Reconstruction-free Cascaded Adaptive Compressive Sensing

Chenxi Qiu, Tao Yue, Xuemei Hu
School of Electronic Science and Engineering, Nanjing University, Nanjing, China
chenxiqiu@smail.nju.edu.cn, yuetao@nju.edu.cn, xuemeihu@nju.edu.cn

Abstract

Scene-aware Adaptive Compressive Sensing (ACS) has constituted a persistent pursuit, holding substantial promise for the enhancement of Compressive Sensing (CS) performance. Cascaded ACS furnishes a proficient multi-stage framework for adaptively allocating the CS sampling based on previous CS measurements. However, reconstruction is commonly required for analyzing and steering the successive CS sampling, which bottlenecks the ACS speed and impedes the practical application in time-sensitive scenarios. Addressing this challenge, we propose a reconstruction-free cascaded ACS method, which requires NO reconstruction during the adaptive sampling process. A lightweight Score Network (ScoreNet) is proposed to directly determine the ACS allocation with previous CS measurements and a differentiable adaptive sampling module is proposed for end-to-end training. For image reconstruction, we propose a Multi-Grid Spatial-Attention Network (MGSANet) that could facilitate efficient multi-stage training and inferencing. By introducing the reconstruction-fidelity supervision outside the loop of the multi-stage sampling process, ACS can be efficiently optimized and achieve high imaging fidelity. The effectiveness of the proposed method is demonstrated with extensive quantitative and qualitative experiments, compared with the state-of-the-art CS algorithms.

1. Introduction

Compressive sensing provides an efficient way to sample the scene information with sub-Nyquist rate [13], which has been applied in a wide range of research fields, such as medical imaging [29], wireless broadcasting [27], ultrafast photography [17] and video snapshot compressive imaging [47, 48]. CS methods with uniform sampling [8, 26, 43, 52, 56, 59] propose to sample each region of the image with the same sampling rate. Since the complexity and content in different image regions are distributed non-uniformly, adaptively allocating different sampling rates based on scene-dependent information is highly promising to realize efficient CS with high reconstruction fidelity. Therefore, different adaptive sampling methods are proposed [6, 34, 38, 49].

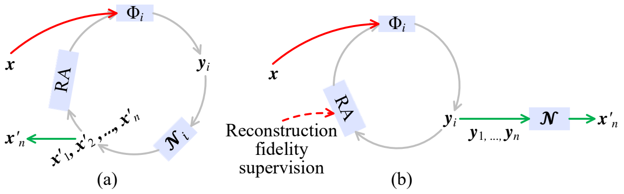
Φ_i
$\boldsymbol{x}$
$\boldsymbol{y}_i$
RA
$\boldsymbol{x}'_n$
$\boldsymbol{x}'_1, \boldsymbol{x}'_2, ..., \boldsymbol{x}'_n$
$\boldsymbol{\mathcal{N}}_i$
(a)
Φ_i
$\boldsymbol{x}$
RA
Reconstruction fidelity supervision
$\boldsymbol{y}_i$
$\boldsymbol{y}_{1,...,}\boldsymbol{y}_n$
$\boldsymbol{\mathcal{N}}$
$\boldsymbol{x}'_n$
(b)

Figure 1. The scheme comparison of n-stage ACS frameworks, (a) the existing ACS methods with n times of reconstruction of $\boldsymbol{x}'_1$, $\boldsymbol{x}'_2$, ..., $\boldsymbol{x}'_n$ in the adaptive sampling loop, and (b) the proposed method which requires no reconstruction during the sampling loop. Only the final image reconstruction with all adaptive measurements, i.e., $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, .., $\boldsymbol{y}_n$, is required. RA denotes the sampling rate allocation module and $\boldsymbol{\mathcal{N}}_i$ denotes the reconstruction at the i-th adaptive sampling stage.

Due to the efficiency of ACS in utilizing scene-dependent information, it has been applied in various fields, such as medical imaging [33], hyperspectral imaging [19], terahertz (THz) imaging [44] and 3D imaging [11, 35].

Generally, adaptive sampling is performed in multi-stage, reconstruction based on previous measurements is required for determining the subsequent adaptive sampling. Existing works propose to analyze the texture or saliency distribution on the previously reconstructed coarse image and allocate higher sampling rates to the regions with richer textures [1, 6, 34, 38, 49]. However, the requirement of image reconstruction in the loop of the adaptive sampling process prevents ACS from efficient sampling for time-sensitive scenarios. As shown in Fig. 1(a), for the existing multi-stage ACS process with n stages, reconstruction lies in the loop of the ACS process, which is required n times for the successive reconstruction of the image and bottlenecks the imaging speed of ACS for practical applications.

In this paper, we propose a reconstruction-free cascaded ACS framework, which requires NO reconstruction during the multi-stage adaptive sampling process. As shown in Fig. 1(b), during the sampling process, the adaptive allocation is determined directly based on the previous CS measurements, and only one reconstruction is required for the final image reconstruction. Specifically, our method proposes a lightweight ScoreNet to score each block based on the previous measurements. A linear programming (LP)-based differentiable adaptive sampling module (ASM) is proposed to implement CS sampling based on the scores, enabling end-to-end optimization of the ACS framework. To realize efficient reconstruction for training and inferencing, we propose a Multi-Grid Spatial-Attention-based reconstruction Network, i.e., MGSANet. The out-of-loop supervision is introduced during the training process to promote the convergence of the overall network. In all, our contributions are concluded as below.

- We propose a reconstruction-free cascaded ACS framework that requires NO reconstruction during the adaptive sampling process and overcomes the bottleneck of imaging speed for multi-stage ACS methods.
- We design a differentiable adaptive sampling method, composed of a lightweight ScoreNet and LP-based adaptive sampling module, that enables end-to-end optimization of the reconstruction-free ACS framework.
- We develop an efficient MGSANet for CS reconstruction to enable efficient training and inferencing. By introducing an out-of-loop reconstruction fidelity supervision during the training process, the proposed ACS framework is optimized with high-fidelity imaging performance.
- The effectiveness of the proposed method is extensively demonstrated by comparing it with state-of-the-art methods and ablation studies.

2. Related work

CS methods with adaptive sampling. Due to the potential of significant improvement in the sampling efficiency of CS with scene-dependent information, ACS has been explored in many fields, such as ghost imaging [1], medical imaging [33], hyperspectral imaging [19], 3D imaging [11, 35], and terahertz (THz) imaging [44], demonstrating the superiority of introducing adaptive sampling. Different from uniformly sampling each region of the image, how to design the ACS framework for efficiency and high fidelity is still an open problem. Existing works propose to sample the image under a two-stage or multiple-stage framework, which realizes the adaptive sampling based on the texture or saliency analysis of the previous reconstruction results. Specifically, several two-stage ACS models [2, 6, 55] are proposed, which uniformly sample the original image and reconstruct the coarse image in the first stage. Then, adaptive sampling based on the analysis of the coarse reconstruction image is implemented in the second stage. Beyond two-stage ACS models, multi-stage-based ACS frameworks are proposed, where adaptive sampling allocation can be achieved by successively accumulating information from the scene, promising efficient utilization of scene-dependent information. Based upon the reconstructed results of the previous measurements, texture analysis based on wavelet transform [1, 11, 19, 38, 41, 49, 54], Fourier transform [24], DCT [28], gradient domain [35], edge detection [44], and fluorescence signal domain [3] is introduced for steering the adaptive sampling allocation to regions with abundant textures. Furthermore, Qiu *et al.* [36, 37] proposes a multi-stage ACS model to allocate the sampling rate based on the measurement error. However, within these multi-stage adaptive sampling processes, image reconstruction based upon previous measurements is commonly required for determining the adaptive allocation of the next stage, which bottlenecks the speed of the sampling process and prevents practical time-sensitive scenarios. In this paper, we propose a reconstruction-free cascade ACS framework, which realizes multi-stage ACS without requiring reconstruction during the adaptive sampling process.

