Image Compressed Sensing Using Non-local Neural Network

Wenxue Cui, *Student Member, IEEE,* Shaohui Liu, *Member, IEEE,* Feng Jiang, *Member, IEEE,* and Debin Zhao, *Member, IEEE*

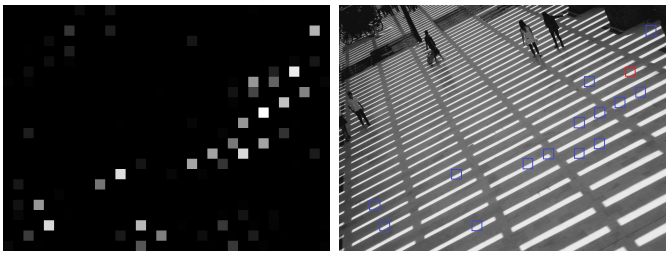
***Abstract*—Deep network-based image Compressed Sensing (CS) has attracted much attention in recent years. However, the existing deep network-based CS schemes either reconstruct the target image in a block-by-block manner that leads to serious block artifacts or train the deep network as a black box that brings about limited insights of image prior knowledge. In this paper, a novel image CS framework using non-local neural network (NL-CSNet) is proposed, which utilizes the non-local self-similarity priors with deep network to improve the reconstruction quality. In the proposed NL-CSNet, two non-local subnetworks are constructed for utilizing the non-local self-similarity priors in the measurement domain and the multi-scale feature domain respectively. Specifically, in the subnetwork of measurement domain, the long-distance dependencies between the measurements of different image blocks are established for better initial reconstruction. Analogically, in the subnetwork of multi-scale feature domain, the affinities between the dense feature representations are explored in the multi-scale space for deep reconstruction. Furthermore, a novel loss function is developed to enhance the coupling between the non-local representations, which also enables an end-to-end training of NL-CSNet. Extensive experiments manifest that NL-CSNet outperforms existing state-of-the-art CS methods, while maintaining fast computational speed.**

***Index Terms*—Image compressed sensing, non-local self-similarity prior, non-local neural network, convolutional neural networks (CNNs).**

I. Introduction

RECENT years have seen significant interest in the compressed sensing (CS) [1], [2], which provides a new paradigm for signal acquisition that performs signal sampling and compression jointly. The CS theory implies that if a signal $x \in \mathbb{R}^N$ is sparse in a certain domain Ψ, it can be reconstructed with high probability from a small number of its linear measurements $y = \Phi x$, where $\Phi \in \mathbb{R}^{M \times N}$ is the sampling matrix with $M \ll N$ and $\frac{M}{N}$ is usually referred to as the sampling rate less than that determined by the Nyquist sampling theorem. The possible reduction of sampling rate is attractive for diverse practical applications, including but not limited to Magnetic Resonance Imaging (MRI) [3], radar imaging [4] and sensor networks [5].

In the study of CS, two main challenges are usually concerned: 1) the design of sampling matrix Φ for efficient signal acquisition and 2) the development of reconstruction solvers for recovering the original signal from its measurements [6]. In recent years, a great deal of algorithms have been proposed to deal with these two challenges. Specifically, for the sampling matrix, the Block-based CS (BCS) [7] is first proposed to squeeze the storage requirement of sampling matrix, in which the measurements are produced in a block-by-block sampling manner. Based on BCS, a variety of sampling matrices have been developed, such as the Gaussian Random Matrix (GRM) [8], [9] and Local Structural-based Measurement Matrix (LSMM) [10]. However, both GRM and LSMM are signal independent, which may result in unsatisfactory reconstructed quality. For the CS reconstruction, many sparsity-regularized-based schemes have been presented, such as the greedy algorithms [11], [12] and the convex-optimization schemes [13], [14]. However, these CS reconstruction methods usually explore image priors to build model and then solve an optimization problem in an iterative fashion, which usually suffer from high computational complexity.



(a) (b)

Fig. 1. The visualization of the learned affinity matrix in the non-local subnetwork of measurement domain (a) and its corresponding image patches on the original image (b). The red block is the current image patch and the blue blocks are the corresponding image patches mapped from the highlighted elements of the affinity matrix, which obviously have the similar structural textures.

Recently, fueled by the powerful learning ability of deep neural networks, a series of deep network-based image CS methods have been proposed. For image sampling, the deep network-based sampling matrix can be optimized jointly with the reconstruction module during the training process [6]. For image reconstruction, the deep network-based schemes usually build a deep mapping from the measurement domain to the image domain, which can be roughly divided into two groups: **1) Training the deep network as a black box:** this kind of method trains the reconstruction network as a black box. Specifically, the early works [15]–[18] usually reconstruct the target image block-by-block and then splice the reconstructed image blocks together into a final image. However, the reconstruction block-by-block usually

This work was supported by the National Natural Science Foundation of China under Grant 61872116. (*Corresponding author: Debin Zhao.*)

Wenxue Cui, Shaohui Liu, Feng Jiang and Debin Zhao are with the Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China and also with the Peng Cheng Laboratory, Shenzhen, 518055, China (e-mail: wenxuecui@stu.hit.edu.cn; shliu@hit.edu.cn; fjiang@hit.edu.cn; dbzhao@hit.edu.cn).

suffers from serious block artifacts especially at low sampling rates [6]. To relieve the block artifacts, some methods [6], [19] attempt to concatenate all image blocks together in the initial reconstruction, and then complete a deep reconstruction in the global image space. Compared with the reconstruction block-by-block, these CS algorithms weaken the block artifacts and achieve higher reconstruction quality. However, these reconstruction networks generally establish a direct mapping from the measurement domain to the image domain in a rude manner and therefore result in limited insights of the image prior knowledge. **2) Interpretable CS reconstruction network:** this kind of method usually integrates the deep network with the iterative optimizers to enjoy a good interpretability. Motivated by the powerful learning capability of deep neural networks, some literatures [20]–[23] attempt to unfold the iterative optimization algorithms (e.g., iterative shrinkage-thresholding algorithm (ISTA) [24] and approximate message passing (AMP) [25]) onto networks to solve the CS reconstruction problem and achieve impressive performance. Obviously, by unfolding the optimization-based solvers, these deep unfolded methods have better interpretability, but these algorithms usually adopt a plain neural network architecture and therefore cannot fully exert the expressiveness of the proposed model for image reconstruction.

In this paper, a novel image CS framework using non-local neural network (dubbed NL-CSNet) is proposed, in which two non-local subnetworks are constructed for utilizing the non-local self-similarity priors in the measurement domain and the multi-scale feature domain respectively. Specifically, in the non-local subnetwork of measurement domain, we establish a long-distance reference between the measurements of different image blocks (as shown in Fig. 1), which efficiently explores the interblock correlations for better initial reconstruction. Analogically, in the non-local subnetwork of multi-scale feature domain, a long-range reference between the non-local structural textures is built, which explores the affinities between the dense feature representations in the multi-scale space for deep reconstruction. In addition, a novel loss function is proposed to enhance the coupling between the non-local representations, which also enables an end-to-end training of the proposed CS framework. Extensive experiments show that NL-CSNet outperforms existing state-of-the-art CS methods, while maintaining fast computational speed.

The main contributions are summarized as follows:

1) A novel image compressed sensing framework using non-local neural network is proposed, which utilizes the non-local self-similarity priors with deep network to improve the reconstruction quality.

2) In the measurement domain, the latent correlations between the measurements of different image blocks are explored for better initial reconstruction, which is favored for the following deep reconstruction.

3) In the multi-scale feature domain, the affinities between the dense feature representations are explored in the multi-scale space for deep reconstruction.

4) A novel loss function is designed to enhance the coupling between the non-local representations, which also provides an end-to-end training of the proposed CS framework.

The remainder of this paper is organized as follows: Section II reviews the recent related works. Section III elaborates the proposed CS framework, including the non-local neural network and the architectures of the networks embedding the non-local neural network in the measurement domain and the multi-scale feature domain respectively. Section IV provides the experimental results and Section V concludes the paper.

II. Background and Related Works

A. Image CS Reconstruction

In CS task, the target image $x \in \mathbb{R}^N$ is reconstructed from its linear measurements $y \in \mathbb{R}^M$. Because $M \ll N$, this inverse problem is typically ill-posed. Recently, in order to solve this inverse problem, a great deal of CS methods have been proposed, which can be roughly grouped into two categories: optimization-based CS reconstruction methods and deep network-based CS reconstruction methods.

Optimization-based CS Methods: Given the linear measurements y, the traditional image CS reconstruction methods usually reconstruct the original image x by solving an inverse optimization problem:

$$\tilde{x} = \arg\min_x \frac{1}{2}\|\Phi x - y\|_2^2 + \lambda \|\Psi x\|_\delta \tag{1}$$

where Ψx indicates the sparse coefficients with respect to the transform Ψ and the sparsity is characterized by the δ norm. λ is the regularization parameter to control the sparsity term. To solve Eq. (1), many sparsity-regularized based methods, such as the greedy algorithms [11], [12] and the convex-optimization algorithms [13], [14], have been proposed. To further enhance the reconstructed quality, more sophisticated structures are established, including minimal total variation [26], [27], wavelet tree sparsity [28], [29], non-local image prior [30], [31] and simple representations in adaptive bases [32]. Many of these approaches have led to significant improvements. However, these optimization-based CS reconstruction algorithms usually suffer from high computational complexity because of their hundreds of iterations, thus limiting the practical applications of CS greatly.

Deep Network-based CS Methods: Driven by the powerful learning capability of deep neural networks, many deep network-based methods have been developed for image CS reconstruction, which can be roughly divided into two groups: 1) training the reconstruction network as a black box and 2) the interpretable CS reconstruction network.

