HUMUS-Net: Hybrid Unrolled Multi-Scale Network Architecture for Accelerated MRI Reconstruction

Anonymous Author(s)
Affiliation
Address
email

Abstract

In accelerated MRI reconstruction, the anatomy of a patient is recovered from a set of under-sampled and noisy measurements. Deep learning approaches have been proven to be successful in solving this ill-posed inverse problem and are capable of producing very high quality reconstructions. However, current architectures heavily rely on convolutions, that are content-independent and have difficulties modeling long-range dependencies in images. Recently, Transformers, the workhorse of contemporary natural language processing, have emerged as powerful building blocks for a multitude of vision tasks. These models split input images into non-overlapping patches, embed the patches into lower-dimensional tokens and utilize a self-attention mechanism that does not suffer from the aforementioned weaknesses of convolutional architectures. However, Transformers incur extremely high compute and memory cost when 1) the input image resolution is high and 2) when the image needs to be split into a large number of patches to preserve fine detail information, both of which are typical in low-level vision problems such as MRI reconstruction, having a compounding effect. To tackle these challenges, we propose HUMUS-Net, a hybrid architecture that combines the beneficial implicit bias and efficiency of convolutions with the power of Transformer blocks in an unrolled and multi-scale network. HUMUS-Net extracts high-resolution features via convolutional blocks and refines low-resolution features via a novel Transformer-based multi-scale feature extractor. Features from both levels are then synthesized into a high-resolution output reconstruction. Our network establishes new state of the art on the largest publicly available MRI dataset, the fastMRI dataset. We further demonstrate the performance of HUMUS-Net on two other popular MRI datasets and perform fine-grained ablation studies to validate our design.

1 Introduction

Magnetic resonance imaging (MRI) is a medical imaging technique that uses strong magnetic fields to picture the anatomy and physiological processes of the patient. MRI is one of the most popular imaging modalities as it is non-invasive and doesn't expose the patient to harmful ionizing radiation. The MRI scanner obtains measurements of the body in the spatial frequency domain also called *k-space*. The data acquisition process is MR is often time-consuming. Accelerated MRI [Lustig et al., 2008] addresses this challenge by undersampling in the k-space domain, thus reducing the time patients need to spend in the scanner. However, recovering the underlying anatomy from undersampled measurements is an ill-posed problem (less measurements than unknowns) and thus incorporating some form of prior knowledge is crucial in obtaining high quality reconstructions. Classical MRI reconstruction algorithms rely on the assumption that the underlying signal is sparse in some transform domain and attempt to recover a signal that best satisfies this assumption in a

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

technique known as compressed sensing (CS) [Candes et al., 2006, Donoho, 2006]. These classical CS techniques have slow reconstruction speed and typically enforce limited forms of image priors.

With the emergence of deep learning (DL), data-driven reconstruction algorithms have far surpassed CS techniques (see Ongie et al. [2020] for an overview). DL models utilize large training datasets to extract flexible, nuanced priors directly from data resulting in excellent reconstruction quality. In recent years, there has been a flurry of activity aimed at designing DL architectures tailored to the MRI reconstruction problem. The most popular models are convolutional neural networks (CNNs) that typically incorporate the physics of the MRI reconstruction problem, and utilize tools from mainstream deep learning (residual learning, data augmentation, self-supervised learning). Comparing the performance of such models has been difficult mainly due to two reasons. First, there has been a large variation in evaluation datasets spanning different scanners, anatomies, acquisition models and undersampling patterns rendering direct comparison challenging. Second, medical imaging datasets are often proprietary due to privacy concerns, hindering reproducibility.

More recently, the fastMRI dataset [Zbontar et al., 2019], the largest publicly available MRI dataset, has been gaining ground as a standard benchmark to evaluate MRI reconstruction methods. An annual competition, the fastMRI Challenge [Muckley et al., 2021], attracts significant attention from the machine learning community and acts a driver of innovation in MRI reconstruction. However, over the past years the public leaderboard has been dominated by a single architecture, the End-to-End VarNet Sriram et al. [2020] [1], with most models concentrating very closely around the same performance metrics, hinting at the saturation of current architectural choices.

In this work, we propose HUMUS-Net: a Hybrid, UNrolled, Multi-Scale network architecture for accelerated MRI reconstruction that combines the advantages of well-established architectures in the field with the power of contemporary Transformer-based models. We utilize the strong implicit bias of convolutions, but also address their weaknesses, such as content-independence and inability to model long-range dependencies, by incorporating a novel multi-scale feature extractor that operates over embedded image patches via self-attention. Moreover, we tackle the challenge of high input resolution typical in MRI by performing the computationally most expensive operations on extracted low-resolution features. HUMUS-Net establishes new state of the art in accelerated MRI reconstruction on the largest available MRI knee dataset. At the time of writing this paper, HUMUS-Net is the only Transformer-based architecture on the highly competitive fastMRI Public Leaderboard. Our results are fully reproducible and the source code is available in the supplementary.

2 Background

2.1 Inverse Problem Formulation of Accelerated MRI Reconstruction

An MR scanner obtains measurements of the patient anatomy in the frequency domain, also referred to as *k-space*. Data acquisition is performed via various receiver coils positioned around the anatomy being imaged, each with different spatial sensitivity. Given a total number of N receiver coils, the measurements obtained by the ith coil can be written as

$$\boldsymbol{k_i} = \mathcal{F}\boldsymbol{S_i}\boldsymbol{x^*} + \boldsymbol{z_i}, \;\; i = 1, .., N,$$

where $\boldsymbol{x^*} \in \mathbb{C}^n$ is the underlying patient anatomy of interest, $\boldsymbol{S_i}$ is a diagonal matrix that represents the sensitivity map of the ith coil, $\mathcal{F}$ is a multi-dimensional Fourier-transform, and z_i denotes the measurement noise corrupting the observations obtained from coil i. We use $\boldsymbol{k} = (\boldsymbol{k}_1, ..., \boldsymbol{k}_N)$ as a shorthand for the concatenation of individual coil measurements and $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ as the corresponding image domain representation after inverse Fourier transformation.

Since MR data acquisition time is proportional to the portion of k-space being scanned, obtaining fully-sampled data is time-consuming. Therefore, in *accelerated MRI* scan times are reduced by undersampling in k-space domain. The undersampled k-space measurements from coil i take the form

$$\tilde{\boldsymbol{k}}_{\boldsymbol{i}} = \boldsymbol{M}\boldsymbol{k_i} \;\; i = 1, .., N,$$

where $\boldsymbol{M}$ is a diagonal matrix representing the binary undersampling mask, that has 0 values for all missing frequency components that have not been sampled during accelerated acquisition.

[1] At the time of writing this paper, the number one model on the leaderboard is AIRS-Net, however the complete architecture/training details as well as the code of this method are not available to the public and we were unable to reproduce their results based on the limited available information.

The forward model that maps the underlying anatomy to coil measurements can be written concisely as $\tilde{\boldsymbol{k}} = \mathcal{A}(\boldsymbol{x}^*)$, where $\mathcal{A}(\cdot)$ is the linear forward mapping and $\tilde{\boldsymbol{k}}$ is the stacked vector of all undersampled coil measurements. Our target is to reconstruct the ground truth object $\boldsymbol{x}^*$ from the noisy, undersampled measurements $\tilde{\boldsymbol{k}}$. Since we have fewer observations than variables to recover, perfect reconstruction in general is not possible. In order to make the problem solvable, prior knowledge on the underlying object is typically incorporated in the form of sparsity in some transform domain. This formulation, known as compressed sensing [Candes et al., 2006, Donoho, 2006], provides a classical framework for accelerated MRI reconstruction [Lustig et al., 2008]. In particular, the above recovery problem can be formulated as a regularized inverse problem

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\| \mathcal{A}(\boldsymbol{x}) - \tilde{\boldsymbol{k}} \right\|^2 + \mathcal{R}(\boldsymbol{x}), \tag{2.1}$$

where $\mathcal{R}(\cdot)$ is a regularizer that encapsulates prior knowledge on the object, such as sparsity in some wavelet domain.

