Optimization-Inspired Compact Deep Compressive Sensing

Jian Zhang, *Member, IEEE*, Chen Zhao, and Wen Gao, *Fellow, IEEE*

***Abstract*—In order to improve CS performance of natural images, in this paper, we propose a novel framework to design an OPtimization-INspired Explicable deep Network, dubbed OPINE-Net, for adaptive sampling and recovery. Both orthogonal and binary constraints of sampling matrix are incorporated into OPINE-Net simultaneously. In particular, OPINE-Net is composed of three subnets: sampling subnet, initialization subnet and recovery subnet, and all the parameters in OPINE-Net (e.g. sampling matrix, nonlinear transforms, shrinkage threshold) are learned end-to-end, rather than hand-crafted. Moreover, considering the relationship among neighboring blocks, an enhanced version OPINE-Net$^+$ is developed, which allows image blocks to be sampled independently but reconstructed jointly to further enhance the performance. In addition, some interesting findings of learned sampling matrix are presented. Compared with existing state-of-the-art network-based CS methods, the proposed hardware-friendly OPINE-Nets not only achieve better performance but also require much fewer parameters and much less storage space, while maintaining a real-time running speed.**

***Index Terms*—Image reconstruction, neural networks, optimization, compressive sensing, interpretable networks.**

I. Introduction

COMPRESSIVE Sensing (CS) theory demonstrates that a signal can be reconstructed with high probability from much fewer acquired measurements than determined by Nyquist sampling theory, when it exhibits sparsity in some transform domain [1], [2]. This novel acquisition strategy is much more hardware-friendly and it enables image or video capturing with a sub-Nyquist sampling rate [3], [4]. In addition, by exploiting the redundancy existing in a signal, CS conducts sampling and compression at the same time, which greatly alleviates the need for high transmission bandwidth and large storage space, enabling low-cost on-sensor data compression. CS has been applied in many practical applications, including but not limited to single-pixel imaging [2], [5], accelerating magnetic resonance imaging (MRI) [6], wireless tele-monitoring [7] and cognitive radio communication [8].

Mathematically, for the original signal $\mathbf{x} \in \mathbb{R}^N$, in the sampling process, its CS measurements are obtained by $\mathbf{y} = \mathbf{\Phi x} \in \mathbb{R}^M$. Here, $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is a linear random projection (matrix). Then, in the recovery process, the purpose is to infer $\mathbf{x}$ from $\mathbf{y}$. Because $M \ll N$, this inverse problem is typically ill-posed, whereby the CS ratio is defined as $\frac{M}{N}$. In this paper, we mainly focus on CS sampling and recovery of natural images.

In the past decade, sparse representation model [9], which assumes that natural images can be sparsely represented by a dictionary, has achieved great success in image processing and compressive sensing [10]–[12]. Recently, researchers realize simultaneously optimizing the sampling matrix and the dictionary for the CS system yields a better signal recovery performance [13]–[15]. Concretely, traditional methods usually consider the problem of simultaneously learning sampling matrix and sparsifying dictionary by exploiting some structured sparsity as an image prior and then solve a sparsity-regularized optimization [13], [16]. Although these methods enjoy the advantage of interpretability, they inevitably suffer from high computational complexity, and they are also faced with the challenges of tuning parameters in their solvers. Fueled by the powerful learning ability of deep networks, several deep network-based image CS algorithms have been recently proposed to jointly optimizing the sampling matrix and the non-linear reconstruction operator [17]–[20]. Compared to optimization-based algorithms, these non-iterative algorithms dramatically reduce time complexity, while achieving impressive reconstruction performance. However, existing network-based CS algorithms for adaptive sampling and recovery are all trained as a *black box*, which limits the insights from the CS domain.

To address the above drawbacks, we combine the merits of both optimization-based and network-based methods and propose a novel optimization-inspired explicable deep network, dubbed OPINE-Net, for adaptive sampling and recovery of image CS. All the parameters involved in OPINE-Net (e.g. nonlinear transforms, shrinkage threshold, step size, etc.) are learned end-to-end using back-propagation. As such, OPINE-Nets enjoy the advantages of fast and accurate reconstruction with well-defined interpretability. As far as we know, OPINE-Net is the first work that maps an optimization problem into deep network for joint adaptive binary sampling and recovery of image CS.

Manuscript received July 1, 2019; revised October 2, 2019; accepted February 15, 2020. Date of publication March 2, 2020; date of current version August 10, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61902009 and in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2017A030310576. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lixin Fan. *(Corresponding author: Jian Zhang.)*

Jian Zhang is with the School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen 518055, China, and also with the Peng Cheng Laboratory, Shenzhen 518052, China (e-mail: zhangjian@pkusz.edu.cn).

Chen Zhao is with the Visual Computing Center, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia (e-mail: chen.zhao@kaust.edu.sa).

Wen Gao is with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China, and also with the Peng Cheng Laboratory, Shenzhen 518052, China (e-mail: wgao@pku.edu.cn).

Digital Object Identifier 10.1109/JSTSP.2020.2977507

In summary, our main contributions are four-fold:

- We present a constrained optimization framework for adaptive sampling and recovery of image CS, and we further propose to solve it with a two-step scheme, based on which we are able to efficiently design our deep network OPINE-Net.
- We propose to incorporate the binary and orthogonal constraints for sampling matrix and the weight-sharing strategy into OPINE-Net simultaneously, which makes the whole network much more hardware-friendly and memory-saving.
- We propose an enhanced multi-block version of OPINE-Net, dubbed OPINE-Net$^+$, by exploiting the inter-block relationship to improve image quality. Compared with other network-based image CS methods, the proposed OPINE-Nets not only achieve the best performance but also have much fewer network parameters and much smaller model size.
- Three interesting findings of learned sampling matrix are presented, which fully verify the feasibility of data-driven joint learning of sampling and recovery for CS.

II. Related Work

According to the way of generating the sampling matrix, we generally group existing CS methods of natural images into two categories: *fixed random Gaussian matrix* and *data-driven adaptively learned matrix*. In what follows, we give a brief review.

Fixed Random Gaussian Matrix: In this case, the sampling matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ is constructed by generating a random Gaussian matrix and then orthogonalizing its rows, i.e. $\mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Applying $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$ yields the CS measurements of the original image $\mathbf{x}$. Then, given $\mathbf{\Phi}$ and $\mathbf{y}$, traditional image CS methods usually reconstruct $\mathbf{x}$ by solving the following optimization problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{\Phi}\mathbf{D}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1, \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{N\times L}$ denotes a sparsifying dictionary, $\boldsymbol{\theta} \in \mathbb{R}^{L\times 1}$ denotes the representation coefficients of $\mathbf{x}$ over $\mathbf{D}$ and the sparsity of the vector $\boldsymbol{\theta}$ is encouraged by the ℓ_1 norm with λ being the regularization parameter. After solving Eq. (1) to obtain $\hat{\boldsymbol{\theta}}$, the CS recovered image is $\hat{\mathbf{x}} = \mathbf{D}\hat{\boldsymbol{\theta}}$.

Many classic domains (e.g. DCT, wavelet [21], and gradient domain [22]) and prior knowledge about transform coefficients (e.g. statistical dependencies [23], structure [24], etc.) have been applied in modeling Eq. (1) [25]–[27]. These traditional image CS reconstruction methods usually require hundreds of iterations to solve Eq. (1) by means of various iterative solvers (e.g. ISTA [28], ADMM [22], or AMP [29]). Quite recently some fast and effective convolutional neural network (CNN) denoisers are trained and integrated into Half Quadratic Splitting (HQA) [30] and alternating direction method of multipliers (ADMM) [31], [32] to solve image inverse problems.

