Universiteit Leiden
The Netherlands

Deep learning for online adaptive radiotherapy

Elmahdy, M.S.E.

Citation

Elmahdy, M. S. E. (2022, March 15). *Deep learning for online adaptive radiotherapy*. Retrieved from https://hdl.handle.net/1887/3278960

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3278960

Note: To cite this publication please use the final published version (if applicable).

6

An Adaptive Intelligence Algorithm for Undersampled Knee MRI Reconstruction

This chapter was adapted from:

N Pezzotti, S Yousefi, M Elmahdy, C Schülke, M Doneva, T Nielsen, S Kastryulin, B Lelieveldt, M van Osch, E de Weerdt, and M Staring. **An Adaptive Intelligence Algorithm for Undersampled Knee MRI Reconstruction**, *IEEE Access*, Pages 204825-204838, 2020.

Abstract

Adaptive intelligence aims at empowering machine learning techniques with the additional use of domain knowledge. In this work, we present the application of adaptive intelligence to accelerate MR acquisition. Starting from undersampled k-space data, an iterative learning-based reconstruction scheme inspired by compressed sensing theory is used to reconstruct the images. We developed a novel deep neural network to refine and correct prior reconstruction assumptions given the training data. The network was trained and tested on a knee MRI dataset from the 2019 fastMRI challenge organized by Facebook AI Research and NYU Langone Health. All submissions to the challenge were initially ranked based on similarity with a known groundtruth, after which the top 4 submissions were evaluated radiologically. Our method was evaluated by the fastMRI organizers on an independent challenge dataset. It ranked #1, shared #1, and #3 on respectively the 8x accelerated multi-coil, the 4x multi-coil, and the 4x single-coil tracks. This demonstrates the superior performance and wide applicability of the method.

6.1 Introduction

Magnetic Resonance Imaging (MRI) is a widely applied non-invasive imaging modality, with excellent soft tissue contrast and high spatial resolution. Unlike Computed Tomography (CT) scanning, MRI does not expose patients to any ionizing radiation, making it a compelling alternative. MR images are essential for clinical assessment of soft tissue as well as functional and structural measurements, which leads to early detection and diagnosis of many diseases. However, MRI is relatively slow compared to other imaging modalities. The total examination time can vary from 15 minutes for knee imaging to an hour or more for cardiac imaging. Remaining still for this long in a confined space is challenging for any patient, being especially difficult for children, elderly and patients under pain. Motion artifacts are not only difficult to correct, which may require a complete re-scan [134]. Furthermore, the acquisition time affects the temporal resolution and subsequently limits the potential of MRI for dynamic imaging, where high temporal resolution and robustness against motion are critical for diagnosis. Moreover, the relatively long scan times lead to high costs that limit the availability of MRI scanners [135]. Therefore, fast acquisition and reconstruction are crucial to improve the performance of current MR scanners, which led in recent years to the development of techniques such as parallel reception, compressed sensing and multi-band accelerations. However, there is still a need for further scan acceleration.

The long acquisition time is intrinsic to the scanner and physics properties of MRI. For the majority of scans performed in clinical practice, this acquisition is done through consecutive reading-out of single lines in k-space. These readouts are constrained by physical limitations of the hardware, the contrast generating principle, and human physiology. The scanning time could be shortened by reducing the number of acquired lines in k-space, i.e. by undersampling the 2D or 3D k-space. However, this could violate the Nyquist criterion, resulting in aliasing and blurriness in the reconstructed images, rendering them unqualified for clinical purpose. Compressed Sensing (CS) and Parallel Imaging are the most common solutions for acceleration by undersampling, while maintaining image quality. Compressed Sensing, the focus of this paper, introduced by Donoho [28], Lustig [29] and Candes [30], leverages the fact that MR images can be compressed in some domain, restoring the missing k-space data through an iterative reconstruction algorithm [31]. Parallel Imaging uses multiple receive coils that provide an additional signal encoding mechanism, allowing to reduce the number of necessary k-space lines to reconstruct an image, thus partially parallelizing the data acquisition [136].

When CS is used to accelerate MR acquisitions, the k-spaces is sampled pseudo-randomly and the image is subsequently reconstructed by promoting a sparse solution. In the optimal setting, the reconstructed image will be identical to the Fourier transform of the full k-space and have a limited number of large coefficients when transformed to the sparse domain. Equation (6.1) shows the optimization function that describes the CS algorithm:

$$\min_{\mathbf{x}}\{\parallel M\mathscr{F}\mathbf{x}-M\mathbf{y}\parallel_2^2+\lambda\parallel\Psi\mathbf{x}\parallel_1\}, \tag{6.1}$$

where $\mathbf{x}$ is the reconstructed image, $\mathbf{y}$ is the fully measured k-space data, $\mathscr{F}$ is the Fourier transform, M (mask) is the undersampling operation, $\Psi\mathbf{x}$ represents the sparsity transform coefficients, and λ is the regularization parameter. The ℓ_1 norm is used to enforce sparsity of the solution in a domain specified by the transformation Ψ. The ℓ_2 norm is used as a similarity measure between the measured k-space data $M\mathbf{y}$ and the reconstructed k-space $M\mathscr{F}\mathbf{x}$, called the "data consistency" term. Note that, in case of multi-coil acquisitions, the data consistency term is given by:

$$\sum_q\parallel M\mathscr{F}\left(\mathbf{S}_q\cdot\mathbf{x}\right)-M\mathbf{y}_q\parallel_2^2, \tag{6.2}$$

where q denotes the coil element and $\mathbf{S_q}$ the corresponding coil sensitivity map. The coil sensitivity maps $\mathbf{S}$ are computed using the fully centered region of k-space. A low-passed version of the coil images $\mathbf{x}_q^{\text{lpf}}$ is obtained by cropping the available region of k-space. The sensitivity map $\mathbf{S}_q$, for the individual coil element is computed as follows:

$$\mathbf{S}_q=\frac{\mathbf{x}_q^{\text{lpf}}}{\sqrt{\sum_j\left(\mathbf{x}_j^{\text{lpf}}\right)^2}} \tag{6.3}$$

To simplify notation, without loss of generality, the single-coil data consistency term will be used throughout this paper.

Recently, deep learning has shown promising results for speeding up MR acquisition by adopting Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN). In contrast to iteratively solving optimization problems, deep learning offers a solution for reconstructing highly-accelerated scans by adopting learnable reconstruction schemes.

The literature of deep learning-based reconstruction algorithms can be divided into two categories [137]. First, data-driven approaches, where a neural network is trained to find the optimal transformation from the zero-filled k-space to the desired reconstruction. Here, the network is completely dependent on the underlying training dataset without any task-specific prior knowledge on the domain; following are selected exemplar algorithms of this approach. Quan *et al.* [138] developed a GAN network for MR reconstruction starting from undersampled data. Their network consists of two consecutive networks, one for reconstruction and one for refining the results. They used a cyclic data consistency term alongside the WGAN loss.

