Degradation-Aware Unfolding Half-Shuffle Transformer for Spectral Compressive Imaging

Anonymous NeurIPS supplementary submission

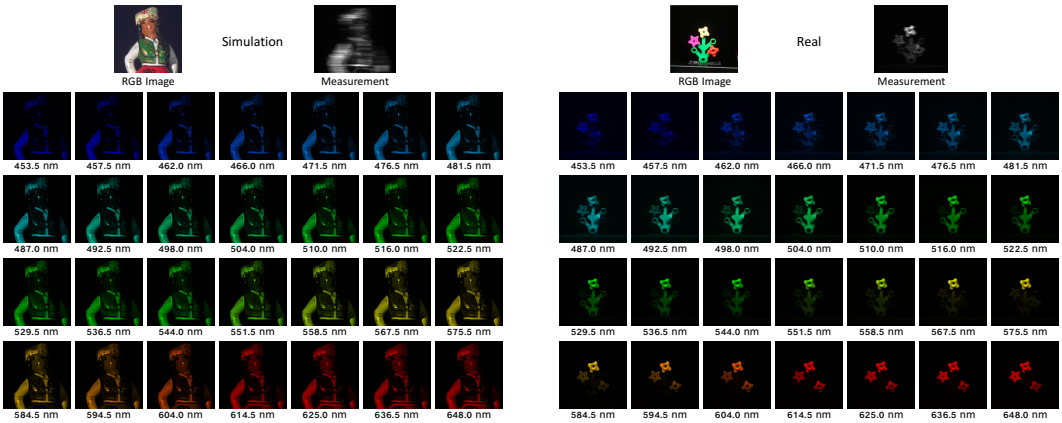

Figure 1: Reconstructed simulation (left) and real (right) spectral images with 28 wavelengths by our DAUHST.

All source codes and pre-trained models will be made publicly available for further research.

In this supplementary material, we share more details that are not in our main paper, including:

(a) Mathematical model of CASSI in Sec. 1

(b) Computational complexity comparisons with other Multi-head Self-Attention (MSA) in Sec. 2

(c) More qualitative comparisons with state-of-the-art (SOTA) methods in Sec. 3

(d) More ablation studies of stage number and mask modulation flexibility in Sec. 4

(e) Limitation of our work in Sec. 5

(f) Broader impact in Sec. 6

(g) Code submission and reproducibility in Sec. 7

1 Mathematical Model of CASSI

We denote the 3D HSI cube (Fig. 2 left) as $\mathbf{X} \in \mathbb{R}^{H\times W\times N_\lambda}$, where H, W, and N_λ represent the HSI's height, width, and total number of wavelengths. Then the mask modulation is conducted as

$$\mathbf{X}'(:,:,n_\lambda) = \mathbf{X}(:,:,n_\lambda) \odot \mathbf{M}^*, \tag{1}$$

where $\mathbf{X}' \in \mathbb{R}^{H\times W\times N_\lambda}$ denotes the modulated signal, $\mathbf{M}^* \in \mathbb{R}^{H\times W}$ denotes a pre-defined coded aperture (physical mask), $n_\lambda \in [1,\dots,N_\lambda]$ indexes the spectral channels, and $\odot$ is the inner product.

After passing through a disperser, the 3D cube $\mathbf{X}'$ becomes tilted and could be considered as sheared along the y-axis. Define $\mathbf{X}'' \in \mathbb{R}^{H\times(W+d(N_\lambda-1))\times N_\lambda}$ as the tilted cube, and λ_c as the reference wavelength, *i.e.*, $\mathbf{X}'[:,:,n_{\lambda_c}]$ is not sheared along the y-axis. Then the dispersion is formulated as

$$\mathbf{X}''(u,v,n_\lambda) = \mathbf{X}'(x, y+d(\lambda_n-\lambda_c), n_\lambda), \tag{2}$$

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

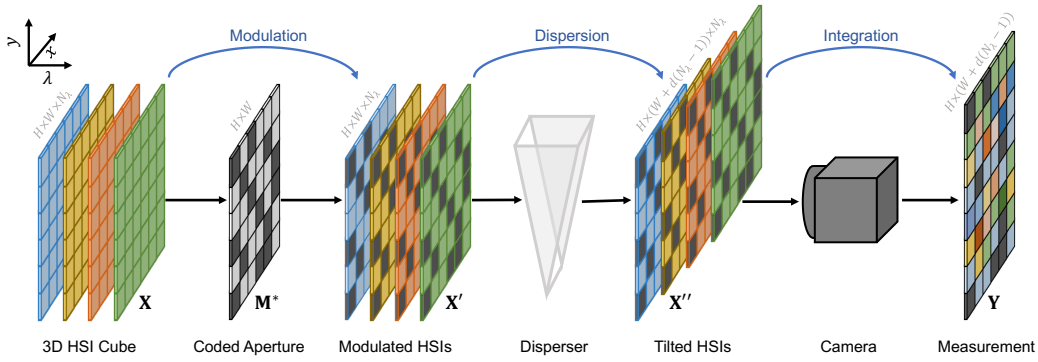

Figure 2: Illustration of a Single Disperser Coded Aperture Snapshot Spectral Imaging (SD-CASSI) system.

