AMS-Net: Adaptive Multi-Scale Network for Image Compressive Sensing

Kuiyuan Zhang, Zhongyun Hua, *Member, IEEE*, Yuanman Li, Yongyong Chen, Yicong Zhou, *Senior Member, IEEE*

***Abstract*—Recently, deep convolutional neural networks have been applied to image compressive sensing (CS) to improve reconstruction quality while reducing computation cost. Existing deep learning-based CS methods can be divided into two classes: sampling image at single scale and sampling image across multiple scales. However, these existing methods treat the image low-frequency and high-frequency components equally, which is an obstruction to get a high reconstruction quality. This paper proposes an adaptive multi-scale image CS network in wavelet domain called AMS-Net, which fully exploits the different importance of image low-frequency and high-frequency components. First, the discrete wavelet transform is used to decompose an image into four sub-bands, namely the low-low (LL), low-high (LH), high-low (HL), and high-high (HH) sub-bands. Considering that the LL sub-band is more important to the final reconstruction quality, the AMS-Net allocates it a larger sampling ratio, while allocating the other three sub-bands a smaller one. Since different blocks in each sub-band have different sparsity, the sampling ratio is further allocated block-by-block within the four sub-bands. Then a dual-channel scalable sampling model is developed to adaptively sample the LL and the other three sub-bands at arbitrary sampling ratios. Finally, by unfolding the iterative reconstruction process of the traditional multi-scale block CS algorithm, we construct a multi-stage reconstruction model to utilize multi-scale features for further improving the reconstruction quality. Experimental results demonstrate that the proposed model outperforms both the traditional and state-of-the-art deep learning-based methods.**

***Index Terms*—Compressive sensing, convolutional neural networks, discrete wavelet transform, block compressive sampling**

I. Introduction

Compressive sensing (CS) is a new signal acquisition technique [1], [2]. It can sample a signal using a measurement matrix at a far lower ratio than the requirement of Nyquist sampling theory and can construct the original signal from its measurements [3]. Because the sampling and compressing operations is performed simultaneously, the CS shows good performance in data acquisition, storage, and transmission [4], [5]. Researchers have applied the CS theory to many fields such as magnetic resonance imaging [6], video compression [7], image coding [8] and snapshot compressive imaging [9].

There are two main tasks in studying the CS: signal sampling and signal reconstruction. For traditional CS schemes, although their measurement matrices can satisfy the restricted isometry property [10], they have low reconstruction performance due to the lack of adaptability for signals with different features. Besides, their reconstruction algorithms [11]–[13] were developed by considering the prior knowledge of the original signal and applying iterative non-linear optimization approaches for reconstruction. These reconstruction algorithms have some disadvantages such as low reconstruction quality, high complexity, and blocking artifacts.

Recently, some deep learning-based image CS models have been proposed to solve the limitations of traditional CS methods. These models implement the reconstruction process using the convolutional neural networks (CNNs) to replace the time-consuming optimization process, and their measurement matrices can be learned adaptively [14]–[17]. Besides, some works first decompose an image using different multi-scale decomposition approaches and then sample these decomposed images across scales to sufficiently utilize the multi-scale features [18]–[20]. Compared with traditional iterative optimization-based algorithms, these deep learning-based methods can significantly improve the reconstruction quality and reduce the time complexity of reconstruction. However, these methods directly sample images at single scale [14]–[17] or across multiple scales [18]–[20] by equally treating the image low-frequency and high-frequency components with the same resources. This limits the reconstruction quality, since the low-frequency components are more important to the reconstruction quality of an image than the high-frequency components, especially at a low sampling ratio [12].

In this paper, we propose a wavelet domain-based adaptive multi-scale image CS network (AMS-Net) by considering the different importance of the image low-frequency and high-frequency components. It can adaptively sample images in the multi-scale domain. Specifically, the discrete wavelet transform (DWT) is used to decompose an image from the spatial domain to the frequency domain, generating the low-low (LL), low-high (LH), high-low (HL), and high-high (HH)

This work was supported in part by the National Natural Science Foundation of China under Grants 62071142 and 62106063, by the Guangdong Basic and Applied Basic Research Foundation under Grants 2021A1515011406, 2022A1515010645, and 2022A1515010819, by the Shenzhen College Stability Support Plan under Grants GXWD20201230155427003-20200824210638001 and GXWD20201230155427003-20200824113231001, and in part by the Foundation for Science and Technology Innovation of Shenzhen under Grant RCBS20210609103708014. (Corresponding author: Zhongyun Hua)

Kuiyuan Zhang, Zhongyun Hua and Yongyong Chen are with School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China (e-mail: zkyhitsz@gmail.com; huazyum@gmail.com; YongyongChen.cn@gmail.com).

Yuanman Li is with College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: yuanmanli@szu.edu.cn)

Yicong Zhou is with Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

sub-bands. Considering that the LL sub-band is the low-frequency components and is more important to the reconstruction quality than the other three sub-bands, we design a dual-channel scalable sampling model that assigns the LL sub-band a larger sampling ratio while assigning the other three sub-bands a smaller one. As different blocks in the sub-bands have different sparsity, we apply a linear sampling resource assigning (LSRA) strategy to these two channels to further adjust the sampling ratios according to block sparsity. By unfolding the reconstruction process of the multi-scale block CS (MS-BCS) strategy [12] using CNN, we construct a multi-stage reconstruction architecture. In each stage, the reconstruction model first applies a dual-channel projection operation in the wavelet domain on the reconstructed image block-by-block, and then implements a full-image denoising operation in the spatial domain to remove the noise and blocking artifacts.

The contributions and novelty of this paper are summarized as follows:

- We present a wavelet domain-based adaptive multi-scale image CS network, which is the first deep learning-based CS network that considers the different importance of image low-frequency and high-frequency components.
- We develop an LSRA strategy to adjust the sampling ratios according to block sparsity and propose a dual-channel scalable sampling model to perform the adaptive sampling tasks at arbitrary sampling ratios.
- Unfolding the traditional reconstruction strategy with CNN, we design a multi-stage reconstruction architecture to exploit the multi-scale features, which can further improve the reconstruction quality.
- We conduct a comprehensive evaluation and the results show that our model outperforms both the traditional and state-of-the-art deep learning-based CS methods.

The rest of this paper is organized as follows. Section II introduces the CS theory and the traditional MS-BCS algorithm. Section III reviews the deep learning-based CS models. Section IV presents the network structure of our AMS-Net. Section V evaluates the performance of the proposed model and compares it with state-of-the-art methods. Section VI gives a conclusion of this paper.

II. Prerequisite Knowledge

A. Compressive Sensing

For a given measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{m\times n}$ with $n >> m$, the CS theory specifies that the sparse signal $\mathbf{x} \in \mathbb{R}^{n\times 1}$ can be sampled as $\mathbf{y} = \mathbf{\Phi x}$ and it can be reconstructed using some reconstruction algorithms [11], [13] from the measurements $\mathbf{y} \in \mathbb{R}^{m\times 1}$. The CS methods can be divided into two classes: single-scale CS methods and multi-scale CS methods. The single-scale CS methods sample and reconstruct images in the spatial domain, while the multi-scale CS methods sample images in the multi-level decomposition domain or reconstruct images using multi-scale information.

B. Multi-Scale Block Compressive Sensing

When processing a two-dimensional (2D) image, the size of the measurement matrix will be quite large if transforming the 2D image into a one-dimensional (1D) signal. To solve this problem, block-based sampling (BCS) methods [12], [21] have been widely used to separately sample each non-overlapping image block with constant size $B \times B$.

Let x_i represent the 1D vector transformed from the i-th block of the image $\mathbf{X}$, and the sampling process is written as

$$y_i = \mathbf{\Phi}_B x_i, \tag{1}$$

where $\mathbf{\Phi}_B \in \mathbb{R}^{n_B\times B^2}$ is a measurement matrix and $y_i \in \mathbb{R}^{n_B\times 1}$ is the measurements of x_i. The sampling ratio (sr) is $sr = n_B/B^2$. The authors in [21] proposed a BCS-SPL method that directly samples image blocks in the spatial domain and uses projected Landweber reconstruction mechanism to reduce blocking artifacts. To utilize the multi-scale features of images, the authors in [12] further proposed the MS-BCS to deploy the BCS-SPL in the multi-level decomposition domain.

1) Multi-Scale Sampling: To sample an image in multi-scale domain, one should first decompose the image to produce L levels of wavelet decomposition coefficients. Then each block $\tilde{\mathbf{x}}_{l,s,i}$ of a sub-band s at level l is sampled as

$$\mathbf{y}_{l,s,i} = \mathbf{\Phi}_l \tilde{\mathbf{x}}_{l,s,i}, \tag{2}$$

where $s \in \{LH, HL, HH\}$, $1 \leq l \leq L$ and $\mathbf{\Phi}_l$ means the measurement matrix for the level l. Since a lower level of the decomposition coefficients is more important to the final reconstruction quality, the sampling resources are adaptively adjusted for each level l.

2) Multi-Scale Reconstruction: First, the initial estimation of each image block is generated by linearly mapping the measurements, which is shown as

$$\hat{x}_{l,s,i}^{(0)} = \mathbf{\Phi}_l^* y_{l,s,i}, \tag{3}$$

where $\mathbf{\Phi}_l^*$ represents the linear mapping matrix and it is the pseudo-inverse matrix of $\mathbf{\Phi}_l$. Then the reconstruction process is an iterative process and each iterative step contains the following two operations.

- Projection in the wavelet domain. This operation is to find a vector that is closer than the current vector $\hat{x}_{l,s,i}^{(t)}$ on the hyperplane $\mathbf{H} = \{\hat{x}_{l,s,i} : \mathbf{\Phi}_l \hat{x}_{l,s,i} = y_{l,s,i}\}$. The projection operation is defined as

$$\begin{aligned} \hat{x}_{l,s,i}^{(t+1)} &= \hat{x}_{l,s,i}^{(t)} + \mathbf{\Phi}_l^* \left(y_{l,s,i} - \mathbf{\Phi}_l \hat{x}_{l,s,i}^{(t)} \right) \\ &= \hat{x}_{l,s,i}^{(t)} + \hat{x}_{l,s,i}^{(0)} - \mathbf{\Phi}_l^* \mathbf{\Phi}_l \hat{x}_{l,s,i}^{(t)}. \end{aligned} \tag{4}$$

- Deblocking and Denoising. Block-based reconstruction and projection may generate some blocking artifacts and noises. Thus, a non-linear mapping

$$\hat{\mathbf{X}}^{(t+1)} = \mathcal{H}\left(\hat{\mathbf{X}}^{(t+1)\prime}\right) \tag{5}$$

is employed to remove the blocking artifacts and noises of the reconstructed image $\hat{\mathbf{X}}^{(t+1)\prime}$, where $\mathcal{H}$ donates the Wiener filtering in the spatial domain or the thresholding operation in the frequency domain, and $\hat{\mathbf{X}}^{(t+1)\prime}$ is the full image generated by applying inverse DWT (IDWT) to the reconstructed blocks $\hat{x}_{l,s,i}^{(t+1)}$.

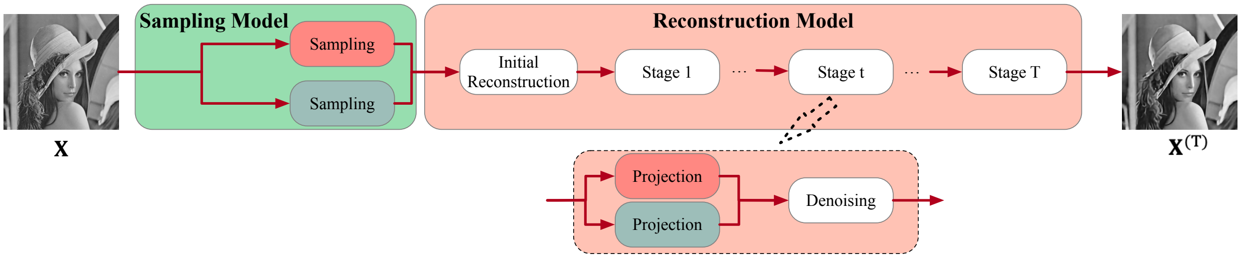


Fig. 1. The framework of the AMS-Net. The $\mathbf{X}$ donates the original image and the $\hat{\mathbf{X}}^{(T)}$ represents the reconstructed image.

