

# Adaptive 3D Convolution for Remote Sensing Image Fusion

Siran Peng, Xiangyu Zhu, *Senior Member, IEEE*, Shang-Qi Deng, Liang-Jian Deng, *Senior Member, IEEE*, and Zhen Lei, *Fellow, IEEE*

**Abstract**—Remote sensing image fusion aims to create a high-resolution multi/hyper-spectral image from a high-resolution image with limited spectral information and a low-resolution image with abundant spectral data. Recently, deep learning (DL) techniques have shown significant effectiveness in this area. Most DL-based methods approach image fusion as a 2D problem by encoding spectral information into feature map channels. However, our research suggests that this strategy introduces notable spectral distortions. In contrast, some methods consider spectral data as an additional dimension, utilizing standard 3D convolutions to preserve spectral information. Nevertheless, in a standard 3D convolutional layer, the same set of kernels is applied across all input regions, which we have found to be sub-optimal for image fusion. Furthermore, standard 3D convolutions necessitate substantial computational resources. To address these challenges, we propose a novel convolutional paradigm called Adaptive 3D Convolution (Ada3D) for remote sensing image fusion. Ada3D applies a unique set of 3D kernels to each input voxel, enabling the capture of fine-grained details. These adaptive kernels are generated through a two-step process: (i) spatial and spectral kernels are derived from their respective image sources; (ii) these two types of kernels are then combined to form content-aware 3D kernels that effectively integrate spatial and spectral information. Additionally, adaptive biases are introduced to enhance the convolutional outcome at the voxel level. Furthermore, we incorporate the group convolution technique to reduce computational complexity. As a result, Ada3D offers full adaptivity in an efficient manner. Evaluation results across five datasets demonstrate that our method achieves state-of-the-art (SOTA) performance, underscoring the superiority of Ada3D. The code is available at <https://github.com/PSRben/Ada3D>.

**Index Terms**—Remote sensing image fusion, pansharpening, hyper-spectral pansharpening, hyper-spectral image super-resolution (HISR), deep learning (DL), 3D convolution.

Siran Peng and Xiangyu Zhu are with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China; the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China. (Emails: pengsiran2023@ia.ac.cn, xiangyu.zhu@ia.ac.cn)

Shangqi Deng is with the State Key Laboratory of Human-Machine Hybrid Augmented Intelligence, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, No.28 Xianning West Road, China. (Email: shangqideng@stu.xjtu.edu.cn)

Liang-Jian Deng is with the School of Mathematical Sciences/Multi-Hazard Early Warning Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan 611731, China. (Emails: liangjian.deng@uestc.edu.cn)

Zhen Lei is with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China; the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China; the Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science and Innovation, Chinese Academy of Sciences, Hong Kong, China. (Email: zhen.lei@ia.ac.cn)

Corresponding author: Liang-Jian Deng.

Accepted by IEEE Transactions on Image Processing (TIP), 2026. DOI: 10.1109/TIP.2026.3689418.

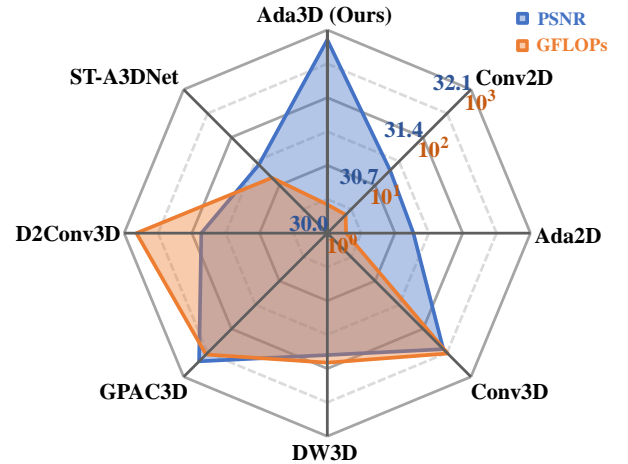


Fig. 1. Comparison of Ada3D with several convolutional paradigms, including standard 2D convolution (Conv2D), adaptive 2D convolution (Ada2D), standard 3D convolution (Conv3D), depth-wise 3D convolution (DW3D), grouped 3D PAC (GPAC3D) [1], D2Conv3D [2], and ST-A3DNet [3]. For fairness, all methods have an equal number of parameters. Quantitative evaluation outcomes on the WDC dataset demonstrates the superior effectiveness and efficiency of Ada3D. For detailed results, please refer to Table VI.