CS Reconstruction Neural Network. With the success of deep learning in computer vision, a series of CS reconstruction algorithms based on deep neural networks [10, 15, 26] have been proposed, which largely improve the efficiency of CS compared to traditional optimization-based algorithms [7, 12, 31, 45]. Recently, transformer [46] has achieved great success in the field of natural language processing, which has also been introduced into CS reconstruction algorithms [16, 39, 52] to capture long-range dependencies. Besides, deep unfolding networks (DUN) that combine traditional optimization algorithms with deep neural networks [8, 9, 42, 53, 56, 57, 59] or transformer [43] are proposed and achieve state-of-the-art performance in CS reconstruction quality. Owing to the proficiency of GPUs in parallel processing, a model with extensive parallelization can achieve markedly greater acceleration than a less parallelized model under identical floating point operations (FLOPs) [30]. Therefore, several multi-branch networks [18, 32] are proposed for their high efficiency, but are underexplored in CS reconstruction. In this paper, we propose a multi-grid spatial attention network, with high parallelism, to achieve both efficient training and reconstruction.

3. Reconstruction-free cascaded ACS method

In this section, we detail the proposed reconstruction-free cascaded ACS method. Specifically, we introduce the proposed reconstruction-free cascaded ACS framework in Sec. 3.1, the proposed differentiable adaptive sampling method in Sec. 3.2, and the MGSANet-based efficient reconstruction network with the designed supervision and training strategy of the overall framework in Sec. 3.3.

3.1. Reconstruction-free cascaded ACS framework

As shown in Fig. 2, we propose a multi-stage framework for realizing reconstruction-free ACS. The overall ACS framework is composed of three main parts, including the multi-stage ACS backbone, the forward adaptive sampling method, and the final image reconstruction with the input

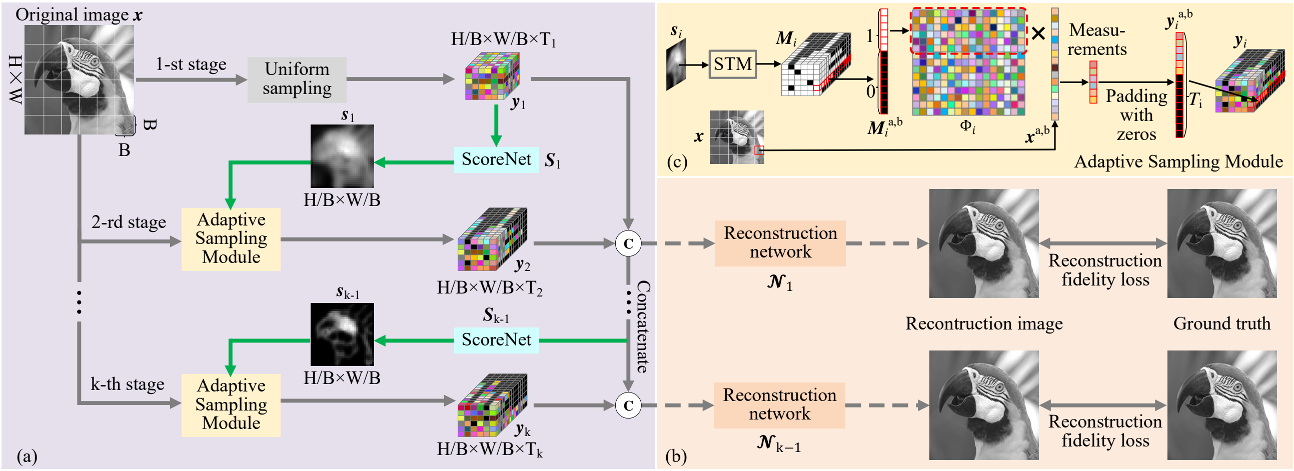

Figure 2. The overview of the proposed reconstruction-free adaptive CS method, the multi-stage adaptive sampling can be conducted without reconstruction until the target CS sampling rate is achieved. (a) The multi-stage ACS sampling process, (b) the reconstruction process and the out-of-loop reconstruction-fidelity supervision during the training process, and (c) the structure of the ASM.

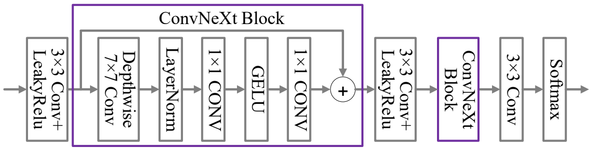

Figure 3. The network structure of ScoreNet.

of all the ACS measurements. Furthermore, the forward adaptive sampling method is composed of ScoreNet and an adaptive sampling module, which first scores the previous measurements of each block and then implements the adaptive sampling based on the score. Specifically, for the overall ACS process, the original image $\boldsymbol{x} \in \mathbb{R}^{H\times W}$ is divided into several non-overlap blocks and flattened to vector form as $\boldsymbol{x}^{a,b} \in \mathbb{R}^{B^2\times 1}$, where H and W are the height and width of the original image, B is the block size, $a \in [1, \frac{H}{B}]$ and $b \in [1, \frac{W}{B}]$ are the horizontal and vertical indexes of the block, where $a, b \in \mathbb{Z}$. In the first stage of ACS, we propose to uniformly sample each block with the same sampling rate to generate uniform measurements $\boldsymbol{y}_1$. Then, the ScoreNet $\boldsymbol{\mathcal{S}}_1$ scores the measurements of each block and outputs the scores $\boldsymbol{s}_1 \in \mathbb{R}^{\frac{H}{B}\times\frac{W}{B}}$. After that, the adaptive sampling module samples each block at different sampling rates according to the scores. Before reaching the target sampling stages, multiple loops of ACS are repeated. In each loop, image blocks are scored with the accumulated ACS measurements, i.e. $\boldsymbol{s}_{i-1} = \mathcal{S}_{i-1}(\boldsymbol{y}_1, ..., \boldsymbol{y}_{i-1})$, and the next adaptive sampling processes are conducted based upon the score, i.e., $\boldsymbol{y}_i = \mathrm{ASM}(\boldsymbol{s}_{i-1}, \Phi_i, \boldsymbol{x})$. Finally, the reconstruction network reconstructs the target image with all ACS measurements, i.e. $\boldsymbol{x}'_k = \mathcal{N}_{k-1}(\boldsymbol{y}_1, ..., \boldsymbol{y}_k)$.