III. Deep Learning-Based CS Methods

Since deep learning-based methods can show great effect in image restoration tasks, they were used to solve the signal sampling and reconstruction problems [14], [15], [22]–[24]. The authors in [22] proposed the first model that uses an stacked denoising autoencoder to reconstruct the image patches from the sampled measurements. Later, the authors in [23] constructed a deep CNN architecture to implement the non-iterative reconstruction process from the sampled measurements. However, the two works use constant measurement matrices to sample images. This limits the reconstruction quality because the constant measurement matrices may lose the generality and adaptability for images with different features. Thus, some models were proposed to simultaneously train the sampling network and reconstruction network [14], [15], [24]. For example, the CSNet* [24] and CSNet^{+} [14] use learnable convolutional layers without overlapping to mimic the sampling process. Then the measurements are transmitted into the reconstruction network to get the final reconstructed image. All these models are single-scale CS methods and they apply sampling and reconstruction in the spatial domain.

To utilize the multi-scale features in the deep learning-based CS methods, the models LAPRAN [25] and SCSNet [16] divide CS measurements into multiple levels. The CS measurements at the lowest level are used to generate the initial reconstruction of the image, while those at higher levels are progressively fused to generate a high-frequency image residual for enhancing the quality of the initial reconstruction. Besides, the model MSRNet [17] applies three parallel channels with different convolution kernel sizes in the reconstruction stage to fuse multi-scale features. The models in [16], [17], [25] exploit multi-scale features in the reconstruction stage but still sample images at single scale in the spatial domain. To directly learn multi-scale features in the sampling stage, the model MS-DCSNet [18] samples the decomposed images block-by-block across multiple wavelet scales. Moreover, it integrates a multi-level wavelet convolutional neural network to further utilize the multi-scale information in the reconstruction stage. Using the same reconstruction architecture but different multi-scale decomposition methods with the MS-DCSNet [18], the models DoC-DCS [19], SS-DCI [20] and P-DCI [20] also develop multi-scale CS architectures to utilize multi-scale features in both the sampling and reconstruction stages. Specifically, the multi-scale decomposition methods in the models DoC-DCS [19], SS-DCI [20] and P-DCI [20] are the difference of convolution, scale-space and pyramid, respectively.

These existing single-scale and multi-scale deep learning-based CS methods equally treat the image low-frequency and high-frequency components by sampling them using the same resources. However, the low-frequency components are usually more important to the final reconstruction quality than the high-frequency components [12]. Sampling the image low-frequency and high-frequency components equally may lead to a low reconstruction quality, especially at low sampling ratios.

IV. AMS-Net

In this section, we present the AMS-Net and Fig. 1 shows its framework. It is an end-to-end structure that contains a dual-channel sampling model and a multi-stage reconstruction model. To demonstrate the sampling and reconstruction processes, we assume that the input image is $\mathbf{X} \in \mathbb{R}^{2H\times 2W\times 1}$, and the image block size is $B \times B$.

A. Sampling Model

The framework of the sampling model is shown as Fig. 2. The original image $\mathbf{X}$ is decomposed by DWT and then sampled by a dual-channel scalable sampling model with LSRA strategy.

1) DWT Decomposition: We use the DWT to decompose the original image $\mathbf{X}$ into four sub-figures, namely the LL, LH, HL and HH sub-bands, and each sub-band is of size $\mathbb{R}^{H\times W\times 1}$. Because the image low-frequency and high-frequency components have different importance to the final reconstruction quality, we use different sampling ratios to sample them. Specifically, we set the LL sub-band as the first channel and denote it as $\mathbf{P} \in \mathbb{R}^{H\times W\times 1}$, and set the LH, HL and HH sub-bands as the second channel and denote it as $\mathbf{Q} \in \mathbb{R}^{H\times W\times 3}$.

Assuming that the target sampling ratio of the input image $\mathbf{X}$ is sr, we set the sampling ratio in sub-band $\mathbf{P}$ as sr_p and that in sub-bands $\mathbf{Q}$ as sr_q. The sr, sr_p and sr_q should satisfy the following equation:

$$\begin{aligned} sr &= \frac{sr_p \times HW + sr_q \times 3HW}{2H \times 2W} \\ &= \frac{1}{4} sr_p + \frac{3}{4} sr_q \end{aligned}. \tag{6}$$

Let the allocation ratio $ar = \frac{sr_p \times HW}{sr \times 2H \times 2W} = \frac{sr_p}{4sr}$ represent the ratio of the measurement number in $\mathbf{P}$ to the total measurement number in $\mathbf{X}$. Then one can obtain that $sr_p = 4sr \times ar$ and $sr_q = \frac{4sr\times(1-ar)}{3}$. When sr is very low, the low-frequency components are critical, and we set a large ar to reconstruct more image approximations. When sr is high,

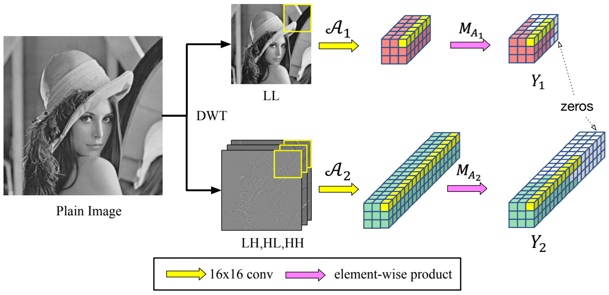


Fig. 2. The sampling model of the AMS-Net. The original image $\mathbf{X}$ is decomposed by DWT to generate LL, LH, HL and HH sub-bands. A dual-channel architecture is used to assign sampling resources for sampling the LL sub-band and the other three sub-bands. The $\mathbf{Y}_1$ and $\mathbf{Y}_2$ denote the measurements of the LL sub-band and the other three sub-bands, respectively.

Algorithm 1 The LSRA strategy.

```
Input: Saliency information $\mathbf{v}$, sampling ratio $sr'$, block size $B$, image size $(H, W, C)$
Output: The measurement number $\mathbf{m}_{ij}$ for each image block
$pixels\_per\_block = B \times B \times C$
$total = \lfloor sr' \times H \times W \times C \rfloor$
$base = \lceil sr'/3 * pixels\_per\_block \rceil$
$rest = total - (base \times \frac{H}{B} \times \frac{W}{B})$
for $i = 1 : \frac{H}{B}$ do
    for $j = 1 : \frac{W}{B}$ do
        $\mathbf{m}_{ij} = base + \lfloor rest \times \mathbf{v}_{ij} \rfloor$
    end for
end for
```

the high-frequency components become important, and we set a relatively small ar to reconstruct more image details. Besides, from the global view, the low-frequency components are more important to the final reconstruction quality than the high-frequency components, since the image approximations contain most of the image information [12]. Thus, the ar should be not smaller than 0.25 to ensure that $sr_p \geq sr_q$.

2) LSRA Strategy: The sampling resources are linearly assigned according to the block sparsity in $\mathbf{P}$ and $\mathbf{Q}$, respectively. Specifically, the saliency information [26] is used to measure the block sparsity. The larger saliency information indicates that the corresponding image block is less sparse and thus should be assigned more sampling resources.

For $\mathbf{P}$, its saliency map [26] is defined as:

$$\begin{aligned} D &= \mathrm{sign}(C_t(\mathbf{P})), \\ F &= \mathrm{abs}\left(C_t^{-1}(D)\right), \\ \mathbf{S} &= G * F^2, \end{aligned} \tag{7}$$

where C_t and C_t^{-1} are the 2D discrete cosine transform and its inverse transform, respectively, and G is a 2D Gaussian low-pass filter. To smooth the calculation results, we further normalize the values of $\mathbf{S}$ into [0, 1]. Assume that $\mathbf{P}_{ij} \in \mathbb{R}^{B \times B \times 1}$ represents an image block in $\mathbf{P}$, where $i \in \{1, 2, \cdots, h\}$ and $j \in \{1, 2, \cdots, w\}$ with $h = H/B$ and $w = W/B$. The quantified saliency information $\mathbf{v}_{ij}$ of each image block $\mathbf{P}_{ij}$ is calculated as

$$\mathbf{v}_{ij} = \sum_{s \in \mathbf{S}_{ij}} s / \sum_{s \in \mathbf{S}} s, \tag{8}$$

where $\mathbf{S}_{ij}$ is the corresponding region of $\mathbf{P}_{ij}$ in $\mathbf{S}$, and $\sum_{i=1}^{h} \sum_{j=1}^{w} \mathbf{v}_{ij} = 1$. For simplicity, we assume that the sub-bands $\mathbf{P}$ and $\mathbf{Q}$ have the same saliency information distribution within blocks.

Algorithm 1 shows the linear sampling resource assigning process. We first initialize the measurements for each image block to avoid the final number of measurements of some blocks being too small, and then linearly assign the rest sampling resources according to the saliency information. Using Algorithm 1 with two group of inputs $\{\mathbf{v}, sr_p, B, (H, W, 1)\}$ and $\{\mathbf{v}, sr_q, B, (H, W, 3)\}$, we can obtain the measurement number m_{ij}^{p} for each image block $\mathbf{P}_{ij}$ and measurement number m_{ij}^{q} for each image block $\mathbf{Q}_{ij}$, respectively. Note that the actual total measurements are less than the target total measurements because of the floor function in Algorithm 1. Then we use these reserved sampling resources to save the saliency information $\mathbf{v}$ and include it in the actual sampling results. Due to the errors that may be caused by the floor operation, the actual sampling ratio (sr_a) and target sampling ratio (sr_t) may have some slight difference. Table I lists the average errors between the sr_a and sr_t for images in Set11 at different sampling ratios.

To illustrate the advantage of our adaptive sampling, we present the measurement allocation maps in the LL coefficients of two images "Cameraman" and "Parrots" in Fig. 3. The image block size is set as 16 × 16. Each value indicates the measurement number to the image block and a larger value means more allocated measurement resources. It can be seen that the image blocks with more details are allocated more measurement resources. For example, the camera in the image "Cameraman" and the eyes in the image "Parrots" have more details, and thus their related image blocks are allocated more measurement resources.

3) Dual-Channel Scalable Sampling: Following existing deep learning-based CS methods [14], [15], we also use the BCS strategy to reduce the memory and computational burden rather than directly sampling the whole image. When applying different sampling ratios to different image blocks, one usually constructs a measurement matrix for each sampling ratio and this will highly increase the parameter number of the model. To solve the problem, we apply a scalable deep compressive sensing method that can perform different sampling tasks with only one measurement matrix. The convolution operation is used to mimic the compressive sampling process on the image block-by-block. Note that our method samples image blocks without overlapping, and thus the kernels of each convolution layer have the same size as the image block.

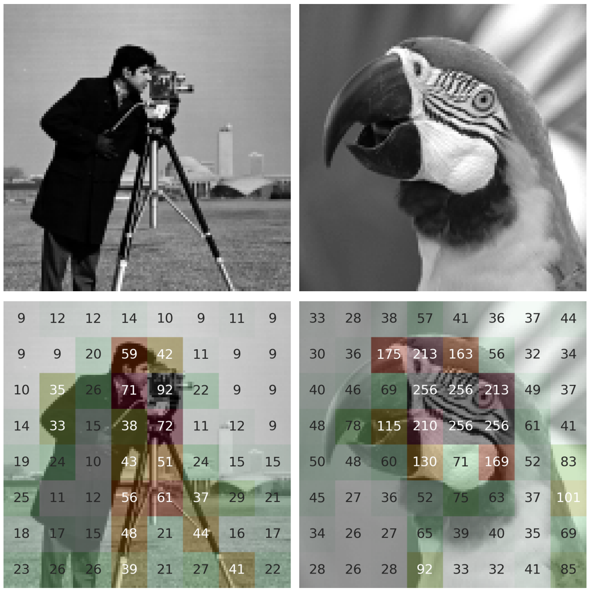
Specifically, to sample $\mathbf{P}$ with an adaptive sampling ratio, we replace the measurement matrix with a convolutional layer $\mathcal{A}_1$ with weights $w_{\mathcal{A}_1} \in \mathbb{R}^{B^2 \times (B \times B \times 1)}$, and the scalable sampling process is defined as

$$\mathbf{Y}_1 = (\mathcal{A}_1 * \mathbf{P}) \odot \mathbf{M}_{\mathcal{A}_1}, \tag{9}$$

where $*$ denotes the convolution operation with stride size $B \times B$, $\odot$ represents the element-wise multiplication, $\mathbf{M}_{\mathcal{A}_1} \in \mathbb{R}^{h \times w \times B^2}$ is the mask to control the activities of measurements

TABLE I
THE AVERAGE ERRORS BETWEEN THE ACTUAL SAMPLING RATIO (sr_a) AND TARGET SAMPLING RATIO (sr_t) ON SET11 DATASET.

