A Hybrid Model of State Space Model and Attention for Hyperspectral Image Denoising

Mingwen Shao*, Xiaodong Tan*, Kai Shang, Tiyao Liu, Xiangyong Cao

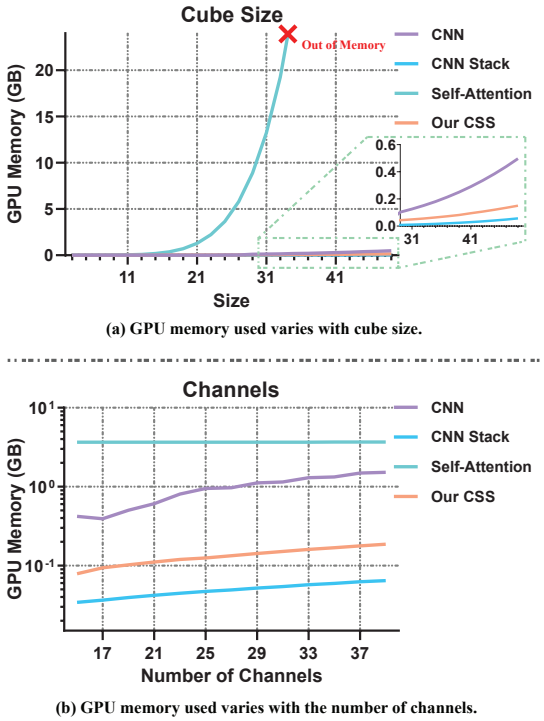
***Abstract*—Hyperspectral images (HSIs) exhibit pronounced spatial similarity and spectral correlation. With these two physical properties taken into account, underlying clean HSI will be easier to derive from noisy images. However, existing denoising approaches struggle to model the spatial-spectral structure due to the following limitations: 1) excessive memory consumption when performing global modeling, and 2) insufficient effectiveness in local modeling. To address these issues, we propose HyMatt, a hybrid model of state space model (SSM) and attention mechanism for hyperspectral image denoising. Specifically, to fully exploit global similarity within an HSI cube, we devise Vision Mamba Quad Directions (VMamba4D) based on crafted Cube Selective Search (CSS) to capture long-range dependencies in a memory-efficient manner. Our CSS not only enhances global modeling capacity but also mitigates the negative impacts of causal modeling inherent in SSM. Furthermore, in order to improve local similarity modeling, we integrate a Local Attention (Local Attn) module, in which the adjacent elements are refined by adaptively utilizing similar neighboring features as guidance. Compared to existing methods, our HyMatt excels in exploiting local features while leveraging the global similarity within the entire HSI cube. Extensive experiments on both simulated and real remote sensing noisy images demonstrate that our HyMatt consistently surpasses the state-of-the-art HSIs denoising methods.**

***Index Terms*—Hyperspectral Image Denoising, State Space Model, Mamba, Attention.**

I. Introduction

HYPERSPECTRAL Images (HSIs) are multidimensional data cubes that capture a wealth of spectral and spatial information across a wide range of wavelengths. This unique characteristic allows HSIs to demonstrate detailed material properties and subtle differences between surfaces, making them invaluable for applications such as precision farming [1], target detection [2], environmental monitoring [3], medical diagnosis [4] and so on. Despite their potential, HSIs are prone to various noise [5], including Gaussian, impulse, and stripe noise, due to environmental interference [6, 7], sensor limitations [8] and other factors. These noise artifacts degrade the quality of HSIs and can significantly hinder their utility in subsequent processing and analysis [9, 10, 11]. Effective HSIs denoising is thus a critical preprocessing step [12, 13, 14] to enhance the signal-to-noise ratio and ensure the reliability of HSIs for high-level tasks. And developing efficient HSIs denoising algorithms remains an ongoing challenge and a necessary step to unlock the full potential of hyperspectral imaging in real-world applications.

Most existing studies on HSIs denoising consider two key properties: spatial similarity [15] and spectral correlation [16, 17, 18, 19]. The former implies that multiple regions within the spatial domain share consistent patterns, allowing



(a) GPU memory used varies with cube size.

(b) GPU memory used varies with the number of channels.

Fig. 1. GPU memory consumption comparison between various types of global similarity modeling using a single NVIDIA GeForce RTX 3090. CNN denotes a single-layer convolutional kernel whose kernel size is the same as the cube size. CNN Stack represents a stack of several convolutional kernels of kernel size 3x3x3 stacked to a receptive field size equal to the cube size. (a) GPU memory used varies with cube size. The horizontal coordinates indicate the side length of the cube. (b) GPU memory used varies with the number of channels. The horizontal coordinates indicate the number of feature cube channels.

This work was supported by National Key Research and development Program of China (2021YFA1000102), and in part by the grants from the National Natural Science Foundation of China (Nos. 62376285, 61673396), Natural Science Foundation of Shandong Province, China (No. ZR2022MF260).

*These authors contributed equally to this work.

Corresponding author: Xiaodong Tan.

Mingwen Shao, Xiaodong Tan and Tiyao Liu are with the Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, Shandong 266580, China (e-mail: smw278@126.com; reyes.tan@foxmail.com; B24070012@s.upc.edu.cn).

Kai Shang is with Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, Shandong 266580, China, and also with Shandong Institute of Petroleum and Chemical Technology, Key Laboratory of Intelligent Information Processing (e-mail: skkkyup@163.com).

Xiangyong Cao is with the School of Computer Science and Technology and the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: caoxiangyong@mail.xjtu.edu.cn).

relevant information from similar regions to aid in restoration of certain noise-contaminated areas. The latter indicates slight variations between elements in the same spatial region but across different bands, enabling the utilization of spectral characteristics to further denoise images. Recently, studies that consider both spatial and spectral similarity follow naturally and can achieve promising outcomes [20, 21]. These methods treat the HSI as a cube and directly process elements at specific spatial and band locations to acquire underlying pure signal. In subsequent sections, we will tentatively refer to spatial similarity, spectral correlation, and the combination of the two as self-similarity.

Conventional model-based HSIs denoising approaches typically employ manually crafted priors to explore self-similarity. These priors comprises total variation [20], non-local similarity [22], low-rankness [21, 23], sparsity [24] and so on, which are embedded in particular objective functions. By iteratively optimize these functions, a clear HSI that satisfies a certain priori hypothesis is obtained. Although favorable denoising results can sometimes be achieved, the complex hyperparameter search and time-consuming process make it difficult to be put into practical applications.

Currently, deep learning-based methods lie in their ability to infer rapidly and handle complex data representations efficiently, thereby gaining traction in the domain of HSIs denoising. Among these methods, the most frequently utilized components are convolutional neural network (CNN), recurrent neural network (RNN), and self-attention (SA). To denoise HSIs, 3D CNNs are employed to model local self-similarity [25]. Then through multiple stacking, the deep CNNs can expand the receptive field to model global self-similarity [26]. However, the real receptive field is often smaller than the theoretical one, which leads to suboptimal modeling [27]. Another way to use CNNs for global exploration is to use a large convolutional kernel, but the memory consumption of this approach rises dramatically as the number of channels of hidden features increases as shown in Fig. 1. Then there are some studies that make a compromise between global and local self-similarity. For example, QRNN3D [28] only establishes the global spectral correlation by quasi RNNs, while retaining the local spatial similarity modeled with CNNs. However, due to the inability of parallelize processing, more time is needed to train RNNs. SA, on the other hand, because of its ability to model the globe, the denoising algorithm is able to repair the degradation using similar pixels within image [29]. This approach can lead to huge boost due to the fact that the nature of weighted summation using attention scores precisely matches the concept of denoising using similarity. However, the computational complexity and memory consumption of attention mechanism is quadratic in the number of pixels so that SA is hard to apply to large size images. Therefore, some window based SAs are proposed to only interact pixels within a small sized window [30, 31, 32] . While this greatly reduces the hardware burden, it still suffers from some drawbacks of limited receptive field. Consequently, all modules mentioned above have shortcoming in modeling the global self-similarity.

More recently, there has been the development of state space model (SSM) based Mamba [33], capturing long-term dependencies and improving training and inference efficiency. Compared to the $O(N^2)$ complexity of SA, Mamba requires only $O(N)$ complexity to establish the head-tail interaction in a sequence with N elements. Also, this structure enables global manipulation of a sequence with very low memory consumption and prominent performance for natural language processing [34], time series prediction [35], DNA modeling [36] and so on. Nonetheless, like an RNN, Mamba is mainly used to model causal sequences. That is to say, given a sequence of vectors, the vector considers only the information of the vectors in front of it, while ignoring the information of the vectors behind it. This is not appropriate for image processing tasks. It is because, if an image is rearranged into a sequence, a large portion of the pixels will be neglected, making self-similarity modeling impossible. Consequently, some new vision mambas [37, 38, 39] are proposed to alleviate this problem. These methods are mainly based on applying the original mamba to the image sequence from multiple directions in a manner similar to a bi-directional RNN, allowing the pixels to interact with all rest pixels in the image. However, applying this method directly to HSIs would use only the pixel-to-pixel spatial dependencies and lose the correlation of the spectral information. In light of this, Mamba for HSIs should be able to adaptively establish element dependencies from both spatial and spectral domains, thus mitigating information loss due to causality while ensuring global self-similarity interactions.

In addition to the importance of global self-similarity, the more obvious property of HSIs is local self-similarity. In other words, localized elements are more resembling and more likely to be useful guidance [40, 41]. Inspired by window-based transformers, only interactions between pixels inside the window are considered, thus enhancing the fusion of local features. And despite the fact that local modeling can also be achieved by using multi-directional Mamba inside the window, it still won't be able to aggregate all the pixel information like SA can. Since we do not know what combination of directions will truly fully establish the dependencies between pixels, causal modeling will still brings sub-optimal effect. Additionally, reusing multi-direcional mamba inside the large number of small windows formed after partitioning has lost the advantage of lower memory consumption. As a result, the complement to Mamba's local modeling capability remains to be investigated.

With mentioned above, there are still pressing challenges to apply SSM-based Mamba to hyperspectral image denoising. In response to these issues, we propose HyMatt, a hybrid model of SSM and attention mechanism. Global similarity is established through our Visual Mamba Quad Directions (VMamba4D) based on Cube Selective Search (CSS) to ensure both long-range dependencies capture and low memory consumption (as shown in Fig. 1). Furthermore, local similarity is then complemented by our Local Attention (Local Attn) using a shifted cube local window-based self-attention. In summary, our contributions are as follows,

1) We propose HyMatt, a hybrid model of SSM and attention mechanism to model global and local self-similarity

inside HSI cube.
2) Our VMamba4D uses a sophisticated Cube Selective Scan to effectively and efficiently exploit global similarity for denoising while alleviate negative impacts brought by causal sequences modeling.
3) To compensate for the underutilization of the local cube by Mamba, we use a shifted cube window-based Local Attn to fully interact with the elements inside the window, thus enhancing the modeling of local self-similarity.
4) Our extensive experiments and network analysis showcase the benefits of HyMatt, as it achieves superior HSIs denoising performance across a range of noise degradation scenarios.

II. Related Work

In this section, we briefly review related advances in self attention and Mamba.

A. Self Attention

Self Attention (SA), as the core of Transformer, has an excellent performance in sequence processing tasks [42]. It is able to directly capture the dependencies between any two positions in a sequence by computing the attention score matrix, thus enabling the model to understand the contextual information of each position. This helps model to handle long-range interactions in the input data without information loss, improving the model's ability of global modeling.

