Dynamic HUMUS-Net for Fast MRI Reconstruction

Jia-Yao He
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
Email: 202203340104@zjut.edu.cn

Yanlin Chen
American International School
123-129 Waterloo Road, Kowloon City District
HongKong, China
Email: chensunny705@gmail.com

Lei Chen
Shenzhen Leo-King Environmental Co. Ltd
Shenzhen, China
Email: chenleibuct@gmail.com

Xuhua Yang Xiao-Xin Li*
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
Email: {xhyang,mordekai}@zjut.edu.cn

2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 979-8-3503-8622-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/BIBM62325.2024.10821980

***Abstract*—To accelerate magnetic resonance imaging (MRI), image reconstruction from under-sampled measurements has been widely used. Recently, the convolutional-Transformer hybrid architecture has dominated the field of MRI reconstruction. To improve calculation performance, two multi-scale (MS) strategies are usually adopted: the one imposed in the *intra*-cascades in a U-shape style and the one lying in the *inter*-cascades in a pyramid manner. The two MS strategies have their own benefits but have not been combined together for boosting performance. In this work, we proposed a dynamic Hybrid Unrolled Multi-Scale Network (dHUMUS-Net) by incorporating the two MS strategies together. A novel Optimal Scale Prediction Network is presented to dynamically estimate the optimal scales for all cascades of dHUMUS-Net. Experiments on the fastMRI dataset demonstrate the effectiveness of our method over the state-of-the-art methods.**

***Index Terms*—fast MRI, unrolled architecture, multi-scale strategy, dynamic network**

I. Introduction

Magnetic resonance imaging (MRI) is widely accepted as a dominant technique for image-guided radiotherapy due to its better function in providing soft-tissue contrast than computed tomography (CT), while avoiding radiation exposure. However, the physical nature of the MR imaging procedure gives rise to the long scanning time, which seriously affects the patient experience and causes high costs. Therefore, accelerated MRI has become a hot research topic, where reconstructing images from undersampled *k*-space measurements is a standard strategy [1]. Recent researches on MRI reconstruction tends to go in the following three overlapping directions.

First, cascades-based network architecture for MRI reconstruction. Most of the advanced methods [2]–[11] adopt the cascades-based network (CBN) architecture derived from the unrolled optimization of the traditional compressed sensing (CS) problem [12]. A typical CBN model consists of several sequentially cascaded subnets, also called *cascades* [6], each of which ends with a data consistency (DC) layer [2] and acts as a refinement step to the final reconstruction [13]. The DC layers are very important to avoid losing or corrupting the sampled *k*-space data in the inputs after long-distance forward mapping. Therefore, the cascades in DUAs are usually very small, focusing only on the local features and having limited receptive fields [14].

Second, Transformer-based MRI reconstruction. Image reconstruction models struggle to enlarge receptive fields to fully capture larger context information for more faithful data recovery. To this end, convolutional neural networks (CNNs) [2], [5], [6], [8], [10] require stacking very deep layers without the DC-like interruption during feature forward mapping. This, however, contradicts with the principle of the CBN architecture. The efforts of using dilated convolutions [15], [16] and cross-cascade connection [8] can only achieve very limited enhancement of receptive fields. With the emergence of vision Transformers [17], [18], the limited receptive-field issue of the CBN architecture can be well addressed [11], [14], [19]–[22] by the modeling power of Transformers in long-range dependency.

Combining the advanced models in the above two subfields has become main solutions for MRI reconstruction. HUMUS-Net [19] and ReconFormer [11] are two representative methods. To improve calculation performance, both HUMUS-Net and ReconFormer adopt the multi-scale strategy. The multi-scale strategy used in HUMUS-Net [19] lies in the *intra*-cascades in a U-shape style, while the multi-scale strategy used in ReconFormer [11] lies in the *inter*-cascades in a pyramid manner. The two multi-scale strategies have their own benefits but have not been combined together for Transformer-based MRI reconstruction.

In this work, we redesign the HUMUS-Net by incorporating the two multi-scale strategies and propose a **d**ynamic **H**ybrid **U**nrolled **Mu**lti-**S**cale Network (dHUMUS-Net) for accelerated MRI reconstruction. The main challenge of using the pyramid structure is how to estimate the optimal scale for each of the cascades. ReconFormer manually sets the scales for all

Corresponding author: Xiao-Xin Li. This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LGF22F020027, in part by the National Natural Science Foundation of China under Grant 62373324 and 62271448.

cascades. This might lead to sub-optimal performance when the acceleration rate or the dataset varies. We show that the optimal scale of a cascade mainly depends on the *repetition level* (RL) of the input image, and a large RL tends to require a large downsampling scale. We give a novel Optimal Scale Prediction Network (OSPN) to estimate the optimal scale of each cascade as per the RL of the input image. By putting OSPN ahead each cascade of HUMUS-Net, dHUMUS-Net can be constructed dynamically. To the best of our knowledge, this is the first attempt to design a dynamic unrolled structure for MRI reconstruction. We show through experiments on the fastMRI dataset that dHUMUS-Net yields higher fidelity reconstructions.