2.2 Deep Learning-based Accelerated MRI Reconstruction

More recently, data-driven deep learning-based algorithms tailored to the accelerated MRI reconstruction problem have surpassed the classical compressed sensing baselines. Convolutional neural networks trained on large datasets have established new state of the art in many medical imaging tasks. The highly popular U-Net [Ronneberger et al., 2015] and other similar encoder-decoder architectures have proven to be successful in a range of medical image reconstruction [Hyun et al., 2018, Han and Ye, 2018] and segmentation Çiçek et al. [2016], Zhou et al. [2018] problems. In the encoder path, the network learns to extract a set of deep, low-dimensional features from images via a series of convolutional and downsampling operations. These concise feature representations are then gradually upsampled and filtered in the decoder to the original image dimensions. Thus the network learns a hierarchical representation over the input image distribution.

Unrolled networks constitute another line of work that has been inspired by popular optimization algorithms used to solve compressed sensing reconstruction problems. These deep learning models consist of a series of sub-networks, also known as cascades, where each sub-network corresponds to an unrolled iteration of popular algorithms such as gradient descent Zhang and Ghanem [2018] or ADMM Sun et al. [2016]. In the context of MRI reconstruction, one can view network unrolling as solving a sequence of smaller denoising problems, instead of the complete recovery problem in one step. Various convolutional neural networks have been employed in the unrolling framework achieving excellent performance in accelerated MRI reconstruction Putzky et al. [2019], Hammernik et al. [2018, 2019]. E2E-VarNet [Sriram et al., 2020] is the current state-of-the-art convolutional model on the fastMRI dataset. E2E-VarNet transforms the optimization problem in (2.1) to the k-space domain and unrolls the gradient descent iterations into T cascades, where the tth cascade represents the computation

$$\hat{\boldsymbol{k}}^{\boldsymbol{t+1}} = \hat{\boldsymbol{k}}^{\boldsymbol{t}} - \mu^t \boldsymbol{M}(\hat{\boldsymbol{k}}^{\boldsymbol{t}} - \tilde{\boldsymbol{k}}) + \mathcal{G}(\hat{\boldsymbol{k}}^{\boldsymbol{t}}), \tag{2.2}$$

where $\hat{\boldsymbol{k}}^{\boldsymbol{t}}$ is the estimated reconstruction in the k-space domain at cascade t, μ^t is a learnable step size parameter and $\mathcal{G}(\cdot)$ is a learned mapping representing the gradient of the regularization term in (2.1). The first term is also known as *data consistency* (DC) term as it enforces the consistency of the estimate with the available measurements.

3 Related Work

Transformers in Vision – Vision Transformer (ViT) [Dosovitskiy et al., 2020], a fully non-convolutional vision architecture, has demonstrated state-of-the-art performance on image classification problems when pre-trained on large-scale image datasets. The key idea of ViT is to split the input image into non-overlapping patches, embed each patch via a learned linear mapping and process the resulting tokens via stacked self-attention and multi-layer perceptron (MLP) blocks. For more details we refer the reader to Appendix F and [Dosovitskiy et al., 2020]. The benefit of Transformers over convolutional architectures in vision lies in their ability to capture long-range dependencies in images via the self-attention mechanism.

Since the introduction of ViT, similar attention-based architectures have been proposed for many other vision tasks such as object detection [Carion et al., 2020], image segmentation [Wang et al., 2021b]

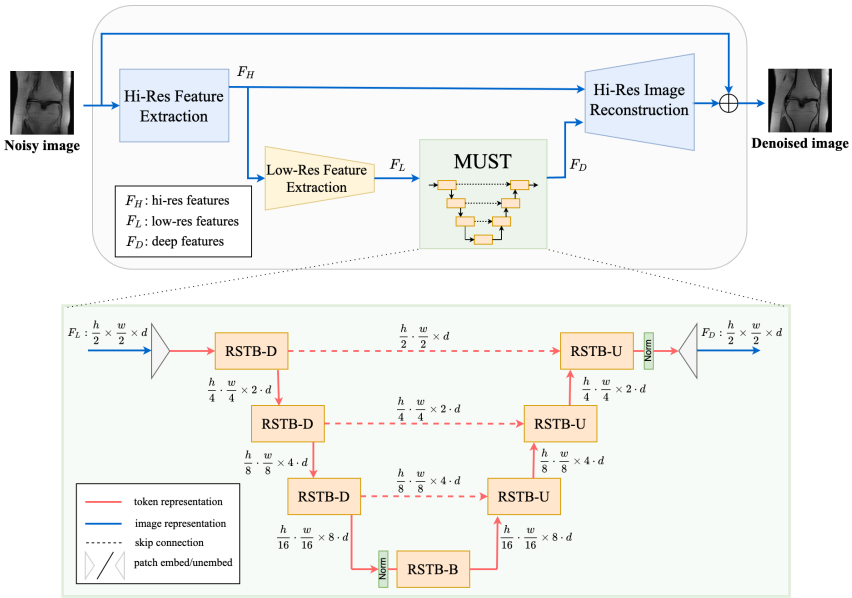


Figure 1: Overview of the HUMUS-Block architecture. First, we extract high-resolution features $\boldsymbol{F_H}$ from the input noisy image through a convolution layer f_H. Then, we apply a convolutional feature extractor f_L to obtain low-resolution features and process them using a Transformer-convolutional hybrid multi-scale feature extractor. The shallow, high-resolution and deep, low-resolution features are then synthesized into the final high-resolution denoised image.

and restoration [Cao et al., 2021b, Chen et al., 2021, Liang et al., 2021, Zamir et al., 2021, Wang et al., 2021c]. A key challenge for Transformers in low-level vision problems is the quadratic compute complexity of the global self-attention with respect to the input dimension. In some works, this issue has been mitigated by splitting the input image into fixed size patches and processing the patches independently [Chen et al., 2021]. Others focus on designing hierarchical Transformer architectures [Heo et al., 2021, Wang et al., 2021b] similar to popular ResNets [He et al., 2015]. Authors in Zamir et al. [2021] propose applying self-attention channel-wise rather than across the spatial dimension thus reducing the compute overhead to linear complexity. Another successful architecture, the Swin Transformer [Liu et al., 2021], tackles the quadratic scaling issue by computing self-attention on smaller local windows. To encourage cross-window interaction, windows in subsequent layers are shifted relative to each other.

Transformers in Medical Imaging – Transformer architectures have been proposed recently to tackle various medical imaging problems. Authors in Cao et al. [2021a] design a U-Net-like architecture for medical image segmentation where the traditional convolutional layers are replaced by Swin Transformer blocks. They report strong results on multi-organ and cardiac image segmentation. In Zhou et al. [2021] a hybrid convolutional and Transformer-based U-Net architecture is proposed tailored to volumetric medical image segmentation with excellent results on benchmark datasets. Similar encoder-decoder architectures for various medical segmentation tasks have been investigated in other works [Huang et al., 2021, Wu et al., 2022]. However, these networks are tailored for image segmentation, a task less sensitive to fine details in the input, and thus larger patch sizes are often used (for instance 4 in Cao et al. [2021a]). This allows the network to process larger input images, as the number of token embedding is greatly reduced, but as we demonstrate in Section 5.2 embedding individual pixels as 1×1 patches is crucial for MRI reconstruction. Thus, compute and memory barriers stemming from large input resolutions are more severe in the MRI reconstruction task and therefore novel approaches are needed.

Promising results have been reported employing Transformers in medical image denoising problems, such as low-dose CT denoising [Wang et al., 2021a, Luthra et al., 2021] and low-count PET/MRI denoising [Zhang et al., 2021]. However, these studies fail to address the challenge of poor scaling to large input resolutions, and only work on small images via either downsampling the original dataset [Luthra et al., 2021] or by slicing the large input images into smaller patches [Wang et al., 2021a]. In contrast, our proposed architecture works directly on the large resolution images that often

arise in MRI reconstruction. Even though some work exists on Transformer-based architectures for supervised accelerated MRI reconstruction [Huang et al., 2022, Lin and Heckel, 2022, Feng et al., 2021], and for unsupervised pre-trained reconstruction [Korkmaz et al., 2022], to the best of our knowledge ours is the first work to demonstrate state-of-the-art results on large-scale MRI datasets such as the fastMRI dataset.

