WTDUN: Wavelet Tree-Structured Sampling and Deep Unfolding Network for Image Compressed Sensing

KAI HAN, Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Institute of Artificial Intelligence, School of Information Science and Technology, Beijing University of Technology, Beijing, China
JIN WANG, School of Computer Science, Beijing University of Technology, Beijing, China
YUNHUI SHI, Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Institute of Artificial Intelligence, School of Information Science and Technology, Beijing University of Technology, Beijing, China
HANQIN CAI, The Department of Statistics and Data Science, University of Central Florida, Orlando, FL, USA and the Department of Computer Science, University of Central Florida, Orlando, FL, USA
NAM LING, The Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA
BAOCAI YIN, Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Institute of Artificial Intelligence, School of Information Science and Technology, Beijing University of Technology, Beijing, China

Deep unfolding networks have gained increasing attention in the field of compressed sensing (CS) owing to their theoretical interpretability and superior reconstruction performance. However, most existing deep unfolding methods often face the following issues: (1) they learn directly from single-channel images, leading to a simple feature representation that does not fully capture complex features; and (2) they treat various image components uniformly, ignoring the characteristics of different components. To address these issues, we propose a novel wavelet-domain deep unfolding framework named WTDUN, which operates directly on the multi-scale wavelet sub-bands. Our method utilizes the intrinsic sparsity and multi-scale structure of wavelet coefficients to achieve a tree-structured sampling and reconstruction, effectively capturing and highlighting the most important features within images. Specifically, the design of tree-structured reconstruction aims to

This work was partially supported by the National Key R&D Program of China (No. 2021ZD0111902), the National Natural Science Foundation of China (Nos. 62272016, 62372018, U21B2038), and the National Science Foundation through grant DMS-2304489.

Authors' Contact Information: Kai Han, Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Institute of Artificial Intelligence, School of Information Science and Technology, Beijing University of Technology, Beijing, China; e-mail: 3479926203@qq.com; Jin Wang, School of Computer Science, Beijing University of Technology, Beijing, China; e-mail: ijinwang@bjut.edu.cn; Yunhui Shi (corresponding author), Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Institute of Artificial Intelligence, School of Information Science and Technology, Beijing University of Technology, Beijing, China; e-mail: syhzm@bjut.edu.cn; Hanqin Cai, the Department of Statistics and Data Science and the Department of Computer Science, University of Central Florida, Orlando, FL, USA; e-mail: hqcai@ucf.edu; Nam Ling, the Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA; e-mail: nling@scu.edu; Baocai Yin, Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Institute of Artificial Intelligence, School of Information Science and Technology, Beijing University of Technology, Beijing, China; e-mail: ybc@bjut.edu.cn.

ACM 1551-6865/2024/12-ART33
https://doi.org/10.1145/3701731

capture the inter-dependencies among the multi-scale sub-bands, enabling the identification of both fine and coarse features, which can lead to a marked improvement in reconstruction quality. Furthermore, a wavelet domain adaptive sampling method is proposed to greatly improve the sampling capability, which is realized by assigning measurements to each wavelet sub-band based on its importance. Unlike pure deep learning methods that treat all components uniformly, our method introduces a targeted focus on important sub-bands, considering their energy and sparsity. This targeted strategy lets us capture key information more efficiently while discarding less important information, resulting in a more effective and detailed reconstruction. Extensive experimental results on various datasets validate the superior performance of our proposed method.

CCS Concepts: • **Computing methodologies** → **Image processing**; **Reconstruction**;

Additional Key Words and Phrases: Compressed sensing, deep unfolding, wavelet tree

ACM Reference format:
Kai Han, Jin Wang, Yunhui Shi, Hanqin Cai, Nam Ling, and Baocai Yin. 2024. WTDUN: Wavelet Tree-Structured Sampling and Deep Unfolding Network for Image Compressed Sensing. *ACM Trans. Multimedia Comput. Commun. Appl.* 21, 1, Article 33 (December 2024), 22 pages.
https://doi.org/10.1145/3701731

1 Introduction

Compressed sensing (CS) is a promising technique for signal acquisition and reconstruction [2, 10]. The target signal is first simultaneously sampled and compressed with linear random transformations. Then, the original signal can be reconstructed exactly from far fewer measurements than that required by the Nyquist sampling rate. Mathematically, a random linear measurement $\boldsymbol{y} \in \mathbb{R}^M$ can be formulated as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^N$ is the original signal and $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is the sampling matrix with $M \ll N$. $r = M/N$ is the sampling rate (or CS ratio). To obtain a reliable reconstruction, traditional CS reconstruction methods commonly solve an optimization problem as follows:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\Psi\boldsymbol{x}\|_1, \tag{1}$$

where $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$ denotes the data-fidelity term and $\Psi\boldsymbol{x}$ is the transform coefficients of $\boldsymbol{x}$ with respect to some transform Ψ, the sparsity of $\Psi\boldsymbol{x}$ is encouraged by the ℓ_1-norm or other sparsity promoting norm with the regularization parameter λ. Currently, CS has been widely applied in various fields such as image reconstruction [8], **magnetic resonance imaging (MRI)** [51], snapshot-compressed imaging [23], and communication.

Over the past decades, numerous efforts have been dedicated to image CS. There are two primary challenges in the field of CS: signal sampling and reconstruction. The construction of the sampling matrix $\boldsymbol{A}$ plays a crucial role in capturing the intrinsic structure of the original signal $\boldsymbol{x}$. Traditionally, $\boldsymbol{A}$ is often chosen as a random Gaussian, Poisson, or Toeplitz matrix to uniformly sample each component of the image. This can ensure accurate signal reconstruction with theoretical guarantees. However, the CS ratio is typically much lower than the Nyquist sampling rate, which usually results in artifacts. This phenomenon may be caused by the aliasing between high- and low-frequency information, which hinders detail and texture recovery during reconstruction. Therefore, it is important to treat different components distinctly to acquire more essential information during the sampling process.

Another key to achieving a superior reconstruction of $\boldsymbol{x}$ lies in the reconstruction algorithm. This has been the focus of numerous studies aimed at improving the quality of reconstructions in the past decades. Traditional optimization-based reconstruction methods often integrate image prior

information [21, 42, 54, 56], such as sparsity in a specific transform domain, as a regularization term within the objective function. Consequently, these methods are capable of solving the sparse regularized optimization problem with precision. Although achieving satisfactory performance, these methods still have room for improvement due to their limited adaptability to diverse signals. Traditional CS methods usually suffer from challenges such as high computational complexity and the need for parameter tuning.

Recently, profiting from the powerful learning and fast computing abilities of deep networks, deep learning-based CS methods have garnered considerable interest. They can directly learn the inverse mapping from the CS measurement domain to the original signal domain [31, 39]. Compared to traditional CS methods, deep learning-based CS methods dramatically reduce time complexity and greatly improve reconstruction performance. However, most existing deep learning-based methods are handcrafted designed and trained as a black box, with limited insights in the CS domain. Thus, deep unfolding-based methods [7, 32, 37, 47, 50, 55] are proposed, which are established by unfolding the first K iterations of a referenced optimization algorithm and transforming all the steps of each iteration into learnable deep network components. Deep unfolding methods can simultaneously maintain accuracy, interpretability, and speed by combining the advantages of traditional optimization-based algorithms and deep networks, such as ISTA-Net [50], ADMM-CSNet [44], and MADUN [33].

To address the above-mentioned issues, in this article, we propose a novel wavelet domain-based deep unfolding framework. This framework introduces wavelet tree-structured reconstruction and **wavelet domain adaptive sampling (WAS)**, which can treat diverse components differently. By utilizing the rich information within the wavelet domain, our method is adapted to capturing intricate image features with greater efficiency. This ensures that the final image is not only visually pleasing but also fidelity to the structure of the original images. Specifically, we design a WAS method that allocates CS measurements based on the difference of each sub-band, thereby customizing the sampling process to the varied features of the image. For CS reconstruction, we design a tree-structured prior to guiding our unfolding network, which can effectively exploit the inter-dependencies at different scales. It can capture key features across multiple scales, allowing the incorporation of information from one scale to enhance details at another scale. Since block partition breaks the global correlation of the whole image, a deblocking module is performed on the whole image in every reconstruction stage. This module serves to eliminate blocking artifacts and exploit contextual information between adjacent phases. Our extensive experimental results on diverse datasets demonstrate that our method can achieve better performance compared with other state-of-the-art CS methods. The main contributions of this article are summarized as follows:

- —A novel wavelet-domain deep unfolding framework for image CS is proposed, which achieves simultaneous WAS and tree-structured reconstruction by fully exploiting the structural sparsity of multi-scale wavelet coefficients.
- —A **wavelet tree-structured prior (WTP)** guided unfolding network is designed, which effectively exploits the inter-dependencies among wavelet sub-bands at different scales. Through an iterative optimization process guided by WTP, accurate recovery of finer textures and sharper edges can be achieved.
- —A WAS method that significantly improves sampling capabilities is developed. Unlike conventional methods that treat all sub-bands uniformly, our approach allows for a targeted focus on significant sub-bands, considering their energy and sparsity, which can capture relevant information effectively while discarding less important details.
- —Extensive experimental results on various datasets validate the supreme performance of our proposed scheme.

Organization. The subsequent sections of this article are organized as follows. Section 2 introduces related works of CS. Section 3 presents the network structure of our approach. Section 4 evaluates the performance of our methods and compares it to other state-of-the-art CS methods. At last, we conclude this article in Section 5.

2 Related Works

The previous CS methods can be divided into two categories: traditional optimization-based methods and deep learning-based methods. In this section, we give a brief review.

2.1 Traditional Optimization-Based Methods

2.1.1 Sampling. For sampling methods, they can be divided into *uniform sampling* and *adaptive sampling*. Uniform sampling methods typically use a random Gaussian, Poisson, or Toeplitz matrix to sample each image block and every component in blocks equally. On the contrary, adaptive sampling methods consider the differences in image structure. Recently, there has been extensive research on adaptive sampling methods, focusing on their capacity to enhance the overall performance of algorithms. Wang et al. [38] use variance as a decision condition to split image blocks into two categories for sampling and reconstruction. Zhu et al. [58] adaptively assign sampling measurements based on different statistical features in image blocks, which significantly improved the reconstruction quality. However, the above studies treat each component of image blocks equally, which may hinder the acquisition of high-quality reconstructed images.