Target sr	0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5
$sr_a - sr_t$	$+9e^{-4}$	$+8e^{-5}$	$+6e^{-6}$	$-6e^{-6}$	$+3e^{-5}$	$+1e^{-4}$	$+1e^{-4}$	$-2e^{-4}$
$(sr_a - sr_t)/sr_t$	9.58%	0.23%	0.006%	-0.01%	0.01%	0.05%	0.03%	-0.02%



9	12	12	14	10	9	11	9
9	9	20	59	42	11	9	9
10	35	26	71	92	22	9	9
14	33	15	38	72	11	12	9
19	24	10	43	51	24	15	15
25	11	12	56	61	37	29	21
18	17	15	48	21	44	16	17
23	26	26	39	21	27	41	22

(a) Sampling ratio = 0.1

33	28	38	57	41	36	37	44
30	36	175	213	163	56	32	34
40	46	69	256	256	213	49	37
48	78	115	210	256	256	61	41
50	48	60	130	71	169	52	83
45	27	36	52	75	63	37	101
34	26	27	65	39	40	35	69
28	26	28	92	33	32	41	85

(b) Sampling ratio = 0.3

Fig. 3. Illustration of the measurement allocation map. The first row shows the LL coefficients of the two images and the second row shows their measurement allocation maps.

for each image block and $\mathbf{Y}_1 \in \mathbb{R}^{h \times w \times B^2}$ is the final sampling results. For each sub-tensor of the mask $\mathbf{M}_{\mathcal{A}_1}$, $\mathbf{M}_{\mathcal{A}_1}[i,j,1:m_{ij}^p] = 1$ and other elements are zeros. Thus, for the measurements of the image block $\mathbf{P}_{ij}$, only $\mathbf{Y_1}[i,j,1:m_{ij}^p]$ are valid measurements and $\mathbf{Y_1}[i,j,m_{ij}^p+1:B^2] = 0$.

Similarly, using a convolutional layer $\mathcal{A}_2$ with weights $w_{\mathcal{A}_2} \in \mathbb{R}^{3B^2 \times (B \times B \times 3)}$, the sampling process for $\mathbf{Q}$ is calculated as

$$\mathbf{Y_2} = (\mathcal{A}_2 * \mathbf{Q}) \odot \mathbf{M}_{\mathcal{A}_2}, \tag{10}$$

where $\mathbf{M}_{\mathcal{A}_2} \in \mathbb{R}^{h \times w \times 3B^2}$ is the corresponding sampling mask and $\mathbf{Y_2} \in \mathbb{R}^{h \times w \times 3B^2}$ is the sampling results of $\mathbf{Q}$. For each sub-tensor of the mask $\mathbf{M}_{\mathcal{A}_2}$, $\mathbf{M}_{\mathcal{A}_2}[i,j,1:m_{ij}^q] = 1$ and other elements are zeros. Thus, for the measurements of the image block $\mathbf{Q}_{ij}$, only $\mathbf{Y_2}[i,j,1:m_{ij}^q]$ are valid measurements and $\mathbf{Y_2}[i,j,m_{ij}^q+1:3B^2] = 0$.

Formally, the whole sampling process can be expressed as

$$\mathbf{Y}_1, \mathbf{Y}_2 = f_{samp}(\mathbf{X}, sr, \mathcal{A}_1, \mathcal{A}_2). \tag{11}$$

To ensure that each image block has at least one measurement, $\frac{hw}{HW}$ and $\frac{hw}{3HW}$ are the smallest sampling ratios that the sampling model can perform on $\mathbf{P}$ and $\mathbf{Q}$, respectively. Thus, the target sampling ratio of the input image should be not smaller than $\frac{hw}{2HW}$. When the target sampling ratio is smaller than this threshold, we set it as the threshold.

B. Reconstruction Model

We construct the reconstruction model by unfolding the reconstruction process of the traditional MS-BCS method. In the traditional MS-BCS method, the matrix $\mathbf{\Phi}_l^*$ in Eq. (3) and Eq. (4) is the pseudo-inverse matrix of $\mathbf{\Phi}_l$. However, since the measurement matrix in our method is generated using learning strategy, the measurement matrix is not orthogonal in the training process, and the real-time calculation of pseudo-inverse will interrupt the backward propagation of gradients. Thus, for simplicity, we set the matrix $\mathbf{\Phi}_l^*$ learnable in Eq. (3) and Eq. (4) to ensure that the model can be trained.

To improve the denoising and deblocking ability, we use a feed-forward learnable denoising block to replace the traditional denoising method in Eq. (5). This can also solve the problem that traditional denoising algorithms cannot propagate backward gradients in the training process. The full-image denoising and deblocking operation is expressed as

$$\hat{\mathbf{X}}^{(t+1)} = \mathbf{X}^{(t+1)\prime} + \mathcal{D}(\mathbf{X}^{(t+1)\prime}), \tag{12}$$

where $\mathcal{D}$ indicates the denoising block containing convolutional layers. The residual learning is used to speed up the training process and improve the reconstruction performance.

The reconstruction model is shown as Fig. 4 and it contains an initial reconstruction module f_{init} and a deep reconstruction module f_{deep}. The original image can be reconstructed from its dual-channel measurements.

1) Initial Reconstruction: Our module f_{init} applies the initial reconstruction illustrated in Eq. (3) on the sampled measurements in the wavelet domain. Similar to the sampling module, the learnable convolutional layers are used to implement the linear mapping matrix to get the initial reconstruction. Specifically, the measurements $\mathbf{Y}_1$ in the first channel is connected to a convolutional layer $\mathcal{A}_1^*$ containing B^2 kernels of size $1 \times 1 \times B^2$. A tensor of size $\mathbb{R}^{h \times w \times B^2}$ is generated by convolving these B^2 kernels on the $\mathbf{Y}_1$ with stride size 1×1. By reshaping the tensor into hw feature maps of size $B \times B \times 1$ and concatenating them into an $H \times W \times 1$ feature map, the initial reconstruction $\hat{\mathbf{P}}^{(0)}$ of $\mathbf{P}$ in the first channel can be generated. Using the similar operation, the initial reconstruction $\hat{\mathbf{Q}}^{(0)}$ of $\mathbf{Q}$ in the second channel can be obtained by connecting the measurements $\mathbf{Y}_2$ to a convolutional layer $\mathcal{A}_2^*$ with $3B^2$ kernels of size $1 \times 1 \times 3B^2$.

The process of the initial reconstruction can be expressed as

$$\begin{cases} \hat{\mathbf{P}}^{(0)} = \Xi(\mathcal{A}_1^* * \mathbf{Y}_1) \\ \hat{\mathbf{Q}}^{(0)} = \Xi(\mathcal{A}_2^* * \mathbf{Y}_2), \end{cases} \tag{13}$$

where Ξ is the reshaping and concatenation operations, and the linear mapping in Eq. (3) is separately implemented using convolution operations with $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$. The final initial

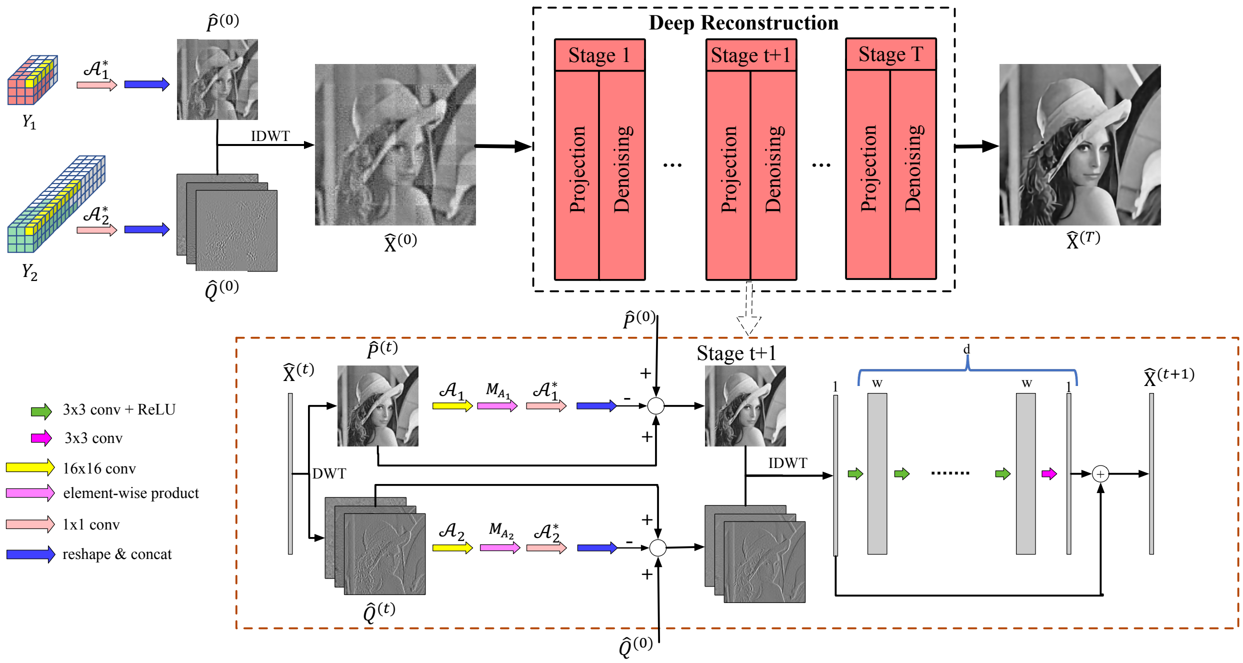


Fig. 4. The reconstruction model of the AMS-Net. The initial reconstruction image $\hat{\mathbf{X}}^{(0)}$ is generated by first linearly mapping the measurements $\mathbf{Y}_1$ and $\mathbf{Y}_2$ and then performing IDWT to the mapping results. Then the deep reconstruction module processes $\hat{\mathbf{X}}^{(0)}$ for T stages to obtain the final reconstructed image $\hat{\mathbf{X}}^{(T)}$. A dual-channel projection operation and a full-image denoising operation are used in each stage, in which the dual-channel projection is performed in the wavelet domain while the denoising operation is performed in spatial domain using d convolutional layers.

reconstruction $\hat{\mathbf{X}}^{(0)}$ is generated by applying the inverse DWT to the combination of $\hat{\mathbf{P}}^{(0)}$ and $\hat{\mathbf{Q}}^{(0)}$.

2) Deep Reconstruction: The deep reconstruction process is performed to the initial reconstruction result $\hat{\mathbf{X}}^{(0)}$ to improve its quality. By unfolding the iterative process in traditional MS-BCS reconstruction algorithm, we divide the module f_{deep} into T stages. Each stage alternatively implements the projection (see Eq. (4)) in the wavelet domain and the full-image denoising (see Eq. (12)) in the spatial domain.

At the $(t+1)$-th stage, let $\hat{\mathbf{X}}^{(t)}$ be the reconstruction result of the previous stage. To perform the projection in Eq. (4) in the wavelet domain, we first decompose the image $\hat{\mathbf{X}}^{(t)}$ into sub-images $\hat{\mathbf{P}}^{(t)}$ and $\hat{\mathbf{Q}}^{(t)}$, in which $\hat{\mathbf{P}}^{(t)}$ represents the LL sub-band and $\hat{\mathbf{Q}}^{(t)}$ represents the stacking of the LH, HL and HH sub-bands. The projection operations for $\hat{\mathbf{P}}^{(t)}$ and $\hat{\mathbf{Q}}^{(t)}$ are expressed as

$$\begin{cases}\hat{\mathbf{P}}^{(t+1)} = \hat{\mathbf{P}}^{(t)} + \hat{\mathbf{P}}^{(0)} - \Xi(\boldsymbol{\mathcal{A}}_1^* * ((\boldsymbol{\mathcal{A}}_1 * \hat{\mathbf{P}}^{(t)}) \odot \mathbf{M}_{\mathcal{A}_1})) \\ \hat{\mathbf{Q}}^{(t+1)} = \hat{\mathbf{Q}}^{(t)} + \hat{\mathbf{Q}}^{(0)} - \Xi(\boldsymbol{\mathcal{A}}_2^* * ((\boldsymbol{\mathcal{A}}_2 * \hat{\mathbf{Q}}^{(t)}) \odot \mathbf{M}_{\mathcal{A}_2})).\end{cases} \tag{14}$$

By applying the IDWT on the combination of $\hat{\mathbf{P}}^{(t+1)}$ and $\hat{\mathbf{Q}}^{(t+1)}$, we can obtain the projection result $\hat{\mathbf{X}}^{(t+1)\prime}$ of the $(t+1)$-th stage.