4 Method

HUMUS-Net combines the efficiency and beneficial implicit bias of convolutional networks with the powerful general representations of Transformers and their capability to capture long-range pixel dependencies. The resulting hybrid network processes information both in image representation (via convolutions) and in patch-embedded token representation (via Transformer blocks). Our proposed architecture consists of a sequence of sub-networks, also called cascades. Each cascade represents an unrolled iteration of an underlying optimization algorithm in k-space (see (2.2)), with an image-domain denoiser, the HUMUS-Block. First, we describe the architecture of the HUMUS-Block, the core component in the sub-network. Then, we specify the high-level, k-space unrolling architecture of HUMUS-Net in Section 4.3.

4.1 HUMUS-Block Architecture

The HUMUS-Block acts as an image-space denoiser that receives an intermediate reconstruction from the previous cascade and performs a single step of denoising to produce an improved reconstruction for the next cascade. It extracts high-resolution, shallow features and low-resolution, deep features through a novel multi-scale transformer-based block, and synthesizes high-resolution features from those. The high-level overview of the HUMUS-Block is depicted in Fig. 1.

High-resolution Feature Extraction– The input to our network is a noisy complex-valued image $\boldsymbol{x}_{in} \in \mathbb{R}^{h \times w \times c_{in}}$ derived from undersampled k-space data, where the real and imaginary parts of the image are concatenated along the channel dimension. First, we extract high-resolution features $\boldsymbol{F_H} \in \mathbb{R}^{h \times w \times d_H}$ from the input noisy image through a convolution layer f_H written as $\boldsymbol{F_H} = f_H(\boldsymbol{x}_{in})$. This initial 3×3 convolution layer provides early visual processing at a relatively low cost and maps the input to a higher, d_H dimensional feature space. Important to note that the resolution of the extracted features is the same as the spatial resolution of the input image.

Low-resolution Feature Extraction– In case of MRI reconstruction, this resolution is typically significantly higher than commonly used image datasets (32×32 - 256×256), posing a significant challenge to contemporary Transformer-based models. Thus we apply a convolutional feature extractor f_L to obtain low-resolution features $\boldsymbol{F_L} = f_L(\boldsymbol{F_H})$, with $\boldsymbol{F_L} \in \mathbb{R}^{h_L \times w_L \times d_L}$ where f_L consists of a sequence of convolutional blocks and spatial downsampling operations. The specific architecture is depicted in Figure 4. The purpose of this module is to perform deeper visual processing and to provide manageable input size to the subsequent computation and memory heavy hybrid processing module. In this work, we choose $h_L = \frac{h}{2}$ and $w_L = \frac{w}{2}$ which strikes a balance between preserving spatial information and resource demands. Furthermore, in order to compensate for the reduced resolution we increase the feature dimension to $d_L = 2 \cdot d_H := d$.

Deep Feature Extraction– The most important part of our model is MUST, a **MU**lti-scale residual **S**win **T**ransformer network. MUST is a multi-scale hybrid feature extractor that takes the low-resolution image representations $\boldsymbol{F_L}$ and performs hierarchical Transformer-convolutional hybrid processing in an encoder-decoder fashion, producing deep features $\boldsymbol{F_D} = f_D(\boldsymbol{F_L})$, where the specific architecture behind f_D is detailed in Subsection 4.2.

High-resolution Image Reconstruction– Finally, we combine information from shallow, high-resolution features $\boldsymbol{F_H}$ and deep, low-resolution features $\boldsymbol{F_D}$ to reconstruct the high-resolution residual image via a convolutional reconstruction module f_R. The residual learning paradigm allows us to learn the difference between noisy and clean images and helps information flow within the network [He et al., 2015]. Thus the final denoised image $\boldsymbol{x}_{out} \in \mathbb{R}^{h \times w \times c_{in}}$ is obtained as $\boldsymbol{x}_{out} = \boldsymbol{x}_{in} + f_R(\boldsymbol{F_H}, \boldsymbol{F_D})$. The specific architecture of the reconstruction network is depicted in Figure 4.

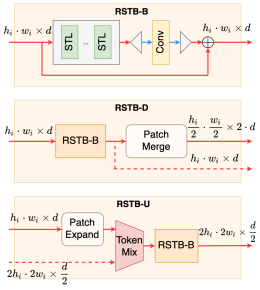


Figure 2: Depiction of different RSTB modules used in the HUMUS-Block.

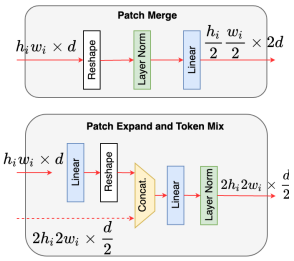


Figure 3: Patch merge and expand operations used in our multi-scale feature extractor.

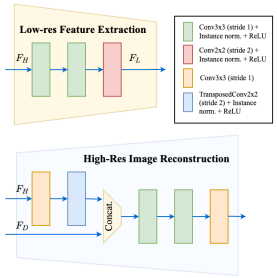


Figure 4: Architecture of convolutional blocks for feature extraction and reconstruction.

4.2 Multi-scale Hybrid Feature Extraction via MUST

The key component to our architecture is MUST, a multi-scale hybrid encoder-decoder architecture that performs deep feature extraction in both image and token representation (Figure 1, bottom). First, individual pixels of the input representation of shape $\frac{h}{2} \times \frac{w}{2} \times d$ are flattened and passed through a learned linear mapping to yield $\frac{h}{2} \cdot \frac{w}{2}$ tokens of dimension d. Tokens corresponding to different image patches are subsequently merged in the encoder path, resulting in a concise latent representation. This highly descriptive representation is passed through a bottleneck block and progressively expanded by combining tokens from the encoder path via skip connections. The final output is rearranged to match the exact shape of the input low-resolution features $\boldsymbol{F_L}$, yielding a deep feature representation $\boldsymbol{F_D}$.

Our design is inspired by the success of Residual Swin Transformer Blocks (RSTB) in image denoising and super-resolution [Liang et al., 2021]. RSTB features a stack of Swin Transformer layers (STL) that operate on tokens via a windowed self-attention mechanism [Liu et al., 2021], followed by convolution in image representation. However, RSTB blocks operate on a single scale, therefore they cannot be readily applied in a hierarchical encoder-decoder architecture. Therefore, we design three variations of RSTB to facilitate multi-scale processing as depicted in Figure 2.

RSTB-B is the bottleneck block responsible for processing the encoded latent representation while maintaining feature dimensions. Thus, we keep the default RSTB architecture for our bottleneck block, which already operates on a single scale.

RSTB-D has a similar function to convolutional downsampling blocks in U-Nets, but it operates on embedded tokens. Given an input with size $h_i \cdot w_i \times d$, we pass it through an RSTB-B block and apply *PatchMerge* operation. *PatchMerge* linearly combines tokens corresponding to 2×2 non-overlapping image patches, while simultaneously increasing the embedding dimension (see Figure 3, top and Appendix H for more details) resulting in an output of size $\frac{h_i}{2} \cdot \frac{w_i}{2} \times 2 \cdot d$. Furthermore, RSTB-D outputs the higher dimensional representation before patch merging to be subsequently used in the decoder path via skip connection.

RSTB-U used in the decoder path is analogous to convolutional upsampling blocks. An input with size $h_i \cdot w_i \times d$ is first expanded into a larger number of lower dimensional tokens through a linear mapping via *PatchExpand* (see Figure 3, bottom and Appendix H for more details). *PatchExpand* reverses the effect of *PatchMerge* on feature size, thus resulting in $2h_i \cdot 2w_i$ tokens of dimension $\frac{d}{2}$. Next, we mix information from the obtained expanded tokens with skip embeddings from higher scales via *TokenMix*. This operation linearly combines tokens from both paths and normalizes the resulting vectors. Finally, the mixed tokens are processed by an RSTB-B block.

4.3 Iterative Unrolling

Architectures derived from unrolling the iterations of various optimization algorithms have proven to be successful in tackling various inverse problems including MRI reconstruction. These architecture can be interpreted as a cascade of simpler denoisers, each of which progressively refines the estimate from the preceding unrolled iteration.