Mardani *et al.* [139] developed a GAN network for CS. The proposed network corrects aliasing artifacts of MR images. Guo et al. [140] proposed a WGAN with recurrent context-awareness to reconstruct MRI images from highly undersampled k-space data. Schlemper et al. propose a cascaded CNN-based compressive sensing (CS) technique for the reconstruction of diffusion tensor cardiac MRI [141]. Yang et al. proposed a conditional GAN-based architecture for de-aliasing and fast CS-MRI [142, 143]. Putzky *et al.* [144] treated the MR reconstruction problem as an inverse problem. They applied the previously introduced invertible Recurrent Inference Machine (i-RIM) model [145], which iteratively updates its current state based on the output of the forward model. The model was trained and evaluated on the single- and multi-coil data at 4x and 8x accelerations from the fastMRI challenge (see Section 6.2 for more details). AUTOMAP [146] reports good reconstruction results with an architecture that learns to directly transform k-space into image data. Lee *et al.* [147] introduced two separate deep residual networks for magnitude and phase. The proposed networks successfully reconstructed images even when obtained with high undersampling factors.

Second, hybrid approaches are presented in the literature. This class of algorithms builds on top of existing reconstruction solutions and integrate learning-based approaches to substitute part of the original computations, often by adopting an unrolled implementation of an iterative algorithm [32]. A notable example is the Variational network presented by Hammernik et al. [33] utilizing learned filters in an existing iterative optimization scheme, while Yang et al. presented the Deep ADMM-Net [34], which extends the Alternating Direction Method of Multipliers (ADMM) [148] approach by integrating learnable operators.

Aggarwal *et al.* [149] introduced a model based deep learning architecture named MoDL to solve the inverse problem, including MR reconstruction. The proposed model consists of a series of recursive linear CNN networks. These networks share weights for regularization and reduction in the number of parameters. The proposed network imitates the CS algorithm and for numerical optimization, the authors introduced a data consistency term using a conjugate gradient (CG) optimization scheme at every iteration. The model was trained on multi-coil brain MR slices from 4 patients and tested on one patient. Ramzi *et al.* [150] provided a reproducible benchmark of deep learning based reconstruction methods on the single-coil part of the fastMRI dataset [151]. The benchmark consists of a U-net [116], cascade net [152], KIKI-net [153], and PD-net [154]. Cascade net has been inspired by a dictionary learning approach [155]. This approach is composed of residual convolutional blocks applied in image space followed by data consistency layers. The data consistency layers enforce the k-space values be close to the original k-space measurements. KIKI-net is a cascaded network where a non-residual convolutional block has been added to perform k-space completion, while PD-net provides a learnable and unrolled version of the Primal Dual Hybrid Gradient optimization algorithm [156]. Seitzer et al. discussed the inadequacy of loss function for training a CS-MRI reconstruction CNN [157]. In that study they proposed a refinement method which incorporates both loss functions in a harmonious way to improve the training stability.

Recently, Zhang and Ghanem [158] developed a deep learning approach called ISTA-Net that mimics the conventional ISTA algorithm, but enriches it by replacing the sparsifying transform and the thresholding with learned operations. The resulting network does not implement a fully iterative algorithm, but it simulates it by adopting a fixed number of iterations, effectively enabling the implementation of a deep neural network that can be trained by the backpropagation algorithm. Inspired by the work of Zhang and Ghanem [158], in this paper we propose a deep-learning based solution, Adaptive-CS-Network, that mimics the ISTA algorithm, but introduces strong prior information, i.e., inductive biases, to better constrain the reconstruction problem. The main contributions of this work are: i) we propose a novel CNN network that integrates and enhances the conventional CS approach; ii) it integrates multiscale sparsification, inspired by wavelet transforms, but in a learnable manner; iii) we adopt domain-specific knowledge, such as data consistency, a prior on known phase behavior, and the location of the background: these computations cannot be easily learned by a CNN; iv) the proposed model exploits the correlation between neighbouring slices by adopting a 2.5D learning approach. In addition, we propose a hierarchical training strategy that leverages the available data. We conducted extensive experiments to investigate the performance of the network, and show that domain specific information is crucial for reconstructing high-quality MR images. The proposed network showed superior performance by winning one, and co-winning a second track out of the three tracks of the fastMRI challenge [151].

6.2 FastMRI challenge

The fastMRI challenge is a challenge organized by Facebook AI Research and NYU Langone Health [151]. The aim of the challenge is to advance and encourage AI-based research in MR reconstruction in order to allow acceleration of the acquisition and, subsequently, to reduce the examination time. The challenge is divided in three tracks: 4x single-coil, 4x multi-coil, and 8x multi-coil accelerations. Eight teams participated in the multi-coil track and 17 teams in the single-coil track [159].

6.2.1 Dataset

The challenge organizers released a large-scale dataset of raw MR data of the knee [160]. The data was acquired with a 2D protocol in the coronal direction with a 15 channel knee coil array using Siemens MR machines at two different field strengths: 1.5T and 3T [151]. The data was acquired using two pulse sequences: a *proton* *density weighting* with (PDFS) and without (PD) fat suppression. The data is divided approximately equally between these pulse sequences. The pixel size is 0.5 mm × 0.5 mm with a slice thickness of 3 mm.

The dataset is divided in 4 categories: training (973 volumes, 34,742 slices), validation (199 volumes, 7,135 slices), test (118 volumes, 4,092 slices), and challenge (104 volumes, 3,810 slices). These numbers are the same for multi-coil and single-coil data, with the exception of the test and challenge categories, where single-coil data has respectively 10 and 12 volumes less than the multi-coil data. The training, validation and test sets were publicly available since late November, 2018, while the challenge set was available since September 2019. The full k-space was available for all the datasets except for the test and challenge sets. Training and validation sets were considered for training and optimizing our model, while the test set was used for evaluating model performance on a public leaderboard. The final model was evaluated by the organizers on the independent challenge set.

The k-space data provided in the challenge were undersampled using a Cartesian mask, where k-space lines are set to zero in the phase encoding direction. The sampling density is dependent on the acceleration rate (4x or 8x), where the sampled lines are randomly selected. All masks, however, are fully sampled in the central area of k-space which corresponds to the low frequencies of the image. For the 4x accelerated scans, this percentage is 8% while it is 4% for 8x acceleration. Besides making the reconstruction problem easier to solve, such lines allow for obtaining a low-pass filtered version of the image that is used to compute the coil sensitivity maps $\mathbf{S}_q$ as presented in Equation (6.3) using a root sum of square approach [151].

6.2.2 Quantitative evaluation

In order to measure the accuracy of the reconstructed volumes $\mathbf{r}$ compared to the target volumes $\mathbf{t}$, the following metrics were considered:

6.2.2.1 Normalized mean square error (NMSE)

measures the square of the Euclidean norm between a pair of images:

$$\mathrm{NMSE} = \frac{||\mathbf{r}-\mathbf{t}||_2^2}{||\mathbf{t}||_2^2} \tag{6.4}$$

6.2.2.2 Peak signal-to-noise ratio (PSNR)

the ratio between the maximum intensity and the underlying distortion noise:

$$\mathrm{PSNR} = 10\log_{10}\frac{\max(\mathbf{t})^2}{\frac{1}{N}||\mathbf{r}-\mathbf{t}||_2^2} \tag{6.5}$$

6.2.2.3 Structural similarity index metric (SSIM)

measures image similarity using human perception aspects [161]. SSIM is calculated by measuring three image distortions including luminance $l(\cdot)$, contrast $c(\cdot)$ and structure $s(\cdot)$:

$$\mathrm{SSIM} = l(\mathbf{r},\mathbf{t})^{\alpha} c(\mathbf{r},\mathbf{t})^{\beta} s(\mathbf{r},\mathbf{t})^{\gamma}, \tag{6.6}$$

where α, β, γ are the distortion weights, here chosen as 1. In this study, similar to the fastMRI challenge, the SSIM score is computed on the magnitude version of the 2D MR scans, leading to grayscale images.