## I. INTRODUCTION

Satellite sensors often face challenges in obtaining high-resolution multi/hyper-spectral images due to hardware constraints. To overcome this, they can simultaneously acquire a high-resolution image with limited spectral information alongside a low-resolution image rich in spectral data. The objective of remote sensing image fusion is to integrate these two types of images, producing a high-resolution output enriched with spectral details. This study focuses on two remote sensing image fusion applications: pansharpening [4] and hyper-spectral pansharpening [5]. Pansharpening combines a high-resolution panchromatic (PAN) image with a low-resolution multi-spectral (LRMS) image to create a high-resolution multi-spectral (HRMS) result. Hyper-spectral pansharpening extends this technique to hyper-spectral images, generating a high-resolution hyper-spectral (HRHS) image from a PAN image and a low-resolution hyper-spectral (LRHS) image.

Traditional pansharpening and hyper-spectral pansharpening studies can be generally divided into three main categories: component substitution (CS) methods [6]–[8], multi-resolution analysis (MRA) approaches [9]–[11], and variational optimization (VO) techniques [12]–[16]. CS-based methods project the LRMS/LRHS image into a transformed domain, where spatial information is considered as an independent component. By re-

placing this component with the PAN image, they produce the desired HRMS/HRHS output. While recognized for their simplicity, low computational demands, and high spatial fidelity, these methods introduce significant spectral distortions. MRA-based approaches leverage an MRA framework to incorporate spatial details from the PAN image into the LRMS/LRHS image, creating an HRMS/HRHS result. These approaches excel in preserving spectral characteristics but may experience notable spatial distortions. In contrast, VO-based techniques focus on revealing the intrinsic relationships between two distinct image types. They typically utilize various forms of prior information to develop optimization models that merge spatial and spectral data. Despite careful design, VO-based techniques often struggle to produce satisfactory fusion results.

In recent years, deep learning (DL) has emerged as the premier solution for tackling image fusion challenges in the remote sensing field [17]–[27]. Inspired by advancements in other low-level vision tasks [28]–[31], most DL-based methods approach image fusion as a 2D problem, encoding spectral information into feature map channels. However, as we show mathematically in Section III-B, this can lead to spectral distortions. Alternatively, some methods approach image fusion as a 3D problem, dedicating an additional dimension to spectral data. Due to the limited variety of foundational DL models in the 3D domain, these methods typically rely on standard 3D convolutions for feature learning. In a standard convolutional layer, the same set of kernels is applied across different regions of various inputs, making the process *content-agnostic* [1]. Section III-C demonstrates that this is sub-optimal for image fusion. Moreover, the increased dimensionality of 3D convolutions substantially raises computational complexity.

Adaptive convolution marks a significant advancement in overcoming the content-agnostic limitations of standard convolution [32]–[34]. By utilizing distinct kernel sets for different input regions, adaptive convolution enhances the extraction of detailed information. Recently, this technique has been applied to remote sensing image fusion, demonstrating considerable promise [35]–[37]. However, its development has largely been confined to the 2D domain, with limited exploration in 3D. Existing approaches like D2Conv3D [2] and ST-A3DNet [3] still depend on globally shared kernels, preventing them from being considered fully adaptive 3D convolution techniques.

The aforementioned situation motivates us to propose Adaptive 3D Convolution (Ada3D), an innovative convolutional paradigm for remote sensing image fusion. Ada3D assigns a unique set of 3D kernels to each input voxel, allowing for the capture of fine-grained details. These adaptive kernels are generated through an intuitive two-step process: (i) spatial kernels are derived from the PAN image, while spectral kernels are extracted from the LRMS/LRHS image; (ii) these two types of kernels are then combined to form content-aware 3D kernels, effectively integrating both spatial and spectral information. In parallel, adaptive biases are computed using a similar strategy to enhance the convolutional outcome on a voxel-by-voxel basis. Additionally, we leverage group convolution to reduce computational complexity. Our experimental results show that Ada3D outperforms other 3D convolutional paradigms while maintaining FLOPs comparable to 2D methods, as depicted

in Fig. 1. Compared to existing approaches that are not fully adaptive, Ada3D establishes a new and efficient paradigm for 3D convolution, achieving voxel-level adaptivity by dynamically generating unique kernels and biases for each location. In conclusion, the *contributions* of this paper are as follows:

- 1) We mathematically illustrate that treating image fusion purely as a 2D problem can result in substantial spectral distortions (Section III-B). Furthermore, we prove that the kernel-sharing approach in standard convolution is sub-optimal for image fusion applications (Section III-C).
- 2) We introduce Ada3D, a novel 3D convolutional paradigm for remote sensing image fusion. In Ada3D, each voxel is processed with a unique set of adaptive kernels. These kernels are derived from two image sources, effectively integrating both spatial and spectral information. We also generate adaptive biases to enhance the convolutional outcome on a voxel-wise basis. Additionally, the group convolution technique is employed to reduce complexity.
- 3) We evaluate our method on five datasets, where Ada3D achieves state-of-the-art (SOTA) performance in qualitative and quantitative assessments. Notably, with the same number of parameters, Ada3D surpasses other 3D convolution paradigms in both effectiveness and efficiency.

