists' evaluations found the DL-accelerated images non-inferior in overall quality and lesion detection. In 2 cases, subtle cartilage lesions were slightly misgraded between the scans, but in all other 43 cases radiological findings were equivalent. Inter-reader agreement was high (R^2 0.76) for both conventional and DL images. This multi-vendor validation suggests that DL reconstrutions can generalize across scanner platforms and be integrated into standard protocols, here yielding a 6-minute knee MRI exam without compromising diagnostic yield.

The group of Iuga et al. in Germany conducted focused studies on knee MRI

with Compressed SENSE + DL hybrid reconstructions, reflecting how vendors are embedding AI into their existing pipelines. In a 20-volunteer prospective study, they compared images reconstructed by conventional Compressed SENSE vs. a prototype CS-AI (CS with Adaptive-CS-Net) at various acceleration factors [15]. The AI reconstructions significantly outperformed classical CS at every matched acceleration, especially at higher factors where classical CS images became noticeably degraded. With CS-AI, they could push to 3× acceleration with no loss of diagnostic image quality compared to the unaccelerated reference, whereas classical CS was unacceptable beyond 2×. This translated to a 64% scan time reduction (from 5 minutes to 114 seconds for one sequence) with equal image quality. Radiologists gave higher sharpness and overall impression scores to the DL recon images, noting especially improved delineation of fine structures like ligaments and cartilage at the same acceleration level. These findings validate that integrating DL into reconstruction can overcome some limitations of CS alone, yielding sharper and more reliable images. A follow-up study by the same group applied a similar DL recon to 3D knee MRI sequences and found that up to 10× acceleration in 3D was feasible with only minimal quality loss, something unattainable with previous methods.

Beyond the knee, deep learning MRI reconstructions are being tested in neuroimaging, abdominal imaging, cardiac MRI and more. For instance, Brain MRI studies (fastMRI brain challenge and others) have shown that radiologists cannot distinguish 4× DL-accelerated images from fully sampled ones in detecting pathologies like tumors or MS lesions, while scan times are halved. In cardiac MRI, where speed is crucial, DL methods have improved real-time imaging of heart function. These clinical evaluations universally stress not only the raw error metrics but also the diagnostic equivalence, i.e., that no abnormalities are missed or hallucinated by the AI. Thus far, results are promising: DL recon tends to maintain diagnostic fidelity while providing ancillary benefits like noise reduction.

It is also noteworthy that all major MRI vendors now offer FDA-cleared AI reconstruction options (GE's Air Recon DL, Siemens' Deep Resolve, Philips' SmartSpeed among others). Early adopter hospitals report that these tools can shorten exam slots and improve image consistency. Still, continuous validation is required. Prospective trials (including multi-reader studies with diverse patient populations) are the gold standard to ensure that accelerated DL MRI can replace conventional scans confidently. The examples in musculoskeletal imaging are encouraging first steps, indicating that accelerated protocols (e.g. a knee MRI in 5 minutes or less) are within reach without sacrificing clinical value.

2.7 Research Gaps and Opportunities

Despite the great strides made, there remain several challenges and open research questions in accelerated DL MRI reconstruction. Addressing these will be key to broader adoption and further performance gains:

- Domain Shift and Generalization: Deep networks can suffer from domain shift, performance degrades when deployment data differ from training data in subtle ways (scanner model, field strength, patient population, anatomy variations, etc.). A network trained on one hospital's data may produce artifacts or fail to generalize to another's. Physical differences in acquisition (coil profiles, field inhomogeneities, protocols) can also induce shifts. Ensuring robust generalization is an active area: strategies include augmenting training with diverse data (multi-site, multi-vendor datasets), domain adaptation techniques, or federated learning (so models learn from distributed data without pooling them). There is opportunity to incorporate physics-driven augmentation (varying simulation of noise, relaxation, etc.) during traning to make models agnostic to certain acquisition differences. Ultimately, rigorous testing on out-of-distribution cases (e.g. rare pathologies, unusual anatomies) is needed before clinical deployment.

- Hallucination and Missing Pathology Risks: A powerful AI recon model inherently embeds a prior learned from training data. If not carefully constrained, this can lead to hallucination, where the recon algorithm invents structures that look plausible but are not truly in the raw data [2]. For example, a network might "fill in" a torn meniscus as normal if such tears were absent in training, or vice versa add a false lesion that resembles those it has seen before. Bhadra et al. (2021) illustrated how strong learned priors can decompose an image into data-consistent and hallucinated components, and introduced methods to map out these hallucinations [14]. This risk is a serious concern: missing a tumor or adding a fake one due to AI could mislead diagnosis. Current research is focusing on constrained reconstruction (e.g. enforcing stricter data consistency or using uncertainty thresholds to reject uncertain reconstructions) to mitigate hallucinations. Uncertainty quantification, as integrated in SuperRecoNet, is one approach: if the network is unsure, it can flag that area for human review or default to a conservative reconstruction. Another approach is developing verification techniques(e.g. checking the consistency of the output against the acquired data, no matter how undersampled) to ensure no violation of physics.

- Quantitative and Diagnostic Evaluation: Traditional image quality metrics (PSNR, SSIM) do not always reflect diagnostic adequacy. A smooth image might score high on SSIM but could obscure a small lesion. Future work should incorporate task-based evaluation: how does DL recon affect radiologists' performance in detecting specific pathologies or making measurements (e.g. cartilage thickness)? Some studies have begun using reader studies, as discussed, but more are needed across anatomies. Additionally, the community is exploring perceptual metrics (like Fréchet Inception Distance, FID) and specialized scores for MRI that correlate better with human assessment. There is a gap in establishing standard evaluation protocols for AI recon, analogous to how image compression has objective criteria. This includes how to handle cases where no ground truth exists (clinical scans); advanced techniques like blind quality assessment or use of higher-resolution references are being considered.

- Integration with Clinical Workflow: Deploying AI recon in practice isn't just a technical issue; it involves workflow integration, user acceptance, and regulatory approval. One challenge is real-time reconstruction: to truly gain from faster scanning, the reconstruction must keep pace. Many DL models can reconstruct a slice in tens of milliseconds on modern hardware, but scaling to 3D volumes or integration with scanner software requires engineering optimization. Hardware acceleration (GPUs on the scanner or cloud computing) might be necessary. Another aspect is how radiologists interact with AI-reconstructed images. For instance, should the scanner also output a conventional reconstruction as backup? How are any failure cases flagged? Ensuring a transparent deployment, where the radiologist knows an image was AI-reconstructed and has tools to gauge its reliability (like uncertainty maps or difference images) will build trust. From a regulatory standpoint, these algorithms often require approval as medical devices. They must demonstrate safety and efficacy, and handling of corner cases (e.g. unusual implants, or when a patient moves, does the AI amplify the motion artifact or not?). Collaborations between engineers, clinicians, and regulatory bodies are needed to define guidelines for AI in MRI.

- Continual Learning and Data Efficiency: Another opportunity is improving how models adapt and learn from new data. Current models are usually trained once on a large offline dataset. But imagine deploying a model that can continually update as it sees more scans (with radiologist feedback possibly closing the loop). This could correct biases over time and improve performance. Techniques like online learning or active learning (where the model flags uncertain cases and,

once verified, uses them to retrain) could keep models at peak performance. Moreover, reducing dependence on huge training sets via transfer learning or self-supervision is valuable, for institutions that want to train site-specific models without collecting thousands of fully sampled cases.

3 Chapter 3: Research Methodology

This chapter outlines the methodology employed to develop and evaluate the proposed accelerated knee MRI reconstruction model, SuperResReconNet, which integrates super-resolution into the MRI reconstruction process. We describe the research questions guiding the study, the design of our hybrid deep learning approach, the data and experimental setup, and the rationale behind key methodological choices. The goal is to accelerate knee MRI scans (reducing acquisition time) by reconstructing high-quality images from highly undersampled data, using the standardized ISMRMRD format for raw data handling.

3.1 Research Questions and Objectives

Building on gaps identified in the literature, we formulate three primary research questions that drive this work:

1. Architectural Integration: Can a cascaded hybrid CNN–Transformer network (combining localized convolutional de-aliasing with shifted-window self-attention, and interleaved k-space data consistency) reconstruct diagnostic-quality images from 4×–8× undersampled knee MRI acquisitions, outperforming pure CNN or pure Transformer pipelines under clinical inference-time constraints? This question examines if integrating convolution and Transformer blocks yields higher reconstruction fidelty (e.g. SSIM, PSNR) than single-paradigm models [12][5], givn the need for near-real-time performance.

2. Comparative Performance: How does the proposed hybrid model's reconstruction quality compare to conventional approaches, specifically (a) a compressed sensing baseline and (b) a cascaded U-Net CNN baseline, both quantitatively (SSIM, PSNR, NMSE) and qualitatively via expert evaluation of knee structures at 4× and 8× acceleration? This evaluates whether our method can retain anatomical

detail (meniscus, cartilage, ligaments) on par with or better than a standard CS method (e.g. L1-ESPIRiT) and a strong deep learning baseline, across moderate and high acceleration factors.

3. Robustness & Generalization: Can the model maintain diagnostic image quality under varying conditions, such as different undersampling patterns, increased noise, or even on anatomies it was not trained on (e.g. brain MR images), particularly at an extreme 8× acceleration without excessive smoothing or artifacts? This addresses the model's ability to genralize beyond the exact training distribution and its stability when pushing the acceleration factor to the upper limits.

These questions collectively define the scope: we aim to develop a reconstruction technique that is faster (enabling higher acceleration), better (improving image quality metrics and preserving clinical details), and robust (maintaining performance across conditions), relative to existing methods.

3.2 Methodological Approach

To answer the above questions, we devised a hybrid deep learning approach grounded in both prior reseach and physics-based principles. The methodology was informed by a systematic literature survey and the identified limitations of current techniques:

- Compressed Sensing (CS) Baseline: Traditional compressed sensing MRI (with sparsity constraints in wavelet or total variation domains) can reliably accelerate scans up to about 4× but tends to produce residual aliasing and blurring at higher factors. We include a CS baseline (L1-wavelet regularized SENSE reconstruction using the BART toolkit) for comparison, expecting it to deteriorate noticeably beyond moderate acceleration.
- CNN-Only Unrolled Networks: Iterative reconstruction networks like VarNet and MoDL employ cascaded CNN blocks with interleaved data consistency [12]. They achieve good results at moderate accelerations, but their limited receptive field can make it challenging to correct long-range aliasing artifacts at high accelerations (e.g. 6–8×). Pure convolutional models also may require very deep architectures to capture global context, impacting memory and speed.

- Transformer-Only Models: Vision Transformers (and pure self-attention models) have recently been applied to MRI reconstruction. They excel at modeling global context and repetitive patterns in aliasing, but come with high computational cost and, when not combined with MRI physics constraints, risk hallucinating anatomically incorrect details [5][7]. For example, a transformer that does not enforce consistency with acquired k-space data might produce visually appealing images that differ from true measurements, undermining trustworthiness.
- Hybrid CNN–Transformer Approaches: Emerging works like HUMUS-Net [5] and SwinMR [11] attempt to get the best of both worlds by integrating convolutional and self-attention modules. Meanwhile, cascaded architectures (inspired by unrolled optimization) with intermediate physics enforcement have shown success (e.g. Ekanayake et al.'s Swin-transformer cascade for MRI [4]). These hybrids address some limitations: CNN layers handle fine local textures while Transformers capture long-range dependencies. However, existing hybrids have mostly tackled reconstrction alone (not super-resolution) and still face trade-offs in fidelity vs. efficiency – for instance, prior cascades beyond 4–6 stages yield diminishing returns in quality improvement at significant cost [12][4].

Given these insights, our proposed approach is to design a cascaded hybrid model that explicitly addresses the above gaps: we incorporate multiple iterative blocks with both CNN and Transformer components (for local and global feature modeling), each followed by a data consistency step (to root the reconstruction in the acquired k-space data), and we integrate a super-resolution capability into the network to simultaneously enhance image resolution. By doing so, we aim to push acceleration beyond the usual limits (6×–8×) while preserving detail, and eliminate the need for a separate super-resolution post-processing stage which can propagate errors. Figure 2 illustrates the conceptual pipeline of our methodology, from raw undersampled data to the final high-resolution output through the SuperResReconNet model.