6.2.3 Radiological evaluation on the challenge dataset

We submitted the reconstructions on the challenge dataset via an online form, which were then evaluated independently by the fastMRI organizers, described in detail by Knoll *et al.* [159]. All submissions were ranked by the SSIM metric, after which only the 4 highest ranking submissions were evaluated by a panel of 7 radiologists. The panel was asked to evaluate the reconstructions on a scale from 1 to 5 on four different categories, where 1 is the best and 5 is the worst. The 4 categories were the rating of artifacts, reconstruction sharpness, perceived contrast-to-noise ratio and diagnostic confidence. The radiological scores were subsequently averaged and translated to a final ranking.

6.3 Methods

In this section we present the background of our solution, first by introducing the Iterative Shrinkage-Thresholding Algorithm (ISTA) [162] and, second, by introducing its deep learning-based variant, ISTA-Net [158]. Then, we present our solution, the Adaptive-CS-Network, that builds on top of the ISTA-Net framework by introducing several improvements, including strong inductive biases derived from domain knowledge on the reconstruction problem.

6.3.1 ISTA background

ISTA is an optimization algorithm to solve (6.1) in an iterative fashion, starting from the reconstruction $\mathbf{x}_0$, which is often obtained by reconstructing the zero-filled undersampled k-space. The initial estimate is refined using the following update rules:

$$\mathbf{r}_{i+1} = \mathbf{x}_i - \rho \mathscr{F}^T (M\mathscr{F}\mathbf{x}_i - M\mathbf{y}), \tag{6.7}$$

$$\mathbf{x}_{i+1} = \arg\min_{\mathbf{x}} \frac{1}{2} \parallel \mathbf{x} - \mathbf{r}_{i+1} \parallel_2^2 + \lambda \parallel \Psi \mathbf{x} \parallel_1, \tag{6.8}$$

where F^T denotes inverse Fourier transform, $\mathbf{r}_{i+1}$ is an update of the estimate $\mathbf{x}_i$, where the error in the measured data $M\mathbf{y}$ is corrected by a step ρ. Equation (6.8) is a special case of the proximal mapping, with a regularization weight λ, and a crucial

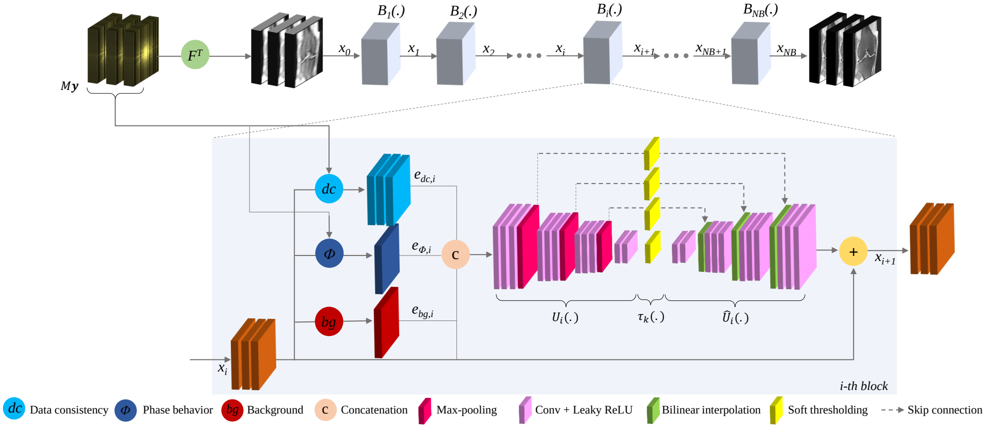

Figure 6.1: Proposed adaptive Adaptive-CS-Net architecture. The input and output of the network are stacks of three consequent knee MR images.

step for optimization algorithms such as ISTA, ADMM [148] and AMP [163]. When Ψ is a wavelet transform **W**, it can be proven that

$$\mathbf{x}_{i+1} = \mathbf{W}^{-1} soft(\mathbf{W}\mathbf{r}_{i+1}, \lambda), \tag{6.9}$$

where $soft$ is the soft-tresholding operator defined as $soft(\mathbf{u}, \lambda) = \max(|\mathbf{u}| - \lambda, 0) \cdot \frac{\mathbf{u}}{|\mathbf{u}|}$. In general, solving (6.8) is not straightforward for non-linear operators Ψ, limiting the applicability of the ISTA framework to simple transforms. Another problem of this family of algorithms, is the difficulty of tuning the hyperparameters λ and ρ in addition to its slow convergence, hence requiring a lot of iterations to achieve the optimal solution of (6.1).

6.3.2 ISTA-Net

Recently, Zhang and Ghanem introduced a deep-learning approach to overcome the limitations of the ISTA framework for image-to-image reconstruction. Their solution, called ISTA-Net [158], replaces the handcrafted transform Ψ with a learned operator $\mathscr{S}(\cdot)$, which consists of a 2D learnable convolution followed by a rectified linear unit (ReLU) and a second convolution. By replacing Ψ with $\mathscr{S}(\cdot)$ in (6.8), we can rewrite the update rule as

$$\mathbf{x}_{i+1} = \arg\min_{\mathbf{x}} \frac{1}{2} \parallel \mathbf{x} - \mathbf{r}_{i+1} \parallel_2^2 + \lambda \parallel \mathscr{S}(\mathbf{x}) \parallel_1, \tag{6.10}$$

and, by defining $\hat{\mathscr{S}}$ as the inverse of $\mathscr{S}$, i.e., $\hat{\mathscr{S}} \circ \mathscr{S} = I$, Zhang and Ghanem propose to update (6.9) as follows:

$$\mathbf{x}_{i+1} = \hat{\mathscr{S}}(soft(\mathscr{S}(\mathbf{r}_{i+1}), \lambda)), \tag{6.11}$$

where $\hat{\mathscr{S}}$ has a similar architecture as $\mathscr{S}$.

The model is trained end-to-end, where the iterations of the ISTA algorithm are "unrolled", i.e., a number b of identical reconstruction blocks are created. Note that in the ISTA-Net approach, the learnable parameters are shared among all the blocks in the unrolled network, unlike our solution. The training loss is defined as a combination of the reconstruction and discrepancy loss:

$$\mathscr{L} = \mathscr{L}_{reconstruction} + \sigma \mathscr{L}_{discrepancy} \tag{6.12}$$

$$\mathscr{L}_{reconstruction} = \| \mathbf{x}_b - \mathscr{F}^T \mathbf{y} \|_2^2 \tag{6.13}$$

$$\mathscr{L}_{discrepancy} = \frac{1}{b} \sum_{i=1}^{b} \| \hat{\mathscr{S}}(\mathscr{S}(\mathbf{x}_i)) - \mathbf{x}_i \|_2^2 \tag{6.14}$$

The reconstruction loss encodes the need for the final reconstruction, defined as $\mathbf{x}_b$, to be as close as possible in the least squares sense to the ground-truth image, i.e., $F^T\mathbf{y}$. The discrepancy loss stimulates that $\hat{\mathscr{S}} \circ \mathscr{S} = I$. The σ parameter allows to control the weight given to the discrepancy loss, and it is chosen to be arbitrarily small, e.g., $\sigma = 0.01$. An extension, called ISTA-Net$^+$ is also presented by the authors, where residual computations are adopted.