For the first group of algorithms (training the reconstruction network as a black box), Mousavi *et al.* first propose a stacked denoising autoencoder (SDA) [33] to capture statistical dependencies between the elements of the signal. However, the fully connected network (FCN) utilized in SDA leads to a huge number of learnable parameters. To relieve this problem, several Convolutional Neural Networks (CNNs) based reconstruction methods [15], [16], [18], [34] are proposed, which usually build a direct mapping from the blocked measurements to the corresponding image blocks. However, these deep network-based CS algorithms usually bring about serious block artifacts [6], [35] (especially at low sampling rates) because of their block-by-block reconstruction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3132489, IEEE Transactions on Multimedia

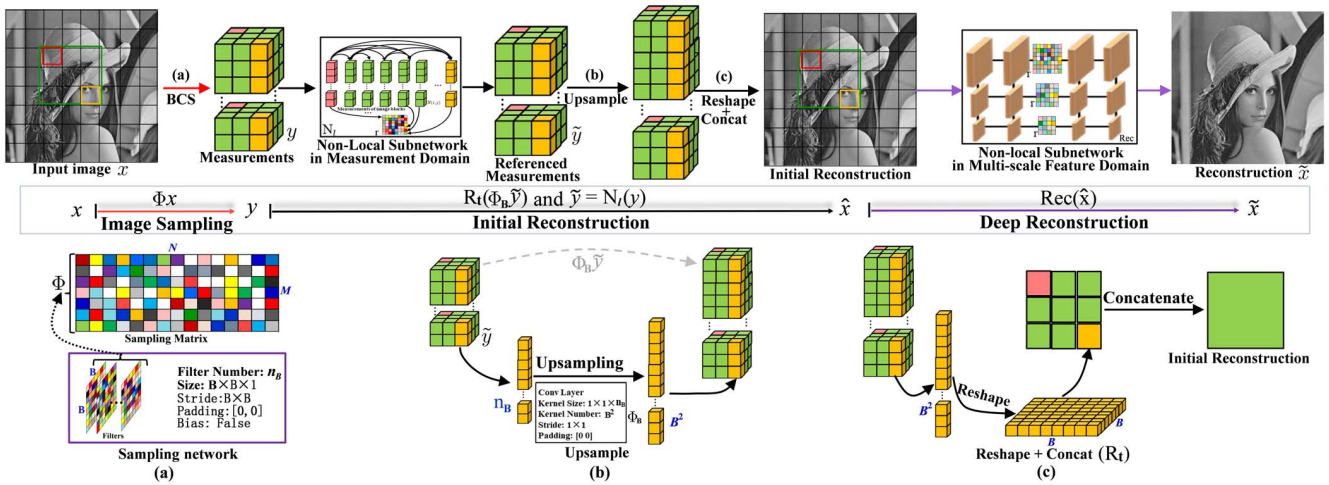


Fig. 2. Diagram of our proposed image CS framework using non-local neural network. For CS reconstruction, two phases are included: initial reconstruction and deep reconstruction. The three sub-figures (a)-(c) show the details of three functional operators. The blue text represents the dimensional description.

In order to solve this problem, some works [16]–[18] try to append a de-blocking algorithm (e.g., BM3D [36]) after these block-by-block reconstruction methods, which usually introduce additional computational burden. To remove the block artifacts further, several literatures [6], [19], [35], [37] attempt to explore the deep image priors in the whole image space. Specifically, these CS methods still adopt the block-by-block sampling, however during the reconstruction, they first concatenate all image blocks together in the initial reconstruction, and then complete a deep reconstruction in the whole image space. Recently, to enhance the applicability of CS framework, several scalable network architectures [19], [38] are proposed, which achieve scalable sampling and reconstruction with only one model. More recently, a novel multi-channel deep network [39] for block-based image CS is proposed, in which the image blocks with different textural complexities are adaptively allocated different sampling rates. Compared to the block-by-block reconstruction methods, these CS algorithms usually achieve much higher reconstruction performance. However, the networks used in these methods are usually trained as a black box, which usually result in limited insights of the image prior knowledge.

For the second group of algorithms (the interpretable CS reconstruction network), the deep networks are usually integrated with the iterative optimizers to enjoy a better interpretability. For example, inspired by the denoising-based iterative thresholding (DIT) algorithm and the AMP algorithm, Metzler *et al.* [20] first propose LDIT and LDAMP respectively for CS reconstruction. Subsequently, Zhang *et al.* [21] propose a deep unfolded version of the popular algorithm ISTA [24], dubbed ISTA-Net, which employs CNNs to learn the appropriate transformation operations and soft thresholding functions to reflect the sparsity of data. However, the performance of [21] is limited because of its random sampling manner and block-by-block reconstruction strategy. To enhance the reconstruction performance, the authors develop an enhanced version of [21], dubbed OPINE-Net [23], in which a learnable sampling matrix and an efficient deblocking strategy are adopted for boosting the reconstruction quality. More recently, by unfolding the iterative denoising process of the well-known AMP algorithm, Zhang *et al.* [22] propose a deep unfolded model, dubbed AMP-Net, to solve the visual image CS problem. By unfolding the optimization-based iterative solvers onto networks, these deep unfolded methods have a better interpretability, but these algorithms usually adopt a plain network architecture and therefore cannot fully exert the expressiveness of the proposed model for image reconstruction.

Recently, motivated by the superiority of the CNN-based denoisers [40], the plug-and-play algorithms [41]–[44] regularized by the deep denoising priors attract much attention for applying to diverse low-level vision tasks. The main idea is that, with the aid of variable splitting algorithms, such as half-quadratic splitting (HQS), it is possible to deal with the fidelity item and prior item separately [42]. Theoretically, the prior item only corresponds to a denoising subproblem [44], which can be solved via a deep CNN denoiser. Based on the above statement, Romano *et al.* [41] propose the paradigm of the Regularization by Denoising (RED): using the denoising engine (such as TNRD [45]) in defining the regularization of the inverse problem. In [42], different CNN denoisers are trained to plug into Half Quadratic Splitting (HQS) algorithm for various low-level vision applications. In [43], Liu *et al.* propose to broaden the current denoiser-centric view of RED by considering priors corresponding to networks trained for

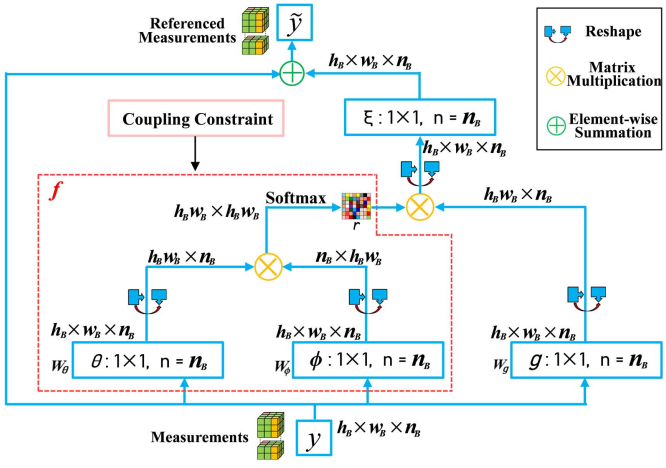


Fig. 3. The details of non-local subnetwork in measurement domain. $n = n_B$ is the number of the convolutional filters of size 1×1. The annotations indicate the dimensional information.

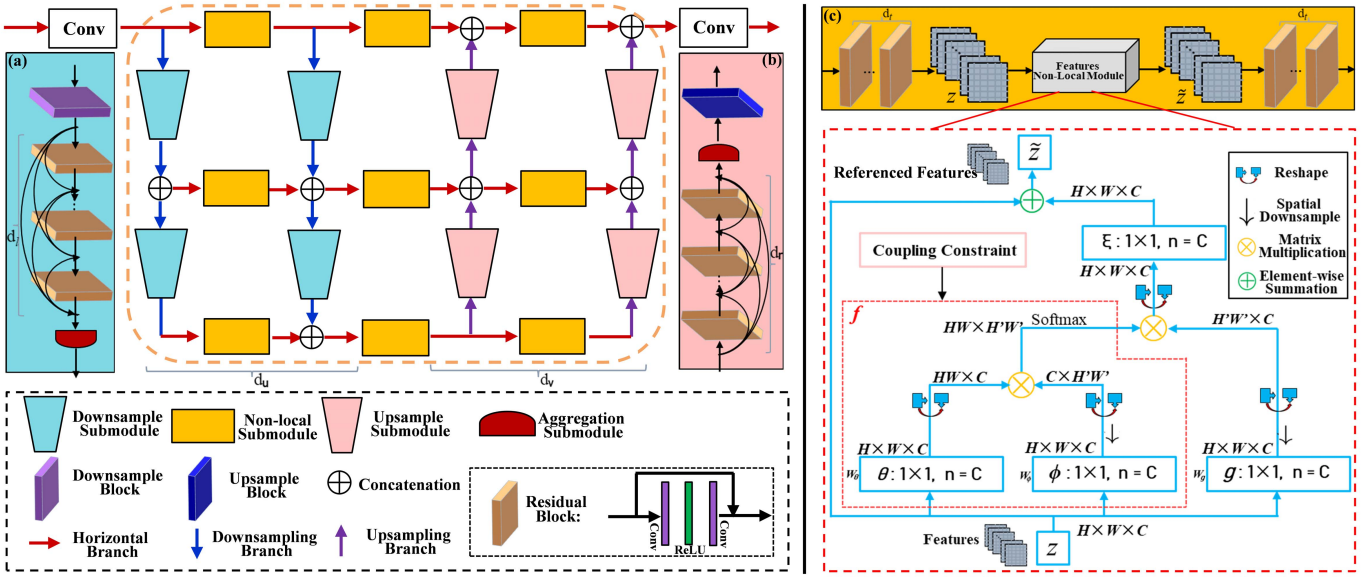


Fig. 4. The architecture details of the proposed non-local subnetwork in multi-scale feature domain (MS-NLNet) in the deep reconstruction phase. (a)-(c) are the architecture details of three submodules: Downsample submodule, Upsample submodule and Non-local submodule.

more general artifact-removal. In [44], Zhang *et al.* expand their previous work [42] and propose a plug-and-play unrolled model using deep denoiser prior for image restoration. By integrating the deep neural network with the iterative optimizers, the deep unrolled models mentioned above can leverage the powerful expressiveness of CNNs. However, these methods usually adopt a plain convolutional neural network and therefore cannot fully characterize the rich image priors for image reconstruction.

B. Non-local Self-Similarity Image Prior

Recent studies show that the image prior models play an important role in diverse image processing tasks. The non-local self-similarity, as a well-known image prior, has been extensively studied, which depicts the repetitiveness of higher-level patterns (e.g., textures and structures) globally positioned in images. Inspired by the success of nonlocal means (NLM) denoising filter [46], a series of nonlocal regularization terms for inverse problems exploiting nonlocal self-similarity property of natural images are emerging [47]–[50]. Due to the utilization of the non-local self-similarity priors, the methods with nonlocal regularization terms usually produce superior results, with sharper image edges and richer image details. Recently, the non-local self-similarity image prior is also applied in many optimization-based CS methods [30], [31], [51], [52]. For example, Zhang *et al.* [31] establish a novel sparse representation model of natural images by mining the nonlocal patches with the similar structures. Chen *et al.* [52] exploit the non-local self-similarity patches and propose a low-rank based CS model for image reconstruction. By introducing the non-local self-similarity image prior, these non-local prior based methods establish the references between the non-local patches with the similar textures and obtain better CS reconstruction performance.