For the full-image denoising operation in Eq. (12), we use the denoising block to remove the noise of $\hat{\mathbf{X}}^{(t+1)\prime}$ to get the $(t+1)$-th reconstruction result $\hat{\mathbf{X}}^{(t+1)}$. In each denoising block, the first $(d-1)$ layers generate w feature maps using 3×3 convolution and the ReLU activation function, and the last layer generates a feature map using 3×3 convolution without activation function. The architecture of the denoising block is designed by removing the batch normalization layer between the convolutional layer and ReLU activation function in the feed-forward denoising model [27].

After finishing all the T stages, the final reconstructed image $\hat{\mathbf{X}}^{(T)}$ can be generated. Formally, let $\mathbb{S}_{\boldsymbol{\Theta}} = \{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \cdots, \boldsymbol{\Theta}_T\}$ represent the learnable weights of all the denoising blocks. Then the total process of the reconstruction model can be expressed as

$$\hat{\mathbf{X}}^{(T)} = f_{rec}(\mathbf{Y}_1, \mathbf{Y}_2, \mathbb{S}_{\boldsymbol{\Theta}}, \boldsymbol{\mathcal{A}}_1, \boldsymbol{\mathcal{A}}_1^*, \boldsymbol{\mathcal{A}}_2, \boldsymbol{\mathcal{A}}_2^*). \tag{15}$$

C. Loss Function

The forward propagation process of the AMS-Net is shown in Algorithm 2, where the whole model can be trained end-to-end and all the parameters are adaptively learned through backward propagation.

The mean squared error (MSE) is widely used as the loss function in many state-of-the-art deep learning-based CS methods [14], [15], [20], since it is differentiable and has faster convergence speed than other loss functions such as l_1 loss and perceptual loss [28]. To obtain the fastest convergence speed and best reconstruction performance, we also use MSE as the loss function to calculate the difference between the original image and its corresponding reconstructed image. Given N training images $\{\mathbf{X}_i\}_{i=1}^N$, the loss function is calculated by

$$\begin{aligned}\mathcal{L} = \frac{1}{2N}\sum_{i=1}^{N} \| f_{rec}(f_{samp}(\mathbf{X}_i, sr_i, \boldsymbol{\mathcal{A}}_1, \boldsymbol{\mathcal{A}}_2),& \\ \mathbb{S}_{\boldsymbol{\Theta}}, \boldsymbol{\mathcal{A}}_1, \boldsymbol{\mathcal{A}}_2, \boldsymbol{\mathcal{A}}_1^*, \boldsymbol{\mathcal{A}}_2^*) - \mathbf{X}_i\|_2^2&\end{aligned}, \tag{16}$$

where sr_i is the target sampling ratio of the i-th original image $\mathbf{X}_i$ in the training set. Using this loss function, the AMS-

Algorithm 2 The forward propagation of the AMS-Net.

```
Input: $\mathbf{X}, sr, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_1^*, \mathcal{A}_2^*, \mathbb{S}_{\mathbf{\Theta}}, T$
Output: $\hat{\mathbf{X}}^{\mathbf{T}}$
procedure SAMPLING($\mathbf{X}, sr, \mathcal{A}_1, \mathcal{A}_2$)
    $sr_p, sr_q = SR\_Assign(sr)$
    $\mathbf{P}, \mathbf{Q} = DWT(X)$
    $\mathbf{M}_{\mathcal{A}_1} = LSRA(sr_p, \mathbf{P})$
    $\mathbf{M}_{\mathcal{A}_2} = LSRA(sr_q, \mathbf{Q})$
    $\mathbf{Y}_1 = (\mathcal{A}_1 * \mathbf{P}) \odot \mathbf{M}_{\mathcal{A}_1}$
    $\mathbf{Y}_2 = (\mathcal{A}_2 * \mathbf{Q}) \odot \mathbf{M}_{\mathcal{A}_2}$
    return $\mathbf{Y}_1, \mathbf{Y}_2$
end procedure
procedure RECONSTRUCTION($\mathbf{Y}_1, \mathbf{Y}_2, \mathcal{A}_1^*, \mathcal{A}_2^*, \mathbb{S}_{\mathbf{\Theta}}, T$)
    $\hat{\mathbf{P}}^{(0)} = \Xi(\mathcal{A}_1^* * \mathbf{Y}_1)$
    $\hat{\mathbf{Q}}^{(0)} = \Xi(\mathcal{A}_2^* * \mathbf{Y}_2)$
    $\hat{\mathbf{X}}^{(\mathbf{0})} = IDWT(\hat{\mathbf{P}}^{(0)}, \hat{\mathbf{Q}}^{(0)})$
    $t = 0$
    while $(t+1) \leq T$ do                    ▷ In the $(t+1)th$ stage
        $\hat{\mathbf{P}}^{(t)}, \hat{\mathbf{Q}}^{(t)} = DWT(\hat{\mathbf{X}}^{\mathbf{t}})$
        $\hat{\mathbf{P}}^{(t+1)} = \hat{\mathbf{P}}^{(t)} + \hat{\mathbf{P}}^{(0)} - \Xi(\mathcal{A}_1^* * ((\mathcal{A}_1 * \hat{\mathbf{P}}^{(t)}) \odot \mathbf{M}_{\mathcal{A}_1}))$
        $\hat{\mathbf{Q}}^{(t+1)} = \hat{\mathbf{Q}}^{(t)} + \hat{\mathbf{Q}}^{(0)} - \Xi(\mathcal{A}_2^* * ((\mathcal{A}_2 * \hat{\mathbf{Q}}^{(t)}) \odot \mathbf{M}_{\mathcal{A}_2}))$
        $\hat{\mathbf{X}}^{(t+1)\prime} = IDWT(\hat{\mathbf{P}}^{(t+1)}, \hat{\mathbf{Q}}^{(t+1)})$
        $\hat{\mathbf{X}}^{(t+1)} = \hat{\mathbf{X}}^{(t+1)\prime} + \mathcal{D}(\hat{\mathbf{X}}^{(t+1)\prime}, \mathbf{\Theta}_{(t+1)})$
        $t = t + 1$
    end while
    return $\hat{\mathbf{X}}^{(t)}$
end procedure
```

Net can quickly minimize the error between the reconstructed image and the original image in the training process.

V. Experimental Results and Performance Evaluations

In this section, we first design the training method to evaluate the performance of the AMS-Net, and then compare it with state-of-the-art schemes in terms of reconstruction quality, visual effect, and model complexity.

A. Experiment Settings

The image block size in the sampling model is set as $B = 16$ and the number of reconstruction stage T is set as 10. The width w and depth d of the denoising block in each reconstruction stage are set as 64 and 5, respectively. For the target sampling ratio sr in the original image, the allocation ratio ar is set as $ar = 0.95$ when $sr \leq 0.05$, and set as $ar = \max(0.25, 0.95 - sr)$ when $sr > 0.05$. The parameter studies about the stage T, the width w and depth d, and the allocation ratio ar are shown in Section V-D.

1) Training: Our experiment uses the same database with other models [14], [16] to build the training dataset. Specifically, the training set is constructed using the training set (200 images) and testing set (200 images) of the BSD500 [29]–[31] dataset. We randomly flip and rotate these original images to extend our dataset, which is the same with the CSNet$^+$ [14]. Then we randomly crop $400 \times 224 = 89600$ gray-scale sub-images of size 128×128 as the training set.

To speed up the training process and improve the reconstruction performance, we use a pre-trained denoising block to initialize the weights of all the denoising blocks in the proposed model. Specifically, following the settings in [27], we pretrain a single denoising block for blind Gaussian denoising task with random noise levels $\sigma \in [0, 55]$. The training epochs are 50 and the batch size is 64. The Adam optimization strategy [32] with a learning rate of 0.0001 is employed to optimize the parameters of the denoising block. Our AMS-Net is trained with batch size 1 for 10 epochs. For every mini-batch, a random sampling ratio within $[0.01, 0.50]$ is set for the used image. The optimizer is the Adam optimizer, and the learning rate decays linearly from 0.0001 to 0.00001 on these ten epochs. The TensorFlow framework is used to implement the proposed model, and all the experiments are implemented on a workstation with two GeForce RTX3090 GPUs and an Intel(R) Core(TM) i9-10920X CPU. It takes about one hour for one epoch in the training process.

2) Testing: Four widely used benchmark datasets, including the Set5 (5 images) [35], Set11 (11 images) [23], Set14 (14 images) [36], and the validating dataset of BSD500 called BSD100 (100 images), are used for evaluation. These widely used datasets contain different image features and can fairly reflect the reconstruction performance of different models. To keep consistency with the training process, we also convert all the color images in these four datasets into gray-scale images and use the obtained gray-scale images as the testing images.

The competing methods contain six traditional CS methods, including the DAMP [33], BCS-FOCUSS [13], TV [11], MH [34], BCS-SPL [21] and MS-BCS [12], five single-scale deep learning-based CS models, including the ReconNet [23], ISTA-Net [37], CSNet$^+$ [14], ISTA-Net^{++} [38] and AMP-Net [15], and six multi-scale deep learning-based CS models, including the SCSNet [16], MSRNet [17], MS-DCSNet [18], DoC-DCS [19], P-DCI [20] and SS-DCI [20]. The reconstruction performance is tested by calculating the Peak Signal-to-Noise Ratio (PSNR) [39] and Structural Similarity Index (SSIM) [40] between the reconstructed image and original image. A higher PSNR or SSIM score means the better reconstruction quality. Besides, the visual quality is also compared by showing the reconstruction results of different methods.

B. Comparison with Traditional CS Methods

First, we compare our AMS-Net with six traditional CS methods, DAMP [33], BCS-FOCUSS [13], TV [11], MH [34], BCS-SPL [21] and MS-BCS [12]. All codes of the traditional CS methods are directly downloaded from the authors' websites, and the default parameters are used for evaluation. Two kinds of measurement matrices are used, including the random measurement matrices (i.e. Gaussian matrices) and learned measurement matrices. The learned measurement matrices are generated as follows. First, two convolutional layers are used to sample the image and recover the initial reconstruction results from the sampled measurements, respectively. Then, using these two learnable layers, we construct a small model and train it using the training set. By converting the learned weights of the sampling layer into matrix, we can get the learned measurement matrices.

TABLE II
PSNR AND SSIM COMPARISONS OF OUR AMS-NET WITH THE TRADITIONAL CS METHODS UNDER MULTIPLE SAMPLING RATIOS (sr).