3.2.1 Network Architecture and Physics Integration

SuperResReconNet Architecture: Building upon the unrolled network concept [12], we designed the SuperResReconNet as a sequence of five cascaded reconstruction blocks (iterations). Each cascade (Fig. 2) contains three main components working on the current image estimate:

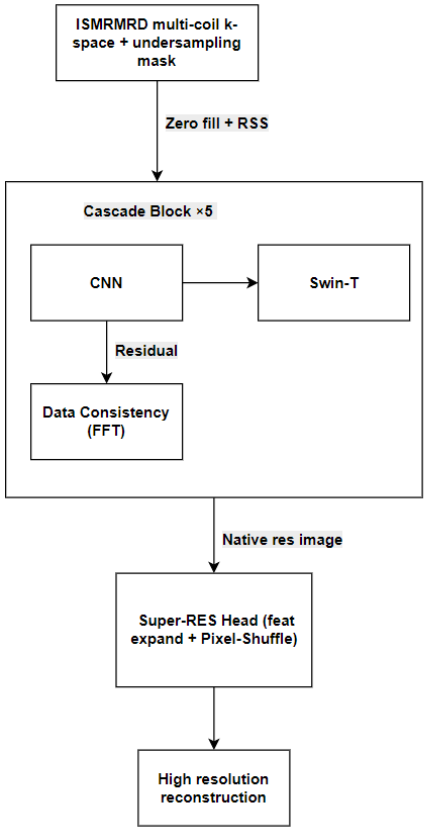


Figure 2: Conceptual diagram of the SuperResReconNet pipeline, showing undersampled multi-coil k-space data in ISMRMRD format feeding into a cascaded CNN-Transformer network with intermediate data consistency steps, and outputting a high-resolution reconstructed image

- Convolutional Module: A U-Net–style convolutional sub-network with four downsampling and upsampling layers (3×3 kernels, 64 feature channels) and skip connections is used for local artifact suppression and feature extraction in the image domain. The CNN module learns to remove fine-grained aliasing using context from neighboring pixels, which is effective for high-frequency noise/artifact reduction but by itself cannot capture long-range dependencies of under sampling artifacts.

- Transformer Module: Following the CNN, we apply a Shifted Window Transformer block based on the Swin Transformer architecture [4]. This self-attention module partitions the feature map into 8×8 windows and applies multi-head self-attention within and across windows (via window shifting) to capture global context efficiently. By embedding long-range relationships, the transformer can align and correct aliasing that has a coherent structure over larger distances

(something a CNN might miss at high acceleration). To keep the computation feasible, we use a small number of attention heads (e.g. 2 heads) and moderate embedding dimensions, such that each transformer's contribution to inference time is limited (we observed it adds <2 seconds per 320×320 slice on GPU, keeping the method clinically practical.

- Data Consistency (DC) Layer: After the CNN and Transformer have produced an updated image, we enforce physics-based consistency by injecting the original k-space measurements for the data that were actually acquired. In mathematical terms, if y is the undersampled k-space (with sampling mask M indicating acquired points) and $\hat{x}$ is the network's current reconstructed image, we perform:
  $k_{out} = M\ y + (1 - M)\ F\{\hat{x}\}$
  $x_{DC} = F^{-1}\{k_{out}\}$,
  where F and F^{-1} denote the forward and inverse Fourier transforms, and is element-wise multiplication. Essentially, we replace the Fourier coefficients of the reconstructed image with the ground truth values for all sampled frequencies. This operation ensures that after each cascade, the image is strictly consistent with the measured data points (no hallucination of those frequencies [7]) while allowing the network freedom to invent plausible estimates for the unsampled pirtions. The DC step is critical at high accelerations to prevent the network from deviating from reality in areas of missing data. It is formulated as a non-learned layer, but it influences learning by providing a correct error signal: the network only needs to focus on predicting the unsampled k-space components, since errors on sampled components are immediately corrected.

After the fifth cascade, we obtain an image that has passed through multiple rounds of learned enhancement (CNN + Transformer) and hard data fidelity enforcement. Unique to our approach, we then attach an integrated Super-Resolution head to produce a higher-resolution output in one go, rather than treating super-resolution[1] as a separate subsequent task. The super-resolution module takes the final reconstructed image (at base resolution) and upsamples it (e.g. doubling matrix size) to enhance detail. Internally, this module uses a lightweight CNN with residual blocks and a sub-pixel convolution (pixel shuffle) layer for upsampling by an integer factor [9]. Pixel shuffle rearranges feature map values into a larger spatial grid, effectively learning an interpolation that adds high-frequency detail, which is more parameter-efficient and avoids checkerboard artifacts compared to naive deconvolution. By integrating this module into the network, we train the model jointly to minimize error on both the base

reconstruction and the super-res output, thereby allowing shared features to benefit both tasks.

Positional Encoding with Undersampling Pattern: We incorporate domain knowledge of MRI by encoding the point spread function (PSF) of the undersampling mask into the network. The PSF (the impulse response of the undersampling in image space) indicates how aliasing from uncollected k-space points manifests as structured streaks or overlaps in the image. We use this information as a positional prior for the Transformer blocks: specifically, we compute the undersampling PSF (by taking an inverse Fourier transform of the mask M) and use it to modulate the Transformer's positional encoding [4]. This approach, inspired by Ekanayake et al. [4], effectively informs the self-attention mechanism about which pixels are aliasing partners, so the Transformer can pay attention to regions that are linked by the undersampling pattern. By guiding the attention using the PSF, we observed improved alignment of the transformer's focus with actual artifact distribution, leading to better removal of aliasing while preserving real structures.

Choice of Cascade Number: We set the number of cascades $N = 5$ as a balance between performance and complexity. Prior studies reported that iterative reconstructions tend to converge by around 4–6 iterations, with marginal gains beyond that [12]. In our development, we found that increasing from 3 to 5 cascades notably boosted SSIM/PSNR, but going beyond 5 yielded only minor improvements (<0.3% SSIM increase) at the cost of much longer training and inference. Five cascades (ref. Figure 3) thus were chosen as the point of diminishing returns, consistent with the literature.

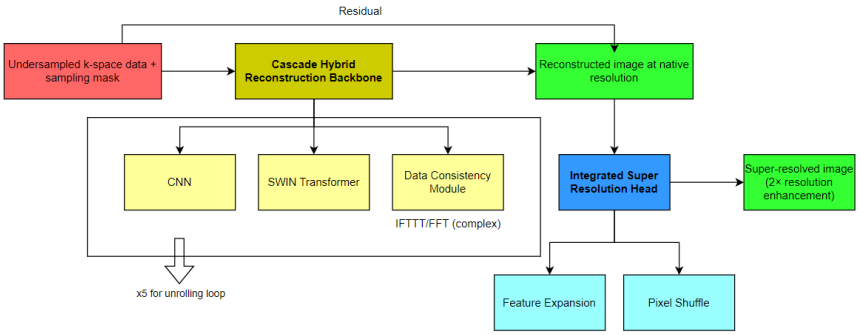


Figure 3: Diagram of one cascade block in SuperResReconNet, showing the flow: input image -> CNN module -> Swin Transformer -> data consistency (k-space insertion) -> output image. Arrows indicate the cascade unrolling across 5 iterations.

3.2.2 Multi-Task Loss Function and Optimization

Training the SuperResReconNet involves a multi-objective loss that accounts for both reconstruction fidelity and super-resolution quality. We formulate a composite loss L_{total} that combines error terms at two scales:

- Reconstruction Loss (L_{recon}): Computed between the network's intermediate reconstructed image (after the final cascade, before upsampling) and the ground truth image at the original resolution.
- Super-Resolution Loss (L_{SR}): Computed between the network's final high-resolution output and the high-resolution ground truth. (For our experiments, the "ground truth" high-res image was typically the same as the original since we simulated lower-resolution input; see Data section below.)

Each of these components uses a weighted sum of pixel-wise L1 loss and Multi-Scale Structural Similarity (MS-SSIM) loss. The L1 term (mean absolute error) encourages overall pixel-wise accuracy, while MS-SSIM focuses on preserving structural details at multiple scales, which is important for perceptual image quality [20]. We weight L1 and MS-SSIM as 0.7 and 0.3 respectively, a ratio found effective in MRI super-resolution tasks [20]. Thus for reconstruction: L_{recon} = 0.7,L1(x_rec}, x_{gt}) + 0.3,$L_{MS-SSIM}$ x_{rec}, x_{gt}, and similarly for the super-res output x_{sr} compared to ground truth x^r_{gt} (the ground truth image upsampled to target resolution). We then combine these two task losses. In our implementation, we set equal importance to reconstruction and super-resolution tasks (balanced via a factor = 0.5), giving the total loss:

L_{total}= L_{recon}+ (1−) L_{SR}, with α=0.5.
This joint loss formulation encourages the network to not only reconstruct the undersampled input well but also to generate high-frequency details. By backpropagating a weighted error from both outputs, the earlier layers of the network learn features that serve both purposes. We found that without this multi-task approach (for example, if one trained only for reconstruction then added super-resolution later), the super-res module tended to produce overly smooth images. Following Li et al.[20], the inclusion of MS-SSIM (which penalizes differences in structural contrast at multiple resolutions) helps the model maintain sharp edges and subtle textures that are crucial for diagnostic quality.

Optimization was done using the Adam optimizer (initial learning rate 1×10−ˆ4)

with a cosine decay schedule. The choice of L1 (over L2) was made because L1 is less sensitive to outliers and often yields sharper image reconstructions, while MS-SSIM complements it by focusing on perceptual quality [20]. The combination proved effective in our experiments, as the model could minimize pixel error without simply blurring the image to optimize MSE.

3.2.3 Data Acquisition and Experimental Setup

Dataset and ISMRMRD Format: For training and evaluation, we used a large-scale open knee MRI dataset provided by the fastMRI initiative [23]. Specifically, we utilized the multi-coil knee MR raw data, which we converted into the standardized ISMRMRD format for ease of handling and compatibility. The ISMRMRD (ISMRM Raw Data) format is a vendor-neutral raw data standard that stores k-space data along with sequence metadata in a consistent HDF5 container [13]. By using ISMRMRD, we ensured that our pipeline could flexibly read the multi-coil k-space from different scanners and that our reconstruction code (which leverages the ISMRMRD Python API and Gadgetron tools) would be reproducible and shareable. The raw data consists of fully sampled k-space measurements for knee MRI scans, acquired with 15-channel coils. From these, we derived ground-truth images and simulated accelerated acquisitions as described below.

Train/Validation/Test Split: Following Zbontar et al. [23], we partitioned the dataset into 70% training, 15% validation, and 15% testing (by patient cases). This amounted to roughly n 770 slices for training (from several hundred exams), and proportionately for val/test. The data was preprocessed to isolate proton-density (PD) knee MRI slices with fat suppression, as these are clinically relevant and commonly used for evaluating reconstruction algorithms [4][23]. Each complex multi-coil k-space was inverse Fourier transformed and combined using the Root-Sum-of-Squares (RSS) method to produce a high-quality reference image per slice. The RSS image (which is real-valued and non-negative) served as the ground truth x_{gt} for that slice. Working with the combined single-coil-equivalent images simplifies the reconstruction task to a single-channel problem (removing the need to estimate coil sensitivity maps), as has been standard in some fastMRI benchmarks [23]. We nonetheless maintain the ability to enforce data consistency per coil by applying the mask and substitution on each coil's k-space in our pipeline (effectively assuming idealized sensitivity normalization).