6.3.3 Adaptive-CS-Network

Starting from the network developed by Zhang and Ghanem, we developed the Adaptive-CS-Network approach. Our solution builds on top of the ISTA-Net solution based on three key innovations, here ordered by importance to the final network performance: i) the use of multi-scale and ii) multi-slice computations, together with iii) the introduction of soft MRI priors. We present them independently, building towards the update rule of the Adaptive-CS-Network model as presented in (6.16). Fig. 6.1 illustrates the proposed network.

First, many non-learned CS algorithms make use of **multi-scale** transforms to sparsify the signal. An example is given in (6.9), where **W** is a wavelet transform; a decomposition of the signal into a set of basis functions at different scales. We include this inductive bias in our design, and adopt a multi-scale transform $\mathscr{U}$, and its inverse $\hat{\mathscr{U}}$. As an additional design choice, we decide to sparsify and learn only the residual, therefore our update rule is written as follows:

$$\mathbf{x}_{i+1} = \hat{\mathscr{U}}(soft(\mathscr{U}(\mathbf{r}_{i+1}), \lambda_{s,f_s})) + \mathbf{r}_{i+1}, \tag{6.15}$$

where $\mathscr{U}$ comprises of 2D convolutions and non-linearities in the form of Leaky-ReLU to counteract the problem of dying neurons. To generate a multiscale representation, a max-pooling layer is used and the resulting features are then processed again by convolutional blocks and non-linearities. The exact design of $\mathscr{U}$ is presented in Fig. 6.1. The feature maps produced at the different scales are then thresholded using the soft-max function. Differently from ISTA-Net$^+$, we learn a lambda parameter and feature channel f_s for each scale s. This approach gives the network the flexibility of tuning the thresholds independently, hence reducing the complexity of the transforms learned by the convolutional operators. Finally, the filtered channels are transformed back into the image domain by the inverse $\hat{\mathscr{U}}$, consisting of interpolation, 2D convolutions and Leaky-ReLU operators. Note that, contrary to the latest literature in deep learning networks, we decided not to adopt strided convolutions for sub- and up-sampling, which would increase the risk of creating checkerboard artifacts [164]; instead we took the more conservative approach of adopting pooling and interpolation layers for achieving better image quality. Overall, the computation represented by $\hat{\mathscr{U}}(soft(\mathscr{U}(\mathbf{r}_{i+1}), \lambda_{s,f_s}))$ is implemented with a UNet-like architecture [116], where the feature maps before the skip connections are filtered according to the parameter λ_{s,f_s}.

Second, it is important to note that the slice thickness of the dataset is much higher than the in-plane resolution. This indicates that inter-slice correlations are less useful for finer scales, and potentially damaging as they will become a confounder for the network. However, such information becomes beneficial at coarser scales, e.g., to facilitate the delineation of the bone in several slices. Since our transform $\mathscr{U}$ is multi-scale by nature, we found it beneficial to inject **neighboring slices** into the model, while leaving it to the network to identify at which scale the information will be used. To reduce the memory footprint of the model, we adopted a 2.5D convolution approach by concatenating neighbouring slices into the input tensor along the channel dimension, enabling to “reinvest” the saved GPU memory as compared to a truly 3D convolution approach, into more unrolled iterations. More details on the number of slices used and the definition of the loss function are given in Section 6.4.3.

Finally, we adopted a hybrid- or nudge- approach to incorporate additional **prior knowledge** into the reconstruction algorithm. We therefore computed additional information derived from the current estimate $\mathbf{x}_i$ together with k-space $M\mathbf{y}$. These soft priors, which are presented in the next section, capture some properties of an MR image that cannot be easily learned by a deep neural network due to the limited size of the receptive field. The priors come in the form of images, and are provided as extra input channel to the transform $\mathscr{U}$. In this way, they are integrated in the computations performed by $\mathscr{U}$ whenever this is beneficial for the optimization of the loss function.

6.3.4 Final design

The overall update for a block B_{i+1} in the Adaptive-CS-Network model is defined as follows:

$$\begin{aligned} \mathbf{x}_{i+1} = B_{i+1}(\mathbf{x}_i) = & \\ & \mathbf{x}_i + \hat{\mathscr{U}}_i\left(soft\left(\mathscr{U}_i\left(\mathbf{x}_i, \mathbf{e}_{\mathrm{dc},i}, \mathbf{e}_{\phi,i}, \mathbf{e}_{\mathrm{bg},i}\right), \lambda_{s,f_s}\right)\right). \end{aligned} \tag{6.16}$$

Each block in the network learns different transforms $\mathscr{U}_i$ and $\hat{\mathscr{U}}_i$, enabling each block to focus on different properties of the image and effectively increasing the network capacity. Note that $\mathscr{U}_i$ and $\hat{\mathscr{U}}_i$ are different for every reconstruction block i.

In our final design, the transform $\mathscr{U}_i$ does not receive the data consistent image $\mathbf{r}_i$, as defined in (6.7), but rather the current estimate $\mathbf{x}_i$ together with the data consistency prior $\mathbf{e}_{\mathrm{dc},i}$ computed as follows:

$$\mathbf{e}_{\mathrm{dc},i} = \mathscr{F}^T(M\mathscr{F}\mathbf{x}_i - M\mathbf{y}). \tag{6.17}$$

This "soft data-consistency" update allows the network to evaluate the reliability of the acquired data and potentially compensate errors in the coil combination defined by $\mathscr{F}$ in (6.1).

The second prior we provided to the network, $\mathbf{e}_{\phi,i}$, represents the known phase response for spin-echo MR sequences. Theoretically, spin-echo sequences have zero phase everywhere in the image. In practice, however, slowly varying phase will occur, i.e. nonzero phase only in the low frequencies, due to hardware and acquisition imperfections. Taking this into account, it is noted that the final reconstructed image should be a real valued image after removal of the slowly varying phase. This information is captured in the following prior:

$$\mathbf{e}_{\phi,i} = \left\{\mathbf{x}_i \cdot \frac{\mathbf{x}^*_{i,\mathrm{lpf}}}{\|\mathbf{x}_{i,\mathrm{lpf}}\|_2}\right\}_{imag}, \tag{6.18}$$

where * denotes the complex conjugate, and lpf refers to low pass filtering. The low pass filter is chosen such that it corresponds to the center part of k-space which is fully sampled. By doing so, the low pass filtered image $\mathbf{x}_{i,\mathrm{lpf}}$ can be derived beforehand only once, hence $\mathbf{x}_{i,\mathrm{lpf}}$ is replaced by $\mathbf{x}_{0,\mathrm{lpf}}$.