## II. RELATED WORKS & MOTIVATIONS

### A. DL Methods for Remote Sensing Image Fusion

Over recent years, DL-based methods have gained dominance in various remote sensing image fusion applications. By leveraging the advantages of neural networks, these methods significantly outperform traditional approaches. Most DL-based studies treat image fusion as a 2D problem, encoding spectral information within feature map channels. Noteworthy contributions to this area include PNN [17], PanNet [18], and U2Net [25]. PNN, in particular, stands out as a groundbreaking accomplishment for introducing DL into remote sensing image fusion. Utilizing three standard 2D convolutional layers, PNN attains SOTA performance at the time of its publication. PanNet applies high-pass filters to extract edge information and employs residual network blocks (ResBlocks) [38] to merge spatial and spectral features. U2Net, on the other hand, integrates modified cross-attention modules within two U-shaped networks, achieving exceptional performance in the pansharpening task. However, these methods often introduce spectral distortions as they alter the representation of spectral information. To tackle this issue, some studies treat image fusion as a 3D problem, incorporating an additional dimension for spectral data. This strategy draws inspiration from successes in the wider domain of remote sensing image processing, where 3D convolutions have demonstrated remarkable effectiveness [39]–[43]. Building on this foundation, notable recent advancements in image fusion include 3DSSR-Net [44] and MSAC-Net [45]. The former utilizes stacked standard 3D convolutional layers to extract features while preserving spectral integrity. The latter proposes an innovative gating mechanism to improve the performance of standard 3D convolutions. Due to their increased dimensionality, these

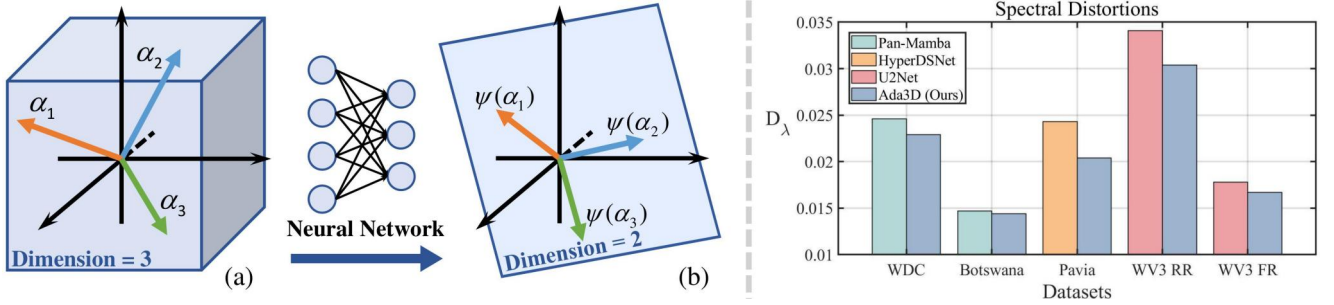


Fig. 2. **Left:** A visual representation of how 2D modeling introduces spectral distortions. (a) The vector space of spectral data, where the number of spectral bands  $L$  is set to 3 for simplicity. (b) The vector space of spectral information, which is encoded into feature map channels via a convolutional layer. This scenario illustrates the case where  $\text{rank}(\mathbf{A}) = 2 < L$ , resulting in a loss of spectral information. **Right:** A comparison of spectral distortions between Ada3D and SOTA 2D methods for image fusion, including Pan-Mamba [46], HyperDSNet [47], and U2Net [25]. Spectral distortions are quantified utilizing the  $D_\lambda$  [48] index, with lower  $D_\lambda$  values indicating fewer spectral distortions. Notably, Ada3D exhibits the lowest spectral distortions across all tested datasets.

methods are burdened by a significant volume of floating-point operations (FLOPs), which hampers their broader adoption.

### B. Adaptive Convolution

Standard convolution employs the same set of kernels across all regions of different inputs, leading to sub-optimal feature extraction and reduced flexibility [1]. In contrast, adaptive convolution assigns unique sets of kernels to each area of every input, allowing for more effective and dynamic feature extraction. The pioneering work in this field is DFN [32], which creates a separate network branch to generate adaptive 2D kernels from image input. Building on this, LS-DFN [49] leverages the attention mechanism to produce kernels with enhanced content information. Later achievements, such as PAC [1] and DDF [34], further highlight the promise of this novel convolutional paradigm. Adaptive convolution has recently been applied to remote sensing image fusion, with notable contributions such as LAGNet [35] and ADKNet [36]. The former introduces an innovative bias production technique and utilizes stacked adaptive convolutional layers to effectively merge spatial and spectral features. The latter proposes a two-branch kernel generator for information integration, demonstrating outstanding generalization capabilities. However, the methods discussed above are confined to 2D space, with only a few efforts made to explore the potential of adaptive convolution in 3D contexts. Existing approaches, such as D2Conv3D [2] and ST-A3DNet [3], continue to rely on globally shared kernels, which prevents them from being classified as genuine adaptive 3D convolution methods.