Set5 (PSNR/SSIM)

sr Methods	0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5	Avg.
DAMP-R [33]	6.54/0.0491	12.08/0.2859	17.73/0.4424	23.24/0.6431	25.12/0.8092	31.65/0.8860	34.46/0.9197	30.97/0.9310	22.72/0.6208
DAMP-L [33]	10.97/0.2967	19.92/0.6062	24.20/0.7293	28.77/0.8426	32.44/0.9181	32.32/0.9413	36.23/0.9557	33.39/0.9605	27.28/0.7813
FOCUSS-R [13]	17.09/0.4531	20.27/0.5737	22.71/0.6439	25.86/0.7356	29.40/0.8267	31.62/0.8741	33.46/0.9045	35.12/0.9262	26.94/0.7422
FOCUSS-L [13]	21.88/0.5520	25.01/0.6863	26.89/0.7630	29.59/0.8481	32.87/0.9136	35.33/0.9429	37.47/0.9595	39.37/0.9699	31.05/0.8294
MH-R [34]	15.70/0.3936	21.21/0.5791	23.93/0.6726	27.44/0.7923	30.63/0.8686	32.60/0.9016	34.25/0.9245	35.77/0.9411	27.69/0.7592
MH-L [34]	19.12/0.5047	25.29/0.7142	27.30/0.7840	30.08/0.8656	33.28/0.9252	35.72/0.9508	37.86/0.9660	39.94/0.9759	31.07/0.8358
TV-R [11]	17.09/0.4135	18.81/0.4788	21.13/0.5550	24.63/0.6939	28.24/0.8182	30.83/0.8795	33.04/0.9172	35.11/0.9423	26.11/0.7123
TV-L [11]	21.40/0.5295	24.53/0.6862	26.34/0.7678	28.89/0.8531	31.65/0.9143	33.63/0.9419	35.45/0.9588	37.27/0.9704	29.89/0.8277
BCS-SPL-R [21]	17.50/0.4705	20.44/0.5744	22.58/0.6492	25.61/0.7376	29.00/0.8219	31.17/0.8675	33.00/0.8984	34.70/0.9212	26.75/0.7426
BCS-SPL-L [21]	22.20/0.5712	25.59/0.7126	27.53/0.7870	30.41/0.8685	33.71/0.9250	36.05/0.9502	38.14/0.9652	40.16/0.9750	31.72/0.8443
MS-BCS-R [12]	-/-	24.63/0.6613	26.27/0.7188	29.46/0.8372	32.65/0.8919	34.90/0.9402	36.15/0.9432	37.40/0.9491	-/-
AMS-Net-R	20.47/0.5529	24.71/0.7050	27.83/0.8013	31.61/0.8778	35.34/0.9314	37.69/0.9526	39.67/0.9643	41.51/0.9716	32.35/0.8446
AMS-Net	**23.00/0.6261**	**27.97/0.7987**	**30.37/0.8563**	**33.41/0.9066**	**36.64/0.9445**	**39.01/0.9616**	**41.25/0.9722**	**43.28/0.9793**	**34.37/0.8807**

Set14 (PSNR/SSIM)

sr Methods	0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5	Avg.
DAMP-R [33]	6.14/0.0424	12.06/0.2656	17.49/0.3987	21.95/0.5557	25.65/0.6995	28.20/0.7800	30.63/0.8361	33.41/0.8813	21.94/0.5574
DAMP-L [33]	11.37/0.2864	18.92/0.5323	22.97/0.6463	26.20/0.7605	29.47/0.8535	31.77/0.8982	33.70/0.9244	35.53/0.9427	26.24/0.7305
FOCUSS-R [13]	17.89/0.4269	20.01/0.5073	21.48/0.5623	23.91/0.6534	26.76/0.7569	28.76/0.8191	30.46/0.8617	32.07/0.8942	25.17/0.6852
FOCUSS-L [13]	20.85/0.4847	23.30/0.5997	24.60/0.6685	26.73/0.7697	29.57/0.8638	31.82/0.9088	33.81/0.9350	35.69/0.9515	28.30/0.7727
MH-R [34]	16.26/0.3594	20.40/0.5074	22.33/0.5869	25.36/0.7011	28.28/0.8070	30.17/0.8572	31.74/0.8896	33.37/0.9169	25.99/0.7032
MH-L [34]	18.54/0.4402	23.49/0.6133	24.86/0.6803	27.04/0.7811	29.99/0.8737	32.14/0.9158	34.44/0.9426	36.38/0.9585	28.36/0.7757
TV-R [11]	17.22/0.3809	18.78/0.4363	20.46/0.4993	23.11/0.6173	26.15/0.7406	28.36/0.8137	30.38/0.8643	32.34/0.9020	24.60/0.6568
TV-L [11]	20.55/0.4738	23.12/0.5994	24.40/0.6681	26.43/0.7685	29.17/0.8599	31.14/0.9044	33.01/0.9321	34.89/0.9512	27.84/0.7697
BCS-SPL-R [21]	17.72/0.4347	19.77/0.5056	21.22/0.5598	23.67/0.6498	26.52/0.7494	28.55/0.8116	30.26/0.8558	31.91/0.8904	24.95/0.6821
BCS-SPL-L [21]	21.06/0.5003	23.66/0.6179	25.05/0.6875	27.30/0.7866	30.30/0.8744	32.47/0.9159	34.43/0.9408	36.39/0.9571	28.83/0.7850
MS-BCS-R [12]	-/-	22.97/0.5729	24.29/0.6343	26.55/0.7430	29.27/0.8274	31.25/0.8945	32.70/0.9064	34.04/0.9198	-/-
AMS-Net-R	19.89/0.4864	23.64/0.6117	25.73/0.6861	28.58/0.7795	32.08/0.8697	34.39/0.9122	36.21/0.9354	37.77/0.9490	29.79/0.7788
AMS-Net	**22.20/0.5470**	**25.63/0.6720**	**27.38/0.7350**	**30.12/0.8182**	**33.34/0.8945**	**35.59/0.9298**	**37.54/0.9499**	**39.23/0.9619**	**31.38/0.8135**

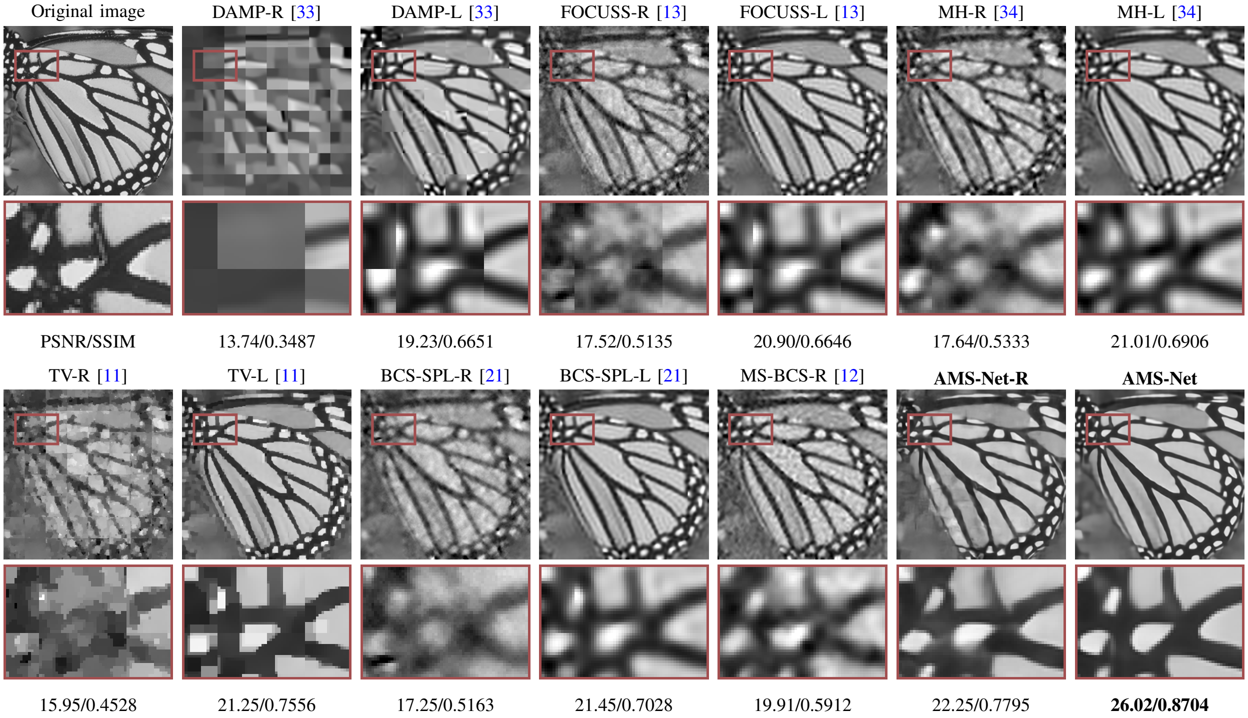


Fig. 5. Reconstructed results of image "butterfly" by our AMS-Net and the traditional CS methods under sampling ratio 0.05.

1) Reconstruction Quality: Table II shows the comparison results of our AMS-Net with traditional CS methods on Set5 and Set14, respectively. For traditional CS methods, the suffix '-R' denotes that the measurement matrices are randomly generated while the suffix '-L' means that the measurement matrices are learned. For our AMS-Net, the suffix '-R' means that the measurement matrices are non-learnable in the training process. The sampling ratio varies from 0.01 to 0.5. Note that the MS-BCS [12] cannot work at a sampling ratio lower than 0.02 and it samples image in the multi-level wavelet domain with different block sizes. Thus, its measurement matrices are difficult to learn. We only test it with random measurement matrices and do not provide the test result at sampling ratio 0.01 for MS-BCS.

As can be see from Table II, when using the learned measurement matrices to replace the random measurement matrices, the reconstruction performance of all the traditional CS methods can be greatly improved, and the average PSNR scores are improved about $3 \sim 5$ dB for every traditional CS methods. Meanwhile, using the learned measurement matrices, the average PSNR scores of our AMS-Net can be improved about $2 \sim 3$ dB than using the random measurement matrices. Thus, it is obvious that the learned measurement matrices are much more effective than these randomly generated measurement matrices. From Table II, we can see that the PSNR and SSIM scores of our AMS-Net are significantly larger than those of the traditional methods. Thus, our AMS-Net can outperform all the traditional CS methods under all the sampling ratios. For example, in the comparison on Set5, when the traditional CS methods use random measurement matrices, the average PSNR score of the AMS-Net is 11.65 dB, 7.43 dB, 6.68 dB, 8.26 dB, 7.62 dB larger than the PSNR scores of the DAMP [33], BCS-FOCUSS [13], MH [34], TV [11], and BCS-SPL [21], respectively, while the corresponding average SSIM score is 0.2599, 0.1385, 0.1215, 0.1684 and 0.1261 larger than these methods, respectively. Besides, when using the random measurement matrices, our AMS-Net still outperforms all other traditional CS methods.

2) Visual Effect: To show the high reconstruction performance of our model, we compare the visual quality of our model with the traditional CS methods using the image “butterfly” from Set5. The comparison results are shown in Fig. 5, where the used sampling ratio is 0.05. As can be seen, there is obvious blocking artifacts in the reconstructed images by the traditional CS methods. However, our model can eliminate these blocking artifacts. This is because the denoising and deblocking modules in our model are learnable, and they can produce better denoising and deblocking performance than the traditional denoising methods. Meanwhile, by comparing the two versions of traditional CS methods, we can find that all the traditional CS methods can obtain better visual quality when using the learned measurement matrices. This visual quality results further verify the high performance of the measurement matrix learning strategy.

C. Comparison with Deep Learning-Based CS Methods

These competing deep learning-based methods, including the ReconNet [23], ISTA-Net [37], CSNet$^+$ [14], ISTA-Net^{++} [38], AMP-Net [15], SCSNet [16], MSRNet [17], DoC-DCS [19], P-DCI [20], SS-DCI [20], MS-DCSNet [18], are all developed recently with high performance. For the ReconNet [23], ISTA-Net [37], CSNet$^+$ [14], SCSNet [16], MSRNet [17], DoC-DCS [19], P-DCI [20], SS-DCI [20] and MS-DCSNet [18], we re-train them using the training set strictly following the details in the original literature. For the ISTA-Net^{++} [38] and AMP-Net [15], we download their pre-trained models from the authors’ websites and run these models on the testing datasets to get the results.

1) Reconstruction Quality: Table III shows the comparison results of different networks on the four testing datasets. As can be seen, the AMS-Net can obtain significantly higher PSNR scores than these deep learning-based CS methods over all sampling ratios and achieve larger SSIM scores in most cases. For example, the average PSNR score of the AMS-Net on Set11 is 2.12 dB, 4.24 dB, 7.06 dB, 2.11 dB, 4.9 dB, 1.47 dB, 2.11 dB, 4.78 dB, 1.80 dB, 2.26 dB, 1.75 dB and 2.14 dB larger than the PSNR scores of the CSNet$^+$ [14], ISTA-Net [37], ReconNet [23], AMP-Net [15], ISTA-Net^{++} [38], SCSNet [16], MSRNet [17], DoC-DCS [19], P-DCI [20], SS-DCI [20] and MS-DCSNet [18], respectively, while the corresponding average SSIM score on Set11 is 0.0160, 0.0789, 0.1376, 0.0088, 0.1114, 0.0148, 0.0689, 0.0098, 0.0142, 0.0081, 0.0148 larger than these methods, respectively.