II. Problem Formulation

An MRI scanner obtains measurements of the patient anatomy in *k*-space via various receiver coils. The fully sampled *k*-space data can be obtained via $\mathrm{y} = \mathcal{A}(\mathrm{x})$, where $\mathrm{x} \in \mathbb{C}^n$ is the underlying patient anatomy of interest and usually has a very high dimension, and $\mathcal{A}$ is the linear forward operator [23] that first multiplies by the sensitivity maps and then applies 2D Fourier Transform (FT). Note that for simplicity, the measurement noise in the forward mapping is omitted. The anatomy image x can be recovered by $\mathrm{x} = \mathcal{A}^*(\mathrm{y})$, where $\mathcal{A}^*$ first applies 2D Inverse Fourier Transform (IFT) and then uses the reduce operator [6] to combine all individual coil images.

To accelerate MRI, only partial *k*-space data y_u is acquired $\mathrm{y}_u = M_u\mathcal{A}(\mathrm{x})$,where M_u is a diagonal matrix representing a binary undersampling mask for $u\times$ acceleration. As y_u is highly undersampled, directly applying $\mathcal{A}^*$ on y_u will lead to a highly aliased reconstruction $\mathrm{x}_u = \mathcal{A}^*(\mathrm{y}_u)$. Deep unrolled architectures (DUAs) [1], [11], [13], [19], [20] are widely used to further perform reconstruction from x_u. However, x_u is usually high dimensional and can lead to high compute cost.

Fortunately, we observe that x_u is *highly compressible*. This is because x_u is obtained from the zero-filled undersampled *k*-space data via using IFT. After IFT, each zero value in *k*-space can spread over all pixels of x_u and thus results in a lot of repeated features in image domain. Such a compressibility can extend to the inputs of the intermediate cascades. In this work, we will explore how to use the compressibility of the aliased input images and the multi-scale strategies of the intra- and inter-cascades to boost the reconstruction performance and efficiency of the Transformer-based DUAs.

III. Method

The network architecture of the proposed **d**ynamic HUMUS-Net (dHUMUS-Net) is illustrated in Fig. 1. dHUMUS-Net consists of T cascades. The main process of each cascade, e.g., Cascade t, can be represented as follows

$$S_t = \mathrm{OSPN}\left(\tilde{\mathrm{x}}_{t-1}\right) \tag{1}$$

$$\hat{\mathrm{x}}_t = \tilde{\mathrm{x}}_{t-1} + \sum_{s\in S_t} \mathrm{dHMUST}_t\left(\tilde{\mathrm{x}}_{t-1}, s\right), \tag{2}$$

where $\tilde{\mathrm{x}}_{t-1} = \mathcal{A}^*\left(\tilde{\mathrm{y}}_{t-1}\right)$ is the reduced reconstruction result of Cascade $t-1$, S_t the set of the possible scales of $\tilde{\mathrm{x}}_{t-1}$ and can be predicted by the proposed Optimal Scale Prediction Network (OSPN), and dHMUST is the main dynamic module and its structure is dynamically determined by the scale of the input data $\tilde{\mathrm{x}}_{t-1}$. We next illustrate dHMUST and OSPN, respectively.

A. Dynamic HMUST (dHMUST)

dHMUST in Cascade t can be defined as follows

$$\mathrm{dHMUST}_t\left(\tilde{\mathrm{x}}_{t-1}, s\right) \triangleq \begin{cases} \mathrm{HMUST}_{t,1}\left(\tilde{\mathrm{x}}_{t-1}\right) & \text{if } s = 1 \\ \mathrm{HMUST}_{t,2}\left(\tilde{\mathrm{x}}_{t-1}\right) & \text{if } s = 2 \\ \mathrm{HMUST}_{t,4}\left(\tilde{\mathrm{x}}_{t-1}\right) & \text{if } s = 4 \\ \mathrm{HMUST}_{t,8}\left(\tilde{\mathrm{x}}_{t-1}\right) & \text{if } s = 8 \end{cases}. \tag{3}$$

Here $\mathrm{HMUST}_{t,s}$ denotes the HMUST (**H**ybrid **Mu**lti-scale residual **S**win **T**ransformer) module with the max-scale s in Cascade t.