### C. Motivations

Most DL-based methods frame image fusion as a 2D vision problem, encoding spectral information within feature map channels. However, our mathematical analysis reveals that this approach can cause significant spectral distortions. Alternatively, some methods introduce an extra dimension for spectral data, treating image fusion as a 3D problem. These methods typically use standard 3D convolutions for feature learning, which we have found to be sub-optimal for image fusion applications. Moreover, standard 3D convolutions require substantial computational resources. By assigning unique sets

of kernels to various input regions, adaptive convolution offers a promising alternative to standard convolution. However, its development has predominantly been confined to 2D space, with limited exploration in 3D contexts. Therefore, we propose Ada3D, an innovative convolutional paradigm for remote sensing image fusion. Ada3D allocates a unique set of 3D kernels to each input voxel. These adaptive kernels consist of spatial and spectral components derived from the PAN and LRMS/LRHS images, facilitating the integration of different information. Additionally, we leverage the group convolution technique to greatly reduce complexity, bringing Ada3D's FLOPs to a level comparable to that of 2D convolution.

## III. MATHEMATICAL ANALYSIS

In this section, we first introduce some basic theories from linear algebra. Leveraging these theories, we mathematically illustrate the limitations of 2D modeling and standard convolution in remote sensing image fusion. This analysis serves as the theoretical foundation for the development of Ada3D. Throughout the rest of this paper, indices and sizes are denoted by  $i$  and  $H$ , vectors by  $\mathbf{x}$ , matrices by  $\mathbf{X}$ , and tensors by  $\mathcal{X}$ .

### A. Basic Theories

**Definition 1.** Given an  $M$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ , its mode- $k$  unfolding can be represented by a matrix  $\mathbf{X} \in \mathbb{R}^{I_k \times (I_1 I_2 \dots I_{k-1} I_{k+1} \dots I_M)}$ . In this unfolding, the element at position  $(i_1, i_2, \dots, i_M)$  in  $\mathcal{X}$  is mapped to the element at position  $(i_k, j)$  in  $\mathbf{X}$ , where the index  $j$  is computed as:

$$j = 1 + \sum_{\substack{n=1 \\ n \neq k}}^M (i_n - 1) \prod_{\substack{m=1 \\ m \neq k}}^{n-1} I_m. \quad (1)$$

**Definition 2.** Let  $\varphi$  be a linear map from an  $n$ -dimensional vector space  $V$  to an  $m$ -dimensional vector space  $U$ . Suppose  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  is a basis for  $V$  and  $(\beta_1, \beta_2, \dots, \beta_m)$  is a basis for  $U$ . For each basis vector  $\alpha_i$  in  $V$ , its image under  $\varphi$ , denoted by  $\varphi(\alpha_i)$ , can be written as a unique linear combination of the basis vectors in  $U$ . Explicitly, we have:

$$(\varphi(\alpha_1), \varphi(\alpha_2), \dots, \varphi(\alpha_n)) = (\beta_1, \beta_2, \dots, \beta_m) \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}. \quad (2)$$

Here, we define the  $m \times n$  matrix on the right as  $\mathbf{A}$ , which represents the linear map  $\varphi$  with respect to two given bases. Let  $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_n)^\top$  be a coordinate vector based on  $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_n)$ . The corresponding coordinate vector  $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_m)^\top$ , expressed in terms of  $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_m)$  after applying the linear map, is given by:

$$\boldsymbol{\mu} = \mathbf{A}\boldsymbol{\lambda}. \quad (3)$$

**Definition 3.** Let  $\mathbf{A}$  be an  $m \times n$  matrix. The rank of  $\mathbf{A}$ , denoted as  $\text{rank}(\mathbf{A})$ , is defined as the maximum number of linearly independent columns (or rows) in the matrix. Thus,  $\text{rank}(\mathbf{A}) \leq \min(m, n)$ . If  $U$  is the vector space spanned by the column vectors of  $\mathbf{A}$ , then the dimension of  $U$  is  $\text{rank}(\mathbf{A})$ .