By exploring the non-local self-similarity prior, several deep network-based CS schemes begin to utilize the non-local priors. For example, Li *et al.* [34] propose a residual network with nonlocal constraint for image CS reconstruction, which considers the non-local self-similarity image prior and adds a non-local operation into the proposed network. While this CS network reconstructs the target image in a block-by-block manner from the measurements acquired by using the Gaussian random sampling matrix. Besides, the non-local operator in [34] only perceives the self-similar information inside the current image block, which ignores the correlations among different image blocks. Compared to [34], the differences of the proposed framework are as follows: 1) Instead of block-by-block reconstruction manner, the proposed CS network reconstructs the target image from the measurements in the whole image space. 2) The sampling matrix of the proposed CS model is optimized jointly with the reconstruction network. 3) The proposed non-local module globally explores the self-similar knowledge in the entire image space, which significantly expands the receptive field. Recently, inspired by Deep Image Prior (DIP) [56], Sun *et al.* [54] propose a non-locally regularized CS network, in which the non-local prior and the deep network prior are both considered for image reconstruction. However, for each image, the network in [54] needs to be trained online driven by the two priors in an iterative fashion, which undoubtedly brings about a high computational cost and a lack of flexibility. By contrast, the proposed CS network generates final test model through pre-training. Specifically, in the training process, the proposed non-local module automatically characterizes the non-local image prior. In the testing process, the pre-trained model can directly capture the long-range dependencies in a feedforward fashion, while maintaining fast computational speed.

Transformer has demonstrated exemplary performance on a broad range of Natural Language Processing (NLP) related tasks [57], [58]. Recently, the breakthroughs from Transformer networks in NLP domain has sparked great interest in the computer vision community [59]. However, these Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality [59], which makes their

TABLE I
AVERAGE PSNR AND SSIM COMPARISONS OF DIFFERENT REPRESENTATIVE CS ALGORITHMS BASED ON THE RANDOM SAMPLING MATRIX AT VARIOUS SAMPLING RATES ON DATASET SET5. BOLD INDICATES THE BEST RESULT, AND UNDERLINE SIGNIFIES THE SECOND BEST RESULT.

Algorithms	Rate=0.01		Rate=0.04		Rate=0.10		Rate=0.20		Rate=0.30		Avg.	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
TV [53]	15.53	0.4554	22.14	0.6076	27.07	0.7865	30.45	0.8709	32.75	0.9107	25.59	0.7262
MH [32]	18.08	0.4472	23.65	0.6337	28.57	0.8211	32.08	0.8881	34.06	0.9158	27.29	0.7412
GSR [31]	18.87	0.4909	24.80	0.7286	29.99	0.8654	**34.17**	**0.9257**	**36.83**	0.9492	28.93	0.7920
ReconNet$_{(CVPR2016)}$ [16]	18.46	0.4492	23.54	0.6189	26.89	0.7518	29.55	0.8348	31.20	0.8738	25.93	0.7057
I-Recon$_{(TCI2018)}$ [17]	21.49	0.5571	**27.26**	0.7607	30.28	0.8496	33.12	0.9023	35.07	0.9357	29.44	0.8011
ISTA-Net$_{(CVPR2018)}$ [21]	18.48	0.4222	23.02	0.6428	28.53	0.8276	– –	– –	34.87	0.9354	– –	– –
ISTA-Net$^{+}_{(CVPR2018)}$ [21]	18.55	0.4408	23.45	0.6619	28.61	0.8315	– –	– –	35.45	0.9408	– –	– –
NLR-CSNet$_{(TMM2020)}$ [54]	21.00	– –	25.07	– –	29.23	– –	– –	– –	– –	– –	– –	– –
DPA-Net$_{(TIP2020)}$ [37]	– –	– –	26.63	0.7767	30.32	0.8713	– –	– –	36.17	**0.9495**	– –	– –
DPIR$_{(TPAMI2021)}$ [44]	17.68	0.4364	26.01	0.7565	29.97	0.8667	33.36	0.9206	35.62	0.9475	28.53	0.7855
NL-CSNet	**22.37**	**0.6031**	26.89	**0.7786**	**30.60**	**0.8753**	33.41	0.9202	35.74	0.9483	**29.80**	**0.8251**

TABLE II
AVERAGE PSNR AND SSIM COMPARISONS OF DIFFERENT REPRESENTATIVE CS ALGORITHMS BASED ON THE LEARNED SAMPLING MATRIX AT VARIOUS SAMPLING RATES ON DATASET SET5. BOLD INDICATES THE BEST RESULT, AND UNDERLINE SIGNIFIES THE SECOND BEST RESULT.

Algorithms	Rate=0.01		Rate=0.04		Rate=0.10		Rate=0.20		Rate=0.30		Avg.	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CSNet$_{(ICME2017)}$ [55]	24.02	0.6378	28.57	0.8226	32.30	0.9015	35.63	0.9451	37.90	0.9630	31.68	0.8540
LapCSNet$_{(ICASSP2018)}$ [35]	24.42	0.6686	– –	– –	32.44	0.9047	– –	– –	– –	– –	– –	– –
SCSNet$_{(CVPR2019)}$ [19]	24.21	0.6468	28.79	0.8314	32.77	0.9083	36.15	0.9487	38.45	0.9655	32.07	0.8601
CSNet$^{+}_{(TIP2020)}$ [6]	24.18	0.6478	28.64	0.8265	32.59	0.9062	36.05	0.9481	38.25	0.9644	31.94	0.8586
BCS-Net$_{(TMM2020)}$ [39]	22.98	0.6103	– –	– –	32.71	0.9030	36.12	0.9483	38.64	0.9694	– –	– –
OPINENet$^{+}_{(JSTSP2020)}$ [23]	22.76	0.6194	29.03	0.8440	33.71	0.9259	– –	– –	– –	– –	– –	– –
AMP-Net$_{(TIP2021)}$ [22]	23.00	0.6488	28.85	0.8375	33.35	0.9162	36.64	0.9532	**39.07**	0.9671	32.18	0.8646
NL-CSNet*	**24.82**	**0.6771**	**29.61**	**0.8571**	**33.84**	**0.9312**	**36.91**	**0.9589**	38.86	**0.9703**	**32.81**	**0.8789**

accuracy unsatisfactory on insufficient data sets. In addition, the inputs of Transformers are usually the linearly embedded patches, which accelerates the exploring of correlations between different image blocks, but weakens the learning ability of local representations to a certain extent. By contrast, the proposed network not only establishes the long-range references between different blocks in measurement domain, but also build the dependencies between dense representations in the multi-scale feature domain.

Graph convolutional network (GCN) is an effective strategy to establish long-range dependence for graph data. Recently, few works [60]–[62] apply GCN to image restoration tasks, and achieve impressive performance. However, the perceptive field in these GCN-based methods is usually limited within a predefined search window by considering the computational complexity [61]. Besides, a fixed number of similar references are explored in each window, which limits the flexibility and adaptability of the method. By contrast, the proposed non-local module perceives information in the whole image space. Furthermore, the proposed method separately explores coarse-grained references and find-grained dependencies in the measurement domain and the multi-scale feature domain, while the GCN-based methods usually capture non-local self-similar knowledge only at the dense feature space.

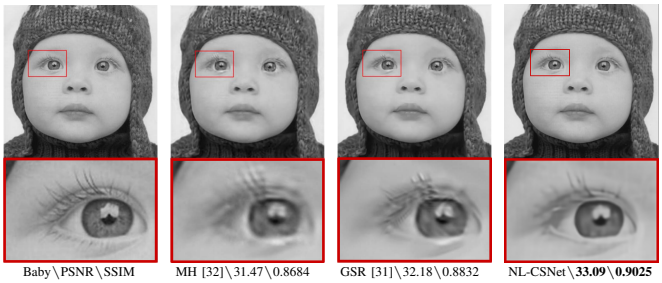


Fig. 5. Visual comparisons of the proposed NL-CSNet and the other optimization-based CS methods on image *Baby* from Set5 [19] in case of sampling rate = 0.1.

III. IMAGE COMPRESSED SENSING USING NON-LOCAL NEURAL NETWORK

In this section, we first give an overview of the proposed whole CS framework, and then detail the structures of the non-local neural network. Finally, we introduce the architectures of the designed networks embedding the non-local neural network in the measurement domain and the multi-scale feature domain respectively.

A. Overview of NL-CSNet

Figure 2 shows the whole network architecture of the proposed NL-CSNet. Specifically, for image sampling, a convolutional layer with specific parameter configurations (shown in Figure 2(a)) is utilized to imitate the block-based sampling process [6], after which a series of measurements are generated. In order to reconstruct the target image from the measurements, a novel image CS reconstruction model using non-local neural network is proposed, in which two non-local subnetworks are constructed for exploring the non-local self-similarity priors in the measurement domain and the multi-scale feature domain respectively. More specifically, given the sampled measurements of all image blocks, the non-local subnetwork of measurement domain first establishes a

TABLE III
AVERAGE PSNR AND SSIM COMPARISONS OF DIFFERENT DEEP NETWORK-BASED CS ALGORITHMS USING RANDOM SAMPLING MATRIX AT DIVERSE SAMPLING RATES ON DATASET SET11. BOLD INDICATES THE BEST RESULT, AND UNDERLINE SIGNIFIES THE SECOND BEST RESULT.