The main process of $\mathrm{HMUST}_{t,s}$ following the definition of HUMUS-Block defined in HUMUS-Net [19]. HUMUS-Block mainly consists of a high-dimension-feature extractor $\mathcal{H}$, a low-dimension-feature extractor $\mathcal{L}$, a deep-and-low-dimension-feature extractor, namely **Mu**lti-scale residual **S**win **T**ransformer (MUST), and a reconstruction operator $\mathcal{R}$. Note that $\mathcal{H}$, $\mathcal{L}$ and $\mathcal{R}$, whereas MUST is a Transformer-convolutional hybrid module. Our HMUST module is formulated by equipping the scale on HUMUS-Block and can be represented as follows

$$\mathsf{H}_{t,s} = \mathcal{H}_{t,s}\left(\tilde{\mathrm{x}}_{t-1}\right) \tag{4}$$

$$\mathsf{L}_{t,s} = \mathcal{L}_{t,s}\left(\mathsf{H}_{t,s}\right) \tag{5}$$

$$\mathsf{D}_{t,s} = \mathrm{MUST}_{t,s}\left(\mathsf{L}_{t,s}\right) \tag{6}$$

$$\mathrm{r}_{t,s} = \mathcal{R}_{t,s}\left(\mathsf{H}_{t,s}, \mathsf{D}_{t,s}\right) \tag{7}$$

where the subscript t, s is used to denote the outputs or the operators used in Cascade t under scale s.

Note that $\mathcal{H}_{t,s}, \mathcal{L}_{t,s}, \mathcal{R}_{t,s}$ are convolutional blocks (ConvBlocks) and their definitions are independent of the scale s. The scale s in the subscript is just used to indicate that the $\mathcal{H}$,$\mathcal{L}, \mathcal{R}$ modules against the scale s are different from those against distinguishing scales. The scale of $\mathrm{HMUST}_{t,s}$ is mainly determined by the max-scale of the adopted MUST module, which can only be 2^n (n is a non-negative integer) due to the definition of the downsampling operator, PatchMerge [19], used in its encoder path. In Eq. (3), we limit the candidate max-scales to be in the set $\mathcal{S}_{\mathrm{MS}} = \{1, 2, 4, 8\}$. The max-scales in $\mathcal{S}_{\mathrm{MS}}$ is set to be ≤ 8 because for the highest $8\times$ acceleration rate used in this work, setting max-scales ≤ 8 is enough to decompose the redundant information in the input image.

B. Optimal Scale Prediction Network (OSPN)

Given an input image, OSPN uses its RL to estimate the set of the optimal downsampling scales that can well adapt to the input data and can be used by dHMUST to dynamically create or choose a HMUST branch. The mapping relationship between the RL and the optimal scale is non-trivial. To model

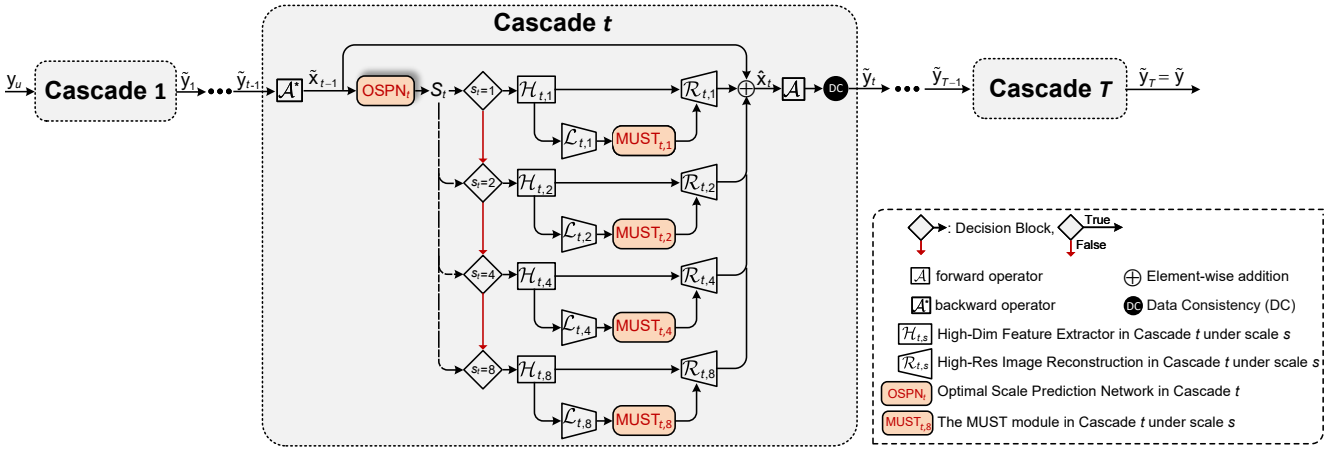

Fig. 1. Overview of the proposed **d**ynamic **H**ybrid **U**nrolled **Mu**lti-**S**cale Network (dHUMUS-Net). The main module is the dHMUST, i.e., the dynamic HMUST (**H**ybrid **Mu**lti-scale residual **S**win **T**ransformer), which consists of several HMUST-based branches, and will be dynamically constructed during training according to the output of OSPN.

the mapping from v_{RL} to the optimal scale s, we use the learning method to build the OSPN model.

Suppose we have a set of training samples $\Gamma = \left\{ \left(\mathrm{x}_u^{(i)}, \mathrm{x}^{(i)} \right) \Big|_{u=r/2}^{r} \right\}_{i=1}^{N}$, where r is the target acceleration rate, and the acceleration rates of the training samples range from $r/2$ to r to ensure that the possible RLs of the outputs of all cascades can be covered as far as possible. For training, we should further quantify the RL and label the optimal scale for each of the training sample in Γ.