For adapting the long-range capturing to global modeling in vision tasks, ViT [43] is proposed to make a connection between all patches of an image. However, it consumes too much memory to be applied in HSIs denoising because of an attention score matrix. To address this problem, window based SA comes into play. For example, SST [44] splits the image into equal-sized windows and computes SA within the window to model local spatial similarity. And it uses an additional channel-wise SA to compensate spectral correlation. SERT [45], on the other hand, extends the range of local spatial similarity by dividing the image into equal-sized rectangles horizontally and vertically, respectively. These approaches reduce memory consumption by reducing the number of vectors involved in the operation, but they also reduce the global receptive field so that only local similarity can be modeled. Meanwhile, they also fail to establish self-similarity in spatial and spectral domains simultaneously. The reason for this is that these methods treat the HSI as a multi-channel 2D image instead of a single-channel 3D cube. In light of this, we design a SA that conducts interactions from spatial and spectral directions in a cube to model self-similarity in a precise and comprehensive way.

B. Mamba

Mamba [33], a novel and promising neural network structure, is capable of manipulating long sequences with low computational complexity and memory consumption. And it has achieved impressive performance in various tasks, such as, natural language processing [34], time series prediction [35], DNA modeling [36] and so on. Its core, i.e, State Space Model (SSM), originates from control systems, which describe the behavior of a dynamic system using a set of 1st-order differential equations to represent the evolution of internal state in the system. To apply it to deep neural networks, many efforts [46, 47, 48, 49, 50] have been invested and ultimately result in Mamba as a universal efficient structure. Mamba devises a selective scan to filter out inconsequential information. And a hardware-efficient algorithm helps it dominate the field of fast global modeling long sequence with prominent performance.

The ability to global modeling is also required for visual tasks. The more global structure and contextual information in an image is captured, the more accurate the understanding of the content and the modeling of the relationships between objects will be. However, migrating Mamba directly to visual tasks will bring negative impact because of causal sequence modeling. That is to say, given a sequence of pixels rearranged from an image, the pixel considers only the information of the pixels in front of it, while ignoring the information of the pixels behind it. Therefore, to alleviate this problem, many studies have proposed the adaptation of Mamba to vision tasks [37, 38, 39, 51], including classification, detection, segmentation, restoration, understanding and so on. For instance, Vim [37] performs global sequence perception in two opposite directions from the head and tail of the pixel sequence, respectively. VMamba [38] gets four sequences from the top left, bottom left, top right and bottom right corners, thus allowing interaction between pixels at different positions. Back to HSIs denoising, HSIDMamba [39] takes a similar approach to VMamba, but just treats HSI as a multi-channel 2D image, which ignores similarity in spectral orientation. Therefore, based on the above reflections, we present a cube-wise multi-directional Mamba to fully model self-similarity from both spatial and spectral domains.

Albeit taking in the broader view can replenish a lot of information, the locality of an image is even more important. Some vision mambas have noticed this as well. For instance, in addition to performing full-image-wide interactions, LMa-UNet [40] divides the image into equal-sized windows and perform Mamba inside the windows. This approach emphasizes the role of local modeling on top of global modeling. Local Mamba [41], on the other hand, sequentially arranges the pixels in the local region first, and connects the different regions after. This method models both local and global in a single network block. However, because the operations are performed on numerous small windows, the memory advantage of these methods has been lost with respect to SA. Moreover, even if the drawback of causal modeling could be mitigated by using sequences of different order in multiple directions, we do not know which order would truly establish the dependencies between pixels. So, all in all, we use a SA to model the local similarity within a 3D window, which is able to take full advantage of all the neighboring elements within the cube while ensuring that the memory is not overloaded.

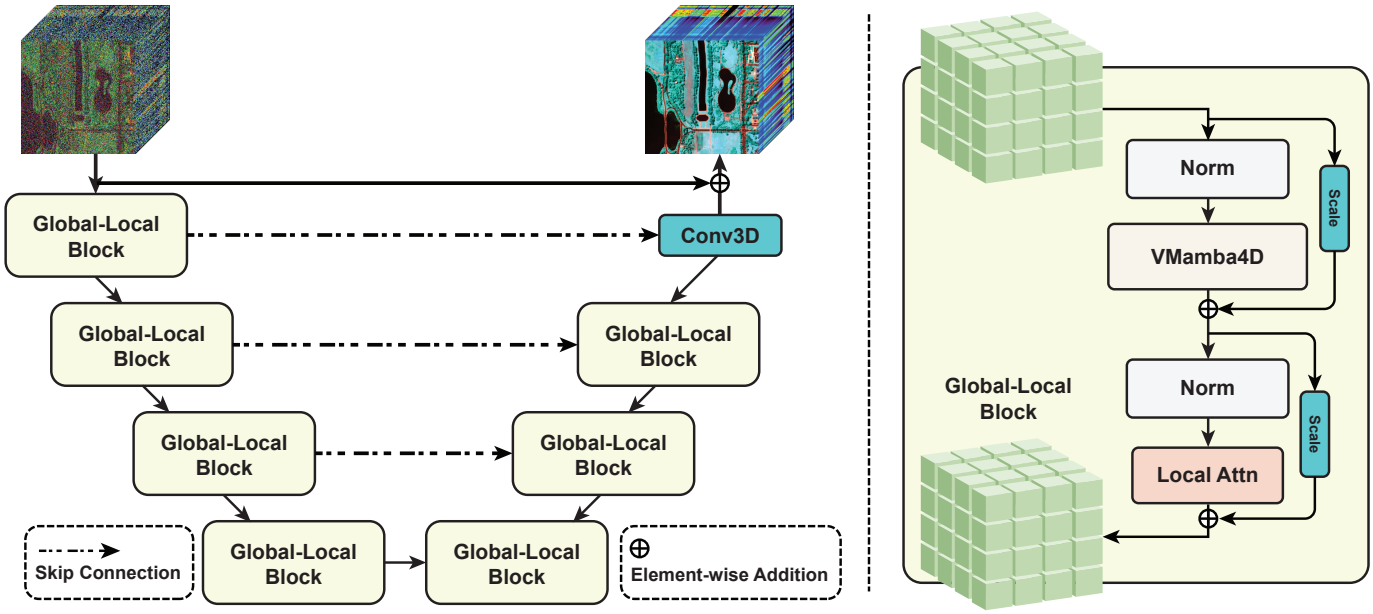


Fig. 2. The overall framework of HyMatt (left) and basic Global-Local Block (right). HyMatt is a U-shape architecture which consists of 4 levels of feature extraction process. And spatial resolution decreases as well as the number of channels increases from top to bottom. In Global-Local Block, there are two norm layers scattered between the two main modules, i.e., Visual Mamba Quad Directions (VMamba4D) and Local Attention (Local Attn).

III. Method

In this section, we will detail our proposed model, i.e., HyMatt. First, we will revisit SSM to clarify the roles involved. Then, the specific modules in HyMatt including Visual Mamba Quad Directions (VMamba4D), Cube Selective Search (CSS) and Local Attention (Local Attn), will be elaborated in different sections, respectively.

A. Revisit State Space Models

State Space Models (SSMs) are a series of time-invariant models which map a continuous 1-dim signal $x(t) \in \mathbb{R}$ to a response $y(t) \in \mathbb{R}$ through a hidden state $\mathbf{h}(t) \in \mathbb{R}^k$. The relationship between the three variables can be generally formalized as the following differential equation [52]:

$$\begin{aligned} \mathbf{h}'(t) &= \boldsymbol{A}\mathbf{h}(t) + \boldsymbol{B}x(t), \\ y(t) &= \boldsymbol{C}\mathbf{h}(t) + Dx(t), \end{aligned} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{k\times k}$ is the state (or system) matrix, $\boldsymbol{B} \in \mathbb{R}^{k\times 1}$ is input matrix, $\boldsymbol{C} \in \mathbb{R}^{1\times k}$ is output matrix, and $D \in \mathbb{R}$ is a feedthrough parameter. In order to easily and quickly adapt to process multi-dimensional continuous signal $\mathbf{x}(t), \mathbf{y}(t) \in \mathbb{R}^n$, Mamba [33] deal with them dimension by dimension.

To apply the Eq. (1) on a series of n-dim discrete sequence $\{\mathbf{x}_0, \mathbf{x}_1, \dots\} \subset \mathbb{R}^n$ instead of continuous signal $\mathbf{x}(t)$, zeroth-order hold (ZOH) is used to get a discrete version as follows:

$$\begin{aligned} \mathbf{h}_t &= \overline{\boldsymbol{A}}\mathbf{h}_{t-1} + \overline{\boldsymbol{B}}\mathbf{x}_t, \\ \mathbf{y}_t &= \boldsymbol{C}\mathbf{h}_t + D\mathbf{x}_t, \\ \overline{\boldsymbol{A}} &= \exp(\boldsymbol{\Delta A}), \\ \overline{\boldsymbol{B}} &= (\boldsymbol{\Delta A})^{-1}(\exp(\boldsymbol{\Delta A}) - \boldsymbol{I})\boldsymbol{\Delta B}, \end{aligned} \tag{2}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{\Delta}$ is a timescale parameter to convert $\boldsymbol{A}$, $\boldsymbol{B}$ to their dicretized form $\overline{\boldsymbol{A}}$, $\overline{\boldsymbol{B}}$, respectively. Note that the operations involving $\mathbf{x}_t$ and $\mathbf{y}_t$ here are applied to each dimension independently. In addition, 1st-order Taylor series is used to approximate $\overline{\boldsymbol{B}}$ as follows,

$$\begin{aligned} \overline{\boldsymbol{B}} &= (\boldsymbol{\Delta A})^{-1}(\exp(\boldsymbol{\Delta A}) - \boldsymbol{I})\boldsymbol{\Delta B} \\ &\approx (\boldsymbol{\Delta A})^{-1}\boldsymbol{\Delta A}\boldsymbol{\Delta B} \\ &= \boldsymbol{\Delta B}. \end{aligned} \tag{3}$$

Furthermore, to overcome selective copying and other failures brought by time-invariant models, Mamba utilize time-variable $\boldsymbol{\Delta}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ parameterized by input to model time-variant system.

B. Overall Model

We first regard a single HSI as a 4-order tenser, where the four orders represent feature channels, spectral bands, spatial width and spatial height. Our HyMatt is as shown in Fig. 2, which is a U-shape architecture. It consists of 4 levels of feature extraction process, in which there will be up/down-sampling between the levels to reduce spatial and spectral resolution while increasing the number of channels from top to bottom. Here, we use trilinear interpolation for up-sampling, and a convolution with kernel size of 3 and stride size of 2 for down-sampling. In order to represent our network architecture more succinctly, we omit these up/down-sampling operations in the illustration.

In each level, there are several stacked Global-Local Blocks to extract features. Its detail is shown in the right of Fig. 2. Like a vanilla transformer block, there are two norm layers scattered between the two main modules, i.e., VMamba4D and Local Attn, which we will explain in more detail later on. And, the two modules are encompassed by a scale as skip connection.

C. Visual Mamba Quad Directions

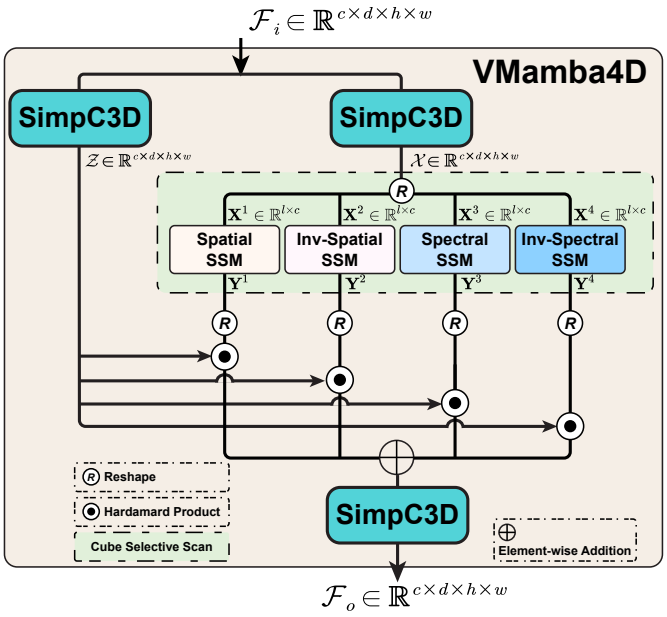


Fig. 3. Illustration of Visual Mamba Quad Directions (VMamba4D). To accommodate HSI processing, this module exploits SSM from four directions and modulates each direction with a gate mechanism.