Algorithms	Rate=0.01		Rate=0.04		Rate=0.10		Rate=0.20		Rate=0.30		Avg.	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SDA$_{(ALLERTON2015)}$ [33]	17.69	0.4376	21.05	0.5720	23.66	0.6794	– –	– –	24.77	0.7191	– –	– –
ReconNet$_{(CVPR2016)}$ [16]	17.54	0.4426	21.24	0.5748	24.07	0.6958	25.54	0.7719	28.72	0.8517	23.42	0.6674
DR^2-Net$_{(CVPR2017)}$ [15]	17.44	0.4294	20.80	0.5806	24.71	0.7175	– –	– –	30.52	– –	– –	– –
IRCNN$_{(CVPR2017)}$ [42]	7.70	– –	17.56	– –	24.02	– –	– –	– –	31.18	– –	– –	– –
LDIT$_{(NIPS2017)}$ [20]	17.58	0.4449	21.45	0.6075	25.56	0.7691	– –	– –	32.69	0.9223	– –	– –
LDAMP$_{(NIPS2017)}$ [20]	17.51	0.4409	21.30	0.5985	24.94	0.7483	– –	– –	32.01	0.9144	– –	– –
I-Recon$_{(TCI2018)}$ [17]	19.20	0.5018	**24.06**	0.7089	25.97	0.7888	27.92	0.8457	31.45	0.9135	25.72	0.7517
DPDNN$_{(TPAMI2018)}$ [63]	17.59	0.4459	21.11	0.6029	24.53	0.7392	– –	– –	32.06	0.9145	– –	– –
ISTA-Net$^+$ $_{(CVPR2018)}$ [21]	17.45	0.4131	21.56	0.6240	26.49	0.8036	– –	– –	**33.70**	0.9382	– –	– –
NN$_{(TCI2020)}$ [64]	17.67	0.4324	20.65	0.5525	22.99	0.6591	– –	– –	27.64	0.8095	– –	– –
DPA-Net$_{(TIP2020)}$ [37]	18.05	0.5011	23.50	**0.7205**	26.99	0.8354	– –	– –	– –	– –	– –	– –
NL-CSNet	**19.59**	**0.5229**	23.74	0.7097	**27.24**	**0.8386**	**30.29**	**0.8910**	33.41	**0.9386**	**26.85**	**0.7802**

long-distance reference between the measurements of different image blocks, which efficiently explores the interblock correlations in the measurement domain for better initial reconstruction. In fact, this kind of reference between the measurements of different image blocks is only a coarse non-local reference because of the non-overlapping block sampling of BCS. In order to further enhance the reconstructed quality from the initial reconstruction, the subnetwork of multi-scale feature domain is responsible for exploring the non-local self-similarity knowledge between the dense feature representations in the multi-scale space, which is capable of building a fine granular reference between the non-local structural textures for deep reconstruction. Furthermore, it is worth noting that a novel loss function is proposed to enhance the coupling of non-local information, which also enables an end-to-end training of the proposed CS framework.

B. Non-Local Neural Network

In the past few years, the non-local neural networks have been proposed in [65] to establish the long-distance references between the non-local representations with the similar structures. For example, given the current signal representation x_i and according to the non-local mean operation [46], the referenced information of x_i by referencing the other signal representations can be expressed as:

$$\hat{x}_i = \frac{1}{\mathcal{C}(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{2}$$

where f is a pairwise function to compute the affinities between the given representations x_i and x_j. The unary function g is used to compute a new representation of x_j. In fact, the non-local behavior in Eq. (2) signifies that the all signal representations ($\forall$) are considered in the operation and the final response is normalized by a factor $\mathcal{C}(x)$.

Obviously, $f(x_i, x_j)$ and $f(x_j, x_i)$ are a pair of symmetrical affinities and they are both actually closely related to the similarity between the given two representations x_i and x_j. In most non-local prior based iterative CS literatures [31], Euclidean distance is usually selected as the similarity criterion between different image patches, which actually is a directionless scalar metric. In other words, given two signal representations x_i and x_j with the similar structures, there actually is a correspondence between $f(x_i, x_j)$ and $f(x_j, x_i)$. For example, if x_i is similar to x_j, we can obtain that x_j is also similar to x_i. Therefore, there is a coupling relationship between $f(x_i, x_j)$ and $f(x_j, x_i)$. Unfortunately, in existing non-local neural networks [65] or their counterpart variants [66], [67], the affinity is generally computed directly in the embedding space (mapped through two linear matrices θ and ϕ), such as the embedded gaussian or the embedded dot product, which does not consider the coupling between $f(x_i, x_j)$ and $f(x_j, x_i)$. In our non-local module, we propose a new constraint as follows:

$$\hat{f}(x_i, x_j) = \arg\min_{f} \| f(x_i, x_j) - f(x_j, x_i) \|_l \tag{3}$$

where the coupling (i.e., the distance) between $f(x_i, x_j)$ and $f(x_j, x_i)$ is characterized by the l norm. Through the above constraint, the consistency between $f(x_i, x_j)$ and $f(x_j, x_i)$ is maintained to a certain extent, which facilitates a mutual reference between the non-local information. It is worth noting that the constraint in Eq. (3) can be embedded into the loss function of the proposed NL-CSNet, which will be elaborated in the Subsection IV-A. The following two subsections introduce the architectures of networks embedding the non-local neural network in the measurement domain and the multi-scale feature domain respectively.

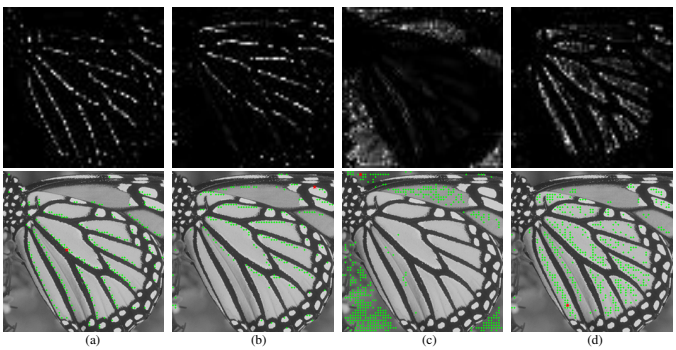


Fig. 6. The visualization of the learned affinity matrix in the non-local subnetwork of multi-scale feature domain (top) and its corresponding positions on the original image (bottom). The red points are the current locations and the green points are the corresponding positions mapped from the highlighted elements of affinity matrix, around which the similar texture is maintained. (a), (b) are edge areas, and (c), (d) are smooth areas.

TABLE IV
AVERAGE PSNR AND SSIM COMPARISONS OF DIFFERENT DEEP NETWORK-BASED CS ALGORITHMS USING LEARNED SAMPLING MATRIX AT DIVERSE SAMPLING RATES ON DATASET SET11. BOLD INDICATES THE BEST RESULT, AND UNDERLINE SIGNIFIES THE SECOND BEST RESULT.

Algorithms	Rate=0.01		Rate=0.04		Rate=0.10		Rate=0.20		Rate=0.30		Avg.	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CSNet$_{(ICME2017)}$ [55]	21.01	0.5560	25.23	0.7538	28.10	0.8514	31.36	0.9141	33.86	0.9448	27.91	0.8040
SCSNet$_{(CVPR2019)}$ [19]	21.04	0.5562	25.48	0.7626	28.48	0.8616	31.82	0.9215	34.62	0.9511	28.29	0.8106
CSNet$^+$$_{(TIP2020)}$ [6]	21.03	0.5566	25.41	0.7602	28.37	0.8580	31.66	0.9203	34.30	0.9490	28.15	0.8088
BCS-Net$_{(TMM2020)}$ [39]	20.88	0.5505	25.44	0.7425	29.43	0.8676	33.06	0.9283	35.06	0.9554	28.77	0.8089
OPINENet$^+$$_{(JSTSP2020)}$ [23]	20.02	0.5362	25.52	0.7879	29.81	0.8904	– –	– –	– –	– –	– –	– –
AMP-Net$^+$$_{(TIP2021)}$ [22]	20.20	0.5581	25.26	0.7722	29.40	0.8779	33.26	0.9405	**36.03**	0.9586	28.83	0.8215
NL-CSNet*	**21.96**	**0.6005**	**26.26**	**0.8108**	**30.05**	**0.8995**	**33.52**	**0.9440**	35.68	**0.9606**	**29.49**	**0.8431**

TABLE V
AVERAGE PSNR AND SSIM COMPARISONS OF RECENT DEEP NETWORK-BASED CS ALGORITHMS USING LEARNED SAMPLING MATRIX AT VARIOUS SAMPLING RATES ON DATASET SET14. BOLD INDICATES THE BEST RESULT, AND UNDERLINE SIGNIFIES THE SECOND BEST RESULT.

Data	Rate	CSNet [55]		SCSNet [19]		CSNet$^+$ [6]		OPINE-Net$^+$ [23]		AMP-Net [22]		NL-CSNet*	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set14	0.01	22.79	0.5628	22.87	0.5631	22.83	0.5630	22.30	0.5508	22.60	0.5723	**23.61**	**0.5862**
	0.04	26.05	0.7164	26.24	0.7210	26.11	0.7196	26.67	0.7306	26.60	0.7212	**27.11**	**0.7460**
	0.10	28.91	0.8119	29.22	0.8181	29.13	0.8169	29.94	0.8415	29.87	0.8130	**30.16**	**0.8527**
	0.20	31.86	0.8908	32.19	0.8945	32.15	0.8941	– –	– –	32.72	0.9024	**32.96**	**0.9150**
	0.30	34.00	0.9276	34.51	0.9311	34.34	0.9297	– –	– –	**35.23**	0.9364	34.88	**0.9405**
Avg.	—	28.72	0.7819	29.01	0.7856	28.91	0.7847	– –	– –	29.40	0.7891	**29.74**	**0.8081**

C. Non-Local Subnetwork in Measurement Domain

In BCS, the image x with size $w \times h$ is first divided into non-overlapping blocks $x_{(i,j)}$ of size $B \times B$, where $i \in \{1, 2, ..., w_B\}$ ($w_B = \frac{w}{B}$) and $j \in \{1, 2, ..., h_B\}$ ($h_B = \frac{h}{B}$) are the position indexes of the current image block. Then, a sampling matrix Φ of size $n_B \times B^2$ is usually used to acquire the CS measurements, i.e., $y_{(i,j)} = \Phi x_{(i,j)}$, where $n_B = \lfloor \frac{M}{N} B^2 \rfloor$ and $\frac{M}{N}$ is the given sampling rate. In fact, each row of the sampling matrix Φ can be considered as a filter. Therefore, we can use a series of convolutional operations to mimic the sampling process [6]. Specifically, in our CS framework, a convolutional layer with the specific configurations as shown in Fig. 2(a) is utilized to imitate the sampling process. Besides, it is worth noting that the measurements between different image blocks are actually independent because of the block-by-block sampling manner in BCS. In order to establish the correlations between the measurements of different image blocks, we embed the non-local neural network into the measurement domain, i.e.,

TABLE VI
AVERAGE RUNNING TIME (IN SECONDS) OF DIFFERENT CS ALGORITHMS FOR RECONSTRUCTING A 256 × 256 IMAGE.