Quantify the RL. As the repeated features are globally distributed in the whole image, we decompose the input image into several sub-images by using the pixel-unshuffle (PU) operator [24] and measure the RL by using the SSIM-based similarities between all PU outputs. Given a PU-factor s ($s \geq 2$), the PU operator can produce s^2 outputs and thus leads to a similarity vector v_s consisting of $\frac{1}{2}s^2\left(s^2-1\right)$ values. Given a PU-factor set $\mathcal{S}_{\mathrm{PU}}$, we represent the RL of an input image as follows

$$v_{\mathrm{RL}} = \left[v_2^\top, v_3^\top, \cdots, v_{|\mathcal{S}_{\mathrm{PU}}|}^\top \right]^\top. \tag{8}$$

Label the optimal scales. We first train the four HMUST modules, defined in (3), respectively, by using the training set Γ. We then label each of the training samples as follows

$$\left[s_1^{(i)}, s_2^{(i)} \right] = \arg\operatorname{sort}_{s \in \mathcal{S}_{\mathrm{MS}}} \mathrm{SSIM}\left(\mathrm{dHMUST}\left(\mathrm{x}_u^{(i)}, s \right), \mathrm{x}^{(i)} \right), \tag{9}$$

where SSIM (Structural Similarity Index Measure) is used to measure the reconstruction performance, and $\mathrm{sort}(\cdot)$ is used to sort the SSIM evaluation values. Note that rather than using the top-1 optimal scale causing the maximum SSIM values, we choose the the the top-2 scales leading to the top two optimal SSIM metrics as the scale labels of the input data $\mathrm{x}_u^{(i)}$. This is mainly because only using the top-1 scale can lead to unstable performance in experiments.

We now have the final training set $\left\{ \left(v_{\mathrm{RL}}^{(i)}, \left[s_1^{(i)}, s_2^{(i)} \right] \right) \right\}_{i=1}^{N\cdot(r/2+1)}$. As the similarity vectors

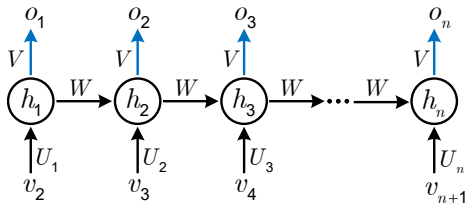

Fig. 2. Architecture of the proposed optimal scale estimation Network (OSPN), where $n = |\mathcal{S}_{\mathrm{MS}}| - 1$.

v_s in $v_{\mathrm{RL}}^{(i)}$ can be considered as a 1D time series, we use an RNN (Recurrent Neural Network) to model the mapping from v_{RL} to $\left[s_1^{(i)}, s_2^{(i)} \right]$.

RNN-based OSPN. We formulate the OSPN by a Recurrent Neural Network (RNN) as shown in Fig. 2.

Formally, we have

$$\begin{aligned} h_s &= \mathrm{ReLU}\left(U_s \cdot v_{s+1} + W \cdot h_{s-1} \right) \\ o_s &= \mathrm{Sigmoid}\left(V \cdot h_s \right) \end{aligned}$$

where $s \in \mathcal{S}_{\mathrm{PU}}$, $h_0 = 0$, $\left\{ U_1, U_2, \cdots, U_{|\mathcal{S}|}, W, V \right\}$ is the parameter set, and o_s is the predicted confidence level under which the optimal scale is set to be s. OSPN can be optimized by minimizing the cross-entropy (CE) loss

$$\min_{\Theta} \sum_{i \in \mathcal{D}} \ell_{\mathrm{CE}}\left(f_{\mathrm{PUSN}}\left(v_{\mathrm{RL}}^{(i)}; \Theta \right), s_{\mathrm{Opt}}^{(i)} \right),$$

where $\mathcal{D}$ is the training set $\left\{ \left(v_{\mathrm{RL}}^{(i)}, s_{\mathrm{Opt}}^{(i)} \right) \right\}_{i=1}^{N\cdot(r/2+1)}$.

The design of dMUST optimizes the scale configuration but will lead to the difficulty of mini-batch-based training. See Section IV for more details.