Following Sriram et al. [2020], we unroll the gradient descent iterations of the inverse problem in (2.1) in k-space domain, yielding the iterative update scheme in (2.2). We apply regularization in image

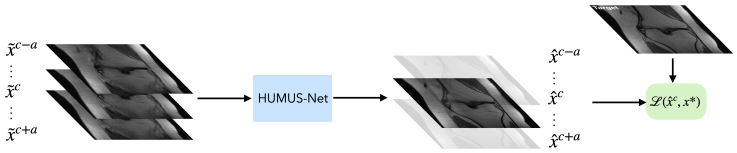


Figure 5: Adjacent slice reconstruction (depicted in image domain for visual clarity): HUMUS-Net takes a volume of adjacent slices $(\tilde{\boldsymbol{x}}^{c-a}, ..., \tilde{\boldsymbol{x}}^{c}, ..., \tilde{\boldsymbol{x}}^{c+a})$ and jointly reconstructs a volume $(\hat{\boldsymbol{x}}^{c-a}, ..., \hat{\boldsymbol{x}}^{c}, ..., \hat{\boldsymbol{x}}^{c+a})$. The reconstruction loss $\mathcal{L}$ is calculated only on the center slice $\hat{\boldsymbol{x}}^{c}$.

domain via our proposed HUMUS-Block, that is we have $\mathcal{G}(\boldsymbol{k}) = \mathcal{F}\left(\mathcal{E}\left(\mathbf{D}\left(\mathcal{R}\left(\mathcal{F}^{-1}(\boldsymbol{k})\right)\right)\right)\right)$, where $\mathbf{D}$ denotes the HUMUS-Block, $\mathcal{R}(\boldsymbol{x_1}, ..., \boldsymbol{x_N}) = \sum_{i=1}^{N} \boldsymbol{S}_i^* x_i$ is the *reduce* operator that combines coil images via the corresponding sensitivity maps and $\mathcal{E}(\boldsymbol{x}) = (\boldsymbol{S}_1\boldsymbol{x}, ..., \boldsymbol{S}_N\boldsymbol{x})$ is the *expand* operator that maps the combined image back to individual coil images. The sensitivity maps can be estimated a priori using methods such as ESPIRiT [Uecker et al., 2014] or learned in an end-to-end fashion via a Sensitivity Map Estimator (SME) network proposed in Sriram et al. [2020]. In this work we aspire to design an end-to-end approach and thus we use the latter method and estimate the sensitivity maps from the low-frequency (ACS) region of the undersampled input measurements during training using a standard U-Net network.

4.4 Adjacent Slice Reconstruction (ASR)

We observe improvements in reconstruction quality when instead of processing the undersampled data slice-by-slice, we jointly reconstruct a set of adjacent slices via HUMUS-Net (Figure 5). That is, if we have a volume of undersampled data $\tilde{\boldsymbol{k}}^{vol} = (\tilde{\boldsymbol{k}}^1, ..., \tilde{\boldsymbol{k}}^K)$ with K slices, when reconstructing slice c, we instead reconstruct the volume $(\tilde{\boldsymbol{k}}^{c-a}, ..., \tilde{\boldsymbol{k}}^{c-1}, \tilde{\boldsymbol{k}}^{c}, \tilde{\boldsymbol{k}}^{c+1}, ..., \tilde{\boldsymbol{k}}^{c+a})$ by concatenating them along the coil channel dimension, where a denotes the number of adjacent slices added on each side. However, we only calculate and backpropagate the loss on the center slice c of the reconstructed volume. The benefit of ASR is that the network can remove artifacts corrupting individual slices as it sees a larger context of the slice by observing its neighbors. Even though ASR increases compute cost, it is important to note that it does not impact the number of token embeddings (spatial resolution is unchanged) and thus can be combined favorably with Transformer-based methods.

5 Experiments

In this section we provide experimental results on our proposed architecture, HUMUS-Net. First, we demonstrate the reconstruction performance of our model on various datasets, including the large-scale fastMRI dataset. Then, we justify our design choices through a set of ablation studies. Our source code is available in the supplementary.

5.1 Benchmark Experiments

We investigate the performance of HUMUS-Net on three different datasets. We use the structural similarity index measure (SSIM) [Wang et al., 2003] as a basis of our evaluation, which is the most common evaluation metric in medical image reconstruction. In all of our experiments, we follow the setup of the fastMRI multi-coil knee track with $8\times$ acceleration in order to provide comparison with state-of-the-art networks of the Public Leaderboard. That is, we perform retrospective undersampling of the fully-sampled k-space data by randomly subsampling 12.5% of whole k-space lines in the phase encoding direction, keeping 4% of lowest frequency adjacent k-space lines. During training, we generate random masks following the above method, whereas for the validation dataset we keep the masks fixed for each k-space volume. For HUMUS-Net, we center crop and pad inputs to 384×384. We compare the reconstruction quality of our proposed model with the current best performing network, E2E-VarNet. For details on HUMUS-Net hyperparameters and training, we refer the reader to the supplementary. For E2E-VarNet, we use the hyperparameters specified in Sriram et al. [2020].

fastMRI – The fastMRI dataset [Zbontar et al., 2019] is the largest publicly available MRI dataset with competitive baseline models and a public leaderboard, and thus provides an opportunity to directly compare different algorithms. Specifically, we run experiments on the multi-coil knee dataset, consisting of close to $35k$ slices in 973 volumes. We use the default HUMUS-Net model defined

fastMRI knee multi-coil $\times 8$ test data			
Method	**SSIM**(↑)	**PSNR**(↑)	**NMSE**(↓)
E2E-VarNet [Sriram et al., 2020]	0.8900	36.9	0.0089
E2E-VarNet† [Sriram et al., 2020]	0.8920	37.1	0.0085
XPDNet [Ramzi et al., 2020]	0.8893	37.2	0.0083
Σ-Net [Hammernik et al., 2019]	0.8877	36.7	0.0091
i-RIM [Putzky et al., 2019]	0.8875	36.7	0.0091
U-Net [Zbontar et al., 2019]	0.8640	34.7	0.0132
HUMUS-Net (ours)	0.8936	37.0	0.0086
HUMUS-Net (ours)†	**0.8945**	**37.3**	**0.0081**

Table 1: Performance of state-of-the-art accelerated MRI reconstruction techniques on the fastMRI knee test dataset. Most models are trained only on the fastMRI training dataset, if available we show results of models trained on the fastMRI combined training and validation dataset denoted by (†).

Validation SSIM			
Method	**fastMRI knee**	**Stanford 2D**	**Stanford 3D**
E2E-VarNet [Sriram et al., 2020]	0.8908	0.8928 ± 0.0168	0.9432 ± 0.0063
HUMUS-Net (ours)	**0.8934**	$\mathbf{0.8954 \pm 0.0136}$	$\mathbf{0.9453 \pm 0.0065}$

Table 2: Validation SSIM of HUMUS-Net on various datasets. For datasets with multiple train-validation split runs we show the mean and standard error of the runs.

Ablation studies						
Method	**Unrolled?**	**Multi-scale?**	**Low-res features?**	**Patch size**	**Embed. dim.**	**SSIM**
Un-SS	✓	✗	✗	1	12	0.9319 ± 0.0080
Un-MS	✓	✓	✗	1	12	0.9357 ± 0.0038
Un-MS-Patch2	✓	✓	✗	2	36	0.9171 ± 0.0075
HUMUS-Net	✓	✓	✓	1	36	$\mathbf{0.9449 \pm 0.0064}$
SwinIR						0.9336 ± 0.0069
E2E-VarNet						0.9432 ± 0.0063

Table 3: Results of ablation studies on HUMUS-Net, evaluated on the Stanford 3D MRI dataset.

above with 3 adjacent slices as input. Table 1 demonstrates our results compared to the best published models from the fastMRI Leaderboard[2] evaluated on the test dataset. Our model establishes new state of the art in terms of SSIM on this dataset by a large margin, and achieves comparable or slightly better performance than other methods in terms of PSNR and NMSE. Moreover, as seen in the second column of Table 2, we evaluated our model on the fastMRI validation dataset as well and compared our results to E2E-VarNet, the best performing model from the leaderboard. We observe similar improvements in terms of the reconstruction SSIM metric to the test dataset. Visual inspection of reconstructions shows that HUMUS-Net recovers very fine details in images that may be missed by other state-of-the-art reconstruction algorithms (see attached figures in the supplementary).