Algorithm	Rate=0.01		Rate=0.1	
	CPU	GPU	CPU	GPU
TV [53]	2.4006	—	2.7405	—
MH [32]	23.1006	—	19.0405	—
GSR [31]	235.6297	—	230.4755	—
SDA [33]	—	0.0045	—	0.0029
ReconNet [16]	0.5193	0.0244	0.5258	0.0289
ISTA-Net [21]	0.9230	0.0390	0.9240	0.0395
ISTA-Net$^+$ [21]	1.3750	0.0470	1.3820	0.0485
CSNet [55]	0.2950	0.0157	0.3014	0.0168
SCSNet [19]	0.5103	0.1050	0.5146	0.1332
CSNet$^+$ [6]	0.8960	0.0262	0.9024	0.0287
NL-CSNet	0.8246	0.0899	0.9075	0.0964

$$\hat{y}_{(i,j)} = \frac{1}{\mathcal{C}(y)} \sum_{\forall p,q} f(y_{(i,j)}, y_{(p,q)}) g(y_{(p,q)}) \tag{4}$$

where $p \in \{1, 2, ..., w_B\}$ and $q \in \{1, 2, ..., h_B\}$ are the position indexes of different image blocks. The function f is responsible for computing the affinities between the measurements of different image blocks. The unary function g computes a representation of the input measurements $y_{(p,q)}$. From Eq. (4), we can get that all measurements ($\forall$) of image x are considered for the current measurement $y_{(i,j)}$. The factor $\mathcal{C}(y)$ is used to normalize the final response. In view of the function f, an extension of the gaussian function is utilized in our model to compute similarity in an embedding space. Besides, considering the coupling between the non-local information as shown in Eq. (3), we define f as

$$f(y_{(i,j)}, y_{(p,q)}) = \arg\min_{e^{\theta^T \phi}} \| \delta_{((i,j),(p,q))} - \delta_{((p,q),(i,j))} \|_F^2 \tag{5}$$

where $\delta_{((i,j),(p,q))} = e^{\theta(y_{(i,j)})^T \phi(y_{(p,q)})}$, $\delta_{((p,q),(i,j))} = e^{\theta(y_{(p,q)})^T \phi(y_{(i,j)})}$, and $\theta(y_{(i,j)}) = W_\theta y_{(i,j)}$, $\phi(y_{(p,q)}) = W_\phi y_{(p,q)}$ are two embeddings. W_θ and W_ϕ indicate two weight matrices to be learned. T represents the transpose operator of a matrix. As above, we set $\mathcal{C}(y) = \sum_{\forall p,q} f(y_{(i,j)}, y_{(p,q)})$. Considering function g, we also use a linear embedding form: $g(y_{(p,q)}) = W_g y_{(p,q)}$, where W_g is a weight matrix to be learned. In detail, we use series of convolutional layers with n kernels of size 1×1 to optimize the learned matrix W_θ, W_ϕ and W_g, where $n = n_B$ in our model. Besides, a residual structure is utilized, i.e., $\tilde{y}_{(i,j)} = y_{(i,j)} + \hat{y}_{(i,j)}$. Fig. 3 shows the details of the non-local subnetwork in the measurement domain. In addition, it is worth noting that a affinity matrix r is generated in the non-local subnetwork of measurement domain, which is composed of the affinities between different measurements, and the elements in the symmetrical positions represent the affinity coefficients between the current two measurements of referring to each other. After the non-local subnetwork in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3132489, IEEE Transactions on Multimedia

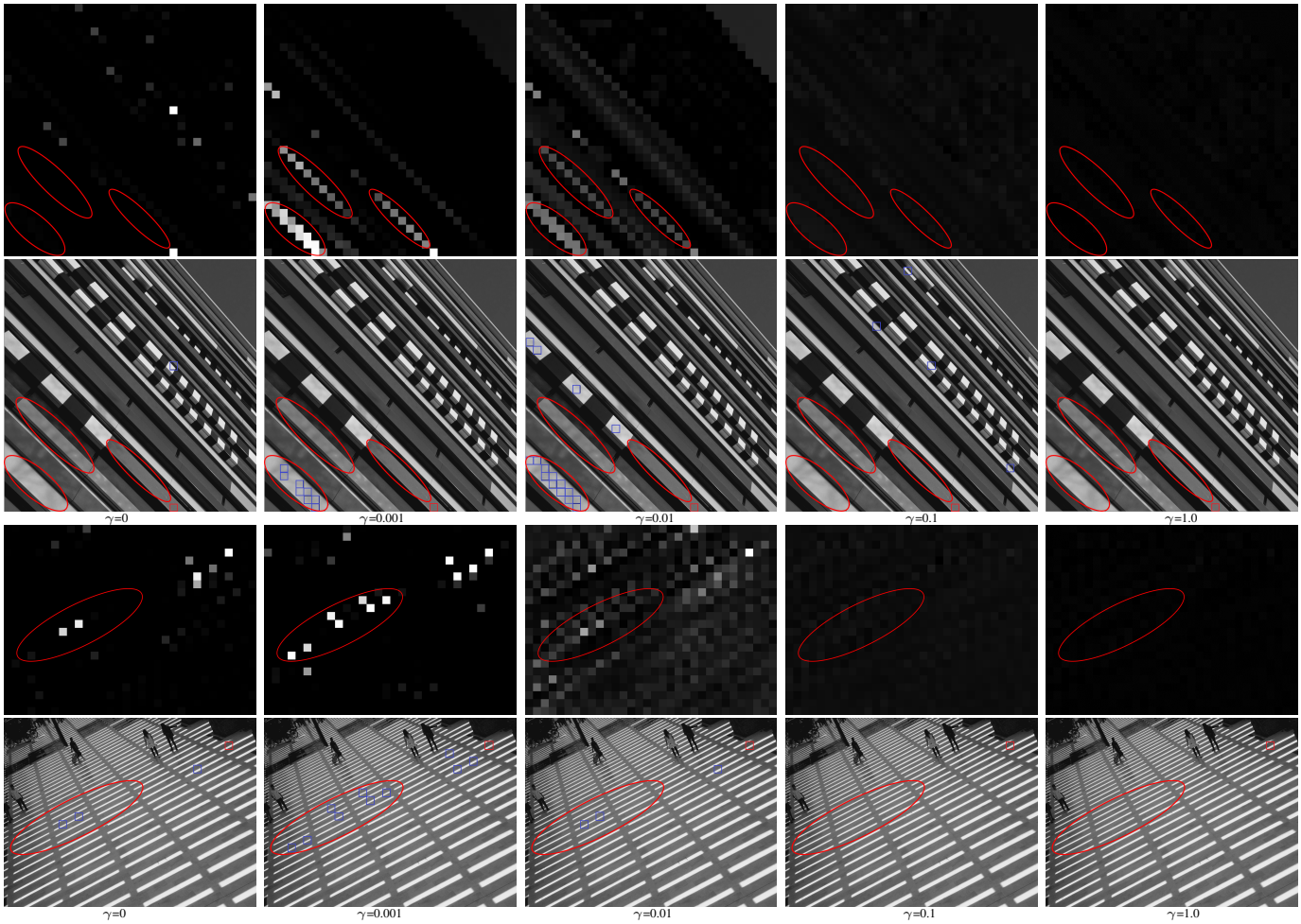


Fig. 7. The visualization of the learned affinity matrix and its corresponding blocks on the original image in terms of different γ. The red blocks are the current image patches and the blue blocks are the corresponding patches mapped from the highlighted elements of affinity matrix.

measurement domain, a series of referenced "measurements" $\tilde{y}_{(i,j)}$ are generated.

In order to reconstruct the initial reconstruction ($\hat{x}$), an upsampling operation is subsequently performed by $\Phi_B \tilde{y}_{(i,j)}$, where Φ_B is a $B^2 \times n_B$ matrix. In our model, we utilize a convolutional layer with B^2 filters of size $1 \times 1 \times n_B$ to learn the upsampling matrix Φ_B (shown in Fig. 2(b)), after which a series of vectors of size $1 \times 1 \times B^2$ are generated. Finaly, two additional operators, i.e., reshape and concatenation (R_t) [6], are appended as shown in Fig. 2(c) to obtain the initial reconstruction. By exploring correlations between the measurements of different image blocks, a better initial reconstruction is usually obtained, which is more favored for the subsequent deep reconstruction. Through the above analysis, we can obtain that the non-local model in the measurement domain builds a coarse-grained reference between the measurements of non-overlapping image blocks. Fig. 1 shows the visual results of the learned affinity matrices in the measurement domain, from which we can observe that our non-local subnetwork is capable of exploring non-local patches with similar structures in the whole image space.

D. Non-Local Subnetwork in Multi-Scale Feature Domain

After the initial reconstruction, the non-local subnetwork in multi-scale feature domain (dubbed MS-NLNet) is appended to further enhance the CS reconstruction. For the structure of MS-NLNet, a series of horizontal and vertical branches are included as shown in Fig. 4 to form a grid architecture [68]. Specifically, the horizontal branch is responsible for the feature extraction and the non-local knowledge exploiting under a certain scale space. The vertical branch mainly focuses on the transformation of different scale spaces, which includes the downsampling and upsampling branches that separately responsible for the down and up sampling operations of the intermediate feature maps. For simplicity, the number of the downsampling and upsampling branches is set as d_u and d_v respectively in our model.

In view of the internal structure details of MS-NLNet, three types of submodules, i.e., downsample, upsample and non-local submodule, are developed. For the first two submodules, the dense connection of the proposed residual blocks (shown in Fig. 4) is utilized and the number of the residual blocks is set as d_l and d_r respectively in these two types of submodules. In addition, it is worth noting that an additional aggregation operator (Conv 1×1) is appended to aggregate the input feature maps together in these two types of submodules. Their structure details are shown in the subgraphs (a) and (b) of Fig. 4. For the non-local submodule, we establish a non-local reference between the feature points (with full channels) in the multi-scale space. For simplicity, we set the scale number of

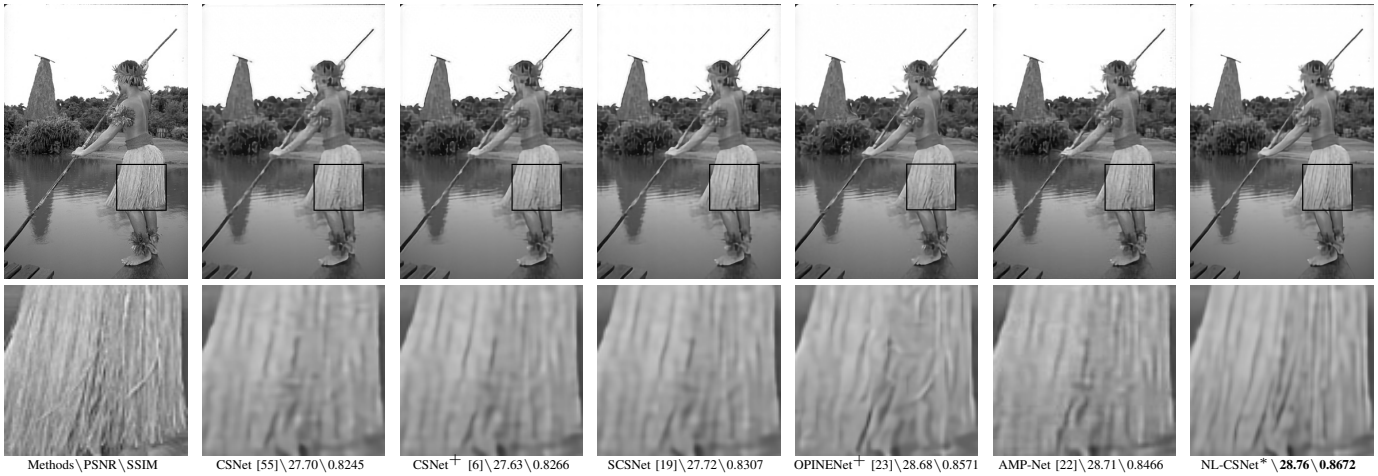


Fig. 8. Visual quality comparisons of deep network-based CS methods using learned sampling matrix on one sample image from BSD68 in case of sampling rate = 0.1.