To realize the optimization of the proposed ACS framework in an end-to-end way, two main challenges are required to be overcome: 1) how to design the differentiable adaptive sampling method for end-to-end training, 2) how to design the reconstruction network, supervision, and training strategy for efficient training, promoting the optimal convergence of the proposed reconstruction-free cascaded ACS framework.

3.2. LP-based differentiable adaptive sampling

To realize adaptive sampling based on previous measurements, we propose two modules, i.e., the ScoreNet module and the adaptive sampling module, which score each block with previous measurements and implement adaptive sampling based on the score. As for the ScoreNet shown in Fig. 3, to avoid introducing too heavy computation, which may hinder the practical application of the proposed ACS, we propose a lightweight architecture, which uses two ConvNeXt [51] blocks to extract the features of the scores. A Softmax layer is equipped in the final layer to ensure the sum of the output scores equals 1. Then, an adaptive sampling module is required allocate the total number of measurements according to the score. To simplify the problem, instead of making the capture-or-not decision for the measurements of a block in the next stage elementwisely, we propose to decide the number of required measurement of the block and select the first corresponding number of rows of the whole measurement matrix $\Phi_i \in \mathbb{R}^{T_i\times B^2}$ to form the real measurement matrix of the block at stage i, as shown in Fig. 2(c). T_i is a hyper-parameter constraining block measurement counts, ensuring the sampling rate does not exceed 1. Specifically, we introduce a Score To Mask (STM) module to generate a binary selection mask $\boldsymbol{M}_i$ based on $\boldsymbol{s}_i$. The measurement of each image block is thus $\boldsymbol{y}_i^{a,b} = \boldsymbol{M}_i^{a,b} \odot (\Phi_i \boldsymbol{x}^{a,b})$, where $\odot$ denotes the dot product. Through introducing an auxiliary variable

$$\boldsymbol{\eta}_i = C\left[\boldsymbol{s}_{i-1}; \boldsymbol{s}_{i-1} - 1/m_i; ...; \boldsymbol{s}_{i-1} - (T_i - 1)/m_i\right], \tag{1}$$

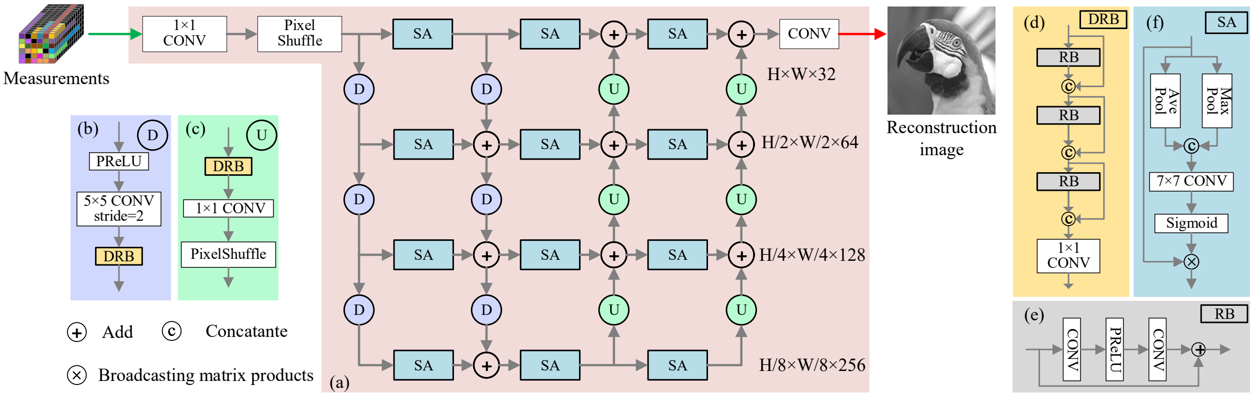

Figure 4. The overall structure of the proposed MGSANet. (a) The backbone of MGSANet, (b) upsampling module, (c) downsampling module, (d) Dense Residual Block, (e) Residual Block, and (f) Spatial Attention module.

where $C[\cdot]$ is the concatenate operation in the third dimension and $\boldsymbol{\eta}_i \in \mathbb{R}^{\frac{H}{B}\times\frac{W}{B}\times T_i}$. m_i is the total number of measurements at stage i. For a given sampling rate r_i of i-th sampling stage and an image with $H \times W$ pixels, $m_i = H \times W \times r_i$. We propose the STM module as

$$\boldsymbol{M}_i = \text{Binarize}(\boldsymbol{\eta}_i - \tau_i), \tag{2}$$

where τ_i is the m_i-th largest value in $\boldsymbol{\eta}_i$. However, Eq. (2) is non-differentiable. To address this issue, we propose to construct an integer linear programming problem with the solution equal to Eq. (2), which can be differentiated with the perturbed optimizer [5]. The constructed LP problem is

$$\begin{aligned} &\arg\max_{\boldsymbol{M}_i\in\mathcal{C}} \langle \boldsymbol{M}_i,\boldsymbol{\eta}_i\rangle, \\ s.t.\ \mathcal{C} = \{&\boldsymbol{M}_i \in \{0,1\}^{\frac{H}{B}\times\frac{W}{B}\times T_i} : \sum_{a,b,t} \boldsymbol{M}_i^{a,b,t} = m_i, \\ &M_i^{a,b,t} \geq M_i^{a,b,t+1}, \forall t \in \{1,...,T_i-1\}\}, \end{aligned} \tag{3}$$

where t indexes the third dimension of $\boldsymbol{M}_i$. $\mathcal{C}$ is the convex polytope set that meets two conditions. The first condition constrains the total number of selected measurements to be m_i, and the second condition denotes that we select the first several rows of the sampling matrix Φ_i. In the training process, the forward and backward propagation of the LP-based differentiable STM are defined below.

Forward propagation:

$$\begin{aligned} \overline{\boldsymbol{M}_i} &= \mathbb{E}_Z \left[\arg\max_{\boldsymbol{M}_i\in\mathcal{C}} \langle \boldsymbol{M}_i, \boldsymbol{\eta}_i + \sigma \boldsymbol{Z} \rangle \right], \\ &= \sum_{q=1}^{Q} \left[\text{STM}(\boldsymbol{\eta}_i + \sigma \boldsymbol{Z}_q) \right], \end{aligned} \tag{4}$$

where Q different uniform Gaussian noise $\boldsymbol{Z}_q$ is added to perturb the input $\boldsymbol{\eta}_i$ and the output is the expectation of the output of the LP module. σ and Q are hyper-parameters.