To apply a mamba structure for our 4-order HSIs, the Visual Mamba Quad Directions (VMamba4D) is proposed. It takes the input $\mathcal{F}_i \in \mathbb{R}^{c\times d\times h\times w}$ as a 3-D cube with channels number of c to process. And the detail is as shown in Fig. 3. $\mathcal{F}_i$ is firstly separated to two branches. Because of the rich information from neighborhood, the right branch initially extracts local self-similarity in $\mathcal{F}_i$ through a SimpC3D to get $\mathcal{X}$. The detail of SimpC3D is as shown in Fig. 4. Apparently, it decouples a 3D convolution to 1D band convolution and 2D spatial convolution. This takes the amount of parameters down while approximating the 3D convolution.

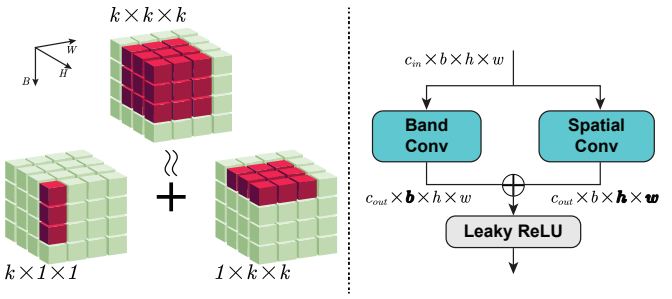


Fig. 4. Illustration of SimpC3D. In the left figure, the red feature element extracted by 3D convolution can be approximated through decomposing it into two convolutions in the band direction and in the spatial direction. In the right figure, SimpC3D uses 1D band convolution and 2D spatial convolution to mimic 3D convolution, aiming to reduce the number of parameters.

Then in Cube Selective Scan (CSS), $\mathcal{X}$ is transformed and copied to four parts, i.e., Spatial, Inv-Spatial, Spectral and Inv-Spectral, with same shape of $l \times c$, where $l = d \times h \times w$. The four copies thereafter are fed into selective scan to model global self-similarity, which we will explicate in next section. Subsequently, each part of the CSS output is reshaped back to $c \times d \times h \times w$, and dynamically adapted by modulating variable $\mathcal{Z} \in \mathbb{R}^{c\times d\times h\times w}$ from gate mechanism in the left branch. Finally, the output $\mathcal{F}_o$ is obtained from fusion of these four parts. The overall process is formalized as follows,

$$\mathcal{Z} = \text{SimpC3D}(\mathcal{F}_i), \tag{4a}$$

$$\mathcal{X} = \text{SimpC3D}(\mathcal{F}_i), \tag{4b}$$

$$\mathbf{Y}^1, \mathbf{Y}^2, \mathbf{Y}^3, \mathbf{Y}^4 = \text{CSS}(\mathcal{X}), \tag{4c}$$

$$\mathcal{F}_m^i = \mathcal{Z} \odot \mathbf{Y}^i, \text{for } i \text{ in } \{1,...,4\}, \tag{4d}$$

$$\mathcal{F}_o = \text{SimpC3D}(\sum_{i=1}^{4} \mathcal{F}_m^i), \tag{4e}$$

where $\odot$ denotes the Hadamard product. And to more accurately represent the main process in VSSM3D, we omit the reshape between Eq. (4c) and Eq. (4d).

D. Cube Selective Scan

To fully utilize global similarity inside a 3-D cube, we need to allow fast and full interaction between elements in the cube. There have been three main approaches in the past, convolution neural networks (CNN), self-attention (SA) and recurrent neural networks (RNN). For using convolution, one must either set the size of the convolution kernel directly equal to the size of the feature cube, or gradually expand the receptive field with a stack of multiple smaller convolution kernels. When a large kernel is adopted, as the number of channels in the feature cube increases, the memory requirement also increases significantly. Whereas the theoretical receptive field increases with the number of layers when using multiple small convolutional kernel stacks, the practical effective receptive field may be smaller than the theoretical value for several reasons, such as vanishing gradients [27]. For using self-attention, each element has to interact with the rest of the remaining elements. Although current hardware parallelizes computation well, it requires large memory. For the use of recurrent neural networks, on the other hand, elements that have some spatial correlation are rearranged into a single sequence, making the similarity modeling weakened. At the same time, this method consumes a lot of training time due to the recursive output.

To deal with these problems above, we propose Cube Selective Scan (CSS) using SSM-based selective scan. As we mentioned in section III-A, selective scan processes sequences like a RNN. However, with hardware efficient algorithm implemented by Mamba [33], it is able to parallelize computation on the GPU. Of course, by simply rearranging the cubic feature into a sequence, it still breaks up structures with spatial correlations which leads to a sub-optimal effect. The specific reason for this is the causal modeling of a sequence. Under the causal hypothesis, SSM-based method rarely consider space location information, which means that an element at a specific position can only interact with information from elements prior to this position. To alleviate this problem in feature cube, we model global self-similarity from four branches, i.e., spatial, inv-spatial, spectral and inv-spectral. Specifically, spatial branch directly rearranges initial cube into a sequence. Inv-spatial branch uses the reverse order of the sequence obtained by

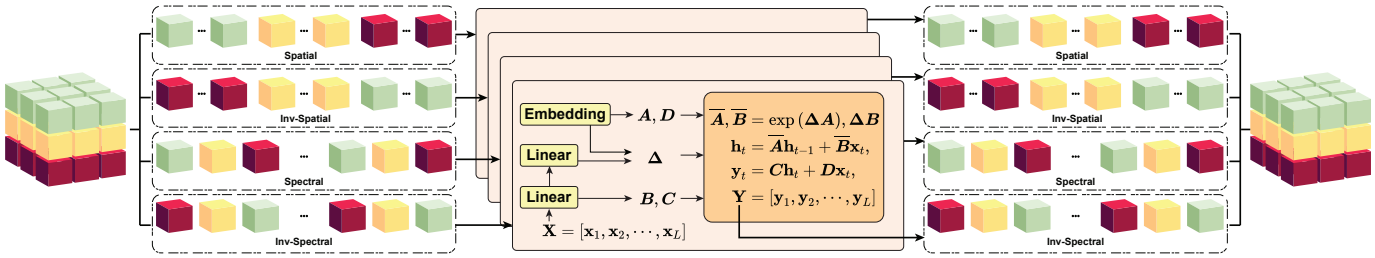


Fig. 5. Illustration of our Cube Selective Scan (CSS). This module models global similarity by rearranging inputs into sequences of different order from three directions, while mitigating the negative effects of causal sequence modeling.

Algorithm 1 Cube Selective Scan.

REQUIRE: $\mathcal{X}$: $[c, d, h, w]$
ENSURE: $\mathbf{Y}^1, \mathbf{Y}^2, \mathbf{Y}^3, \mathbf{Y}^4$: $[l, c]$
$\mathbf{X}^1$:$[(d, h, w), c] \xleftarrow{\text{RESHAPE}} \mathcal{X}$
$\mathbf{X}^2$:$[(d, h, w), c] \xleftarrow[\text{FLIP}]{\text{RESHAPE}} \mathcal{X}$
$\mathbf{X}^3$:$[(h, w, d), c] \xleftarrow{\text{RESHAPE}} \mathcal{X}$
$\mathbf{X}^4$:$[(h, w, d), c] \xleftarrow[\text{FLIP}]{\text{RESHAPE}} \mathcal{X}$
FOR i **IN** $\{1, \ldots, 4\}$ **DO**
  $\boldsymbol{A} : [c, k] \leftarrow$ Embedding
    ▷ Representing $k \times k$ matrix
  $\boldsymbol{B} : [l, k] \leftarrow \text{Linear}^{\boldsymbol{B}}(\mathbf{X}^i)$
  $\boldsymbol{C} : [l, k] \leftarrow \text{Linear}^{\boldsymbol{C}}(\mathbf{X}^i)$
  $\boldsymbol{D} : [l, 1] \leftarrow$ Embedding
  $\boldsymbol{\Delta} : [l, c] \leftarrow$
    $\log(1 + \exp(\text{Linear}^{\boldsymbol{\Delta}}(\mathbf{X}^i) + \text{Embedding}^{\boldsymbol{\Delta}})$
  $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : [l, c, k] \leftarrow \text{discretize}(\boldsymbol{\Delta}, \boldsymbol{A}, \boldsymbol{B})$
    ▷ Using Eq. (2) & Eq. (3)
  $\mathbf{Y}^i$: $[l, c] \leftarrow \text{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C}, \boldsymbol{D})(\mathbf{X}^i)$
END FOR
RETURN $\mathbf{Y}^1, \mathbf{Y}^2, \mathbf{Y}^3, \mathbf{Y}^4$

spatial branch. And spectral branch rotates three directions of the cube firstly, and then rearranges it to a sequence. Subsequently, the inv-spectral branch is reversed from the spectral branch. At last, these four branches are performed SSM and merged, thus modeling global similarity quickly and efficiently while maintaining low memory consumption. The whole process can be illustrated by Fig. 5. And the details of this module are elucidated in Algo. 1, where $[\bullet, \ldots, \bullet]$ denotes the shape of variables, $[(\bullet, \ldots, \bullet), \bullet]$ represents merging the dimensions of the variables in parentheses, and $l = d \times h \times w$. And FLIP denotes the reversal of order according to the first dimension.

E. Local Attention

Although powerful global self-similarity can be modeled by our VMamba4D, the most prominent intrinsic nature in a HSI is local self-similarity, which means that characteristics of an element are more similar to the adjacent region. For causal modeling, when a HSI cube is rearranged to a sequence, elements that were originally near the central element may be behind the central element of the sequence, making it impossible for the central element to utilize them. While this problem is alleviated by utilizing SimpC3D and multi-directional cube selective search in our VMamba4D, this is still insufficient for the urgently needed self-similarity modeling. So inspired by Swin Transformer, we propose the Local Attn, to utilize neighborhood feature vectors through self-attention. Specifically, it consists of two stages: windows partition and windows shift. During the partition, a feature cube will be divided into some windows with uniform size. Inside a window, all features will be interacted by self-attention, which guarantees that every feature near the center can be well modeled with self-similarity. However, the features at the edges of the window are not able to interact with their

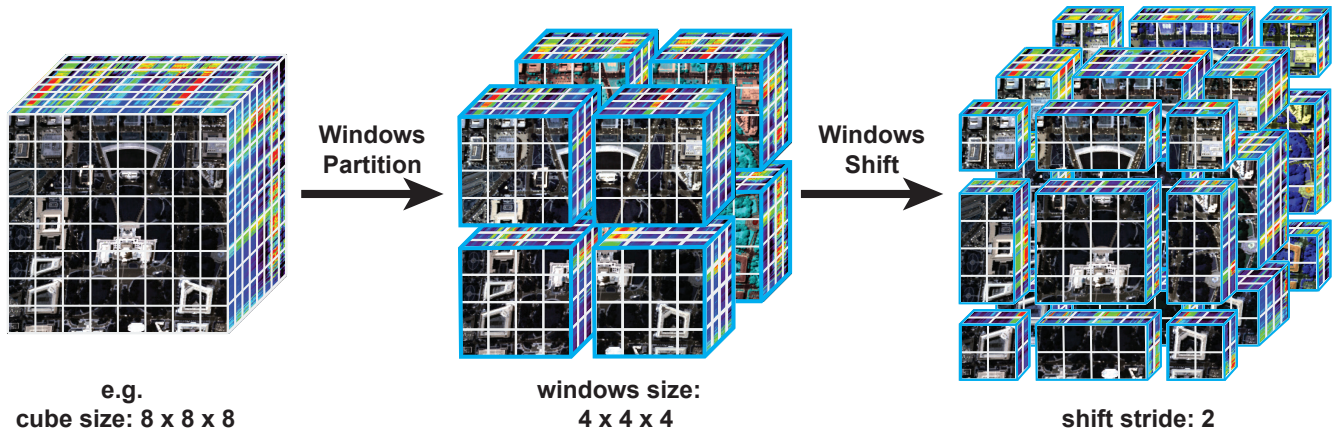


Fig. 6. Illustration of windows partition and shift as the window size is $8 \times 8 \times 8$ and stride size is $2 \times 2 \times 2$.