our MS-NLNet as S_B and there are S_N non-local submodules in each scale space. Therefore, the non-local operation in the multi-scale feature domain can be expressed as:

$$\hat{z}^{s_t}_{(i,j)} = \frac{1}{\mathcal{C}(z^{s_t})} \sum_{\forall p,q} f(z^{s_t}_{(i,j)}, z^{s_t}_{(p,q)}) g(z^{s_t}_{(p,q)}) \tag{6}$$

where $z^{s_t}_{(i,j)}$ indicates the current feature points (with full channels) of the input feature maps in the t-th non-local submodule of the s-th scale space, where $t = \{1, 2, ...S_N\}$ and $s = \{1, 2, ...S_B\}$. (i, j) and (p, q) are the position indexes of the feature points. It is clear that for the current feature point $z^{s_t}_{(i,j)}$, all feature points ($\forall$) in the current feature maps are concerned to generate the corresponding referenced feature point $\hat{z}^{s_t}_{(i,j)}$. In addition, a residual structure is utilized, i.e., $\tilde{z}^{s_t}_{(i,j)} = z^{s_t}_{(i,j)} + \hat{z}^{s_t}_{(i,j)}$. The structure details of the non-local subnetwork in the multi-scale feature domain are shown in Fig. 4. For the functions f and g, they have the similar definitions as that in the measurement domain (shown in Eq. (5)), i.e.,

$$f(z^{s_t}_{(i,j)}, z^{s_t}_{(p,q)}) = \underset{e^{\theta^T \phi}}{\arg\min} \| \delta^{s_t}_{((i,j),(p,q))} - \delta^{s_t}_{((p,q),(i,j))} \|^2_F \tag{7}$$

where $\delta^{s_t}_{((i,j)\{p,q))} = e^{\theta(z^{s_t}_{(i,j)})^T \phi(z^{s_t}_{(p,q)})}$, $\delta^{s_t}_{((p,q)\{i,j))} = e^{\theta(z^{s_t}_{(p,q)})^T \phi(z^{s_t}_{(i,j)})}$, and $\theta(y_{(i,j)}) = W_\theta y_{(i,j)}$, $\phi(y_{(p,q)}) = W_\phi y_{(p,q)}$, $g(y_{(p,q)}) = W_g y_{(p,q)}$. W_θ, W_ϕ and W_g indicate the weight matrices to be learned. Similar to the non-local subnetwork in the measurement domain, we use series of convolutional layers with n kernels of size 1×1 to optimize these learned matrices, where n is equal to the channel number of the input feature maps. Analogically, $\mathcal{C}(z^{s_t}) = \sum_{\forall p,q} f(z^{s_t}_{(i,j)}, z^{s_t}_{(p,q)})$ is used to normalize the final response.

In fact, with the dimensional increase of the input signal, the resource consumptions of the non-local neural networks increase rapidly [65]. In order to alleviate this problem, a downsampling operator ($\downarrow$ in Fig. 4) is used in our non-local neural network. Besides, it is worth noting that an affinity matrix is also generated in each non-local submodule, i.e., r^{s_t} ,which is composed of the affinities between different feature points, and the elements in the symmetrical positions represent the affinity coefficients between two feature points of referring to each other. Fig. 6 shows the visual results of the learned affinity matrices in the multi-scale feature domain, from which we can observe that the non-local textures with the similar structures are explored efficiently. Compared to the non-local subnetwork in the measurement domain, the non-local subnetwork in the multi-scale feature domain builds a fine-grained reference between the dense feature representations of the feature maps.

IV. EXPERIMENTAL RESULTS

In this section, we first elaborate the loss function, and then demonstrate the experimental settings and implementation details as well as the experimental comparisons against the existing state-of-the-art CS methods.

A. Loss Function

Given the input image x_i, the mission of the proposed NL-CSNet is to narrow down the gap between the output and the target image x_i. In addition, considering the coupling constraint between the non-local self-similarity knowledge, the total loss function can be expressed as

$$L = L_r + \gamma L_c \tag{8}$$

where L_r and L_c are the reconstruction loss and the non-local coupling loss respectively. γ is a hyper parameter to control the non-local coupling loss item. Due to the non-local neural network is embedded into the measurement domain and multi-scale feature domain in the proposed NL-CSNet, the non-local coupling loss L_c can be divided into two subitems: $L_c = \gamma_u L_u + \gamma_v L_v$, where L_u and L_v are the non-local coupling loss in terms of these two types of non-local models respectively. γ_u and γ_v are the regularization parameters to control the trade-off between these two subitems.

TABLE VII
THE EXPERIMENTAL RESULTS (PSNR) FOR ANALYZING THE CONTRIBUTIONS OF DIFFERENT FUNCTIONAL MODULES IN TERMS OF VARIOUS SAMPLING RATES ON DATASET BSD68.

Coupling	NLM	MSN	NLF	0.01	0.04	0.1	0.3
✗	✓	✓	✓	22.71	25.36	27.79	32.27
✓	✗	✓	✓	22.78	25.39	27.86	32.28
✓	✓	✗	✓	22.40	25.02	27.49	31.95
✓	✓	✓	✗	22.65	25.25	27.70	32.13
✓	✓	✓	✓	**22.85**	**25.48**	**27.95**	**32.36**

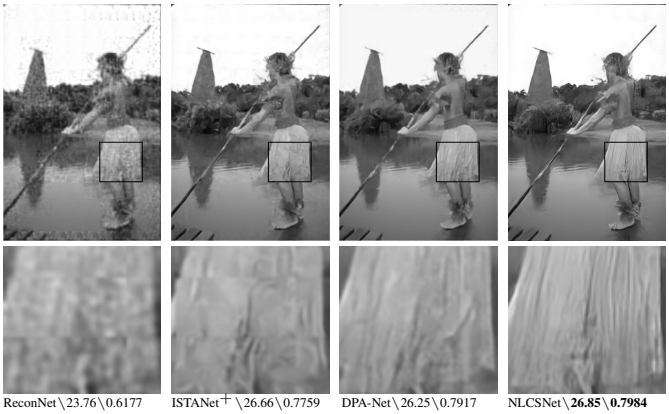


Fig. 9. Visual quality comparisons of deep network-based CS methods using random sampling matrix on one sample image from BSD68 in case of sampling rate = 0.1. (The ground truth image is shown in Fig. 8.)

In order to explain the loss function more conveniently, the outputs of NL-CSNet are defined and analyzed below. Specifically, given the input image x_i, through the pipeline of the proposed entire CS framework, three entities are revealed, i.e., r_i, $\{r_i^{s_t}\}$ and $\tilde{x}_i$. More specifically, r_i indicates the produced affinity matrix of the non-local subnetwork in the measurement domain, $\{r_i^{s_t}\}$ signifies the set of the affinity matrices for the non-local subnetwork in the multi-scale feature domain ($r_i^{s_t}$ is the affinity matrix of the t-th non-local submodule in the s-th scale space, where $s = \{1, 2, ...S_B\}$, $t = \{1, 2, ...S_N\}$). $\tilde{x}_i$ is the revealed reconstructed image.

For the reconstruction loss L_r, we directly use the L2 norm to constrain the distance between the reconstructed image $\tilde{x}_i$ and the ground truth x_i. i.e.,

$$L_r = \frac{1}{2K}\sum_{i=1}^{K}\|\tilde{x}_i - x_i\|_F^2 \tag{9}$$

where K indicates the batchsize of the training data.

For the non-local coupling loss item L_u in the measurement domain, we constrain the affinity matrix r_i according to Eq. (5). Since the elements in the symmetrical positions of r_i are the affinity coefficients between the given two measurements of referencing to each other, the loss function L_u can be expressed as

$$L_u = \frac{1}{2K}\sum_{i=1}^{K}\|r_i - {r_i}^T\|_F^2 \tag{10}$$

where T is the transpose operator of the affinity matrix.

Analogically, for the non-local coupling loss item L_v in the multi-scale feature domain, since multiple non-local submodules are included in the multi-scale space, the loss function L_v can be expressed as

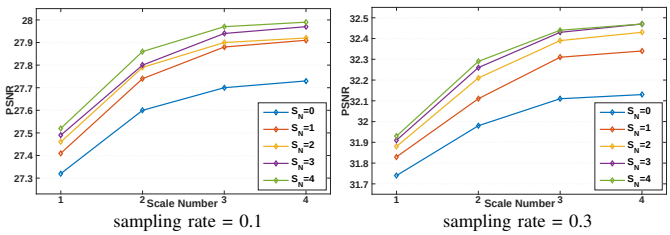
$$L_v = \frac{1}{2K}\sum_{i=1}^{K}\sum_{s=1}^{S_B}\sum_{t=1}^{S_N}\|r_i^{s_t} - r_i^{s_t T}\|_F^2 \tag{11}$$

where S_B indicates the number of the scale spaces and S_N is the number of the non-local submodules in each scale space.

B. Implementation and Training Details

In the proposed CS framework, we set block size B=32 as the previous works [6], [21]. For the hyperparameters S_B and S_N of the deep reconstruction subnetwork MS-NLNet, Fig. 10 reveals the relationship between these two hyperparameters and the reconstruction quality, from which we observe that with the increase of the hyperparameters, the quality becomes more and more insensitive to them. In our model, we set the number of the horizontal branches as 3, which implies 3 scale spaces are considered, i.e., $S_B = 3$. In each scale space, we set the number of non-local submodules as 3, i.e., $S_N = 3$. The kernel number of the convolutional layers (channel number C of the feature maps) in the three horizontal branches are separately set as 16, 32 and 64. Besides, for the vertical branches, we set the number of both downsampling and upsampling branches as 3, i.e., $d_u = d_v = 3$ (two downsampling and upsampling branches are shown in Fig. 4). In terms of the number of the residual blocks, we set $d_l = d_r = 3$ in our model. In addition, the numbers of the residual blocks (d_t) in the non-local submodules of the three horizontal branches are different and we set them as 1, 2 and 3 respectively. To reduce the resource consumptions of the non-local neural network in the multi-scale feature domain, we use the same spatial downsampling trick (↓ in Fig. 4) as [65]. Specifically, for the three scale spaces in our framework, we do downsampling operations at the scale factors of 16, 4 and 1 respectively. Furthermore, for the downsample block and upsample block in the vertical branches, a convolutional layer with stride 2 and a pixelshuffle layer is applied respectively for feature maps downsampling and upsampling. It is worth noting that the kernel size of all the convolutional layers are set as 3×3, except the convolutional layers in the aggregation submodules and the non-local networks. For training configurations, we initialize the convolutional filters using the same method as [69] and pad zeros around the boundaries to keep the size of feature maps the same as the input.