2.1.2 Reconstruction. In the field of CS reconstruction, most researchers have focused on two directions: image domain-based methods and wavelet domain-based methods.

Methods categorized in the former group directly sample the original image blocks, which mainly include convex optimization algorithm [6], greedy matching pursuit algorithm [24], orthogonal matching pursuit algorithm [28], Bayesian algorithm [19], and gradient descent algorithm [49]. Afterward, to further improve the reconstruction performance, some elaborate priors have been applied to CS reconstruction, such as denoising prior, total variation prior, and group sparsity prior. Specifically, Li et al. [21] develop a total variation regularization constraint to improve the local smoothness. Zhang et al. [53] present a group sparse representation to enhance image sparsity and exploit non-local self-similarity for image recovery.

In the latter class of methods, the original image is first decomposed through a multi-layer wavelet transform. Subsequently, the resulting wavelet sub-bands are sampled in blocks. Finally, an inverse transform is applied to obtain the recovered image. The wavelet coefficients of images are organized in a quadtree structure. Most values of the wavelet coefficients cluster around zero, while only a small part shows significant amplitude. Such structure leads to strong sparsity which can greatly improve the imaging speed [4] and highly enhance reconstruction quality. For example, MS-BCS-SPL [15] is a CS algorithm to reconstruct multi-scale measurements in the wavelet domain, which can recover more details and sharper edges.

Although traditional optimization methods have strong theoretical benefits, they still suffer from some issues such as excessive complexity, time-consuming, and poor real-time performance, which limit the broad application in practice.

2.2 Deep Learning-Based Methods

2.2.1 Sampling. In contrast to traditional CS methods, some deep learning methods still use random matrix [55] to obtain CS measurements, while others adopt convolutional [31] or fully connected [12, 22, 40] layer to model sampling matrix. Recently, Zhou et al. [57] propose a multi-channel model named BCS-Net using a channel-specific sampling network to realize adaptive CS

ratio allocation. Fan et al. [13] utilize convolutional layers to extract high-dimensional features, which are subsequently compressed through a series of convolutional layers to acquire more sampling information. Although satisfactory reconstruction results can be achieved, there is still much room to improve the performance due to its lack of adaptability to the vast diversity of signals.

2.2.2 Reconstruction. As for deep learning-based reconstruction methods, they can be roughly categorized into deep network methods and deep unfolding methods.

Deep Network Methods. For classical deep network methods, Mousavi et al. [26] first exploit a stack denoising auto-encoder to recover images from CS measurements. Afterward, Lohit et al. [22] design a **convolutional neural network (CNN)** based model called ReconNet to improve the image reconstruction quality. Shi et al. [31] propose a residual CNN-based framework dubbed CSNet to jointly train the sampling matrix and the reconstruction network. Huang et al. [17, 18] design a reconstruction network to acquire high-quality reconstruction by exploiting the Gaussian Scale Mixture prior. Goodfellow et al. [16] use generative adversarial networks for CS reconstruction. Fang et al. [14] successfully improve the reconstruction quality through the analysis of image features in the frequency domain. DDS-Net [20], a novel dynamic network, efficiently customizes denoising processes for various image noise levels. Moreover, the application of transformer in image reconstruction has also yielded significant achievements [29, 46].

Deep Unfolding Methods. Deep unfolding methods realize mathematical optimization iterations by designing deep networks. Therefore, deep unfolding methods combine the advantages of traditional iterative optimization methods and classical deep network methods. For example, Zhang et al. [50] extend the popular algorithm ISTA to ISTA-Net for optimizing a ℓ_1-norm CS model. Afterward, Zhang et al. [55] propose a CS model called AMP-Net by unfolding the denoising process of the approximate message-passing algorithm, incorporating a deblocking module in each stage to remove blocking artifacts. Metzler et al. [25] design LDIT and LDAMP from the denoise-based iterative thresholding algorithm. Yang et al. [44] propose a deep learning framework named ADMM-CSNet by the redesign of CS-MRI.

Compared with traditional iterative optimization-based methods, deep learning-based methods greatly reduce computational complexity and time complexity. Although the above deep learning-based methods have achieved excellent success in the field of CS, there are still some issues that texture details are not recovered well enough and no sparsity prior knowledge is combined.

To address the above-mentioned issues, we propose a novel wavelet domain-based deep unfolding framework to simultaneously enable WAS and tree-structured reconstruction. In the sampling process, we present a WAS method to allocate CS measurements based on the different importance of multi-scale sub-bands. The reconstruction process adopts a tree-structured prior guided model that takes advantage of the structure sparsity in multi-scale sub-bands. This model enables the recovery of more structural and finer detail components.

3 Proposed Method

3.1 Overall Architecture

As shown in Figure 1, WTDUN is an end-to-end deep unfolding model that integrates an optimization algorithm with a deep network, and it comprises a sampling module, an initialization module, and K reconstruction stages. Traditional optimization algorithm [1, 11] solves the CS reconstruction problem by iterating between the following two update steps:

$$\boldsymbol{z}^{k-1} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{k-1}, \tag{2}$$

$$\boldsymbol{x}^{k} = \zeta_k(\boldsymbol{A}^T\boldsymbol{z}^{k-1} + \boldsymbol{x}^{k-1}), \tag{3}$$

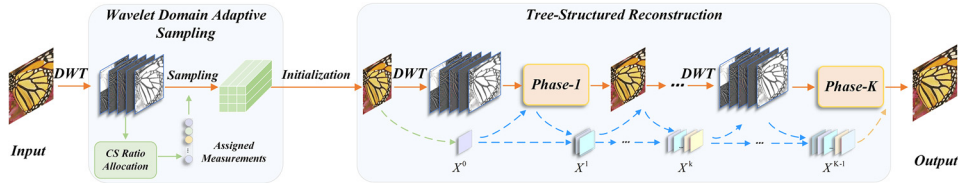

Fig. 1. Illustration of WTDUN, which consists of a sampling module, an initial module, and K reconstruction modules. WTDUN, Wavelet Tree-Structured Sampling and Deep Unfolding Network.

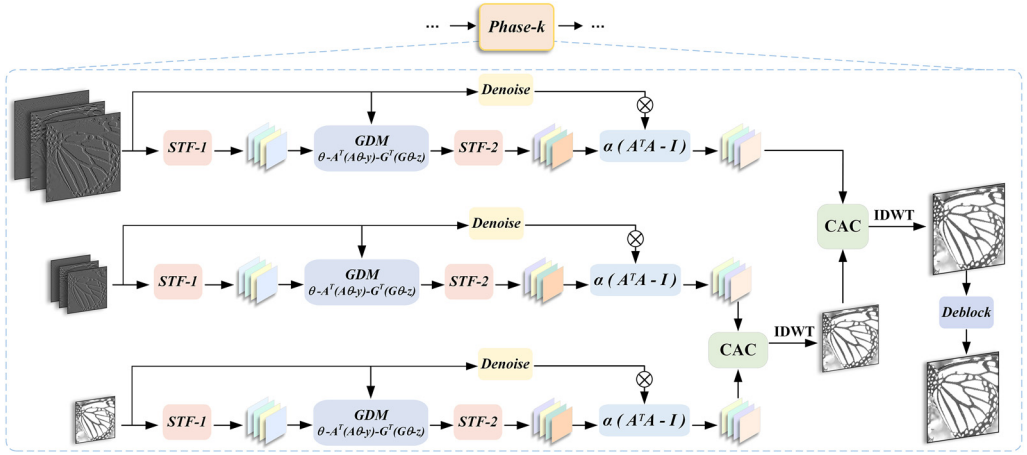

Fig. 2. The detailed design of one single phase in WTDUN. TheGDM represents Equation (15). STF-1 and STF-2 denote Equations (12) and (17), respectively. GDM, gradient descent module; STF, soft threshold function.

where k is the number of iterations, $\boldsymbol{A}^T$ is the transpose of sampling matrix $\boldsymbol{A}$, and $\zeta(\cdot)$ is an ideal image denoiser with a denoising perspective [55]:

$$\boldsymbol{A}^T\boldsymbol{z}^{k-1} + \boldsymbol{x}^{k-1} = \overline{\boldsymbol{x}} + \boldsymbol{e}, \tag{4}$$

where $\boldsymbol{e} = (\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I})(\overline{\boldsymbol{x}} - \boldsymbol{x}^{k-1})$ denotes a noise term. The clean signal can be obtained by calculating:

$$\boldsymbol{x}^k = \boldsymbol{A}^T\boldsymbol{z}^{k-1} + \boldsymbol{x}^{k-1} - (\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I})(\overline{\boldsymbol{x}} - \boldsymbol{x}^{k-1}). \tag{5}$$

By combining with the WTP, we transform Equation (5) into the kth reconstruction stage of our network, as shown in Figure 2. Considering the different importance of image high- and low-frequency components, a WAS module is constructed to enable adaptive sampling as in Figure 3. Then in subsections, each module will be elaborated separately.