where (u, v) indicates the coordinate system on the detector plane, λ_n denotes the wavelength of the n_λ-th spectral channel, d represents the shifting step, and $d(\lambda_n - \lambda_c)$ signifies the spatial shifting for the n_λ-th channel on $\mathbf{X}'$. Since the sensor integrates all the light within the wavelength range $[\lambda_{\min}, \lambda_{\max}]$, the compressed measurement at the detector $y(u, v)$ can be modelled as

$$y(u, v) = \int_{\lambda_{\min}}^{\lambda_{\max}} x''(u, v, n_\lambda) \mathrm{d}\lambda, \tag{3}$$

where x'' denotes the continuous representation of $\mathbf{X}''$. Then we discretize Eq. (3) as

$$\mathbf{Y} = \sum_{n_\lambda=1}^{N_\lambda} \mathbf{X}''(:, :, n_\lambda) + \mathbf{N}, \tag{4}$$

where $\mathbf{Y} \in \mathbb{R}^{H \times (W + d(N_\lambda - 1))}$ denotes the 2D compressed measurement that captures the information and $\mathbf{N} \in \mathbb{R}^{H \times (W + d(N_\lambda - 1))}$ represents the imaging noise generated by the detector.

To simplify the notations, we define $\mathbf{M} \in \mathbb{R}^{H \times (W + d(N_\lambda - 1)) \times N_\lambda}$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{H \times (W + d(N_\lambda - 1)) \times N_\lambda}$ as the shifted version of the mask $\mathbf{M}^*$ and original HSI signal $\mathbf{X}$ corresponding to different wavelengths:

$$\begin{aligned} \mathbf{M}(u, v, n_\lambda) &= \mathbf{M}^*(x, y + d(\lambda_n - \lambda_c)), \\ \tilde{\mathbf{X}}(u, v, n_\lambda) &= \mathbf{X}(x, y + d(\lambda_n - \lambda_c), n_\lambda). \end{aligned} \tag{5}$$

Subsequently, $\mathbf{Y}$ in Eq. (4) can be reformulated as

$$\mathbf{Y} = \sum_{n_\lambda=1}^{N_\lambda} \tilde{\mathbf{X}}(:, :, n_\lambda) \odot \mathbf{M}(:, :, n_\lambda) + \mathbf{N}. \tag{6}$$

Vectorization. We define $\mathbf{y} = \mathrm{vec}(\mathbf{Y})$ and $\mathbf{n} = \mathrm{vec}(\mathbf{N}) \in \mathbb{R}^n$ as the vectorization of matrices $\mathbf{Y}$ and $\mathbf{N}$, where $\mathrm{vec}(\cdot)$ concatenates all the columns of a matrix as one single vector and $n = H(W + d(N_\lambda - 1))$. Similarly, we have $\tilde{\mathbf{x}}^{(n_\lambda)} = \mathrm{vec}(\tilde{\mathbf{X}}(:, :, n_\lambda))$, resulting in the vector $\mathbf{x} = \mathrm{vec}([\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(N_\lambda)}]) \in \mathbb{R}^{nN_\lambda}$. We denote the sensing matrix as

$$\mathbf{\Phi} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_{N_\lambda}] \in \mathbb{R}^{n \times nN_\lambda}, \tag{7}$$

where $\boldsymbol{D}_{n_\lambda} = \mathrm{diag}(\mathrm{vec}(\mathbf{M}(:, :, n_\lambda)))$ is a diagonal matrix with $\mathrm{vec}(\mathbf{M}(:, :, n_\lambda))$ as the diagonal elements. As such, Eq. (6) can be reformulated in a vectorized version as

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}. \tag{8}$$

Eq. (8) is similar to the compressive sensing [1, 2] as $\mathbf{\Phi}$ is a fat matrix, *i.e.*, more colums than rows. However, since $\mathbf{\Phi}$ has the very special structure as in Eq. (7), most existing compressive sensing theories can not fit in our applications. Note that $\mathbf{\Phi}$ is highly sparse with at most nN_λ nonzero elements. Yet, it has been proved that the signal can still be reconstructed even when $N_\lambda > 1$ [3, 4].