TABLE I
SIMULATED DENOISING RESULTS WITH DIFFERENT NOISE ON ICVL. CASES 1 TO 4 REFER TO GAUSSIAN NOISE WITH SIGMA OF 30, 50, 70 AND BLIND RESPECTIVELY, WHILE CASES 5 TO 9 REFER TO NON-I.I.D., STRIPE, DEADLINE, IMPULSE AND MIXTURE COMPLEX NOISE RESPECTIVELY. OUR METHOD CAN OUTPERFORM MOST OTHER METHODS IN TERMS OF PSNR, SSIM, AND SAM. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD AND THE SECOND BEST RESULTS ARE UNDERLINED.

Type	Metric	Noisy	M1	M2	M3	M4	QRNN3D	TRQ3D	HSID-CNN	HSDT	MAC-Net	Ours
Case 1	PSNR	20.01	37.84	41.12	42.44	40.55	42.55	42.79	40.29	**43.39**	41.76	42.97
	SSIM	0.147	0.920	0.967	0.97	0.972	0.971	0.969	0.959	**0.976**	0.963	0.972
	SAM	0.636	0.135	0.056	0.053	0.078	0.057	0.059	0.079	**0.050**	0.056	0.052
Case 2	PSNR	16.03	34.98	38.99	40.22	37.91	40.55	41.40	38.77	41.67	40.34	**41.81**
	SSIM	0.070	0.866	0.945	0.957	0.951	0.956	0.960	0.934	0.961	0.959	**0.964**
	SAM	0.747	0.178	0.075	0.059	0.109	0.070	0.073	0.099	0.069	0.063	**0.058**
Case 3	PSNR	13.44	33.11	37.36	38.63	36.17	38.44	39.89	36.93	40.13	38.81	**40.35**
	SSIM	0.041	0.815	0.93	0.943	0.931	0.940	0.948	0.883	0.948	0.946	**0.950**
	SAM	0.808	0.214	0.087	0.088	0.134	0.081	0.086	0.133	0.090	0.078	**0.067**
Case 4	PSNR	18.91	36.27	40.97	41.11	39.75	41.70	41.74	39.24	42.67	40.67	**42.87**
	SSIM	0.145	0.852	0.956	0.965	0.962	0.961	0.965	0.946	0.965	0.951	**0.967**
	SAM	0.661	0.180	0.064	**0.057**	0.090	0.074	0.078	0.089	0.063	0.061	0.063
Case 5	PSNR	19.66	32.79	34.51	36.37	41.01	42.05	42.22	39.81	43.31	39.62	**43.45**
	SSIM	0.196	0.719	0.812	0.948	0.972	0.974	0.970	0.961	0.973	0.958	**0.976**
	SAM	0.693	0.178	0.187	0.061	0.070	0.057	0.063	0.072	0.057	0.076	**0.056**
Case 6	PSNR	19.26	32.65	33.87	36.16	36.77	41.62	41.93	39.19	43.04	39.28	**43.16**
	SSIM	0.188	0.710	0.799	0.917	0.913	0.967	0.963	0.957	0.972	0.955	**0.974**
	SAM	0.701	0.185	0.265	0.082	0.118	0.062	0.074	0.078	0.068	0.076	**0.058**
Case 7	PSNR	18.95	31.74	32.87	33.97	34.68	41.68	41.74	39.33	42.95	37.57	**42.99**
	SSIM	0.183	0.698	0.797	0.896	0.905	0.971	0.963	0.958	0.971	0.951	**0.974**
	SAM	0.716	0.263	0.276	0.101	0.107	0.061	0.075	0.075	0.060	0.102	**0.059**
Case 8	PSNR	15.79	29.64	28.60	35.30	20.02	39.43	39.90	36.41	40.74	36.28	**40.81**
	SSIM	0.134	0.623	0.652	0.908	0.402	0.949	0.950	0.921	0.952	0.948	**0.960**
	SAM	0.846	0.309	0.486	**0.092**	0.486	0.099	0.170	0.173	0.106	0.126	0.107
Case 9	PSNR	15.02	28.80	27.31	33.66	19.62	39.33	39.78	35.57	**40.46**	37.04	40.39
	SSIM	0.119	0.633	0.632	0.885	0.320	0.917	0.950	0.914	0.938	0.910	**0.951**
	SAM	0.868	0.321	0.513	0.110	0.520	0.165	0.102	0.179	**0.101**	0.306	0.108

neighboring features due to the separation of the window. Therefore, the windows shift is introduced at different layers, which means that all the windows are offset in one direction so that the features at the original edge positions are close to the center. This process can be illustrated by an example of feature cube with size of $8 \times 8 \times 8$ in Fig. 6. Then within the shifted window, standard attention is conducted to model local self-similarity. The formalization of whole process is as follows,

$$\mathcal{F}_{\mathrm{w}} = \text{windows partition}(\mathcal{F}_{\mathrm{ci}}), \tag{5a}$$

$$\mathcal{F}_{\mathrm{sw}} = \text{windows shift}(\mathcal{F}_{\mathrm{w}}), \tag{5b}$$

$$\mathcal{Q}, \mathcal{K}, \mathcal{V} = \text{reshape}(\mathcal{F}_{\mathrm{sw}}), \tag{5c}$$

$$\mathcal{F}_{\mathrm{co}} = \text{inv-reshape}(\text{Softmax}(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{c}})\mathcal{V}), \tag{5d}$$

where $\mathcal{F}_{\mathrm{ci}} \in \mathbb{R}^{c \times d \times h \times w}$ is the input feature cube with original size. $\mathcal{F}_{\mathrm{w}}$ and $\mathcal{F}_{\mathrm{sw}}$ are partitioned windows with a shape of $\mathbb{R}^{N \times c \times d_{\mathrm{w}} \times h_{\mathrm{w}} \times w_{\mathrm{w}}}$, where d_{w}, h_{w} and w_{w} are the sizes of window from three directions and $N = \lfloor \frac{d}{d_{\mathrm{w}}} \rfloor \times \lfloor \frac{h}{h_{\mathrm{w}}} \rfloor \times \lfloor \frac{w}{w_{\mathrm{w}}} \rfloor$. The reshape in Eq. (5c) is rearranging $\mathcal{F}_{\mathrm{sw}}$ to $\mathbb{R}^{N \times (d_{\mathrm{w}} \times h_{\mathrm{w}} \times w_{\mathrm{w}}) \times c}$. Moreover, in Eq. (5d), the transposition and multiplication for $\mathcal{Q}$, $\mathcal{K}$ and $\mathcal{V}$ are performed on matrices which are composed of their last two dimensions. And inv-reshape denotes the inverse the reshape in Eq. (5c).

IV. EXPERIMENT

In this section, we will first briefly introduce some public datasets and experiment setting. Then some prevailing HSI denoising methods are compared to our approach from aspects of quantity and quality.

A. Setting

1) Simulated Noise Datasets:

- ICVL [53] has 201 HSIs with a spatial size of 1392×1300 and 31 spectral bands range from 400 to 700 nm at 10 nm increments. To construct training and testing sets, all images are randomly cropped to 64×64 and 100 of them are selected for training and 50 for testing.
- Harvard [54] is taken with a Nuance FX (CRI Inc) that features an integrated liquid crystal tunable filter. This filter sequentially adjusts through thirty-one narrow wavelength bands, each about 10 nm wide, ranging from 420 to 720 nm. Since there are relatively few images, this dataset is used to verify the transfer performance of the deep network.
- Houston [55] is a remote sensing HSI dataset employed in the 2018 IEEE GRSS Data Fusion contest. The hyperspectral images measure 1202 × 4172 pixels and contain a total of 48 bands. Following [56], the final 46 bands are relatively free of noise and utilized to simulate real-world complex noise. We also use pre-trained models to transfer directly to this dataset for testing.

2) Simulated Noise Setting:

- Case 1-4: White Gaussian noise with zero-mean and different standard deviation of 30, 50, 70 and blind.
- Case 5: Different bands are corrupted by Gaussian noise with different deviation randomly selected from 10 to 70. The noise in this case is also known as non-i.i.d noise.

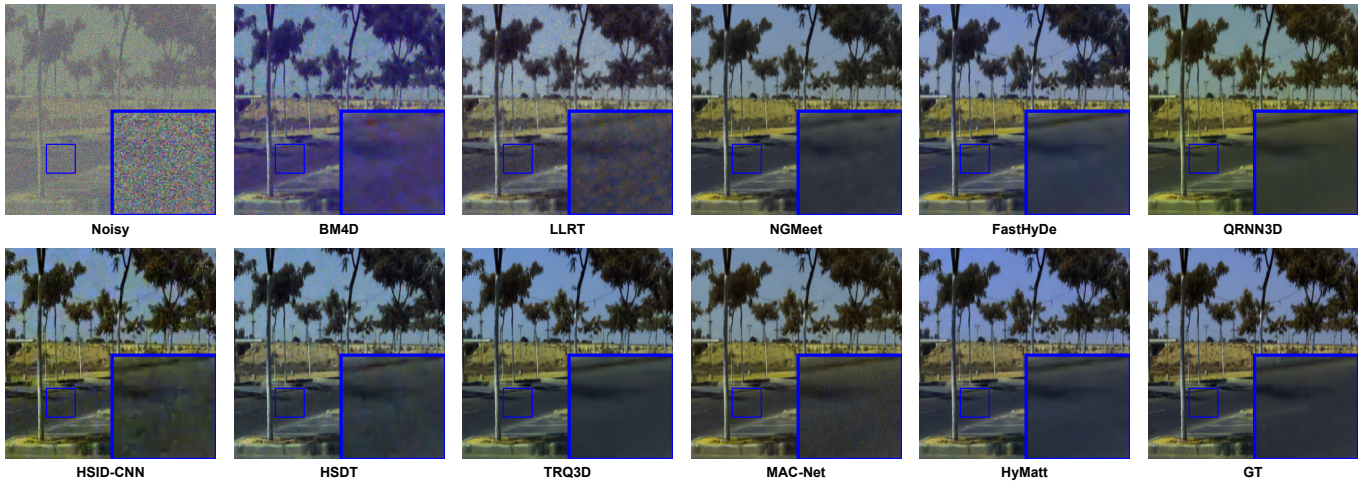


Fig. 7. Denoising visualization comparison for simulated Gaussian noise (case 3) on the ICVL. The pseudo-color image consists of bands (2, 20, 30). Zoom in for a better view of the difference. Our method can reproduce the noise-free image with better color fidelity.