Backward propagation: The backpropagation can be achieved with the Jacobian matrix and the Jacobian of the above forward propagation can be calculated as

$$\begin{aligned} J_{\boldsymbol{s}}\boldsymbol{M}_i &= \mathbb{E}_Z \left[\arg\max_{\boldsymbol{M}_i\in\mathcal{C}} \langle \boldsymbol{M}_i, \boldsymbol{\eta}_i + \sigma \boldsymbol{Z} \rangle \boldsymbol{Z}^T/\sigma \right], \\ &= \sum_{q=1}^{Q} \left[\text{STM}\left(\boldsymbol{\eta}_i + \sigma \boldsymbol{Z}_q\right) \boldsymbol{Z}_q^T/\sigma \right]. \end{aligned} \tag{5}$$

3.3. MGSANet-based reconstruction network

For efficient training and inferencing, we propose a multi-grid spatial attention network, as shown in Fig. 4. Due to GPUs' parallel computing strength, networks with greater parallelism achieve faster acceleration than less parallel models at the same FLOPs [30]. Consequently, we propose the integration of a multi-grid structure as the fundamental backbone of our architecture. This structure facilitates the distribution of features across multiple branches, enabling parallel processing, as depicted in Fig. 4(a). Specifically, we use the downsampling module and the upsampling module to generate multi-scale features as shown in Fig. 4(b) and Fig. 4(c). A dense residual block (DRB) is incorporated for feature processing within each downsampling or upsampling module. The DRB comprises 3 residual blocks (RB) [20] with dense connections [21], as shown in Fig. 4(d) and Fig.4(e). The downsampling operation is implemented through a convolution layer with the stride set to 2, and the upsampling operation is performed using a pixel shuffle layer [40], in conjunction with a convolution layer with a 1×1 kernel.

Besides, to enhance the model's capability of focusing on the informative regions, we incorporate the spatial attention (SA) module [50] within the horizontal branches as shown in Fig. 4(f). We use 4 scales with 2× scale factor between two adjacent horizontal branches, and the sizes of features in each horizontal branch are $H \times W \times 32$, $H/2 \times W/2 \times 64$, $H/4 \times W/4 \times 96$, and $H/8 \times W/8 \times 128$ from top to bottom respectively.

Loss function. For each adaptive stage, we use l_1 loss as pixel loss to supervise the reconstructed result $\boldsymbol{x}_i'$. Besides, to reconstruct visually pleasing results, we introduce Structure Similarity Index Measure (SSIM) loss [60]. The total reconstruction fidelity loss function is

$$\begin{aligned}\mathcal{L}_i &= \mathcal{L}_i^{\text{pixel}} + \beta\mathcal{L}_i^{\text{SSIM}} \\ &= \|\boldsymbol{x}_i' - \boldsymbol{x}\|_1 + \beta(1 - \text{SSIM}(\boldsymbol{x}_i', \boldsymbol{x})),\end{aligned} \tag{6}$$

where β is the loss-balancing hyper-parameter.

Out-of-loop reconstruction-fidelity supervision. As for the training process, we propose to train the ScoreNet stage-by-stage. After finishing the training of the current stage, we fix the parameters of the current stage and train the next stage. Since there is no image reconstruction in the adaptive sampling loop, introducing sufficient supervision to promote the convergence of the proposed ACS framework is important. In our paper, we propose reconstruction fidelity supervision outside the multi-stage ACS loop to optimize the ScoreNet in the training process. As shown in Fig. 2, the reconstruction fidelity loss of each ACS stage is introduced out-of-loop to supervise and promote the convergence of the training process. Note that supervision is only introduced during the training process, and NO reconstruction is required during the adaptive sampling process.

Training strategy. Furthermore, the training process of each adaptive stage is divided into two phases: end-to-end training and fine-tuning. In the first phase, and we end-to-end train all parameters, including the sampling matrix, the ScoreNet, and the reconstruction network. Q is set to 500. $\overline{\boldsymbol{M}_i}$ is not binary but an averaged value of perturbed inputs which is different from the testing process, so in the first phase we linearly decay σ from 0.005 to 0 when training the ScoreNet to keep consistent with the testing process. Besides, to avoid overfitting, we randomly shuffle the mask $\overline{\boldsymbol{M}_i}$ in the third dimension. In the second phase, we fix the parameters of ScoreNet and fine-tune the sampling matrix and reconstruction network.

4. Experiments

4.1. Implementaion details

For the network training, we use the same training dataset with [10] which contains BSDS500 [4] train dataset and the VOC2012 [14] train dataset. We randomly crop 128 × 128 sub-image from the training dataset in the training process. We use Adam optimizer [25] to train our model with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$. The Y channel of the images in the YUV color space is utilized. The batch size is set as 32. The block size B is set to 8. We train a 5-stage model with a 5% sampling rate for each stage, so adaptive sampling rates of 10%, 15%, 20%, and 25% can be achieved with only one model. The maximum sampling rate of a block is 100% and the maximum number of measurements is evenly allocated to each adaptive stage, i.e. $T_1 = 3$, $T_2 = 15$, $T_3 = 15$, $T_4 = 16$, and $T_5 = 15$. The loss-balancing hyper-parameter β is empirically set to 0.1. For the end-to-end training phase of each adaptive stage, we train 100 epochs, the initial learning rate is set to 2×10^{-4} and multiplied by 0.8 for every 25 epochs. For the finetuning phase of each adaptive stage, we train 300 epochs, the initial learning rate is set to 2×10^{-4} and multiplied by 0.5 at the 150, 250, 280, and 290 epochs. Two commonly used test sets Set11 [26] and Urban100 [22] are adopted for evaluating the performance. We use the Peak Signal-to-Noise Ratio (PSNR) and SSIM to evaluate the quality of the reconstruction results. All the experiments are implemented on the PyTorch platform with an Intel XEON Gold 6326 CPU and an NVIDIA RTX 4090 GPU.

4.2. Comparisons with state-of-the-art CS methods

We compare our proposed model with state-of-the-art (SOTA) non-adaptive CS methods and ACS methods proposed in recently years. The non-adaptive CS methods includes AMP-Net [59], OPINE-Net$^+$ [57], COAST [53], NL-CS [10], MADUN [42], TransCS [39], FSOINet [8], CSFormer [52], TCS-Net [16] and OCTUF$^+$ [43], while the ACS methods includes ACCSNet [37], CASNet [6] and AMS-Net [58]. It is worth mentioning that AMS-Net [58] designs its adaptive sampling scheme with the ground truth image accessible, which limits its applicability in many scenarios. The quantitative comparison is summarized in Tab. 1, we can observe that our proposed model can outperform the SOTA CS methods at the sampling rates of 10%, 15%, 20% and 25%. Specifically, on the Set11 test set, our proposed model can outperform AMP-Net, OPINE-Net$^+$, COAST, NL-CS, MADUN, CASNet, TransCS, FSOINet, CSFormer, TCS-Net, OCTUF$^+$, CASNet and AMS-Net by 1.88 dB/0.026, 1.67 dB/0.0202, 1.73 dB/0.0192, 1.48 dB/0.0121, 1.09 dB/0.01, 1.65 dB/0.0183, 0.79 dB/0.83, 2.2 dB/0.0195, 2.59 dB/0.0239, 0.51 dB/0.007, 0.93 dB/0.0091 and 0.14dB/0.0191 in terms of PSNR/SSIM for average, respectively. Furthermore, we compare our proposed method with SOTA CS methods on the Urban100 test set which contains 100 more textured architectural images with high resolutions. The texture distributions are quite non-uniform in the high-resolution images, which leads to the non-uniform sampling rate of different regions required for the high-quality reconstruction. As shown in Tab. 1, benefits from the learned ScoreNet, our proposed methods can achieve different sampling rates in different regions, thus can outperform the SOTA CS methods with a large margin in the Urban100 test set. Fig. 5 shows the visual reconstruction results, we can observe that the reconstruction results of our proposed method are closer to the ground truth and have clearer texture details. In all, through comparison with the SOTA methods, we demonstrate the superiority of our proposed method both quantitatively and qualitatively.