Given the compressed measurement $\mathbf{y}$ captured by the camera and sensing matrix $\mathbf{\Phi}$ calibrated based on pre-design, one practical task of CASSI is to solve $\mathbf{x}$, which is also the topic of our work.

2 Computational Complexity Comparisons of Different MSA Modules

In this section, we compare the computational complexity of global MSA (G-MSA) [5], window-based MSA (W-MSA) [6] and our HS-MSA. Since the computational complexity of position embedding is negligible, we omit it for brevity and focus on comparing the self-attention calculation scheme. We denote the input tokens as $\mathbf{X}_{in} \in \mathbb{R}^{H \times \hat{W} \times C}$, where $\hat{W} = W + d(N_\lambda - 1)$. Subsequently, $\mathbf{X}_{in}$ is linearly projected into $query$ $\mathbf{Q} \in \mathbb{R}^{H \times \hat{W} \times C}$, key $\mathbf{K} \in \mathbb{R}^{H \times \hat{W} \times C}$, and $value$ $\mathbf{V} \in \mathbb{R}^{H \times \hat{W} \times C}$ as

$$\mathbf{Q} = \mathbf{X}_{in}\mathbf{W}^{\mathbf{Q}},\ \mathbf{K} = \mathbf{X}_{in}\mathbf{W}^{\mathbf{K}},\ \mathbf{V} = \mathbf{X}_{in}\mathbf{W}^{\mathbf{V}}, \tag{9}$$

where $\mathbf{W}^{\mathbf{Q}}, \mathbf{W}^{\mathbf{K}}, \mathbf{W}^{\mathbf{V}} \in \mathbb{R}^{C \times C}$ are learnable parameters and biases are omitted for simplification.

2.1 Global Multi-head Self-Attention

For G-MSA, $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are split along the channel dimension into N heads: $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^N], \mathbf{K} = [\mathbf{K}^1, \ldots, \mathbf{K}^N], \mathbf{V} = [\mathbf{V}^1, \ldots, \mathbf{V}^N]$. The dimension of each head is $d_h = \frac{C}{N}$. Then G-MSA samples all the tokens as key and $query$ to calculate the self-attention $\mathbf{A}_g^i$ inside each head as

$$\mathbf{A}_g^i = \text{softmax}(\frac{\mathbf{Q}^i {\mathbf{K}^i}^{\mathsf{T}}}{\sqrt{d_h}})\mathbf{V}^i, \quad i = 1, \ldots, N. \tag{10}$$

Subsequently, the outputs of N heads are concatenated along the spectral dimension and then undergo a linear projection to generate the output feature map $\mathbf{X}_{out} \in \mathbb{R}^{H \times \hat{W} \times C}$ as

$$\mathbf{X}_{out} = \sum\nolimits_{i=1}^{N} \mathbf{A}_g^i \mathbf{W}_g^i, \tag{11}$$

where $\mathbf{W}_g^i \in \mathbb{R}^{d_h \times C}$ are learnable parameters. The computational complexity of G-MSA is

$$O(\text{G-MSA}) = 4H\hat{W}C^2 + 2(H\hat{W})^2 C, \tag{12}$$

where the first term comes from the linear projection in Eq. (9) and head merging in Eq. (11), the second term comes from the calculation of the self-similarity and content aggregation in Eq. (10).

2.2 Window-based Multi-head Self-Attention

W-MSA firstly partitions $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ into non-overlapping windows with size $M \times M$ and reshapes them into $\mathbf{Q}_w, \mathbf{K}_w, \mathbf{V}_w \in \mathbb{R}^{\frac{H\hat{W}}{M^2} \times M^2 \times C}$. Subsequently, $\mathbf{Q}_w, \mathbf{K}_w, \mathbf{V}_w$ are split along the channel dimension into N heads: $\mathbf{Q}_w = [\mathbf{Q}_w^1, \ldots, \mathbf{Q}_w^N], \mathbf{K}_w = [\mathbf{K}_w^1, \ldots, \mathbf{K}_w^N], \mathbf{V}_w = [\mathbf{V}_w^1, \ldots, \mathbf{V}_w^N]$. Then W-MSA samples the tokens inside each window to calculate the self-attention $\mathbf{A}_w^i$ in each head:

$$\mathbf{A}_w^i = \text{softmax}(\frac{\mathbf{Q}_w^i {\mathbf{K}_w^i}^{\mathsf{T}}}{\sqrt{d_h}})\mathbf{V}_w^i, \quad i = 1, \ldots, N. \tag{13}$$

Finally, the results of N heads are aggregated to generate the output feature $\mathbf{X}_{out} \in \mathbb{R}^{H \times \hat{W} \times C}$ as

$$\mathbf{X}_{out} = \sum\nolimits_{i=1}^{N} \mathbf{A}_w^i \mathbf{W}_w^i, \tag{14}$$

where $\mathbf{W}_w^i \in \mathbb{R}^{d_h \times C}$ are learnable parameters. The computational complexity of W-MSA is

$$O(\text{W-MSA}) = 4H\hat{W}C^2 + 2M^2 H\hat{W}C, \tag{15}$$

where the first term comes from the linear projection in Eq. (9) and head merging in Eq. (14), the second term comes from the calculation of the self-similarity and content aggregation in Eq. (13).

