UFC-Net: Unrolling Fixed-point Continuous Network for Deep Compressive Sensing

Xiaoyang Wang
Northwestern Polytechnical University
Xi'an 710072, China
wxyang@mail.nwpu.edu.cn

Hongping Gan*
Northwestern Polytechnical University
Xi'an 710072, China
ganhongping@nwpu.edu.cn

Abstract

*Deep unfolding networks (DUNs), renowned for their interpretability and superior performance, have invigorated the realm of compressive sensing (CS). Nonetheless, existing DUNs frequently suffer from issues related to insufficient feature extraction and feature attrition during the iterative steps. In this paper, we propose **U**nrolling **F**ixed-point **C**ontinuous **Net**work (**UFC-Net**), a novel deep CS framework motivated by the traditional fixed-point continuous optimization algorithm. Specifically, we introduce Convolution-guided Attention Module (CAM) to serve as a critical constituent within the reconstruction phase, encompassing tailored components such as Multi-head Attention Residual Block (MARB), Auxiliary Iterative Reconstruction Block (AIRB), etc. MARB effectively integrates multi-head attention mechanisms with convolution to reinforce feature extraction, transcending the confinement of localized attributes and facilitating the apprehension of long-range correlations. Meanwhile, AIRB introduces auxiliary variables, significantly bolstering the preservation of features within each iterative stage. Extensive experiments demonstrate that our proposed UFC-Net achieves remarkable performance both on image CS and CS-magnetic resonance imaging (CS-MRI) in contrast to state-of-the-art methods.*

1. Introduction

Compressive sensing (CS) represents an innovative technique that facilitates efficient signal acquisition and transmission [17]. It can acquire signals at significantly lower sampling rates than those prescribed by conventional algorithms while simultaneously achieving superior reconstruction quality [2], resulting in substantial savings in storage and transmission costs. Consequently, CS has emerged as a potent tool for addressing practical challenges across numerous domains, including snapshot compressive imaging [10, 31], hyperspectral compressive imaging [24, 49], image restoration [27, 56, 61], magnetic resonance imaging (MRI) [11, 12, 26, 32], among others [15, 33, 45].

*Corresponding author

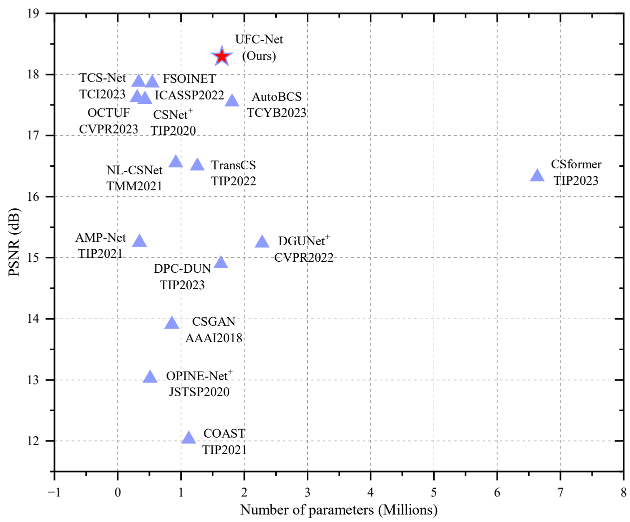

Figure 1. The PSNR performance comparison of our UFC-Net and fourteen SOTA methods on the CIFAR10 [29] dataset at a sampling rate $\delta = 0.01$.

Mathematically, the sampling phase of CS can be formulated as $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^N$ denotes the original signal, $\mathbf{A} \in \mathbb{R}^{M \times N} (M \ll N)$ represents the sampling matrix, and $\mathbf{y} \in \mathbb{R}^M$ is the measurements. The sampling rate is $\delta = \frac{M}{N}$. The intractable ill-posed issue of reconstructing $\mathbf{x}$ is reformulated into an optimization problem, exemplified by ℓ_1-regularized least squares problem as follows:

$$\min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x}\|_{\ell_1} + \frac{\mu}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_{\ell_2}^2, \tag{1}$$

where μ serves as a penalty coefficient, striking a balance between the data fitting term and the ℓ_1-norm regularization term. In order to effectively apply CS theory to practical scenarios, researchers have proposed various reconstruction algorithms [1, 3, 6, 8, 22, 23, 34] to address the ill-posed problem described by Eq. (1). Traditional image CS recovery can be broadly categorized into three major classes: iterative sparse reconstruction algorithms, matching pursuit algorithms, and other convex optimization methods. Among these, iterative sparse reconstruction algorithms have garnered significant popularity, for instance, iterative shrinkage/thresholding algorithm (ISTA) [3], and two accelerated ISTA algorithms, i.e., FISTA [3] and TwIST [6]. However, these aforementioned methods usually utilize fixed constraints of soft thresholds. In contrast, fixed-point continuation method (FPC) [22, 23] introduces an expansive strategy regarding soft thresholds, which can enhance the balance between modeling the sparsity of signals and adapting to the complexity of signal structures. Refer to the *Sect. 1 of* ***Supplementary*** for the principles of FPC.

Recently, amidst the flourishing landscape of deep learning (DL), some researchers have developed a series of DL-based image CS reconstruction algorithms, which typically can be categorized into two distinct paradigms. One approach involves pure model-based CS architectures, which model the signal reconstruction problem as an end-to-end learning task, such as MAC-Net [7], AutoBCS [18], TCS-Net [19]. While these approaches achieve high-quality reconstructions, pure model-based image CS methods are often regarded as black-box models, posing challenges in elucidating their internal operational principles.

In contrast, inspired by classical iterative thresholding algorithms, deep unfolding image CS methods such as ISTA-Net [57], DPC-DUN [40], OCTUF [42], generally utilize convolutional neural networks (CNNs) or Transformer architectures to decompose iterative threshold problems into multiple iterative steps mapped into deep networks. Although these deep unfolding image CS networks can offer a degree of interpretability while achieving high-quality reconstructions, there still exists the problem of insufficient feature extraction due to the feature representation structure based on CNNs or Transformer architectures. Furthermore, the intrinsic necessity for the output of reconstructed images at each iteration stage in the reconstruction process introduces inherent information loss.

To address the aforementioned challenges, we introduce a deep unfolding CS method based on the well-known FPC algorithm, denoted as UFC-Net. It unfurls the iterative recovery steps into a fixed number of concatenated blocks, with customized Gradient Descent Update (GDU) module and Convolution-guided Attention Module (CAM) to enhance feature extraction and minimize feature loss in the inter-reconstruction stages. As depicted in Fig. 2, CAM comprises two specialized sub-modules, i.e., Multi-head Attention Residual Block (MARB) and Auxiliary Iterative Reconstruction Block (AIRB). MARB synergizes convolution with multi-head attention mechanisms to obtain local features and contextual relationships within the image in each iterative reconstruction stage, and AIRB harnesses auxiliary variables to amplify information exchange between iterative stages. In summary, our main contributions are as follows:

- We propose UFC-Net for CS, which effectively harness the advantages of FPC and deep neural networks, thereby manifesting commendable performance and interpretability both on image CS and CS-MRI.
- We introduce MARB for UFC-Net, which integrates attention mechanisms and convolutional neural networks, excelling in capturing local attributes while simultaneously introducing long-range dependencies of images inside each iteration stage.
- We customize AIRB for UFC-Net, which incorporates auxiliary variables to further amplify inter-stage information interaction between each iteration stage, thereby preserving a greater abundance of features.

Extensive experiments underscore that our proposed UFC-Net exhibits formidable performance, surpassing state-of-the-art CS algorithms.