Recently, several deep network-based image CS reconstruction algorithms have been proposed to learn the representation from training data and to reconstruct test data from their CS measurements [33]–[36]. Furthermore, the tremendous success of deep learning for many image processing applications has also led researchers to consider relating iterative optimization methods to neural net-works [37]–[41]. In particular, some optimization-inspired deep unrolling networks are proposed to achieve state-of-the-art performance for CS recovery in the case of fixed random Gaussian matrix [42]–[44].

Data-Driven Adaptively Learned Matrix: In this case, the sampling matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ is adaptively learned by the training dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_b}\}$. To optimize the sampling matrix and the dictionary simultaneously, traditional methods usually formulate it by minimizing the following problem:

$$\min_{\mathbf{D},\mathbf{\Phi},\boldsymbol{\theta}_i} \sum_i^{N_b} \left\{ \frac{1}{2}\|\mathbf{\Phi}\mathbf{D}\boldsymbol{\theta}_i - \mathbf{\Phi}\mathbf{x}_i\|_2^2 + \beta\|\mathbf{D}\boldsymbol{\theta}_i - \mathbf{x}_i\|_2^2 + \lambda\|\boldsymbol{\theta}_i\|_1 \right\}, \tag{2}$$

where $\boldsymbol{\theta}_i$ denotes the representation coefficients of each $\mathbf{x}_i$ over $\mathbf{D}$ [13], [16]. The above problem can be solved by utilizing the alternating-minimization based methods. The main idea is to alternatively update one variable while fixing the others. After obtaining the learned $\mathbf{D}$ and $\mathbf{\Phi}$, CS recovery problem will become Eq. (1). Based on some well-studied image formation models, these methods enjoy the advantage of well-defined interpretability. However, they usually require hundreds of iterations to solve Eq. (1) for CS recovery, which inevitably gives rise to high computational cost. In addition, Eq. (2) only works well for small image patches (such as 8×8), since solving Eq. (2) will become inefficient and even impractical if the dimension of the dictionary is high or the size of training dataset is very large [13], [16].

Lately, some deep networks are developed to jointly optimizing the sampling matrix and the non-linear recovery operator [17]–[20], [45], [46]. In particular, Adler *et al*. propose to utilize a fully-connected network to perform both the block-based linear sensing and non-linear reconstruction stages. Lohit *et al*. propose to add one fully-connected layer as the sampling matrix in front of ReconNet [35] for simultaneous sampling and recovery. Shi *et al*. [19] and Du *et al*. [18] separately propose to adopt a convolution layer to mimic the sampling matrix and utilize all-convolutional networks for CS recovery. Obviously, the network-based CS methods not only jointly train the sampling and recovery stages, but also are non-iterative, which dramatically reduces time complexity as compared with their optimization-based counterparts. However, existing networks for joint learning of sampling matrix and recovery operator are either fully-connected or repetitive convolutional layers. We believe that their lack of structural diversity is the bottleneck for further performance improvement.

In order to address the drawbacks of existing networks-based CS methods in the case of data-driven adaptively learned matrix and inspired by the success of optimization-inspired network in the case of fixed random Gaussian matrix, we propose to a novel optimization-inspired deep structured network, dubbed OPINE-Net, for adaptive sampling and recovery of image CS. We will detail our OPINE-Net in next section.

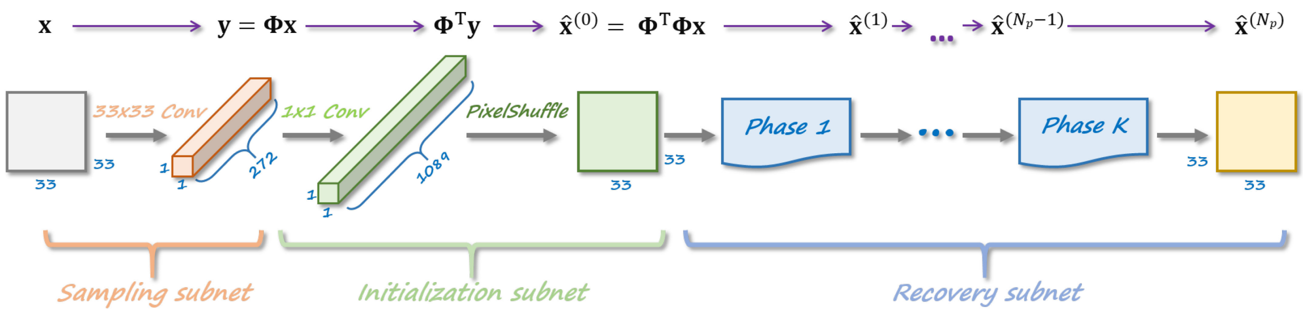

Fig. 1. Illustrations of our proposed OPINE-net framework. Specifically, OPINE-Net is composed of three subnets: Sampling Subnet (SS), Initialization Subnet (IS) and Recovery Subnet (RS).

III. Proposed OPINE-Net Framework

In this section, we first present a constrained optimization framework for adaptive sampling and recovery of image CS. Then, we propose to solve it with a two-step scheme, based on which we are able to efficiently design our deep network OPINE-NET. Finally, in order to address the issue of blocking artifacts introduced by block-based sampling and recovery, we extend the intra-block training of OPINE-NET to inter-block training to enhance the CS recovery quality. Compared with other network-based image CS methods, the proposed OPINE-Net not only achieves the best performance but also has fewer network parameters. More details are provided below.

A. Problem Formulation

Assume we have a training dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_b}\}$, where N_b is the number of image blocks. Instead of using the synthesis sparse model as in Eq. (2), we adopt a constrained analysis sparse model and introduce a nonlinear sparsifying transform $\mathcal{F}$ for modelling the image CS problem. Without losing generality, we consider two typical types of constraints associated with $\mathbf{\Phi}$. One is $\mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. To facilitate practical hardware implementation, we further introduce a second constraint — $\mathbf{\Phi}$ is binary, i.e. each element of $\mathbf{\Phi}$ is either 1 or -1. Both constraints are represented as the set $\mathbf{\Omega}(\mathbf{\Phi})$, and thus the proposed optimization framework is formulated as

$$\min_{\hat{\mathbf{x}}_i, \mathbf{\Phi}, \mathcal{F}} \sum_{i}^{N_b} \left\{ \frac{1}{2} \|\mathbf{\Phi}\hat{\mathbf{x}}_i - \mathbf{\Phi}\mathbf{x}_i\|_2^2 + \lambda \|\mathcal{F}(\hat{\mathbf{x}}_i)\|_1 \right\} \quad \text{s.t.} \quad \mathbf{\Omega}(\mathbf{\Phi}), \tag{3}$$

where $\hat{\mathbf{x}}_i$ denotes the recovered image block.

Next, to map the optimization in Eq. (3) in an efficient way, we propose to implement it in a two-step scheme. Concretely, the first step is to design the network architecture based on the unconstrained version of Eq. (3), i.e. Eq. (3) without the constraints $\mathbf{\Omega}(\mathbf{\Phi})$. Then, the second step is to enforce the constraints $\mathbf{\Omega}(\mathbf{\Phi})$ back by incorporating them into the network to form a complete OPINE-Net.

B. Architecture Design of OPINE-Net

In this subsection, we will elaborate on the architecture design of the proposed OPINE-Net according to Eq. (3) without $\mathbf{\Omega}(\mathbf{\Phi})$. Fig. 1 illustrates the overall architecture of OPINE-Net, which is composed of three sub-networks: sampling subnet, initialization subnet, and recovery subnet. We will describe the design of these three sub-networks in detail in the following subsections.