Note that the SSIM score is calculated by a sliding Gaussian window in the spatial domain. However, our proposed model performs the sampling and projection operations in the wavelet domain, and this may slightly affect the SSIM scores of our method. Although our model gets a little smaller SSIM scores than the AMP-Net [15] under some sampling ratios, it can obtain the best SSIM score on average among all the methods. This indicates that our model can get the best reconstruction quality. Besides, compared with AMP-Net [15], our AMS-Net has some other important advantages. First, it can achieve much larger PSNR and SSIM scores when the sampling ratio is very low, which indicates that it shows much better performance in heavy compression tasks (see Fig. 6 and Table III). Second, our AMS-Net is a scalable method while AMP-Net [15] is not. This indicates that our method can sample images at arbitrary sampling ratios with only one-time training. However, a single AMP-Net [15] model can sample images only at a fixed sampling ratio. Thus, to perform tasks with n sampling ratios, the AMP-Net [15] should train n models and thus has much more parameters than our AMS-Net, as shown in Table VII.

2) Visual Effect: We use the image “Parrots” in Set11 dataset to show the visual quality of the reconstructed images by different deep learning-based CS methods and Fig. 6 shows the reconstruction results under the sampling ratio 0.05. As can be seen, the reconstructed images by the ReconNet [23], ISTA-Net [37], ISTA-Net^{++} [38] and MSRNet [17] have obvious blocking artifacts. This is because these methods focus on the reconstruction of each image block individually, and don’t consider the correlations between adjacent blocks. On the contrary, there aren’t obvious blocking artifacts in the reconstructed images generated by other models, because these models all apply denoising and deblocking operations to the full-image on the reconstruction stage. As shown in Fig. 6, the image reconstructed by our AMS-Net contains more details and sharp edges of the original image than the images reconstructed by other methods. Thus, compared with other methods, our AMS-Net has stronger ability to remove blocking artifacts and reconstruct images with higher visual quality.

TABLE III
PSNR AND SSIM COMPARISONS OF OUR AMS-NET WITH THE DEEP LEARNING-BASED CS METHODS UNDER MULTIPLE SAMPLING RATIOS (sr).

Set5 (PSNR/SSIM)

sr Methods	0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5	Avg.
CSNet$^+$ [14]	22.78/0.6094	26.49/0.7665	28.60/0.8316	31.54/0.8956	34.86/0.9396	36.95/0.9581	38.67/0.9692	40.63/0.9774	32.56/0.8684
ISTA-Net [37]	19.32/0.4669	22.61/0.6138	24.82/0.7130	28.31/0.8272	32.56/0.9083	35.17/0.9400	37.24/0.9574	39.31/0.9701	29.92/0.7996
ReconNet [23]	18.44/0.4660	21.91/0.5863	23.57/0.6409	26.06/0.7332	28.97/0.8185	31.25/0.8738	32.71/0.8956	34.46/0.9226	27.17/0.7421
AMP-Net [15]	22.42/0.6183	26.56/0.7789	28.62/0.8380	32.10/0.9024	35.55/0.9427	37.87/0.9608	39.77/0.9717	41.60/**0.9793**	33.06/0.8740
ISTA-Net^{++} [38]	12.60/0.3081	16.67/0.4658	25.41/0.7539	30.22/0.8702	33.94/0.9250	36.27/0.9485	38.13/0.9621	39.95/0.9725	29.15/0.7758
SCSNet [16]	22.78/0.6136	26.47/0.7654	28.69/0.8331	31.58/0.8968	34.86/0.9404	36.87/0.9586	38.80/0.9701	40.81/0.9786	32.61/0.8696
MSRNet [17]	22.06/0.5723	25.17/0.7019	27.34/0.7888	29.03/0.8265	32.07/0.8946	33.58/0.9117	33.89/0.9114	34.85/0.9239	29.75/0.8164
P-DCI [20]	22.72/**0.6280**	26.21/0.7719	28.51/0.8320	31.47/0.8942	34.62/0.9375	36.49/0.9541	38.13/0.9638	39.78/0.9732	32.24/0.8693
SS-DCI [20]	22.05/0.6054	26.67/0.7864	29.02/0.8456	31.89/0.9017	35.00/0.9407	37.38/0.9604	39.01/0.9699	40.99/0.9781	32.75/0.8735
MS-DCSNet [18]	22.70/0.6218	26.37/0.7691	28.33/0.8297	31.19/0.8928	34.62/0.9344	36.75/0.9540	38.40/0.9648	40.29/0.9741	32.33/0.8676
DoC-DCS [19]	22.06/0.5995	26.63/0.7870	28.90/0.8457	31.98/0.9016	34.94/0.9419	37.38/0.9600	39.05/0.9701	40.72/0.9768	32.71/0.8728
AMS-Net	**23.00**/0.6261	**27.97/0.7987**	**30.37/0.8563**	**33.41/0.9066**	**36.64/0.9445**	**39.01/0.9616**	**41.25/0.9722**	**43.28/0.9793**	**34.37/0.8807**

Set14 (PSNR/SSIM)

sr Methods	0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5	Avg.
CSNet$^+$ [14]	21.59/0.5309	24.30/0.6477	25.71/0.7125	27.96/0.8028	31.01/0.8846	33.14/0.9232	35.01/0.9454	36.97/0.9606	29.46/0.8010
ISTA-Net [37]	18.56/0.4122	21.24/0.5293	23.07/0.6122	25.74/0.7237	29.19/0.8354	31.70/0.8913	33.90/0.9246	35.99/0.9479	27.43/0.7346
ReconNet [23]	18.01/0.4175	20.65/0.5056	22.03/0.5586	24.02/0.6419	26.40/0.7404	28.43/0.8123	29.79/0.8494	31.47/0.8869	25.10/0.6766
AMP-Net [15]	21.65/0.5435	24.62/0.6629	26.13/0.7267	28.77/**0.8182**	32.00/0.8938	34.38/**0.9301**	36.34/**0.9512**	38.22/**0.9650**	30.26/0.8114
ISTA-Net^{++} [38]	12.32/0.2506	16.09/0.3863	23.35/0.6445	27.32/0.7714	30.72/0.8628	33.07/0.9074	34.98/0.9341	36.73/0.9525	26.82/0.7137
SCSNet [16]	21.61/0.5323	24.31/0.6478	25.79/0.7139	27.99/0.8041	30.98/0.8855	33.15/0.9239	35.08/0.9466	37.08/0.9619	29.50/0.8020
MSRNet [17]	21.10/0.5026	23.47/0.6074	24.98/0.6822	26.31/0.7388	28.93/0.8415	30.41/0.8726	30.80/0.8783	31.66/0.8934	27.21/0.7521
P-DCI [20]	21.65/0.5411	24.06/0.6494	25.67/0.7114	28.05/0.8031	30.87/0.8842	32.85/0.9189	34.55/0.9395	35.68/0.9492	29.17/0.7996
SS-DCI [20]	21.22/0.5251	24.45/0.6641	25.98/0.7264	28.27/0.8143	31.18/0.8896	33.55/0.9278	35.27/0.9464	37.42/0.9625	29.67/0.8070
MS-DCSNet [18]	21.57/0.5396	24.20/0.6470	25.71/0.7123	27.86/0.8011	30.98/0.8818	33.04/0.9188	34.96/0.9432	36.73/0.9563	29.38/0.8000
DoC-DCS [19]	21.29/0.5204	24.43/0.6651	25.93/0.7264	28.34/0.8153	31.12/0.8909	33.40/0.9265	35.31/0.9474	37.08/0.9592	29.61/0.8064
AMS-Net	**22.20/0.5470**	**25.63/0.6720**	**27.38/0.7350**	**30.12/0.8182**	**33.34/0.8945**	**35.59**/0.9298	**37.54**/0.9499	**39.23**/0.9619	**31.38/0.8135**

Set11 (PSNR/SSIM)

sr Methods	0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5	Avg.
CSNet$^+$ [14]	21.06/0.5588	24.13/0.7169	26.06/0.7913	28.66/0.8644	32.10/0.9219	34.65/0.9509	36.78/0.9655	38.93/0.9763	30.30/0.8433
ISTA-Net [37]	18.16/0.4394	20.83/0.5731	22.90/0.6746	26.32/0.8010	30.55/0.8973	33.33/0.9343	35.63/0.9549	37.72/0.9687	28.18/0.7804
ReconNet [23]	17.44/0.4417	20.20/0.5446	21.66/0.6081	24.03/0.7076	26.88/0.8023	29.23/0.8631	30.83/0.8892	32.58/0.9172	25.36/0.7217
AMP-Net [15]	20.20/0.5581	24.11/0.7252	26.06/0.7987	29.40/0.8779	33.21/0.9334	36.03/0.9586	38.28/**0.9715**	40.33/**0.9804**	30.95/0.8505
ISTA-Net^{++} [38]	11.30/0.2375	14.81/0.3832	22.88/0.7048	28.34/0.8531	32.33/0.9217	34.86/0.9478	36.94/0.9628	38.73/0.9727	27.52/0.7479
SCSNet [16]	21.11/0.5616	24.13/0.7166	26.10/0.7939	28.66/0.8659	32.02/0.9232	34.63/0.9511	36.81/0.9663	39.04/0.9775	30.31/0.8445
MSRNet [17]	20.66/0.5287	23.15/0.6572	25.12/0.7496	26.71/0.7984	29.67/0.8746	31.29/0.8974	31.62/0.8975	32.93/0.9197	27.64/0.7904
P-DCI [20]	21.36/0.5747	24.04/0.7204	26.01/0.7931	28.88/0.8674	32.15/0.9242	34.44/0.9482	36.28/0.9615	38.11/0.9713	30.16/0.8451
SS-DCI [20]	20.86/0.5566	24.50/0.7411	26.50/0.8079	29.04/0.8741	32.38/0.9268	35.30/0.9560	37.30/0.9687	39.46/0.9787	30.67/0.8512
MS-DCSNet [18]	21.30/0.5734	24.15/0.7188	26.05/0.7927	28.58/0.8648	32.20/0.9215	34.63/0.9478	36.65/0.9627	38.68/0.9740	30.28/0.8445
DoC-DCS [19]	20.86/0.5475	24.54/0.7422	26.31/0.8041	29.19/0.8753	32.25/0.9273	35.26/0.9539	37.31/0.9685	39.20/0.9770	30.62/0.8495
AMS-Net	**21.65/0.5795**	**25.69/0.7426**	**27.91/0.8181**	**31.23/0.8867**	**34.99/0.9406**	**37.46/0.9599**	**39.37**/0.9702	**41.05**/0.9770	**32.42/0.8593**

BSD100 (PSNR/SSIM)

sr Methods	0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5	Avg.
CSNet$^+$ [14]	22.47/0.5150	24.37/0.6123	25.45/0.6742	27.25/0.7671	29.75/0.8615	31.81/0.9111	33.68/0.9405	35.59/0.9604	28.80/0.7803
ISTA-Net [37]	19.80/0.4215	21.86/0.5133	23.17/0.5825	25.14/0.6861	27.87/0.8013	29.95/0.8656	31.82/0.9070	33.68/0.9367	26.66/0.7143
ReconNet [23]	19.07/0.4242	21.48/0.4990	22.55/0.5452	24.06/0.6201	26.00/0.7155	27.79/0.7931	29.06/0.8350	30.66/0.8784	25.08/0.6638
AMP-Net [15]	22.29/0.5231	24.46/0.6231	25.54/0.6840	27.62/0.7787	30.34/0.8702	32.56/**0.9178**	34.58/**0.9466**	36.57/**0.9653**	29.25/0.7886
ISTA-Net^{++} [38]	13.38/0.2868	17.14/0.4134	23.22/0.6103	26.16/0.7265	28.85/0.8305	30.93/0.8869	32.81/0.9227	34.64/0.9472	25.89/0.7030
SCSNet [16]	22.47/0.5159	24.37/0.6116	25.48/0.6753	27.26/0.7688	29.76/0.8630	31.78/0.9118	33.70/0.9418	35.69/0.9621	28.81/0.7813
MSRNet [17]	22.22/0.4992	23.89/0.5858	25.03/0.6538	26.12/0.7148	28.29/0.8225	29.64/0.8609	30.09/0.8696	31.04/0.8914	27.04/0.7373
P-DCI [20]	22.58/0.5205	24.42/0.6174	25.43/0.6745	27.27/0.7679	29.71/0.8627	31.55/0.9070	33.29/0.9356	34.68/0.9509	28.62/0.7795
SS-DCI [20]	22.30/0.5095	24.56/0.6263	25.65/0.6868	27.48/0.7790	29.91/0.8665	32.13/0.9175	33.93/0.9441	35.94/0.9638	28.99/0.7867
MS-DCSNet [18]	22.56/0.5205	24.41/0.6161	25.45/0.6750	27.16/0.7662	29.75/0.8600	31.78/0.9097	33.64/0.9392	35.57/0.9595	28.79/0.7808
DoC-DCS [19]	22.24/0.5035	24.58/0.6270	25.61/0.6862	27.50/0.7790	29.92/0.8689	31.95/0.9147	33.93/0.9441	35.78/0.9623	28.94/0.7857
AMS-Net	**23.05/0.5281**	**25.38/0.6327**	**26.66/0.6949**	**28.78/0.7821**	**31.66/0.8710**	**33.92/0.9178**	**36.06**/0.9456	**38.22**/0.9624	**30.47/0.7918**

D. Discussion

In this section, we evaluate the influence of model parameters and model structure on the reconstruction performance of our AMS-Net.