3.2 WAS

3.2.1 CS Measurement Ratio Allocation. The CS measurement ratio is usually significantly lower than the Nyquist sampling rate, leading to undesirable artifacts. This phenomenon can be attributed to aliasing between high- and low-frequency information, which may prevent the accurate recovery of image details and textures during reconstruction. To mitigate the problem, we employ a Haar wavelet to separate the image into different frequency components. Subsequently, CS measurements for each component are assigned based on their importance.

$$\boldsymbol{\theta}_s = \Psi\boldsymbol{X}, \tag{6}$$

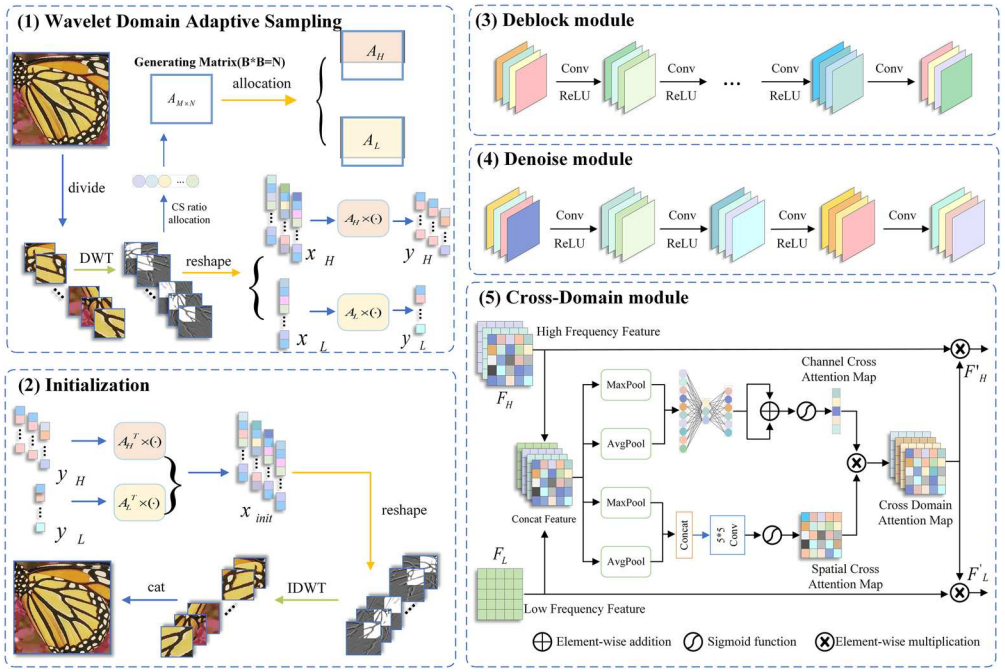

Fig. 3. Detailed design of each module. (1) Indicates the sampling process; (2) is the initial reconstruction process; (3) is the deblock module, which is composed of six convolution layers, with a ReLU activation layer between the adjacent convolution layers; (4) is the denoise module, which is composed of four convolution layers, with a ReLU activation layer between the adjacent convolution layers; (5) is the cross-sub-band fusion module (CAC). CAC, cross-domain attention.

where $\Psi = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_n]$ is the transform basis, $s \in \{LL_l, HL_i, LH_i, HH_i\}$, $1 \leq i \leq l$, i means the wavelet decomposition level, $\boldsymbol{\theta}$ means each wavelet sub-band of image $\boldsymbol{X}$. After multi-level wavelet decomposition, we obtain a sequence of sub-bands. Based on the property that the lowest frequency in the wavelet domain can carry the image energy and the high-frequency sub-bands can express the image textures and details, we design an algorithm to realize adaptive CS ratio allocation. We use the mean to evaluate sub-band energy and the SD to measure sub-band sparsity.

In the allocation algorithm, we initially calculate the SD σ and mean μ of absolute values $|\boldsymbol{\theta}|$ for each wavelet sub-band. Subsequently, we introduce η as a control parameter to adjust the contribution of σ and μ when determining the allocation weight. The allocation weight W is obtained by summing up the products of σ, μ, and their respective control parameters. Next, we compute the percentages P_H and P_{LL} representing high-frequency (W_H) and low-frequency (W_{LL}) contributions, respectively. By utilizing the percentages P_H and P_{LL}, we can derive the corresponding quantities of CS measurements. Subsequently, an upper bound Θ is established to prevent sample overflow. In cases where the number of low-frequency measurements surpasses this threshold Θ, it is adjusted to match the number of low-frequency CS measurements. Any additional CS measurements are then allocated to other wavelet sub-bands. Given that high-frequency sub-bands in images exhibit a tree structure following multi-level wavelet decomposition, these sub-bands at different scales demonstrate similar structures. Rather than considering sparsity when assigning measurements to high frequencies, our focus lies solely on their energy levels. Consequently, CS measurements

are allocated among them based on their respective means at different levels. Finally, we set a bias term ($\sum_{i=0}^{n} bias_i = 0$) to increase the flexibility of this allocation method.

3.2.2 Block-by-Block Sampling. We employ the block-by-block sampling method to independently sample each non-overlapping image block with a fixed size of $n \times n$. Initially, an image $\boldsymbol{X} \in \mathbb{R}^{H \times W}$ is divided into blocks $\boldsymbol{x}_j \in \mathbb{R}^{n \times n}, j \in \{1, 2, \cdots, J\}$. Subsequently, these blocks are decomposed into multi-scale wavelet sub-bands $\boldsymbol{\theta}_s \in \mathbb{R}^{n_s \times n_s}$. Before sampling, each sub-band is vectorized as a vector. As illustrated in Figure 3, the sampling process of each sub-band $\boldsymbol{\theta}_s$ at decomposition level i within block $\boldsymbol{x}_j$ can be described as follows:

$$\boldsymbol{y}_s = \boldsymbol{\Phi}_s \boldsymbol{\Psi} \boldsymbol{x}_j = \boldsymbol{A}_s \boldsymbol{\theta}_s, \tag{7}$$

where $\boldsymbol{y}_s \in \mathbb{R}^{M_s \times J}$ is the CS measurements of $\boldsymbol{x}_j$ and $H \cdot W = J \cdot n^2$, $s \in \{LL_l, HL_i, LH_i, HH_i\}$, $1 \leq i \leq l$, i means the wavelet decomposition level. $\boldsymbol{A}_s \in \mathbb{R}^{M_s \times n^2}$ is the multi-scale sampling operator, which consists of two components (a multi-scale transform $\boldsymbol{\Psi}$ and a multi-scale block-based measurement process $\boldsymbol{\phi}$). $\boldsymbol{\Psi}$ produces L levels after multi-level wavelet decomposition, therefore, $\boldsymbol{\phi}$ consists of L different block-based sampling operators, one for each level.

3.3 WTP-Guided Reconstruction

The reconstruction module exploits the quad-tree representation of wavelet coefficients to capture the inter-dependencies among coefficients at different scales in an image. This representation enables us to identify groups of coefficients that can either be zero or non-zero simultaneously, thereby providing valuable structural insights for enhancing image reconstruction quality. Specifically, the WTP-guided reconstruction model can be described as follows:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \beta(\|\boldsymbol{\theta}\|_1 + \|\boldsymbol{G}\boldsymbol{\theta}\|_2), \tag{8}$$

where $\boldsymbol{G}$ represents the set of all parent-child groups for the wavelet tree. And we adopt weighted ℓ_1- and ℓ_2-norms in the second term in Equation (8) to impose the tree-structured sparsity within wavelet sub-bands.

3.3.1 Initial Reconstruction Module. The initial reconstruction module is illustrated by Equation (9). The wavelet coefficients of the original image can be initially reconstructed from its multi-scale measurements as shown in Figure 3:

$$\boldsymbol{\theta}_s = {\boldsymbol{A}_s}^T \boldsymbol{y}_s, \tag{9}$$

where $s \in \{LL_l, HL_i, LH_i, HH_i\}$, $1 < i < L$. There is only one low-frequency sub-band, so LL is not annotated with subscripts "i." ${\boldsymbol{A}_s}^T \in \mathbb{R}^{n^2 \times M_s}$ represents the linear mapping matrix and it is the pseudo-inverse matrix of $\boldsymbol{A}_s$. Then we transform $\boldsymbol{\theta}_s$ into full-image $\boldsymbol{X}^0$ by **inverse wavelet transform (IDWT)**.

3.3.2 Deep Reconstruction Module. The deep reconstruction process is performed on the initial reconstruction result $\boldsymbol{X}^0$ and improves its quality. We divide the reconstruction model into K stages. Each stage alternatively implements the projection in the wavelet domain and the full-image deblocking in the spatial domain. With $\boldsymbol{z} = \boldsymbol{G}\boldsymbol{\theta}$ constraint, Equation (8) can be rewritten as

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \beta(\|\boldsymbol{\theta}\|_1 + \|\boldsymbol{z}\|_2) + \frac{\lambda}{2}\|\boldsymbol{z} - \boldsymbol{G}\boldsymbol{\theta}\|_2^2. \tag{10}$$

The variables z and $\boldsymbol{\theta}$ are coupled together in the minimization of Equation (10), thus solving them simultaneously is computationally intractable. Therefore, we divide the problem into two sub-problems, namely $\boldsymbol{\theta}$-sub-problem and z-sub-problem, which are described as follows.

z-Sub-Problem. The goal of the z-sub-problem is to generate an intermediate variable to simplify the optimization problem. The z-sub-problem is equivalent to solving the following optimization problem:

$$z^k = \underset{z}{\mathbf{argmin}} \ \beta\|z\|_2 + \frac{\lambda}{2}\|z - G\boldsymbol{\theta}^{k-1}\|_2^2. \tag{11}$$

The mapping process for solving the corresponding optimization z-sub-problem is described as follows, which can be solved by using a **soft threshold function (STF)**. It is mapped into a deep network module in our framework.

$$z^k = \mathbf{max}\left(\|G\boldsymbol{\theta}^{k-1}\|_2 - \frac{\beta}{\lambda}, 0\right)\frac{G\boldsymbol{\theta}^{k-1}}{\|G\boldsymbol{\theta}^{k-1}\|_2}. \tag{12}$$

We implement the tree-structured group sparsity prior via Equation (12). Guided by the WTP, our unfolding network effectively exploits the inherent structural information embedded in an image. The incorporation of the wavelet tree structure as a guiding principle in our model enables us to exploit the hierarchical nature of wavelet decomposition, facilitating the capture of both local and global features present in an image. By integrating this prior into our framework, we are better equipped to handle complex images characterized by diverse textures and structures. As shown in Figure 7, it exhibits superior reconstruction performance on the complex dataset Urban100. Moreover, our unfolding network capitalizes on the inter-dependency between sub-bands at different scales, allowing for leveraging information from one scale to enhance or refine details at another scale. Specifically, the STF consists of the following steps, including:

(1) We concatenate the sub-bands of different groups in the channel dimension and treat a channel as a group. Then a **channel-wise global embedding (CGE)** is performed to $wg = \boldsymbol{\theta}^{k-1}$, such as CGE($\boldsymbol{\theta}^{k-1}$). It is composed of channel-wise attention, a ReLU layer, and a 1×1 convolutional.
(2) Based on the assumption that a group with higher energy requires a larger threshold, we calculate a group-wise threshold in the shrinkage as follows:

$$\boldsymbol{\theta}^{k-1} = wg * \boldsymbol{\theta}^{k-1}. \tag{13}$$

(3) Applying the group-wise soft-shrinkage operation Equation (12) with the new $\boldsymbol{\theta}^{k-1}$. The soft-shrinkage operation is efficiently implemented by using a ReLU function.