Finally, we adopt a simple approach to estimate the location in $\mathbf{x}_i$ where the background is found, which is common in parallel imaging techniques. The following prior is applied:

$$\mathbf{e}_{\mathrm{bg},i} = \frac{\mathbf{x}_i}{\|\mathbf{x}_{i,\mathrm{lpf}}\|_2}. \tag{6.19}$$

This prior will penalize estimated signal content where $\|\mathbf{x}_{i,\mathrm{lpf}}\|$ is low, i.e., within the background. Again, $\mathbf{x}_{i,\mathrm{lpf}}$ is replaced by $\mathbf{x}_{0,\mathrm{lpf}}$. Because $\mathbf{x}_{0,\mathrm{lpf}}$ is based on the fully measured central part of k-space, the image is artefact free albeit at low spatial resolution, leading to a reliable background identification.

In Fig. 6.1 the design of the Adaptive-CS-Network is shown, including the multi-scale transforms, the multi-slice computation and the priors provided as input. Note how the spin-echo and background priors are computed only for the central slice, in order to save GPU memory.

6.3.5 Network training and implementation details

We implemented our models in PyTorch [165]. All the optimization experiments were performed on an NVIDIA V100 GPU with 16 GB RAM and the final network was trained on two NVIDIA V100 GPU with 16 GB RAM. In order to run as many experiments as possible given the challenge deadline, model optimization (see Section 6.4) was done with a relatively small model (≤ 10 blocks), which we trained for 20 epochs. All the optimization networks were trained and validated on the highest acceleration rate of the challenge, i.e. 8x and for single-coil data, except for the number of the blocks which was performed for both 4x and 8x, and for the priors which are more relevant for the multi-coil data. Since the ground truth for the test set was not available, all the quantitative comparisons were only done on the validation set.

For the challenge, we trained the final model using the training and validation datasets for 25 epochs and accelerations randomly selected from 2x to 10x. The residual connections designed on a per-iteration basis, facilitates the learning and prevents the degradation of the error gradient throughout the architecture. The model was subsequently fine-tuned on eight data sub-populations identified by the acceleration (4x and 8x), the protocol (PD and PDFS) and the scanner field strength (1.5T and 3T). Fine-tuning was then performed for 10 epochs on the sub-populations. This procedure was performed independently for the single- and multi-coil datasets, resulting in a total of 8 models. All models were trained using an exponentially decaying learning rate of 10^{-4}. The final models have 33M trainable parameters each; for the single-coil data this leads to an inference time of approximately 327 ms, while it takes approximately 518 ms to compute the reconstruction of a multi-coil dataset on an NVIDIA V100 GPU.

6.4 Experiments and results: Model optimization

In this section we present how we optimized the network configuration, on a smaller model with 10 reconstruction blocks, using the quantitative measures reported in Section 6.2.2 for validation. We performed experiments on the number of the blocks, the loss functions, the influence of using adjacent slices, the optimizer, and the soft priors. A repeated measure one-way ANOVA test was performed on the SSIM values using a significance level of $p = 0.05$. P-values are only stated for the comparisons between the best method and the other methods. In all the experiments a learning rate of 0.0001 was used.

6.4.1 Number of blocks

The proposed model consists of multiple blocks, related to the number of unrolled iterations of the ISTA scheme. Increasing the number of blocks leads to an increase in the number of parameters of the model, and subsequently training time and GPU memory usage as well as an increase in risk of overfitting. In this experiment we investigated the effect of the number of the blocks on the quality of reconstructed images. Tests were ran with the 2D network for 4x and 8x acceleration rates without neighboring slices, MSE as loss function, RMSprop as optimizer, and with the Unet-like

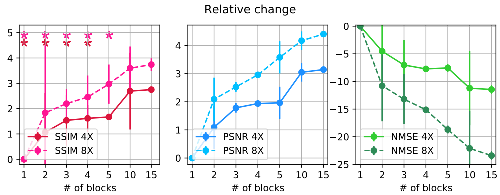

Figure 6.2: The effect of the number of blocks on performance, using the 4x and 8x single-coil validation data. The variance values are shown by the bars. The stars in the first plot show one-way ANOVA statistical significance.

Table 6.1: The effect of the loss function on performance, using the 8x single-coil validation data. Stars denote one-way ANOVA statistical significance.

Loss function	SSIM	NMSE	PSNR	p-value
	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	
MSE	0.657±0.149*	0.046±0.029	30.2±2.8	≪0.001
Perceptual loss	0.664±0.157*	0.061±0.044	29,2±3.2	≪0.001
Huber	0.664±0.148*	0.062±0.041	29.1±3.0	≪0.001
ℓ_1	0.664±0.148*	0.062±0.041	29.1±3.0	≪0.001
SSIM	0.662±0.145*	0.065±0.041	28.9±2.8	≪0.001
MSSIM [168]	0.671±0.143*	0.050±0.034	30.1±3.1	≪0.001
Eq. (6.21)	**0.673**±0.143	**0.048**±0.033	**30.3**±3.1	

architecture of 16 filter maps for each convolutional layer. Fig. 6.2 reports the relative changes to a single block of our quantitative metrics. Based on the experiments, increasing the number of the blocks will improve the performance of the network. Therefore, the final network was configured with the maximum number of blocks that could be fitted into GPU memory: 25 blocks. However, for the optimization experiments below only 10 blocks were employed to limit the duration of the training.

6.4.2 Loss functions

In this experiment we investigated the effect of a wide range of differentiable loss functions on the performance of our network. Here, we used the single slice reconstruction network with only 10 blocks, RMSprop as the optimizer, and 16 filter maps for each convolutional layer. The models were trained for 20 epochs to ensure convergence of the model. The evaluated loss functions included MSE, perceptual loss (PL) [166], ℓ_1, Huber [167] and multi-scale structural similarity index (MSSIM) [168]. The PL loss function was calculated using a pre-trained VGG-16 at layers relu1_2, relu2_2, and relu3_3.

MSSIM [168] builds upon SSIM (see Section 6.2.2.3) by incorporating structural similarity at multiple image resolutions, thereby supplying more flexibility compared to SSIM, and is defined as follows:

$$\mathrm{MSSIM} = \left[l_M(\mathbf{r^c},\mathbf{t^c})\right]^{\alpha_M} \prod_{i=1}^{M} \left[c_i(\mathbf{r^c},\mathbf{t^c})\right]^{\beta_i} \left[s_i(\mathbf{r^c},\mathbf{t^c})\right]^{\gamma_i}, \tag{6.20}$$

where $\mathbf{r}^c, \mathbf{t}^c$, denote the reconstructed and target images respectively, M is the number of scales used, l_M, c_i and s_i are the luminance, contrast, and structure as defined in [161], α_M, β_i, and γ_i are the weights of the distortion factors at different resolution levels. We adopted the same weights as reported in [168].