Stanford 2D – Next, we run experiments on the Stanford2D FSE [Cheng] dataset, a publicly available MRI dataset consisting of scans from various anatomies (pelvis, lower extremity and more) in 89 fully-sampled volumes. We randomly sample 80% of volumes as train data and use the rest for validation. We randomly generate 3 different train-validation splits this way to reduce variations in the presented metrics. As slices in this dataset have widely varying shapes across volumes, we center crop the target images to keep spatial resolution within 384×384. We use the default HUMUS-Net defined above with single slices as input. Our results comparing the best performing MRI reconstruction model with HUMUS-Net is shown in the third column of Table 2. We present the mean SSIM of all runs along with the standard error. We achieve improvements of similar magnitude as on the fastMRI dataset. These results demonstrate the effectiveness of HUMUS-Net on a more diverse dataset featuring multiple anatomies.

Stanford 3D – Finally, we evaluate our model on the Stanford Fullysampled 3D FSE Knees dataset [Sawyer et al., 2013], a public MRI dataset including 20 volumes of knee MRI scans. We generate train-validation splits using the method described for Stanford 2D and perform 3 runs. We use the default HUMUS-Net network with single slices as input. The last column of Table 2 compares our results to E2E-VarNet, showing improvements of similar scales as on other datasets we have investigated in this work. This experiment demonstrates that HUMUS-Net performs well not only on large-scale MRI datasets, but also on smaller problems.

[2] https://fastmri.org/leaderboards

5.2 Ablation Studies

In this section, we motivate our design choices through a set of ablation studies. We start from SwinIR, a general image reconstruction network, and highlight its weaknesses for MRI. Then, we demonstrate step-by-step how we addressed these shortcomings and arrived at the HUMUS-Net architecture. We train the models on the Stanford 3D dataset. More details can be found in the supplementary. The results of our ablation studies are summarized in Table 3.

First, we investigate SwinIR, a state-of-the-art image denoiser and super-resolution model that features a hybrid Transformer-convolutional architecture. In order to handle the $10\times$ larger input sizes (320×320) in our MRI dataset compared to the input images this network has been designed for (128×128), we reduce the embedding dimension of SwinIR to fit into GPU memory (16 GB). We find that compared to models designed for MRI reconstruction, such as E2E-VarNet (last row in Table 3) SwinIR performs poorly. This is not only due to the reduced network size, but also due to the fact that SwinIR is not specifically designed to take the MRI forward model into consideration.

Next, we unroll SwinIR and add a sensitivity map estimator. We refer to this model as *Un-SS*. Due to unrolling, we have to further reduce the embedding dimension of the denoiser and also decrease the depth of the network in order to fit into GPU memory. *Un-SS*, due to its small size, performs slightly worse than vanilla SwinIR and significantly lags behind the E2E-VarNet architectures. We note that SwinIR operates over a single, full-resolution scale, whereas state-of-the-art MRI reconstruction models typically incorporate multi-scale processing in the form of U-Net-like architectures.

Thus, we replace SwinIR by MUST, our proposed multi-scale hybrid processing unit, but keep the embedding dimension in the largest-resolution scale fixed. The obtained network, which we call *Un-MS*, has overall lower computational cost when compared with *Un-SS*, however as Table 3 shows MRI reconstruction performance has significantly improved compared to both *Un-SS* and vanilla SwinIR, which highlights the efficiency of our proposed multi-scale feature extractor. Reconstruction performance is limited by the low dimension of patch embeddings, which we are unable to increase further due to our compute and memory constraints originating in the high-resolution inputs.

The most straightforward approach to tackle the challenge of high input resolution is to increase the patch size. To test this idea, we take *Un-MS* and embed 2×2 patches of the inputs, thus reducing the number of tokens processed by the network by a factor of 4. We refer to this model as *Un-MS-Patch2*. This reduction in compute and memory load allows us to increase network capacity by increasing the embedding dimension 3-folds (to fill GPU memory again). However, *Un-MS-Patch2* performs *much* worse than previous models using patch size of 1. For classification problems, where the general image context is more important than small details, patches of 16×16 or 8×8 are considered typical [Dosovitskiy et al., 2020]. Even for more dense prediction tasks such as medical image segmentation, patch size of 4×4 have been used successfully Cao et al. [2021a]. However, our experiments suggest that in low-level vision tasks such as MRI reconstruction using patches larger than 1×1 may be detrimental due to loss of crucial high-frequency detail information.

Our approach to address the heavy computational load of Transformers for large input resolutions where increasing the patch size is not an option is to process lower resolution features extracted via convolutions. That is we replace MUST in *Un-MS* by a HUMUS-Block, resulting in our proposed HUMUS-Net architecture. We train a smaller version of HUMUS-Net with the same embedding dimension as *Un-MS*. As seen in Table 3, our model achieves the best performance across all other proposed solutions, even surpassing E2E-VarNet. This series of incremental studies highlights the importance of each architectural design choice leading to our proposed HUMUS-Net architecture.

6 Conclusion

In this paper, we introduce HUMUS-Net, an unrolled, Transformer-convolutional hybrid network for accelerated MRI reconstruction. HUMUS-Net achieves state-of-the-art performance on the fastMRI dataset and greatly outperforms all previous published and reproducible methods. We demonstrate the performance of our proposed method on two other MRI datasets and perform fine-grained ablation studies to motivate our design choices and emphasize the compute and memory challenges of Transformer-based architectures on low-level and dense computer vision tasks such as MRI reconstruction. A limitation of our current architecture is that it requires fixed-size inputs, which we intend to address with a more flexible design in the future. This work opens the door for the adoption of a multitude of promising techniques introduced recently in the literature for Transformers, which we leave for future work.

References

Emmanuel J. Candes, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-Unet: Unet-like pure Transformer for medical image segmentation. *arXiv:2105.05537*, May 2021a. arXiv: 2105.05537.

Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution Transformer. *arXiv preprint arXiv:2106.06847*, 2021b.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with Transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021.

Joseph Y. Cheng. Stanford 2D FSE. http://mridata.org/list?project=Stanford 2D FSE.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.

David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929 [cs]*, 2020.

Chun-Mei Feng, Yunlu Yan, Huazhu Fu, Li Chen, and Yong Xu. Task Transformer network for joint MRI reconstruction and super-resolution. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 307–317, 2021.

Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P Recht, Daniel K Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic Resonance in Medicine*, 79(6):3055–3071, 2018.

Kerstin Hammernik, Jo Schlemper, Chen Qin, Jinming Duan, Ronald M Summers, and Daniel Rueckert. Σ-net: Systematic evaluation of iterative deep neural networks for fast parallel MR image reconstruction. *arXiv preprint arXiv:1912.09278*, 2019.

Yoseob Han and Jong Chul Ye. Framing U-Net via deep convolutional framelets: Application to sparse-view CT. *IEEE Transactions on Medical Imaging*, 37(6):1418–1429, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021.

Jiahao Huang, Yingying Fang, Yinzhe Wu, Huanjun Wu, Zhifan Gao, Yang Li, Javier Del Ser, Jun Xia, and Guang Yang. Swin Transformer for fast MRI. *arXiv preprint arXiv:2201.03230*, 2022.

Xiaohong Huang, Zhifang Deng, Dandan Li, and Xueguang Yuan. Missformer: An effective medical image segmentation Transformer. *arXiv preprint arXiv:2109.07162*, 2021.

Chang Min Hyun, Hwa Pyung Kim, Sung Min Lee, Sungchul Lee, and Jin Keun Seo. Deep learning for undersampled MRI reconstruction. *Physics in Medicine & Biology*, 63(13):135007, 2018.

Yilmaz Korkmaz, Salman UH Dar, Mahmut Yurt, Muzaffer Özbey, and Tolga Cukur. Unsupervised MRI reconstruction via zero-shot learned adversarial Transformers. *IEEE Transactions on Medical Imaging*, 2022.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using Swin Transformer. *arXiv:2108.10257*, 2021.

Kang Lin and Reinhard Heckel. Vision Transformers enable fast and robust accelerated MRI. In *Medical Imaging with Deep Learning*, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using shifted windows. *arXiv:2103.14030*, 2021.