IV. Network Training

We now consider how to train the dHUMUS-Net. We construct a set of training samples $\mathbf{\Gamma} = \left\{ \left(\mathrm{x}_u^{(i)}, \mathrm{x}^{(i)}, \left[s_1^{(i)}, s_2^{(i)} \right] \right) \right\}_{i=1}^{N}$, where u is the target

acceleration rate. We choose normalized mean square error (NMSE) as the loss function:

$$\frac{1}{|\Gamma|}\sum_{(\mathrm{x}_u,\mathrm{x})\in\Gamma}\frac{1}{\|\mathrm{x}\|_2}\left\|\text{dHUMUS-Net}\left(\mathrm{x}_u;\Theta\right)-\mathrm{x}\right\|_2. \tag{10}$$

Training is carried out by minimizing the loss function (10) using mini-batch gradient descent. However, different inputs might have different optimal scales, leading to different structures of the cascades of our dHUMUS-Net, and forbidding randomly choosing training samples to form a mini-batch. Fortunately, we experimentally found that training samples usually have stable *optimal scale trajectories*. As each sample has an optimal scale for each cascade, the sequence of these optimal scales can be considered as a *trajectory*. One can verify that training samples with the same optimal scale in the first cascade tend to have identical optimal scale trajectories.

We therefore forms the mini-batches according to the optimal scales. Specifically, for each of the optimal scale $\left[s_1^{(i)}, s_2^{(i)}\right] \in \mathcal{S}_{\mathrm{MS}} = \{1, 2, 4, 8\}$, we create a mini-batch $\mathcal{B}_s$ and split $\mathcal{B}_s$ into two subsets $\mathcal{B}_s^{(1)}$ and $\mathcal{B}_s^{(2)}$, where $\mathcal{B}_s^{(1)}$ consists of the training samples with the optimal scales whereas $\mathcal{B}_s^{(2)}$ is composed of the training samples with scales larger than the optimal ones. Moreover, we set $\left|\mathcal{B}_s^{(1)}\right| : \left|\mathcal{B}_s^{(2)}\right| = 7:3$ if s is not the largest optimal scale in $\mathcal{S}_{\mathrm{MS}}$; otherwise, we set $\left|\mathcal{B}_s^{(1)}\right| : \left|\mathcal{B}_s^{(2)}\right| = 1:0$. For training with $\mathcal{B}_s$, we use a fixed optimal scale trajectory, which is the one labeled for the sample randomly selected from $\mathcal{B}_s^{(1)}$.

By combining the above loss function and the mini-batch training strategy with the training setting of a specific DCN, we can fully use the training data to train our dHUMUS-Net efficiently.

V. Experimental Results

In this section, we performed extensive experiments on publicly available datasets to comprehensively evaluate the effectiveness of our method.

A. Experimental Setting and Implementation Details

We utilized the knee datasets from the fastMRI challenge [23] for our experiments. This dataset comprises two MRI modalities: proton-density weighting with (PDFS) and without (PD) fat suppression. It includes both single-coil and multi-coil tasks, with 973 volumes (34,742 slices) for training and 199 volumes (7,135 slices) for validation. The raw *k*-space data have matrix sizes of 640×368 or 640×372.

We focused on acceleration rates (AR) of 4×, 8×, and 16×, employing equispaced undersampling masks for their practical implementability in MRI scanners [6]. The undersampled *k*-space data and zero-filled images were generated according to the process described in Section II.

For a comprehensive evaluation, we utilized data from multiple MRI scanners: Skyra, Biograph, Prisma, and Aera. This diverse dataset allows us to assess the robustness and generalizability of our method across different imaging systems.

Our model was implemented using PyTorch and trained on NVIDIA Tesla V100-PCIE GPUs. We used the Adam optimizer with an initial learning rate of 1e-4, which was reduced by a factor of 0.1 every 40 epochs. The model was trained for a total of 60 epochs with a batch size of 5. For evaluation metrics, we used the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR). The training was conducted on a system with 8 Tesla V100-PCIE GPUs, each with 32GB of memory, running CUDA version 11.4 and NVIDIA driver version 470.82.01.

B. Ablation Studies

Two scale configuration strategies for dHUMUS-Net should be compared: fixed scale (FS) and OSPN-based dynamic scale (DS). For FS, we choose three max-scales 2, 4, and 8 from $\mathcal{S}_{\mathrm{MS}}$. For DS, we need to consider the mini-batch generation strategy, which suggests that the samples in the two parts, $\mathcal{B}_s^{(1)}$ and $\mathcal{B}_s^{(2)}$, of a mini-batch $\mathcal{B}_s$ should satisfy a certain ratio. We use OSPN($\mathcal{B}_s^{(1)}$:$\mathcal{B}_s^{(2)}$), specifically, OSPN(1:0) and OSPN(7:3), to represent how the mini-batch generation strategy cooperates with OSPN.

Table I reports the comparison results. For FC, we can see that max-scale=4, rather than max-scale=2 or max-scale=8, leads to the optimal performance for AR=4. It means that manually setting the max-scales has the risk to lead to sub-optimal performance. Using OSPN(1:0) can overcome this issue and lead to the optimal performance compared to fixed max-scales. Furthermore, using OSPN(7:3) outperforms using OSPN(1:0) and can achieve the best performance under all acceleration rates. These results collectively highlight the dynamic design of our proposed methods, demonstrating its crucial role in achieving high-quality MRI reconstruction.