$\boldsymbol{\theta}$-Sub-Problem. The $\boldsymbol{\theta}$-sub-problem in Equation (10) can be written as follows:

$$\boldsymbol{\theta}^k = \underset{\boldsymbol{\theta}}{\mathbf{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{z}^k - \boldsymbol{G}\boldsymbol{\theta}\|_2^2 + \beta\|\boldsymbol{\theta}\|_1. \tag{14}$$

The $\boldsymbol{\theta}$-sub-problem in Equation (14) can be solved through the optimization algorithm of denoising perspective:

$$\boldsymbol{r^k} = \mathbf{GDM}(\boldsymbol{\theta}^{k-1}, \boldsymbol{z}^k, \lambda) = \boldsymbol{\theta}^{k-1} - \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\theta}^{k-1} - \boldsymbol{y}) - \lambda \boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{\theta}^{k-1} - \boldsymbol{z}^k), \tag{15}$$

$$\boldsymbol{\theta}^k = \underset{\boldsymbol{\theta}}{\mathbf{argmin}} \ \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{r}^k\|_2^2 + \beta\|\boldsymbol{\theta}\|_1, \tag{16}$$

where GDM denotes the gradient descent module at $\boldsymbol{r}^k$, as shown in Figure 2. And $\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{z}^k - \boldsymbol{G}\boldsymbol{\theta}\|_2^2$ is a convex smooth function with Lipschitz constant. The optimization problem in

Equation (16) can be solved as follows:

$$\hat{\boldsymbol{\theta}}^k = \mathbf{max}(\|\boldsymbol{r}^k\|_2 - w_i * \beta, 0)\frac{\boldsymbol{r}^k}{\|\boldsymbol{r}^k\|_2}, \tag{17}$$

$$\boldsymbol{\theta}^k = \hat{\boldsymbol{\theta}}^k - (\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I})\mathfrak{R}_k(\boldsymbol{\theta}^{k-1}). \tag{18}$$

We use the denoising block $\mathfrak{R}_k(\cdot)$ to remove the noise of $\boldsymbol{\theta}^{k-1}$ to get the kth reconstruction result $\boldsymbol{\theta}^k$. Each denoising module consists of four 3×3 convolutional layers. There is a ReLU activation function between adjacent convolutional layers. A channel attention module in our network is used to learn the corresponding weights $\{w_i | i = 1, 2, ...n\}$ of different image patch groups for improving flexibility.

In each deblocking block $\mathfrak{B}_k(\cdot)$, it consists of six convolutional layers. The ReLU activation function is applied between adjacent convolution layers. By combining $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_H$ using the IDWT, we can obtain the reconstruction result $\boldsymbol{X}^k$ of the kth phase, which combines the output set $[\boldsymbol{X}^0, \boldsymbol{X}^1, ..., \boldsymbol{X}^{k-1}]$ of the previous $k-1$ phases as input to deblock module. The deblock module integrated inter-context information can effectively remove block artifacts of full-image. We use a **cross-domain attention (CAC)** module before the inverse wavelet for better collaborative reconstruction of high-frequency and low-frequency components. The process of image deblocking in the kth reconstruction stage can be expressed as

$$\boldsymbol{X}^k = \mathbf{IDWT}(\mathbf{CAC}(\boldsymbol{\theta}_L^k, \boldsymbol{\theta}_H^k)), \tag{19a}$$

$$\boldsymbol{X}^k_{\text{output}} = \boldsymbol{X}^k - \mathfrak{B}_k(cat(\boldsymbol{X}^0, \boldsymbol{X}^1, ..., \boldsymbol{X}^k)). \tag{19b}$$

The input of $\mathfrak{B}_k(\cdot)$ is a set of whole concatenated images rather than each image block. Figure 3 illustrates the structures of the denoise module, deblock module, and CAC [43] module.

3.4 Loss Functions

In this article, we adopt the MSE as the difference metric between the mth ground-truth $\boldsymbol{X}_{\text{m}}$ in the training set and the final recovered result $\boldsymbol{X}^K_{\text{m}}$. The loss function is formulated as follows:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N_{\text{a}}N_{\text{b}}}\|\widetilde{\boldsymbol{X}}_{\text{m}} - \boldsymbol{X}^K_{\text{m}}\|_2^2, \tag{20}$$

where N_{a} denotes the size of $\widetilde{\boldsymbol{X}}_{\text{m}}$ and N_{b} denotes the size of the training set.

The high-frequency coefficients potentially converge to zero by performing convolution operations on them. To address this issue, we develop a texture loss function that aims to prevent the high-frequency coefficients from converging to zero:

$$\mathcal{L}_{\text{texture}} = \mathbf{max}(\|\widetilde{\boldsymbol{\theta}}_{\text{H}}\| - \|\boldsymbol{\theta}^k_{\text{H}}\| + \epsilon, 0), \tag{21}$$

where $\widetilde{\boldsymbol{\theta}}_{\text{H}}$ is the high-frequency coefficients of the original image in the training set, $\boldsymbol{\theta}^k_{\text{H}}$ is the reconstruction result at kth stage. ϵ is a bias term that prevents the texture from disappearing, which is a tensor of the same size as the high-frequency sub-bands and has values only at non-zero elements.

The initial loss denoted as the MSE between the output result $\boldsymbol{X}_{\text{init}}$ of the initial reconstruction module and the corresponding original image in the training set, defined as

$$\mathcal{L}_{\text{init}} = \frac{1}{N_{\text{a}}N_{\text{b}}}\|\widetilde{\boldsymbol{X}}_{\text{m}} - \boldsymbol{X}_{\text{init}}\|_2^2. \tag{22}$$

Finally, an end-to-end loss function is designed as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \gamma * \mathcal{L}_{\text{texture}} + \mu * \mathcal{L}_{\text{init}}, \tag{23}$$

where γ and μ are the regularization parameters. In our experiments, γ and μ are set to 0.001 and 0.01, respectively.

4 Experimental Results and Performance Evaluation

In this section, our proposed method is first compared with other state-of-the-art methods in terms of both objective reconstruction quality and subjective visual quality. Then a series of ablation experiments are implemented to further evaluate the effects of each functional module in our method. Finally, the model complexity is compared with other methods.

4.1 Experiment Settings

We use 400 images from the BSD500 dataset for training and validation. In the training process, two training sets are generated for models. (1) Training set with the size 64×64 is randomly extracted from images in BSDS500. (2) Training set with the size 128×128 is randomly extracted from images in BSDS500. We also unfold the whole testing image in this way during the testing process. We use the PyTorch toolbox and train our model using the Adam solver on an NVIDIA RTX 3090 GPU. All models are trained for 200 epochs with batch size 32 and learning rate 0.0001. Before training, the control parameter α is initialized as 1 and other trainable parameters are initialized randomly.

For testing results, we conduct extensive experiments on some widely used datasets: Set5 [30], Set11 [5], Set14, and Urban100 [33]. To ensure fairness, we evaluate the performance with two quality evaluation metrics: **Peak Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index (SSIM)**.

4.2 Comparison with Other Methods

In this section, we select several typical deep learning-based CS methods for comparison with our method, including DPA-Net [36], BCS-Net [57], OPINE-Net [52], AMP-Net [55], COAST-Net [48], MADUN-Net [33], ISTA-Net++ [47], FHDUN-Net [9], ULAMP-Net [41], DPUNet [45], SODAS-Net [35], and DPC-DUN [34]. Apart from some experimental results provided by the authors, the results of the other comparison methods are retrained on the Ubuntu 16.04 system under the PyTorch 1.7 framework and use Python 3.7. Moreover, we have also compared our method with some model-based methods on the Set11 dataset, as shown in Table 5. The source code of comparison methods comes from the official code published by their authors. Tables 1–4 show the test results of WTDUN and other methods on Set5, Set11, Set14, and Urban100. These tables contain the average PSNR (dB)/SSIM where the best is marked in bold and show the reconstruction results at different sampling rates of {50%, 40%, 30%, 20%, 10%}.

4.2.1 Reconstruction Quality. From Tables 1–4, it can be seen that WTDUN can achieve higher PSNR than those deep learning-based CS methods at all sampling ratios. As for SSIM, our WTDUN is better than most methods. It can be seen that MADUN-Net and ISTA-Net++ perform worse at low CS ratios of 10%. DPA-Net does not work well at all CS ratios. Compared to other methods, for example, the gain of our method with PSNR is about 1.01–5.57 dB at 50% CS ratio and 0.20–2.22 dB at 10% CS ratio on dataset Urban100. The gain of our method with SSIM is about 0.0021–0.0301 at 50% CS ratio and 0.0057$\sim$ −0.0561 at 10% CS ratio on Urban100.