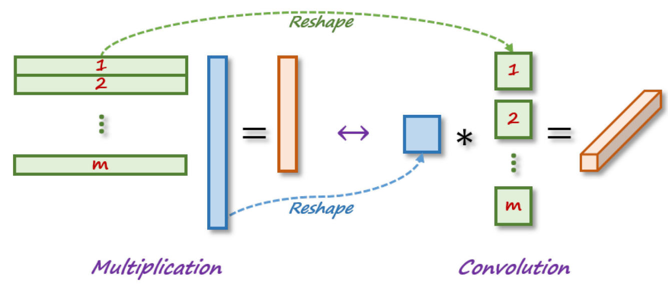

Fig. 2. Illustration of the equivalent transformation from matrix multiplication to matrix convolution.

1) Sampling Subnet (SS): In this paper, we denote a block of size $\sqrt{N} \times \sqrt{N}$ by its vector representation $\mathbf{x} \in \mathbb{R}^N$, and the linear measurements of a block by $\mathbf{y} \in \mathbb{R}^M$, which is obtained via $\mathbf{y} = \mathbf{\Phi x}$, where $\mathbf{\Phi}$ is a measurement matrix.

Viewing the measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ as a learnable network parameter, we reshape it into M filters in the same way, i.e. each of which is of size $\sqrt{N} \times \sqrt{N}$, as shown in Fig. 2. By this means, we can equivalently mimic the CS sampling process $\mathbf{y} = \mathbf{\Phi x} \in \mathbb{R}^M$ using a convolutional layer without bias, which we call sampling subnet (SS). Fig. 1 illustrates a concrete example of sampling an image block $\mathbf{x}$ of size 33×33 with CS sampling rate 25%. The sampling subnet exploits a convolution layer using 272 filters of size 33×33 to obtain the CS measurements $\mathbf{y}$, which is represented by a tensor of size $1 \times 1 \times 272$. Note that the advantage of using a convolutional layer in SS is that it can be easily extended to multi-block training, which will be shown in the following.

2) Initialization Subnet (IS): Inspired by traditional optimization, given $\mathbf{y} = \mathbf{\Phi x}$ as the output of the SS, the proposed initialization subnet (IS) utilizes $\mathbf{\Phi}^\top \mathbf{y}$ as the OPINE-net initialization, denoted by $\hat{\mathbf{x}}^{(0)}$. To be concrete, IS is composed of two consecutive operations: a 1×1 convolution layer and a pixelshuffle layer. We first reshape $\mathbf{\Phi}^\top \in \mathbb{R}^{N \times M}$ into N filters, each of which is of kernel size $1 \times 1 \times M$. With these filters, a 1×1 convolution layer is utilized to obtain $\mathbf{\Phi}^\top \mathbf{y}$, which is actually a tensor of size $1 \times 1 \times N$. Then, we adopt the pixelshuffle layer to reshape a tensor $1 \times 1 \times N$ into a tensor $\sqrt{N} \times \sqrt{N} \times 1$.

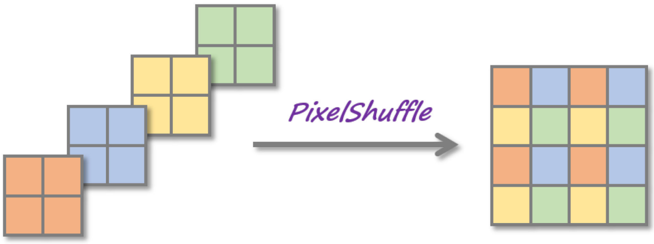

Fig. 3. Illustration of the pixelshuffle operation in the proposed initialization subnet (IS).

The pixelshuffle layer is clearly depicted in Fig. 3, which was first introduced for sub-pixel convolution for image super-resolution [47]. As shown in Fig. 1, a tensor of size $1 \times 1 \times 272$ is transformed into a tensor of size $33 \times 33 \times 1$ through IS. In fact, IS is an efficient convolutional implementation of $\mathbf{\Phi}^\top$, which can be easily extended to multi-block training and will also be used in the following recovery subnet.

Compared with existing deep network based CS methods that introduce extra $M \times N$ parameters for initialization [17]–[20], our proposed IS only utilizes $\mathbf{\Phi}$ and requires no extra parameters. Because the amount $M \times N$ is usually quite large, our proposed OPINE-Net clearly reduces the number of network parameters.

3) Recovery Subnet (RS): Regarding $\mathbf{\Phi}$ and $\mathcal{F}$ as learnable network parameters and given the measurements $\mathbf{\Phi x}$, Eq. (3) without the constraints $\mathbf{\Omega}(\mathbf{\Phi})$ is reduced to the following expression (subscript i is omitted without confusion):

$$\min_{\hat{\mathbf{x}}} \frac{1}{2}\|\mathbf{\Phi}\hat{\mathbf{x}} - \mathbf{\Phi x}\|_2^2 + \lambda\|\mathcal{F}(\mathbf{x})\|_1. \tag{4}$$

Obviously, Eq. (4) becomes the CS recovery with fixed $\mathbf{\Phi}$. As discussed before, several recent optimization inspired networks are developed for CS recovery, such as ISTA-Net$^+$ and ADMM-Net [42], [44]. Considering the simplicity and interpretability, in this paper, we adopt the framework of ISTA-Net$^+$ to efficiently solve Eq. (4). However, it is worth noting that the proposed OPINE-Net can also be trivially extended to other deep unrolling networks to solve Eq. (4).

To be specific, Eq. (4) can be efficiently solved with iterative shrinkage-thresholding algorithm (ISTA) by iterating the following two update steps:

$$\mathbf{r}^{(k)} = \hat{\mathbf{x}}^{(k-1)} - \rho\mathbf{\Phi}^\top(\mathbf{\Phi}\hat{\mathbf{x}}^{(k-1)} - \mathbf{\Phi x}), \tag{5}$$

$$\hat{\mathbf{x}}^{(k)} = \arg\min_{\hat{\mathbf{x}}} \frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{r}^{(k)}\|_2^2 + \lambda\|\mathcal{F}(\hat{\mathbf{x}})\|_1. \tag{6}$$

ISTA-Net$^+$ consists of a fixed number of phases, and each phase corresponds to one iteration in traditional ISTA, as illustrated in Fig. 4. In particular, each phase of ISTA-Net$^+$ is composed of $\mathbf{r}^{(k)}$ and $\hat{\mathbf{x}}^{(k)}$ modules, which are corresponding to the above two update steps Eq. (5) and Eq. (6).

Here, to preserve the ISTA structure, $\mathbf{r}^{(k)}$ **module** is directly defined according to Eq. (5), in which the step size ρ becomes a learnable parameter.

To map Eq. (6) into deep network, first define a linear operator $\mathcal{R}(\cdot)$ as $\mathcal{R} = \mathcal{G} \circ \mathcal{D}$, where $\mathcal{D}$ corresponds to N_f filters (each of size 3×3 in the experiments). Then define $\mathcal{F} = \mathcal{H} \circ \mathcal{D}$, where $\mathcal{H}$ consists of two linear convolutional operators and one rectified linear unit (ReLU). Next, define the left inverse of $\mathcal{H}$ as $\widetilde{\mathcal{H}}$, i.e., satisfying the symmetry constraint $\widetilde{\mathcal{H}} \circ \mathcal{H} = \mathcal{I}$. Therefore, with the learnable parameters $\{\mathcal{H}, \widetilde{\mathcal{H}}, \mathcal{D}, \mathcal{G}, \delta\}$, the $\hat{\mathbf{x}}^{(k)}$ **module** to solve Eq. (6) is expressed as:

$$\hat{\mathbf{x}}^{(k)} = \mathbf{r}^{(k)} + \mathcal{G}(\widetilde{\mathcal{H}}(soft(\mathcal{H}(\mathcal{D}(\mathbf{r}^{(k)})), \delta))). \tag{7}$$

Different from ISTA-Net$^+$, in this paper, we generalize $\mathcal{G}$, and set it as a composition of several convolutional operators and ReLUs. Furthermore, our recovery subnet (RS) includes N_p phases. From experiments, we found sharing weights across all the phases does not affect the final performance in adaptive sampling and recovery. Therefore, we restrict that each phase in our RS shares the same weights, which greatly reduces the number of parameters in RS.