C. Comparison with Existing Methods

To validate the effectiveness of our method, we conducted extensive comparisons against state-of-the-art methods, including DnCn [2], *k*SPCN [25], ReconFormer [11], and HUMUS-Net [19], under the same experimental setting. Table II presents the quantitative evaluation results of our comparisons for acceleration rates of 4×, 8×, and 16×.

Our dHUMUS-Net demonstrated superior performance, surpassing competitors in both PSNR and SSIM metrics across all acceleration rates. At 4× acceleration, dHUMUS-Net achieved the highest average PSNR and SSIM, significantly outperforming the suboptimal method, HUMUS-Net. This superiority continues at higher acceleration rates, dHUMUS-Net maintains its superior performance at 8× and 16× acceleration rates, respectively.

Notably, our approach not only achieved optimal SSIM and PSNR values but also maintained competitive training times, as shown in the last column of Table II. The computational efficiency of dHUMUS-Net can be attributed to the design of dynamic scale prediction, where the optimal scales are used to adapt to the data, thus saving the network capacities and computation resources in most of the cascades.

TABLE I
ABLATION STUDY RESULTS

Max-Scale	AR=4		AR=8		AR=16	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2	40.53±0.016	0.9562±0.009	38.53±0.009	0.9431±0.015	35.72±0.007	0.9147±0.016
4	41.12±0.017	0.9662±0.023	38.32±0.017	0.9397±0.024	35.59±0.025	0.9134±0.014
8	40.39±0.021	0.9551±0.240	39.14±0.026	0.9513±0.018	36.39±0.017	0.9194±0.026
OSPN(1:0)	41.16±0.016	0.9669±0.190	39.21±0.014	0.9524±0.013	36.5±0.024	0.9203±0.027
OSPN(7:3)	41.29±0.012	0.9689±0.012	39.35±0.015	0.9546±0.015	36.61±0.016	0.9216±0.017

TABLE II
COMPARISON WITH EXISTING METHODS

AR	Methods	Aera		Biograph		Prisma		Skyra		Avg		Training hours
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
4	D5C5	35.97±0.015	0.9154±0.017	37.37±0.018	0.9253±0.017	37.43±0.015	0.9272±0.022	37.21±0.025	0.9228±0.031	36.83±0.014	0.9196±0.013	22.96
	kSPCN	36.24±0.021	0.9176±0.024	37.66±0.015	0.9302±0.018	37.69±0.008	0.9309±0.017	37.47±0.008	0.9277±0.028	37.10±0.025	0.9233±0.024	21.74
	ReconFormer	38.81±0.027	0.9431±0.023	41.97±0.009	0.9695±0.015	41.69±0.014	0.9635±0.013	41.59±0.017	0.9623±0.015	40.46±0.021	0.9554±0.019	97.44
	HUMUS-Net	39.38±0.013	0.9502±0.019	42.50±0.016	0.9733±0.014	42.17±0.019	0.9699±0.021	41.91±0.022	0.9639±0.016	40.97±0.013	0.9605±0.011	88.56
	dHUMUS-Net	**39.71±0.019**	**0.9582±0.011**	**44.81±0.023**	**0.9858±0.019**	**44.55±0.021**	**0.9837±0.025**	**42.72±0.016**	**0.9761±0.025**	**41.29±0.012**	**0.9689±0.012**	50.12
8	D5C10	34.69±0.014	0.8953±0.026	37.06±0.014	0.9265±0.026	37.05±0.026	0.9264±0.016	35.89±0.017	0.9127±0.014	35.53±0.018	0.9062±0.025	30.36
	kSPCN	35.11±0.026	0.9041±0.015	37.48±0.017	0.9277±0.015	37.44±0.018	0.9273±0.022	36.29±0.018	0.9182±0.021	35.94±0.017	0.9128±0.022	25.72
	ReconFormer	36.68±0.014	0.9225±0.027	40.31±0.021	0.9553±0.026	40.23±0.009	0.9547±0.013	38.75±0.021	0.9419±0.016	38.03±0.024	0.9345±0.029	124.98
	HUMUS-Net	37.27±0.018	0.9246±0.013	40.86±0.023	0.9594±0.011	40.71±0.025	0.9571±0.008	39.42±0.019	0.9503±0.022	38.64±0.012	0.9394±0.017	118.08
	dHUMUS-Net	**37.94±0.020**	**0.9361±0.024**	**41.62±0.015**	**0.9624±0.023**	**41.41±0.016**	**0.9607±0.011**	**40.17±0.016**	**0.9533±0.015**	**39.35±0.015**	**0.9546±0.015**	54.92
16	D5C15	31.37±0.024	0.8784±0.018	31.39±0.009	0.8791±0.014	31.57±0.021	0.8804±0.024	31.13±0.025	0.8736±0.028	31.32±0.011	0.8767±0.026	37.26
	kSPCN	31.98±0.016	0.8846±0.025	32.05±0.012	0.8856±0.019	32.20±0.012	0.8883±0.017	31.74±0.017	0.8821±0.019	31.95±0.022	0.8838±0.015	28.58
	ReconFormer	34.19±0.019	0.8958±0.013	36.93±0.010	0.9261±0.028	37.16±0.015	0.9294±0.013	35.60±0.024	0.9135±0.015	35.17±0.007	0.9069±0.014	161.04
	HUMUS-Net	34.87±0.025	0.9011±0.012	37.69±0.016	0.9307±0.017	37.71±0.012	0.9311±0.021	36.43±0.013	0.9204±0.013	35.91±0.012	0.9127±0.013	142.92
	dHUMUS-Net	**35.49±0.028**	**0.9107±0.023**	**38.34±0.025**	**0.9391±0.019**	**38.39±0.026**	**0.9399±0.016**	**37.13±0.020**	**0.9287±0.023**	**36.61±0.016**	**0.9216±0.017**	60.28