Table 1. Performance comparison with state-of-the-art CS algorithms on Set11 [26] and Urban100 [22] test sets.

Datasets	Methods		10%	15%	20%	25%	Average
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set11	Non-adaptive sampling	AMP-Net [59]	29.40/0.8779	31.56/0.9119	33.27/0.9338	34.63/0.9481	32.22/0.9179
		OPINE-Net$^+$ [57]	29.81/0.8884	31.73/0.9176	33.32/0.9381	34.86/0.9509	32.43/0.9237
		COAST [53]	30.01/0.8963	31.99/0.9217	33.50/0.9399	33.98/0.9407	32.37/0.9247
		NL-CS [10]	30.05/0.8995	31.93/0.9268	33.52/0.9440	34.99/0.9568	32.62/0.9318
		MADUN [42]	29.96/0.8988	32.38/0.9293	34.04/0.9475	35.66/0.9601	33.01/0.9339
		TransCS [39]	29.54/0.8877	31.69/0.9189	33.49/0.9411	35.06/0.9548	32.45/0.9256
		FSOINet [8]	30.46/0.9023	32.60/0.9312	34.39/0.9492	35.80/0.9595	33.31/0.9356
		MR-CCSNet [15]	-/-	-/-	-/-	34.77/0.9546	-/-
		CSFormer [52]	29.21/0.8784	31.64/0.9181	33.34/0.9386	34.81/0.9527	31.90/0.9244
		TCS-Net [16]	29.04/0.8834	30.84/0.9139	32.20/0.9317	33.94/0.9508	31.51/0.9200
		OCTUF$^+$ [43]	30.73/0.9036	32.92/0.9332	34.61/0.9500	36.10/0.9607	33.59/0.9369
	Adaptive sampling	ACCSNet [37]	29.76/0.8847	31.86/0.9139	33.61/0.9309	-/-	-/-
		CASNet [6]	30.36/0.9014	32.47/0.9301	34.19/0.9485	35.67/0.9591	33.17/0.9348
		AMS-Net [58]	**31.23**/0.8867	33.25/0.9196	34.99/0.9406	36.35/0.9522	33.96/0.9248
		Ours	31.05/**0.9177**	**33.56/0.9420**	**35.16/0.9543**	**36.62/0.9617**	**34.10/0.9439**
Urban100	Non-adaptive sampling	AMP-Net [59]	26.04/0.8151	28.02/0.8664	29.60/0.8989	30.89/0.9202	28.64/0.8751
		OPINE-Net$^+$ [57]	26.93/0.8397	28.42/0.8784	30.06/0.9082	31.86/0.9308	29.32/0.8893
		COAST [53]	26.76/0.8414	28.67/0.8846	30.14/0.9102	31.10/0.9168	29.17/0.8882
		NL-CS [10]	27.37/0.8492	29.18/0.8909	30.50/0.9166	31.93/0.9332	29.75/0.8975
		MADUN [42]	27.00/0.8558	29.14/0.8981	30.87/0.9248	32.54/0.9347	29.89/0.9033
		TransCS [39]	26.72/0.8413	28.33/0.8818	30.07/0.9131	31.72/0.9330	29.21/0.8923
		FSOINet [8]	27.53/0.8627	29.60/0.9029	31.23/0.9268	32.62/0.9430	30.25/0.9089
		CSFormer [52]	27.92/0.8458	29.76/0.8896	31.31/0.9166	32.43/0.9332	30.36/0.8963
		TCS-Net [16]	25.86/0.8284	27.59/0.8744	28.82/0.9000	30.11/0.9236	28.10/0.8816
		OCTUF$^+$ [43]	27.92/0.8652	30.02/0.9057	31.63/0.9292	33.08/0.9453	30.66/0.9113
	Adaptive sampling	ACCSNet [37]	27.80/0.8422	29.62/0.8793	31.08/0.9009	-/-	-/-
		CASNet [6]	27.46/0.8616	29.42/0.9005	30.91/0.9237	32.20/0.9396	30.00/0.9063
		AMS-Net [58]	28.04/0.8399	30.23/0.8869	31.90/0.9147	33.23/0.9328	30.85/0.8936
		Ours	**29.09/0.8979**	**31.27/0.9254**	**32.81/0.9405**	**34.27/0.9504**	**31.86/0.9286**

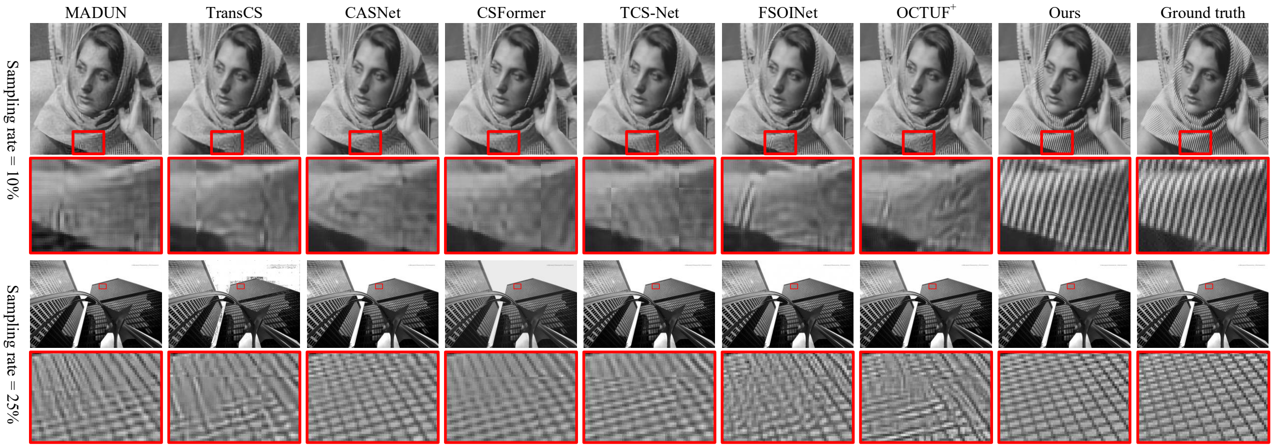

Figure 5. Visual comparison with the state-of-the-art CS algorithms. Top row: *Barbara* from Set11 [26] with sampling rate = 10%, bottom row: *img_062* from Urban100 [22] with sampling rate = 25%.

4.3. Ablation study

Adaptive sampling. To explore the effectiveness of adaptive sampling, we conduct an experiment on the model performance with and without adaptive sampling. In the case of without adaptive sampling, we uniformly sample each image block and reconstruct the image by the proposed MGSANet. As shown in Tab. 2, compared to uniform sampling, adaptive sampling can achieve more efficient sampling at the same sampling rates, resulting in significant improvements in the reconstruction results. From the visual results shown in Fig. 6, thanks to our proposed adaptive sampling method being able to allocate more samples to areas that are more difficult to reconstruct, the reconstruction results based on adaptively sampled measurements have clearer texture details.