2.3 Half-Shuffle Multi-head Self-Attention

HS-MSA firstly splits $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ into two equal parts along the channel dimension as

$$\mathbf{Q} = [\mathbf{Q}_l, \mathbf{Q}_{nl}], \quad \mathbf{K} = [\mathbf{K}_l, \mathbf{K}_{nl}], \quad \mathbf{V} = [\mathbf{V}_l, \mathbf{V}_{nl}], \tag{16}$$

where $\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l \in \mathbb{R}^{H\times\hat{W}\times\frac{C}{2}}$ are fed into the *local branch* to capture local contents, while $\mathbf{Q}_{nl}, \mathbf{K}_{nl}, \mathbf{V}_{nl} \in \mathbb{R}^{H\times\hat{W}\times\frac{C}{2}}$ pass through the *non-local branch* to model non-local dependencies.

Local Branch. The *local branch* computes MSA within position-specific windows. More specifically, $\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l$ are divided into non-overlapping windows with size $M \times M$. Then they are reshaped into $\mathbb{R}^{\frac{H\hat{W}}{M^2}\times M^2\times\frac{C}{2}}$. Subsequently, $\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l$ are split into $h = \frac{N}{2}$ heads along the channel wise: $\mathbf{Q}_l = [\ \mathbf{Q}_l^1, \ldots, \mathbf{Q}_l^h\ ]$, $\mathbf{K}_l = [\ \mathbf{K}_l^1, \ldots, \mathbf{K}_l^h\ ]$, $\mathbf{V}_l = [\ \mathbf{V}_l^1, \ldots, \mathbf{V}_l^h\ ]$. The dimension of each head is $d_h = \frac{C}{2h}$. Then the local self-attention $\mathbf{A}_l^i$ is calculated inside each head as

$$\mathbf{A}_l^i = \text{softmax}(\frac{\mathbf{Q}_l^i\ {\mathbf{K}_l^i}^\mathsf{T}}{\sqrt{d_h}})\ \mathbf{V}_l^i, \quad i = 1, \ldots, h. \tag{17}$$

Non-local Branch. The *non-local branch* computes cross-window interactions through shuffle operations inspired by ShuffleNet [7]. In particular, $\mathbf{Q}_{nl}, \mathbf{K}_{nl}, \mathbf{V}_{nl} \in \mathbb{R}^{H\times\hat{W}\times\frac{C}{2}}$ are firstly divided into non-overlapping windows with size $M \times M$. Then their shapes are transposed from $\mathbb{R}^{\frac{H\hat{W}}{M^2}\times M^2\times\frac{C}{2}}$ to $\mathbb{R}^{M^2\times\frac{H\hat{W}}{M^2}\times\frac{C}{2}}$ to shuffle the positions of tokens and establish inter-window dependencies. Subsequently, $\mathbf{Q}_{nl}, \mathbf{K}_{nl}, \mathbf{V}_{nl}$ are also split into h heads: $\mathbf{Q}_{nl} = [\mathbf{Q}_{nl}^1, \ldots, \mathbf{Q}_{nl}^h], \mathbf{K}_{nl} = [\mathbf{K}_{nl}^1, \ldots, \mathbf{K}_{nl}^h], \mathbf{V}_{nl} = [\mathbf{V}_{nl}^1, \ldots, \mathbf{V}_{nl}^h]$. Then the non-local self-attention $\mathbf{A}_{nl}^i$ is computed as

$$\mathbf{A}_{nl}^i = \text{softmax}(\frac{\mathbf{Q}_{nl}^i\ {\mathbf{K}_{nl}^i}^\mathsf{T}}{\sqrt{d_h}})\ \mathbf{V}_{nl}^i, \quad i = 1, \ldots, h. \tag{18}$$