Zhao et al. [169] reported that a linear combination of SSIM and ℓ_1 preserves the different properties of an image better than each separately: SSIM encourages the network to preserve structural and contrast information, while ℓ_1 enforces sensitivity to uniform biases to preserve luminance [170]. Since MSSIM reached higher metric values than SSIM (see Table 6.1), we deployed a weighted summation of MSSIM [168] and ℓ_1:

$$L = \alpha \mathrm{MSSIM}(\mathbf{r^c},\mathbf{t^c}) + (1-\alpha)\|\mathbf{r^c} - \mathbf{t^c}\|_1, \tag{6.21}$$

where $\alpha = 0.84$ was chosen, following Zhao et al. [169]. Note that, compared to the ISTA-Net approach, we found it beneficial not to adopt the discrepancy loss as presented in Eq. (6.12) for two reasons. First, we empirically found that tuning the loss multiplier θ is not straightforward, leading to sub-optimal results in terms of the reconstruction loss. Secondly, computing the discrepancy loss is very demanding in terms of GPU memory, requiring to perform a second forward pass where only the thresholding operation is ignored. While feasible, it requires to make the model significantly smaller in terms of learnable parameters, hence reducing model performance significantly.

Table 6.1 reports the quantitative results for the different loss functions. The weighted linear combination of MSSIM and ℓ_1 yielded the best results, where the p-values indicate that the improvement achieved thanks to our modifications is highly consistent across all scans, despite the small improvements on SSIM-values. Fig. 6.3 shows two example results for the different loss functions, confirming the favorable results for the model trained using a combination of MSSIM and ℓ_1. Therefore, this loss function was selected for training the final model. For the remainder of the experiments, MSSIM is used as loss function.

6.4.3 Multi-slice network

The resolution of the images in the dataset is anisotropic with a voxel size of $0.5 \times 0.5 \times 3$ mm^3. Due to the correlation between adjacent slices with respect to anatomical structures in MRI images, we performed an experiment to assess whether inclusion of neighbouring slices into the reconstruction might improve the performance. We compared the 2D scheme using only the center slice with three alternative 2.5D

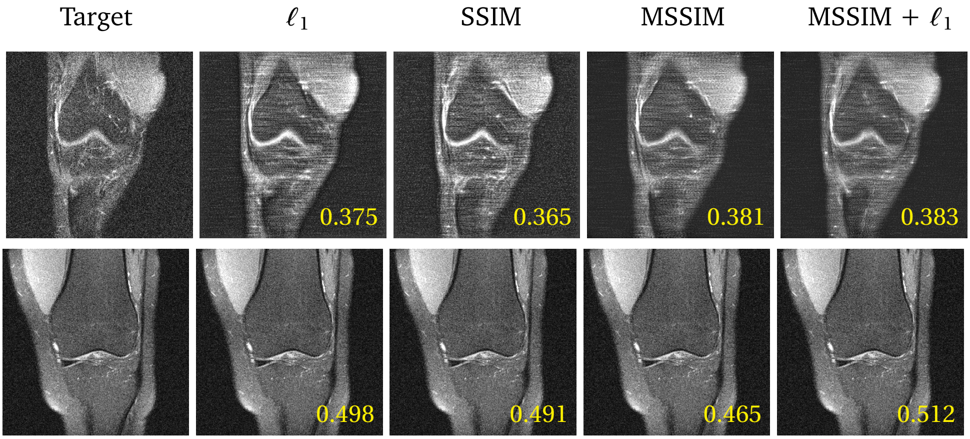

Figure 6.3: Two examples of single-coil 4x for the different loss functions. A small network is used to test several losses. SSIM values are shown in yellow.

Table 6.2: The effect of adopting a 2.5D approach on the 8x single-coil data using the small model. W denotes the loss weight applied to the neighboring slices. Stars denote one-way ANOVA statistical significance.

Network	SSIM $\mu \pm \sigma$	NMSE $\mu \pm \sigma$	PSNR $\mu \pm \sigma$	p-value
2D	0.671±0.143*	0.050±0.034	30.1±3.1	≪0.001
2.5D W0.1	0.549±0.128*	0.089±0.034	26.8±2.1	≪0.001
2.5D W0.2	0.548±0.128*	0.090±0.033	26.8±2.1	≪0.001
2.5D	**0.674±0.143**	**0.048±0.033**	**30.3±3.1**	

schemes: i) the neighboring slices were used together with the center slice as input, but only the center slice was used in the loss function (network 2.5D); ii) and iii) the neighboring slices are also used in the loss, with different weights (0.1 vs 0.2 for the neighbors; 1.0 for the center slice). To compute the first and last slice, we pad the volume with replicas of the edge slices. MSSIM was used for the loss function, 10 blocks, RMSprop as the optimizer, and 16 feature maps.

Table 6.2 shows the results of this experiment, showing that the 2.5D schema very consistently improves over the 2D scheme, and that the loss should only be defined on the center slice. For the final model, this scheme was selected.

6.4.4 Optimizer

We experimented with different optimizers including RMSprop, rectified Adam (RAdam) [130], LookAhead [171] and Ranger [172]. RAdam exploits a dynamic rectifier to adjust the adaptive momentum of Adam [173]. LookAhead not only uses an adaptive learning rate and accelerated schemes but also iteratively updates two sets of weights,

Table 6.3: The effect of the optimizer on performance, using the 8x single-coil validation data. Stars denote one-way ANOVA statistical significance.

Optimizer	SSIM $\mu \pm \sigma$	NMSE $\mu \pm \sigma$	PSNR $\mu \pm \sigma$	p-value
RMSprop	0.673±0.143*	0.048±0.033	30.3±3.1	$\ll$0.001
LookAhead	0.668±0.140*	0.050±0.032	30.0±2.9	$\ll$0.001
Ranger	0.668±0.140*	0.050±0.032	30.0±2.9	$\ll$0.001
RAdam	**0.674±0.141**	**0.048±0.032**	**30.3±3.0**	

Table 6.4: Adaptive-CS-Net vs ISTA-Net$^+$ on the 8x single-coil dataset. ISTA-Net$^+$ has 0.75M trainable parameters, while ISTA-Net-L$^+$ and A-CS-Net have 2.12M trainable parameters. Stars denote one-way ANOVA statistical significance.

Model	SSIM $\mu \pm \sigma$	NMSE $\mu \pm \sigma$	PSNR $\mu \pm \sigma$	p-value
ISTA-Net$^+$	0.547±0.117*	0.169±0.022	23.8±1.9	$\ll$0.001
ISTA-Net-L$^+$	0.543±0.119*	0.103±0.038	26.2±2.0	$\ll$0.001
A-CS-Net	**0.671±0.143**	**0.050±0.034**	**30.1±3.1**	

i.e. fast and slow weights. Ranger combines Radam and LookAhead optimizers into a single one. We used the 2D network with 10 blocks and 16 feature maps for each layer, and MSSIM the loss function.

Table 6.3 tabulates the results for the different optimizers. Since the best results were obtained for the RAdam optimizer, very consistently improving over the other optimizers, this was used for the final network.