Michael Lustig, David L. Donoho, Juan M. Santos, and John M. Pauly. Compressed sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72–82, 2008.

Achleshwar Luthra, Harsh Sulakhe, Tanish Mittal, Abhishek Iyer, and Santosh Yadav. Eformer: Edge enhancement based Transformer for medical image denoising. *arXiv preprint arXiv:2109.08044*, 2021.

Matthew J Muckley, Bruno Riemenschneider, Alireza Radmanesh, Sunwoo Kim, Geunu Jeong, Jingyu Ko, Yohan Jun, Hyungseob Shin, Dosik Hwang, Mahmoud Mostapha, et al. Results of the 2020 fastMRI challenge for machine learning MR image reconstruction. *IEEE transactions on Medical Imaging*, 40(9):2306–2317, 2021.

Gregory Ongie, Ajil Jalal, Christopher A. Metzler Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 2020.

Patrick Putzky, Dimitrios Karkalousos, Jonas Teuwen, Nikita Miriakov, Bart Bakker, Matthan Caan, and Max Welling. I-RIM applied to the fastMRI Challenge. *arXiv:1910.08952*, 2019.

Zaccharie Ramzi, Philippe Ciuciu, and Jean-Luc Starck. XPDNet for MRI reconstruction: an application to the 2020 fastMRI Challenge. *arXiv preprint arXiv:2010.07290*, 2020.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

Anne Marie Sawyer, Michael Lustig, Marcus Alley, Phdmartin Uecker, Patrick Virtue, Peng Lai, and Shreyas Vasanawala. Creation of fully sampled MR data repository for compressed sensing of the knee. 2013.

Anuroop Sriram, Jure Zbontar, Tullie Murrell, Aaron Defazio, C. Lawrence Zitnick, Nafissa Yakubova, Florian Knoll, and Patricia Johnson. End-to-end variational networks for accelerated MRI reconstruction. In *Medical Image Computing and Computer Assisted Intervention*, pages 64–73, 2020.

Jian Sun, Huibin Li, Zongben Xu, et al. Deep ADMM-Net for compressive sensing MRI. *Advances in Neural Information Processing Systems*, 29, 2016.

Martin Uecker, Peng Lai, Mark J Murphy, Patrick Virtue, Michael Elad, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. *Magnetic Resonance in Medicine*, 71(3):990–1001, 2014.

Dayang Wang, Zhan Wu, and Hengyong Yu. TED-net: Convolution-free T2T VisionTransformer-based encoder-decoder dilation network for low-dose CT denoising. In *International Workshop on Machine Learning in Medical Imaging*, pages 416–425, 2021a.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid Vision Transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021b.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general U-shaped Transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021c.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

Yixuan Wu, Kuanlun Liao, Jintai Chen, Danny Z Chen, Jinhong Wang, Honghao Gao, and Jian Wu. D-Former: A U-shaped dilated Transformer for 3D medical image segmentation. *arXiv preprint arXiv:2201.00462*, 2022.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient Transformer for high-resolution image restoration. *arXiv:2111.09881*, 2021.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: An open dataset and benchmarks for accelerated MRI. *arXiv:1811.08839*, 2019.

Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1828–1837, 2018.

Lipei Zhang, Zizheng Xiao, Chao Zhou, Jianmin Yuan, Qiang He, Yongfeng Yang, Xin Liu, Dong Liang, Hairong Zheng, Wei Fan, et al. Spatial adaptive and Transformer fusion network (STFNet) for low-count PET blind denoising with MRI. *Medical Physics*, 2021.

Hong-Yu Zhou, Jiansen Guo, Yinghao Zhang, Lequan Yu, Liansheng Wang, and Yizhou Yu. nnFormer: Interleaved Transformer for volumetric segmentation. *arXiv preprint arXiv:2109.03201*, 2021.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested U-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Conclusions.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We were unable to identify possible negative effects on society.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Full code is available in the supplementary. Source code will be published upon paper acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Supplementary
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the standard error of runs in our tables.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplementary.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We attach source code in the Supplementary.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A HUMUS-Net baseline details

Our default model has 3 RSTB-D downsampling blocks, 2 RSTB-B bottleneck blocks and 3 RSTB-U upsampling blocks with $3-6-12$ attention heads in the D/U blocks and 24 attention heads in the bottleneck block. For Swin Transformers layers, the window size is 8 for all methods and MLP ratio ($hidden_dim/input_dim$) of 2 is used. Each RSTB block consists of 2 STLs with embedding dimension of 66. We use 8 cascades of unrolling with a U-Net as sensitivity map estimator (same as in E2E-VarNet) with 16 channels.

We center crop and reflection pad the input images to 384×384 resolution for HUMUS-Net and use the complete images for VarNet. In all experiments, we minimize the SSIM loss between the target image $\boldsymbol{x}^*$ and the reconstruction $\hat{\boldsymbol{x}}$ defined as

$$\mathcal{L}_{SSIM}(\boldsymbol{x}^*, \hat{\boldsymbol{x}}) = 1 - SSIM(\boldsymbol{x}^*, \hat{\boldsymbol{x}}).$$

fastMRI experiments– We train HUMUS-Net using Adam for 50 epochs with a learning rate of 0.0001, dropped by a factor of 10 at epoch 40 and apply adjacent slice reconstruction with 3 slices. We run the experiments on $8\times$ Titan RTX 5000 24GB GPUs with a per-GPU batch size of 1.

Stanford 3D experiments– We train for 25 epochs with a learning rate of 0.0001, dropped by a factor of 10 at epoch 20 and reconstruct single slices. We run the experiments on $4\times$ Quadro RTX 5000 16GB GPUs with a per-GPU batch size of 1.

Stanford 2D experiments– We train for 50 epochs with a learning rate of 0.0001, dropped by a factor of 10 at epoch 40 and reconstruct single slices. Moreover, we crop reconstruction targets to fit into a 384×384 box. We run the experiments on $8\times$ Quadro RTX 5000 16GB GPUs with a per-GPU batch size of 1.

B Ablation study experimental details

Here we discuss the experimental setting and hyperparameters used in our ablation study in Section 5.2. In all experiments, we train the models on the Stanford 3D dataset for 3 different train-validation splits and report the mean and standard error of the results. We use Adam optimizer with learning rate 0.0001 and train for 25 epochs, decaying the learning rate by a factor of 10 at 20 epochs. For all unrolled methods, we unroll 12 iterations in order to provide direct comparison with the best performing E2E-VarNet, which uses the same number of cascades. We reconstruct single slices and do not use the method of adjacent slice reconstruction discussed in 4.4. In models with sensitivity map estimator,we used the default U-Net with 8 channels (same as default E2E-VarNet). For Swin Transformers layers, the window size is 8 for all methods and MLP ratio ($hidden_dim/input_dim$) of 2 is used. Further hyperparameters are summarized in Table 4.

Method	Embedding dim.	# of STLs in RSTB	# attention heads	Patch size
Un-SS	12	$2-2-2$	$6-6-6$	1
Un-MS	12	$D/U:\ 2-2-2,\ B:\ 2$	$D/U:\ 3-6-12,\ B:\ 24$	1
Un-MS-Patch2	36	$D/U:\ 2-2-2,\ B:\ 2$	$D/U:\ 3-6-12,\ B:\ 24$	2
HUMUS-Net	36	$D/U:\ 2-2-2,\ B:\ 2$	$D/U:\ 3-6-12,\ B:\ 24$	1
SwinIR	66	$6-6-6-6$	$6-6-6-6$	1

Table 4: Ablation study experimental details. We show the number of STL layers per RSTB blocks and number of attention heads for multi-scale networks in downsampling (D) , bottleneck (B) and upsampling (U) paths separately.

We run the experiments on $4\times$ Quadro RTX 5000 16GB GPUs with a per-GPU batch size of 1.