Subsequently, $\mathbf{A}_{nl}^i \in \mathbb{R}^{M^2\times\frac{H\hat{W}}{M^2}\times d_h}$ is unshuffled by being transposed to shape $\mathbb{R}^{\frac{H\hat{W}}{M^2}\times M^2\times d_h}$. Then the outputs of the *local branch* and *non-local branch* are aggregated by a linear projection as

$$\mathbf{X}_{out} = \sum\nolimits_{i=1}^{h} \mathbf{A}_l^i \mathbf{W}_l^i + \sum\nolimits_{i=1}^{h} \mathbf{A}_{nl}^i \mathbf{W}_{nl}^i, \tag{19}$$

where $\mathbf{W}_l^i$, $\mathbf{W}_{nl}^i \in \mathbb{R}^{d_h\times C}$ are learnable parameters. The computational complexity of HS-MSA is

$$O(\text{HS-MSA}) = 4H\hat{W}C^2 + M^2H\hat{W}C + \frac{H^2\hat{W}^2}{M^2}C, \tag{20}$$

where the first term comes from linear projection in Eq. (9) and head merging in Eq. (19), the second and third terms come from self-similarity calculation and content aggregation in Eq. (17) and Eq. (18).

Discussion. W-MSA suffers from limited receptive fields within position-specific windows. In contrast, our HS-MSA enjoys global receptive fields and can capture long-range dependencies. However, instead of globally sampling all tokens like global MSA, HS-MSA builds the inter-window correlations by shuffle operations. The self-attention is still calculated in the local window but with tokens from non-local regions. Thus, the computational cost of HS-MSA is much cheaper than that of global MSA. In implementation, $H = W = 256, d = 2, C = N_\lambda = 28, M = 8$. Compared with global MSA, HS-MSA only requires 0.89% computational cost, showing its efficiency advantage.

3 More Qualitative Comparisons with State-of-the-Art Methods

3.1 Simulation HSI Reconstruction

All Spectral Channels. Fig. 1 (left) shows the reconstructed simulation images of *Scene* 6 with 28 spectral channels. DAUHST-9stg successfully recovers the desired HSIs of *Scene* 6 at all wavelengths.

Comparisons with SOTA methods. Fig. 3 depicts the reconstructed simulation HSI comparisons of *Scene* 5, 7, and 8 with 4 out of 28 spectral channels. Nine SOTA algorithms and DAUHST-9stg are included. Please zoom in for a better view. As can be seen from the reconstructed HSIs (bottom) and the zoomed-in patches (top-right) of the selected yellow boxes that other competitors fail to restore high-frequency HSI contents. They are favorable to yield over-smooth results sacrificing fine-grained details and structural textures, or introducing unpleasant artifacts. In contrast, our DAUHST-9stg is more effective in producing perceptually-pleasing and sharp images, and maintaining the spatial smoothness of the homogeneous regions without introducing artifacts. Additionally, we plot the spectral density curves (top-medium) corresponding to the picked regions of the green boxes in the RGB image (top-left). The highest correlation and coincidence between our curves and the ground-truth curves demonstrate the spectral-wise consistency reconstruction effectiveness of DAUHST.