Differentiable ASM with the perturbed optimizer. In this paper, we model the STM module in ASM as an LP problem and introduce the perturbed optimizer to make the

Table 2. Ablation experiments of adaptive sampling on Set11 [26] and Urban100 [22] datasets. The best PSNR is marked in bold.

Dataset	Sampling module	Rate			
		10%	15%	20%	25%
Set11	Uniform	30.24	32.70	34.18	35.35
	Adaptive	**31.05**	**33.56**	**35.16**	**36.62**
Urban100	Uniform	28.10	30.49	31.76	32.92
	Adaptive	**29.09**	**31.27**	**32.81**	**34.27**

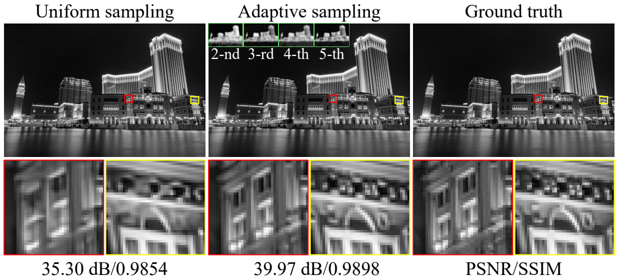

Figure 6. Ablation study of adaptive sampling at sampling rate = 25% on *img_085* from Urban100 [22]. The scores for adaptive sampling of each stage are shown in the top of the middle column.

ASM differentiable. As the differentiable ASM is a key component to train ScoreNet end-to-end, we explore the effectiveness of the differentiable ASM by training our model with and without the perturbed optimizer. As shown in Tab. 3, the model trained with the perturbed optimizer outperforms the model trained without the perturbed optimizer on Set11 [26] and Urban100 [22] test sets with large margin. The ASM without the perturbed optimizer is non-differentiable, which interrupts the entire backpropagation process, resulting in the inability to optimize the parameters of the ScoreNet. After introducing the differentiable ASM, the parameters of the ScoreNet can be optimized through backpropagation under the supervision of reconstruction fidelity. Fig. 7 shows the visualization of scores, the scores are normalized to [0, 255]. We can observe that the output scores of the ScoreNet trained without differentiable ASM are irregular, leading to limited performance in the reconstruction results. While introducing the differentiable ASM, significant improvements in the reconstruction results are achieved at the same sampling rate.

Table 3. Ablation experiments of differentiable ASM on Set11 [26] and Urban100 [22] datasets.

Datasets	Perturbed optimizer	Rate=10%	Rate=15%	Rate=20%
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set11	w/o	30.38/0.9085	32.37/0.9355	33.92/0.9505
	w/	**31.05/0.9177**	**33.56/0.9420**	**35.16/0.9543**
Urban100	w/o	28.33/0.8831	30.23/0.9165	31.62/0.9352
	w/	**29.09/0.8979**	**31.27/0.9254**	**32.81/0.9504**

Out-of-loop reconstruction fidelity supervision. In the training process, we propose to optimize the ScoreNet of each adaptive stage with the supervision of reconstruction fidelity. We conduct ablation experiments on the stage-by-stage reconstruction fidelity supervised training strategy.

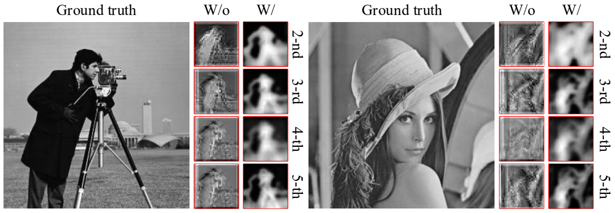

Figure 7. The visualization of scores generated by the 5-stage model trained with (w/) and without (w/o) perturbed optimizer on *cameraman* and *lena256* from Set11 testset [26]. In addition to the 1-st stage, the 2-nd to 5-th stages implement adaptive sampling based on the scores.

We train multi-stage models (3 stages for sampling rate = 15% and 5 stages for sampling rate = 25%), and the reconstruction fidelity loss is only introduced in the last stage. As shown in Tab. 4, introducing reconstruction fidelity supervision of each stage can effectively improve the quality of reconstruction results, especially for the sampling rate = 25%. When the number of stages is higher, the reconstruction quality of introducing reconstruction fidelity supervision for each stage is more significantly improved.

Table 4. Ablation experiments of the stage-by-stage reconstruction fidelity-driven training method on Set11 [26] and Urban100 [22].

Datasets	Reconstruction fidelity-driven	Rate=15%	Rate=25%
		PSNR/SSIM	PSNR/SSIM
Set11	Last stage	33.02/0.9340	35.65/0.9611
	Each stage	**33.56/0.9420**	**36.62/0.9617**
Urban100	Last stage	30.82/0.9142	33.23/0.9493
	Each stage	**31.27/0.9254**	**34.27/0.9504**

4.4. Effectiveness of MGSANet

Attention mechanism. We conduct an ablation experiment on the attention module. We replace the SA module with the channel attention (CA) [50] module and the convolutional block attention module (CBAM) [50], where CBAM is a combination of the SA module and the CA module. As shown in Tab. 5, the results of the SA-based model are slightly better than the CA-based model on the Urban100 [22] dataset, and on the Set11 [26] dataset, the two results are comparable and both better than the CBAM-based model. In addition, the computational complexity of the SA-based and the CA-based models are very close, and both are smaller than the CBAM-based model. Adding the attention module may not always lead to performance improvement, but may result in a decrease in the performance of the backbone network [23]. In summary, we adopt the SA module as the attention module.

Number of scales. We also explore the model performance with different numbers of scales. Our proposed MGSANet has 4 horizontal branches as shown in Fig. 4, and the size of features in each horizontal branch is $H \times W \times 32$, $H/2 \times W/2 \times 64$, $H/4 \times W/4 \times 96$ and $H/8 \times W/8 \times 128$ from top to bottom respectively. The models with 2 and 3

Table 5. MGSANet with different attention mudule on Set11 [26] and Urban100 [22] datasets at sampling rate = 25%.

Dataset	Attention module	FLOPs (G)	Performance	
			PSNR	SSIM
Set11	CA	**202**	35.33	**0.9612**
	CBAM	218	35.18	0.9601
	SA	**202**	**35.35**	0.9611
Urban100	CA	**1550**	32.90	0.9484
	CBAM	1672	32.57	0.9461
	SA	**1550**	**32.92**	**0.9487**

scales only contain the top 2 and 3 horizontal branches. Besides, we add a horizontal branch at the bottom to form the model with 5 scales, and the size of features in the branch is $H/16 \times W/16 \times 256$. As shown in Tab. 6, when the number of scales increases from 2 to 4, the performance of the model can be greatly improved. However, when the number of scales increases to 5, the performance of the model increases very little and even decreases on the Urban100 [22] test set. Besides, the model with 5 scales has more parameters and computational complexity, therefore, we adopt the model with 4 scales.