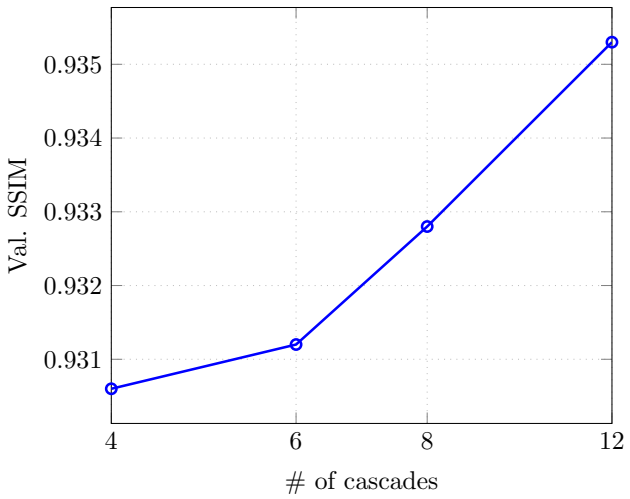


Figure 6: Validation SSIM as a function of number of cascades (unrolled itarations) in HUMUS-Net on the Stanford 3D dataset. We observe a steady increase in reconstruction performance with more cascades.

C Number of unrolled iterations: ablation study

The number of cascades in unrolled networks has a fundamental impact on their performance. Deeper networks typically perform better, but also incur heavy computational and memory cost. In this experiment we investigate the scaling of HUMUS-Net with respect to the number of iterative unrollings.

We perform an ablation study on the Stanford 3D dataset with $\times 8$ acceleration for a fixed training-validation split, where 20% of the training set has been set aside for validation. We reconstruct single slices, without applying adjacent slice reconstruction. We plot the highest SSIM throughout training on the validation dataset for each network on Figure 6.

We observe improvements in reconstruction performance with increasing number of cascades, and this improvement has not saturated yet in the range of model sizes we have investigated. In our main experiments, we select 8 cascades due to memory and compute limitations for deeper models. However, our experiment suggests that HUMUS-Net can potentially obtain even better reconstruction results given enough computational resources.

D Adjacent slice reconstruction: ablation study

In this section, we demonstrate that even though adjacent slice reconstruction is in general helpful, it is not the main reason why HUMUS-Net performs better than E2E-VarNet.

To this end, we add adjacent slice reconstruction to E2E-VarNet and investigate its effect on model performance. We run experiments on the Stanford 3D dataset for 3 different random train-validation splits. We use adjacent slice reconstruction with 3 slices. We have found that the default learning rate for E2E-VarNet is not optimal with the increased input size, therefore we tune the learning rate using grid search and set it to 0.0005. We compare the results with HUMUS-Net, where we match the number of cascades in E2E-VarNet, and use the default embedding dimension of 66. We do not use adjacent slice reconstruction when training HUMUS-Net in order to ablate its effect.

The results are summarized in Table 5. We observe that ASR boosts the reconstruction quality of E2E-VarNet. However, ASR alone cannot close the gap between the two models, as HUMUS-Net without ASR still outperforms the best E2E-VarNet model with ASR.

Model	ASR?	SSIM	PSNR	NMSE
E2E-VarNet	✗	0.9432 ± 0.0063	40.0 ± 0.6	0.0203 ± 0.0006
E2E-VarNet	✓	0.9457 ± 0.0064	40.4 ± 0.7	0.0186 ± 0.0007
HUMUS-Net	✗	$\mathbf{0.9467 \pm 0.0059}$	$\mathbf{40.6 \pm 0.6}$	$\mathbf{0.0178 \pm 0.0005}$

Table 5: Results of the adjacent slice reconstruction ablation study on the Stanford 3D dataset. Mean and standard error over 3 random train-validation splits is shown. ASR improves the performance of E2E-VarNet. However, HUMUS-Net outperforms E2E-VarNet in all cases even without ASR.

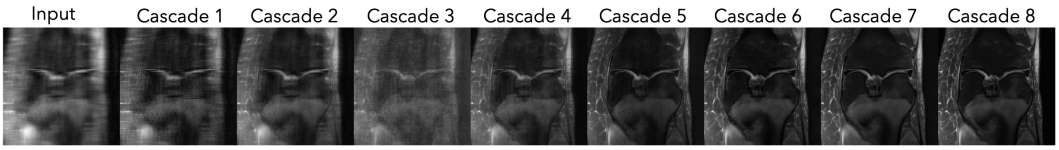


Figure 7: Visualization of intermediate reconstructions in HUMUS-Net.

E Iterative denoising visualization

Here, we provide more discussion on the iterative denoising interpretation of unrolled networks, such as HUMUS-Net. Consider the regularized inverse problem formulation of MRI reconstruction in (2.1) and design the architecture based on unrolling the gradient descent steps on the above problem. Applying data consistency corresponds to the gradient of the first term in (2.2), whereas we learn the gradient of the regularizer parameterized by the HUMUS-Block architecture. This can be conceptualized as alternating between enforcing consistency with the measurements and applying a denoiser based on some prior, represented by the HUMUS-Block in our architecture.

We show an example of a sequence of intermediate reconstructions in order to support the denoising intuition in Figure 7. We plot the magnitude image of reconstructions at the outputs of consecutive HUMUS-Blocks, along with the model input zero-padded reconstruction. We point out that in general the intermediate reconstruction quality (for instance in terms of SSIM) is not necessarily an increasing function of cascades. This is due to the highly non-linear nature of the mapping represented by the neural network, which might not always be intuitive and interpretable.

F Vision Transformer terminology overview

Here we provide a brief overview of the terms related to the Transformer architecture, some of which has been carried over from the Natural Language Processing (NLP) literature. We introduce the key concepts through the Vision Transformer architecture, which has inspired most Transformer-based architectures in computer vision. For more details, we refer the reader to [Dosovitskiy et al., 2020].

Traditional Transformers for NLP receive a sequence of 1D token embeddings. Each token may represent for instance a word in a sentence. To extend this idea to 2D images, the input $x \in \mathbb{R}^{H \times W \times C}$ image (H and W stand for spatial dimensions, C for channel dimension) is split into N patches of size $P \times P$, and each patch is flattened into a 1D vector to produce an input of flattened patches x_f with shape $N \times P^2C$:

$$x_f = \text{PatchFlatten}(x).$$

In order to set the latent dimension of the input entering the Vision Transformer, a trainable linear mapping $E \in \mathbb{R}^{P^2C \times D}$ is applied to the flattened patches, resulting in the so called patch embeddings z of shape $N \times D$:

$$z = x_f E.$$

In order to encode information with respect to the position of image patches, a learnable positional embedding $E_{pos} \in \mathbb{R}^{N \times D}$ is added to the patch embeddings:

$$z_0 = z + E_{pos}.$$

The input to the Transformer encoder is this $N \times D$ representation, which we also refer to in the paper as token representation, as each row in the representation corresponds to a token (in our case an image patch) in the original input.

G Detailed validation results

Dataset	Model	SSIM(↑)	PSNR(↑)	NMSE(↓)
fastMRI	E2E-VarNet	0.8908	36.8	0.0092
	HUMUS-Net	**0.8934**	**37.0**	**0.0090**
Stanford 2D	E2E-VarNet	0.8928 ± 0.0168	$\mathbf{33.9 \pm 0.7}$	0.0339 ± 0.0037
	HUMUS-Net	$\mathbf{0.8954 \pm 0.0136}$	33.7 ± 0.6	$\mathbf{0.0337 \pm 0.0024}$
Stanford 3D	E2E-VarNet	0.9432 ± 0.0063	40.0 ± 0.6	0.0203 ± 0.0006
	HUMUS-Net	$\mathbf{0.9453 \pm 0.0065}$	$\mathbf{40.4 \pm 0.6}$	$\mathbf{0.0187 \pm 0.0009}$

Table 6: Detailed validation results of HUMUS-Net on various datasets. For datasets with multiple train-validation split runs we show the mean and standard error of the runs.

H Additional figures

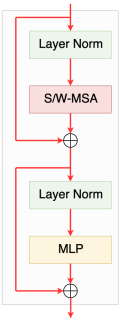


Figure 8: Swin Transformer Layer, the fundamental building block of the Residual Swin Transformer Block.