C. Constraint Incorporation

In this subsection, we will show how to incorporate the two constraints in $\mathbf{\Omega}(\mathbf{\Phi})$ into OPINE-Net simultaneously.

For the orthogonal constraint $\mathbf{\Phi\Phi}^\top = \mathbf{I}$, we design an orthogonal loss term, denoted by $\mathcal{L}_{orth} = \frac{1}{M^2}\|\mathbf{\Phi\Phi}^\top - \mathbf{I}\|_F^2$, and propose to directly enforce this constraint into the loss function of OPINE-Net.

For the binary constraint, considering that $\mathbf{\Phi}$ should also satisfy the above orthogonal constraint, we introduce an auxiliary variable denoted by $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{M \times N}$ and define $\mathbf{\Phi} = \alpha\, BinarySign(\tilde{\mathbf{\Phi}})$, where α is actually a learnable scale factor parameter and $BinarySign(\cdot)$ is an element-wise operation defined below

$$BinarySign(z) = 1 \text{ if z} >= 0 \quad \text{or} \quad -1 \text{ if z} < 0. \tag{8}$$

Furthermore, in order to use back-propagation, we define the derivative of $BinarySign(\cdot)$ as a constant function, i.e. $BinarySign'(z) = 1$. Therefore, in practical implementation, the real learnable parameter is $\tilde{\mathbf{\Phi}}$, and we use $\alpha\, Binary Sign(\tilde{\mathbf{\Phi}})$ to replace $\mathbf{\Phi}$ in OPINE-Net.

Experiments demonstrate that the above schemes for constraint incorporation are very effective and efficient.

D. Network Parameters and Loss Function

In light of previous descriptions, Eq. (3) has been successfully mapped into our proposed OPINE-Net. Concretely, the learnable parameter set in OPINE-Net, denoted by $\mathbf{\Theta}$, includes the scale factor α and the auxiliary variable $\tilde{\mathbf{\Phi}}$ in the sampling subnet, the step size ρ, the parameters of the transforms $\mathcal{D}(\cdot), \mathcal{H}(\cdot), \widetilde{\mathcal{H}}(\cdot), \mathcal{G}(\cdot)$, and the shrinkage threshold δ in the recovery subnet. As such, $\mathbf{\Theta} = \{\alpha, \tilde{\mathbf{\Phi}}, \rho, \delta, \mathcal{D}(\cdot), \mathcal{H}(\cdot), \widetilde{\mathcal{H}}(\cdot), \mathcal{G}(\cdot)\}$. Note that all these parameters will be learned as neural network parameters and all phases in recovery subnet share the same parameters.

Given the training dataset $\{(\mathbf{x}_i)\}_{i=1}^{N_b}$, OPINE-Net first takes $\mathbf{x}_i$ as input and generates the reconstructed result, denoted by $\hat{\mathbf{x}}_i^{(N_p)}$ as output. Note that, the purpose is to reduce the discrepancy between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i^{(N_p)}$ (N_p denotes the total number of phases in recovery subnet) while satisfying the symmetry

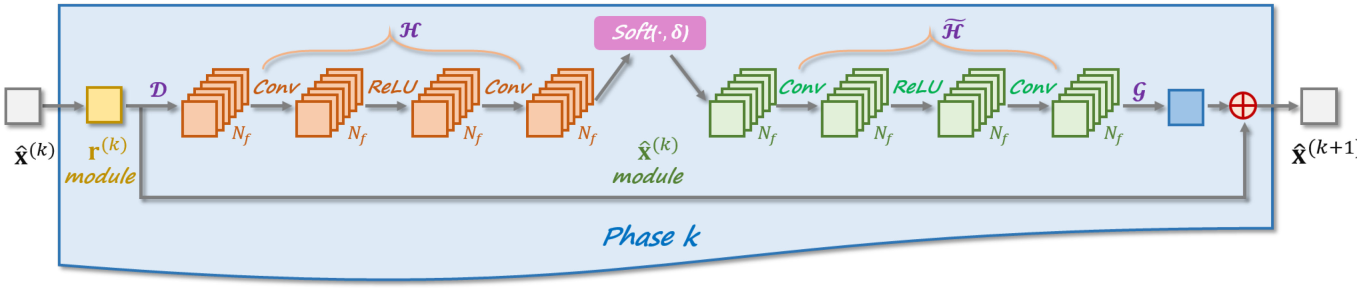

Fig. 4. Illustration of k^{th} phase in our recovery subnet (RS). Specifically, RS is composed of N_p phases, and each phase corresponds to one iteration in optimization. Here, N_f denotes the number of feature maps.

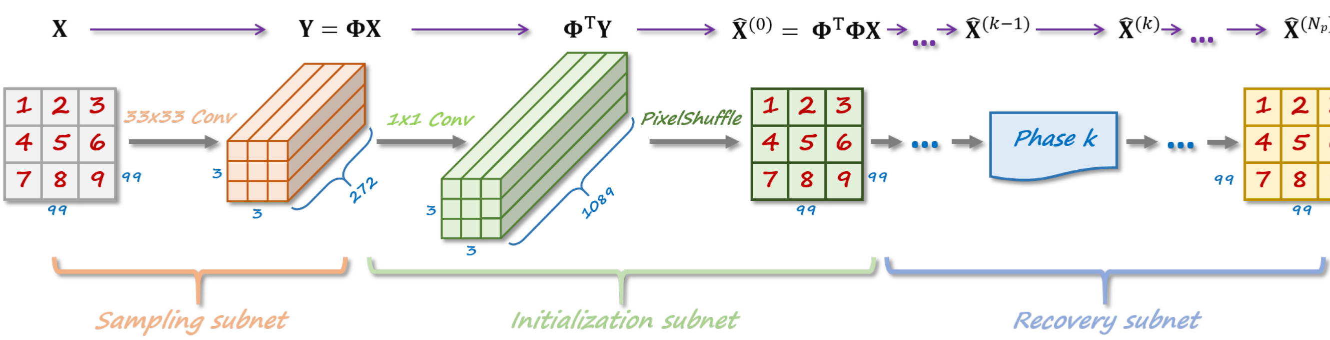

Fig. 5. Illustrations of our proposed OPINE-Net$^+$ framework, which allows image blocks to be sampled independently but recovered jointly, greatly suppressing blocking artifacts.

constraint $\widetilde{\mathcal{H}} \circ \mathcal{H} = \mathcal{I}$ and the orthogonal constraint and the binary constraint. Therefore, we design the end-to-end loss function for OPINE-Net as follows:

$$\mathcal{L}_{total}(\mathbf{\Theta}) = \mathcal{L}_{discrepancy} + \gamma\mathcal{L}_{symmetry} + \mu\mathcal{L}_{orth},$$
$$\text{with:}\begin{cases}\mathcal{L}_{discrepancy} = \frac{1}{N_b N}\sum_{i=1}^{N_b}\|\mathbf{x}_i^{(N_p)} - \mathbf{x}_i\|_F^2,\\ \mathcal{L}_{symmetry} = \frac{1}{N_z}\sum_{i=1}^{N_b}\sum_{k=1}^{N_p}\|\widetilde{\mathcal{H}}(\mathcal{H}(\mathbf{z}_i^{(k)})) - \mathbf{z}_i^{(k)}\|_F^2,\\ \mathcal{L}_{orth} = \frac{1}{M^2}\|\mathbf{\Phi}\mathbf{\Phi}^\top - \mathbf{I}\|_F^2.\end{cases} \tag{9}$$

where $\mathbf{z}_i^{(k)} = \mathcal{D}(\mathbf{r}_i^{(k)})$ and N_z denotes the number of elements in $\mathbf{z}_i^{(k)}$. $\|\cdot\|_F^2$ stands for the Frobenius norm of a matrix or a tensor, N_b denotes the total number of training blocks of size $\sqrt{N} \times \sqrt{N}$, γ, μ are the regularization parameters. In our experiments, γ and μ are set to 0.01.