4.2.2 Visual Effect. The visual quality comparison of different CS methods on test images is shown in Figures 4–7 at 10% CS ratio. It can be seen that the reconstructed images from COAST and ISTA-Net++ have obvious blocking artifacts. This is because these methods only focus on the individual reconstruction of each block and do not consider the correlations between neighboring image blocks. On the contrary, there are no obvious blocking artifacts in the reconstructed images generated by MADUN, OPINE-Net, AMP-Net, and our method, since these models all perform

Table 1. Average PSNR (dB)/SSIM Performance Comparisons on Set5 with Different CS Sampling Rates

Method	Set5 PSNR (dB)/SSIM				
	10%	20%	30%	40%	50%
DPA-Net [36]	30.32/0.8713	[a]/[a]	36.17/0.9495	38.05/0.9632	39.57/0.9716
AMP-Net [55]	31.95/0.9017	*35.49*/0.9419	*37.86*/0.9606	39.70/0.9713	41.51/0.9791
COAST [48]	30.50/0.8794	34.18/0.9298	36.48/0.9515	38.33/0.9645	40.21/0.9744
ISTA-Net++ [47]	29.61/0.8563	33.33/0.9173	35.62/0.9427	37.40/0.9575	38.94/0.9678
MADUN [33]	31.11/0.8910	34.80/0.9363	37.25/0.9561	39.29/0.9693	41.18/0.9784
ULAMP [41]	30.78/0.8774	31.94/0.8868	36.39/0.9599	38.20/0.9693	40.97/*0.9827*
DPUNet [45]	31.80/0.9079	35.38/*0.9458*	37.54/0.9618	39.44/0.9716	41.10/0.9783
Ours (64 × 64)	*32.14*/*0.9123*	35.39/0.9443	37.82/*0.9622*	*39.94*/*0.9734*	*42.09*/0.9824
Ours (128 × 128)	**32.23**/**0.9132**	**35.56**/**0.9461**	**38.04**/**0.9637**	**40.14**/**0.9746**	**42.18**/**0.9827**

The best and second-best results are highlighted in bold and italics, respectively.
[a]Indicates the case where the result at the corresponding sampling rate is missing in the original paper.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

Table 2. Average PSNR (dB)/SSIM Performance Comparisons on Set11 with Different CS Sampling Rates

Method	Set11 PSNR (dB)/SSIM				
	10%	20%	30%	40%	50%
DPA-Net [36]	26.99/0.8354	[a]/[a]	[a]/[a]	35.04/0.9565	36.73/0.9670
BCS-Net [57]	29.36/0.8650	32.87/0.9254	35.40/0.9527	36.52/0.9640	39.58/0.9734
COAST [48]	28.69/0.8618	32.54/0.9251	35.04/0.9501	37.13/0.9648	38.94/0.9744
ISTA-Net++ [47]	27.62/0.8358	31.66/0.9127	34.23/0.9427	36.28/0.9593	37.94/0.9693
MADUN [33]	29.29/0.8768	33.30/0.9355	36.00/0.9576	38.09/0.9700	39.86/0.9774
FHDUN [9]	29.53/0.8859	[a]/[a]	36.12/*0.9589*	38.04/0.9696	[a]/[a]
DPUNet [45]	29.30/0.8815	33.17/0.9357	35.75/0.9581	37.90/*0.9705*	39.69/0.9782
SODAS-Net [35]	28.89/0.8669	32.20/0.9243	35.55/0.9543	37.74/0.9680	39.60/0.9769
DPC-DUN [34]	29.40/0.8798	33.10/0.9334	35.88/0.9570	37.98/0.9694	39.84/0.9778
Ours (64 × 64)	*29.53*/*0.8867*	*33.58*/*0.9371*	*36.27*/0.9585	*38.26*/0.9698	*40.22*/*0.9784*
Ours (128 × 128)	**29.64**/**0.8877**	**33.81**/**0.9399**	**36.41**/**0.9597**	**38.45**/**0.9708**	**40.38**/**0.9789**

The best and second-best results are highlighted in bold and italics, respectively.
[a]Indicates the case where the result at the corresponding sampling rate is missing in the original paper.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

denoising and deblocking operations on the full image at each reconstruction stage. Compared with COAST, ISTA-Net++, MADUN, OPINE-Net, and AMP-Net algorithms, images recovered by our WTDUN have richer texture details and sharper edges than other methods as shown in Figures 5 and 7. Therefore, our WTDUN has a stronger ability to reconstruct high-quality images compared to other state-of-the-art methods.

4.3 Ablation Study

To evaluate the contribution of each component in our WTDUN, we design several variants of the proposed model in which some functional modules are selectively discarded or replaced. Tables 6 and 7 show comparative experimental results on Set11 and Urban100 at 20% and 50% CS ratios,

Table 3. Average PSNR (dB)/SSIM Performance Comparisons on Set14 with Different CS Sampling Rates

Method	Set14 PSNR (dB)/SSIM				
	10%	20%	30%	40%	50%
AMP-Net [55]	*28.69*/0.8171	*31.95*/0.8933	34.27/0.9293	36.26/*0.9505*	38.10/*0.9647*
COAST [48]	27.41/0.7799	30.71/0.8672	33.10/0.9106	35.12/0.9369	36.93/0.9549
ISTA-Net++ [47]	26.75/0.7549	30.09/0.8518	32.40/0.8999	34.26/0.9287	35.90/0.9477
MADUN [33]	27.97/0.7914	31.50/0.8790	34.05/0.9194	36.05/0.9439	37.96/0.9592
DPUNet [45]	28.49/*0.8226*	31.61/**0.8963**	33.95/*0.9308*	35.93/0.9505	37.66/0.9628
Ours (64 × 64)	28.61/0.8216	31.89/0.8908	*34.36*/0.9280	*36.27*/0.9486	*38.31*/0.9644
Ours (128 × 128)	**28.71**/**0.8235**	**32.10**/*0.8944*	**34.57**/**0.9308**	**36.53**/**0.9507**	**38.49**/**0.9653**

The best and second-best results are highlighted in bold and italics, respectively.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

Table 4. Average PSNR (dB)/SSIM Performance Comparisons on Urban100 with Different CS Sampling Rates

Method	Urban100 PSNR (dB)/SSIM				
	10%	20%	30%	40%	50%
DPA-Net [36]	24.55/0.7841	[a]/[a]	29.47/0.9034	31.09/0.9311	32.08/0.9447
OPINE-Net [52]	26.56/0.8345	30.07/0.9088	32.64/0.9419	34.66/0.9600	36.64/0.9727
AMP-Net [55]	25.96/0.8133	29.50/0.8974	32.07/0.9352	34.22/0.9569	36.16/0.9706
COAST [48]	25.94/0.8038	29.70/0.8940	32.20/0.9317	34.21/0.9528	35.99/0.9665
ISTA-Net++ [47]	24.78/0.7607	28.55/0.8687	31.08/0.9152	33.10/0.9402	34.86/0.9560
DPUNet [45]	26.10/0.8226	29.71/0.9027	32.23/0.9378	34.30/0.9573	36.10/0.9693
SODAS-Net [35]	26.23/0.8084	29.51/0.8950	33.15/0.9412	35.28/0.9599	37.14/0.9721
Ours (64 × 64)	*26.58*/*0.8358*	*30.58*/*0.9109*	*33.18*/*0.9431*	*35.26*/*0.9610*	*37.39*/*0.9738*
Ours (128 × 128)	**26.77**/**0.8402**	**30.82**/**0.9142**	**33.44**/**0.9459**	**35.60**/**0.9629**	**37.65**/**0.9748**

The best and second-best results are highlighted in bold and italics, respectively.
[a]Indicates the case where the result at the corresponding sampling rate is missing in the original paper.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

which include five functional modules. Tables 8 and 9 show the comparison of performing wavelet decomposition at different levels. Table 10 demonstrates the effect of different reconstruction phase numbers k on Urban100 and Set11 at 10% CS ratio. These Tables 1–4 show the effect of different patch sizes of image blocks on reconstruction quality on these datasets: Set5, Set11, Set14, and Urban100.

4.3.1 Validating the Capability of WAS. In this subsection, we validate the capability of the WAS method. CS measurements allocation method is used to adaptively allocate CS measurements according to the target sampling ratio. To test the effect of WAS, we train our network in settings with and without WAS on Urban100 and Set11, as shown in Tables 6 and 7. Compared to the case without WAS, the average PSNR scores of WTDUN can be improved by about 1.5–2.5 dB at 20% and 50% CS ratios. The average SSIM scores of our WTDUN can be improved by about 0.01–0.05 at 20% and 50% CS ratios with WAS. Ablation results demonstrate that WAS can effectively improve image reconstruction quality.

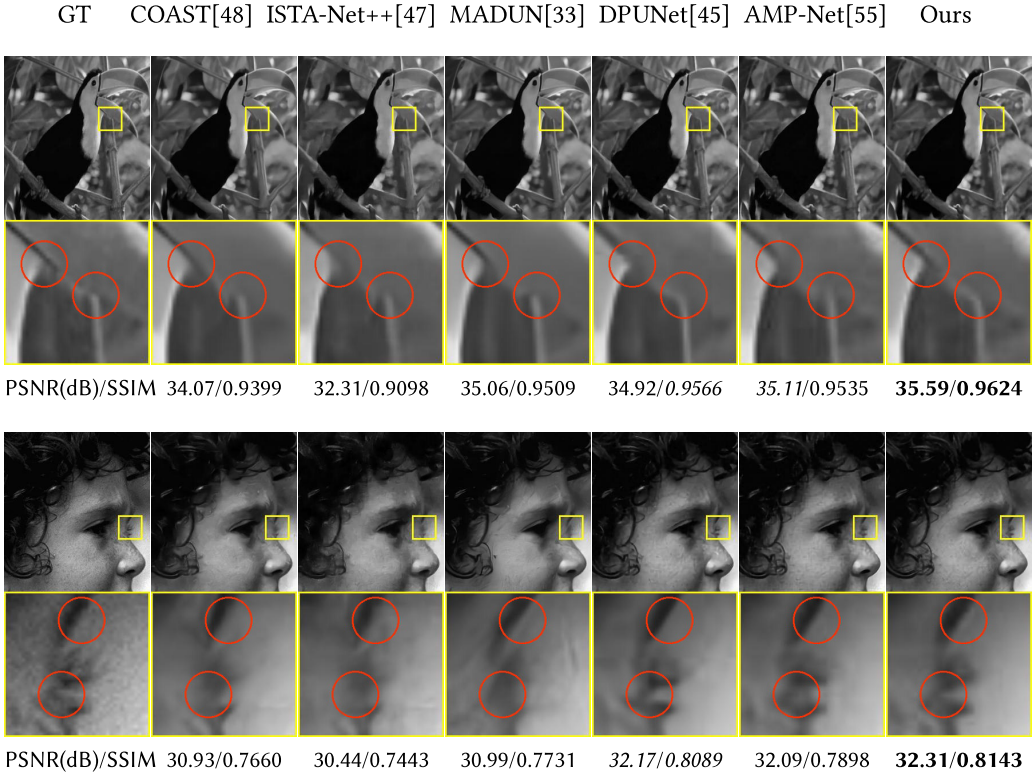

Fig. 4. Visual quality comparisons between our proposed method and recently state-of-the-art CS methods on Set5 at 10% CS ratio. The best and second-best results are highlighted in bold and italics, respectively.