2. Related Works

Pure Model-based Image CS Methods. This category of methods represents a paradigm that formulates the image reconstruction problem as an end-to-end learning task, mapping features between measurement data and the original signal without explicitly establishing a mathematical model. Using feature processing units as classification criteria, researchers have delved into three distinct categories of model-based CS architectures, one of the types roots in CNNs, such as CSNet [39], MAC-Net [7], NL-CSNet [13], BNN [36], ASGLD [46] and AutoBCS [18]. An alternative CS paradigm is built upon the foundation of Transformer/Attention mechanisms, such as TCS-Net [19]. Furthermore, hybrid CS architectures combining both CNNs and Transformer/Attention approaches have garnered significant popularity, such as DPA-Net [44] and CSformer [52]. These two CS methods employ a dual-path architecture: the former uses two paths to respectively deal with texture and structure, while the latter introduces separate CNN and Transformer pathways. In comparison to unfolding DL-based image CS techniques, these pure model-based CS models are often regarded as black-box models, rendering it challenging to expound their internal operational principles.

Deep unfolding Image CS Methods. Deep unfolding networks (DUNs) typically transforms the classical iterative CS recovery algorithms into deep neural networks using feature architectures such as CNNs or Transformer/Attention. A quintessential illustration is ISTA-Net [57], which adeptly tackles the proximal mapping problem of sparse signals in CS by embedding a limited num-

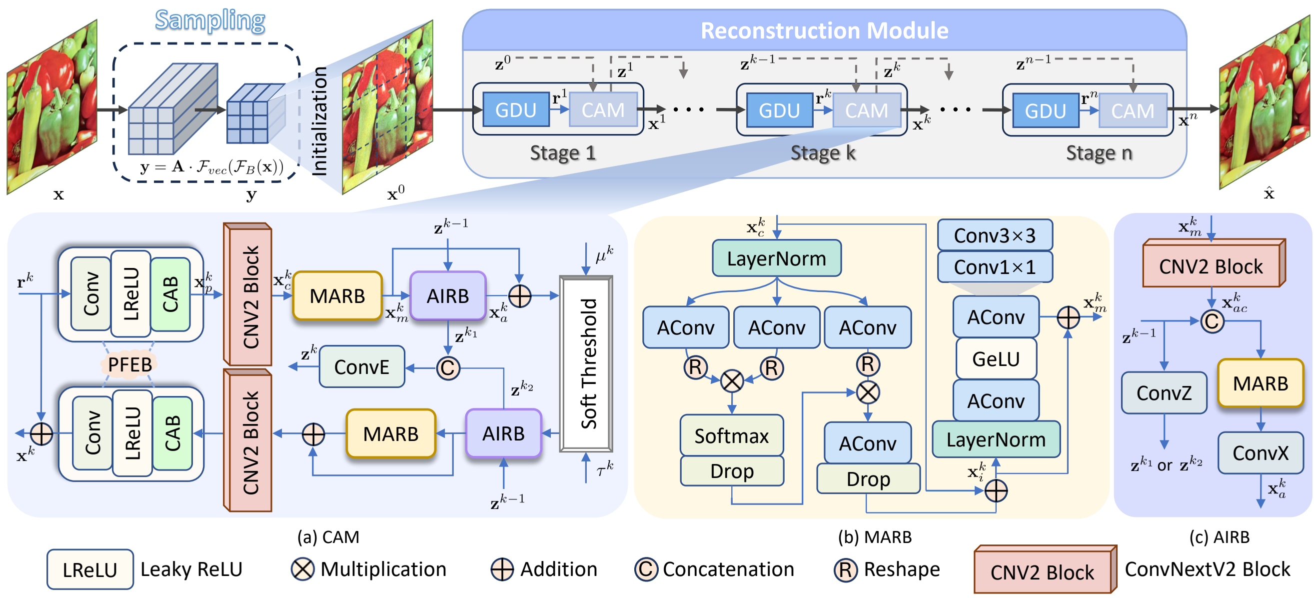

Figure 2. The proposed UFC-Net architecture. The topmost row represents the holistic structure, where the reconstruction module is comprised of n iterative recovery stages. Each stage corresponds to an iteration of the FPC algorithm and is constituted by GDU and CAM. The second row provides an in-depth delineation of each iterative stage, along with the details of the customized MARB and AIRB.

ber of predetermined CNN blocks. Moreover, Zhang et al. introduced AMP-Net [62], which leverages the deep iterative CNN denoising paradigm into the Approximate Message Passing algorithm (AMP) for image CS. Song et al. introduced OCTUF [42], a lightweight DUN method, which incorporates ISTA algorithm and leverages cross-attention Transformers to enable efficient image reconstruction. Other classical examples include OPINE-Net$^+$ [58], COAST [53], FSOINET [9], TransCS [38], DGUNet [35], DPC-DUN [40], LTwIST [20], among others [14, 21, 41].

These DUNs can enhance the interpretability of neural network methods, facilitating the understanding of signal recovery process. Due to the local inductive bias of CNNs or the limited local modeling capability of Transformers, these CS networks may inadequately extract and capture image features. As a result, they usually face certain challenges, including diminished image quality and block artifacts occurring in specific regions of the reconstructed images, particularly at low sampling rates. Furthermore, hindered by the image output requirements at each iterative stage, inadequate information exchange between recovery stages has impacted the representational capacity of DUNs methods.

3. The Proposed UFC-Net

3.1. Overall Architecture

The framework of our proposed UFC-Net is illustrated in Fig. 2, which encompasses sampling module and reconstruction module.

In the sampling module, we use block-based function $\mathcal{F}_B(\cdot)$ and vectorization function $\mathcal{F}_{vec}(\cdot)$ to expedite image acquisition. Subsequently, the sampling matrix $\mathbf{A}$ is adopted to obtain the measurements $\mathbf{y}$ from the segmented and vectorized images. Mathematically, the CS sampling can be modeled as $\mathbf{y} = \mathbf{A} \cdot \mathcal{F}_{vec}(\mathcal{F}_B(\mathbf{x}))$.

Following this, we first use $\mathbf{A}^T$(the transpose of $\mathbf{A}$) to achieve the initial estimation $\mathbf{x}^0$ from $\mathbf{y}$, i.e., $\mathbf{x}^0 = \mathbf{A}^T \cdot \mathbf{y}$. Afterward, $\mathbf{x}^0$ undergoes iterative updates through n repeated recovery stages within the reconstruction module to result in the estimate $\mathbf{x}^n$. Ultimately, by means of the inverse operations of $\mathcal{F}_B(\cdot)$ and $\mathcal{F}_{vec}(\cdot)$, that is, $\mathcal{F}_B^{-1}(\cdot)$ and $\mathcal{F}_{vec}^{-1}(\cdot)$, the final reconstructed image $\hat{\mathbf{x}}$ is obtained, i.e.,

$$\hat{\mathbf{x}} = \mathcal{F}_B^{-1}(\mathcal{F}_{vec}^{-1}(\mathbf{x}^n)). \tag{2}$$

Particularly, each stage in the reconstruction module encompasses both a Gradient Descent Update (GDU) module and a Convolution-guided Attention Module (CAM), corresponding to the iterative procedures of gradient descent and proximal mapping in the FPC algorithm, respectively. Concisely, the CAM in the k^{th} stage can be outlined as follows:

$$(\mathbf{x}^k, \mathbf{z}^k) = \mathrm{CAM}(\mathbf{r}^k, \mathbf{z}^{k-1}, \nu^k), \tag{3}$$

where $\mathbf{r}^k$, $\mathbf{z}^{k-1}$ and ν^k are inputs, while $\mathbf{x}^k$ and $\mathbf{z}^k$ represent outputs of k^{th} stage. Specifically, $\nu^k = \frac{\tau^k}{\mu^k}$, τ^k and μ^k are parameters related to soft threshold operator ν^k. Detailed explanations of these two sub-modules are provided in subsections 3.2 and 3.3. Overall, the proposed UFC-Net can be referenced in Algorithm 1.

Algorithm 1 Overall Process of UFC-Net

Input: Original image $\mathbf{x}$, sampling matrix $\mathbf{A}$, initial reconstruction matrix $\mathbf{A}^T$, the number of iteration stage n, gradient update step size $\lambda^{1\sim n}$, FPC parameters $\mu^{1\sim n}$, $\tau^{1\sim n}$, $\beta^{1\sim n}$, γ.