**Definition 4.** Let  $\mathbf{A}$  be an  $m \times n$  coefficient matrix,  $\mathbf{x}$  be an  $n$ -dimensional vector of variables, and  $\mathbf{b}$  be an  $m$ -dimensional vector of constants. Then, the non-homogeneous system of linear equations can be expressed in the form  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . Let  $\overline{\mathbf{A}} = (\mathbf{A}|\mathbf{b})$  be the augmented matrix. The number of solutions  $N$  under different conditions is represented as follows:

$$N = \begin{cases} 1, & \text{rank}(\mathbf{A}) = \text{rank}(\overline{\mathbf{A}}) = n \\ \infty, & \text{rank}(\mathbf{A}) = \text{rank}(\overline{\mathbf{A}}) < n \\ 0, & \text{rank}(\mathbf{A}) \neq \text{rank}(\overline{\mathbf{A}}) \end{cases}. \quad (4)$$

### B. Comparison between 2D & 3D Modeling

Most DL-based methods treat image fusion as a 2D problem. They typically encode spectral information into feature map channels using a standard 2D convolutional layer. Let's denote the spectral data as  $\mathcal{X} \in \mathbb{R}^{H \times W \times L}$  (with  $H$ ,  $W$ , and  $L$  representing the height, width, and number of spectral bands, respectively), the resulting feature map as  $\mathcal{Y} \in \mathbb{R}^{H \times W \times C}$  (with  $C$  indicating the number of channels in the feature map), and the convolutional layer as  $f(\cdot)$ . Then, the encoding process of spectral information can be described as follows:

$$\mathcal{Y} = f(\mathcal{X}). \quad (5)$$

If the kernel size is  $1 \times 1$ , we perform mode-3 unfolding on  $\mathcal{X}$  and  $\mathcal{Y}$  to express the function  $f(\cdot)$  mathematically. This results in matrices  $\mathbf{X} \in \mathbb{R}^{L \times (HW)}$  and  $\mathbf{Y} \in \mathbb{R}^{C \times (HW)}$ . Given that each spectral band corresponds to a distinct wavelength range, the columns of  $\mathbf{X}$  can be considered as coordinate vectors with respect to a basis  $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_L)$ , where  $\boldsymbol{\alpha}_i \in \mathbb{R}^L$ . Additionally, the columns of  $\mathbf{Y}$  can be viewed as coordinate vectors in terms of the standard basis  $(\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_C)$ , where  $\mathbf{e}_i \in \mathbb{R}^C$ . Then, Equation 5 can be rewritten as:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}, \quad (6)$$

where  $\mathbf{A} \in \mathbb{R}^{C \times L}$  is the weight matrix of  $f(\cdot)$ . According to Definition 2,  $\mathbf{A}$  can be interpreted as the matrix representation of the linear map  $\psi$ , which is the abstract form of the function  $f(\cdot)$ .  $\psi$  maps spectral information into feature map channels with respect to  $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_L)$  and  $(\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_C)$ . Therefore,  $\psi(\boldsymbol{\alpha}_i)$  is precisely the  $i$ -th column of  $\mathbf{A}$ . As stated in Definition 3, the dimension of the vector space spanned by  $(\psi(\boldsymbol{\alpha}_1), \psi(\boldsymbol{\alpha}_2), \dots, \psi(\boldsymbol{\alpha}_L))$  is equal to  $\text{rank}(\mathbf{A})$ . When  $\text{rank}(\mathbf{A})$  is smaller than  $L$ , the dimension of the vector space

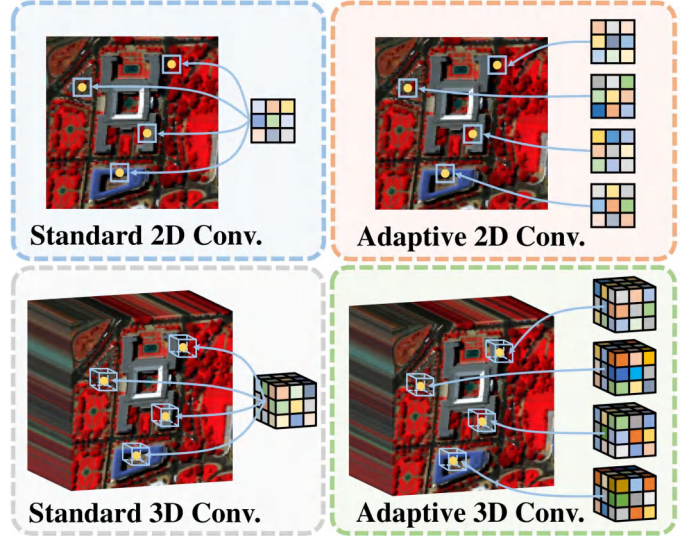


Fig. 3. Graphical illustration of standard 2D convolution, adaptive 2D convolution, standard 3D convolution, and adaptive 3D convolution.

spanned by  $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_L)$ , spectral information loss occurs. This happens because a lower-dimensional vector space possesses a reduced capacity to encapsulate the information contained in complex data. Additionally, a detailed explanation for the scenario where the kernel size is  $k \times k$  is provided in the supplementary material. We now present the following remark.