4.3.2 Validating the Capability of WTP. In this subsection, we validate the capability of the wavelet tree-structured reconstruction model. The WTP is used to fully exploit the correlation within multi-scale sub-bands. To test the effect of WTP, we trained our model in settings with and without WTP. As shown in Tables 6 and 7, the PSNR scores of our WTDUN with WTP are improved by about 0.08–0.2 dB. It can be seen that our model can obtain higher PSNR/SSIM when the WTP is integrated into our framework.

*4.3.3 Validating the Capability of Deblocking, **Memory Context (MC)**, and CAC.* In this subsection, we validate the capability of the Deblock module, MC module and CAC module. The comparison results are presented in Tables 6 and 7. Of the three mentioned modules, the deblock module brings the largest improvement in PSNR/SSIM, with a gain of about 1–2.5 dB for PSNR and 0.02–0.1 for SSIM. The gain for MC and CAC is almost the same, about 0.2 dB for PSNR and 0.01 for SSIM.

4.3.4 Validating the Influence Image Patch Size. In this subsection, we validate the influence of different image patch sizes. Tables 1–4 show that the reconstruction performance can be improved with the increase of the block size. Larger patch size can help our model capture more contextual information, which is vital for reconstruction methods. The contextual information enables the method to more effectively estimate and fill in missing or corrupted areas of the image, resulting in a more accurate and visually coherent reconstruction.

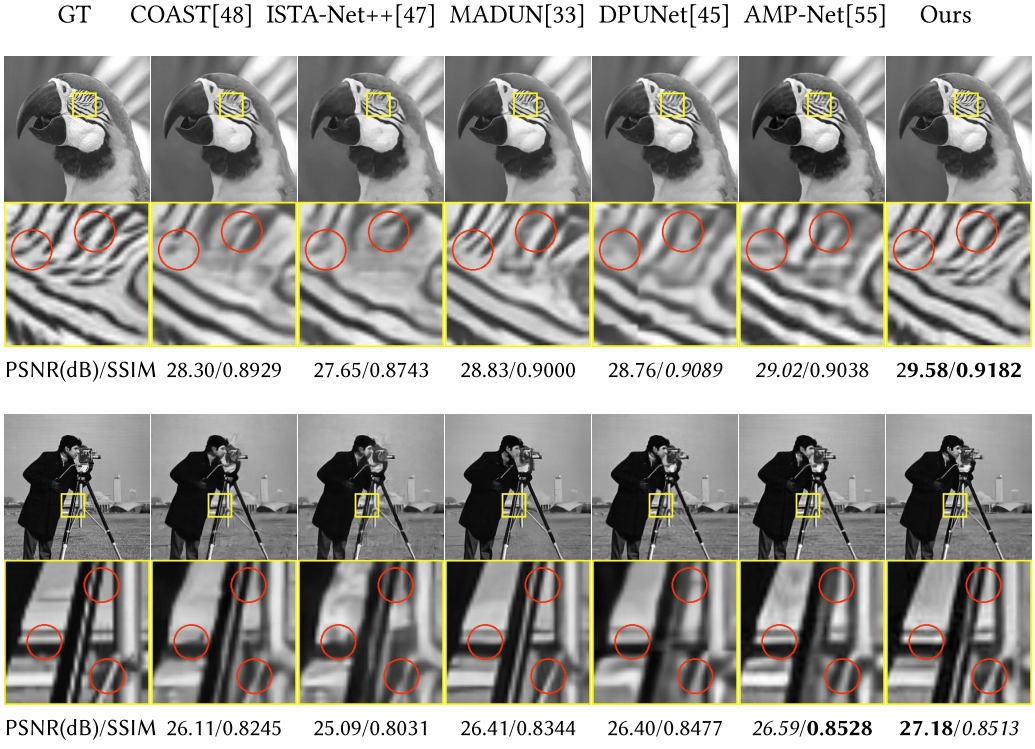

Fig. 5. Visual quality comparisons between our WTDUN and recently state-of-the-art CS methods on Set11 at 10% CS ratio. The best and second-best results are highlighted in bold and italics, respectively.

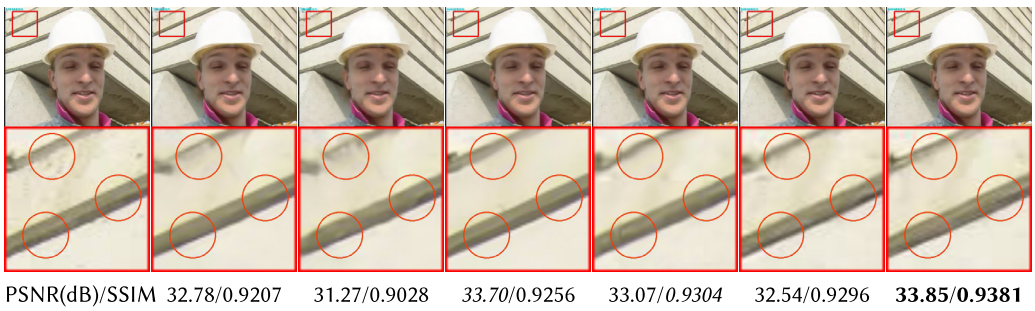

Fig. 6. Visual quality comparisons between our WTDUN and recently state-of-the-art CS methods on Set14 at 10% CS ratio. The best and second-best results are highlighted in bold and italics, respectively.

4.3.5 Validating the Capability of Wavelet Decomposition at Different Levels. In this subsection, we validate the capability of wavelet decomposition at different levels on Set11 and Urban100. Tables 8 and 9 show that the best reconstruction results can be obtained at two-level wavelet decomposition. Multi-level wavelet decomposition can increase image sparsity, which can lead to better reconstruction quality. Therefore, the PSNR/SSIM of the two-level and three-level wavelet decomposition are better than the one-level wavelet decomposition. Since we use 64×64 image

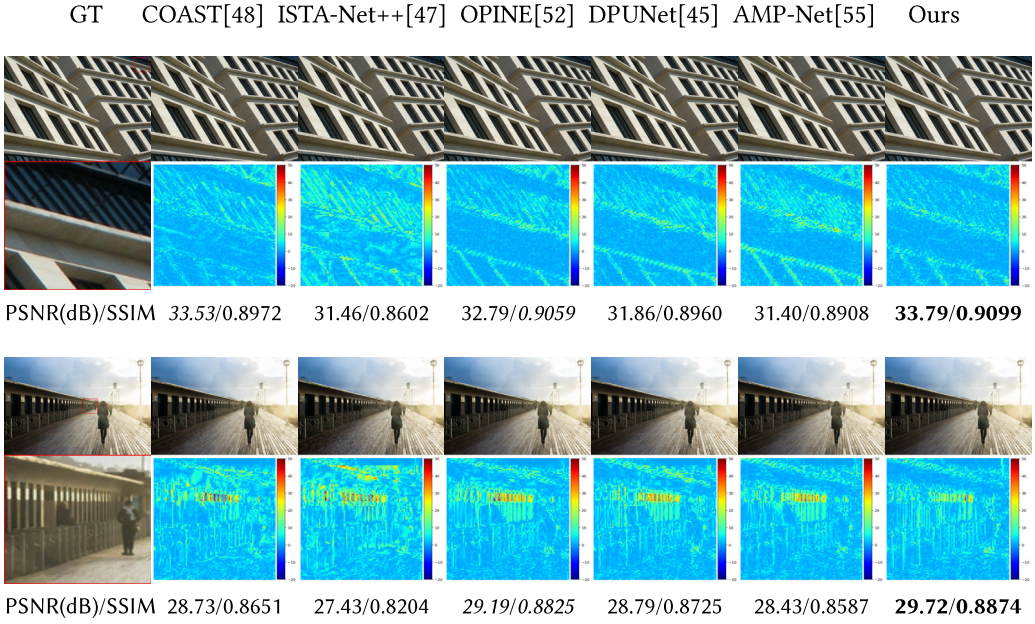

Fig. 7. Visual quality comparisons between the proposed WTDUN and recently state-of-the-art CS methods on Urban100 at 10% CS ratio. To better illustrate the difference, we show the residual heatmap between the reconstructed image patch and the GT patch. The best and second-best results are highlighted in bold and italics, respectively.

Table 5. Average PSNR/SSIM Comparisons with Other Multi-Scale Sampling Methods on Set11 Dataset

Method	Sampling Rate				
	10%	20%	30%	40%	50%
BCS-SPL-DWT [27]	23.31/0.7056	27.68/0.8057	28.41/0.8614	30.32/0.8977	32.13/0.9241
MS-BCS-SPL [15]	27.00/0.8293	30.21/0.8893	32.51/0.9398	34.01/0.9428	35.33/0.9489
MS-DCSNet [3]	28.58/0.8648	32.20/0.9215	34.63/0.9478	36.65/0.9627	38.68/0.9740
Ours (64 × 64)	*29.53/0.8867*	*33.58/0.9371*	*36.27/0.9585*	*38.26/0.9698*	*40.22/0.9784*
Ours (128 × 128)	**29.64/0.8877**	**33.81/0.9399**	**36.41/0.9597**	**38.45/0.9708**	**40.38/0.9789**

The best and second-best results are highlighted in bold and italics, respectively.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

blocks as input to our model, we can get some 8×8 wavelet sub-bands after three-level decomposition. Because the wavelet sub-bands of size 8×8 are too small for deep networks, the useful feature cannot be effectively extracted, resulting in poor reconstruction quality. Thus, the PSNR/SSIM of the two-level wavelet decomposition is better than that of the three-level wavelet decomposition.

4.3.6 Validating the Capability of Different Reconstruction Stage K. To investigate the selection of $K = 9$ as the optimal value for our CS model, we conduct a series of comparative experiments. These experiments show that when K goes beyond 9, the improvement in PSNR/SSIM becomes marginal. To balance the performance and complexity, $K = 9$ is selected as the number of iterations in our CS reconstruction model, as depicted in Figure 8 and Table 10.