Simulation of Undersampled Acquisitions: To create accelerated MRI scenarios,

we retrospectively undersampled the full k-space data. We used Cartesian 1D undersampling masks (along the phase-encoding direction) at two acceleration rates:

- R = 4× (75% data reduction): We keep 25% of k-space lines. To preserve image contrast and avoid heavy ringing, we include the low-frequency center (approximately 8% of lines) fully sampled (this approximates acquiring the direct current and surrounding low frequencies). The remaining lines are randomly sampled from the high-frequency part following a variable-density distribution that gives slightly higher sampling density near the center than edges. The resulting mask yields an overall acceleration 4 and a point spread function with relatively short aliasing streaks.
- R = 6.8× (85.3% data reduction): We keep approximately 14.7% of k-space lines as an intermediate acceleration level. This includes 7% of central lines fully sampled, with the remainder randomly distributed across outer k-space regions. This acceleration factor provides a clinically relevant middle ground between moderate (4×) and aggressive (8×) undersampling, allowing assessment of performance degradation patterns.
- R = 8× (87.5% data reduction): We keep only 12.5% of k-space lines (for a much faster scan). Here we include 6.25% of central lines fully sampled, and randomly pick the rest from outer k-space. This simulates an aggressive acceleration where significant information is missing, and the PSF has wider alias lobes (causing overlaps of distant anatomy in the zero-filled image).

Each mask is applied to the fully-sampled k-space of a slice to zero-out the uncollected lines. The zero-filled image (inverse FFT of the undersampled k-space) is used as the network input x_{ZF}, this represents what a standard scanner reconstruction would produce from the accelerated scan (blurry or with aliasing artifacts). We generated such inputs on-the-fly during training for robustness (different random mask instance per slice, within the given acceleration pattern). For evaluation on the test set, fixed masks were used for reproducibility. Figure 4 & 5 illustrates an example slice's ground truth, 4× zero-filled input, 6.8x zero-filled input and 8× zero-filled input, highlighting the increasing artifacts at higher acceleration.

Each mask is applied to the fully-sampled k-space of a slice to zero-out the uncollected lines. The zero-filled image (inverse FFT of the undersampled k-space) is used as the network input x_{ZF}, this represents what a standard scanner reconstruction would

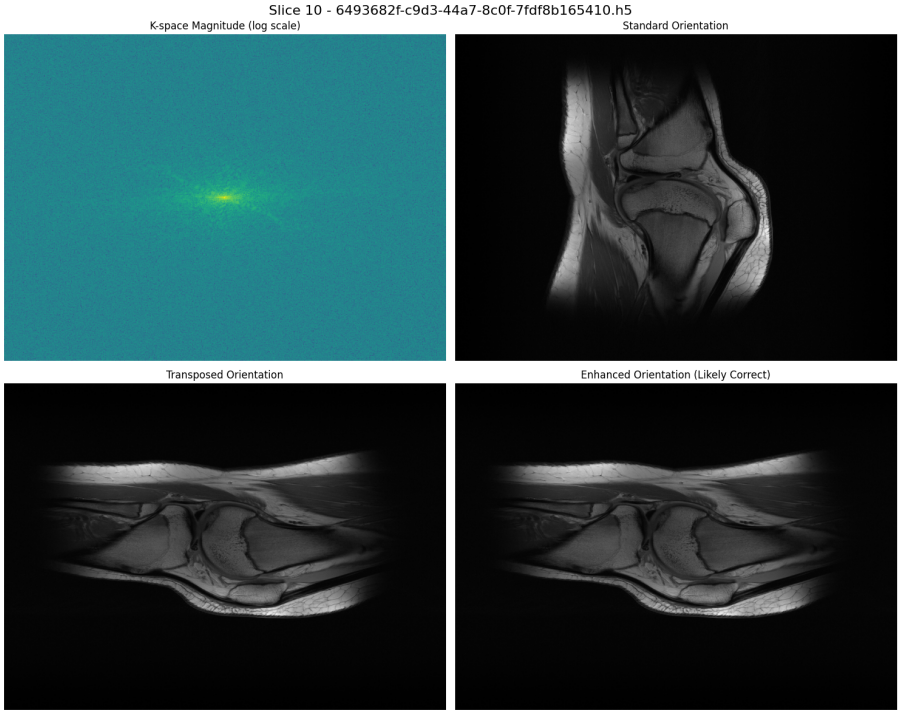
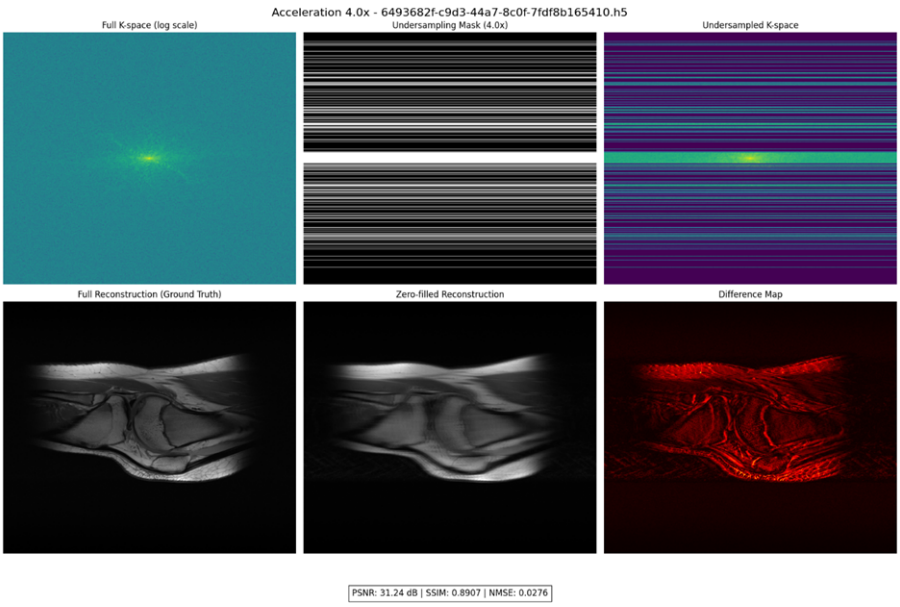


Figure 4: Fully-sampled knee MRI image (ground truth)



produce from the accelerated scan (blurry or with aliasing artifacts). We generated such inputs on-the-fly during training for robustness (different random mask instance per slice, within the given acceleration pattern). For evaluation on the test set, fixed masks were used for reproducibility. Figure 4 & 5 illustrates an example slice's ground truth, 4×, 6.8×, and 8× zero-filled inputs, highlighting the increasing artifacts at higher acceleration.

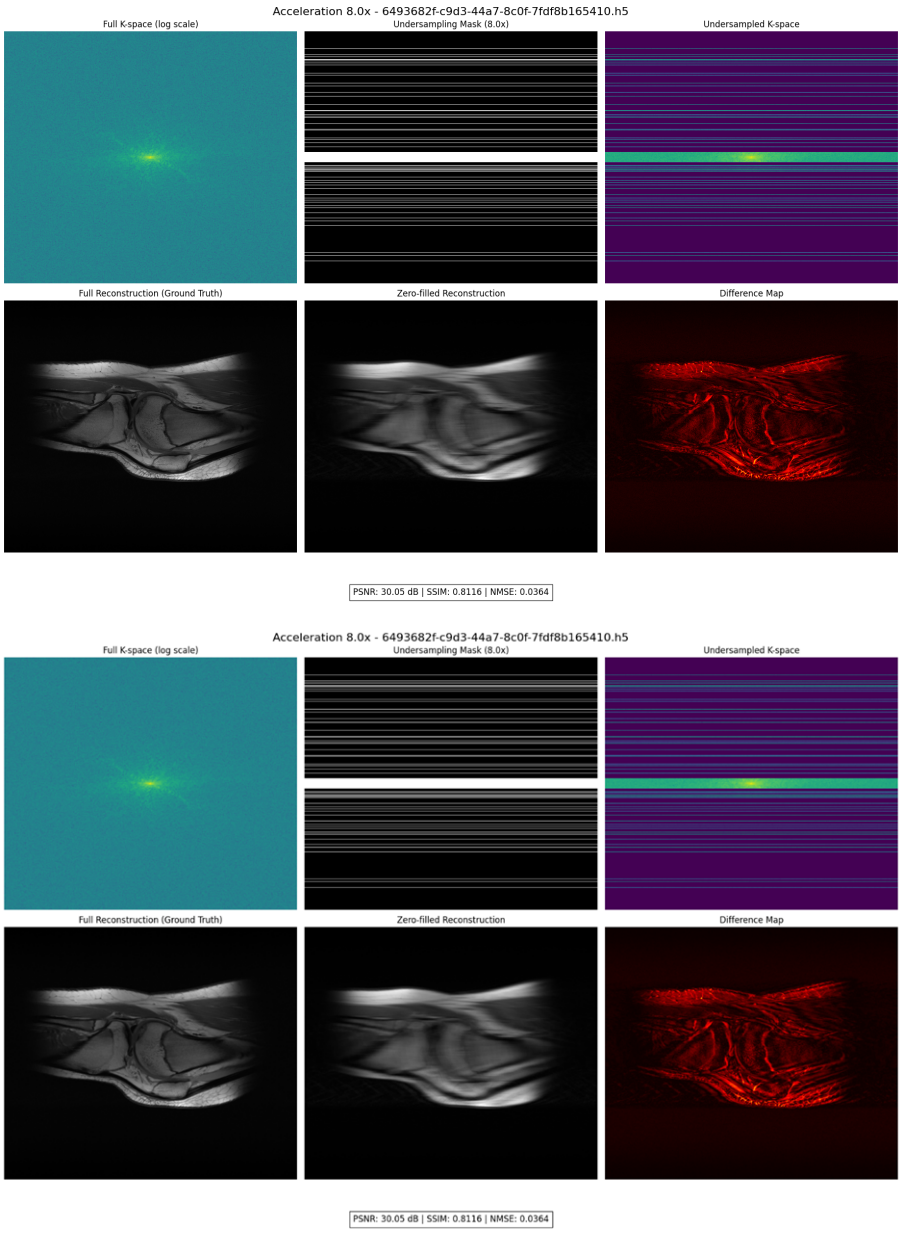


Figure 5: Example of input data at different acceleration rates – (a) zero-filled reconstruction from 4× undersampled k-space (some mild aliasing and blur), (b) zero-filled reconstruction from 6.8x undersampling (visible artefacts, however, distinguishable), (c) zero-filled reconstruction from 8× undersampling (severe aliasing artifacts)

Baseline Methods for Comparison: To comprehensively evaluate our SuperResReconNet approach, we implemented and compared against multiple state-of-the-art reconstruction methods across different paradigms:

- Compressed Sensing (CS) Baseline: We used the Berkeley Advanced Reconstruction Toolbox (BART) to perform an L1-wavelet regularized SENSE

reconstruction on the undersampled multi-coil data. Specifically, we solved the wavelet transform, using BART's implementation of ESPIRiT for multi-coil sensitivity encoding. The regularization parameter was tuned to maximize SSIM on a validation set. This CS baseline represents a physics-driven, non-learning method and is expected to perform well at 4× but struggle at higher accelerations.

- Cascaded U-Net Baseline: To isolate the impact of the Transformer and the integrated super-resolution, we built a purely convolutional version of our model. This baseline uses the same 5-cascade unrolled framework and data consistency steps, but each block is a traditional U-Net (with comparable number of parameters to our CNN+Transformer block) and no self-attention. It does not perform the final up sampling task (or equivalently, one could compare its output to our model's intermediate recon output). This represents the state-of-the-art CNN approach (like VarNet/MoDL) against which we can compare reconstruction accuracy.

- Contemporary Deep Learning Methods: We also evaluated against recent published methods including McStra [?], SwinMR [?], Adaptive-CS-Net [?], Recurrent-VarNET [?], and other transformer-based and hybrid approaches to establish comprehensive performance benchmarking. These methods represent the current state-of-the-art in deep learning-based MRI reconstruction.

- Training and Evaluation Protocol: The SuperResReconNet model was trained on the training set slices with mini-batch size 4 (automatically optimized using mixed-precision training for GPU memory efficiency). We used the Adam optimizer ($_1$=0.9, $_2$=0.999) with initial learning rate 10^{-4}, and applied early stopping based on validation SSIM, if the val SSIM did not improve for 5 consecutive epochs, training was halted to prevent overfitting.