We use training set (400 images) from BSD500 [70] dataset and the training set of VOC2012 [71] as our training data. Specifically, in the training process, we use a batch size of 8 and randomly crop the size of patches to 128×128. We augment the training data in two ways: (1) Rotate the images by 90°, 180° and 270° randomly. (2) Flip the images horizontally with a probability of 0.5. We use the Pytorch toolbox and train our model using the Adaptive moment estimation (Adam) solver on a NVIDIA GTX 1080Ti GPU. In addition, we set $\gamma = 0.001$, $\gamma_u = 1.0$ and $\gamma_v = 1.0$ in our model. We set the momentum to 0.9 and the weight decay to 1e-4. The learning rate is initialized to 1e-4 for all layers and decreased by a factor of 2 for every 30 epochs. We train our model for 200 epochs totally and 1000 iterations are performed for each epoch. Therefore 200×1000 iterations are completed



sampling rate = 0.1 sampling rate = 0.3

Fig. 10. The relationship between the hyperparameters (S_B, S_N) and the reconstruction quality in the proposed model. The X-axis represents the scale number S_B, the Y-axis is the reconstruction quality, and the curves signify the relationship between S_B and reconstruction quality under different S_N.

Fig. 11. Visual quality comparisons of deep network-based CS methods using learned sampling matrix on one sample image from BSD68 in case of sampling rate = 0.1.

for the whole training process.

C. Comparisons with State-of-the-art Methods

To evaluate the performance of the proposed CS framework, we conduct the comparisons against the existing CS methods in terms of two aspects: reconstruction quality comparisons and running speed comparisons. Depending on whether the sampling matrix can be learned, the comparisons of the CS reconstruction methods are provided in two aspects: comparisons with the random sampling matrix-based CS methods and the learned sampling matrix-based CS methods. a) For the random sampling matrix-based CS methods, the optimization-based and the deep network-based reconstruction methods are both concerned. Specifically, considering the optimization-based methods, three representative CS schemes are selected, i.e., TV [53], MH [32] and GSR [31]. In view of the deep network-based methods, more than ten CS algorithms are considered, including: SDA [33], ReconNet [16], LDIT [20], LDAMP [20], DPDNN [63], I-Recon [17], ISTA-Net [21], DR^2-Net [15], IRCNN [42], NN [64], NLR-CSNet [54], DPA-Net [37] and DPIR [44]. b) For the learned sampling matrix-based CS methods, seven recent literatures, i.e., CSNet [55], LapCSNet [35], SCSNet [19], CSNet$^+$ [6], BCS-Net [39], OPINE-Net$^+$ [23] and AMP-Net [22] participate in the comparison in our experiments.

For the fairness of comparison, two model variants, dubbed NL-CSNet and NL-CSNet*, are generated in our experiments by using random sampling matrix and learned sampling matrix. For NL-CSNet, the orthogonalized Gaussian random matrix [37], [54] is utilized in our experiments, and during the training process, the sampling matrix remains unchanged. For NL-CSNet*, the sampling matrix is optimized jointly with the reconstruction network. For testing data, we carry out extensive experiments on several representative benchmark datasets: Set5 [19], Set14 [35], Set11 [72] and BSD68 [37], which are widely used in the previous CS-related works. To ensure the fairness of the comparison, we evaluate the reconstruction performance with two widely used quality evaluation metrics: PSNR and SSIM in terms of various sampling rates.

1) Reconstruction Quality Comparisons: In our experiments, the compared CS methods are divided into the following two groups: random sampling matrix-based CS methods and learned sampling matrix-based CS methods.

Random sampling matrix-based CS methods: For the random sampling matrix-based CS methods, two types of CS algorithms, i.e., the optimization-based methods and the deep network-based methods are mainly concerned. **a)** For the optimization-based methods, since the performance of the method GSR is better than the other two algorithms TV and MH, we mainly analyze the experimental results compared with GSR in our experiments. The quantitative results on dataset Set5 are shown in Table I, from which we can get that the proposed NL-CSNet outperforms other optimization-based CS methods. Specifically, for the given five sampling rates from 0.01 to 0.30 (i.e., 0.01, 0.04, 0.10, 0.20 and 0.30), the proposed framework achieves on average 0.87dB and 0.0331 gains in PSNR and SSIM compared against GSR on the dataset Set5. Especially at low sampling rate (e.g., 0.01), the proposed method can obtain more than 3dB gain compared against GSR. The visual comparisons are displayed in Fig. 5, from which we observe that the proposed NL-CSCNet is capable of preserving more structural details compared with the other optimization-based CS methods. **b)** For the deep network-based CS methods, Tables I and III respectively present the experimental comparisons in terms of the given sampling rates on two datasets Set5 and Set11, from which we can observe that the proposed NL-CSNet achieves competitive or even superior performance against the existing deep network-based CS schemes. The visual comparisons are shown in Fig. 9, from which we observe that the proposed NL-CSCNet is capable of preserving more texture information and recovering richer structural details compared against the other deep network-based CS methods that use random sampling matrix.

Learned sampling matrix-based CS methods: For the learned sampling matrix-based CS methods, the sampling matrix is optimized jointly with the reconstruction module, which facilitates the collaborations between the sampling and reconstruction. For comparative fairness, in our experimental variant NL-CSNet*, the sampling matrix is also optimized jointly with the reconstruction process. Tables II, IV and V separately present the experimental comparisons in terms of the given five sampling rates (i.e., 0.01, 0.04, 0.10, 0.20

and 0.30) on three datasets (Set5, Set11 and Set14), from which we can get that the proposed NL-CSNet* performs much better than the other deep network-based CS schemes. In the compared CS methods, the recent algorithms SC-SNet [19], CSNet$^+$ [6] and AMP-Net [22] can obtain the best reconstruction performance. For simplicity, we mainly analyze the experimental results compared with these three representative CS algorithms. Specifically, (1) On the dataset Set5, the proposed framework achieves on average 0.74dB, 0.87dB, 0.63dB and 0.0188, 0.0203, 0.0143 gains in PSNR and SSIM compared against these three deep network-based CS algorithms in terms of the given sampling rates. (2) On the dataset Set11, the proposed framework achieves on average 1.20dB, 1.34dB, 0.66dB and 0.0325, 0.0343, 0.0216 gains in PSNR and SSIM compared against the three CS methods in terms of different sampling rates. (3) On the dataset Set14, the proposed framework achieves on average 0.73dB, 0.83dB, 0.34dB and 0.0225, 0.0234, 0.0190 gains in PSNR and SSIM in terms of various sampling rates. The visual comparisons are shown in Figs. 8 and 11, from which we observe that the proposed method is capable of preserving more details and retaining sharper edges compared to the other representative deep network-based CS methods.

2) Running Speed Comparisons: To verify the efficiency of the proposed CS framework, we also display the reconstruction speed of different CS methods. Specifically, we evaluate the runtime on the same platform with 3.30 GHz Intel i7 CPU (32G RAM) plus NVIDIA GRX 1080Ti GPU (11G Memory). Table VI shows the average running time comparisons (in second) between different CS methods (including the optimization-based and deep network-based CS methods) for reconstructing a 256×256 image at two sampling rates of 0.01 and 0.10. It is worth noting that the running time result of the algorithm SDA is copied from [16], and for the other CS methods, we test them on the same platform with their source codes downloaded from the authors' websites. In addition, the optimization-based CS schemes are implemented based on CPU device. In contrast, we test all the deep network-based CS methods on both the CPU and GPU devices. The running speed comparison results show that the deep network-based methods run faster than the optimization-based methods. Furthermore, the proposed NL-CSNet basically remains at the same order of magnitude as the other existing deep network-based methods and achieves a more faster reconstruction compared to the other optimization-based CS algorithms.

D. Ablation Studies and Discussions

As noted above, the proposed CS framework achieves higher reconstruction quality. In this subsection, we mainly analyze the contribution of each submodule of the proposed CS framework. For image sampling, as shown in Tables I and II as well as Tables III and IV, the learned sampling matrix obtains more than 2dB gain on average compared to the Gaussian random sampling matrix. For image reconstruction, in order to evaluate the benefits of each part of the proposed CS reconstruction network, we design several counterpart versions of the proposed reconstruction model, in which some functional modules are selectively discarded or retained. Table VII shows the experimental results on the dataset BSD68 [37], in which four functional modules, i.e., non-local coupling loss item (Coupling), non-local in the measurement domain (NLM), multi-scale network architecture (MSN) and non-local in the multi-scale feature domain (NLF) are considered. Specifically, the check marks and the cross marks indicate the reserving and discarding of the corresponding functional modules respectively. It should be noted that when MSN is discarded, we only use the first horizontal branch of MS-NLNet to reconstruct the target image. The experimental results in Table VII reveal that the multi-scale network architecture (MSN) brings a maximum gain, and the non-local in the measurement domain (NLM) brings a minimum gain. Besides, the non-local in the multi-scale feature domain (NLF) can bring more growth against that of the measurement domain (NLM).

In view of the non-local coupling loss item (Coupling) of the proposed model, its mission is to enhance the coupling between the non-local self-similarity knowledge. In order to choose a appropriate regularization parameter γ in Eq. (8), we conduct a large number experimental comparisons in case of different settings of γ. Considering $\gamma = 0$ as a baseline (same with the traditional non-local neural networks [65]), we observe that with the increase of γ, the reconstructed quality can be improved to a certain extent, but when γ is too large, the reconstruction will be corroded slightly. Moreover, the visual comparisons of the affinity matrices produced in the non-local subnetwork of measurement domain in terms of different γ are shown in Fig. 7, from which we can get that when $\gamma = 0$, the localization of similar patches is inefficient, and when γ is too large, the learned affinities are tend to 0. Through experimental analysis, we finally set γ as 0.001. It is worth noting that when the non-local coupling loss item (Coupling) is discarded, it actually is the traditional non-local neural network [65]. Table VII shows that the proposed non-local coupling loss item is capable of improving the CS reconstruction performance to a certain extent. As mentioned above, each of the aforementioned four functional modules in the proposed NL-CSNet can enhance the CS reconstruction quality to varying degrees.