E. Enhanced Multi-Block Version: OPINE-Net$^+$

From Fig. 1, one can clearly see that each image block is sampled and reconstructed independently, which will inevitably result in blocking artifacts and decrease image quality. In order to exploit the inter-block relationship and improve image quality, we furthermore design an enhanced multi-block version of OPINE-Net, dubbed OPINE-Net$^+$. As illustrated in Fig. 5, instead of one block of size 33×33, we adopt a larger image block of size 99×99 as input for training, denoted by $\mathbf{X}$. Obviously, $\mathbf{X}$ can be divided into nine blocks, i.e. $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_9\}$. Due to stride=33 in the convolution layer in the sampling subnet and the efficient convolutional design of $\mathbf{\Phi}$ and $\mathbf{\Phi}^\top$ in OPINE-Net, the proposed OPINE-Net$^+$ allows image blocks of size 33×33 to be sampled independently but reconstructed jointly.

IV. Experimental Results

For fair comparison, we use the same set of 91 images as in [35] for training. The training data $\{\mathbf{x}_i\}_{i=1}^{N_b}$ is first generated by randomly extracting the luminance component of 88,912 image blocks (each of size 33×33), i.e. $N_b = 88912$ and $N = 1089$ for OPINE-Net and 43,340 image blocks (each of size 99 $\times$ 99) for OPINE-Net$^+$, respectively. Then, for a given range of CS ratios $\{1\%, 4\%, 10\%, 25\%, 50\%\}$, we train the OPINE-Nets separately for adaptive sampling and recovery of image CS, obtaining the corresponding learned sampling matrices $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$. In practice, the training of OPINE-Net$^+$ is accelerated by fine-tuning OPINE-Net for one epoch. All the experiments are performed on a workstation with Intel Core i7-6820 CPU and GTX1080Ti GPU by PyTorch. Adam optimization [48] is used with a batch size of 64. [1] Training OPINE-Nets with phase number $N_p = 9$ in recovery subnet roughly takes 10 hours. For testing, we utilize three widely used benchmark datasets: Set11 [35], BSD68 [49] and Urban100 [50]. Note that we deal with color images in the transformed YCbCr space and conduct an independent operation for each channel. The CS recovered

[1]The sources codes and training models of OPINE-Net and OPINE-Net$^+$ will be made [Online]. Available: https://jianzhang.tech/projects/OPINENet

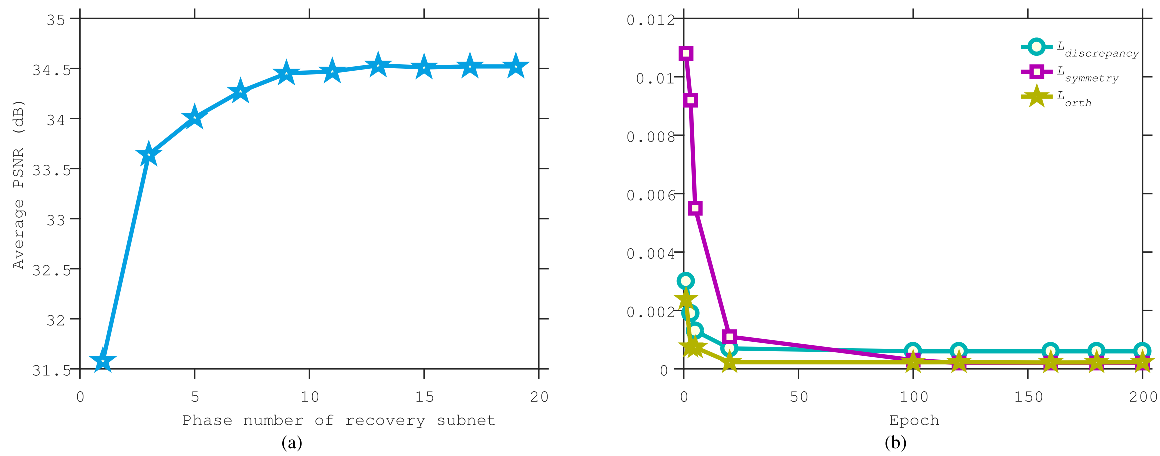

Fig. 6. (a) Average PSNR curves for Set11 by OPINE-Net with various phase numbers in the cases of CS ratio = 25%; (b) The progression curves of $\mathcal{L}_{discrepancy}$, $\mathcal{L}_{symmetry}$, $\mathcal{L}_{orth}$ achieved by OPINE-Net in training with various epoch numbers in the case of CS ratio = 25%.

results are evaluated with PSNR and SSIM [51] on Y channel (*i.e.*, luminance).

A. Study of Phase Number and Convergence

To determine a proper phase number N_p, we plot the average PSNR curves by OPINE-Net for Set11 with respect to different phase numbers in the cases of CS ratio = 25%, as shown in Fig. 6(a). One can observe that the PSNR curves increase as phase number N_p increases; however, the curves are almost flat when $N_p \geq 9$. Thus, considering the trade-off between computational complexity and recover performance, we set N_p to be 9 for our OPINE-Nets by default.

Fig. 6(b) further illustrates the progression of three types of losses, i.e. $\mathcal{L}_{discrepancy}$, $\mathcal{L}_{symmetry}$ and $\mathcal{L}_{orth}$ in Eq. (9) achieved by OPINE-Net with respect to epoch number in training in the case of CS = 25% and $N_p = 9$. Clearly, OPINE-Net converges very fast and all three losses decrease consistently. In particular, the losses $\mathcal{L}_{symmetry}$ and $\mathcal{L}_{orth}$ are eventually close to zero, indicating that the learned OPINE-Net satisfies the corresponding two constraints.

B. Ablation Studies and Discussions

By default, our propose OPINE-Net has three constraints, i.e. binary constraint (BC) of $\mathbf{\Phi}$, orthogonal constraint (OC) of $\mathbf{\Phi}$, and shared weights (SW) across phases in recovery subnet. In this subsection, we will investigate the performance effect of these three $\mathbf{\Phi}$. constraints and give some interesting findings about the learned Table I shows the ablation investigation on the effects of BC, OC, and SW. From Table I, we can observe that these three constraints do not impair the final performance of OPINE-Net. In fact, BC and OC play as the role of network regularization and always improve the performance a little. Note that BC makes the proposed OPINE-Net more hardware-friendly and greatly reduces the storage of $\mathbf{\Phi}$. SW also reduces the storage of the parameters in recovery subnet from N_p phases to one phase. We further visualize the convergence process of four typical combinations in Fig. 7. We use the performance of ISTA-Net$^+$ with fixed random Gaussian matrix as a reference. 'None' means the case without using the above three constraint. Clearly, all curves converge stably to the same result. The curves with fewer constraints usually have faster speed and OC does not affect the convergence speed.