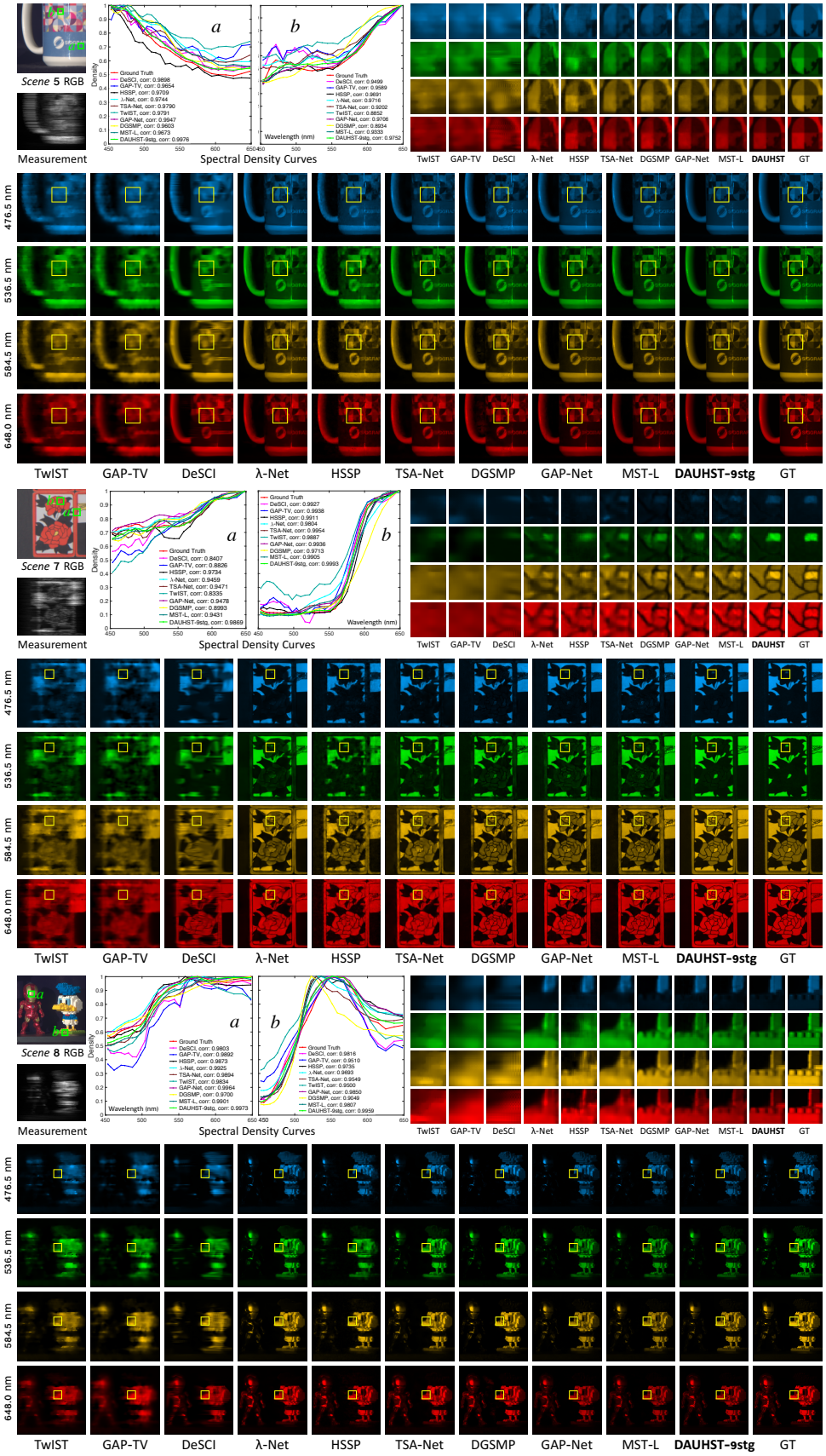

Figure 3: Reconstructed simulation HSIs of *Scene* 5, 7, and 8 with 4 out of 28 spectral channels. Nine SOTA methods and our DAUHST are included. The spectral curves correspond to the green boxes of the RGB image.

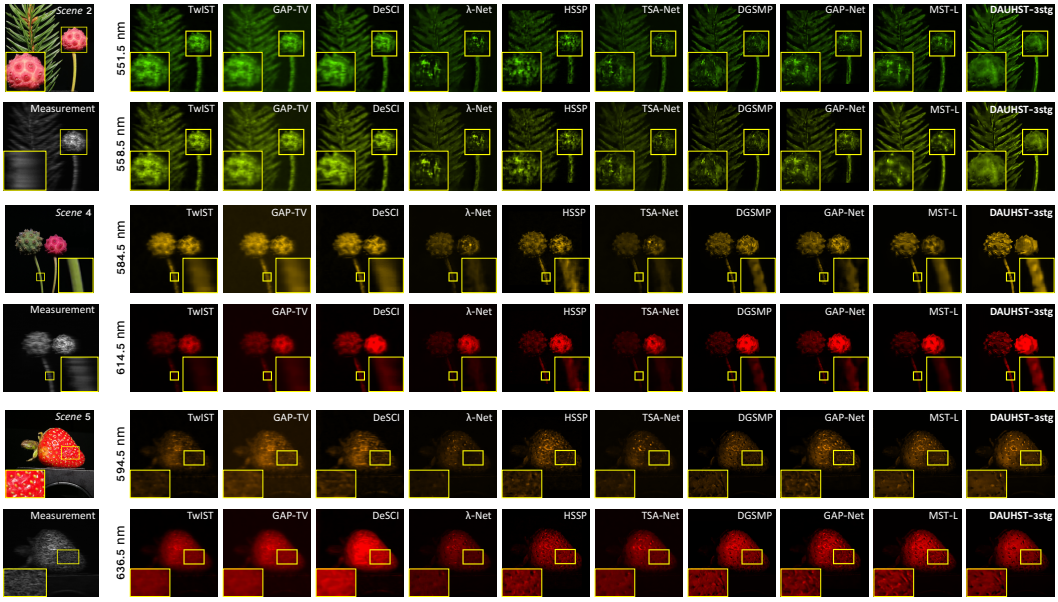

Figure 4: Reconstructed real HSI comparisons on *Scene* 2 (top), 4 (middle), and 5 (bottom) with 2 out of 28 spectral channels. Nine SOTA methods and DAUHST-3stg are included. Our DAUHST-3stg is superior to other SOTA methods in detailed content reconstruction and real noise suppression. Please zoom in for a better view.