Output: Reconstructed image $\hat{\mathbf{x}}$

1: **Learnable parameters:** $\mathbf{A}$, $\mathbf{A}^T$, $\lambda^{1\sim n}$, $\mu^{1\sim n}$, $\tau^{1\sim n}$, $\beta^{1\sim n}$, ζ

2: **Sampling module:**
$\mathbf{y} = \mathbf{A} \cdot \mathcal{F}_{vec}(\mathcal{F}_B(\mathbf{x}))$

3: **Initial reconstruction:**
$\mathbf{x}^0 = \mathbf{A}^T \cdot \mathbf{y}$

4: **Iterative reconstruction:**

5: **while** $1 \leq k \leq n$ **do**

6: $\quad \mathbf{x}^{k-1} = \mathcal{F}_{vec}(\mathbf{x}^{k-1})$

7: $\quad \mathbf{r}^k = \mathcal{F}_{vec}^{-1}(\mathbf{x}^{k-1} - \lambda^k \mathbf{A}^T(\mathbf{A}\mathbf{x}^{k-1} - \mathbf{y}))$

8: $\quad \mu^k = \min(\mu^{k-1} \cdot \beta^k, \overline{\mu})$

9: $\quad \nu^k = \tau^k / \mu^k$

10: $\quad (\mathbf{x}^k, \mathbf{z}^k) = \mathrm{CAM}(\mathbf{r}^k, \mathbf{z}^{k-1}, \nu^k)$

11: $\quad k \leftarrow k + 1$

12: **end while**

13: $\hat{\mathbf{x}} = \mathcal{F}_B^{-1}(\mathcal{F}_{vec}^{-1}(\mathbf{x}^n))$

3.2. Gradient Descent Update Module

During each stage of reconstruction module, we initiate the procedure with a GDU approach, which involves the computation of gradients pertaining to data fidelity terms $\frac{1}{2}||\mathbf{A}\mathbf{x}-\mathbf{y}||_2^2$, aimed at reducing the disparities between measurements and the reconstructed image. Mathematically, GDU in the k^{th} stage can be formalized as follows:

$$\mathbf{r}^k = \mathcal{F}_{vec}^{-1}(\mathbf{x}^{k-1} - \lambda^k \mathbf{A}^T(\mathbf{A}\mathbf{x}^{k-1} - \mathbf{y})), \tag{4}$$

where

$$\mathbf{x}^{k-1} = \mathcal{F}_{vec}(\mathbf{x}^{k-1}), \tag{5}$$

where $\mathbf{x}^{k-1}$ corresponds to the output of the $(k-1)^{th}$ stage, notably, when $k = 1$, it represents the initial reconstruction value, $\mathbf{x}^0$. $\mathbf{A}^T$ designates the transposition of sampling matrix $\mathbf{A}$. λ^k denotes the step size for gradient descent updates, serving as a learnable parameter that iteratively changes during backpropagation. $\mathbf{r}^k$ signifies the outcome of gradient computations for $\mathbf{x}^{k-1}$. In other words, $\mathbf{r}^k$ is the generated preliminary reconstruction result.

3.3. Convolution-guided Attention Module

The Convolution-guided Attention Module (CAM) seamlessly succeeds the GDU. In the k^{th} stage, CAM takes the output $\mathbf{r}^k$ from GDU and the auxiliary varible $\mathbf{z}^{k-1}$ as its inputs, yielding $\mathbf{x}^k$ and $\mathbf{z}^k$ as outputs. Notably, it is no necessity for computing $\mathbf{z}^n$ in the n^{th} stage.

The entire CAM, with the soft threshold at its core, elucidates a non-linear solution for the proximal mapping of sparsity-related problems with intricate variations. It commences with a Preliminary Feature Extraction Block (PFEB), followed by the CNV2 block (a ConvNext V2 block [47]), which contributes to the deepening of feature extraction.

Subsequently, we introduce Multi-head Attention Residual Block (MARB) to facilitate the capture of both local attributes and long-range dependencies. Following this, we customize the Auxiliary Iterative Reconstruction Block (AIRB) to preserve a more extensive range of multi-channel information. The residual connection is thoughtfully applied both preceding and succeeding the AIRB, enhancing the network’ s capacity to model complex data. Next, sparsity is elevated through the application of soft thresholding. Afterwards, we execute a process that entirely reverses the actions taken before soft thresholding. Ultimately, a residual connection is established with the input $\mathbf{r}^k$ to yield the output $\mathbf{x}^k$ for the k^{th} stage. Furthermore, the intermediate values $\mathbf{z}^{k_1}$ and $\mathbf{z}^{k_2}$ generated by the two AIRB before and after soft thresholding are cascaded and processed through a convolutional operation to yield $\mathbf{z}^k$, which then serves as input for the subsequent stage, as depicted in Fig. 2.

Hence, our proposed UFC-Net is not only proficient in acquiring local features but also adept at establishing distant dependencies, all while meticulously preserving features between each iterative stage. Certainly, we sequentially delve into the inner workings of PFEB, MARB, AIRB and soft threshold.

Preliminary Feature Extraction Block. As shown in Fig. 2(a), in the pursuit of representing intricate image features, we first perform a convolutional operation $\mathrm{Conv}(\cdot)$ to transform the initial single-channel image input into a multi-channel format containing 32 channels. Then, we apply a leaky ReLU activation function $\mathrm{LReLU}(\cdot)$. Subsequently, we integrate a channel attention mechanism $\mathrm{C}(\cdot)$ to finely calibrate the significance of various dimensions. This calibration process serves a dual purpose, both improving feature representation and reducing redundant information. These procedures can be mathematically formalized as follows:

$$\mathbf{x}_p^k = \mathcal{P}(\mathbf{r}^k) = \mathrm{C}(\mathrm{LReLU}(\mathrm{Conv}(\mathbf{r}^k))), \tag{6}$$

where $\mathcal{P}(\cdot)$ represents the PFEB operation, while $\mathbf{x}_p^k$ signifies the output of this module.

Multi-head Attention Residual Block. In order to transcend the local intrinsic properties inherent in pure convolutions and capture long-distance dependencies and spatial relationships within images, we design a Multi-head Attention Residual Block (MARB), as visualized in Fig. 2(b). Before embarking on a detailed exposition, it is imperative to clarify the introduction of a structure denoted as Aconv, which represents the sequence of 3×3 convolution kernels and 1×1 convolution kernels. We designate $\mathbf{x}_c^k$ as the output

Table 1. PSNR (dB) and SSIM comparisons of UFC-Net and several state-of-the-art methods on large datasets CIFAR10 and CIFAR100 [29] at various sampling rates $\delta \in \{0.01, 0.04, 0.05, 0.10\}$.

Datasets	δ	CSGAN [28] (AAAI2018)		CSNet$^+$ [39] (TIP2020)		OPINE-Net$^+$ [58] (JSTSP2020)		NL-CSNet [13] (TMM2021)		FSOINET [9] (ICASSP2022)		DGUNet$^+$ [35] (CVPR2022)		AutoBCS [18] (TCYB2023)		CSformer [52] (TIP2023)		OCTUF [42] (CVPR2023)		UFC-Net (our method)	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CIFAR10	0.01	13.91	0.2993	17.59	0.2913	13.03	0.1147	16.55	0.3012	17.86	0.3202	15.24	0.2718	17.55	0.3094	16.32	0.2850	17.61	0.3075	**18.29** (0.43↑)	**0.3561** (0.0359↑)
	0.04	16.54	0.4364	20.75	0.5853	19.24	0.5630	20.11	0.5731	21.48	0.6301	19.41	0.5563	21.08	0.6260	20.15	0.5727	21.41	0.6306	**21.73** (0.25↑)	**0.6415** (0.0109↑)
	0.05	/	/	21.69	0.6581	19.80	0.6212	20.99	0.6298	22.22	0.6876	20.22	0.5922	22.19	0.6818	21.06	0.6294	22.27	0.6890	**22.48** (0.21↑)	**0.6930** (0.0040↑)
	0.10	18.35	0.5788	23.98	0.8027	22.33	0.7626	23.64	0.7833	25.21	0.8283	22.68	0.7629	24.47	0.8207	23.68	0.7814	25.26	0.8289	**25.39** (0.13↑)	**0.8308** (0.0019↑)
CIFAR100	0.01	13.87	0.3058	17.72	0.3066	12.70	0.1172	16.68	0.3256	18.18	0.3410	15.51	0.2918	17.73	0.3232	16.48	0.3010	17.80	0.3218	**18.64** (0.46↑)	**0.3729** (0.0319↑)
	0.04	16.56	0.4410	21.01	0.5946	19.50	0.5745	20.67	0.834	22.01	0.6403	19.74	0.5647	21.50	0.6337	20.52	0.5820	21.90	0.6411	**22.23** (0.22↑)	**0.6515** (0.0104↑)
	0.05	/	/	21.99	0.6643	20.09	0.6301	21.27	0.6389	22.76	0.6958	20.67	0.6059	22.67	0.6909	21.38	0.6377	22.78	0.6967	**23.00** (0.22↑)	**0.7007** (0.0040↑)
	0.10	18.41	0.5786	24.22	0.8031	22.67	0.7643	24.01	0.7845	25.78	0.8319	23.01	0.7631	24.95	0.8218	24.05	0.7830	25.86	0.8323	**25.96** (0.10↑)	**0.8334** (0.0011↑)