Next, we give three interesting findings about the learned $\mathbf{\Phi}$ obtained by OPINE-Net with different constraint combinations. 1) Define the sampling matrix learned in the case of 'None' as

TABLE I
ABLATION INVESTIGATION OF NETWORK CONSTRAINTS: BINARY CONSTRAINT (BC) OF $\mathbf{\Phi}$, ORTHOGONAL CONSTRAINT (OC) OF $\mathbf{\Phi}$, AND SHARED WEIGHTS (SW) IN RECOVERY SUBNET. WE OBSERVE THE BEST PERFORMANCE (PSNR) ON SET11 IN THE CASE OF CS RATIO = 25%

	Different combinations of constraints in OPINE-Net						
BC	×	✓	×	×	✓	✓	✓
OC	×	×	✓	×	✓	×	✓
SW	×	×	×	✓	×	✓	✓
PSNR	34.31	34.39	34.43	34.41	34.47	34.43	34.44

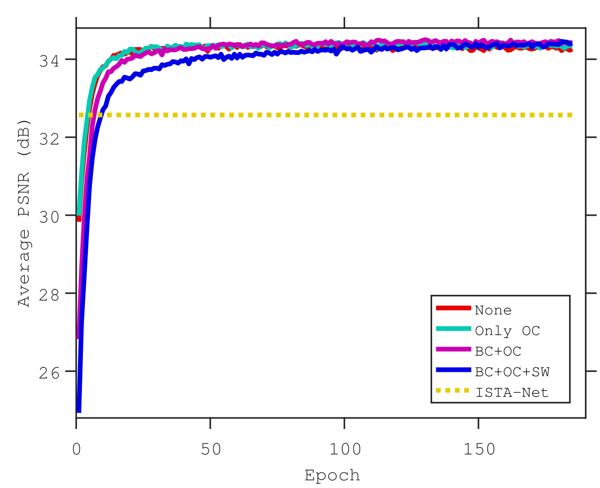

Fig. 7. Convergence analysis on four combinations of constraints. The curves for each combination are based on the PSNR on Set11 in the case of CS ratio = 25%.

TABLE II

AVERAGE PSNR/SSIM PERFORMANCE COMPARISONS WITH DIFFERENT CS RATIOS. THE PROPOSED OPINE-NET^{+} ACHIEVES THE BEST PERFORMANCE, WHICH IS LABELED IN BOLD

Dataset	CS ratio	ISTA-Net^{+} [44]	BCS [17]	CSNet [19]	AdapReconNet [20]	OPINE-Net	OPINE-Net^{+}
Set11	1%	17.42/0.4029	19.15/0.4410	19.87/0.4977	19.63/0.4848	19.87/0.5070	**20.15/0.5340**
	4%	21.32/0.6037	23.19/0.6633	23.93/0.7338	23.87/0.7279	25.04/0.7730	**25.69/0.7920**
	10%	26.64/0.8087	26.04/0.7971	27.59/0.8575	27.39/0.8521	29.33/0.8825	**29.81/0.8884**
	25%	32.59/0.9254	29.98/0.8932	31.70/0.9274	31.75/0.9257	34.44/0.9491	**34.86/0.9509**
	50%	38.11/0.9707	34.61/0.9435	37.19/0.9700	35.87/0.9625	39.88/0.9790	**40.17/0.9797**
Set68	1%	19.14/0.4158	21.24/0.4624	21.91/0.4958	21.50/0.4825	21.80/0.4972	**22.11/0.5140**
	4%	22.17/0.5486	23.94/0.6193	24.63/0.6564	24.30/0.6491	24.87/0.6709	**25.20/0.6825**
	10%	25.32/0.7022	26.07/0.7537	27.02/0.7864	26.72/0.7821	27.54/0.7966	**27.82/0.8045**
	25%	29.36/0.8525	29.18/0.8729	30.22/0.8918	30.10/0.8901	31.28/0.9034	**31.51/0.9061**
	50%	34.04/0.9424	33.18/0.9400	34.82/0.9590	33.60/0.9479	36.12/0.9646	**36.35/0.9660**
Urban100	1%	16.90/0.3846	18.97/0.4363	19.26/0.4632	19.14/0.4510	19.45/0.4808	**19.82/0.5006**
	4%	19.83/0.5377	21.55/0.5986	21.96/0.6430	21.92/0.6390	22.91/0.6930	**23.36/0.7114**
	10%	24.04/0.7378	23.58/0.7230	24.76/0.7899	24.55/0.7801	26.44/0.8298	**26.93/0.8397**
	25%	29.78/0.8954	26.75/0.8410	28.13/0.8827	28.21/0.8841	31.40/0.9270	**31.86/0.9308**
	50%	35.24/0.9614	30.65/0.9129	32.97/0.9503	31.88/0.9434	36.88/0.9729	**37.23/0.9741**

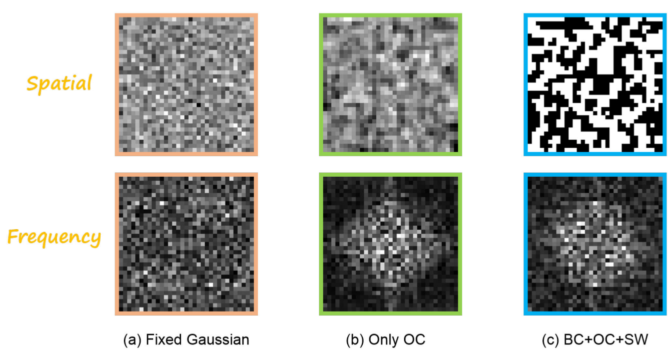

Fig. 8. Visualization of one row in the traditional fixed random Gaussian matrix (left), the learned matrix with 'OC' (middle) and the learned matrix with 'BC+OC+SW' (right).

$\mathbf{\Phi}_{No}$. We observe that, although there is no constraints, $\mathbf{\Phi}_{No}$ still satisfies the orthogonal constraint in the following form: $\mathbf{\Phi}_{No}\mathbf{\Phi}_{No}^{\top} = \eta\mathbf{I}$, where η usually varies at each training. This verifies the necessity of OC again. 2) Define the sampling matrix learned in the cases of 'OC' and 'BC+OC+SW' as $\mathbf{\Phi}_{OC}$ and $\mathbf{\Phi}_{All}$, respectively. Denote the fixed random Gaussian matrix, as $\mathbf{\Phi}_{FG}$. In the cases with same CS ratios, We reshape one row in $\mathbf{\Phi}_{FG}$, $\mathbf{\Phi}_{OC}$ and $\mathbf{\Phi}_{All}$ into [33 33] and visualize them in Fig. 8, along with their frequency. Obviously, the rows representing a filter in $\mathbf{\Phi}_{OC}$ and $\mathbf{\Phi}_{All}$ are more structured and are more like a low-pass filter, instead of being random as one in $\mathbf{\Phi}_{FG}$. 3) Define the normalized $\mathbf{\Phi}_{No}$ as $\tilde{\mathbf{\Phi}}_{No} = \mathbf{\Phi}_{No}/\sqrt{\eta}$. We plot the histograms of $\tilde{\mathbf{\Phi}}_{No}$, $\mathbf{\Phi}_{OC}$ and $\mathbf{\Phi}_{FG}$ in the cases of CS ratio $= 25\%$ and ratio $= 50\%$, respectively, as shown in Fig. 9. Surprisingly, in each case, these three matrices have the same distribution, which naturally leads to the following two inferences. The first one is the learned sampling matrix retains the same properties as the fixed random Gaussian matrix, such as RIP [52]. The second one is the feasibility of data-driven CS sampling matrix learning has been fully verified.

C. Comparison With State-of-the-Art Methods

We compare our proposed OPINE-Net with four recent representative deep network-based CS methods, namely ISTA-Net^{+} [44], BCS [17], CSNet [19] and AdapReconNet [20]. ISTA-Net^{+} does not involve sampling matrix learning, but generates state-of-the-art CS recovery results using fixed random Gaussian sampling matrix. The other four competing methods are able to learn adaptive sampling and recovery for image CS.