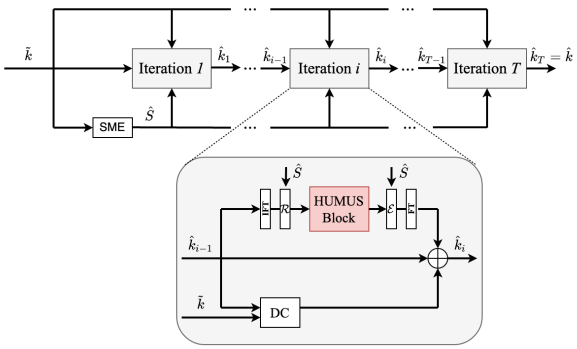


Figure 9: Depiction of iterative unrolling with sensitivity map estimator (SME). HUMUS-Net applies a highly efficient denoiser to progressively improve reconstructions in a cascade of sub-networks.

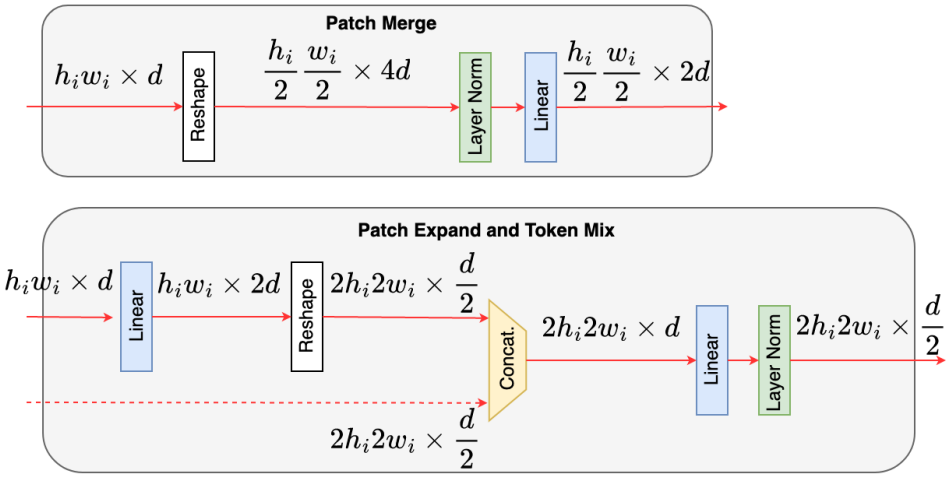
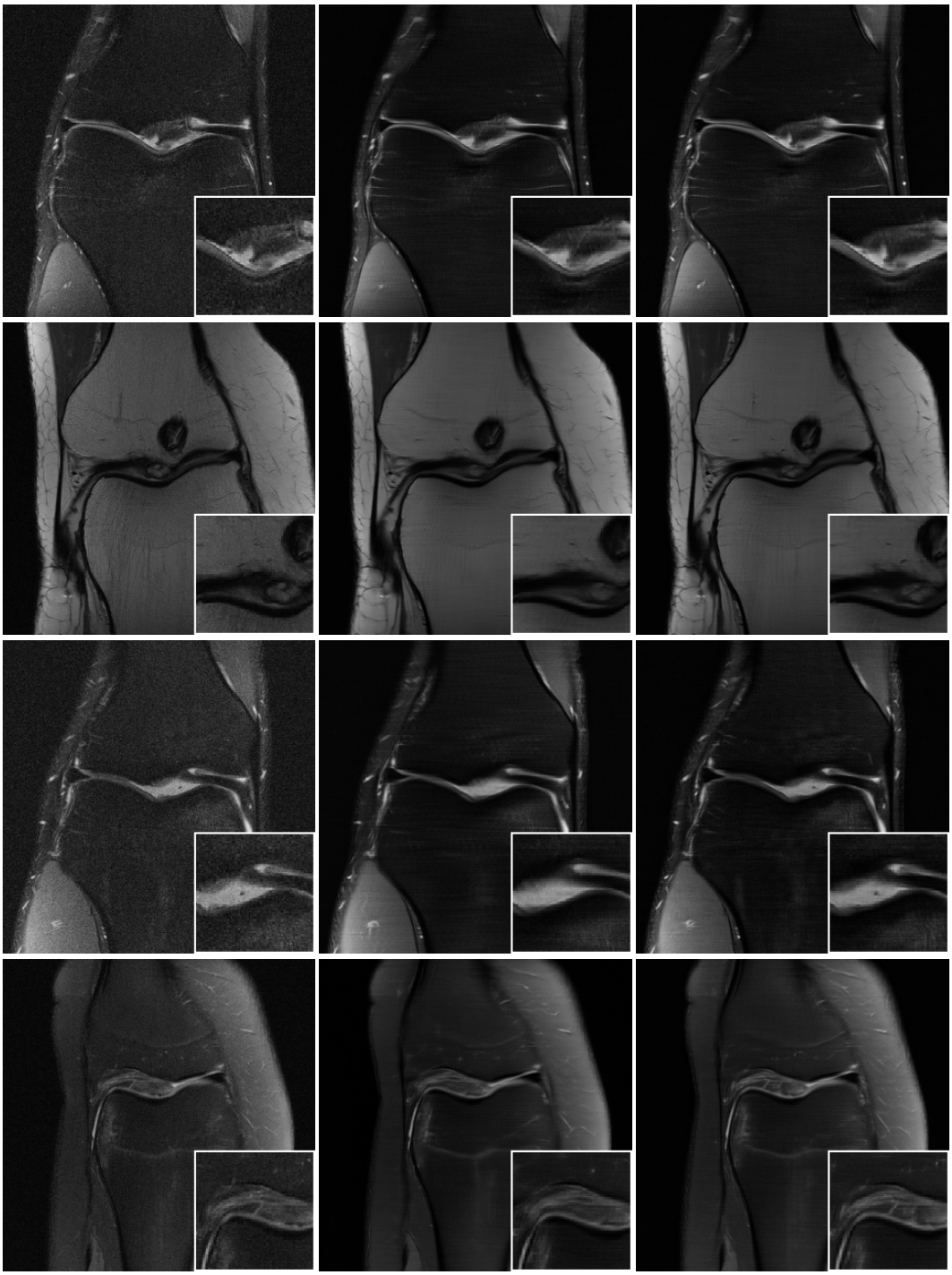


Figure 10: Patch merge and expand operations used in MUST.



Ground truth **E2E-VarNet** **HUMUS-Net (ours)**

Figure 11: Visual comparison of reconstructions from the fastMRI knee dataset with $\times 8$ acceleration. HUMUS-Net reconstructs fine details on MRI images that other state-of-the-art methods may miss.

I Results on additional accelerations

We observe that HUMUS-Net achieves state-of-the-art performance across a wide range of acceleration ratios. We perform experiments on the Stanford 3D dataset using a small HUMUS-Net model with only 6 cascades, embedding dimension of 66, adjacent slice reconstruction with 3 slices and for

the sake of simplicity we removed the residual path from the HUMUS-Block. We call this model *HUMUS-Net-small*. We set the learning rate to 0.0002 and train the model with Adam optimizer. We generate 3 random training-validation splits on the Stanford 3D dataset (same splits as in other Stanford 3D experiments in this paper), and show the mean and standard error of the results in Table .

Acceleration	**Model**	**SSIM(↑)**	**PSNR(↑)**	**NMSE(↓)**
4x	E2E-VarNet	0.9623 ± 0.0038	42.9 ± 0.5	0.0103 ± 0.0001
	HUMUS-Net-small	$\mathbf{0.9640 \pm 0.0040}$	$\mathbf{43.3 \pm 0.5}$	$\mathbf{0.0096 \pm 0.0002}$
8x	E2E-VarNet	0.9432 ± 0.0063	40.0 ± 0.6	0.0203 ± 0.0006
	HUMUS-Net-small	$\mathbf{0.9459 \pm 0.0065}$	$\mathbf{40.4 \pm 0.7}$	$\mathbf{0.0184 \pm 0.0008}$
12x	E2E-VarNet	0.9259 ± 0.0084	37.7 ± 0.7	0.0347 ± 0.0018
	HUMUS-Net-small	$\mathbf{0.9283 \pm 0.0052}$	$\mathbf{38.0 \pm 0.5}$	$\mathbf{0.0298 \pm 0.0019}$

Table 7: Experiments on various acceleration factors on the Stanford 3D dataset. Mean and standard error of 3 random training-validation splits is shown.