of the CNV2 block with $\mathbf{x}_p^k$ as its input, and then employ $\mathbf{x}_c^k$ as the input for MARB. Then we commence with layer normalization operation $\mathrm{LN1}(\cdot)$ [50], and subsequently engage a multi-head attention mechanism, $\mathrm{MHSA}(\cdot)$, to enable the network to learn the interrelationships between distinct regions within the image. Furthermore, the output of $\mathrm{MHSA}(\cdot)$ is established a residual connection with $\mathbf{x}_c^k$, leading to the intermediary value $\mathbf{x}_i^k$. The process can be formally represented as follows:

$$\mathbf{x}_i^k = \mathbf{x}_c^k + \mathrm{MHSA}(\mathrm{LN1}(\mathbf{x}_c^k)). \tag{7}$$

Then, $\mathbf{x}_i^k$ undergoes another layer normalization $\mathrm{LN2}(\cdot)$, further is processed via an AConv structure $\mathrm{AConv1}(\cdot)$, incorporating a GeLU activation function $\mathrm{GeLU}(\cdot)$, followed by another AConv structure $\mathrm{AConv2}(\cdot)$. Ultimately, MARB concludes with a residual connection with $\mathbf{x}_i^k$, yielding the definitive output $\mathbf{x}_m^k$ of the MARB in the k^{th} stage. This is mathematically formulated as follows:

$$\mathbf{x}_t^k = \mathrm{AConv1}(\mathrm{LN2}(\mathbf{x}_i^k)), \tag{8}$$

$$\mathbf{x}_m^k = \mathbf{x}_i^k + \mathrm{AConv2}(\mathrm{GeLU}(\mathbf{x}_t^k)), \tag{9}$$

where $\mathbf{x}_t^k$ signifies the intermediary output resultant from the sequential application of the $\mathrm{LN2}(\cdot)$ and $\mathrm{AConv1}(\cdot)$.

Auxiliary Iterative Reconstruction Block. As shown in Fig. 2(c), we design Auxiliary Iterative Reconstruction Block (AIRB) to preserve a broader spectrum of channel-specific information. Collectively, takig $\mathbf{x}_m^k$ and $\mathbf{z}^{k-1}$ as inputs, AIRB engenders two outputs, namely, $\mathbf{x}^k$ and $\mathbf{z}^{\{k_1,k_2\}}$[1].

To be specific, the process unfolds as follows. Initially, we attain $\mathbf{x}_{ac}^k$ via the CNV2 block $\mathrm{CB}(\cdot)$ taking $\mathbf{x}_m^k$ as input. Moreover, $\mathbf{x}_{ac}^k$ is concatenated seamlessly with the auxiliary variable $\mathbf{z}^{k-1}$ originating from the previous stage. The amalgamation outcome undergoes MARB, and then culminates in a convolution layer $\mathrm{ConvX}(\cdot)$, yielding one of the outputs, denoted as $\mathbf{x}_a^k$. The formal expression of this process is delineated as follows:

$$\mathbf{x}_a^k = \mathrm{ConvX}(\mathrm{MARB}(\mathcal{F}_c(\mathrm{CB}(\mathbf{x}_m^k), \mathbf{z}^{k-1}))), \tag{10}$$

[1]In the experiment, one of the outputs from AIRB before the soft thresholding is $\mathbf{z}^{k_1}$, and after is $\mathbf{z}^{k_2}$.

where $\mathcal{F}_c(\cdot,\cdot)$ denotes the concatenation of two inputs. Furthermore, an additional convolution operation $\mathrm{ConvZ}(\cdot)$ is applied to the input $\mathbf{z}^{k-1}$, resulting in the output $\mathbf{z}^{k_1}$:

$$\mathbf{z}^{k_1} = \mathrm{ConvZ}(\mathbf{z}^{k-1}). \tag{11}$$

In particular, when $k = 1$, $\mathbf{z}^0$ is derived from a preprocessing of $\mathbf{r}^1$. To elaborate, this entails a convolution operation $\mathrm{Conv_I}(\cdot)$ followed by the CNV2 block layer $\mathrm{CB_I}(\cdot)$, which is formulated as follow:

$$\mathbf{z}^0 = \mathrm{CB_I}(\mathrm{Conv_I}(\mathbf{r}^1)). \tag{12}$$

Soft Threshold. During the execution of the soft thresholding operation, the soft thresholding operator ν^k in the k^{th} stage is computed by the division of two parameters, τ^k and μ^k. Among these, τ^k is initialized according to $\tau^k = \min\{1 + 1.665(1-\delta), 1.999\}$. Moreover, μ^1 is initialized as $\mu^1 = \tau^1/(\gamma \cdot ||\mathbf{x}^0||_\infty)$. The constant, denoted as γ, is established with a fixed value of 0.9. Particularly, we introduce the parameter β^k to derive the extrapolated sequence of μ^k.

$$\mu^k = \min(\mu^{k-1} \cdot \beta^k, \overline{\mu}), \tag{13}$$

where β^k is initialized as 2, and $\overline{\mu}$ is a constant with 5000. It is imperative to emphasize that the parameters $\mu^{1\sim n}, \tau^{1\sim n}, \beta^{1\sim n}, \lambda^{1\sim n}$ are all trainable, eliminating the need for manual customization, which significantly bolsters the adaptability, intelligence, and versatility of our UFC-Net. Subsequent to the soft threshoding, we execute a procedure entirely antithetical to the previous soft thresholding process.

Furthermore, in the concluding segment of the k^{th} stage, we amalgamate the two auxiliary variables, $\mathbf{z}^{k_1}$ and $\mathbf{z}^{k_2}$, which are derived from the AIRB both before and after the soft-thresholding process. This fusion is achieved through concatenation and a convolution layer $\mathrm{ConvE}(\cdot)$ to yield $\mathbf{z}^k$, serving as the auxiliary variable input for the subsequent AIRB stage. This process is mathematically represented as follows:

$$\mathbf{z}^k = \mathrm{ConvE}(\mathcal{F}_c(\mathbf{z}^{k_1}, \mathbf{z}^{k_2})). \tag{14}$$

Table 2. PSNR (dB) and SSIM comparisons of UFC-Net and competing methods on datasets Set11 [30], Set14 [55], Urban100 [25], and General100 [16] at different sampling rates $\delta \in \{0.01, 0.04, 0.10, 0.25\}$.