Table 6. The Effect of WAS Method, WTP, Deblock, Memory Context (MC) Module, and CAC on Reconstruction Quality

Cases	Modules					PSNR/SSIM	
	WAS	WTP	Deblock	MC	CAC	20%	50%
(a)		✓	✓	✓	✓	28.59/0.8649	35.61/0.9600
(b)	✓		✓	✓	✓	30.45/0.9010	37.38/0.9710
(c)	✓	✓		✓	✓	28.05/0.8588	34.43/0.9552
(d)	✓	✓	✓		✓	30.30/0.8995	37.12/0.9703
(e)	✓	✓	✓	✓		30.31/0.9010	36.97/0.9699
WTDUN	✓	✓	✓	✓	✓	**30.58/0.9109**	**37.39/0.9738**

The sampling ratio of CS measurement is 20% and 50% on the Urban100 dataset. The best results are highlighted in bold.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

Table 7. The Effect of WAS Method, WTP, Deblock, MC Module, and CAC on Reconstruction Quality

Cases	Modules					PSNR/SSIM	
	WAS	WTP	Deblock	MC	CAC	20%	50%
(a)		✓	✓	✓	✓	31.63/0.9080	38.02/0.9675
(b)	✓		✓	✓	✓	33.46/0.9302	40.21/0.9766
(c)	✓	✓		✓	✓	31.67/0.9051	38.06/0.9681
(d)	✓	✓	✓		✓	33.41/0.9305	39.98/0.9758
(e)	✓	✓	✓	✓		33.43/0.9311	39.91/0.9759
WTDUN	✓	✓	✓	✓	✓	**33.58/0.9371**	**40.22/0.9784**

The sampling ratio of CS is 20% and 50% on the Set11 dataset. The best results are highlighted in bold.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

Table 8. The Effect of Multi-Level Wavelet Transform on Reconstruction Quality

Method	PSNR (dB)/SSIM		
	10%	30%	50%
Ours-level1	29.50/0.8791	35.26/0.9581	40.00/0.9762
Ours-level2	**29.53/0.8867**	**36.27/0.9585**	**40.22/0.9784**
Ours-level3	29.41/0.8808	36.18/0.9578	40.05/0.9781

The sampling ratio of CS is 10%, 30%, and 50% on the Set11 dataset. The best results are highlighted in bold.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

4.4 Model Complexity

In this section, we compare the model complexity of different CS approaches in terms of parameter capacity and time complexity. The average running time is used to evaluate the actual reconstruction efficiency. Note that the average running time and the number of parameters are tested by reconstructing nine test images of size 256×256.

Table 9. The Effect of Multi-Level Wavelet Transform on Reconstruction Quality

Method	PSNR (dB)/SSIM		
	10%	30%	50%
Ours-level1	26.52/0.8265	31.83/0.9390	37.01/0.9702
Ours-level2	**26.58/0.8358**	**33.18/0.9431**	**37.39/0.9738**
Ours-level3	26.42/0.8297	33.17/0.9430	37.26/0.9732

The sampling ratio of CS is 10%, 30%, and 50% on the Urban100 dataset.
The best results are highlighted in bold.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

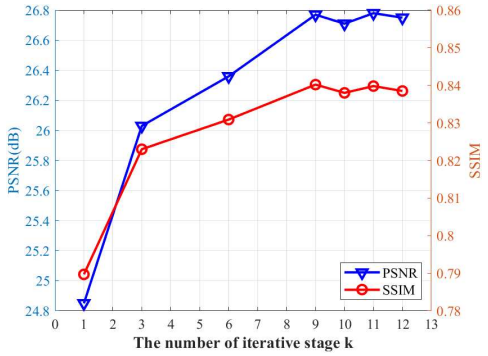

Fig. 8. The PSNR/SSIM curve with the increase of iterative stage K.

4.4.1 Parameter Capacity. Table 11 shows the number of sampling matrix parameters for ISTA-Net, ISTA-Net++, OPINE-Net, AMP-Net, COAST, MADUN, DPUNet, and our method. For a normal task, our method can achieve better reconstruction quality and faster reconstruction speed compared to MADUN with a similar number of parameters. Even though our method samples with larger patches (64×64), the number of parameters we use only increases by about 0.4 Mb parameters rather than nearly four times parameters compared to small-size patches (33×33).

4.4.2 Time Complexity. As shown in Table 11, the average running time of deep learning methods on GPU is less than 0.2 seconds. The reconstruction process of traditional CS methods is implemented iteratively until convergence or the maximum iteration step is reached. As a result, the reconstruction speed of traditional methods is lower than deep learning methods and deep unfolding methods. Moreover, GPUs have advantages over CPUs in terms of computational power and parallel computing capability. This can greatly reduce the running time of deep learning-based CS and deep unfolding approaches.

All CS models shown in Table 11 have different time complexities. Owing to the superior computing power of GPUs, the slight difference in the running time of these methods is not significant. Since there is only a small difference in the running time of models, image reconstruction quality is more important for deep learning-based methods. In summary, our method achieves a better accuracy-complexity tradeoff than other state-of-the-art methods.

Table 10. PSNR/SSIM of Models with Different Reconstruction Numbers k on Urban100 at the CS Ratio of 10%

Method	PSNR (dB)/SSIM			
	$k = 1$	$k = 3$	$k = 6$	$k = 9$
ISTA [50]	22.03/0.6502	23.07/0.7001	23.48/0.7176	23.56/0.7230
ISTA++ [47]	17.45/0.3585	21.87/0.6322	23.29/0.7072	24.14/0.7421
AMP-Net [55]	**25.03**/*0.7842*	*25.63*/*0.8042*	*25.96*/*0.8133*	*26.05*/*0.8156*
Ours	*24.85*/**0.7897**	**26.03**/**0.8230**	**26.36**/**0.8309**	**26.58**/**0.8358**

The best results are highlighted in bold.
PSNR, peak signal-to-noise ratio; SSIM, structural similarity index.

Table 11. Model Complexity Comparison between WTDUN and Other CS Methods

Methods	Parameters		
	PN (Mb)	PM (Mb)	Time (s)
ISTA-Net [50]	1.05	2.57	0.083
OPINE-Net [52]	2.13	4.18	0.099
AMP-Net [55]	2.13	5.40	0.072
COAST [48]	-	8.56	0.093
ISTA-Net++ [47]	2.13	5.80	0.082
MADUN [33]	2.13	23.04	0.177
DPUNet [45]	-	12.1	0.071
Ours	2.50	19.72	0.170

PN and PM are the number of the learnable matrix parameters and total parameters, respectively. Running time is computed at 50% CS ratio.

5 Conclusions

We propose a novel wavelet domain framework named WTDUN for image CS, which enables simultaneous sampling based on sub-band characteristics and reconstruction guided by WTP. To achieve better CS sampling, we have designed an algorithm that allocates CS measurements based on the importance of each sub-band. Additionally, we have developed a wavelet domain sampling method to achieve adaptive sampling, thereby effectively enhancing the acquisition capability of sampling information. For CS reconstruction, we propose a tree-structured prior guided unfolding network. This innovative method effectively maintains a similar structure among wavelet sub-bands within a group by fully exploiting their inherent sparsity. By leveraging the advantages of tree-structured sparsity, our method significantly enhances the quality of image reconstruction compared to other state-of-the-art CS methods. Furthermore, as we increase the number of reconstruction modules, we observe further improvements in the reconstructed results. Our future work will focus on extending our model to exploit structural differences between blocks and facilitate effective interaction among different stages of reconstruction.

References

[1] Amir Beck and Marc Teboulle. 2009. A Fast Iterative Shrinkage-Thresholding Algorithm with Application to Wavelet-Based Image Deblurring. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09)*, 693–696.

[2] E. J. Candes, J. Romberg, and T. Tao. 2006. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Transactions on Information Theory* 52, 2 (2006), 489–509.

[3] Thuong Nguyen Canh and Byeungwoo Jeon. 2018. Multi-Scale Deep Compressive Sensing Network. In *Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP '18)*, 1–4.

[4] Cheng Chen and Junzhou Huang. 2014. The Benefit of Tree Sparsity in Accelerated MRI. *Medical Image Analysis* 18, 6 (2014), 834–842.

[5] Jiwei Chen, Yubao Sun, Qingshan Liu, and Rui Huang. 2020. Learning Memory Augmented Cascading Network for Compressed Sensing of Images. In *Proceedings of the European Conference on Computer Vision*, 513–529.

[6] Scott Saobing Chen, David L. Donoho, and Michael A. Saunders. 1998. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing* 20 (1998), 33–61.

[7] Wenjun Chen, Chunling Yang, and Xin Yang. 2022. FSOINET: Feature-Space Optimization-Inspired Network for Image Compressive Sensing. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '22)*, 2460–2464.

[8] Zan Chen, Xingsong Hou, Xueming Qian, and Chen Gong. 2018. Efficient and Robust Image Coding and Transmission Based on Scrambled Block Compressive Sensing. *IEEE Transactions on Multimedia* 20, 7 (2018), 1610–1621.

[9] Wenxue Cui, Shaohui Liu, and Debin Zhao. 2022. Fast Hierarchical Deep Unfolding Network for Image Compressed Sensing. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2739–2748.

[10] D. L. Donoho. 2006. Compressed Sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306.

[11] David L. Donoho, Arian Maleki, and Andrea Montanari. 2010. Message Passing Algorithms for Compressed Sensing: I. Motivation and Construction. In *Proceedings of the 2010 IEEE Information Theory Workshop on Information Theory (ITW '10),* 1–5.

[12] Jiang Du, Xuemei Xie, Chenye Wang, Guangming Shi, Xun Xu, and Yuxiang Wang. 2019. Fully Convolutional Measurement Network for Compressive Sensing Image Reconstruction. *Neurocomputing* 328 (2019), 105–112.

[13] Zi-En Fan, Feng Lian, and Jia-Ni Quan. 2022. Global Sensing and Measurements Reuse for Image Compressed Sensing. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '22)*, 8944–8953.