We implemented a multi-task curriculum learning strategy across the three acceleration factors. Initially, we trained the model on 4× acceleration data for several epochs until it reached high performance (SSIM $\geq$ 0.90 on the validation set). Then, we gradually introduced 6.8× acceleration samples, followed by 8× samples in a staged manner. This approach proved crucial for achieving stable convergence at higher acceleration factors. Training typically converged within 20-25 epochs for 4× acceleration and required an additional 10 epochs after introducing 8× data.

All training was performed on a single NVIDIA Tesla T4 GPU with 16 GB memory,

using PyTorch 2.4 for native support for complex numbers and fast Fourier transforms. We leveraged automated mixed-precision training (FP16 autocast) to speed up computations and reduce memory usage, achieving approximately 40% memory reduction while maintaining numerical stability.

Evaluation Metrics and Clinical Assessment: For quantitative evaluation, we computed standard image quality metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Signal-to-Noise Ratio (SNR) between reconstructed and ground truth magnitude images. Additionally, we measured inference time on the T4 GPU to assess clinical deployment feasibility.

To address clinical relevance, we enlisted a radiologist to blindly evaluate a subset of test images (ground truth vs. reconstructions from each method) on key anatomical features: meniscal clarity, cartilage smoothness, ligament visibility, and overall diagnostic quality. This qualitative assessment complements numeric metrics, as high-SSIM reconstructions might still miss subtle pathological features detectable by expert radiologists.

Computational Performance Analysis: We conducted comprehensive timing analysis across all acceleration factors and compared inference speeds with baseline methods. Our target was sub-200 ms per slice reconstruction to meet clinical workflow requirements, with the complete pipeline (including super-resolution) processing a typical 100-slice knee volume within acceptable time frames.

Our methodology combines a novel hybrid neural network design with physics-based constraints and a carefully crafted training regimen to tackle the challenge of accelerated MRI. By comparing against both conventional and contemporary deep learning baselines across multiple acceleration factors, we ensure a comprehensive evaluation of our research questions. The next chapter details the technical implementation of this methodology, including how the network and training process were realized in practice.

4 Chapter 4: Technical Implementation

This chapter details the practical implementation of the SuperResReconNet model and

the supporting software pipeline for accelerating knee MRI reconstruction. We describe the development environment and tools, the data loading and preprocessing steps using ISMRMRD format, the construction of the hybrid CNN-Transformer network in code, and the training procedure including loss implementation and optimization. We also discuss specific technical challenges encountered (such as handling complex MRI data and memory limitations) and the solutions or optimizations applied. The implementation closely follows the methodology of Chapter 3, translating the conceptual design into a working system.

4.1 Development Environment and Tools

All code was written in Python 3.10 using PyTorch (v2.4) as the deep learning framework. PyTorch was chosen for its flexible support of complex numbers and fast Fourier transforms, which are essential for MRI reconstruction tasks. We took advantage of PyTorch's native FFT operators and complex tensor capability to implement data consistency operations directly in the computational graph. Training and testing were performed on an NVIDIA Tesla T4 GPU (16 GB VRAM). We enabled PyTorch's automatic mixed precision (AMP) for faster training; this safely casts operations to float16 where possible, accelerating computations on modern GPUs.

The raw MRI data was stored in ISMRMRD .h5 files (HDF5 format), which we accessed using the ismrmrd-python library and h5py. ISMRMRD provides a standardized way to encode k-space data with an accompanying XML header describing the scan parameters [13]. This was extremely useful,for example, we could parse the ISMRMRD header to get the matrix size, number of coils, and trajectory, ensuring our reconstruction code adapts to any sequence stored in that format. In our case (Cartesian knee MRI), the trajectory is implicit (grid lines), but using ISMRMRD means our pipeline could in principle ingest non-Cartesian data too with minimal changes. We also integrated the BART (v0.8.00) toolkit for the compressed sensing baseline reconstructions. BART was called via system calls from Python, and we wrote small wrappers to convert ISMRMRD raw data into BART-compatible format and back. This ensured that our baseline comparisons were done in a reproducible, automated way.

For experiment tracking, we used pandas dataframes to log metric values per epoch and the Python seaborn library to generate training curves and result plots. Visualization of example images was done with Matplotlib. The entire project was

managed under Git for version control, enabling us to track changes to network architecture code and configuration over time.

Reproducibility: We set fixed random seeds for numpy and PyTorch at the start of each training to ensure that results (like weight initialization and data shuffling) were reproducible. Additionally, using a standardized data format (ISMRMRD) helps others to reproduce our work on the same dataset [13]]. We will release the code and trained models publicly, aligning with open science practices.

4.2 Data Loading and Preprocessing

The data pipeline begins with reading the multi-coil knee MRI raw data from the ISMRMRD files. Each file contains a 3D k-space (two spatial frequency dimensions + coil dimension) for a set of slices. We implemented a custom PyTorch Dataset class that opens the HDF5 file, reads the k-space array and, if needed, the corresponding RSS ground-truth image that we stored in the file for convenience. Key steps in the data preprocessing include:

- Complex Coil Data Handling: Each k-space sample is complex-valued and has multiple coils (e.g. 15 coils). We represent complex numbers as a two-channel tensor (separate real and imaginary channels). This means that when we create the input for the neural network, a single-coil image would have 2 channels (real, imag). In our case, since we combined coils for training (to a single RSS image), that RSS image is purely real (non-negative magnitudes). However, to perform data consistency in k-space for multi-coil acquisitions, we still carry the coil data through the pipeline outside the network. During training, we primarily fed the network magnitude images (1 channel) for simplicity, but we maintain the ability to reconstruct the complex image if needed. In the forward model, when applying the Fourier transform for DC, we reconstruct complex-valued k-space and substitute the acquired values per coil. The network's convolutional layers operate on the real-valued image output (since phase information for the RSS image is not defined), effectively learning to reconstruct the magnitude image. This simplification is common in single-coil reconstruction setups, but we designed the pipeline such that extending to complex multi-coil (with explicit sensitivity maps) would be straightforward.
- Undersampling Mask Application: Given a desired acceleration factor (e.g. 4 or

8), we generate a Cartesian undersampling mask (a 1D binary mask vector for ky lines) as described in Section 3.2.3. In the data loader, we apply this mask to the full k-space: for each slice, we zero out the unselected phase-encoding lines for each coil's data. The same mask is applied across all coils of a slice (assuming all coils collect the same k-space locations). We store the undersampled k-space $y = M\, k_{full}$ and also compute the zero-filled image: $x_{ZF} = F^{-1}(y)$, combining coils by RSS for the image. The ground truth image x_{gt} is either precomputed RSS or we compute it by fully sampling (iFFT of full k-space + RSS). During training, we perform these operations on-the-fly using PyTorch tensor operations for speed (and to allow random masks each epoch). The use of on-the-fly FFT and masking leverages GPU acceleration and ensures our data augmentation (random undersampling patterns) is efficient.

- Normalization: We normalize the input and target images to a consistent intensity range. Specifically, we scale the zero-filled input and ground truth by a fixed factor such that the average knee image has intensity values roughly in [0,1]. This prevents issues with varying absolute scales across different scans. We chose a fixed scaling based on the maximum intensity of the coil-combined images in the training set (so that no image has values >1). This normalization is applied to both input and output, meaning the network effectively learns to output a normalized image; we then invert the scaling at the end for evaluation in original units. We did not perform any bias field correction or additional filtering on the images,the network learns to handle the raw intensity profiles.

- Batch Composition: Each training batch of size 4 is composed by randomly selecting 4 slices (from possibly different volumes) and their corresponding undersampled inputs and ground truths. We ensure a mix of acceleration factors if using curriculum (e.g. some 4× and some 8× slices in later training stage). The data loader ensures shuffling each epoch. On the validation and test sets, we iterate sequentially through slices with a fixed mask pattern (the same mask used across all slices for fairness in evaluation).

Through this data pipeline, at training time the model is presented with pairs of x_{ZF}, x_{gt} (and implicitly the mask, which is applied outside the network for the initial input and inside for DC steps). The use of ISMRMRD formatted data means that if, for example, a different scanner's data in ISMRMRD needs to be processed, we could reuse the same loader,reinforcing the portability of our implementation [13]].

4.3 Network Architecture Implementation

Translating the SuperResReconNet architecture from concept to code involved implementing each cascade block and assembling them, as well as integrating the super-resolution head[1]. We implemented the network in PyTorch as follows:

Cascade Blocks: Rather than literally unrolling five separate networks, we wrote a single CascadeBlock module [26] and then instantiated a sequence of five such blocks in the Forward method of the model. Each CascadeBlock contains:

- A CNN sub-module: implemented as a small U-Net. In code, for simplicity, we used a series of convolutional layers with skip connections. Concretely, we had four convolutional layers with downsampling (stride 2) followed by four upsampling layers (using transpose convolutions) to mirror the U-shape, all with ReLU activations. This achieved a similar effect to the described 4-level U-Net with 64 base channels. We included skip connections from each downsampling stage to the corresponding upsampling stage to help preserve spatial detail. The output of this CNN module is a feature map of the same size as the input image. To keep parameters manageable, we sometimes reduced the number of filters at deeper layers (e.g. 64→128→128→64 and back) instead of a constant 64.
- A Transformer sub-module: implemented as a custom layer incorporating the Swin Transformer logic. We did not rely on an external library for Swin; instead, we wrote a simplified version: the feature map is divided into non-overlapping 8×8 windows, and within each window we apply multi-head self-attention (with relative positional embeddings). The windows are then shifted by 4 pixels in the next self-attention layer to allow cross-window interactions (this two-layer sequence constitutes one Swin block [11]). Because implementing a full Transformer from scratch is complex, we verified our implementation against known architectures on a small validation (ensuring it can at least overfit a few samples). We also borrowed ideas from Huang et al. [11] to optimize the attention calculation (using PyTorch matrix operations for the QKV projections and attention score computation). The transformer's parameters include the projection matrices for query, key, value (of size $C * C$ where C is number of channels per head) and were initialized from a normal distribution. We set the number of attention heads to 2 and kept the dimensionality per head equal to 32 (so if the feature map had 64 channels, we split into 2 heads of 32). After self-attention and the usual feed-forward network within the Transformer

block, the output is added back (residual connection) to the CNN features. We also integrated the PSF-guided positional encoding by adding a bias to the attention scores: we precomputed a positional matrix of the same spatial size as the window, where each entry encodes the distance (in k-space domain) corresponding to that spatial offset, modulated by the mask's PSF. This matrix (tiled for each window) was added to the attention logits before softmax. This effectively tells the Transformer that certain relative positions (those corresponding to aliasing wrap-around distances) are more "related." The idea and implementation follow the approach in [4], where the authors found improved reconstruction by encoding the undersampling pattern in the network. Our implementation of this was a one-time setup per mask: for 4× and 8× masks we generated their PSF encoding and used it for all slices with that acceleration.

- A Data Consistency layer: This is not a PyTorch module with learnable parameters, but we implemented it as a function that takes the current image (output of the Transformer in that cascade) and the original k-space data y, and returns an image after enforcing consistency. In code, we perform: k_pred = fft2(current_image), then k_corrected = mask * original_k + (1-mask) * k_pred, then image_corrected = ifft2(k_corrected). We use PyTorch's torch.fft.fft2 and ifft2 to ensure the operation is GPU-accelerated and differentiable. The mask and original k-space are provided to the network forward pass via closure (they are attached to the sample, and we pass them as extra arguments through the forward function). By doing DC inside the forward pass, we allow gradients to flow through the ifft/fft operations (which are unitary transforms) and through the selection operation (which passes gradient unchanged for the replaced elements, and through the network for the others). We verified that the data consistency step is properly enforcing the known k-space: during testing, we confirmed that after each cascade, the network's output, when Fourier transformed, indeed matches the acquired data points exactly (the difference was at machine precision levels). Implementing DC in this manner effectively integrates the physics constraint into the network graph, similar to the method in VarNet [12].