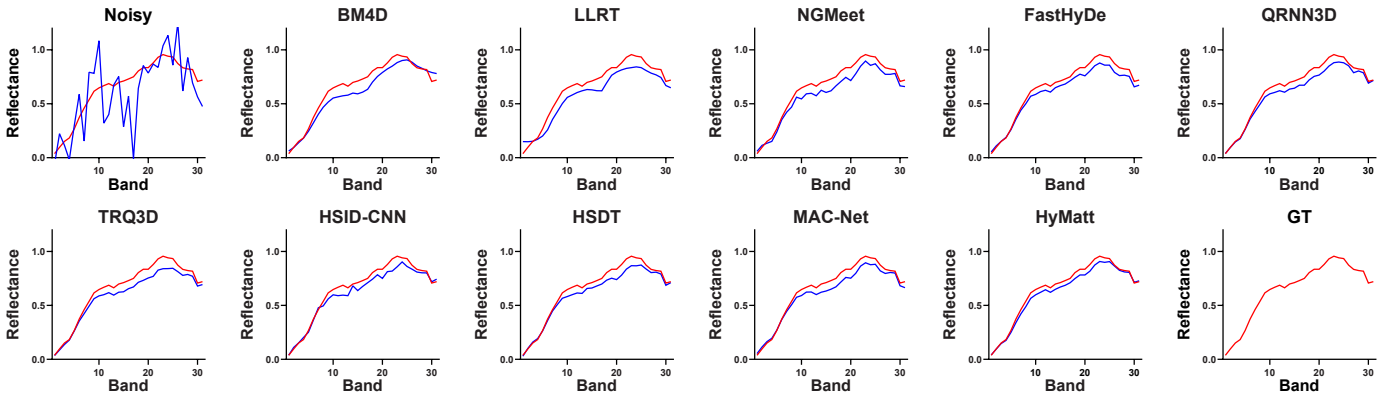


Fig. 8. Reflectance curve of Gaussian denoising results at point (297, 260) on the ICVL. Our method produces the curve that most closely resembles the ground truth.

- Case 6: Based on case 5, stripe noise is added to five to fifteen percent of the columns in the randomly selected one-third of bands.
- Case 7: Based on case 5, deadline noise is added to five to fifteen percent of the columns in the randomly selected one-third of bands.
- Case 8: Based on case 5, impulse noise with different intensity ranged from 0.1 to 0.7 is added to the randomly selected one-third of bands.
- Case 9: Each band is randomly contaminated by at least one of the types of noise mentioned in cases 5-8.

3) Real Noise Datasets:

- Urban[1] includes an remote sensing image measuring 307×307 pixels and encompassing 210 spectral bands, spanning from 400 to 2500 nm, resulting in a spectral resolution of 10 nm. Some of these bands are affected by severe complex noise.

4) Methods used for comparison: Model-based methods include BM4D [57], LLRT [58], NGMeet [59], FastHyDe [60], LRMR [61], NMoG [62], LRTDTV [63] and FastHyMix [64]. Deep learning-based methods include HSID-CNN [65], QRNN3D [66], TRQ3D [67], HSDT [68] and MAC-Net [69]. Since the first four model-based methods are designed to remove Gaussian noise while others are primarily used to remove complex noise, so we use different methods for comparison under different noise settings. Specifically, for cases 1 to 4 of Gaussian noise, M1, M2, M3 and M4 represent the first four model-based methods. And for cases 5 to 9 of complex noise, M1, M2, M3 and M4 represent the latter four model-based methods.

5) Hardware & Criteria: All model-based methods are implemented in MATLAB on AMD Ryzen 5700X @ 3.00GHz. And other methods are evaluated on a single NVIDIA A6000. PSNR, SSIM and SAM are as metrics to quantify denoising performance.

B. Denoising Performance on Simulated Noise

1) ICVL: For cases 1 to 4, all deep learning based methods are trained with blind i.i.d Gaussian noise (i.e. case 4) and tested directly on other cases. And for cases 5 to 9, these methods are trained with mixture noise (i.e. case 9) and tested directly on other cases.

[1]Available at https://rslab.ut.ac.ir/data

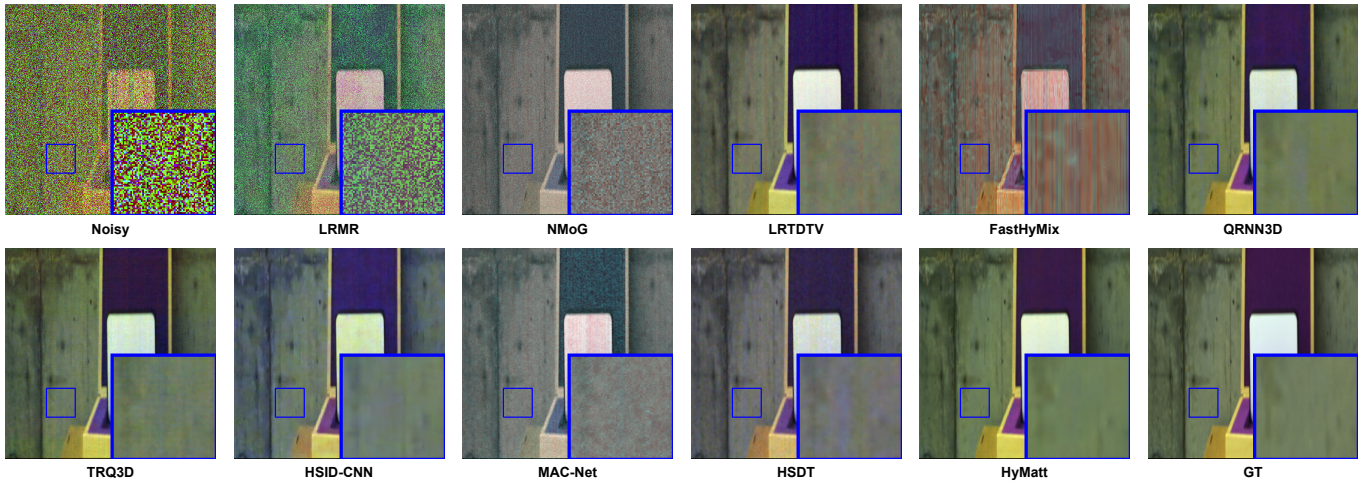


Fig. 9. Denoising visualization comparison for simulated complex noise (case 9) on the ICVL. The pseudo-color image consists of bands (2, 20, 30). Zoom in for a better view of the difference. Our method can reproduce the noise-free image with better color fidelity.

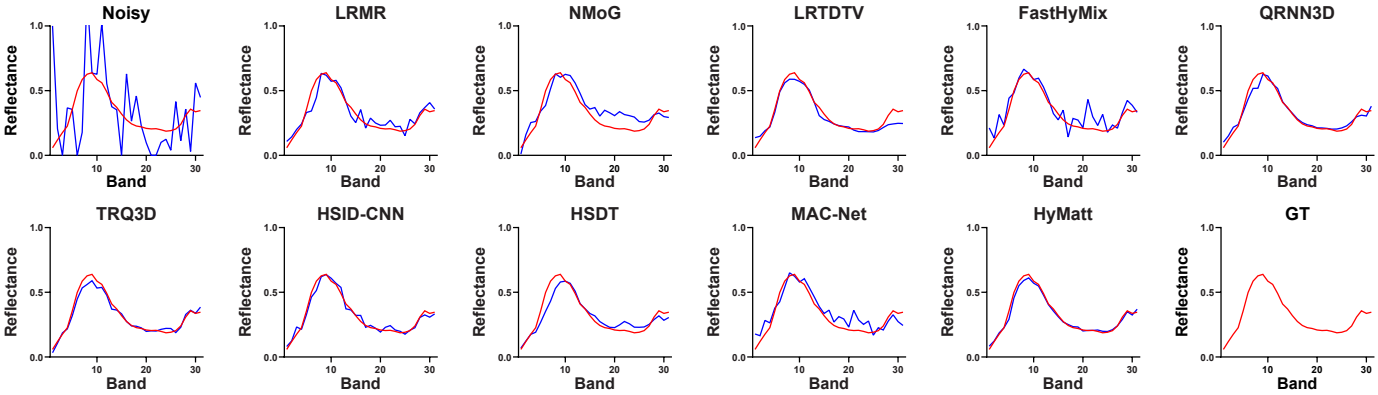


Fig. 10. Reflectance curve of complex denoising results at point (431, 361) on the ICVL. Our method produces the curve that most closely resembles the ground truth.

As shown in Tab. I, our HyMatt can outperform other performance in most noise scenarios with the quantitative result. In the cases of Gaussian noise, the performance of all methods decays with increasing intensity. And some methods show great potential. For instance, QRNN3D has a small PSNR decay (2.9 dB) under i.i.d. Gaussian noise while ensuring a comparable denoising effect. NGMeet, as a traditional method, achieves an acceptable performance without the need for extra data for training. While for cases of complex noise, it is clear to see that the effectiveness of most of the methods is significantly reduced in the presence of impulse noise. Thus, their metrics in case 9 are closer to the metrics in case 8.

We also visualize the denoised images in Fig. 7 (under case 3) and Fig. 9 (under case 9). It is obvious that some methods fail to denoise Gaussian noise in Fig. 7, such as BM4D, LLRT, and MAC-Net. In addtion, NGMeet, FastHyMix, HSID-CNN and HSDT blur the image. QRNN3D even makes the whole image yellowish. For denoising case 9 in Fig. 9, we can see that many approaches incorrectly estimate color and preserve much of the noise. As for QRNN3D, while the color and noise remain generally acceptable, purple artifacts appear in some areas.

As well, we plot the reflection curves of the denoised image versus the ground truth in Fig. 8 and Fig. 10. It is clear to see that our method is more accurate relative to other methods both in terms of predicted curve trend and curve values.

2) Harvard: Deep learning algorithm effectiveness is also reflected in generalizability. Because the spectral range of the Harvard dataset and the specific parameters of each band are similar to ICVL, it is most appropriate to use it for generalization experiments. We only conduct this experiments under Gaussian noise setting. For complex noise appearing in real remote sensing image, we will simulate it on Houston dataset.

As shown in Tab. II, our method can achieve good PSNR in most cases and other metrics at a comparable level. Due to comprehensive cooperation between the model-based and deep learning-based methods, MAC-Net demonstrates its powerful transfer capability. And for traditional method with a zero-shot inference manner, FastHyDe also achieves an acceptable performance.

Then we also visualize denoised images in Fig. 11. It is interesting to see that the results of some methods are whitish compared to ground truth. And most methods can't perfectly

TABLE II
SIMULATED DENOISING RESULTS WITH DIFFERENT NOISE ON HARVARD. CASES 1 TO 4 REFER TO GAUSSIAN NOISE WITH SIGMA OF 30, 50, 70 AND BLIND RESPECTIVELY. OUR METHOD HAS BETTER TRANSFERABILITY THAN OTHER DEEP LEARNING-BASED METHODS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD AND THE SECOND BEST RESULTS ARE UNDERLINED.