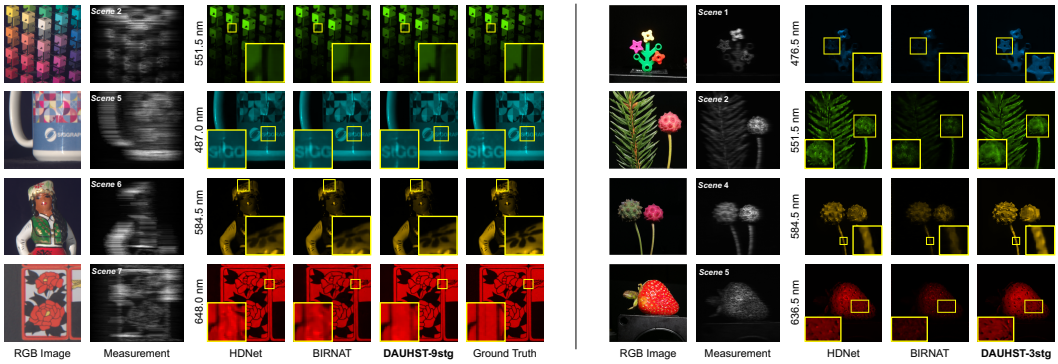

Figure 5: Qualitative comparisons of HDNet, BIRNAT, and our DAUHST on simulation (left) *Scene* 2, 5, 6, 7 and real (right) *Scene* 1, 2, 4, 5. Our DAUHST yields more visually pleasant results. Zoom in for a better view.

3.2 Real HSI Reconstruction

All Spectral Channels. Fig. 1 (right) depicts 28 spectral channels of the reconstructed real HSIs on *Scene* 1. Our DAUHST-3stg reliably reconstructs all the spectral channels of the desired HSI signal.

Comparisons with SOTA methods. Fig. 4 shows the reconstructed real HSI comparisons of *Scene* 2 (top), 4 (middle), and 5 (bottom) with 2 out of 28 spectral channels. Our DAUHST-3stg is superior to other methods in fine-grained content reconstruction, spectral density responses, and real noise suppression. These results suggest the robustness and generalization ability of the proposed DAUHST.

3.3 Visual Comparisons with HDNet and BIRNAT

Bearing the space constraints and resolution of figures in concern, we provide the qualitative comparisons of HDNet [8], BIRNAT [9], and our DAUHST in Fig. 5. It can be clearly observed that our DAUHST achieves more visually pleasant results on all simulation (left) and real (right) scenes.

Stage	1	3	5	7	9
PSNR	34.36	37.21	37.75	38.20	38.36
SSIM	0.932	0.959	0.962	0.968	0.967
Params (M)	0.73	2.08	3.44	4.79	6.15
FLOPS (G)	9.72	27.17	44.61	62.05	79.50

(a) Ablation of the stage number.

Method	TSA-Net [10]	DGSMP [11]	**DAUHST-3stg**
Mask-0	31.46 ↓ 0.00 %	32.63 ↓ 00.00 %	37.21 ↓ 0.00 %
Mask-1	29.18 ↓ 7.24 %	28.50 ↓ 12.66 %	36.43 ↓ 2.10 %
Mask-2	29.10 ↓ 7.50 %	27.87 ↓ 14.59 %	36.55 ↓ 1.77 %
Mask-3	29.01 ↓ 7.79 %	27.91 ↓ 14.47 %	36.38 ↓ 2.23 %

(b) Ablation of mask modulation flexibility.

Table 1: Ablation studies on simulation datasets [12, 13]. PSNR, SSIM, Params, and FLOPS are reported.

4 More Ablation Studies

4.1 Number of Stages

We conduct ablation to study how the performance and costs of DAUHST change with the stage number in Tab. 1a. The performance improves when we gradually increase the stage number. We notice that a 3-stage DAUHST can achieve a very impressive PSNR result of 37.21 dB.

4.2 Mask Modulation Flexibility

We change the mask by randomly cropping it with size 256×256 from the real mask of size 660×660 to evaluate the flexibility of DAUHST for different signal modulations. The results are reported in Tab. 1b, where 'Mask-0' indicates the original mask used in training. Compared with the two SOTA methods TSA-Net (↓ 7.51% on average) and DGSMP (↓ 13.91%), our DAUHST-3stg declines by much smaller margins (↓ 2.03%) when the mask changes. These results suggest that DAUHST is more robust and flexible for large-scale SCI reconstruction.