**Remark 1.** Approaching image fusion as a 2D problem may cause significant spectral distortions, particularly when the feature map has fewer channels than the spectral bands.

In the 3D modeling of image fusion, spectral data is stored in an additional dimension, maximizing the representation capability of spectral information. Experimental outcomes, as shown on the right side of Fig. 2, indicate that 3D modeling consistently reduces spectral distortions compared to 2D modeling. Furthermore, the left side of the figure visually demonstrates how 2D modeling causes spectral distortions.

### C. Comparison between Standard & Adaptive Convolutions

Given that image fusion aims to reconstruct a high-resolution image enriched with spectral information, the convolutional output should accurately reflect the physical properties of the scene rather than being an abstract representation of the input data. Consequently, we assume that each convolutional layer has a specific, intended output that aligns with this goal. In this context, we examine a simplified 2D convolution model, where the input data is represented as  $\tilde{\mathbf{A}} \in \mathbb{R}^{H \times W}$  and the target output as  $\mathbf{B} \in \mathbb{R}^{H \times W}$ . In a 2D convolutional layer with a kernel size of  $k \times k$ , the value of  $\mathbf{B}(i, j)$  is calculated as the weighted sum of the elements within a  $k \times k$  neighborhood centered at  $\tilde{\mathbf{A}}(i, j)$ . To mathematically express this operation, we begin by converting these neighborhoods into vectors. This conversion is achieved by stacking the elements of each neighborhood in a row-wise manner. The resulting vectors are then organized into an  $H \times W \times k^2$  tensor, denoted as  $\mathbf{A}$ . Next, we perform mode-3 unfolding on  $\mathbf{A}$  and transpose the

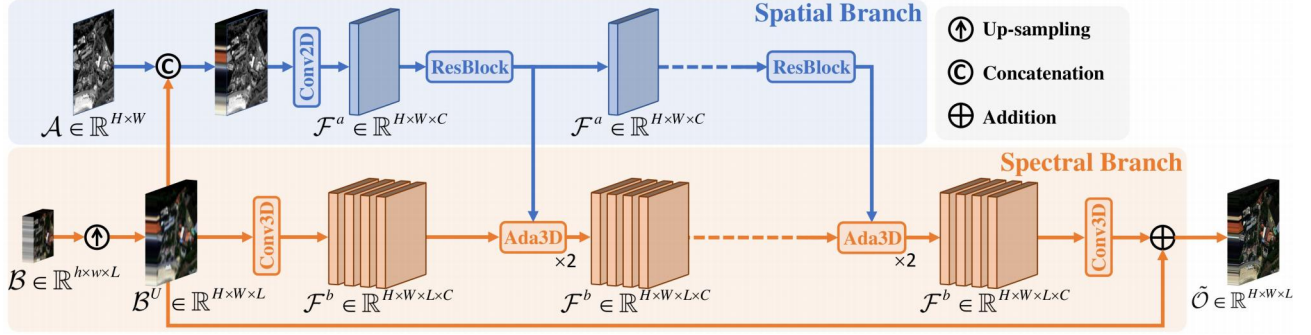


Fig. 4. Graphical illustration of our network architecture, which consists of two main sections: a spatial branch and a spectral branch. The former focuses on extracting spatial details, while the latter captures spectral characteristics and progressively achieves information integration using Ada3D blocks.

resulting matrix, yielding  $\mathbf{A} \in \mathbb{R}^{(HW) \times k^2}$ . Additionally, we flatten  $\mathcal{B}$  row-wise to obtain  $\mathbf{b} \in \mathbb{R}^{HW}$ .

In a standard convolutional layer, the same set of kernels is applied across all regions of the input. For simplicity, we denote the kernel weights as a vector  $\mathbf{x} \in \mathbb{R}^{k^2}$ . Thus, the standard 2D convolution can be mathematically expressed as:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (7)$$

According to Definition 4, Equation 17 can be viewed as a non-homogeneous system, with  $\mathbf{A}$  denoting the coefficient matrix,  $\mathbf{x}$  the vector of variables, and  $\mathbf{b}$  the vector of constants. Let  $\bar{\mathbf{A}} = (\mathbf{A}|\mathbf{b}) \in \mathbb{R}^{(HW) \times (k^2+1)}$  be the augmented matrix. When  $\text{rank}(\mathbf{A}) \neq \text{rank}(\bar{\mathbf{A}})$ , the system has no solution, indicating that the standard convolution cannot yield the desired output.