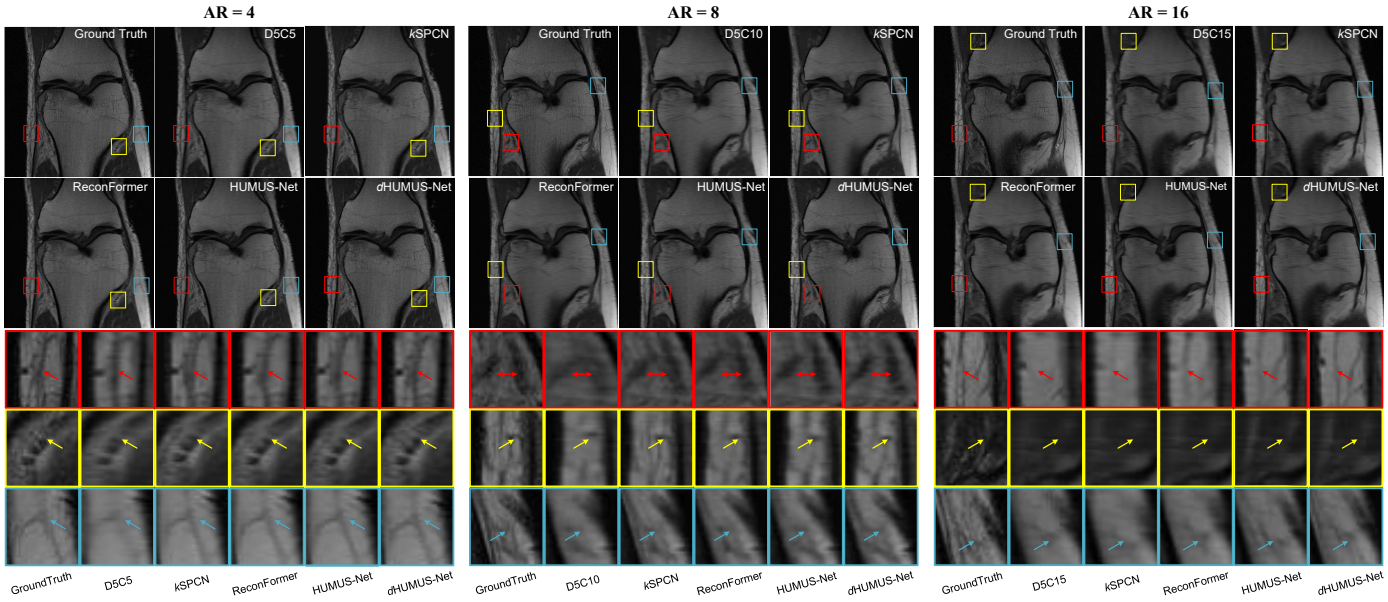

Fig. 3. Visual comparison of reconstruction results at different acceleration rates. The first two rows shows the reconstruction results of the compared methods. The bottom three rows present the zoomed-in views of selected regions (highlighted by colored boxes) for detailed comparison.

Fig. 3 provides a visual comparison of the reconstruction results from different methods against the three acceleration rates 4×, 8×, and 16×. As can be seen, our dHUMUS-Net can restore more fine-scaled information and anatomy structures than the other methods. Also, dHUMUS-Net can well remove the aliasing artifacts produced by HUMUS-Net.

Both the quantitative and qualitative evaluation results show that the performance improvements are consistent across different MRI scanners (Aera, Biograph, Prisma, and Skyra), demonstrating the robustness and generalizability of our method. Also, the results highlight the superior reconstruction quality and efficiency of our dHUMUS-Net, particularly at high acceleration rates, making it a promising solution for accelerated MRI reconstruction in clinical settings.

VI. Conclusions

We provide a dHUMUS-Net to resolve the high dimension and high repetition level (RL) issues in MRI reconstruction by incorporating the intra- and inter-cascade multi-scale strategies into the unrolled Transformer-convolutional hybrid architecture. OSPN and dHMUST are designed for dynamic module selection according to the RL of the input data. Experiments on the fastMRI dataset demonstrated the effectiveness of our method. Future work will focus on better design of the dHMUST and make dHMUST can be configured with more max-scales beyond 2^n so that the model can better adapt to data. We will also study the adaption of OSPN on other well-known convolution-based [5], [6], [10], [26] and Transformers-based [11], [20] unrolled architectures.