5 Limitation

The main limitation of our work is that the performance improvement of our method comes with lowering down the inference speed and increasing the model complexity. Specifically, the Params, FLOPS, and depth of network increase with the stage number of DAUHST. For instance, compared with DAUHST-1stg, DAUHST-9stg achieves 4.00 dB improvement but requires 8.18× FLOPS, 8.42× Params, and 4.04× inference time. To tackle this limitation, we will study how to improve the restoration performance without increasing the model complexity and sacrificing the inference speed.

6 Broader Impact

HSI reconstruction is one of the core tasks in snapshot compressive imaging (SCI) and has been studied for decades. Compared with normal RGB images, HSIs have more spectral bands to store richer information of the desired scenes. Hence, HSIs are widely applied in many computer vision related tasks, such as medical imaging [14, 15, 16], object tracking [17, 18, 19], remote sensing [20, 21, 22], and so on. Nowadays, billions of 3D HSIs are compressed by SCI systems. Therefore, how to reconstruct the original 3D HSI signal from the 2D compressed measurement is worth studying. Our algorithm, DAUHST, is capable of reconstructing HSIs more efficiently and accurately than all existing SOTA methods.

Until now, HSI reconstruction techniques have no negative social impact yet. Our proposed DAUHST does not present any negative foreseeable societal consequence, either.

7 Code Submission and Reproducibility

We provide the **source code** and **pre-trained** models to reproduce the main results in Tab. 1 of our paper. Please refer to the folder 'code' and read the file 'README.md' for detailed instructions. **All the source codes and pre-trained models will be released to the public for further research.**

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, 2006.

[2] C. Emmanuel, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, 2006.

[3] S. Jalali and X. Yuan, "Compressive imaging via one-shot measurements," in *IEEE International Symposium on Information Theory (ISIT)*, 2018.

[4] S. Jalali and X. Yuan, "Snapshot compressed sensing: Performance bounds and algorithms," *IEEE Transactions on Information Theory*, 2019.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.

[7] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *CVPR*, 2018.

[8] X. Hu, Y. Cai, J. Lin, H. Wang, X. Yuan, Y. Zhang, R. Timofte, and L. V. Gool, "Hdnet: High-resolution dual-domain learning for spectral compressive imaging," in *CVPR*, 2022.

[9] Z. Cheng, B. Chen, R. Lu, Z. Wang, H. Zhang, Z. Meng, and X. Yuan, "Recurrent neural networks for snapshot compressive imaging," *TPAMI*, 2022.

[10] Z. Meng, J. Ma, and X. Yuan, "End-to-end low cost compressive spectral imaging with spatial-spectral self-attention," in *ECCV*, 2020.

[11] T. Huang, W. Dong, X. Yuan, J. Wu, and G. Shi, "Deep gaussian scale mixture prior for spectral compressive imaging," in *CVPR*, 2021.

[12] J.-I. Park, M.-H. Lee, M. D. Grossberg, and S. K. Nayar, "Multispectral imaging using multiplexed illumination," in *ICCV*, 2007.

[13] I. Choi, M. Kim, D. Gutierrez, D. Jeon, and G. Nam, "High-quality hyperspectral reconstruction using a spectral prior," in *Technical report*, 2017.

[14] V. Backman, M. B. Wallace, L. Perelman, J. Arendt, R. Gurjar, M. Muller, Q. Zhang, G. Zonios, E. Kline, and T. McGillican, "Detection of preinvasive cancer cells," *Nature*, 2000.

[15] G. Lu and B. Fei, "Medical hyperspectral imaging: a review," *Journal of Biomedical Optics*, 2014.

[16] Z. Meng, M. Qiao, J. Ma, Z. Yu, K. Xu, and X. Yuan, "Snapshot multispectral endomicroscopy," *Optics Letters*, 2020.

[17] M. H. Kim, T. A. Harvey, D. S. Kittle, H. Rushmeier, R. O. P. J. Dorsey, and D. J. Brady, "3d imaging spectroscopy for measuring hyperspectral patterns on solid objects," *ACM Transactions on on Graphics*, 2012.

[18] Z. Pan, G. Healey, M. Prasad, and B. Tromberg, "Face recognition in hyperspectral images," *TPAMI*, 2003.

[19] H. V. Nguyen, A. Banerjee, and R. Chellappa, "Tracking via object reflectance using a hyperspectral video camera," in *CVPRW*, 2010.

[20] M. Borengasser, W. S. Hungate, and R. Watkins, "Hyperspectral remote sensing: principles and applications," *CRC press*, 2007.

[21] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, 2004.

[22] Y. Yuan, X. Zheng, and X. Lu, "Hyperspectral image superresolution by transfer learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017.