Table 6. Model performance of MGSANet with different number of scales on Set11 [26] and Urban100 [22] datasets. The best PSNR is marked in bold.

Datasets	Rate	Number of scales			
		2	3	4	5
Set11	15%	31.78	32.54	32.70	**32.73**
	25%	34.43	34.48	35.35	**35.36**
Urban100	15%	29.22	30.21	**30.48**	30.38
	25%	31.60	31.75	**32.91**	32.76

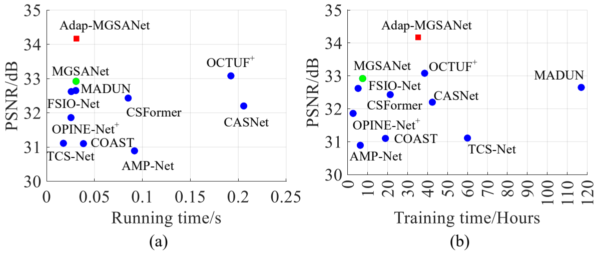

Figure 8. Comparison on performance and efficiency of various CS algorithms at sampling rate = 25% on Urban100 dataset [22].

Running speed. We compare the efficiency and reconstruction quality of our model and the SOTA models. As shown in Fig. 8(a), on the Urban100 [22] dataset, the reconstruction quality of MGSANet is slightly lower than OCTUF$^+$ [43], but the reconstruction speed of MGSANet can reach more than 6× that of OCTUF$^+$. Furthermore, the training time of MGSANet is short (less than 8 hours) as shown in Fig. 8(b), which is important for multi-stage training. Besides, as shown in Tab. 7, although MGSANet requires more FLOPs, its highly parallelized framework utilizes the potential of the GPU, enabling efficient and accurate reconstruction. In summary, our proposed MGSANet can achieve good trade-off in reconstruction quality and efficiency. Moreover, we also show the reconstruction efficiency of the MGSANet with adaptive sampling (i.e. Adap-MGSANet), we can observe that the reconstruction quality of Adap-MGSANet greatly outperforms the SOTA methods with comparable training and inferencing speed.

Table 7. Comparison of the FLOPs and running time on Urban100 dataset [22] at sampling rate = 25%.

Methods	CSFormer	FSIONet	OCTUF$^+$	CASNet	MGSANet
FLOPs (T)	0.243	0.202	0.362	0.826	1.550
Time (s)	0.0851	0.0257	0.1923	0.2057	0.0307

4.5. Sensitivity to noise

In practical scenarios, the efficacy of the model may be affected by noise. In order to evaluate the robustness of our proposed model to noise, we add Gaussian noise with different standard deviation levels, similar to [43]. We compare our proposed method with different SOTA methods at different noise levels at sampling rates = 10% and 25%. As shown in Fig. 9, our proposed method outperforms the SOTA CS methods with standard variances noise from 0 to 8 (the range of pixel values is [0, 255]).

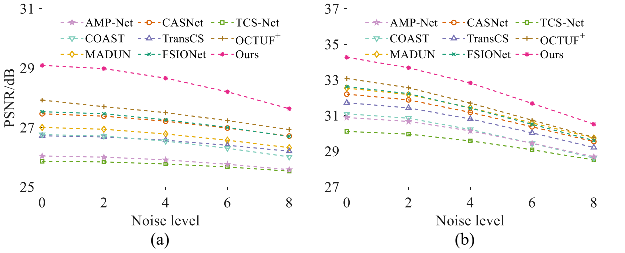

Figure 9. The results of different noise levels on Urban100 [22] dataset at sampling rate = (a) 10% and (b) 25%.

5. Conclusion

In this paper, we introduce a novel reconstruction-free cascaded Adaptive Compressive Sensing (ACS) framework, which obviates the need for reconstruction at the adaptive sampling process. A lightweight ScoreNet is proposed to allocate sampling rates based on the previous CS measurements and a differentiable adaptive sampling module is designed for end-to-end training. Furthermore, we propose a Multi-Grid Spatial-Attention Network (MGSANet) for efficient multi-stage training and reconstruction. By incorporating reconstruction fidelity supervision outside the adaptive sampling loop, we optimize ACS for high-quality imaging. Extensive quantitative and qualitative experiments demonstrate the effectiveness of our proposed method compared with state-of-the-art CS algorithms.

6. Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2022YFA1207200, and NSFC Projects under Grant 61971465.

References

[1] Marc Aβmann and Manfred Bayer. Compressive adaptive computational ghost imaging. *Scientific Reports*, 3(1):1–5, 2013. 1, 2

[2] Ali Akbari, Diana Mandache, Maria Trocan, and Bertrand Granado. Adaptive saliency-based compressive sensing image reconstruction. In *IEEE International Conference on Multimedia & Expo Workshops*, pages 1–6, 2016. 2

[3] Milad Alemohammad, Jaewook Shin, and Mark A Foster. Adaptively scanned compressive multiphoton microscopy. In *CLEO: Science and Innovations*, pages SW4J–6. Optica Publishing Group, 2018. 2

[4] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010. 5

[5] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *Advances in Neural Information Processing Systems*, 33:9508–9519, 2020. 4

[6] Bin Chen and Jian Zhang. Content-aware scalable deep compressed sensing. *IEEE Transactions on Image Processing*, 31:5412–5426, 2022. 1, 2, 5, 6

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

[8] Wenjun Chen, Chunling Yang, and Xin Yang. Fsoinet: feature-space optimization-inspired network for image compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2460–2464. IEEE, 2022. 1, 2, 5, 6

[9] Zan Chen, Wenlong Guo, Yuanjing Feng, Yongqiang Li, Changchen Zhao, Yi Ren, and Ling Shao. Deep-learned regularization and proximal operator for image compressive sensing. *IEEE Transactions on Image Processing*, 30:7112–7126, 2021. 2

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

[11] Huidong Dai, Guohua Gu, Weiji He, Ling Ye, Tianyi Mao, and Qian Chen. Adaptive compressed photon counting 3d imaging based on wavelet trees and depth map sparse representation. *Optics Express*, 24(23):26080–26096, 2016. 1, 2

[12] Weisheng Dong, Guangming Shi, Xin Li, Yi Ma, and Feng Huang. Compressive sensing via nonlocal low-rank regularization. *IEEE Transactions on Image Processing*, 23(8): 3618–3632, 2014. 2

[13] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 1

[14] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8:5, 2011. 5

[15] Zi-En Fan, Feng Lian, and Jia-Ni Quan. Global sensing and measurements reuse for image compressed sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8954–8963, 2022. 2, 6

[16] Hongping Gan, Minghe Shen, Yi Hua, Chunyan Ma, and Tao Zhang. From patch to pixel: A transformer-based hierarchical framework for compressive image sensing. *IEEE Transactions on Computational Imaging*, 9:133–146, 2023. 2, 5, 6

[17] Liang Gao, Jinyang Liang, Chiye Li, and Lihong V Wang. Single-shot compressed ultrafast photography at one hundred billion frames per second. *Nature*, 516(7529):74–77, 2014. 1