6.4.5 Adaptive-CS-Net vs ISTA-Net$^+$

In this experiment, we compare the proposed model to ISTA-Net$^+$ [158]. For this experiment, a 2D network with 10 blocks and 16 feature maps per layer was used, SSIM as loss function, and RAdam as the optimizer. Since ISTA-Net$^+$ uses a much smaller single scale architecture with much fewer network parameters, we added an experiment increasing the feature maps for ISTA-Net$^+$ such that the number of parameters was the same as for our architecture. According to the results reported in Table 6.4, the proposed model outperforms ISTA-Net$^+$ significantly. Figure 6.4 shows a qualitative comparison between ISTA-Net$^+$ and Adaptive-CS-Net on the single-coil 4x dataset. Although for the first image Adaptive-CS-Net reconstructed a better output in terms of the anatomical structure, the output of ISTA-Net-L$^+$ has a higher SSIM value. This implies that the radiological evaluation is a complementary step to judge the quality of the results.

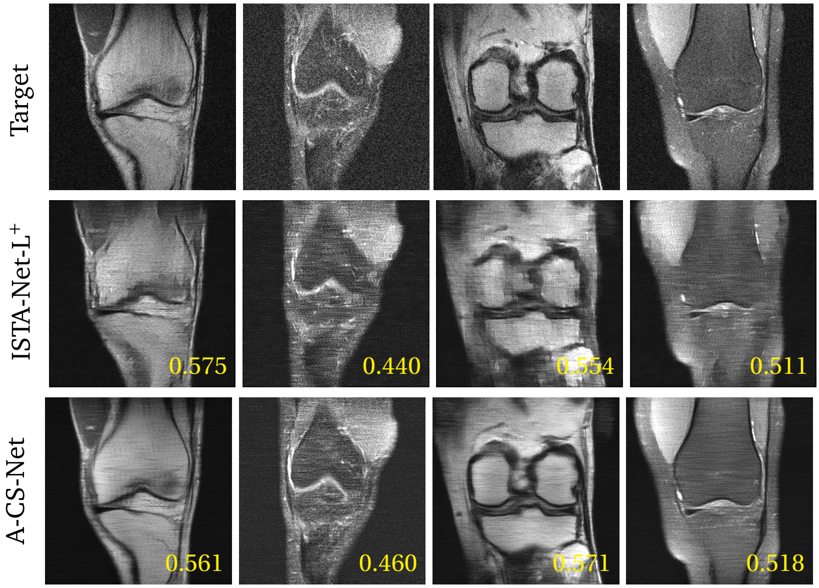

Figure 6.4: Qualitative comparison of Adaptive-CS-Net vs ISTA-Net$^+$ on the single-coil 4x dataset. The SSIM values are shown in yellow.

Table 6.5: The effect of adding priors to the final network on performance, using the multi-coil test data.

Acceleration	prior	SSIM	NMSE	PSNR
4x	–	0.772	**0.025**	30.98
	+	**0.773**	0.028	**33.49**
8x	–	0.674	**0.038**	**30.90**
	+	**0.675**	0.044	30.27

6.4.6 Soft priors

To assess the contribution of the additional soft priors, we compared the full model against a version without known phase behaviour $\mathbf{e}_{\phi,i}$ and without background information $\mathbf{e}_{\text{bg},i}$. Visually, we observed only small differences. To verify the differences in a realistic setting, we submitted the results to the public leaderboard of the fastMRI challenge. As shown in Table 6.5, the network with all priors performed better in terms of the SSIM metric, although the results worsened in terms of NMSE and PSNR. Despite the fact that the improvement was minimal, we decided to adopt all priors for the final model to ensure our participation in the last challenge phase, since the selection was based on SSIM.

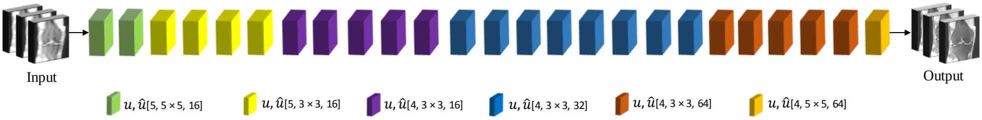
Input
Output
$u, \hat{u}$[5, 5 × 5, 16]
$u, \hat{u}$[5, 3 × 3, 16]
$u, \hat{u}$[4, 3 × 3, 16]
$u, \hat{u}$[4, 3 × 3, 32]
$u, \hat{u}$[4, 3 × 3, 64]
$u, \hat{u}$[4, 5 × 5, 64]

Figure 6.5: Final network, each block has the same structure as shown in Fig. 6.1 and is defined by $\mathscr{U}, \hat{\mathscr{U}}$[number of scales, kernel size, number of feature maps in the first scale]. For all layers Leaky ReLU was used as the activation function.

6.5 Adaptive-CS-NET: Submitted model

In this section, we describe the configuration of the submitted model [174] and analyze the resulting reconstructions. The final performance is evaluated with the quantitative metrics on the test and challenge datasets, and by presenting the radiological scores for the challenge dataset as performed by the fastMRI challenge organizers.

Following our model optimization study, the configuration of the final model was determined as follows. The linear combination of MSSIM and ℓ_1 (6.21) was chosen as the loss function. The 2.5D scheme was chosen with two neighboring slices, with the loss applied only on the central slice. For training the model, the RAdam optimizer was deployed. Fig. 6.5 shows the structure of the final network. Each block is determined by three parameters for the denoiser: 1) the number of scales for the denoiser $\mathscr{U}, \hat{\mathscr{U}}$, 2) the kernel size used in the convolutions and, 3) the number of feature maps in the first convolutional layer, which is then doubled at each scale. According to the experiments presented in Fig. 6.2, the number of reconstruction blocks greatly affects the reconstruction performance, empirically observing that performance still improves when 15 blocks are used. The available GPU memory is a limiting factor when designing a deep neural network. To allow for a large number of blocks, we chose a different design in each block, mixing a less powerful design (16 filters) with more powerful ones (64 filters). By adopting this strategy, our final design contained 25 reconstruction blocks and has 33M parameters.

Fig. 6.6 shows example results of the final network for the multi-coil track from the validation dataset. Fig. 6.7 shows examples from the test and challenge datasets. Table 6.6 shows the SSIM, NMSE, and PSNR values for the test and challenge set (as described in Section 6.2.1), for the images with and without fat suppression and both combined, for both single- and multi-coil MRI scans. For the radiological evaluation, our method scored 2.285, 1.286, and 2.714 for multi-coil 4x, multi-coil 8x, and single-coil 4x, respectively (the closer to 1, the better). The average runtimes for the model are 518 and 327 milliseconds for the multi-coil and the single-coil data, respectively. More details on the results for the challenge were presented in [159].

Table 6.6: Results for the final model for single- and multi-coil data on the test and challenge dataset.