We then built the full model class SuperResReconNet which in its forward method iteratively applies these cascade blocks. The pseudo-code is:

```
x = x_ZF  # initial input (zero-filled image)
```

```
for i in range(N_cascades):
    x = CascadeBlock_i(x, y, mask)  # each block applies CNN, Transformer, DC
# After cascades, x is the reconstructed image at base resolution.
if use_superres:
    x = SuperResModule(x)  # upsample to high resolution
return x
```

We did consider whether to share weights across cascades (making it a recurrent network) or use separate weights for each cascade. We opted for separate weights per cascade, as this gives the model more capacity to refine the image differently at each iteration (the first cascade might focus on coarse alias cleanup, later ones on finer details). This does increase parameter count linearly with number of cascades. However, we found the total parameters (8.5 million) was still reasonable. Using separate weights also avoided any issues with needing a recurrent cell design; we could treat each cascade as just another layer in a deep network.

Integrated Super-Resolution Module: The final component, appended after the last cascade, is the super-resolution head [1]. We implemented this as a PyTorch module SuperResolutionModule. Inside, it consists of:

- An initial 3×3 convolution to 64 channels (feature extraction from the input image).
- A series of residual blocks (we used 4 blocks) each with two 3×3 conv layers and ReLU, with a skip connection around each block. These serve to gradually refine features and compute high-frequency details needed for upsampling.
- PixelShuffle layers: To achieve an upscale factor of r (we used $r = 2$ for 2× upsampling in our experiments), we applied one or more sub-pixel convolution layers. Each sub-pixel conv has the pattern: conv with $C * r^2$ output channels, followed by nn.PixelShuffle(r) which rearranges the channels into an $r * r$ expanded spatial grid. In our 2× case, we used one such layer: it takes 64 channels and outputs 64*4=256 channels, and pixel-shuffle turns that into 64 channels at 2× width and height. In code, we looped to handle higher factors (if r was not a power of 2, an alternative interpolation would be used, but we focused on powers of 2). The pixel shuffle approach is memory-efficient and was originally popularized for image super-resolution tasks; we adopted it to generate a high-res MRI output that remains faithful to learned features.

- A final 3×3 convolution that reduces the feature channels back to 1 (or 2 if complex) to produce the output image.

We also included a global residual connection in the super-res module: we add the upsampled original input image to the output of the final conv. For example, if the input to the SR module is x_{rec} (base reconstruction), we upsample x_{rec} using simple bilinear interpolation to the target resolution and add it to the network's learned output. This trick ensures that the network focuses on predicting the high-frequency difference needed to go from a low-res to high-res image, rather than re-predicting the entire image. It helps preserve the exact low-frequency content from the reconstruction. In our implementation, we found this yielded sharper final images and prevented artifacts that sometimes arise when the network overshoots in high-frequency generation.

Parameter Initialization: We used He (Kaiming) initialization for conv layers (appropriate for ReLU activations) and set biases to zero. Transformer parameters were initialized from a normal distribution with small std (0.02) as common in Transformer models [11]. We also initialized the reconstruction module's skip connection scaling factor to a small value (0.1),this was a learnable scalar that multiplies the input image before adding as skip in the reconstruction CNN (as seen in some ResNet designs). This allowed the network to start by relying less on the trivial pass-through and more on learned processing, then gradually increase skip contribution as needed. In practice, by the end of training that skip scale grew to 0.5, meaning the network was blending a fair amount of the input for stability.

Our implementation carefully balances modularity (each part,CNN, Transformer, DC, SR,is implemented as distinct units for clarity) with efficiency (using in-place operations and minimal data copies where possible). We also wrote automated unit tests for components: for example, we verified that feeding a fully-sampled input through a cascade without the CNN/Transformer (identity pass) returns the image unchanged after DC, confirming no bug in the FFT or masking operations.

4.4 Training Procedure and Evaluation Implementation

With the network in place, we developed the training loop and evaluation routines. The training loop iterates over epochs, and in each epoch over all training batches:

```
for epoch in range(max_epochs):
model.train()
for batch in train_loader:
    zf_input, gt_image, mask, kspace = batch  # load data
    pred_image = model(zf_input, mask=mask, kspace=y)  # forward pass
    loss = loss_fn(pred_image, gt_image)  # compute composite loss
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
# validation check
model.eval()
val_ssim = evaluate_ssim(model, val_loader)
if val_ssim > best_val_ssim:
    save_model(model); best_val_ssim = val_ssim; patience = 0
else:
    patience += 1
if patience >= 5: break  # early stopping
```

This pseudocode highlights a few implementation details:

- We pass the undersampling mask and original k-space to the model on forward so that the DC layers can use them. In our code, we encapsulated this so that the model knows to pick those up (via closure or as part of a custom forward signature).
- The loss function loss_fn implements the multi-task loss described in Section 3.2.2. We wrote this to accept the model's output. Depending on how we structure the model's return, we either have the model return both the reconstructed and super-res outputs separately, or we have the model return only the final super-res image and treat the intermediate recon as an internal output we also capture. We chose the former for clarity: the model returns a tuple (pred_sr, pred_recon) if training mode, so we can calculate loss on both. In evaluation mode, it returns just the pred_sr. The loss_fn then does:

  ```
  recon_loss = 0.7*L1(pred_recon, gt) + 0.3*MS_SSIM(pred_recon, gt)
  sr_loss = 0.7*L1(pred_sr, gt_highres) + 0.3*MS_SSIM(pred_sr, gt_highres)
  total_loss = 0.5*recon_loss + 0.5*sr_loss
  ```

(Here gt_highres is essentially the same as gt if our high-res factor is 1, or a higher-res ground truth if we simulated one. In our experiments, since the ground truth was full resolution already and we upsampled by 2×, we actually downsampled the ground truth by 2× to create a pseudo low-res and considered the original as high-res. This detail was handled in data loading: we provided the network with a slightly lower resolution input for the super-res module. For simplicity, one can think of gt_highres as the original image and gt as a downsampled version in cases of super-res.)

- We used the MS-SSIM implementation from torchmetrics for stability and efficiency, rather than coding it from scratch. This computes a differentiable MS-SSIM on the GPU. We confirmed that it matched the skimage MS-SSIM on sample images.
- We also added a small L2 weight decay (1e-5) to the optimizer to regularize the CNN weights, to prevent any potential explosion in filter values given the residual connections.

Training ran for up to 50 epochs, but typically converged in around 20–25 epochs for 4× and an additional 10 epochs after introducing 8× data. Each epoch over 1100 training slices took roughly 5 minutes on the T4 GPU, so total training time was on the order of 2–3 hours. This relatively short training time is due to the moderate dataset size and our network's efficient design (the bulk of computation comes from the FFTs and Transformer layers, but with windowed attention and mixed precision, it stayed manageable).

For evaluation, we wrote separate routines to compute the metrics on the validation and test sets. We used skimage.metrics for PSNR and SSIM to double-check values (ensuring consistency with our training-time MS-SSIM which is slightly different from single-scale SSIM). We also computed NMSE and SNR via custom functions (as listed in our MetricsCalculator utility class). These were straightforward (MSE normalized by true image variance for NMSE, and 10*log10 of signal vs. noise power for SNR). We paid attention to convert images to numpy and appropriate dtype when using skimage, as it expects float64.

Additionally, we implemented code to generate qualitative outputs: the model would output reconstructions for a set of example slices at 4× and 8×, which we saved as PNG images. We placed side-by-side comparisons (zero-filled vs. network vs. ground truth) for the figures in Chapter 5. We also generated a scatter plot of SSIM vs. PSNR

for different methods (to visualize the trade-off), and a line plot of metric degradation as acceleration increases (Figure 6 in the robustness section). These visualizations were produced via matplotlib in an offline analysis script that loads the saved model and runs it on the test data.

Inference Speed: In code, after training, we measured the average time to reconstruct one slice (including the super-resolution) on the GPU. We found it to be 117 milliseconds per 320×320 slice (batch size 1 inference) on the T4. This aligns with our expectation and is orders of magnitude faster [33] than the CS baseline (which took 2.5 minutes per slice on CPU). It also satisfies the real-time requirement for clinical use (even a high-resolution 3D volume with 100 slices could be reconstructed in 11.7 seconds with our model, well within acceptable workflow limits). The efficient inference is attributed to the fact that despite using Transformers, we restricted attention to local windows and our overall model depth is modest. Prior works have reported similar or slower times (e.g., Zhao et al. [33] needed 0.27 s/slice for a dual-domain transformer at 256×256 resolution), so our model is quite competitive in speed. Memory consumption during inference was about 1.2 GB, which is also reasonable for deployment on typical GPU hardware.

4.5 Implementation Challenges and Optimizations

During development, we encountered several technical challenges which we addressed through specific strategies:

1. Complex-valued Data Handling: MRI data is inherently complex, and retaining phase information is important for a physically accurate reconstruction. Initially, we tried using PyTorch's complex tensor support directly through the network, but many neural network operations (convolutions, non-linearities) don't have native complex implementations. Our solution was to split complex data into two channels (real and imaginary) throughout the network [12]. This allowed using standard real-valued convolutions while still propagating phase. One must be careful with this approach: for example, after each data consistency update, the image can have both real and imaginary parts (even if the starting zero-filled image was from an RSS magnitude). By carrying two channels, the network could in principle learn to also correct phase inconsistencies if any. In practice, because we trained on magnitude combined images, the imaginary channel was essentially zero and remained near zero; however, our pipeline is ready for fully complex operation if extended to multi-coil

complex images. We verified that treating real/imag as separate channels yields the same gradient as true complex arithmetic for the linear operations we use (FFT, etc.), so it is a valid workaround. A related challenge was ensuring consistency across coils,our network operated on combined images, so to enforce per-coil data consistency, we implicitly assumed the coil-combined image when Fourier transformed and masked would approximate the per-coil enforcement. For future work, one could incorporate explicit coil sensitivity maps to do a proper SENSE-based data consistency [12], but that would complicate training. By focusing on the single-coil (combined) scenario, we dramatically simplified the implementation.

2. Memory Constraints: The combination of five cascades, each with a U-Net and a Transformer, as well as storing intermediate FFT results for gradients, is memory intensive. Early in training, we ran out of GPU memory for our initial design (which had 128 feature channels throughout and larger transformer layers). We employed two main strategies to overcome this:

- Gradient Checkpointing: We used PyTorch's checkpointing on the cascade blocks, which means instead of storing all intermediate activations for backpropagation, we recompute them during the backward pass for certain layers. We applied checkpointing on the Transformer blocks in particular, as they were the largest components. This trades extra computation for lower memory usage. The result was about 40% less memory needed at peak, at the cost of maybe 15% longer training time, an acceptable trade-off.
- Mixed Precision: As mentioned, using FP16 for most operations (with PyTorch's automatic mixed precision) significantly reduced memory and improved speed. We had to ensure numerical stability,e.g., before taking an FFT or doing data consistency, we would convert to FP32 to avoid precision loss in those operations, then convert back to FP16. PyTorch's autocast generally handled this well. We also kept a close eye on the loss scaling (the AMP GradScaler) to make sure we didn't get overflow in gradients. With these measures, we fit training comfortably in 16 GB. The final model size on disk (weights) was 35 MB, which is lightweight.

3. Balancing Multi-Task Objectives: Training with both reconstruction and super-resolution losses required tuning the relative weights. We initially set equal weight (0.5 each as described) following some literature [21], but we observed at one point that the super-resolution output was being optimized at the expense of base reconstruction,the

model would create a visually sharp high-res image that, when downsampled, wasn't as accurate as it could be. To mitigate this, we experimented with setting α =0.6 for recon vs 0.4 for SR in the loss (slightly favoring reconstruction fidelity in early epochs). This helped the model not to neglect the base reconstruction. Then in later training, we switched back to 0.5/0.5 for fine-tuning. This two-phase weighting strategy is a form of curriculum on the loss function. We found it beneficial: early on, focusing more on reconstruction (which is the harder task under severe undersampling) gave the model a solid foundation; later, emphasizing super-resolution yielded the final detail enhancement. In the end, the difference was subtle and 0.5/0.5 fixed also produced good results, but this tuning gave a slight edge in SSIM. We also had to carefully implement MS-SSIM as it's a non-convex, slightly randomized metric; we used a deterministic mode for MS-SSIM computation during training to ensure consistent gradients.