1) Depth and Width of the Denoising Block: The number of parameters in our AMS-Net is influenced by the width w and depth d of the denoising block. Here, we investigate the parameter capacity by setting w as 32 and 64, and setting d ranging from 3 to 11. As can be seen from Table IV, when fixing d and changing w from 32 to 64, the average PSNR scores can be improved about 0.2 dB. Besides, the reconstruction quality gradually rises until d grows to 9. When $d > 9$, the model appears to be over-fitting and slightly declines in the reconstruction performance. Thus, to achieve a relatively high reconstruction quality and small parameter

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3198323

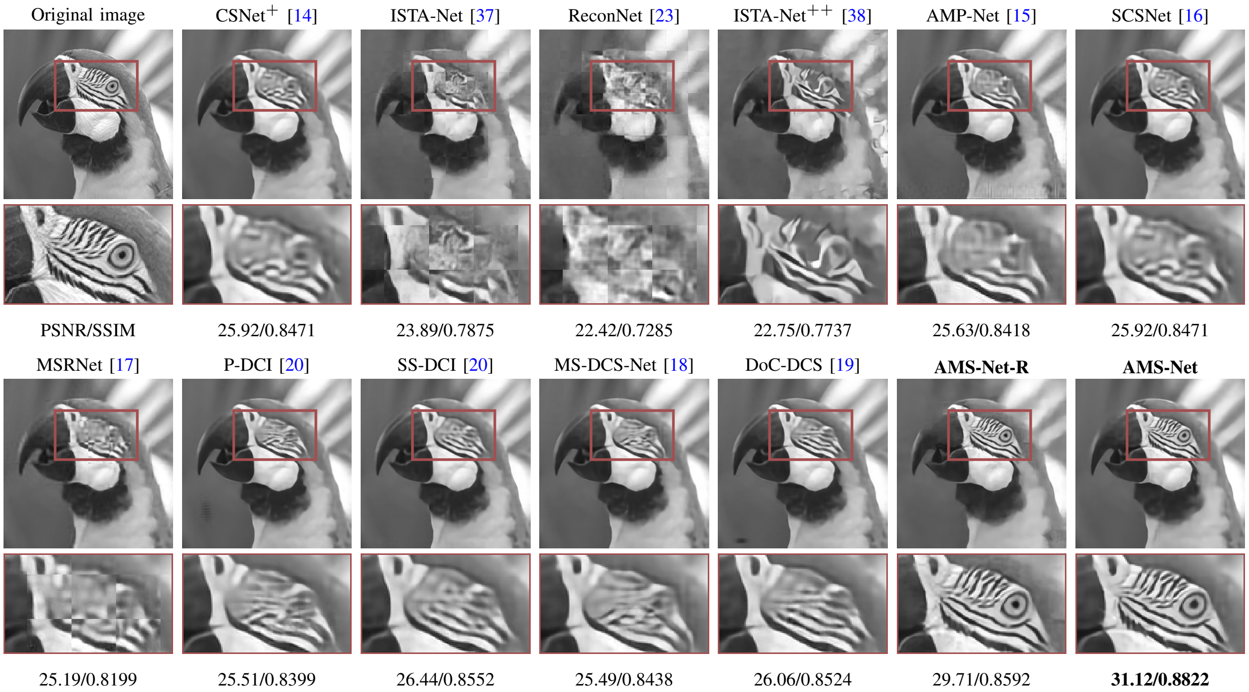


Fig. 6. Reconstructed results of image "Parrots" by our AMS-Net and the deep learning-based CS methods under sampling ratio 0.05.

TABLE IV
THE INFLUENCE OF THE NUMBER OF DENOISING STAGES T, THE WIDTH w AND DEPTH d OF THE DENOISING BLOCK, THE LSRA STRATEGY L, AND THE PROJECTION OPERATION P ON THE AVERAGE RECONSTRUCTION PSNR (DB) PERFORMANCE ON SET11 OF OUR PROPOSED AMS-NET.

L	P	d	T	Sampling Ratio ($w = 32/w = 64$)								Avg.
				0.01	0.03	0.05	0.1	0.2	0.3	0.4	0.5	
✓	✓	5	4	21.18/21.36	25.04/25.21	27.24/27.50	30.42/30.69	34.18/34.48	36.74/37.06	38.68/39.02	40.26/40.66	31.72/32.00
✓	✓	5	6	21.36/21.50	25.26/25.51	27.50/27.75	30.80/31.02	34.52/34.79	37.03/37.29	38.96/39.24	40.59/40.90	32.00/32.25
✓	✓	5	8	21.48/21.56	25.47/25.52	27.69/27.83	30.95/31.17	34.67/34.94	37.18/37.43	39.12/39.37	40.79/41.03	32.17/32.36
✓	✓	5	10	21.55/21.65	25.47/25.69	27.77/27.91	31.00/31.23	34.75/34.99	37.24/37.46	39.18/39.37	40.86/41.05	32.22/32.42
✓	✓	5	12	21.51/21.64	25.48/25.70	27.78/28.00	30.98/31.22	34.70/34.99	37.21/37.42	39.15/39.29	40.84/40.87	32.21/32.39
✓	✓	3	10	21.27/21.34	25.10/25.23	27.25/27.44	30.50/30.67	34.29/34.47	36.83/37.05	38.86/39.09	40.55/40.83	31.83/32.01
✓	✓	7	10	21.61/21.83	25.69/25.91	27.92/28.17	31.08/31.37	34.84/35.19	37.33/37.60	39.27/39.53	40.92/41.20	32.33/32.60
✓	✓	9	10	21.73/21.84	25.78/25.98	28.05/28.27	31.27/31.44	35.01/35.23	37.51/37.64	39.44/39.60	41.10/41.30	32.49/32.66
✓	✓	11	10	21.79/21.88	25.84/26.03	28.12/28.26	31.27/31.43	35.01/35.27	37.47/37.66	39.38/39.60	41.05/41.28	32.49/32.68
	✓	5	10	21.00/21.15	24.50/24.62	26.65/26.75	29.69/29.87	33.38/33.60	36.00/36.23	37.20/36.84	38.28/37.93	30.84/30.87
✓		5	10	20.65/20.79	24.11/24.33	26.07/26.39	28.85/29.12	31.99/32.30	34.42/34.71	36.28/36.57	37.50/37.84	29.98/30.26

capacity, we set $w = 64$ and $d = 5$ in our model.

2) Projection Operation: By utilizing the projection operation, our model can integrate the structure advantages of the traditional CS method with the powerful learning ability of CNN. Besides, the projection operation only reuses the learned measurement matrices and the linear mapping matrices to update the image blocks. Thus, it does not introduce any extra parameters. As shown in Table IV, when removing the projection operation, the average reconstruction PSNR scores of our model will degrade about $0.9 \sim 3$ dB on each sampling ratio. This demonstrates the effectiveness of the projection operation for image reconstruction.

3) Deep Reconstruction Module: To demonstrate the effectiveness of our deep reconstruction module, we construct two new models, the AMS-CSNet$^+$ and AMS-SCSNet, by replacing the deep reconstruction module of our AMS-Net with the deep reconstruction module in CSNet$^+$ [14] and SCSNet [16], respectively. Table V shows the comparison results on Set11 and the suffix "-NoProj" means that no projection operation is used in each reconstruction stage. As can be seen from Table V, the performance of the AMS-CSNet$^+$ and AMS-SCSNet are not superior compared to the original models. Besides, with more convolutional layers, the AMS-NoProj achieves similar reconstruction performance with the AMS-CSNet$^+$ and AMS-SCSNet, which indicates that the reconstruction performance cannot be improved by simply stacking more convolutional layers. Compared to other models, the proposed AMS-Net with projection operation can significantly improve the reconstruction performance. This demonstrates that the projection operation is very important for our adaptive multi-scale sampling model.

4) Reconstruction Stages: The reconstruction module of our AMS-Net has multiple reconstruction stages T. To evaluate the influence of T, we train the AMS-Net with multiple reconstruction stages ranging from 4 to 12. As can be seen from Table IV, the reconstruction quality can be improved with the increasing of T. However, when $T > 10$, the

TABLE V
THE INFLUENCE OF THE DEEP RECONSTRUCTION MODULE ON RECONSTRUCTION PERFORMANCE. THE TESTING SET IS SET11.

Methods	Sampling Ratio (PSNR)			
	0.01	0.1	0.3	0.5
CSNet$^+$ [14]	21.06	28.66	34.65	38.93
AMS-CSNet$^+$	20.59	28.99	34.93	38.14
SCSNet [16]	21.11	28.66	34.63	39.04
AMS-SCSNet	20.59	28.79	34.56	37.72
AMS-NoProj	20.79	29.12	34.71	37.84
AMS-Net	21.65	31.23	37.46	41.05

TABLE VI
THE INFLUENCE OF THE ALLOCATION RATIO ar ON RECONSTRUCTION QUALITY. THE TESTING SET IS SET11.

Max. ar	Sampling Ratio (PSNR/SSIM)			
	0.01	0.1	0.3	0.5
0.95	21.65/0.5795	31.23/0.8867	37.46/0.9599	41.05/0.9770
0.85	21.38/0.5667	31.24/0.8839	37.59/0.9585	40.73/0.9745
0.75	21.26/0.5650	31.19/0.8816	37.48/0.9566	39.86/0.9709
0.65	21.14/0.5609	31.03/0.8780	37.16/0.9544	38.29/0.9653

model appears to be over-fitting and slightly declines in the reconstruction performance. Thus, to achieve a relatively high reconstruction quality, we set $T = 10$ in our model.

5) LSRA Strategy: Considering that the block sparsities of the two groups of sub-bands in the wavelet domain are distributed unevenly, our model adaptively allocates sampling resources to different image blocks of the two groups of sub-bands using the LSRA strategy. As can be seen from the fourth row and the penultimate row of Table IV, the average reconstruction PSNR scores of the proposed model can improve about $0.5 \sim 3$ dB on each sampling ratio when using the LSRA strategy. This proves the effectiveness of the LSRA strategy on reconstruction performance.

6) Allocation Ratio: The allocation ratio ar is used to adaptively assign sampling resources according to the target sampling ratio. To test the influence of the ar on reconstruction performance, we train the AMS-Net with different settings of ar. The maximum value of ar is ranging from 0.65 to 0.95, and Table VI shows the comparison results. As can be seen, our model can get relatively high PSNR and SSIM scores when the maximum value of ar is 0.95. Thus, we set the maximum value of ar to 0.95 in the experiment settings.

E. Model Complexity

This section compares the model complexity of different CS methods in terms of parameter capacity and time complexity. The average running time is used to evaluate the actual reconstruction efficiency. Besides, we use the number of floating-point operations (FLOPs) [41] to quantify the time complexity, since it is commonly used in the time complexity comparison of deep learning-based models [6]. Note that the average running time and the number of FLOPs are calculated by reconstructing an image of size 256×256. The traditional CS methods are implemented in MATLAB software and run on CPU, while the deep learning-based methods are tested on GPU. Table VII shows the comparison results and the suffix "$T\cdot$" of our AMS-Net represents the number of reconstruction stages. As the DoC-DCS [19], P-DCI [20], SS-DCI [20] and MS-DCSNet [18] have the same reconstruction architecture, we only list the results of DoC-DCS [19] for comparison.