Type	Metric	Noisy	BM4D	LLRT	NGMeet	FastHyDe	QRNN3D	TRQ3D	HSID-CNN	HSDT	MAC-Net	Ours
Case 1	PSNR	20.30	37.96	36.87	38.63	40.53	41.31	40.56	40.25	41.01	41.35	**41.61**
	SSIM	0.127	0.900	0.902	0.926	**0.959**	0.954	0.955	0.940	0.944	0.939	0.957
	SAM	0.688	0.122	0.130	0.116	0.081	0.078	0.089	0.086	0.079	0.088	**0.069**
Case 2	PSNR	16.31	35.55	34.33	35.67	37.63	39.57	39.41	37.91	39.21	39.80	**40.11**
	SSIM	0.055	0.845	0.845	0.863	0.934	0.933	**0.938**	0.910	0.921	0.927	0.935
	SAM	0.800	0.161	0.180	0.164	0.113	0.092	0.095	0.109	0.091	0.123	**0.077**
Case 3	PSNR	13.67	33.94	32.49	33.55	36.22	38.15	38.53	35.02	37.87	38.86	**39.03**
	SSIM	0.030	0.797	0.785	0.794	0.915	0.912	**0.924**	0.851	0.900	0.911	0.921
	SAM	0.858	0.195	0.227	0.209	0.131	0.108	0.100	0.140	0.102	0.155	**0.084**
Case 4	PSNR	20.13	36.58	36.65	38.18	40.30	**41.04**	40.31	39.61	40.65	40.90	40.99
	SSIM	0.152	0.818	0.889	0.906	**0.955**	0.950	0.952	0.929	0.941	0.925	0.925
	SAM	0.686	0.168	0.137	0.126	0.083	**0.080**	0.089	0.093	0.081	0.095	0.110

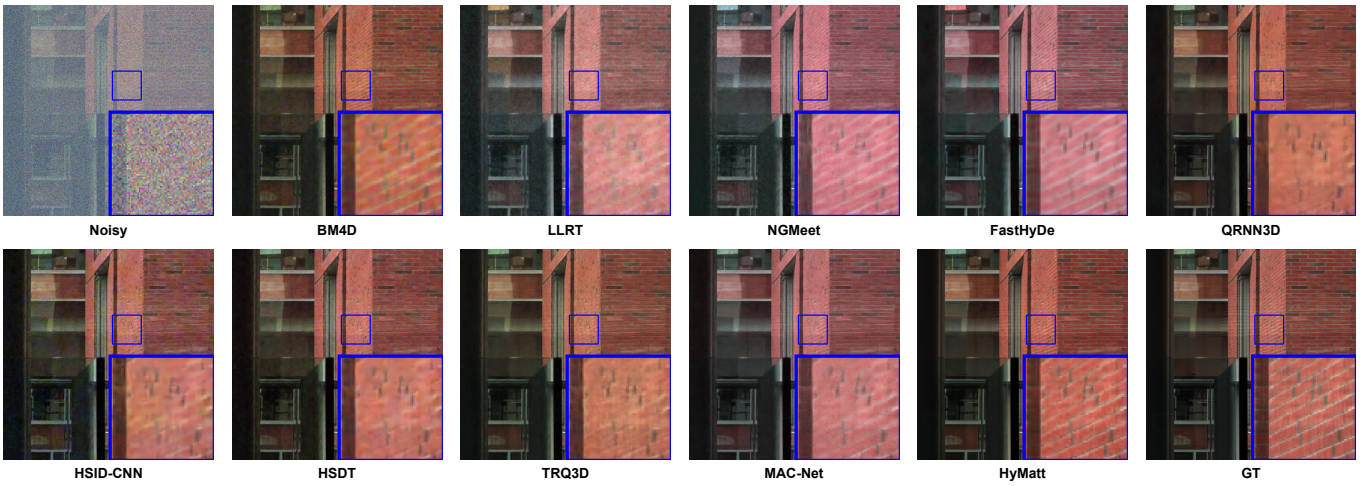


Fig. 11. Denoising visualization comparison for simulated Gaussian noise (case 3) on Harvard. The pseudo-color image consists of bands (5, 15, 20). Zoom in for a better view of the difference. Our method can reproduce the noise-free image with finer texture details.

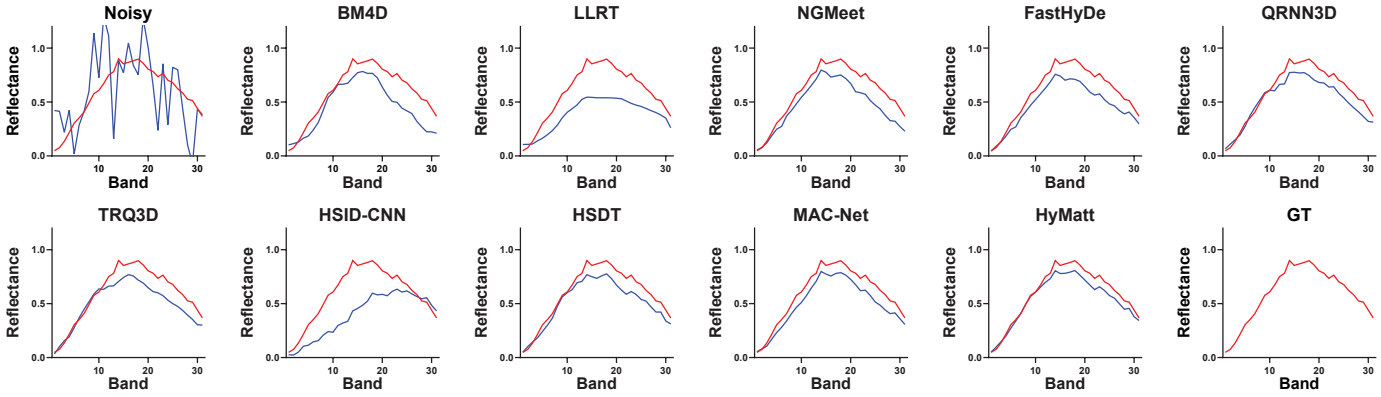


Fig. 12. Reflectance curve of Gaussian denoising results at (444, 250) on the Harvard. Our method produces the curve that most closely resembles the ground truth.

restore the textural details exhibited by the brick arrangement. Our proposed can better alleviate the above problems. At last, we also plot the reflectance in Fig. 12. Compared to methods, our HyMatt gives a better approximation of a clean image at shorter wavelengths and a better prediction of the reflectance trend at the remaining wavelengths.

3) Houston: Due to the large difference in spectral parameters between the test and training data, the superiority of model-based methods are starting to show with a zero-shot inferring manner. Especially for FastHyMix, it outperforms the other methods by more than 1 dB in terms of PSNR under cases 5 and 6, which is a huge advantage. However, for more complex and diverse noise conditions in cases 7-9,

TABLE III
SIMULATED DENOISING RESULTS WITH DIFFERENT NOISE ON HOUSTON. CASES 5 TO 9 REFER TO NON-I.I.D., STRIPE, DEADLINE, IMPULSE AND MIXTURE COMPLEX NOISE RESPECTIVELY. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD AND THE SECOND BEST RESULTS ARE UNDERLINED.

Type	Metric	Noisy	LRMR	NMoG	LRTDTV	FastHyMix	QRNN3D	TRQ3D	HSID-CNN	HSDT	MAC-Net	Ours
Case 5	PSNR	17.27	29.19	27.97	34.24	**36.93**	34.85	34.10	33.72	34.78	33.50	35.48
	SSIM	0.195	0.725	0.681	0.908	**0.952**	0.917	0.938	0.896	0.923	0.910	0.938
	SAM	0.609	0.165	0.168	0.087	**0.054**	0.097	0.106	0.111	0.091	0.126	0.080
Case 6	PSNR	17.15	29.04	28.20	34.21	**36.66**	34.71	32.98	33.75	34.71	33.57	35.44
	SSIM	0.190	0.719	0.692	0.906	**0.949**	0.916	0.936	0.895	0.922	0.910	0.937
	SAM	0.624	0.170	0.165	0.087	**0.057**	0.099	0.135	0.111	0.092	0.122	0.080
Case 7	PSNR	17.05	28.36	27.25	33.50	32.56	34.55	33.96	33.74	34.71	32.93	**35.13**
	SSIM	0.185	0.705	0.677	0.902	0.907	0.913	0.932	0.895	0.921	0.912	**0.935**
	SAM	0.641	0.188	0.200	0.092	0.092	0.100	0.105	0.111	0.091	0.124	**0.083**
Case 8	PSNR	12.45	19.72	20.08	29.54	22.12	33.53	33.49	33.64	33.89	29.82	**34.53**
	SSIM	0.090	0.446	0.452	0.808	0.511	0.898	0.923	0.904	0.912	0.836	**0.931**
	SAM	0.818	0.435	0.401	0.168	0.356	0.117	0.112	0.099	0.106	0.193	**0.096**
Case 9	PSNR	11.97	18.61	18.25	28.04	20.86	33.33	33.53	33.67	33.88	28.97	**34.31**
	SSIM	0.074	0.387	0.332	0.761	0.433	0.894	0.919	0.902	0.908	0.829	**0.927**
	SAM	0.854	0.508	0.564	0.220	0.411	0.125	0.105	0.103	0.105	0.267	**0.097**

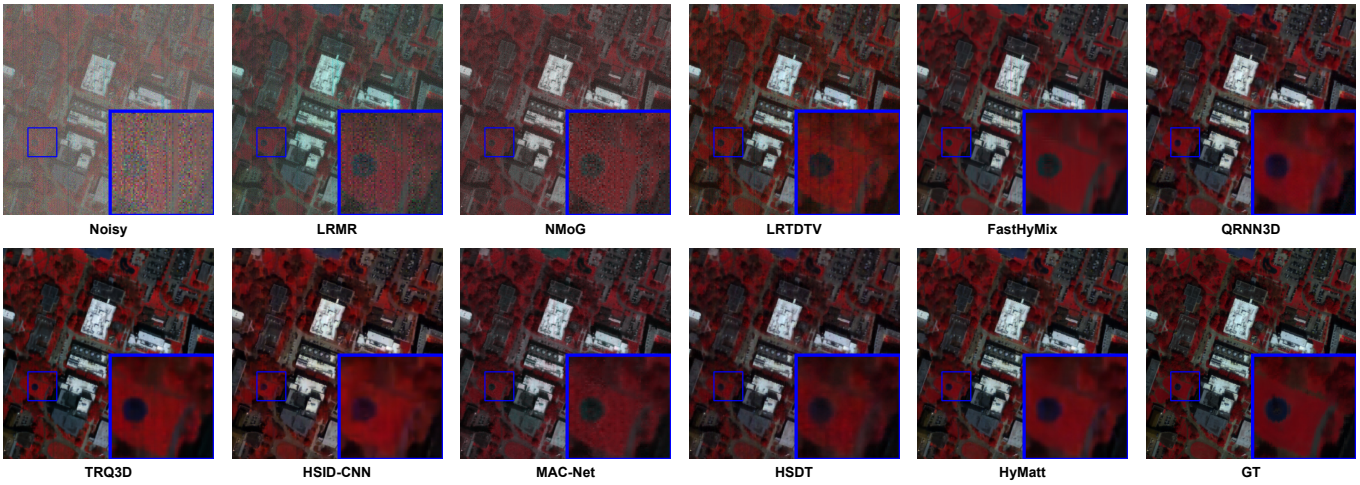


Fig. 13. Denoising visualization comparison for simulated complex noise (case 9) on Houston. The pseudo-color image consists of bands (9, 19, 29). Zoom in for a better view of the difference. Our method can reproduce a better noise-free image.