V. Conclusion

In this paper, we propose a novel image compressed sensing framework using non-local neural network, which utilizes the non-local self-similarity priors with deep network to improve the reconstruction quality. In the proposed model, two non-local subnetworks are designed to establish the long-distance references between the non-local self-similarity knowledge in the measurement domain and multi-scale feature domain respectively. Specifically, in the subnetwork of measurement domain, the affinities between the measurements of different image blocks are explored for mining the interblock correlations. Analogically, in the subnetwork of multi-scale feature domain, the dependencies between the non-local feature representations are explored for building a reference mechanism between the non-local structural textures. Experimental results demonstrate that the proposed CS framework achieves much higher reconstruction performance and better perceptual image quality against other state-of-the-art CS methods.

REFERENCES

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

[2] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[3] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, 2008.

[4] J. H. G. Ender, "On compressive sensing applied to radar," *Signal Processing*, vol. 90, no. 5, pp. 1402–1414, 2010.

[5] S. Li, L. D. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, 2013.

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

[7] L. Gan, "Block compressed sensing of natural images," *Proceedings of the international conference on digital signal processing*, pp. 403–406, 2007.

[8] M. Vehkaper, Y. Kabashima, and S. Chatterjee, "Analysis of regularized ls reconstruction and random matrix ensembles in compressed sensing," *IEEE Transactions on Information Theory (TIT)*, vol. 62, no. 4, pp. 2100–2124, 2016.

[9] W. Cui, F. Jiang, X. Gao, W. Tao, and D. Zhao, "Deep neural network based sparse measurement matrix for image compressed sensing," *IEEE International Conference on Image Processing (ICIP)*, pp. 3883–3887, 2018.

[10] X. Gao, J. Zhang, W. Che, X. Fan, and D. Zhao, "Block-based compressive sensing coding of natural images by local structural measurement matrix," *IEEE Data Compression Conference (DCC)*, pp. 133–142, 2015.

[11] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[12] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory (TIT)*, vol. 53, no. 12, pp. 4655–4666, 2007.

[13] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[14] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[15] H. Yao, F. Dai, D. Zhang, Y. Ma, S. Zhang, Y. Zhang, and Q. Tian, "DR2-Net: Deep residual reconstruction network for image compressive sensing," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-iterative reconstruction of images from compressively sensed measurements," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 449–458, 2016.

[17] S. Lohit, K. Kulkarni, R. Kerviche, P. Turaga, and A. Ashok, "Convolutional neural networks for noniterative reconstruction of compressively sensed images," *IEEE Transactions on Computational Imaging (TCI)*, 2018.

[18] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu, "DR2-Net: Deep residual reconstruction network for image compressive sensing," *Neurocomputing*, vol. 359, no. SEP.24, pp. 483–493, 2019.

[19] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Scalable convolutional neural network for image compressed sensing," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12290–12299, 2019.

[20] C. A. Metzler, A. Mousavi, and R. G. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," *International Conference on Neural Information Processing Systems (NIPS)*. Red Hook, NY, USA: Curran Associates Inc., pp. 1770–1781, 2017.

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

[22] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu, "AMP-Net: Denoisingbased deep unfolding for compressive image sensing," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 1487–1500, 2021.

[23] J. Zhang, C. Zhao, and W. Gao, "Optimization-inspired compact deep compressive sensing," *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, vol. 14, no. 4, pp. 765–774, 2020.

[24] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[25] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

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

[27] C. Li, W. Yin, and Y. Zhang, "Tval3: Tv minimization by augmented lagrangian and alternating direction agorithm 2009," 2013.

[28] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Modelbased compressive sensing," *IEEE Transactions on Information Theory (TIT)*, vol. 56, no. 4, pp. 1982–2001, 2010.

[29] C. Hegde, P. Indyk, and L. Schmidt, "A fast approximation algorithm for tree-sparse recovery," *IEEE International Symposium on Information Theory*, pp. 1842–1846, 2014.

[30] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 8, pp. 3618–3632, 2014.

[31] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 8, pp. 3336–3351, 2014.

[32] 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)*, pp. 1193–1198, 2011.

[33] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," *in 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1336–1343, 2015.

[34] W. Li, F. Liu, L. Jiao, and F. Hu, "Multi-scale residual reconstruction neural network with non-local constraint," *IEEE Access*, vol. 7, pp. 70910–70918, 2019.

[35] W. Cui, H. Xu, X. Gao, S. Zhang, F. Jiang, and D. Zhao, "An efficient deep convolutional laplacian pyramid architecture for cs reconstruction at low sampling ratios," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[36] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing (TIP)*, vol. 16, no. 8, pp. 2080–2095, 2007.

[37] Y. Sun, J. Chen, Q. Liu, B. Liu, and G. Guo, "Dual-path attention network for compressed sensing image reconstruction," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 9482–9495, 2020.

[38] K. Xu, Z. Zhang, and F. Ren, "Lapran: A scalable laplacian pyramid reconsructive adversarial network for flexible compressive sensing reconstruction," *Springer European Conference on Computer Vision (ECCV)*, pp. 491–507, 2018.

[39] S. Zhou, Y. He, Y. Liu, C. Li, and J. Zhang, "Multichannel deep networks for block-based image compressive sensing," *IEEE Transactions on Multimedia (TMM)*, doi:10.1109/TMM.2020.3014561, 2020.

[40] 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 (TIP)*, vol. 26, no. 7, pp. 3142–3155, 2017.

[41] Y. Romano, M. Elad, and P. Milanfar, "The Little Engine that Could: Regularization by Denoising (RED)," *Siam Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[42] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2808–2817, 2017.

[43] J. Liu, Y. Sun, C. Eldeniz, W. Gan, H. An, and U. S. Kamilov, "RARE: Image reconstruction using deep priors learned without groundtruth," *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, vol. 14, no. 6, pp. 1088–1099, 2020.

[44] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, doi: 10.1109/TPAMI.2021.3088914, 2021.

[45] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 6, pp. 1256–1272, 2017.

[46] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 60–65, 2005.
[47] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing (TIP)*, vol. 22, no. 4, pp. 1620–1630, 2013.
[48] S. Kindermann, S. Osher, and P. W. Jones, "Deblurring and denoising of images by nonlocal functionals," *Siam Journal on Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1091–1115, 2005.
[49] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE Transactions on Image Processing (TIP)*, vol. 17, no. 7, pp. 1047–1060, 2008.
[50] M. Jung, X. Bresson, T. F. Chan, and L. A. Vese, "Nonlocal mumford-shah regularizers for color image restoration," *IEEE Transactions on Image Processing (TIP)*, vol. 20, no. 6, pp. 1583–1598, 2011.
[51] J. Zhang, D. Zhao, C. Zhao, R. Xiong, S. Ma, and W. Gao, "Compressed sensing recovery via collaborative sparsity," *IEEE Data Compression Conference (DCC)*, pp. 287–296, 2012.
[52] C. Zhao, J. Zhang, S. Ma, and W. Gao, "Nonconvex Lp Nuclear Norm based ADMM Framework for Compressed Sensing," *IEEE Data Compression Conference (DCC)*, pp. 161–170, 2016.
[53] C. Li, W. Yin, H. Jiang, and Y. Zhang, "An efficient augmented lagrangian method with applications to total variation minimization," *Computational Optimization and Applications*, vol. 56, no. 3, pp. 507–530, 2013.
[54] Y. Sun, Y. Yang, Q. Liu, J. Chen, X. T. Yuan, and G. Guo, "Learning non-locally regularized compressed sensing network with half-quadratic splitting," *IEEE Transactions on Multimedia (TMM)*, vol. 22, no. 12, pp. 3236–3248, 2020.
[55] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 877–882, 2017.
[56] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9446–9454, 2018.
[57] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[58] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.
[59] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[60] D. Valsesia, G. Fracastoro, and E. Magli, "Image denoising with graphconvolutional neural networks," *IEEE International Conference on Image Processing (ICIP)*, pp. 2399–2403, 2019.
[61] G. F. Diego Valsesia and E. Magli, "Deep graph-convolutional image denoising," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 8226–8237, 2020.
[62] S. Zhou, J. Zhang, W. Zuo, and C. C. Loy, "Cross-scale internal graph neural network for image super-resolution," *arXiv preprint arXiv: 2006.16673*, 2020.
[63] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 41, no. 10, pp. 2305–2318, 2019.
[64] D. Gilton, G. Ongie, and R. Willett, "Neumann networks for linear inverse problems in imaging," *IEEE Transactions on Computational Imaging (TCI)*, vol. 6, pp. 328–343, 2020.
[65] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, 2018.
[66] K. Yue, "Compact generalized non-local network," *Conference and Workshop on Neural Information Processing Systems(NIPS)*, 2018.
[67] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, Ccnet: Criss-cross attention for semantic segmentation, in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 603–612, 2019.
[68] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," *arXiv preprint arXiv:1707.07958*, 2017.
[69] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.
[70] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 5, pp. 898–916, 2011.
[71] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (VOC2012) development kit," *Pattern Analysis, Statistical Modelling and Computational Learning*, 2011.
[72] J. Chen, Y. Sun, Q. Liu, and R. Huang, "Learning memory augmented cascading network for compressed sensing of images," *Springer European Conference on Computer Vision (ECCV)*, pp. 513–529, 2020.

Wenxue Cui received the bachelors degree from Northeast Forestry University, Harbin, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology (HIT), Harbin, China. Since 2018, he has been with the Peng Cheng Laboratory. His research interests include data compression, image and video processing, computer vision, multimedia security and deep network.

Shaohui Liu received the B.S. degree in computation mathematics and its application software, the M.S. degree in computation mathematics, and the Ph.D. degree in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 1999, 2001, and 2007, respectively. Since 2018, he has been with the Peng Cheng Laboratory. He is currently an Associate Professor with the School of Computer Science and Technology, HIT. His research interests include data compression, image and video processing, and multimedia security.

Feng Jiang received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 2001, 2003, and 2008, respectively. Since 2018, he has been with the Peng Cheng Laboratory. He is currently a Professor with the Department of Computer Science, HIT, and a Visiting Scholar with the School of Electrical Engineering, Princeton University. His research interests include computer vision, pattern recognition, and image and video processing.

Debin Zhao received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 1985, 1988, and 1998, respectively. He is currently a Professor with the Department of Computer Science, HIT. Since 2018, he has been with the Peng Cheng Laboratory. He has published over 200 technical articles in refereed journals and conference proceedings in the areas of image and video coding, video processing, video streaming and transmission, and deep network.