In an adaptive convolutional layer, each input region is processed with a unique set of kernels. We denote the weights of these adaptive kernels as  $\mathbf{X} \in \mathbb{R}^{(HW) \times k^2}$ . The simplified adaptive convolution can be mathematically expressed as:

$$\mathbf{A}(i, \cdot) \mathbf{X}(i, \cdot)^\top = \mathbf{b}(i), \quad i = 1, 2, \dots, HW. \quad (8)$$

If we consider  $\mathbf{A}(i, \cdot)$  and  $\bar{\mathbf{A}}(i, \cdot)$  as row matrices, we have  $\text{rank}(\mathbf{A}(i, \cdot)) = \text{rank}(\bar{\mathbf{A}}(i, \cdot)) \leq k^2$ . This relationship suggests that solutions exist theoretically, indicating that the adaptive convolution can generate the desired output.

For comprehensive explanations of the remaining cases, please refer to the supplementary material. Additionally, Fig. 3 provides a graphical illustration of standard and adaptive convolutions in both 2D and 3D domains for better understanding. In conclusion, we present the following remark.

**Remark 2.** *The kernel-sharing strategy of standard convolution limits its information processing capabilities compared to adaptive convolution, making it sub-optimal for image fusion.*

## IV. METHODOLOGY

### A. Notations

The PAN image is represented as  $\mathcal{A} \in \mathbb{R}^{H \times W}$ , where  $H$  and  $W$  denote its height and width, respectively. Additionally, the LRMS/LRHS image is represented as  $\mathcal{B} \in \mathbb{R}^{h \times w \times L}$ , with  $h = H/4$  and  $w = W/4$ , and  $L$  indicating the number of spectral bands. Furthermore, the up-sampled LRMS/LRHS image, the generated HRMS/HRHS image, and the ground-truth (GT)

image are denoted as  $\mathcal{B}^U \in \mathbb{R}^{H \times W \times L}$ ,  $\tilde{\mathcal{O}} \in \mathbb{R}^{H \times W \times L}$ , and  $\mathcal{O} \in \mathbb{R}^{H \times W \times L}$ , respectively. In general, our network takes  $\mathcal{A}$  and  $\mathcal{B}$  as inputs to produce an output  $\tilde{\mathcal{O}}$ , which is supervised by  $\mathcal{O}$ . The network comprises two main sections: a spatial branch and a spectral branch. The former generates spatial feature maps denoted by  $\mathcal{F}^a \in \mathbb{R}^{H \times W \times C}$ , where  $C$  represents the number of channels. In addition, the latter produces spectral feature maps denoted as  $\mathcal{F}^b \in \mathbb{R}^{H \times W \times L \times C}$ . Moreover, the adaptive 3D kernels have a size of  $k \times k \times k$ .

### B. Network Architecture

Since the primary focus of this study is not on the network architecture itself, we adopt a relatively simple design, as illustrated in Fig. 4, to validate the effectiveness of Ada3D. This architecture consists of two main sections: a spatial branch and a spectral branch. The former is responsible for extracting spatial details, while the latter captures spectral characteristics and progressively performs image fusion.

In the spatial branch, we start by concatenating  $\mathcal{A}$  and  $\mathcal{B}^U$  to integrate spatial information. Subsequently, a standard  $3 \times 3$  convolutional layer is applied to adjust the number of channels, producing the spatial feature map  $\mathcal{F}^a$ . Following that, we utilize a series of ResBlocks to extract spatial features.

In the spectral branch, we first up-sample  $\mathcal{B}$  using either Bicubic or PixelShuffle to enhance its spatial resolution, resulting in  $\mathcal{B}^U$ . Next, we apply a standard  $3 \times 3 \times 3$  convolutional layer to generate a feature vector for each voxel in  $\mathcal{B}^U$ , producing  $\mathcal{F}^b$ . Subsequently, a series of Ada3D blocks are employed to effectively merge spectral and spatial information. We then utilize another standard  $3 \times 3 \times 3$  convolutional layer to combine the feature vector values of each voxel. The result is added to  $\mathcal{B}^U$ , yielding the desired output,  $\tilde{\mathcal{O}}$ .

### C. Kernel Generator

The kernel generator is meticulously designed for the efficient production of adaptive 3D kernels that integrate both spatial and spectral information. As illustrated in Fig. 5, it consists of three components: a spatial kernel generator, a spectral kernel generator, and a combination section.