[14] Jing Fang, Yinbo Yu, Zhongyuan Wang, Xin Ding, and Ruimin Hu. 2023. An Image Arbitrary-Scale Super-Resolution Network Using Frequency-Domain Information. *ACM Transactions on Multimedia Computing, Communications, and Applications* 20, 3 (2023), 1–23.

[15] James E. Fowler, Sungkwang Mun, and Eric W. Tramel. 2011. Multiscale Block Compressed Sensing with Smoothed Projected Landweber Reconstruction. In *Proceedings of the 2011 19th European Signal Processing Conference*, 564–568.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2672–2680.

[17] Tao Huang, Weisheng Dong, Xin Yuan, Jinjian Wu, and Guangming Shi. 2021. Deep Gaussian Scale Mixture Prior for Spectral Compressive Imaging. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '21)*, 16211–16220.

[18] Tao Huang, Xin Yuan, Weisheng Dong, Jinjian Wu, and Guangming Shi. 2023. Deep Gaussian Scale Mixture Prior for Image Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 9 (2023), 10778–10794.

[19] Shihao Ji, Ya Xue, and Lawrence Carin. 2008. Bayesian Compressive Sensing. *IEEE Transactions on Signal Processing* 56, 6 (2008), 2346–2356.

[20] Zutao Jiang, Changlin Li, Xiaojun Chang, Ling Chen, Jihua Zhu, and Yi Yang. 2023. Dynamic Slimmable Denoising Network. *IEEE Transactions on Image Processing* 32 (2023), 1583–1598.

[21] Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang. 2013. An Efficient Augmented Lagrangian Method with Applications to Total Variation Minimization. *Computational Optimization and Applications* 56 (2013), 507–530.

[22] Suhas Lohit, Kuldeep Kulkarni, Ronan Kerviche, Pavan Turaga, and Amit Ashok. 2018. Convolutional Neural Networks for Noniterative Reconstruction of Compressively Sensed Images. *IEEE Transactions on Computational Imaging* 4, 3 (2018), 326–340.

[23] Jiawei Ma, Xiao-Yang Liu, Zheng Shou, and Xin Yuan. 2019. Deep Tensor ADMM-Net for Snapshot Compressive Imaging. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV '19)*, 10222–10231.

[24] S. Mallat and Z. Zhang. 1992. Adaptive Time-Frequency Decomposition with Matching Pursuits. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, 7–10.

[25] Christopher A. Metzler, Ali Mousavi, and Richard G. Baraniuk. 2017. Learned D-AMP: Principled Neural Network Based Compressive Image Recovery. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17)*, 1770–1781.

[26] Ali Mousavi, Ankit B. Patel, and Richard G. Baraniuk. 2015. A Deep Learning Approach to Structured Signal Recovery. In *Proceedings of the 2015 53rd Annual Allerton Conference on Communication, Control, and Computing*, 1336–1343.

[27] Sungkwang Mun and James E. Fowler. 2009. Block Compressed Sensing of Images Using Directional Transforms. In *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP '09)*, 3021–3024.
[28] Sujit Kumar Sahoo and Anamitra Makur. 2015. Signal Recovery from Random Measurements via Extended Orthogonal Matching Pursuit. *IEEE Transactions on Signal Processing* 63, 10 (2015), 2572–2581.
[29] Minghe Shen, Hongping Gan, Chao Ning, Yi Hua, and Tao Zhang. 2022. TransCS: A Transformer-Based Hybrid Architecture for Image Compressed Sensing. *IEEE Transactions on Image Processing* 31 (2022), 6991–7005.
[30] Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. 2019. Scalable Convolutional Neural Network for Image Compressed Sensing. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '19),* 12282–12291.
[31] Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. 2020. Image Compressed Sensing Using Convolutional Neural Network. *IEEE Transactions on Image Processing* 29 (2020), 375–388.
[32] Wuzhen Shi, Shaohui Liu, Feng Jiang, and Debin Zhao. 2021. Video Compressed Sensing Using a Convolutional Neural Network. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 2 (2021), 425–438.
[33] Jiechong Song, Bin Chen, and Jian Zhang. 2021. Memory-Augmented Deep Unfolding Network for Compressive Sensing. In *Proceedings of the 29th ACM International Conference on Multimedia*, 4249–4258.
[34] Jiechong Song, Bin Chen, and Jian Zhang. 2023. Dynamic Path-Controllable Deep Unfolding Network for Compressive Sensing. *IEEE Transactions on Image Processing* 32 (2023), 2202–2214.
[35] Jiechong Song and Jian Zhang. 2023. SODAS-Net: Side-Information-Aided Deep Adaptive Shrinkage Network for Compressive Sensing. *IEEE Transactions on Instrumentation and Measurement* 72 (2023), 1–12.
[36] Yubao Sun, Jiwei Chen, Qingshan Liu, Bo Liu, and Guodong Guo. 2020. Dual-Path Attention Network for Compressed Sensing Image Reconstruction. *IEEE Transactions on Image Processing* 29 (2020), 9482–9495.
[37] Yubao Sun, Ying Yang, Qingshan Liu, Jiwei Chen, Xiao-Tong Yuan, and Guodong Guo. 2020. Learning Non-Locally Regularized Compressed Sensing Network with Half-Quadratic Splitting. *IEEE Transactions on Multimedia* 22, 12 (2020), 3236–3248.
[38] Anhong Wang, Lei Liu, Bing Zeng, and Huihui Bai. 2011. Progressive Image Coding Based on an Adaptive Block Compressed Sensing. *IEICE Electronics Express* 8 (2011), 575–581.
[39] Xuemei Xie, Chenye Wang, Jiang Du, and Guangming Shi. 2018. Full Image Recover for Block-Based Compressive Sensing. In *Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME '18)*, 1–6.
[40] Xuemei Xie, Yuxiang Wang, Guangming Shi, Chenye Wang, Jiang Du, and Xiao Han. 2017. Adaptive Measurement Network for CS Image Reconstruction. In *Proceedings of the 2nd CCF Chinese Conference Computer Vision*, 407–417.
[41] Jingyu Yang, Xiangjun Yin, Mengxi Zhang, Huihui Yue, Xingyu Cui, and Huanjing Yue. 2022. Learning Image Formation and Regularization in Unrolling AMP for Lensless Image Reconstruction. *IEEE Transactions on Computational Imaging* 8 (2022), 479–489.
[42] Peihao Yang, Linghe Kong, Meikang Qiu, Xue Liu, and Guihai Chen. 2021. Compressed Imaging Reconstruction with Sparse Random Projection. *ACM Transactions on Multimedia Computing, Communications, and Applications* 17, 1 (2021), 1–25.
[43] Yuxiang Yang, Qi Cao, Jing Zhang, and Dacheng Tao. 2022. CODON: On Orchestrating Cross-Domain Attentions for Depth Super-Resolution. *International Journal of Computer Vision* 130, 2 (2022), 267–284.
[44] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. 2020. ADMM-CSNet: A Deep Learning Approach for Image Compressive Sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 3 (2020), 521–538.
[45] Yixiao Yang, Ran Tao, Kaixuan Wei, and Ying Fu. 2022. Dynamic Proximal Unrolling Network for Compressive Imaging. *Neurocomputing* 510 (2022), 203–217.
[46] Dongjie Ye, Zhangkai Ni, Hanli Wang, Jian Zhang, Shiqi Wang, and Sam Kwong. 2023. CSformer: Bridging Convolution and Transformer for Compressive Sensing. *IEEE Transactions on Image Processing* 32 (2023), 2827–2842.
[47] Di You, Jingfen Xie, and Jian Zhang. 2021. ISTA-NET++: Flexible Deep Unfolding Network for Compressive Sensing. In *Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME '21)*, 1–6.
[48] Di You, Jian Zhang, Jingfen Xie, Bin Chen, and Siwei Ma. 2021. COAST: COntrollable Arbitrary-Sampling NeTwork for Compressive Sensing. *IEEE Transactions on Image Processing* 30 (2021), 6066–6080.
[49] Roman Zeyde, Michael Elad, and Matan Protter. 2012. On Single Image Scale-Up Using Sparse Representations. In *Proceedings of the 7th International Conference on Curves and Surfaces*, 711–730.
[50] Jian Zhang and Bernard Ghanem. 2018. ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '18),* 1828–1837.
[51] Jian Zhang, Zhenyu Zhang, Jingfen Xie, and Yongbing Zhang. 2022. High-Throughput Deep Unfolding Network for Compressive Sensing MRI. *IEEE Journal of Selected Topics in Signal Processing* 16, 4 (2022), 750–761.
[52] Jian Zhang, Chen Zhao, and Wen Gao. 2020. Optimization-Inspired Compact Deep Compressive Sensing. *IEEE Journal of Selected Topics in Signal Processing* 14, 4 (2020), 765–774.

[53] Jian Zhang, Debin Zhao, and Wen Gao. 2014. Group-Based Sparse Representation for Image Restoration. *IEEE Transactions on Image Processing* 23, 8 (2014), 3336–3351.
[54] Jian Zhang, Debin Zhao, Ruiqin Xiong, Siwei Ma, and Wen Gao. 2014. Image Restoration Using Joint Statistical Modeling in a Space-Transform Domain. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 6 (2014), 915–928.
[55] Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. 2021. AMP-Net: Denoising-Based Deep Unfolding for Compressive Image Sensing. *IEEE Transactions on Image Processing* 30 (2021), 1487–1500.
[56] Chen Zhao, Siwei Ma, Jian Zhang, Ruiqin Xiong, and Wen Gao. 2017. Video Compressive Sensing Reconstruction via Reweighted Residual Sparsity. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 6 (2017), 1182–1195.
[57] Siwang Zhou, Yan He, Yonghe Liu, Chengqing Li, and Jianming Zhang. 2021. Multi-Channel Deep Networks for Block-Based Image Compressive Sensing. *IEEE Transactions on Multimedia* 23 (2021), 2627–2640.
[58] Shuyuan Zhu, Bing Zeng, and Moncef Gabbouj. 2014. Adaptive Reweighted Compressed Sensing for Image Compression. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS '14)*, 1–4.

Received 18 December 2023; revised 18 May 2024; accepted 4 October 2024