Table II clearly shows that our proposed OPINE-Net and OPINE-Net^{+} outperform all the other competing methods by a large margin across all the CS ratios. Note that ISTA-Net^{+} can be regarded as a special case of our proposed OPINE-Net when the sampling matrix is fixed. OPINE-Nets achieved more than 2 dB PSNR gains on average than ISTA-Net^{+}, which fully illustrates the necessity of adaptive sampling. Compared with the remaining four methods, the performance improvement of OPINE-Net mainly comes from the network structure inspired by optimization. Accordingly, the superior performance by OPINE-Net verifies the effectiveness of designing optimization-inspired deep network for joint learning of sampling and recovery. Furthermore, the enhanced version OPINE-Net^{+} achieved about 0.4 dB PSNR gain on average than OPINE-Net.

In Fig. 10, we show the reconstructions of all six methods of two images when the CS ratio is 10% and 25% respectively. The proposed OPINE-Net is able to recovery more details and sharper edges than other competing methods, and OPINE-Net^{+} achieves better results than OPINE-Net by further reducing blocking artifacts. More visual comparisons of OPINE-Net and OPINE-Net^{+} in the cases of CS ratio $= 4\%$ and ratio $= 10\%$ are shown in Fig. 11, which clearly verifies the superiority of OPINE-Net^{+}.

D. Study of Model Size and Computational Time

Table III provides a comparison of model size and computational time for various methods in the case of CS ratio $=$ 50%. Since BCS exploits all fully-connected layers, it has the most parameters and the largest model size. Compared with the other three CNN-based methods, our OPINE-Net reduces the parameters by half due to that no additional parameters are introduced in the initialization subnet. Remember that the learned $\mathbf{\Phi}$ by OPINE-Net is binary. If we use one bit instead of 4 bytes to represent one element in the $\mathbf{\Phi}$, then the model

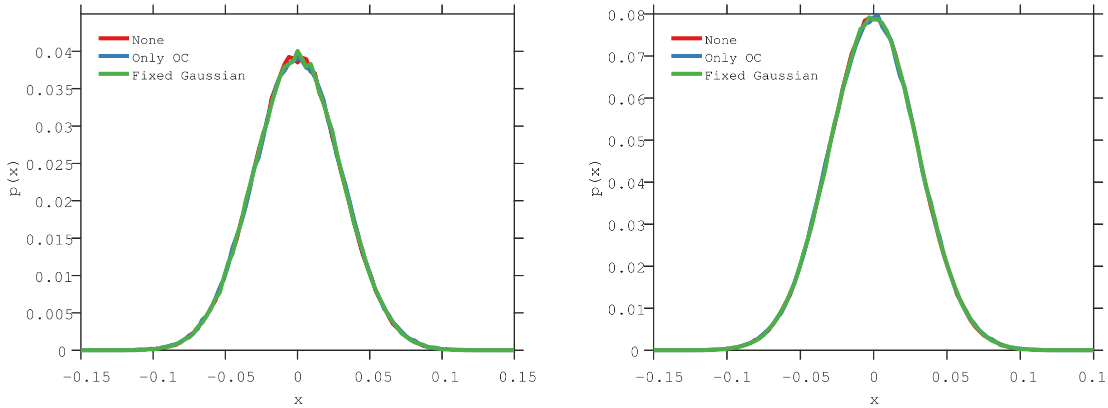

Fig. 9. Visualization of histograms of $\tilde{\mathbf{\Phi}}_{No}$, $\mathbf{\Phi}_{OC}$ and $\mathbf{\Phi}_{FG}$ in the cases of CS ratio $= 25\%$ and ratio $= 50\%$. Clearly, these three matrices have the same distribution.

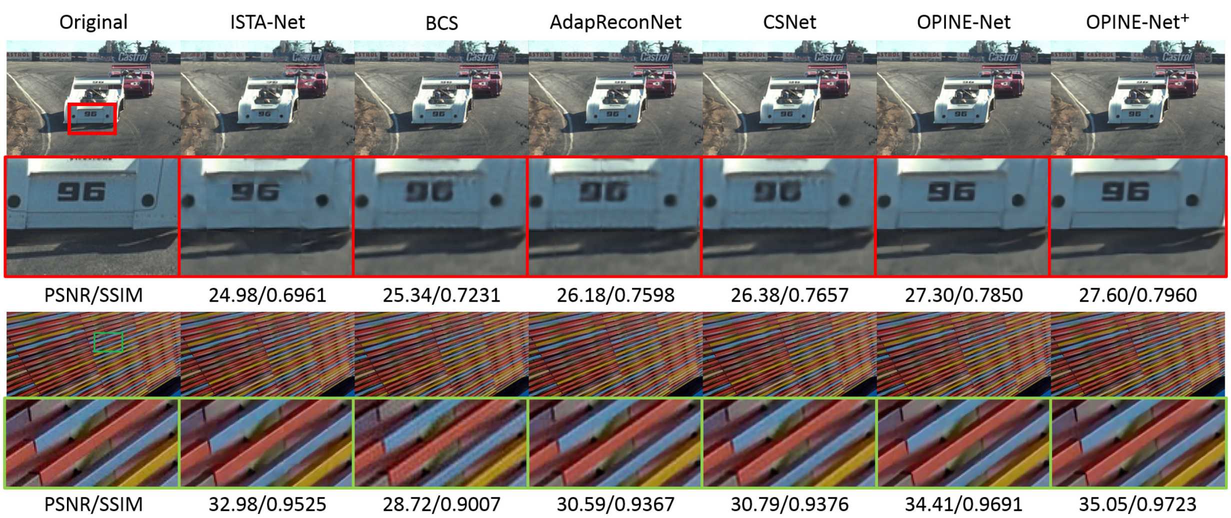

Fig. 10. Visual comparison of all the competing CS methods. The proposed OPINE-Nets are able to recovery more details and sharper edges than other competing methods.

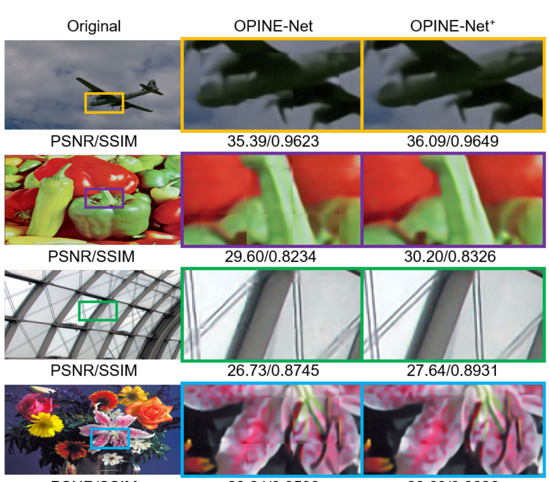

Fig. 11. Visual comparison of OPINE-Net and OPINE-Net$^+$. The proposed OPINE-Net$^+$ achieves better results than OPINE-Net by further reducing blocking artifacts.

TABLE III
COMPARISON OF COMPUTATIONAL TIME AND MODEL SIZE

	BCS	AdapReconNet	CSNet	OPINE-Net
#Para	7.76M	1.15M	1.17M	0.62M
Size	31.05MB	4.62MB	4.67MB	2.48MB
Time	0.0018s	0.0027s	0.0007s	0.0101s

size of OPINE-Net can be further reduced to 0.31 MB from 2.48 MB. The last row records the average running time on a 512×512 image with GPU. Note that the computational time of OPINE-Net is less than 15 millisecond (ms), which leads to more than 60 frames-per-second (FPS).