References

[1] Y. Chen, C.-B. Schönlieb, P. Lio, T. Leiner, P. L. Dragotti, G. Wang, D. Rueckert, D. Firmin, and G. Yang, "AI-based reconstruction for fast MRI–a systematic review and meta-analysis," *Proceedings of the IEEE*, vol. 110, no. 2, pp. 224–245, 2022.

[2] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for dynamic MR image reconstruction," *IEEE transactions on Medical Imaging*, vol. 37, no. 2, pp. 491–503, 2018.

[3] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, "Learning a variational network for reconstruction of accelerated MRI data," *Magnetic resonance in medicine*, vol. 79, no. 6, pp. 3055–3071, 2018.

[4] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1828–1837.

[5] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 521–538, 2020.

[6] A. Sriram, J. Zbontar, T. Murrell, A. Defazio, C. L. Zitnick, N. Yakubova, F. Knoll, and P. Johnson, "End-to-end variational networks for accelerated MRI reconstruction," in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*. Cham: Springer International Publishing, 2020, pp. 64–73.

[7] S. Wang, H. Cheng, L. Ying, T. Xiao, Z. Ke, H. Zheng, and D. Liang, "DeepcomplexMRI: Exploiting deep residual network for fast parallel MR imaging with complex convolution," *Magnetic Resonance Imaging*, vol. 68, pp. 136–147, 2020.

[8] X.-X. Li, Z. Chen, X.-J. Lou, J. Yang, Y. Chen, and D. Shen, "Multimodal MRI acceleration via deep cascading networks with peer-layer-wise dense connections," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 329–339.

[9] D. You, J. Xie, and J. Zhang, "ISTA-Net++: Flexible deep unfolding network for compressive sensing," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2021, pp. 1–6.

[10] E. Z. Chen, P. Wang, X. Chen, T. Chen, and S. Sun, "Pyramid convolutional RNN for MRI image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 41, no. 8, pp. 2033–2047, 2022.

[11] P. Guo, Y. Mei, J. Zhou, S. Jiang, and V. M. Patel, "ReconFormer: Accelerated MRI econstruction using recurrent Transformer," *IEEE transactions on medical imaging*, vol. 43, no. 1, pp. 582–593, 2023.

[12] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: the application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[13] D. Liang, J. Cheng, Z. Ke, and L. Ying, "Deep magnetic resonance image reconstruction: Inverse problems meet neural networks," *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 141–151, 2020.

[14] D. Hu, Y. Zhang, J. Zhu, Q. Liu, and Y. Chen, "TRANS-Net: Transformer-enhanced residual-error alternative suppression network for MRI reconstruction," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[15] B. Zhou and S. K. Zhou, "DuDoRNet: Learning a dual-domain recurrent network for fast MRI reconstruction with deep t1 prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4273–4282.

[16] Y. Ren, W. Jiang, and Y. Liu, "A complex-valued dual-domain dilated convolution neural network for brain mri reconstruction," in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022, pp. 1144–1149.

[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision Transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[18] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin Transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1833–1844.

[19] Z. Fabian, B. Tinaz, and M. Soltanolkotabi, "HUMUS-Net: Hybrid unrolled multi-scale network architecture for accelerated MRI reconstruction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 306–25 319, 2022.

[20] J. Huang, Y. Fang, Y. Wu, H. Wu, Z. Gao, Y. Li, J. Del Ser, J. Xia, and G. Yang, "Swin Transformer for fast MRI," *Neurocomputing*, vol. 493, pp. 281–304, 2022.

[21] C.-M. Feng, Y. Yan, G. Chen, Y. Xu, Y. Hu, L. Shao, and H. Fu, "Multimodal transformer for accelerated mr imaging," *IEEE Transactions on Medical Imaging*, vol. 42, no. 10, pp. 2804–2816, 2023.

[22] B. Zhou, N. Dey, J. Schlemper, S. S. M. Salehi, C. Liu, J. S. Duncan, and M. Sofka, "DSFormer: A dual-domain self-supervised Transformer for accelerated multi-contrast MRI reconstruction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4966–4975.

[23] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno *et al.*, "fastMRI: An open dataset and benchmarks for accelerated MRI," *arXiv preprint arXiv:1811.08839*, 2018.

[24] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1905–1914.

[25] X.-X. Li, W.-H. Zheng, Q. Zhou, H. Hu, and L. Chen, "Deep k-space partition-based convolutional networks for fast multimodal mri reconstruction," in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2023, pp. 1280–1285.

[26] G. Yiasemis, J.-J. Sonke, C. SÃ¡nchez, and J. Teuwen, "Recurrent variational network: A deep learning inverse problem solver applied to the task of accelerated MRI reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 722–731.