TABLE VII
MODEL COMPLEXITY COMPARISONS OF DIFFERENT CS METHODS WITH n SAMPLING RATIOS. THE PN$_1$ AND PN$_2$ ARE THE PARAMETER NUMBERS OF THE LEARNABLE MATRICES AND ALL OTHER CONVOLUTIONAL LAYERS, RESPECTIVELY, AND $SR = sr_1 + sr_2 + \cdots + sr_n$ IS THE SUM OF THE n SAMPLING RATIOS.

Methods	Scalable Sampling	PN$_1$ (M)	PN$_2$ (M)	FLOPs (G)	Time (s)
DAMP [33]	**True**	-	-	-	8.231
FOCUSS [13]	**True**	-	-	-	2.254
MH [34]	**True**	-	-	-	3.743
TV [11]	**True**	-	-	-	0.533
BCS-SPL [21]	**True**	-	-	-	1.506
MS-BCS [12]	**True**	-	-	-	0.910
ReconNet [23]	False	$1.05 \times SR$	$\mathbf{0.02} \times n$	**3.2**	**0.002**
CSNet$^+$ [14]	False	$2.10 \times SR$	$0.37 \times n$	6.82	0.005
ISTA-Net [37]	False	$1.05 \times SR$	$0.34 \times n$	37.12	0.008
AMP-Net [15]	False	$2.10 \times SR$	$0.34 \times n$	22.42	0.017
ISTA-Net^{++} [38]	**True**	-	0.76	49.45	0.015
SCSNet [16]	**True**	2.10	1.41	11.62	0.003
MSRNet [17]	False	$2.10 \times SR$	$2.05 \times n$	268.8	0.011
DoC-DCS [19]	False	$2.10 \times SR$	$16.29 \times n$	82.53	0.017
AMS-Net-$T4$	**True**	**1.31**	0.45	29.31	0.005
AMS-Net-$T6$	**True**	**1.31**	0.67	43.95	0.008
AMS-Net-$T8$	**True**	**1.31**	0.89	58.60	0.010
AMS-Net-$T10$	**True**	**1.31**	1.12	73.25	0.011

1) Parameter Capacity: As can be seen from Table VII, the SCSNet [16], ISTA-Net^{++} [38] and our AMS-Net use scalable sampling strategies and train only one model for multiple sampling ratios, which significantly reduce the parameter capacity and training time. In contrast, all other models need to train a model for each sampling ratio. However, a method should be available for different sampling ratios. Thus, when performing tasks with n sampling ratios, they need to train n models for these n sampling ratios, which greatly increases the parameter capacity and training time. As a result, for a normal task (e.g., $n > 10$), our AMS-Net can achieve the second smallest parameter capacity and only has larger parameter capacity than ISTA-Net^{++} [38]. Thus, our AMS-Net can achieve a small parameter capacity while keeping the best reconstruction quality.

2) Time Complexity: As shown in Table VII, it takes several seconds for traditional CS methods to reconstruct the images, while the average running times for deep learning-based methods are below 0.1 seconds when running on GPU. This is because the reconstruction process of traditional CS methods is implemented iteratively until achieving convergence or reaching the maximum iteration steps. Besides, compared with CPU, the GPU has advantages in calculation ability and parallel computation. This can heavily reduce the running time of the deep learning-based CS methods.

Among all the deep learning-based CS methods, the ReconNet [23] has the least FLOPs but the lowest reconstruction quality (see Table III), because it has fewer convolutional layers than other models. Our model has a comparative number of FLOPs compared to other models. Besides, with a small number of reconstruction stages, our AMS-Net can achieve a faster reconstruction speed and less FLOPs while maintaining a good reconstruction performance (see Table IV). In general, these deep learning-based CS methods have different time complexity. However, due to the strong computation efficiency of GPU, the slightly different running times of these methods do not make much sense. Since all running speeds are almost the same magnitude, the reconstruction quality is more important for a deep learning-based method. Our AMS-Net can achieve better reconstruction quality than other methods, especially at low sampling ratios (see Table III and Fig. 6).

VI. Conclusion

In this paper, we proposed a dual-channel deep network for adaptive multi-scale image CS, called AMS-Net, which fully exploits the different importance of the image low-frequency and high-frequency components in the wavelet domain. An original image is decomposed into four sub-images using the DWT, namely the LL, LH, HL and HH sub-bands. Since the LL sub-band is low-frequency components and more important to the reconstruction quality than the other three sub-bands, the AMS-Net allocates the LL sub-band a larger sampling ratio while allocating the other three sub-bands a smaller one. Considering the different sparsity in different blocks, an LSRA strategy is proposed to further adjust the sampling resources block-by-block on the two groups of sub-bands, respectively. Then a dual-channel scalable sampling model is developed to apply adaptive sampling tasks in the wavelet domain at arbitrary sampling ratios. Furthermore, by unfolding the reconstruction process of the traditional multi-scale block CS algorithm, we propose a multi-stage reconstruction architecture to utilize multi-scale features for further enhancing the reconstruction quality. Comparison results demonstrate that our AMS-Net can outperform the traditional CS methods and state-of-the-art deep leaning-based CS methods.

References

[1] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] Q. Jiang, S. Li, Z. Zhu, H. Bai, X. He, and R. C. de Lamare, "Design of Compressed Sensing System With Probability-Based Prior Information," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 594–609, 2020.

[4] B. Zhang, D. Xiao, and Y. Xiang, "Robust coding of encrypted images via 2D compressed sensing," *IEEE Transactions on Multimedia*, vol. 23, pp. 2656–2671, 2021.

[5] Y. Zhang, P. Wang, L. Fang, X. He, H. Han, and B. Chen, "Secure Transmission of Compressed Sampling Data Using Edge Clouds," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6641–6651, 2020.

[6] M. Ran, W. Xia, Y. Huang, Z. Lu, P. Bao, Y. Liu, H. Sun, J. Zhou, and Y. Zhang, "MD-Recon-Net: A Parallel Dual-Domain Convolutional Neural Network for Compressed Sensing MRI," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 1, pp. 120–135, 2021.

[7] S. Zheng, J. Chen, X.-P. Zhang, and Y. Kuo, "A New Multihypothesis Based Compressed Video Sensing Reconstruction System," *IEEE Transactions on Multimedia*, vol. 23, pp. 3577–3589, 2021.

[8] X. Yuan and R. Haimi-Cohen, "Image Compression Based on Compressive Sensing: End-to-End Comparison With JPEG," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2889–2904, 2020.

[9] X. Yuan, D. J. Brady, and A. K. Katsaggelos, "Snapshot compressive imaging: Theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 65–88, 2021.

[10] E. J. Candes and M. B. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[11] C. Li, W. Yin, and Y. Zhang, "User's guide for TVAL3: TV minimization by augmented lagrangian and alternating direction algorithms," *CAAM report*, vol. 20, no. 46-47, p. 4, 2009.

[12] J. E. Fowler, S. Mun, and E. W. Tramel, "Multiscale block compressed sensing with smoothed projected landweber reconstruction," in *2011 19th European Signal Processing Conference*, 2011, pp. 564–568.

[13] A. S. Unde and P. Deepthi, "Block compressive sensing: Individual and joint reconstruction of correlated images," *Journal of Visual Communication and Image Representation*, vol. 44, pp. 187–197, 2017.

[14] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image compressed sensing using convolutional neural network," *IEEE Transactions on Image Processing*, vol. 29, pp. 375–388, 2020.

[15] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu, "AMP-Net: Denoising-Based Deep Unfolding for Compressive Image Sensing," *IEEE Transactions on Image Processing*, vol. 30, pp. 1487–1500, 2021.

[16] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Scalable Convolutional Neural Network for Image Compressed Sensing," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 282–12 291.

[17] R. Liu, S. Li, and C. Hou, "An End-to-End Multi-Scale Residual Reconstruction Network for Image Compressive Sensing," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 2070–2074.

[18] T. N. Canh and B. Jeon, "Multi-scale deep compressive sensing network," in *2018 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2018, pp. 1–4.

[19] T. N. Canh and B. Jeon, "Difference of convolution for deep compressive sensing," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 2105–2109.

[20] T. N. Canh and B. Jeon, "Multi-scale deep compressive imaging," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 86–97, 2021.

[21] Lu Gan, "Block Compressed Sensing of Natural Images," in *2007 15th International Conference on Digital Signal Processing*, 2007, pp. 403–406.

[22] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A Deep Learning Approach to Structured Signal Recovery," *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1336–1343, Sep. 2015.

[23] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Measurements," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 449–458.

[24] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 877–882.

[25] K. Xu, Z. Zhang, and F. Ren, "LAPRAN: A scalable Laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 485–500.

[26] Y. Yu, B. Wang, and L. Zhang, "Saliency-Based Compressive Sampling for Image Signals," *IEEE Signal Processing Letters*, vol. 17, no. 11, pp. 973–976, 2010.

[27] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[28] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[29] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 416–423.

[30] X. Fu, M. Wang, X. Cao, X. Ding, and Z.-J. Zha, "A model-driven deep unfolding method for jpeg artifacts removal," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[31] C. Mou, J. Zhang, X. Fan, H. Liu, and R. Wang, "Cola-net: Collaborative attention network for image restoration," *IEEE Transactions on Multimedia*, vol. 24, pp. 1366–1377, 2021.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[33] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From Denoising to Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 5117–5144, 2016.

[34] C. Chen, E. W. Tramel, and J. E. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 1193–1198.

[35] M. Bevilacqua, A. Roumy, C. Guillemot, and M. line Alberi Morel, "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in *Proceedings of the British Machine Vision Conference*, 2012, pp. 135.1–135.10.

[36] R. Zeyde, M. Elad, and M. Protter, "On Single Image Scale-Up Using Sparse-Representations," in *Curves and Surfaces*, 2012, pp. 711–730.

[37] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1828–1837.

[38] D. You, J. Xie, and J. Zhang, "ISTA-NET^{++}: Flexible Deep Unfolding Network for Compressive Sensing," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

[39] J. Korhonen and J. You, "Peak signal-to-noise ratio revisited: Is simple beautiful?" in *2012 Fourth International Workshop on Quality of Multimedia Experience*, 2012, pp. 37–38.

[40] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[41] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.

Kuiyuan Zhang received his M.S. degree in Computer Science and Technology from Harbin Institute of Technology, Shenzhen, China. He is currently pursuing the Ph.D degree in Computer Science and Technology from Harbin Institute of Technology, Shenzhen, China. His current research interests include image compressive sensing and image encryption.

Zhongyun Hua (S'14-M'16) received the B.S. degree from Chongqing University, Chongqing, China, in 2011, and the M.S. and Ph.D. degrees from University of Macau, Macau, China, in 2013 and 2016, respectively, all in software engineering.

He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include chaotic system, image processing and multimedia security. He has published more than fifty papers on the subject, receiving more than 3500 citations. He currently serves as an Associate Editor for *International Journal of Bifurcation and Chaos*.

Yuanman Li (Member, IEEE) received the B.Eng. degree in software engineering from Chongqing University, Chongqing, China, in 2012, and the Ph.D. degree in computer science from the University of Macau, Macau, in 2018.

From 2018 to 2019, he was a Post-Doctoral Fellow with the State Key Laboratory of Internet of Things for Smart City, University of Macau. He is currently an Assistant Professor with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include data representation, multimedia security and forensics, computer vision, and machine learning.

Yongyong Chen received the B.S. and M.S. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Macau, Macau, in 2020. He is currently an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He has published more than 30 research papers in top-tier journals and conferences, including IEEE TIP, IEEE TNNLS, IEEE TMM, IEEE TCSVT, IEEE TGRS, IEEE TCI, IEEE JSTSP, Pattern Recognition and ACM MM. His research interests include image processing, data mining, and computer vision.

Yicong Zhou (M'07-SM'14) received the B.S. degree in electrical engineering from Hunan University in Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. Dr. Zhou is a Professor in the Department of Computer and Information Science at University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a Fellow of SPIE (the Society of Photo-Optical Instrumentation Engineers) and was recognized as one of "World's Top 2% Scientists" and one of "Highly Cited Researchers" in 2020 and 2021. He received the Third Price of Macao Natural Science Award as a sole winner in 2020 and a co-recipient in 2014. Dr. Zhou has been a leading Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society since 2015. He serves as an Associate Editor for *IEEE Transactions on Neutral Networks and Learning Systems*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Geoscience and Remote Sensing*, and four other journals.