V. Conclusion and Future Work

Inspired by traditional optimization, we propose a novel framework to design a structured deep network for adaptive sampling and recovery of image compressive sensing (CS), which is dubbed OPINE-Net, as well as its enhanced version OPINE-Net$^+$. With the incorporated orthogonal and binary constraints of sampling matrix. the proposed OPINE-Nets possess well-defined explicability, and make full use of the merits of both optimization-based and network-based CS methods. All the parameters in OPINE-Nets are discriminately learned end-to-end. Some interesting findings of learned $\mathbf{\Phi}$ are presented. Compared with existing network-based methods, the proposed hardware-friendly OPINE-Nets reduce the number of learnable parameters by half and achieves about $8\times$ model compression rate improvement. What's more, OPINE-Nets greatly improve upon the results of state-of-the-art CS methods, while maintaining a real-time speed. Since the developed framework is quite general, one direction of interest is to extend OPINE-Net to video application or to other scenarios with joint sampling and recovery.

REFERENCES

[1] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[2] M. F. Duarte *et al.*, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[3] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, "CS-MUVI: Video compressive sensing for spatial-multiplexing cameras," in *Proc. IEEE Int. Conf. Comput. Photography*, 2012, pp. 1–10.

[4] A. Liutkus *et al.*, "Imaging with nature: Compressive imaging using a multiply scattering medium," *Scientific Reports*, vol. 4, 2014, Art. no. 5552.

[5] F. Rousset, N. Ducros, A. Farina, G. Valentini, C. DAndrea, and F. Peyrin, "Adaptive basis scan by wavelet prediction for single-pixel imaging," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 36–46, Mar. 2017.

[6] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid mr imaging," *Magn. Reson. Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[7] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ECG via block sparse Bayesian learning," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 2, pp. 300–309, Feb. 2013.

[8] S. K. Sharma, E. Lagunas, S. Chatzinotas, and B. Ottersten, "Application of compressive sensing in cognitive radio communications: A survey," *IEEE Commun. Surv. Tut.*, vol. 18, no. 3, pp. 1838–1860, Jul.–Sep. 2016.

[9] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[10] J. Zhang, C. Zhao, D. Zhao, and W. Gao, "Image compressive sensing recovery using adaptively learned sparsifying basis via l0 minimization," *Signal Process.*, vol. 103, pp. 114–126, 2014.

[11] C. Zhao, S. Ma, J. Zhang, R. Xiong, and W. Gao, "Video compressive sensing reconstruction via reweighted residual sparsity," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 27, no. 6, pp. 1182–1195, Jun. 2016.

[12] C. Zhao, J. Zhang, S. Ma, X. Fan, Y. Zhang, and W. Gao, "Reducing image compression artifacts by structural sparse representation and quantization constraint prior," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 27, no. 10, pp. 2057–2071, Oct. 2017.

[13] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009.

[14] H. Bai, G. Li, S. Li, Q. Li, Q. Jiang, and L. Chang, "Alternating optimization of sensing matrix and sparsifying dictionary for compressed sensing," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1581–1594, Mar. 2015.

[15] C. Lu, H. Li, and Z. Lin, "Optimized projections for compressed sensing via direct mutual coherence minimization," *Signal Process.*, vol. 151, pp. 45–55, 2018.

[16] T. Hong and Z. Zhu, "Online learning sensing matrix and sparsifying dictionary simultaneously for compressive sensing," *Signal Process.*, vol. 153, pp. 188–196, 2018.

[17] A. Adler, D. Boublil, and M. Zibulevsky, "Block-based compressed sensing of images via deep learning," in *Proc. Int. Workshop Multimedia Signal Process.*, 2017, pp. 1–6.

[18] J. Du, X. Xie, C. Wang, G. Shi, X. Xu, and Y. Wang, "Fully convolutional measurement network for compressive sensing image reconstruction," *Neurocomputing*, vol. 328, pp. 105–112, 2019.

[19] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2017, pp. 877–882.

[20] S. Lohit, K. Kulkarni, R. Kerviche, P. Turaga, and A. Ashok, "Convolutional neural networks for noniterative reconstruction of compressively sensed images," *IEEE Trans. Comput. Imag.*, vol. 4, no. 3, pp. 326–340, Sep. 2018.

[21] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proc. IEEE Int. Conf. Image Process.*, 2009, pp. 3021–3024.

[22] C. Li, W. Yin, H. Jiang, and Y. Zhang, "An efficient augmented Lagrangian method with applications to total variation minimization," *Comput. Optim. Appl.*, vol. 56, no. 3, pp. 507–530, 2013.

[23] Y. Kim, M. S. Nadar, and A. Bilgin, "Compressed sensing using a Gaussian scale mixtures model in wavelet domain," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 3365–3368.

[24] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3488–3497, Sep. 2009.

[25] J. Zhang, D. Zhao, C. Zhao, R. Xiong, S. Ma, and W. Gao, "Image compressive sensing recovery via collaborative sparsity," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 3, pp. 380–391, Sep. 2012.

[26] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3618–3632, Aug. 2014.

[27] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[28] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[29] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[30] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2808–2817.

[31] J. H. R. Chang, C.-L. Li, B. Póczos, B. V. K. V. Kumar, and A. C. Sankaranarayanan, "One network to solve them all — solving linear inverse problems using deep projection models," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 5889–5898.

[32] C. Zhao, J. Zhang, R. Wang, and W. Gao, "CREAM:CNN-REgularized ADMM framework for compressive-sensed image reconstruction," *IEEE Access*, vol. 6, pp. 76 838–76 853, 2018.

[33] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Proc. IEEE Annu. Allerton Conf. Commun., Control, Comput.*, 2015, pp. 1336–1343.

[34] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, "Deep fully-connected networks for video compressive sensing," *Digit. Signal Process.*, vol. 72, pp. 9–18, 2018.

[35] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 449–458.

[36] A. Mousavi and R. G. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2272–2276.

[37] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[38] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 2774–2781.

[39] Y. Chen, W. Yu, and T. Pock, "On learning optimized reaction diffusion processes for effective image restoration," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5261–5269.

[40] U. S. Kamilov and H. Mansour, "Learning optimal nonlinearities for iterative thresholding algorithms," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 747–751, May 2016.

[41] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 370–378.
[42] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 10–18.
[43] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1772–1783.
[44] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 1828–1837.
[45] A. Mousavi, G. Dasarathy, and R. G. Baraniuk, "Deepcodec: Adaptive sensing and recovery via deep convolutional neural networks," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2017, p. 744.
[46] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Scalable convolutional neural network for image compressed sensing," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 12 290–12 299.
[47] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1874–1883.
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[49] 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 *Proc. IEEE Int. Conf. Comput. Vision*, 2001, pp. 416–423.
[50] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5197–5206.
[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[52] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9–10, pp. 589–592, 2008.

Jian Zhang (Member, IEEE) received the B.S. degree from the Department of Mathematics, Harbin Institute of Technology (HIT), Harbin, China, in 2007, and the M.Eng. and Ph.D. degrees from the School of Computer Science and Technology, HIT, in 2009 and 2014, respectively.

He is currently an Assistant Professor with the School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China. His research interests include image/video compression and restoration, optimization, and deep learning.

Chen Zhao received the B.S. degree from the School of Software Engineering, Sichuan University, Chengdu, China, in 2010 and the Ph.D. degree from the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, in 2016.

She is currently a Postdoctoral Fellow with the King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. Her research interests include action recognition and image/video processing.

Wen Gao (Fellow, IEEE) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He is currently a Professor of computer science with Peking University, Beijing, China. He has authored extensively including five books and more than 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics.

Dr. Gao served or serves on the editorial board for several journals, such as IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON IMAGE PROCESSING, *EURASIP Journal of Image Communications*, *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ISCAS, ICME, and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations. He is a member of Chinese Academy of Engineering, and a Fellow of ACM.