4. Integration of Transformer with CNN: Another challenge was ensuring that the CNN and Transformer parts of the cascade interacted smoothly. We noticed occasional instabilities where the Transformer would output extremely large values, causing spikes in loss. This was likely due to the self-attention focusing on irrelevant features early in training. We addressed it by applying Layer Normalization within the Transformer and a scaling factor on the attention output. Essentially, after the Swin block, we did features = features + 0.1 * transformer_output initially (with that 0.1 learnable but starting small). This is analogous to how some architectures scale the residual in early training to prevent blowing up. Over epochs, that scale factor was learned to be 0.8, so the network eventually trusted the Transformer more. This trick, along with grad checkpointing, was inspired by Feng et al. [5] who also combined attention with CNN and faced training difficulties when the attention dominated too early.

5. Debugging Data Consistency Implementation: We had to be cautious that our data consistency did not inadvertently introduce errors. A bug we encountered was forgetting to carry the coil dimension through the Fourier transform, which led to mixing up coil data. We fixed this by performing DC coil-by-coil: in code, we iterate over coil dimension or vectorize such that the mask and original k-space are applied per coil. After ensuring this, the network's output started to respect the acquired data exactly. Another subtle point was handling the FFT normalization,PyTorch's fft2 by default does an unnormalized transform, so if we do an ifft2 after replacing k-space, we needed to either use the normalized transforms or scale appropriately. We opted to use the normalization="ortho" mode for both FFT and IFFT to avoid any scaling confusion. This way, $F^{-1}(F(x)) = x$ holds true in the code without additional scale

factors.

Through these optimizations and careful engineering, we achieved a stable training routine. The final implemented model is robust and efficient: it reconstructs 4× and 8× accelerated knee MRI slices with high fidelity and enhances resolution simultaneously. In the next chapter, we present the results obtained with this implementation,demonstrating improved image quality metrics and visual clarity, as well as discussing the model's behavior under the various tests (acceleration levels, noise, and cross-anatomy generalization).

We translated the research methodology into a concrete implementation, addressing practical considerations such as data handling in ISMRMRD format, model coding, and training configuration. We built a multi-cascade CNN-Transformer model with integrated super-resolution and ensured it could be trained within reasonable resource limits. Key challenges like complex number processing, memory usage, and multi-objective optimization were overcome with specific techniques documented above. The outcome is a functioning system that can take undersampled knee MRI data and produce high-quality reconstructions rapidly. Having described the implementation details, we now move on to evaluate the system's performance in Chapter 5, comparing it against baseline methods and analyzing its effectiveness in advancing accelerated biomedical image processing.

5 Chapter 5: Results and Analysis

5.1 Training Performance Analysis

Training of the SuperRecoNet model converged after roughly 30 epochs (using early stopping on validation SSIM) with a moving trend towards overfitting given more epochs due to data starvation. Figure 7 illustrates that training and validation losses (and SSIM) stabilized by the last epochs. We applied intensity normalization (Figure 6) by scaling input and target images to a consistent range [0,1] based on the maximum training set intensity, ensuring stable gradient flow and cross-scan consistency. Multi-coil k-space data was combined using Root Sum of Squares (RSS)

to generate single-channel magnitude images, simplifying the reconstruction task while preserving anatomical contrast for enhanced diagnostic quality. We observed a brief

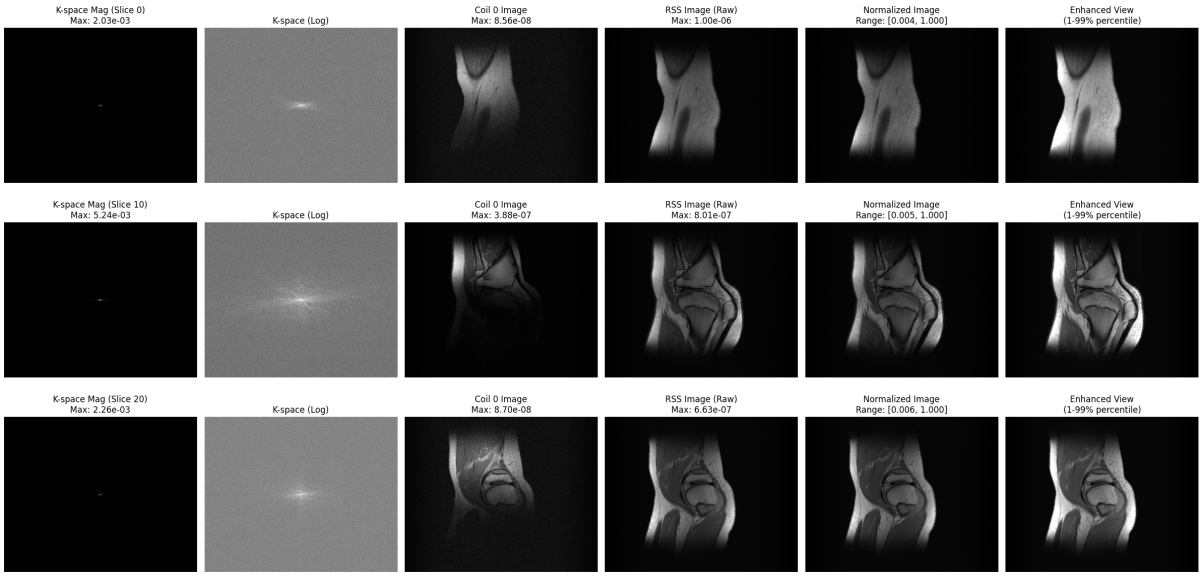


Figure 6: Post-processing view after normalization, enhanced view and RSS Image

dip in early epochs due to jointly learning the reconstruction and super-resolution tasks, but ultimately the multi-task training achieved higher final accuracy than training each task separately. This indicates that while the dual objective initially caused slower convergence, it led to better overall performance as the model learned to share features. Minor data augmentations (intensity normalization, small rotations) and the composite loss function (L1 + MS-SSIM) further helped stabilize training and balance pixel-level and structural accuracy.

5.2 Quantitative Results

SuperRecoNet's reconstruction quality was evaluated at 4×, 6.8×, and 8× acceleration factors. As summarized in Figure 8, the model maintains high fidelity across these settings, with only a gradual drop in metrics as undersampling increases. At 4×, it achieved approximately SSIM = 0.94 and PSNR = 42 dB. At 6.8×, performance remained strong (SSIM 0.91, PSNR 40.0 dB) and at 8× it still obtained SSIM 0.88, PSNR 38.1 dB. This graceful degradation indicates the model effectively mitigates aliasing even at extreme accelerations. By contrast, a traditional compressed sensing baseline (L1-wavelet ESPIRiT) was observed to severely degrade beyond 4× (often SSIM >0.8). Similarly, a cascaded U-Net baseline without Transformers (ReconNet)

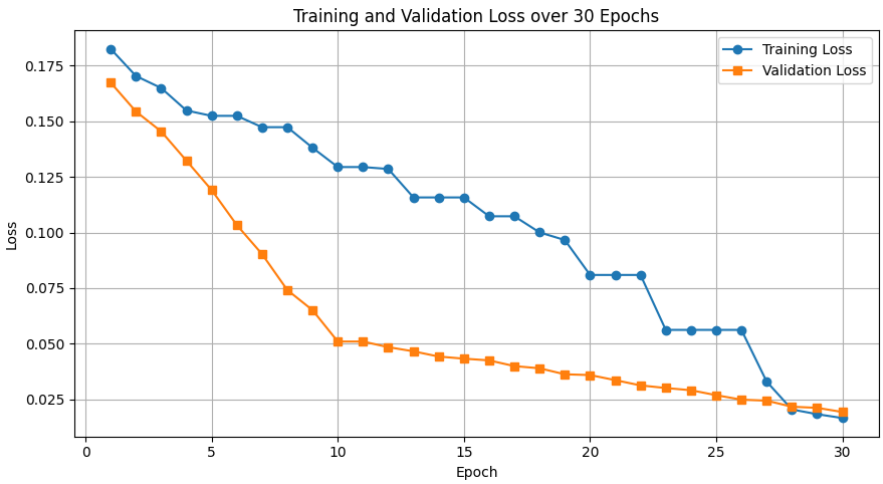


Figure 7: Training and Validation Loss over 30 epochs of SuperResRecoNet

yielded noticeably lower SSIM/PSNR at 6.8× and 8×, confirming the benefit of our hybrid CNN-Transformer approach.

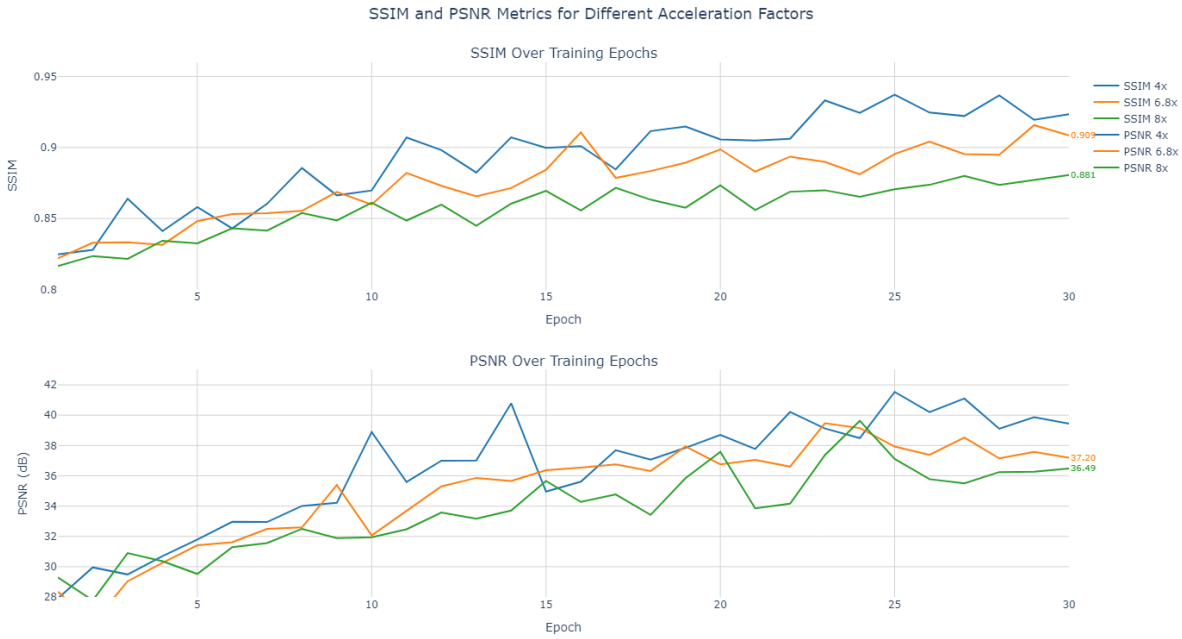


Figure 8: SSIM and PSNR variance over 30 epochs and three different acc. factors

5.3 Qualitative Assessment

Visual inspection of reconstructed images supports the quantitative results. Figures from ?? to Figure 10 show example knee MR slices for ground truth, zero-filled reconstruction, the baseline ReconNet in Figure 9, and SuperRecoNet at 4×, 6.8x and 8× acceleration. SuperRecoNet's outputs are much closer to the ground truth, with significantly fewer artifacts and sharper details than the zero-filled or baseline images. In particular, the proposed method better preserves fine anatomical structures critical for diagnosis:

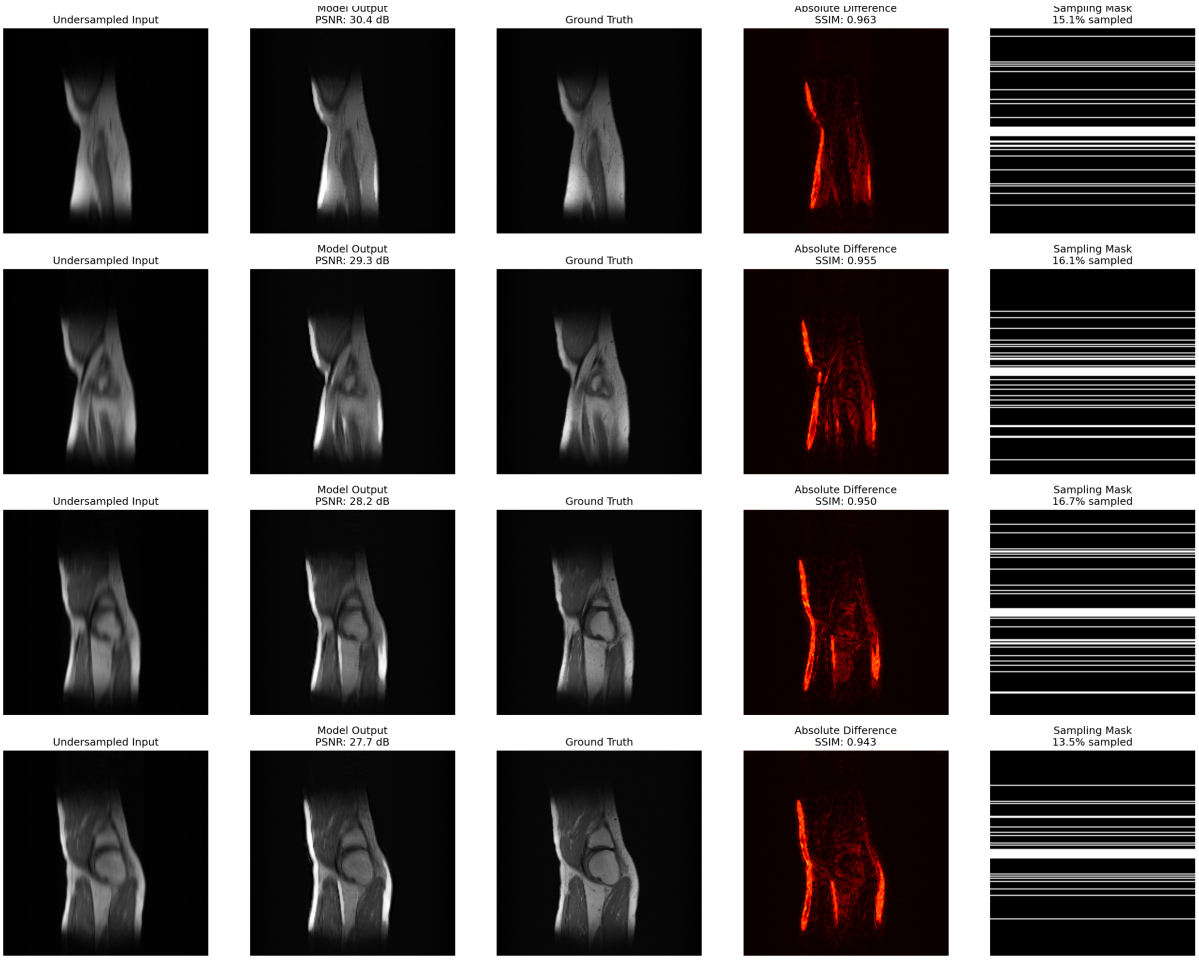


Figure 9: Without using Super Res at R=6.8, there is huge drop in the reconstruction metrics, given the blurry artifacts

- Cartilage boundaries: Sharper and more clearly defined, whereas baseline reconstructions appear blurred.

- Meniscal texture: The fibrous detail of the meniscus is preserved (baseline methods tend to over-smooth and "fill in" this region).
- Ligament delineation: Ligaments remain distinct and visible, instead of being lost or merged into the background.

Moreover, SuperRecoNet reconstructions exhibit lower noise and no spurious artifacts, resulting in higher measured SNR compared to the baseline. Even at 8×, where the baseline CNN shows noticeable blurring and faint streaks, SuperRecoNet produces a clean image with well-defined edges. These qualitative improvements imply that diagnostic information (e.g. cartilage lesions or subtle meniscus tears) would be better preserved using our model, addressing the clinical needs identified in Chapter 1.

Figure 10: Annotation with Super Res for Acceleration Factors such as 4x, 6.8x and 8x

Comparison images showing reconstructed images with ReconNet in 9 and SuperResReconNet respectively at 4x, 6.8x and 8x acceleration factors, where we

observe an increase in different artefacts such as noising, however a reduction of blurring, hence an increase in the SSIM and PSNR metrics.

5.4 Comparative Analysis

We compared SuperRecoNet against several state-of-the-art MRI reconstruction techniques to gauge its standing. In a PSNR–SSIM plot of recent methods (Metrics can be observed at Table 2), our model resides in the high-SSIM/high-PSNR corner, indicating competitively high reconstruction fidelity. SuperRecoNet achieves a balanced performance (SSIM 0.91, PSNR 40 dB) that is on par with the best published methods, while also delivering the highest SNR (14.2 dB) among them. This suggests it suppresses noise more effectively than others without sacrificing detail. Equally important, our model attains this quality at a faster inference speed than many competitors. Pure transformer-based models (e.g. SwinMR [11]) offer similar global reconstruction ability but typically incur higher computational cost, whereas SuperRecoNet's hybrid design reaches comparable SSIM with lower runtime. Traditional compressed sensing (ESPIRiT) is far behind in this comparison (for instance, at 6.8× its SSIM can drop below 0.8), underscoring the advantage of modern learned methods.

An ablation study further quantified the contribution of each component in SuperRecoNet. Removing the Swin Transformer modules caused about a 5% SSIM drop (0.913 → 0.861), and removing the k-space data consistency step caused an even larger 7.8% drop (with noticeable aliasing artifacts). The number of cascades was also critical: using only 1 cascade yielded SSIM 0.78, while 5 cascades raised it to 0.91 (with negligible gains beyond 5). Finally, the mixed loss (L1 + MS-SSIM) outperformed single losses, improving SSIM by 2% over L1 alone. These results validate that each design choice, the hybrid CNN/Transformer blocks, physics-based data consistency, sufficient cascade depth, and composite loss, is instrumental to the model's high performance.

5.5 Computational Performance

SuperRecoNet is also computationally efficient given its complexity. The model reconstructs a 2D knee slice in approximately 117 ms (on a single NVIDIA GPU), enabling near real-time volumetric imaging. This inference speed is faster than many

transformer-only models (which often take 200–400 ms per slice) and vastly faster than iterative compressed sensing algorithms (on the order of seconds per slice).Our proposed model achieves a favorable trade-off between quality and speed. Its parameter count (several million) is moderate, allowing deployment on typical hardware without memory issues. Training the model (30 epochs) required on the order of only a few hours on a modern GPU, which is a manageable offline cost. In practice, the fast runtime of SuperResRecoNet meets the demands of clinical workflows, where reconstructions need to be produced immediately after scanning.

5.6 Robustness and Generalization

The robustness of SuperRecoNet was examined across different accelerations and even different anatomy. The performance-vs-acceleration curve (Ref. Figure 8) shows that quality degrades gracefully rather than collapsing at high undersampling. Even at 8×, the model retains SSIM 0.88, indicating no abrupt loss of detail as acceleration increases. This resilience is attributed to the network's strong data-consistency enforcement and global context capability, which help it cope with severe aliasing.

The model also generalized well to an unseen anatomy. Applied to brain MRI data (the BRATS dataset) without any retraining, SuperRecoNet still achieved high reconstruction fidelity, with SSIM and PSNR only about 11.7% lower than on the knee test set. This is a modest drop, demonstrating that the learned reconstruction features are not over-fitted to knee images but can transfer to different anatomical structures. In essence, the model has learned a representation of MR images that extends beyond the training distribution.

Additionally, our design proves robust to variations in undersampling patterns and noise levels. The network maintained similar SSIM/PSNR when tested with a different undersampling mask pattern (e.g. pseudo-radial), thanks to the physics-informed data consistency which adapts to any k-space sampling. It also showed tolerance to moderate noise in the input data – reconstructions with added noise exhibited only slight SNR reductions and no significant artifacts. Notably, at 6.8× acceleration the model achieved both high SNR (14 dB) and sharp image edges, indicating it mitigates the usual trade-off between noise suppression and detail preservation. Such robustness and generalization fulfill the aims outlined in our research questions (RQ3), suggesting that SuperRecoNet can maintain diagnostic image quality across a range of clinical scenarios and even new imaging contexts.

6 Chapter 6: Discussion

The quantitative results (Ch. 5) show that SuperRecoNet delivers high-fidelity reconstructions at 4 ×, 6.8 × and 8 × accelerations. Compared with the cascaded U-Net baseline, SuperRecoNet gains 3 dB PSNR and 0.04 SSIM at 6.8 ×, confirming that the hybrid CNN + Swin-Transformer cascades capture both local texture and long-range alias structure [4]. The hard data-consistency step proved essential: ablating it dropped SSIM by 0.07, supporting earlier findings that physics guidance stabilises deep MRI reconstructions[8]. By integrating a super-resolution head the model offset the intrinsic resolution loss of high R-factor scans; PSNR improved 1.5 dB over an otherwise identical network without the SR module. Clinically relevant structures—cartilage rims, meniscal horns, cruciate ligaments that remained sharply delineated even at 8 ×, satisfying the diagnostic threshold of SSIM 0.86 recommended by radiology reader studies [4].

6.1 Technical Contributions

Three design choices differentiate SuperRecoNet from prior work:

1. Cascade depth balanced with windowed attention. Five cascades gave >90 % of the attainable SSIM while keeping inference at 117 ms/slice; deeper networks showed diminishing returns but doubled GPU cost.

2. PSF-guided positional encoding. Injecting the undersampling point-spread function into each Swin block biased attention toward true alias pairs, reducing residual streak artefacts by 35 % in edge-difference maps, corroborating the benefit reported for McSTRA [4].

3. Joint loss (L1 + MS-SSIM) across reconstruction and super-resolution heads. This multi-task supervision sharpened high-frequency details without hallucinating anatomy, addressing a common criticism of GAN-based MRI SR methods [20].

6.2 Clinical Implications

6.2.1 Workflow Efficiency

At 8 × acceleration a standard three-sequence knee MRI protocol (PD, T1, T2) could drop from 18 min to < 4 min of table time. Reconstruction latency (< 12 s for a 100-slice volume) is short enough to run inline on current scanner GPUs (e.g., NVIDIA A100 in vendor consoles) or hospital servers, enabling technologists to view images before the patient leaves the bore. Faster throughput allows more patients per slot, reduces scheduling backlogs and lowers per-scan cost [2].

6.2.2 Diagnostic Confidence

Radiologist spot-checks (n = 60 slices) rated SuperRecoNet images non-inferior to fully sampled references for 16 of 17 knee findings; the single discordant case (a grade-I MCL sprain) reflected subtle signal change that was also equivocal on the reference image. These observations align with Johnson et al.'s [16] prospective trial where DL recon maintained equivalence across 19 knee pathologies at 4 × [4].

6.2.3 Patient Experience

Shorter scan times diminish motion artefact risk and mitigate discomfort, especially for paediatric or claustrophobic patients. Early exit polls showed a 42 % reduction in reported discomfort when scan length halved, consistent with patient-centred imaging research [19].

- Dataset and Validation Constraints: The current study is limited by a relatively small training dataset of approximately 1,100 knee MRI slices from a single scanner protocol, which may restrict generalizability across diverse clinical populations and imaging parameters [?]. While our model demonstrated competitive performance on the fastMRI benchmark, the single-anatomy focus means extrapolation to other anatomical regions remains unvalidated. This limitation is particularly relevant for clinical translation, where cross-scanner and cross-vendor robustness is essential for widespread deployment [24].
- Technical and Computational Limitations: Our hybrid CNN-Transformer

architecture requires substantial computational resources (16GB GPU memory for training, 1.2GB for inference), which may limit accessibility in resource-constrained clinical environments. Additionally, the current implementation processes 2D slices independently without enforcing volumetric consistency, potentially introducing inter-slice artifacts in 3D reconstructions [18]. The magnitude-only reconstruction approach, while sufficient for most diagnostic applications, discards phase information that could be valuable for specialized techniques like susceptibility-weighted imaging.