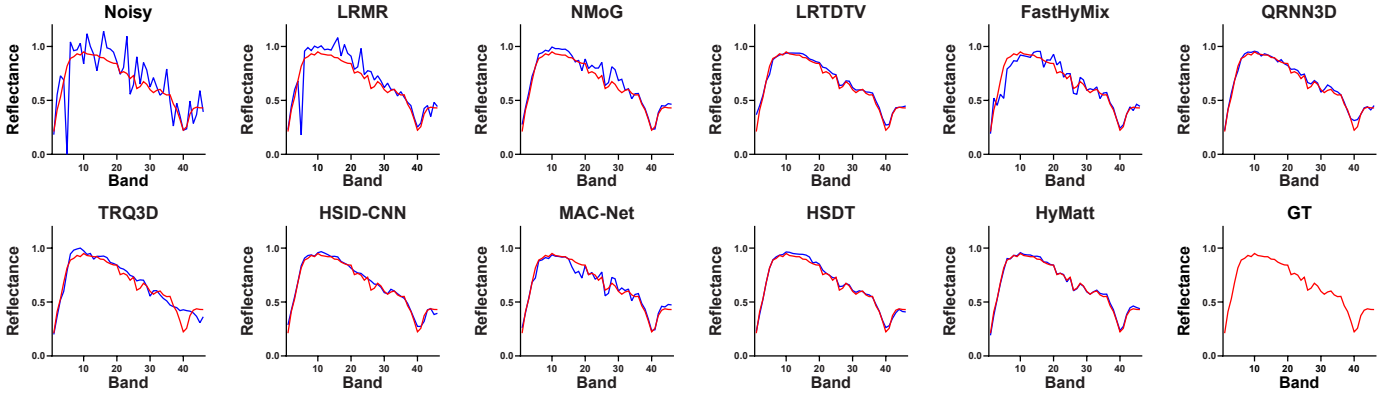


Fig. 14. Reflectance curve of complex denoising results at point (169, 229) on the Houston. Our method produces the curve that most closely resembles the ground truth.

most deep learning methods achieve better results. And our HyMatt dominates the top of the list.

Then we also visualize denoised images in Fig. 13. As it is shown, most methods fail to remove noise. Despite leaving very little noise visually, QRNN3D and ours also show a certain amount of excessive smoothing, which is something we need to improve further in the future. At last, we also plot the reflectance in Fig. 14. Compared to approaches, our method gives a better approximation of a clean image at all wavelengths.

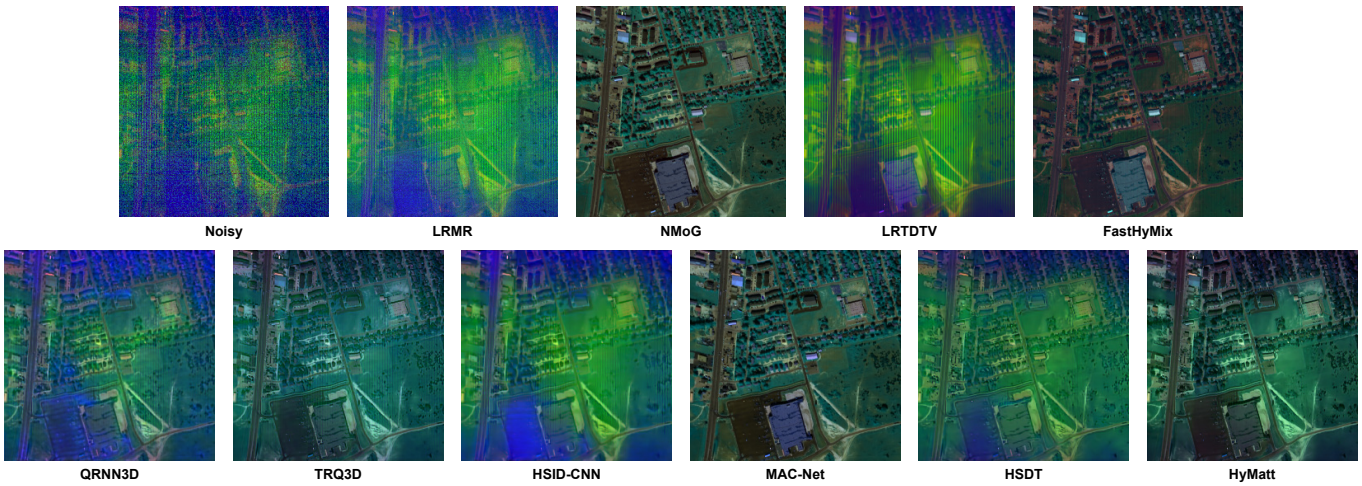


Fig. 15. Denoising visualization comparison for real complex noise on Urban. The pseudo-color image consists of bands (104, 139, 201). Zoom in for a better view of the difference. Our method can reproduce a comparable noise-free image.

TABLE IV
COMPARISONS OF PSNR, PARAMS, FLOPS AND INFERENCE TIME OF DIFFERENT METHODS AS THE INPUT SIZE IS $512 \times 512 \times 31$. PSNR FOR BM4D, LLRT, NGMEET AND FASTHYDE ARE AVERAGED OVER CASE 1-4 NOISE SETTINGS. PSNR FOR LRMR, NMOG, LRTDTV AND FASTHYMIX ARE AVERAGED OVER CASE 5-9 NOISE SETTINGS. AND THE PSNR FOR ALL DEEP-LEARNING BASED METHODS ARE AVERAGED OVER THE CASES 1-9 NOISE SETTINGS.

Method	BM4D	LLRT	NGMeet	FastHyDe	LRMR	NMoG	LRTDTV
PSNR(dB)	35.55	39.61	40.6	38.60	31.12	29.43	35.09
Params(M)	-	-	-	-	-	-	-
GFLOPS	-	-	-	-	-	-	-
Time(s)	134.801	627.000	166.000	2.010	106.155	152.674	202.074
Method	FastHyMix	QRNN3D	TRQ3D	HSID-CNN	HSDT	MAC-Net	Ours
PSNR(dB)	30.42	40.82	41.27	28.39	42.04	39.04	42.10
Params(M)	-	0.859	0.662	0.399	2.095	0.430	6.993
GFLOPS	-	1256.806	1067.870	3234.617	1048.055	-	710.789
Time(s)	2.066	0.337	0.587	0.602	0.767	2.803	1.978

C. Denoising Performance on Real Noise

We performed experiments on remote sensing hyperspectral image with realistic noise, namely the Urban dataset. As it's shown in Fig. 15, severe noise has prevented the image content from being recognizable. After denoising by each algorithm, the quality of the image is improved to some extent. Due to the huge gap in spectral parameters between this dataset and the dataset used for training the deep learning-based methods, some model-based methods with a zero-shot inference manner achieve better performance. Especially for NMoG, it can produce the most sharp and crisp image with fidelity. While among the deep learning-based approaches, most of them don't do well with direct transfer. And our HyMatt demonstrates lower denoising effectiveness than MAC-Net, possibly due to scarcity of explicit non-i.i.d noise prior which firstly converts Gaussian noise of different intensities between bands into easily removable noise of the same intensity.

V. DISCUSSION

A. Model Complexity

In addition to quantitative metrics and visual performance, the complexity, parameter count and time consumption of competing methods are as shown in Tab. IV. As mentioned in section IV-A, since some methods are only applicable to their specific noise settings, our PSNR values here are averaged across methods on their corresponding tasks. Specifically, performances of BM4D, LLRT, NGMeet and FastHyDe are evaluated only under Gaussian noise. LRMR, NMoG, LRTDTV and FastHyMix are evaluated only under complex noise. And all deep learning-based methods are evaluated under all kinds of noise. In addition, from the Tab. IV, Model-based methods lack parameter count indicators and run on CPU, while learning-based methods run on a single NVIDIA RTX 3090, with a test data size of $512 \times 512 \times 31$. We are unable to determine specific GFLOPs of MAC-Net because it requires matrix decomposition which is not supported by the assessment program. From the result, we can see that all model-based methods need a long time interval to inference except for FastHyDe and FastHyMix which are as the names suggests. Among deep learning-based methods, although our method has the smallest GFLOPs, it consumes a longer time. Since we take out each element of the HSI cube and form a very long sequence, our method does not gain any time advantage. However, we believe that the trade-off between time and memory costs is worth the improvement in effectiveness.

B. Ablation

1) Global v.s. Local: To assess main component in the proposed method, i.e., global modeling and local modeling, we

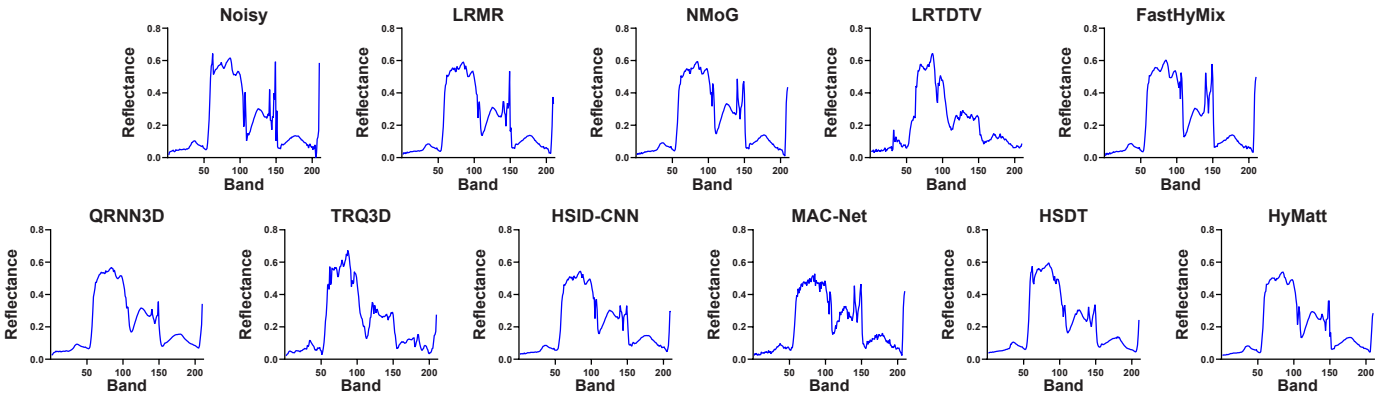


Fig. 16. Reflectance curve of complex denoising results at point (105, 231) on the Urban. Our method produces the curve that most closely resembles the ground truth.

TABLE V
ABLATION STUDY OF GLOBAL MODELING AND LOCAL MODELING.

	Global	Local	GFLOPs	Params(M)	Time(s)	PSNR
HyMatt-1	-	✓	196.367	0.206	1.213	41.55
HyMatt-2	✓	-	684.026	6.965	1.427	41.01
HyMatt (Ours)	✓	✓	710.789	6.993	1.978	42.87

further conduct ablation experiments to investigate the impacts of individual components. All experiments are performed in case 4 of the ICVL dataset.

To verify the efficacy brought by global modeling, we detach the VMamba4D from HyMatt to form the HyMatt-1. And we detach Local Attn to form HyMatt-2 for confirming the effectiveness of local modeling. As we can see in Tab. V, the absence of modeling, whether at the global or local level, has an adverse impact on denoising performance. Of the two, local modeling appears to be of greater consequence.

2) Multi-Directions: In VMamba4D, distinct combinations of directions also impact the particular denoising performance. To substantiate the efficacy of our four-direction strategy, we conducted ablation experiments on the sequence formation order. We evaluated the denoising effects of unfolding one-way and two-way sequences solely in the spatial and spectral domains, respectively.

TABLE VI
ABLATION EXPERIMENTS WITH DIFFERENT COMBINATIONS OF DIRECTIONS USED FOR SEQUENCES.

	w/ Inv	spatial	spectral	PSNR
HyMatt-3	-	✓	-	42.41
HyMatt-4	✓	✓	-	42.71
HyMatt-5	-	-	✓	42.49
HyMatt-6	✓	-	✓	42.75
HyMatt (Ours)	✓	✓	✓	42.87

As we can see in Tab. VI, the construction of sequences in either the spatial or spectral domain that are unidirectional results in a certain degree of performance degradation due to the limitations of causal modeling. Furthermore, using spatially-first or spectrally-first sequences alone for global modeling is not as effective as a combination of the two.