Datasets	δ	ISTA-Net$^+$ [57] (CVPR2018)		DPA-Net [44] (TIP2020)		MAC-Net [7] (ECCV2020)		AMP-Net [62] (TIP2021)		COAST [53] (TIP2021)		TransCS [38] (TIP2022)		DPC-DUN [40] (TIP2023)		TCS-Net [19] (TCI2023)		LTwIST [20] (TCSVT2023)		UFC-Net (our method)	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set11	0.01	17.42	0.4130	18.20	0.5101	18.25	0.4002	20.19	0.5578	12.39	0.2636	20.13	0.5063	18.01	0.4600	21.09	0.5504	20.98	0.5469	**21.24** (0.15↑)	**0.5607** (0.0029↑)
	0.04	21.55	0.6238	24.26	0.7541	24.21	0.6981	25.24	0.7719	23.54	0.7157	25.39	0.7880	24.37	0.7497	25.45	0.7863	25.71	0.7900	**25.92** (0.21↑)	**0.7943** (0.0043↑)
	0.10	26.46	0.8035	27.66	0.8530	27.67	0.8181	29.37	0.8775	28.69	0.8607	29.51	0.8873	29.40	0.8800	29.05	0.8835	29.84	0.8930	**30.15** (0.31↑)	**0.8960** (0.0030↑)
	0.25	32.43	0.9236	32.38	0.9311	32.90	0.9243	34.61	0.9480	33.95	0.9404	35.02	0.9544	34.72	0.9480	33.95	0.9540	35.00	0.9546	**35.42** (0.40↑)	**0.9567** (0.0021↑)
Set14	0.01	18.20	0.4012	18.30	0.4613	18.43	0.3974	21.55	0.5301	13.49	0.2740	20.88	0.4849	19.01	0.4550	21.63	0.5218	21.48	0.5190	**21.79** (0.16↑)	**0.5324** (0.0023↑)
	0.04	22.07	0.5707	23.69	0.6531	23.71	0.6171	25.42	0.6996	23.67	0.6417	25.45	0.7129	24.32	0.6630	25.25	0.7072	25.44	0.7112	**25.67** (0.22↑)	**0.7163** (0.0034↑)
	0.10	25.98	0.7288	26.29	0.7690	26.40	0.7381	28.70	0.8179	27.41	0.7772	28.79	0.8340	28.04	0.7951	28.19	0.8284	28.82	0.8342	**29.10** (0.28↑)	**0.8363** (0.0021↑)
	0.25	30.61	0.8699	30.15	0.8812	30.67	0.8742	33.12	0.9136	32.04	0.8921	33.34	0.9239	32.76	0.9022	32.22	0.9205	33.40	0.9241	**33.81** (0.41↑)	**0.9259** (0.0018↑)
Urban100	0.01	16.66	0.3733	16.36	0.4150	16.39	0.3637	19.55	0.5016	12.90	0.2616	18.96	0.4395	17.28	0.4214	19.61	0.4946	19.46	0.4886	**19.69** (0.08↑)	**0.5041** (0.0025↑)
	0.04	19.65	0.5368	21.64	0.6486	21.60	0.6120	22.73	0.6819	21.40	0.6331	23.25	0.7114	22.35	0.6767	22.94	0.7035	23.01	0.7061	**23.37** (0.12↑)	**0.7195** (0.0081↑)
	0.10	23.48	0.7200	24.55	0.7841	24.49	0.7465	25.92	0.8144	25.90	0.8021	26.74	0.8416	26.94	0.8358	25.88	0.8290	26.76	0.8463	**27.55** (0.61↑)	**0.8583** (0.0120↑)
	0.25	28.89	0.8830	28.80	0.8944	28.79	0.8798	30.79	0.9188	31.07	0.9165	31.75	0.9329	32.33	0.9320	30.12	0.9241	31.79	0.9349	**32.82** (0.49↑)	**0.9423** (0.0074↑)
General100	0.01	19.00	0.4698	19.37	0.5436	19.72	0.4857	22.68	0.6109	12.85	0.2960	21.65	0.5414	19.92	0.5361	22.59	0.5977	22.69	0.5989	**23.08** (0.39↑)	**0.6145** (0.0036↑)
	0.04	23.74	0.6545	25.96	0.7472	26.17	0.7169	26.91	0.7689	25.91	0.7352	27.23	0.7841	26.60	0.7529	26.58	0.7712	27.53	0.7935	**27.92** (0.39↑)	**0.7988** (0.0053↑)
	0.10	28.52	0.8100	29.05	0.8497	29.70	0.8275	30.77	0.8712	30.61	0.8572	31.38	0.8916	31.15	0.8714	29.91	0.8749	31.91	0.8990	**32.31** (0.40↑)	**0.9014** (0.0024↑)
	0.25	34.31	0.9248	33.71	0.9316	34.83	0.9283	35.93	0.9493	35.77	0.9405	37.05	0.9599	36.49	0.9479	34.64	0.9505	37.31	0.9616	**37.75** (0.44↑)	**0.9624** (0.0008↑)

3.4. Loss Function

We employ the mean squared error (MSE) as the loss function, encompassing all trainable parameters denoted as $\boldsymbol{w}$. The original image $\mathbf{x}$ serves as the ground truth, and the reconstructed value $\hat{\mathbf{x}}$ derived from measurements $\mathbf{y}$ corresponding to $\mathbf{x}$ serves as the network' s output, thus formulating the loss function as:

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{2\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left\| \mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)} \right\|_2^2, \tag{15}$$

where $\mathcal{N}$ denotes the aggregate count of training images, $\mathbf{x}^{(i)}$ represents the i^{th} trainable image, and $\boldsymbol{w} = \{\mathbf{A}, \mathbf{A}^T, \lambda^{1\sim n}, \mu^{1\sim n}, \tau^{1\sim n}, \beta^{1\sim n}, \zeta\}$, where ζ represents the convolution bias, etc.

4. Experiments

4.1. Experimental Settings

The default number of iterative stages of our UFC-Net is 10. The batch size is 64 during training. The default feature channel number is set to 32. Gradient descent step sizes $\lambda^{1\sim n}$ are initialized to 0.5, and the parameters $\mu^{1\sim n}$, $\tau^{1\sim n}$ and $\beta^{1\sim n}$ are initialized as elucidated in the soft threshold at the end of subsection 3.3. All our experiments are implemented based on the PyTorch 1.12.0 framework with a GeForce RTX 3090 GPU. Refer to the *Sect. 2 of **Supplementary*** to obtain more experimental settings. Our source code is available at UFC-Net.

4.2. Comparisons on UFC-Net and Competing Methods

In this subsection, we undertake a comparative analysis between the proposed UFC-Net and other eighteen state-of-the-art algorithms, which encompass ISTA-Net$^+$ [57], CSGAN [28], CSNet$^+$ [39], DPA-Net [44], OPINE-Net$^+$ [58], MAC-Net [7], NL-CSNet [13], AMP-Net [62], COAST [53], FSOINET [9], TransCS [38], DGUNet$^+$ [35], AutoBCS [18], TCS-Net [19], DPC-DUN [40], CSformer [52], OCTUF [42] and LTwIST [20]. The outcomes are delineated in Tab. 1, Tab. 2 and Fig. 3. Specifically, the best results in the tables are indicated in bold, while the second-best outcomes are designated by underlining.

Firstly, with $\delta \in \{0.01, 0.04, 0.05, 0.10\}$, we perform a comparison of ten algorithms including CSGAN, CSNet$^+$, OPINE-Net$^+$, NL-CSNet, FSOINET, DGUNet$^+$, AutoBCS, CSformer, OCTUF and our proposed UFC-Net on the large-scale CIFAR10 and CIFAR100 datasets [29]. The results in Tab. 1 illustrate the exceptional performance of our proposed UFC-Net, demonstrating its optimal performance across all scenarios. For instance, in comparison to OCTUF, our UFC-Net demonstrates remarkable enhancements in PSNR (percentage gains) and SSIM (percentage gains) on the CIFAR10 dataset when $\delta = 0.01$, with increases of 0.68dB ($\sim$3.86%) and 0.0486 ($\sim$15.80%), respectively. Similarly, on the CIFAR100 dataset, UFC-Net achieves significant improvements of 0.84dB ($\sim$4.72%) and 0.0511 ($\sim$15.88%) in PSNR and SSIM, respectively.