Dataset	Coil		Detail	SSIM	NMSE	PSNR
Test	multi	4x	ALL	0.928	0.005	39.9
			PD	0.961	0.002	41.7
			PDFS	0.891	0.009	37.9
		8x	ALL	0.888	0.009	36.8
			PD	0.937	0.005	38.5
			PDFS	0.843	0.013	35.3
	single	4x	ALL	0.777	0.027	33.7
			PD	0.877	0.010	36.9
			PDFS	0.685	0.043	30.7
		8x	ALL	0.680	0.042	30.5
			PD	0.777	0.019	32.4
			PDFS	0.575	0.067	28.5
Challenge	multi	4x	ALL	0.927	0.005	39.9
		8x	ALL	0.901	0.009	37.4
	single	4x	ALL	0.751	0.030	32.7

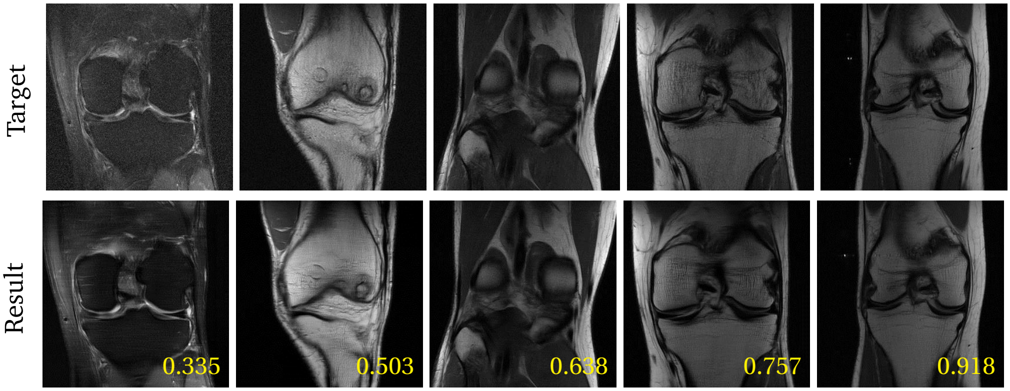

Figure 6.6: Example results of the final model for the multi-coil track accelerated by 8x on the validation dataset. Top row depicts the target image, bottom row the reconstructed images with the SSIM value in yellow.

6.6 Discussion

In this paper we propose a general method, named Adaptive-CS-Net, for reconstructing undersampled MRI data, combining ideas from compressed sensing theory with ideas from MR physics and deep learning. The method was developed in the context of the 2019 fastMRI challenge, which focused on accelerating knee MR imaging. The proposed network is an unrolled iterative learning-based reconstruction scheme, in which a large number of reconstruction blocks refine the MR image by denoising the

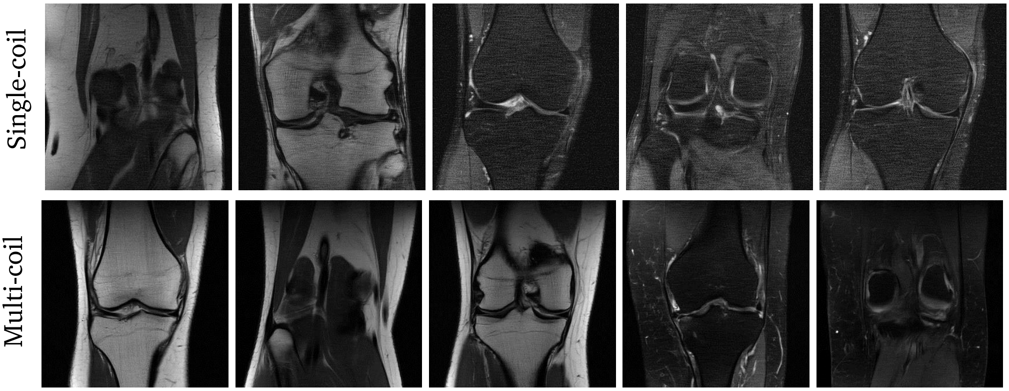

Figure 6.7: Example results of the final network from the test and challenge datasets, for which no ground truth reconstructions are available.

signal in a learned and multi-scale fashion. Moreover, we added neighboring slices as input to the sparsifying transform, as well as a number of soft priors that encode MRI domain knowledge.

The main driver of the performance of our network is the multi-scale architecture, as demonstrated in a direct comparison with ISTA-Net$^+$ that is corrected by the number of trainable parameters. According to the experimental results on the number of blocks for 4x and 8x accelerations of both single- and multi-coil data, we showed that the number of blocks has a large impact on model performance. Therefore, it was decided to use the maximum number of blocks that we could fit into the GPU memory, where we adopted different model designs for the different blocks to save memory. It might be expected that beyond a certain number of blocks, overfitting of the data might occur. However, signs of overfitting were not observed during training and the final number of blocks was only marginally larger than tested in the optimization experiments. Whether further increase in the number of blocks could result in even better performance could be the topic of further experiments. This would, however, need better hardware, as the current design is memory- and time-bound during training. With the current configuration, final model training took approximately 7 days on two V100 GPUs.

We experimented with a large variety of loss functions. Results showed that the linear summation of MSSIM and ℓ_1 performed best. Figure 6.6 showed that poor SNR data yield very low SSIM scores. Surprisingly, within high SNR data, a large variance of SSIM scores is also found. This highlights the fact that further research is required in order to develop better quality metrics. Moreover, we defined a 2.5D scheme to train the network in which three adjacent slices were reconstructed while the loss function was calculated only for the central slice. The proposed scheme outperformed the 2D network as well as 2.5D networks in which the loss was calculated over all slices. By incorporating the neighbouring slices, the network can exploit existing correlations into the reconstruction of the target slice, which is our main target as defined by our loss. It can be expected that for MRI acquisition with less asymmetric voxel sizes, the inclusion of information of neighbouring slices would become more important. However, weighing the loss of the neighbouring slices resulted in less optimal results since it forces the network to solve a more difficult problem: the network has to reconstruct multiple slices instead of a single one. This reduces the effective network capacity per slice, leading to a degradation of the reconstruction performance. We tested different optimizers, where the newly introduced RAdam outperformed the others and we used it for training the final network. We also incorporated prior knowledge, including data consistency, known phase behaviour and background discrimination to support the network in the reconstruction process. We observed that these priors provided only limited extra performance to the network, resulting in visually similar images and minimal difference in the metrics.

We can conclude that the Adaptive-CS-Net is sufficiently powerful to learn directly from the data how to reconstruct the undersampled k-space, being the multi-scale structure and the use of many reconstruction blocks the main driver of our performance. As a future work, we want to better understand how much the network is relying on the priors by adopting interpretable AI techniques such as differentiable image parameterizations for feature visualization [175]. Stronger use of the priors via the loss function is an additional option.

As mentioned before, the radiologist scores were based on the visual quality of the reconstructed images and not on diagnostic interchangeability. Therefore, designing a network based on the diagnosis can be considered a point for further research. We furthermore observed that optimizing for SSIM was needed for reaching the final stage of the challenge, but is not necessarily an ideal representative of radiological image quality. This observation was very recently confirmed in a comparative study by others [176]. The proposed method outperforms the benchmark networks, including U-net [116], cascade net [152], KIKI-net [153], and PD-net [154], on the single-coil track as reported in [150]. It outperforms as well the i-RIM model [145] on the Multi-coil track but not the single coil track [159].

6.7 Conclusion

In this paper we propose an adaptive intelligence algorithm called Adaptive-CS-Net, which was developed in the context of the 2019 fastMRI challenge. In the two clinically relevant tracks of the challenge, using multi-coil MRI acquisitions, the proposed method was leading, while on a simulated single-coil track the method ranked 3rd.

6.8 Acknowledgements

Facebook AI Research and NYU Langone Health are acknowledged for organizing the 2019 fastMRI challenge.