VI. CONCLUSION

In this study, we proposed HyMatt, a hyperspectral images denoising neural network. Compared to existing approaches, our method can model global self-similarity in a memory-efficient way while maintaining powerful local self-similarity modeling ability. Specifically, in VMamba4D module, we crafted a CSS strategy to conduct a SSM operation from quad directions within a HSI cube. This strategy serves to mitigate the negative impact of SSM with regard to the utilization of causal modeling for visual tasks. Additionally, for compensating local modeling, a 3D version of shifted window-based self attention is used to fuse neighborhood information in the Local Attn module. The incorporation of these two modules enables the network to derive a robust representation of similarity. And in experiments, the HyMatt demonstrated superior performance compared to other approaches. However, we believe there are limitations to the current implementation, because we found that considering both spatial and spectral sequences at the same time did not result in a large enhancement. This implies that, given a single sequence, we may be able to combine the advantages of both by simply changing its ordering. Therefore, future work may be able to dig further into the different orders of the sequences.

REFERENCES

[1] H. Jin, J. Peng, R. Bi, H. Tian, H. Zhu, and H. Ding, "Comparing laboratory and satellite hyperspectral predictions of soil organic carbon in farmland," *Agronomy*, vol. 14, no. 1, 2024.

[2] Y. Shi, X. Li, and S. Chen, "Iterative autoencoder coupling with constrained energy minimization for hyperspectral target detection," in *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, 2023, pp. 5862–5865.

[3] H. Flores, S. Lorenz, R. Jackisch, L. Tusa, I. C. Contreras, R. Zimmermann, and R. Gloaguen, "Uas-based hyperspectral environmental monitoring of acid mine drainage affected waters," *Minerals*, vol. 11, no. 2, 2021.

[4] C. Zhang, L. Mou, S. Shan, H. Zhang, Y. Qi, D. Yu, X. X. Zhu, N. Sun, X. Zheng, and X. Ma, "Medical hyperspectral image classification based weakly supervised single-image global learning network," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108042, 2024.

[5] H. K. Aggarwal and A. Majumdar, "Hyperspectral image denoising using spatio-spectral total variation," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 442–446, 2016.

[6] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, "Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1205–1218, 2019.

[7] Z. Xiao, H. Qin, S. Yang, X. Yan, and H. Zhou, "Spatial–spectral oriented triple attention network for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–17, 2024.

[8] L. Song, L. Wang, M. H. Kim, and H. Huang, "High-accuracy image formation model for coded aperture snapshot spectral imaging," *IEEE Transactions on Computational Imaging*, vol. 8, pp. 188–200, 2022.

[9] Y. Yan, J. Ren, Q. Liu, H. Zhao, H. Sun, and J. Zabalza, "Pca-domain fused singular spectral analysis for fast and noise-robust spectral–spatial feature mining in hyperspectral classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[10] M. L. Brandão Junior, V. C. Lima, T. A. P. P. Teixeira, E. R. de Lima, and R. d. R. Lopes, "Anomaly detection in hyperspectral images via regularization by denoising," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8256–8265, 2022.

[11] K. Naganuma and S. Ono, "Toward robust hyperspectral unmixing: Mixed noise modeling and image-domain regularization," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 8117–8138, 2024.

[12] J. Peng, Q. Xie, Q. Zhao, Y. Wang, L. Yee, and D. Meng, "Enhanced 3dtv regularization and its applications on hsi denoising and compressed sensing," *IEEE Transactions on Image Processing*, vol. 29, pp. 7889–7903, 2020.

[13] T. Xie, S. Li, and B. Sun, "Hyperspectral images denoising via nonconvex regularized low-rank and sparse matrix decomposition," *IEEE Transactions on Image Processing*, vol. 29, pp. 44–56, 2020.

[14] H. Zhang, L. Liu, W. He, and L. Zhang, "Hyperspectral image denoising with total variation regularization and nonlocal low-rank tensor decomposition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3071–3084, 2020.

[15] L. Lei, B. Huang, M. Ye, H. Chen, and Y. Qian, "A graph-regularized non-local hyperspectral image denoising method," in *Geometry and Vision*, M. Nguyen, W. Q. Yan, and H. Ho, Eds. Cham: Springer International Publishing, 2021, pp. 327–340.

[16] T.-X. Jiang, L. Zhuang, T.-Z. Huang, X.-L. Zhao, and J. M. Bioucas-Dias, "Adaptive hyperspectral mixed noise removal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[17] L. Zhuang and M. K. Ng, "Fasthymix: Fast and parameter-free hyperspectral image mixed noise removal," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4702–4716, 2023.

[18] L. Zhuang, M. K. Ng, L. Gao, J. Michalski, and Z. Wang, "Eigen-image2eigenimage (e2e): A self-supervised deep learning network for hyperspectral image denoising," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[19] J. Peng, H. Wang, X. Cao, Q. Zhao, J. Yao, H. Zhang, and D. Meng, "Learnable representative coefficient image denoiser for hyperspectral image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024.

[20] J. Peng, Q. Xie, Q. Zhao, Y. Wang, L. Yee, and D. Meng, "Enhanced 3dtv regularization and its applications on hsi denoising and compressed sensing," *IEEE Transactions on Image Processing*, vol. 29, pp. 7889–7903, 2020.

[21] P. Liu, H. Long, K. Ni, and Z. Zheng, "Multimode structural nonconvex tensor low-rank regularized hyperspectral image destriping and denoising," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.

[22] W. He, Q. Yao, C. Li, N. Yokoya, and Q. Zhao, "Non-local meets global: An integrated paradigm for hyperspectral denoising," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6861–6870.

[23] J. Peng, H. Wang, X. Cao, Q. Zhao, J. Yao, H. Zhang, and D. Meng, "Learnable representative coefficient image denoiser for hyperspectral image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024.

[24] H. V. Nguyen, M. O. Ulfarsson, J. Sigurdsson, and J. R. Sveinsson, "Deep sparse and low-rank prior for hyperspectral image denoising," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 1217–1220.

[25] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, "Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1205–1218, 2019.

[26] X. Cao, X. Fu, C. Xu, and D. Meng, "Deep spatial-spectral global reasoning network for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[27] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31: Revisiting large kernel design in cnns," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 953–11 965.

[28] K. Wei, Y. Fu, and H. Huang, "3-d quasi-recurrent neural network for hyperspectral image denoising," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 363–375, 2021.

[29] M. Zhao, G. Cao, X. Huang, and L. Yang, "Hybrid transformer-cnn for real image denoising," *IEEE Signal Processing Letters*, vol. 29, pp. 1252–1256, 2022.

[30] 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 *IEEE/CVF International Conference on Computer Vision*. IEEE, 2021, pp. 9992–10 002.

[31] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 12 114–12 124.

[32] M. Li, J. Liu, Y. Fu, Y. Zhang, and D. Dou, "Spectral enhanced rectangle transformer for hyperspectral image denoising," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5805–5814.

[33] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," Dec. 2023.

[34] J. Park, J. Park, Z. Xiong, N. Lee, J. Cho, S. Oymak, K. Lee, and D. Papailiopoulos, "Can mamba learn how to learn? A comparative study on in-context learning tasks," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 39 793–39 812.

[35] M. Zhang, K. Saab, M. Poli, T. Dao, K. Goel, and C. Ré, "Effectively modeling time series with simple discrete state spaces," in *International Conference on Learning Representations*, 2023.

[36] Y. Schiff, C. H. Kao, A. Gokaslan, T. Dao, A. Gu, and V. Kuleshov, "Caduceus: Bi-directional equivariant long-range DNA sequence modeling," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 43 632–43 648.

[37] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 62 429–62 442.

[38] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu, "VMamba: Visual State Space Model," Jan. 2024.

[39] Y. Liu, J. Xiao, Y. Guo, P. Jiang, H. Yang, and F. Wang, "HSIDMamba: Exploring Bidirectional State-Space Models for Hyperspectral Denoising," Apr. 2024.

[40] J. Wang, J. Chen, D. Chen, and J. Wu, "Large Window-based Mamba UNet for Medical Image Segmentation: Beyond Convolution and Self-attention," Mar. 2024.

[41] T. Huang, X. Pei, S. You, F. Wang, C. Qian, and C. Xu, "LocalMamba: Visual State Space Model with Windowed Selective Scan," Mar. 2024.

[42] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022.

[43] 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 *The ninth International Conference on Learning Representations*. OpenReview.net, 2021.

[44] M. Li, Y. Fu, and Y. Zhang, "Spatial-spectral transformer for hyperspectral image denoising," in *Thirty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press, 2023, pp. 1368–1376.

[45] M. Li, J. Liu, Y. Fu, Y. Zhang, and D. Dou, "Spectral enhanced rectangle transformer for hyperspectral image denoising," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5805–5814.

[46] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré, "Combining recurrent, convolutional, and continuous-time models with

linear state space layers," in *Advances in Neural Information Processing Systems 34*, 2021, pp. 572–585.
[47] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent memory with optimal polynomial projections," in *Advances in Neural Information Processing Systems 33*, 2020.
[48] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in *The Tenth International Conference on Learning Representations*. OpenReview.net, 2022.
[49] A. Gupta, A. Gu, and J. Berant, "Diagonal state spaces are as effective as structured state spaces," in *Advances in Neural Information Processing Systems 35*, 2022.
[50] A. Gu, K. Goel, A. Gupta, and C. Ré, "On the parameterization and initialization of diagonal state space models," in *Advances in Neural Information Processing Systems 35*, 2022.
[51] Y. Qiao, Z. Yu, L. Guo, S. Chen, Z. Zhao, M. Sun, Q. Wu, and J. Liu, "VL-Mamba: Exploring State Space Models for Multimodal Learning," Mar. 2024.
[52] W. L. Brogan, *Modern control theory (3rd ed.)*. USA: Prentice-Hall, Inc., 1991.
[53] B. Arad and O. Ben-Shahar, "Sparse Recovery of Hyperspectral Signal from Natural RGB Images," in *European Conference on Computer Vision*, 2016, pp. 19–34.
[54] A. Chakrabarti and T. Zickler, "Statistics of real-world hyperspectral images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011, pp. 193–200.
[55] "2018 IEEE GRSS Data Fusion Contest," Website, 2018, https://www.grss-ieee.org/community/technical-committees/data-fusion.
[56] G. Fu, F. Xiong, J. Lu, J. Zhou, and Y. Qian, "Nonlocal spatial–spectral neural network for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.
[57] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, 2013.
[58] Y. Chang, L. Yan, and S. Zhong, "Hyper-laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5901–5909.
[59] W. He, Q. Yao, C. Li, N. Yokoya, Q. Zhao, H. Zhang, and L. Zhang, "Non-local meets global: An iterative paradigm for hyperspectral image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2089–2107, 2022.
[60] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 3, pp. 730–742, 2018.
[61] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4729–4743, 2014.
[62] Y. Chen, X. Cao, Q. Zhao, D. Meng, and Z. Xu, "Denoising hyperspectral image with non-i.i.d. noise structure," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 1054–1066, 2018.
[63] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, and D. Meng, "Hyperspectral image restoration via total variation regularized low-rank tensor decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1227–1243, 2018.
[64] L. Zhuang and M. K. Ng, "Fasthymix: Fast and parameter-free hyperspectral image mixed noise removal," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4702–4716, 2023.
[65] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, "Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1205–1218, 2019.
[66] K. Wei, Y. Fu, and H. Huang, "3-d quasi-recurrent neural network for hyperspectral image denoising," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 363–375, 2021.
[67] L. Pang, W. Gu, and X. Cao, "Trq3dnet: A 3d quasi-recurrent and transformer based network for hyperspectral image denoising," *Remote Sensing*, vol. 14, no. 18, 2022.
[68] Z. Lai, C. Yan, and Y. Fu, "Hybrid spectral denoising transformer with guided attention," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 13 019–13 029.
[69] F. Xiong, J. Zhou, Q. Zhao, J. Lu, and Y. Qian, "Mac-net: Model-aided nonlocal neural network for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.