Furthermore, we extend our comparative experiments to widely employed datasets, including Set11 [30], Set14 [55], Urban100 [25], and General100 [16]. We compare UFC-Net with other nine algorithms, including ISTA-Net$^+$, DPA-Net, MAC-Net, AMP-Net, COAST, TransCS, DPC-DUN, TCS-Net and LTwIST, as elaborated in Tab. 2. These results unequivocally illustrate the substantial advantage of our proposed UFC-Net. For instance, on the Urban100 dataset, our UFC-Net demonstrates respective enhancements of 0.61dB ($\sim$2.26%) and 0.0120 ($\sim$1.42%) in PSNR and SSIM with $\delta = 0.10$, compared to the second-best algorithm.

Moreover, we arrange the reconstructed images generated by our UFC-Net and other eight algorithms in Fig. 3. To facilitate observations, we directly provide enlarged views of selected regions. It is discernible that our proposed UFC-Net excels in reconstructing images of superior quality, achieving enriched details and crisper lines.

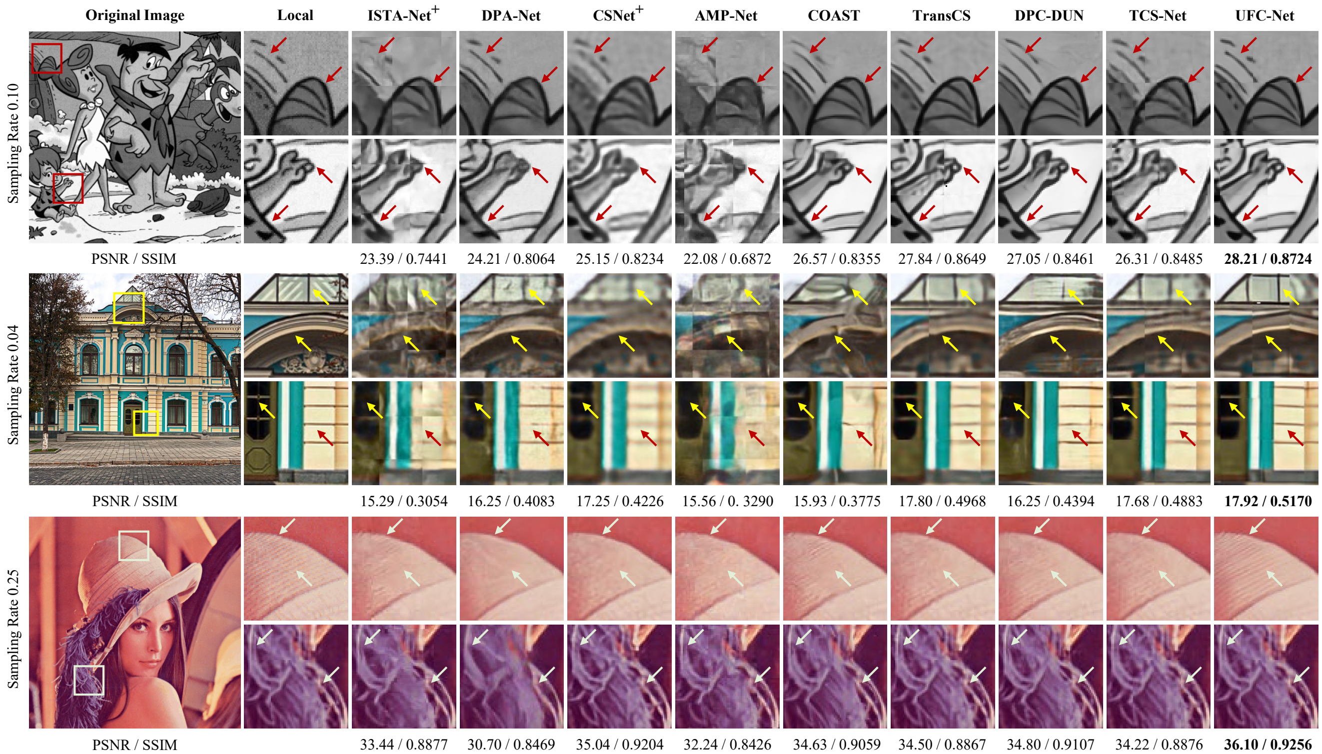

Figure 3. Comparisons of reconstructed images between UFC-Net and eight competing algorithms. The first row pertains to comparative images from the flintstones in Set11 [30] at $\delta = 0.10$. The second row corresponds to comparative images from Urban100 dataset [25] at $\delta = 0.04$, while the third row relates to comparative images from lenna in Set14 [55] at $\delta = 0.25$. Regions of interest are magnified, with distinctive differences indicated by arrows for ease of observation.

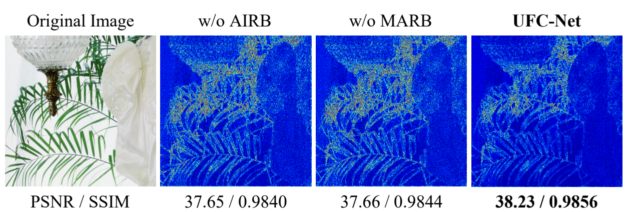

Figure 4. Visual analysis of the reconstructed images and corresponding error maps generated by various models derived from UFC-Net, conducted on McM18 dataset [60] at $\delta = 0.25$.

4.3. UFC-Net without MARB or AIRB

To substantiate the effectiveness of our customized modules, we produce two variants of UFC-Net by selectively excluding modules MARB and AIRB, denoted as w/o MARB and w/o AIRB, respectively.

Experimental comparisons are conducted between UFC-Net along with the two aforementioned variants on widely adopted datasets Set5, Set11 and McM18 at different sampling rates $\delta \in \{0.10, 0.25\}$. From the results illustrated in Tab. 3, it is evident that the removal of any module leads to a substantial decline in the overall performance, underscoring the beneficial contributions of the proposed MARB and AIRB. The assessment metrics of reconstructed images and corresponding error maps shown in Fig. 4 collectively indicate that both our proposed MARB and AIRB modules excel at capturing richer image textures and fine-grained details, leading to higher detail fidelity.

Table 3. PSNR (dB) and SSIM comparisons of UFC-Net with different components on datasets Set5 [5], Set11 [30] and McM18 [60] at different sampling rates.

Methods	δ	Set5		Set11		McM18	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
w/o AIRB	0.10	32.30	0.9138	29.83	0.8917	31.71	0.8967
	0.25	36.88	0.9585	35.16	0.9549	37.03	0.9606
w/o MARB	0.10	32.23	0.9137	29.81	0.8905	31.74	0.8977
	0.25	36.92	0.9589	35.11	0.9547	37.05	0.9610
UFC-Net	0.10	**32.51**	**0.9170**	**30.15**	**0.8960**	**31.97**	**0.9011**
	0.25	**37.08**	**0.9597**	**35.42**	**0.9567**	**37.24**	**0.9619**

Additionally, we visually analyze the feature maps of our proposed UFC-Net and variants at different stages in *Sect. 3.1 of* ***Supplementary***. We also conduct ablation experiments of our UFC-Net with different numbers of iterative stages in *Sect. 3.2 of* ***Supplementary***, and evaluate the UFC-Net performance under Gaussian noise, and salt and pepper noise in *Sect. 3.3 of* ***Supplementary***.

Table 4. Complexity and running-time on GPU comparisons of UFC-Net and different methods with an input image of 256×256 at the sampling rate $\delta = 0.10$.

Methods	DGUNet$^+$	DPC-DUN	OCTUF	LTwIST	UFC-Net
GPU/s	0.044	0.053	0.031	0.040	0.036
Params (M)	2.287	1.634	0.400	23.491	1.741
GFLOPs	97.79	74.15	189.30	110.74	109.00

4.4. Complexity Analysis

This subsection performs a comprehensive complexity analysis for five SOTA models: DGUNet$^+$, DPC-DUN, OCTUF, LTwIST and our UFC-Net at $\delta = 0.10$. We employ single-channel images with dimensions of 256×256 as inputs. This analysis encompasses various aspects, including running time on GPU (the averaged outcome of 100 repetitions), the number of parameters, and the giga floating-point operations (GFLOPs).