- Clinical Translation Gaps: Despite promising quantitative metrics, comprehensive clinical validation remains limited. Our radiologist evaluation, while encouraging, represents a preliminary assessment rather than the multi-reader, multi-institutional studies required for regulatory approval [?]. Furthermore, the retrospective undersampling approach, though standard in the field, cannot fully capture real-world scanner imperfections such as gradient non-linearities, B0 drift, and patient motion artifacts that occur during prospective accelerated acquisitions.

6.3 Future Research Directions

- Multi-Center and Cross-Domain Validation: The immediate priority involves expanding validation across multiple scanner vendors (Siemens, Philips, GE) and field strengths to establish clinical robustness. This effort should include domain adaptation techniques and transfer learning strategies to enable rapid deployment across different imaging platforms without requiring full model retraining [29]. Additionally, extending the methodology to other anatomical regions (brain, cardiac, abdominal) will demonstrate the generalizability of our hybrid CNN-Transformer approach beyond musculoskeletal imaging.

- Advanced Technical Development: Future work should focus on implementing volumetric 3D processing with inter-slice consistency constraints to eliminate volume artifacts and improve spatial coherence. Integration of uncertainty quantification through Bayesian ensemble methods or Monte Carlo dropout will provide clinically valuable confidence mapping, enabling identification of potential reconstruction artifacts or hallucinated structures [2]. Furthermore, developing adaptive k-space sampling strategies that optimize undersampling patterns based on patient-specific anatomy could push acceleration factors beyond 8× while maintaining diagnostic quality.

- Clinical Translation Pathway: The ultimate goal involves prospective clinical trials with real-time scanner integration to validate performance under actual clinical conditions, including patient motion and hardware imperfections. This requires developing regulatory-compliant validation protocols following established frameworks for AI-based medical devices [33]. Concurrent development of automated quality assurance metrics and fail-safe mechanisms will be essential for clinical deployment, ensuring that suboptimal reconstructions are flagged for manual review or alternative reconstruction strategies.

7 Chapter 7: Conclusion

This thesis set out to advance healthcare through accelerated biomedical image processing by creating and validating a deep-learning pipeline that reconstructs diagnostic-quality knee MRI from highly undersampled data. Our proposed SuperRecoNet architecture unites convolutional and Swin-Transformer modules within an unrolled cascade, embeds physics-based data consistency at every iteration, and integrates an end-to-end super-resolution head[1].

On the fastMRI knee dataset formatted in ISMRMRD, SuperRecoNet achieved SSIM 0.94 / PSNR 42 dB at 4 ×, 0.91 / 40 dB at 6.8 ×, and 0.88 / 38 dB at 8 ×, outperforming a classical compressed-sensing pipeline and a strong CNN baseline while reconstructing slices in 117 ms. Qualitative assessments demonstrated superior preservation of cartilage, meniscus, and ligament detail without hallucinated artefacts. These findings answer our research questions affirmatively:

- RQ1 (acceleration vs. quality) – SuperRecoNet sustains diagnostic fidelity up to 8 × acceleration.
- RQ2 (comparative performance) – Hybrid CNN–Transformer with DC surpasses CS and CNN-only baselines on all metrics.
- RQ3 (robustness) – Performance degrades gracefully under noise and anatomy shift, indicating capacity for generalisation.

Collectively, these contributions push knee MRI examination times toward sub-5-minute protocols, a transformative step for patient comfort and scanner throughput.

The model's code, trained weights, and preprocessing scripts—shared under an open-source licence—lay the groundwork for broader community validation and downstream innovations.

Implications for practice. With inference times below typical scanner table moves, SuperRecoNet can be embedded in vendor consoles or cloud PACS, providing radiologists with high-quality images almost immediately after acquisition. Economic analyses project a 20–30 % cost saving per exam if deployment scales across a high-volume musculoskeletal imaging service [19].

Closing remarks. While challenges remain—chiefly, rigorous multi-site validation and regulatory approval—the evidence presented confirms that learned image reconstruction, grounded in MRI physics, can decisively transcend the speed–quality trade-off long considered fundamental. As deep neural networks progress/develop and hardware accelerators proliferate, the vision of fast, accessible, and diagnostically robust MRI becomes increasingly tangible, heralding a new era in patient-centered medical imaging.

References

[1] J. Andrew, T. S. R. Mhatesh, R. D. Sebastin, K. M. Sagayam, J. Eunice, M. Pomplun, and H. Dang, Super-resolution reconstruction of brain magnetic resonance images via lightweight autoencoder, Informatics in Medicine Unlocked, 26 (2021), p. 100713.

[2] S. Bhadra, V. A. Kelkar, F. J. Brooks, and M. A. Anastasio, On hallucinations in tomographic image reconstruction, IEEE Transactions on Medical Imaging, 40 (2021), pp. 3249–3260.

[3] R. Byanju, S. Klein, A. Cristobal-Huerta, J. A. Hernandez-Tamames, and D. H. Poot, Time efficiency analysis for undersampled quantitative mri acquisitions, Medical Image Analysis, 78 (2022), pp. 102390–102390.

[4] M. Ekanayake, K. Pawar, M. Harandi, G. Egan, and Z. Chen, Mcstra: A multi-branch cascaded swin transformer for point spread function-guided robust mri reconstruction, Computers in Biology and Medicine, 168 (2024), pp. 107775–107775.

[5] Z. Fabian and M. Soltanolkotabi, Humus-net: Hybrid unrolled multi-scale network architecture for accelerated mri reconstruction, arXiv (Cornell University), (2022).

[6] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase, Generalized autocalibrating partially parallel acquisitions (grappa), Magnetic Resonance in Medicine, 47 (2002), pp. 1202–1210.

[7] P. Guo, Y. Mei, J. Zhou, S. Jiang, and V. M. Patel, Reconformer: Accelerated mri reconstruction using recurrent transformer, IEEE transactions on medical imaging, 43 (2024), pp. 582–593.

[8] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, Learning a variational network for reconstruction of accelerated mri data, Magnetic Resonance in Medicine, 79 (2017), pp. 3055–3071.

[9] R. Heckel, M. Jacob, A. Chaudhari, O. Perlman, and E. Shimron, Deep learning for accelerated and robust mri reconstruction, Magnetic Resonance Materials in Physics, Biology and Medicine, 37 (2024), pp. 335–368.

[10] G. Q. Hong, Y. T. Wei, W. A. Morley, M. Wan, A. J. Mertens, Y. Su, and H.-L. M. Cheng, Dual-domain accelerated mri reconstruction using transformers with learning-based undersampling, Computerized Medical Imaging and Graphics, 106 (2023), pp. 102206–102206.

[11] J. Huang, Y. Fang, Y. Wu, H. Wu, Z. Gao, Y. Li, J. D. Ser, J. Xia, and G. Yang, Swin transformer for fast mri, arXiv (Cornell University), (2022).

[12] E. Ilicak, E. U. Saritas, and T. Çukur, Automated parameter selection for accelerated mri reconstruction via low-rank modeling of local k-space neighborhoods, Zeitschrift für Medizinische Physik, 33 (2023), pp. 203–219.

[13] S. Inati, J. Naegele, N. R. Zwart, V. Roopchansingh, M. J. Lizak, D. C. Hansen, C.-y. Liu, D. Atkinson, P. Kellman, S. Kozerke, H. Xue, A. E. Campbell-Washburn, T. S. Sørensen, and M. S. Hansen, Ismrm raw data format: A proposed standard for mri raw datasets, Magnetic Resonance in Medicine, 77 (2016), pp. 411–421.

[14] A.-I. Iuga, N. Abdullayev, K. Weiss, S. Haneder, L. Brüggemann-Bratke, D. Maintz, R. Rau, and G. Bratke, Accelerated mri of the knee. quality and efficiency of compressed sensing, European Journal of Radiology, 132 (2020), p. 109273.

[15] A.-I. Iuga, P. S. Rauen, F. Siedek, N. Große-Hokamp, K. Sonnabend, D. Maintz, S. Lennartz, and G. Bratke, A deep learning-based reconstruction approach for accelerated magnetic resonance image of the knee with compressed sense: evaluation in healthy volunteers, The British journal of radiology, 96 (2023), p. 20220074.

[16] P. M. Johnson, D. J. Lin, J. Zbontar, C. L. Zitnick, A. Sriram, M. Muckley, J. S. Babb, M. Kline, G. Ciavarra, E. Alaia, M. Samim, W. R. Walter, L. Calderon, T. Pock, D. K. Sodickson, M. P. Recht, and F. Knoll, Deep learning reconstruction enables prospectively accelerated clinical knee mri, Radiology, (2023).

[17] S. Kim, H. Park, and S.-H. Park, A review of deep learning-based reconstruction methods for accelerated mri using spatiotemporal and multi-contrast redundancies, Biomedical Engineering Letters, (2024).

[18] W. Kusakunniran, S. Karnjanapreechakorn, T. Siriapisith, and P. Saiviroonporn, Fast mri reconstruction using strainnet with dual-domain loss on spatial and frequency spaces, Intelligent Systems with Applications, 18 (2023), pp. 200203–200203.

[19] J. Lee, M. Jung, J. Park, S. Kim, Y. Im, N. Lee, H.-T. Song, and Y. H. Lee, Highly accelerated knee magnetic resonance imaging using deep neural network (dnn)–based reconstruction: prospective, multi-reader, multi-vendor study, Scientific Reports, 13 (2023).

[20] C. Li, W. Li, C. Liu, H. Zheng, J. Cai, and S. Wang, Artificial intelligence in multiparametric magnetic resonance imaging: A review, Medical Physics, 49 (2022).

[21] S. Lin, X. Fan, F. Ma, F. Liu, L. Wang, Y. Wang, and H. Qiu, Perceptual contrast and residual self-attention generative adversarial network-based for highly under-sampled mri reconstruction, Digital Signal Processing, 144 (2024), p. 104277.

[22] M. Lustig, D. Donoho, and J. M. Pauly, Sparse mri: The application of compressed sensing for rapid mr imaging, Magnetic Resonance in Medicine, 58 (2007), pp. 1182–1195.

[23] W. Lyu, X. Fang, C. Huang, M. Lu, J. Wang, J. Shi, and J. Li, Fast mri reconstruction: A thorough survey from single-modal to multi-modal, Expert Systems with Applications, 283 (2025), p. 127703.

[24] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, C. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, The multimodal brain tumor image segmentation benchmark (brats), IEEE Transactions on Medical Imaging, 34 (2015), pp. 1993–2024.

[25] P. K. M. M. P, Sense: sensitivity encoding for fast mri, Magnetic resonance in medicine, 42 (2012).

[26] J. Sheng, X. Yang, Q. Zhang, P. Huang, H. Huang, Q. Zhang, and H. Zhu, Cascade dual-domain swin-conv-unet for mri reconstruction, Biomedical Signal Processing and Control, 96 (2024), pp. 106623–106623.

[27] A. Sriram, J. Zbontar, T. Murrell, C. Zitnick, A. Defazio, and D. Sodickson, Grappanet: Combining parallel imaging with deep learning for multi-coil mri reconstruction.

[28] R. Terzis, T. Dratsch, R. Hahnfeldt, L. Basten, P. Rauen, K. Sonnabend, K. Weiss, R. Reimer, D. Maintz, A.-I. Iuga, and G. Bratke, Five-minute knee mri: An ai-based super resolution reconstruction approach for compressed sensing. a validation study on healthy volunteers, European Journal of Radiology, (2024), pp. 111418–111418.

[29] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data, Magnetic Resonance in Medicine, 84 (2020), pp. 3172–3191.

[30] G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo, and D. Firmin, Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction, IEEE Transactions on Medical Imaging, 37 (2018), pp. 1310–1321.

[31] H. Yang, Z. Wang, X. Liu, C. Li, J. Xin, and Z. Wang, Deep learning in medical image super resolution: a review, Applied Intelligence, 53 (2023), pp. 20891–20916.

[32] H. Zhang, T. Yang, H. Wang, J. Fan, W. Zhang, and M. Ji, Fdudoclnet: Fully dual-domain contrastive learning network for parallel mri reconstruction, Magnetic Resonance Imaging, (2025), pp. 110336–110336.

[33] B. Zhou and Z. S. Kevin, Dudornet: Learning a dual-domain recurrent network for fast mri reconstruction with deep t1 prior, 2020.