[18] Yuanbiao Gou, Peng Hu, Jiancheng Lv, Joey Tianyi Zhou, and Xi Peng. Multi-scale adaptive network for single image denoising. *Advances in Neural Information Processing Systems*, 35:14099–14112, 2022. 2

[19] Jürgen Hahn, Christian Debes, Michael Leigsnering, and Abdelhak M Zoubir. Compressive sensing and adaptive direct sampling in hyperspectral imaging. *Digital Signal Processing*, 26:113–126, 2014. 1, 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 4

[22] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 5, 6, 7, 8

[23] Zhongzhan Huang, Senwei Liang, Mingfu Liang, Wei He, Haizhao Yang, and Liang Lin. The lottery ticket hypothesis for self-attention in convolutional neural network. *arXiv preprint arXiv:2207.07858*, 2022. 7

[24] Hongzhi Jiang, Shuguang Zhu, Huijie Zhao, Bingjie Xu, and Xudong Li. Adaptive regional single-pixel imaging based on the fourier slice theorem. *Optics Express*, 25(13):15118–15130, 2017. 2

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[26] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 449–458, 2016. 1, 2, 5, 6, 7, 8

[27] Chengbo Li, Hong Jiang, Paul Wilford, Yin Zhang, and Mike Scheutzow. A new compressive video sensing framework for mobile broadcast. *IEEE Transactions on Broadcasting*, 59 (1):197–205, 2013. 1

[28] Jiying Liu and Cong Ling. Adaptive compressed sensing using intra-scale variable density sampling. *IEEE Sensors Journal*, 18(2):547–558, 2017. 2

[29] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr

imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007. 1

[30] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision*, pages 116–131, 2018. 2, 4

[31] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 2

[32] Yiqun Mei, Yuchen Fan, Yulun Zhang, Jiahui Yu, Yuqian Zhou, Ding Liu, Yun Fu, Thomas S Huang, and Humphrey Shi. Pyramid attention network for image restoration. *International Journal of Computer Vision*, 131(12):3207–3225, 2023. 2

[33] R Monika and Samiappan Dhanalakshmi. An efficient medical image compression technique for telemedicine systems. *Biomedical Signal Processing and Control*, 80: 104404, 2023. 1, 2

[34] David B Phillips, Ming-Jie Sun, Jonathan M Taylor, Matthew P Edgar, Stephen M Barnett, Graham M Gibson, and Miles J Padgett. Adaptive foveated single-pixel imaging with dynamic supersampling. *Science Advances*, 3(4): e1601782, 2017. 1

[35] Yan Qian, Ruiqing He, Qian Chen, Guohua Gu, Feng Shi, and Wenwen Zhang. Adaptive compressed 3d ghost imaging based on the variation of surface normals. *Optics Express*, 27 (20):27862–27872, 2019. 1, 2

[36] Chenxi Qiu and Xuemei Hu. Adacs: Adaptive compressive sensing with restricted isometry property-based error-clamping. *IEEE TPAMI*, pages 1–18, 2024. 2

[37] Chenxi Qiu, Tao Yue, and Xuemei Hu. Adaptive and cascaded compressive sensing. *arXiv:2203.10779*, 2022. 2, 5, 6

[38] Florian Rousset, Nicolas Ducros, Andrea Farina, Gianluca Valentini, Cosimo D'Andrea, and Françoise Peyrin. Adaptive basis scan by wavelet prediction for single-pixel imaging. *IEEE Transactions on Computational Imaging*, 3(1): 36–46, 2016. 1, 2

[39] Minghe Shen, Hongping Gan, Chao Ning, Yi Hua, and Tao Zhang. Transcs: a transformer-based hybrid architecture for image compressed sensing. *IEEE Transactions on Image Processing*, 31:6991–7005, 2022. 2, 5, 6

[40] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. 4

[41] Fernando Soldevila, Eva Salvador-Balaguer, P Clemente, Enrique Tajahuerce, and Jesús Lancis. High-resolution adaptive imaging with a single photodiode. *Scientific Reports*, 5 (1):1–9, 2015. 2

[42] Jiechong Song, Bin Chen, and Jian Zhang. Memory-augmented deep unfolding network for compressive sensing. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4249–4258, 2021. 2, 5, 6

[43] Jiechong Song, Chong Mou, Shiqi Wang, Siwei Ma, and Jian Zhang. Optimization-inspired cross-attention transformer for compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6174–6184, 2023. 1, 2, 5, 6, 8

[44] Rayko I Stantchev, David B Phillips, Peter Hobson, Samuel M Hornett, Miles J Padgett, and Euan Hendry. Compressed sensing with near-field thz radiation. *Optica*, 4(8): 989–992, 2017. 1, 2

[45] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[47] Lishun Wang, Miao Cao, Yong Zhong, and Xin Yuan. Spatial-temporal transformer for video snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[48] Ping Wang, Lishun Wang, and Xin Yuan. Deep optics for video snapshot compressive imaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10646–10656, 2023. 1

[49] Qiong Wang, Qiurong Yan, Suhui Deng, Hui Wang, Chenglong Yuan, and Yuhao Wang. Iterative adaptive photon-counting compressive imaging based on wavelet entropy automatic threshold acquisition. *IEEE Photonics Journal*, 11 (5):1–13, 2019. 1, 2

[50] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision*, pages 3–19, 2018. 4, 7

[51] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023. 3

[52] Dongjie Ye, Zhangkai Ni, Hanli Wang, Jian Zhang, Shiqi Wang, and Sam Kwong. Csformer: Bridging convolution and transformer for compressive sensing. *IEEE Transactions on Image Processing*, 2023. 1, 2, 5, 6

[53] Di You, Jian Zhang, Jingfen Xie, Bin Chen, and Siwei Ma. Coast: Controllable arbitrary-sampling network for compressive sensing. *IEEE Transactions on Image Processing*, 30:6066–6080, 2021. 2, 5, 6

[54] Wen-Kai Yu, Ming-Fei Li, Xu-Ri Yao, Xue-Feng Liu, Ling-An Wu, and Guang-Jie Zhai. Adaptive compressive ghost imaging based on wavelet trees and sparse representation. *Optics Express*, 22(6):7133–7144, 2014. 2

[55] Ying Yu, Bin Wang, and Liming Zhang. Saliency-based compressive sampling for image signals. *IEEE Signal Processing Letters*, 17(11):973–976, 2010. 2

[56] Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive

sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1828–1837, 2018. 1, 2

[57] Jian Zhang, Chen Zhao, and Wen Gao. Optimization-inspired compact deep compressive sensing. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):765–774, 2020. 2, 5, 6

[58] Kuiyuan Zhang, Zhongyun Hua, Yuanman Li, Yongyong Chen, and Yicong Zhou. Ams-net: Adaptive multi-scale network for image compressive sensing. *IEEE Transactions on Multimedia*, 2022. 5, 6

[59] Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. Amp-net: Denoising-based deep unfolding for compressive image sensing. *IEEE Transactions on Image Processing*, 30:1487–1500, 2020. 1, 2, 5, 6

[60] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2016. 5