As illustrated in Tab. 4, the results indicate that our UFC-Net boasts an exceptionally fast reconstruction time, and that the parameters and GFLOPs of UFC-Net, while not optimal, still showcase a relatively lower levels of intricacy.

4.5. UFC-Net for CS-MRI

In the pursuit of elevating the profundity and scope of our method, we extend our proposed UFC-Net from the field of image CS to CS-MRI. The principal distinction lies in the redefinition of sampling matrix $\mathbf{A}$ involved UFC-Net as $\mathbf{A} = \mathcal{M} \cdot \mathcal{T}$, where $\mathcal{M}$ represents the Cartesian matrix and $\mathcal{T}$ denotes the discrete Fourier transform, while the other components of the reconstruction module remain unaltered. Please refer to the *Sect. 2 of* ***Supplementary*** for experimental settings in CS-MRI.

We conduct a comparative evaluation of the proposed UFC-Net and nine meticulously crafted CS-MRI methods encompassing Zero-filled [4], DC-CNN [37], ISTA-Net$^+$ [57], RDN [43], CDDN [63], ADMM-CSNet [51], PUERT [48], HiTDUN [59], and LTwIST [20] at $\delta \in \{0.05, 0.10.0.15\}$. It is imperative to emphasize that these approaches are retrained on the provided FastMRI knee dataset [54] to ensure equitable comparisons.

As delineated in the Tab. 5, the comparative findings unequivocally demonstrate the pronounced superiority of our UFC-Net. For example, when compared to the second-ranking algorithm, HiTDUN, at $\delta = 0.15$, UFC-Net demonstrates a substantial increment of 0.36dB ($\sim 1.24\%$) in PSNR and a improvement of 0.0121 ($\sim 1.73\%$) in SSIM. Furthermore, we showcase the reconstructed knee images produced by UFC-Net alongside those of other comparative algorithms at three sampling rates, as shown in Fig. 5 and *Fig. 1 of* ***Supplementary***, respectively. It is conspicuously evident that the images generated by our UFC-Net exhibit superior clarity in capturing fine textural details.

Table 5. PSNR (dB) and SSIM comparisons of several CS-MRI methods on datasets FastMRI [54] at different sampling rates $\delta \in \{0.05, 0.10, 0.15\}$.

Methods	Sampling Rate					
	0.05		0.10		0.15	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Zero-filled [4]	23.23	0.4564	25.38	0.5314	26.31	0.5763
DC-CNN [37]	26.81	0.5432	27.07	0.5899	27.92	0.6624
ISTA-Net$^+$ [57]	27.02	0.5498	27.20	0.5946	28.85	0.6922
RDN [43]	27.00	0.5471	27.32	0.5967	28.89	0.6911
ADMM-CSNet [51]	27.06	0.5512	27.43	0.5999	28.79	0.6898
CDDN [63]	27.13	0.5535	27.42	0.6015	28.93	0.6941
PUERT [48]	26.93	0.5465	27.53	0.6043	28.51	0.6815
HiTDUN [59]	27.08	0.5513	27.62	0.6057	29.10	0.6977
LTwIST [20]	27.23	0.5517	27.70	0.6003	29.03	0.6914
UFC-Net	**27.43**	**0.5642**	**28.01**	**0.6202**	**29.46**	**0.7098**

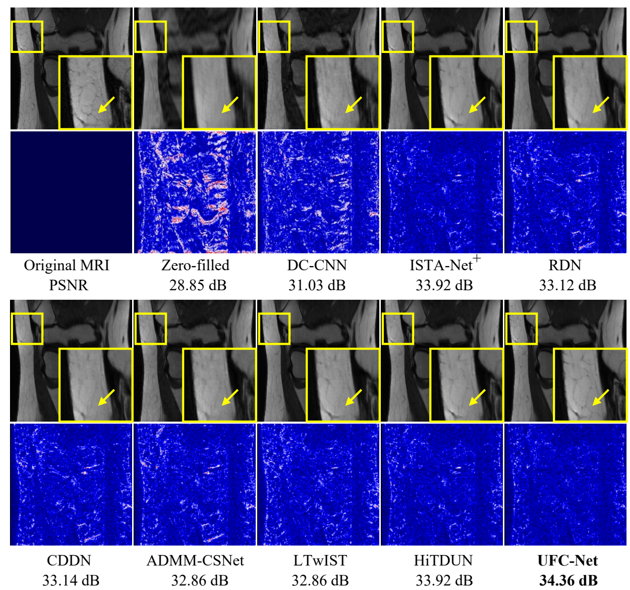

Figure 5. Comparisons of reconstruction images and error maps of competing methods and our UFC-Net on the dataset FatMRI when $\delta = 0.15$. Local areas are zoomed in for better comparisons.

5. Conclusions

In this paper, we propose a unrolling DL-based fixed-point continuous network for CS, denoted as UFC-Net, which effectively expands the FPC optimization algorithm into a deep neural network paradigm. Particularly, we customize GDU and CAM, including MARB and AIRB components. MARB strengthens feature extraction capability, while AIRB enhances feature fusion and reduces inter-stage feature loss during the iterative reconstruction process. Substantial experiments on image CS and CS-MRI tasks prove to the state-of-the-art performance of our UFC-Net.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62101455.

References

[1] Manya V. Afonso, Jos M. Bioucas-Dias, and Mrio A. T. Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, 20(3):681–695, 2011. 2

[2] Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010. 1

[3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 2

[4] Matt A Bernstein, Sean B Fain, and Stephen J Riederer. Effect of windowing and zero-filled reconstruction of MRI data on spatial resolution and acquisition strategy. *Journal of Magnetic Resonance Imaging*, 14(3):270–280, 2001. 8

[5] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference (BMVC)*, pages 135.1–135.10, 2012. 7

[6] JosÉ M. Bioucas-Dias and MÁrio A. T. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007. 2

[7] Jiwei Chen, Yubao Sun, Qingshan Liu, and Rui Huang. Learning memory augmented cascading network for compressed sensing of images. In *European Conference on Computer Vision (ECCV)*, pages 513–529. Springer, 2020. 2, 6

[8] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001. 2

[9] Wenjun Chen, Chunling Yang, and Xin Yang. FSOINET: Feature-space optimization-inspired network for image compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2460–2464, 2022. 3, 5, 6

[10] Ziheng Cheng, Bo Chen, Ruiying Lu, Zhengjue Wang, Hao Zhang, Ziyi Meng, and Xin Yuan. Recurrent neural networks for snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2264–2281, 2023. 1

[11] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-Closer-Diffuse-Faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12403–12412, 2022. 1

[12] Hyungjin Chung, Dohoon Ryu, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Solving 3D inverse problems using pre-trained 2D diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22542–22551, 2023. 1

[13] Wenxue Cui, Shaohui Liu, Feng Jiang, and Debin Zhao. Image compressed sensing using non-local neural network. *IEEE Transactions on Multimedia*, 25:816–830, 2021. 2, 5, 6

[14] Wenxue Cui, Xingtao Wang, Xiaopeng Fan, Shaohui Liu, Chen Ma, and Debin Zhao. G2-DUN: Gradient guided deep unfolding network for image compressive sensing. In *Proceedings of ACM International Conference on Multimedia*, pages 7933–7942, 2023. 3

[15] Chao Deng, Yuanlong Zhang, Yifeng Mao, Jingtao Fan, Jinli Suo, Zhili Zhang, and Qionghai Dai. Sinusoidal sampling enhanced compressive camera for high speed imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):1380–1393, 2021. 1

[16] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision (ECCV)*, pages 391–407, 2016. 6

[17] Marco F. Duarte and Richard G. Baraniuk. Kronecker compressive sensing. *IEEE Transactions on Image Processing*, 21(2):494–504, 2012. 1

[18] Hongping Gan, Yang Gao, Chunyi Liu, Haiwei Chen, Tao Zhang, and Feng Liu. AutoBCS: Block-based image compressive sensing with data-driven acquisition and noniterative reconstruction. *IEEE Transactions on Cybernetics*, 53 (4):2558–2571, 2023. 2, 5, 6

[19] Hongping Gan, Minghe Shen, Yi Hua, Chunyan Ma, and Tao Zhang. From patch to pixel: A transformer-based hierarchical framework for compressive image sensing. *IEEE Transactions on Computational Imaging*, 9:133–146, 2023. 2, 6

[20] Hongping Gan, Xiaoyang Wang, Lijun He, and Jie Liu. Learned two-step iterative shrinkage thresholding algorithm for deep compressive sensing. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. 3, 6, 8

[21] Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 6:328–343, 2020. 3

[22] Elaine T Hale, Wotao Yin, and Yin Zhang. A fixed-point continuation method for l_1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, 43:44, 2007. 2

[23] Elaine T. Hale, Wotao Yin, and Yin Zhang. Fixed-Point Continuation for l_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008. 2

[24] Wei He, Naoto Yokoya, and Xin Yuan. Fast hyperspectral image recovery of dual-camera compressive hyperspectral imaging via non-iterative subspace-based fusion. *IEEE Transactions on Image Processing*, 30:7170–7183, 2021. 1

[25] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6, 7

[26] Mathews Jacob, Jong Chul Ye, Leslie Ying, and Mariya Doneva. Computational MRI: Compressive sensing and beyond. *IEEE Signal Processing Magazine*, 37(1):21–23, 2020. 1

[27] Jerin Geo James, Pranay Agrawal, and Ajit Rajwade. Restoration of non-rigidly distorted underwater images using a combination of compressive sensing and local polynomial image representations. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7838–7847, 2019. 1

[28] Maya Kabkab, Pouya Samangouei, and Rama Chellappa. Task-aware compressed sensing with generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 5, 6

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1, 5, 6

[30] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. ReconNet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 449–458, 2016. 6, 7

[31] Yang Liu, Xin Yuan, Jinli Suo, David J. Brady, and Qionghai Dai. Rank minimization for snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2990–3006, 2019. 1

[32] Morteza Mardani, Enhao Gong, Joseph Y. Cheng, Shreyas S. Vasanawala, Greg Zaharchuk, Lei Xing, and John M. Pauly. Deep generative adversarial neural networks for compressive sensing MRI. *IEEE Transactions on Medical Imaging*, 38 (1):167–179, 2019. 1

[33] Julien N. P. Martel, Lorenz K. Müller, Stephen J. Carey, Piotr Dudek, and Gordon Wetzstein. Neural sensors: Learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1642–1653, 2020. 1

[34] Christopher A. Metzler, Arian Maleki, and Richard G. Baraniuk. From denoising to compressed sensing. *IEEE Transactions on Information Theory*, 62(9):5117–5144, 2016. 2

[35] Chong Mou, Qian Wang, and Jian Zhang. Deep generalized unfolding networks for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17399–17410, 2022. 3, 5, 6

[36] Tongyao Pang, Yuhui Quan, and Hui Ji. Self-supervised bayesian deep learning for image recovery with applications to compressive sensing. In *European Conference on Computer Vision (ECCV)*, pages 475–491. Springer, 2020. 2

[37] Jo Schlemper, Jose Caballero, Joseph V. Hajnal, Anthony N. Price, and Daniel Rueckert. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Transactions on Medical Imaging*, 37(2):491–503, 2018. 8

[38] Minghe Shen, Hongping Gan, Chao Ning, Yi Hua, and Tao Zhang. TransCS: A transformer-based hybrid architecture for image compressed sensing. *IEEE Transactions on Image Processing*, 31:6991–7005, 2022. 3, 6

[39] Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. Image compressed sensing using convolutional neural network. *IEEE Transactions on Image Processing*, 29:375–388, 2020. 2, 5, 6

[40] Jiechong Song, Bin Chen, and Jian Zhang. Dynamic path-controllable deep unfolding network for compressive sensing. *IEEE Transactions on Image Processing*, 32:2202–2214, 2023. 2, 3, 6

[41] Jiechong Song, Bin Chen, and Jian Zhang. Deep memory-augmented proximal unrolling network for compressive sensing. *International Journal of Computer Vision*, 131(6): 1477–1496, 2023. 3

[42] Jiechong Song, Chong Mou, Shiqi Wang, Siwei Ma, and Jian Zhang. Optimization-inspired cross-attention transformer for compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6174–6184, 2023. 2, 3, 5, 6

[43] Liyan Sun, Zhiwen Fan, Yue Huang, Xinghao Ding, and John Paisley. Compressed sensing MRI using a recursive dilated network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 8

[44] Yubao Sun, Jiwei Chen, Qingshan Liu, Bo Liu, and Guodong Guo. Dual-path attention network for compressed sensing image reconstruction. *IEEE Transactions on Image Processing*, 29:9482–9495, 2020. 2, 6

[45] Lishun Wang, Miao Cao, Yong Zhong, and Xin Yuan. Spatial-temporal transformer for video snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):9072–9089, 2023. 1

[46] Weixi Wang, Ji Li, and Hui Ji. Self-supervised deep image restoration via adaptive stochastic gradient langevin dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1979–1988, 2022. 2

[47] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16133–16142, 2023. 4

[48] Jingfen Xie, Jian Zhang, Yongbing Zhang, and Xiangyang Ji. PUERT: Probabilistic under-sampling and explicable reconstruction network for CS-MRI. *IEEE Journal of Selected Topics in Signal Processing*, 16(4):737–749, 2022. 8

[49] Ting Xie, Licheng Liu, and Lina Zhuang. Plug-and-play priors for multi-shot compressive hyperspectral imaging. *IEEE Transactions on Image Processing*, 32:5326–5339, 2023. 1

[50] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 5

[51] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(3):521–538, 2020. 8

[52] Dongjie Ye, Zhangkai Ni, Hanli Wang, Jian Zhang, Shiqi Wang, and Sam Kwong. CSformer: Bridging convolution and transformer for compressive sensing. *IEEE Transactions on Image Processing*, 32:2827–2842, 2023. 2, 5, 6

[53] Di You, Jian Zhang, Jingfen Xie, Bin Chen, and Siwei Ma. COAST: Controllable arbitrary-sampling network for com-

pressive sensing. *IEEE Transactions on Image Processing*, 30:6066–6080, 2021. 3, 6

[54] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. FastMRI: An open dataset and benchmarks for accelerated MRI. *arXiv preprint arXiv:1811.08839*, 2018. 8

[55] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference Curves and Surfaces*, pages 711–730. Springer, 2012. 6, 7

[56] Zhiyuan Zha, Bihan Wen, Xin Yuan, Joey Tianyi Zhou, Jiantao Zhou, and Ce Zhu. Triply complementary priors for image restoration. *IEEE Transactions on Image Processing*, 30:5819–5834, 2021. 1

[57] Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1828–1837, 2018. 2, 6, 8

[58] Jian Zhang, Chen Zhao, and Wen Gao. Optimization-inspired compact deep compressive sensing. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):765–774, 2020. 3, 5, 6

[59] Jian Zhang, Zhenyu Zhang, Jingfen Xie, and Yongbing Zhang. High-Throughput deep unfolding network for compressive sensing mri. *IEEE Journal of Selected Topics in Signal Processing*, 16(4):750–761, 2022. 8

[60] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging*, 20(2): 023016, 2011. 7

[61] Xinyuan Zhang, Xin Yuan, and Lawrence Carin. Nonlocal low-rank tensor factor analysis for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8232–8241, 2018. 1

[62] Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. AMP-Net: Denoising-based deep unfolding for compressive image sensing. *IEEE Transactions on Image Processing*, 30:1487–1500, 2021. 3, 6

[63] Hao Zheng, Faming Fang, and Guixu Zhang. Cascaded dilated dense network with two-step data consistency for mri reconstruction. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 8