The spatial kernel generator focuses on creating spatial kernels from  $\mathcal{A}$ . To achieve this, we first employ two  $3 \times 3$  convolutional layers to extract spatial details and expand the

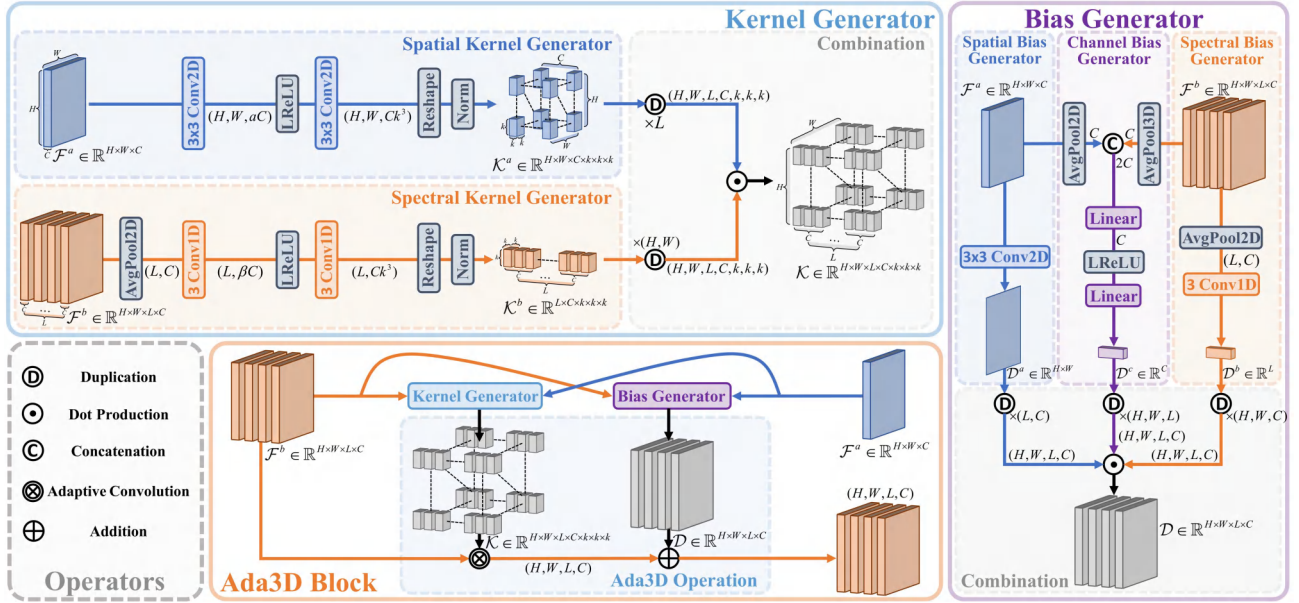


Fig. 5. The Ada3D block. Taking  $\mathcal{F}^a$  and  $\mathcal{F}^b$  as inputs, the kernel and bias generators produce adaptive kernels  $\mathcal{K}$  and biases  $\mathcal{D}$ . The acquired  $\mathcal{K}$  and  $\mathcal{D}$  are then applied to  $\mathcal{F}^b$ , facilitating the integration of spatial and spectral information. The 1D and 2D convolutional layers use kernel sizes of 3 and  $3 \times 3$ , both with a stride and padding of 1. Normalization is applied over each  $k \times k \times k$  kernel field, and the number of hidden channels is determined by  $\alpha$  and  $\beta$ .

TABLE I

COMPARISON OF PARAMETERS AND FLOPS ACROSS VARIOUS CONVOLUTIONAL PARADIGMS. THE NUMBER OF OUTPUT CHANNELS IS SET TO  $C$ .

Convolutions	Standard 3D	Depth-wise 3D	PAC3D	Ada3D (Ours)
Params	$C^2 k^3$	$Ck^3 + C^2$	$27C^3 k^3$	$3(3\alpha + \beta)C^2(k^3 + 1)$
FLOPs	$2HWLC^2 k^3$	$2HWLk^3 + 2HWLC^2$	$54HWLC^3 k^3$	$3HWLk^3 + 18\alpha HWC^2(k^3 + 1) + 6\beta LC^2(k^3 + 1)$
FLOPs / Params	$2HWL$	$2HWL$	$2HWL$	$\approx (HWL/C + 6\alpha HW + 2\beta L)/(3\alpha + \beta)$

channel number of  $\mathcal{F}^a$ , resulting in a feature map of size  $H \times W \times (Ck^3)$ . Notably, the channel count is  $Ck^3$  rather than  $C^2 k^3$ , indicating that we employ the group convolution technique with the group number set to  $C$ . Subsequently, this feature map is reshaped into a sixth-order tensor, which represents the spatial kernels, denoted as  $\mathcal{K}^a \in \mathbb{R}^{H \times W \times C \times k \times k \times k}$ . Essentially,  $\mathcal{K}^a$  can be interpreted as  $HW$  sets of 3D kernels, with each set corresponding to a unique spatial position.

The spectral kernel generator aims to produce spectral kernels from  $\mathcal{B}$ . To begin with, we apply 2D global average pooling to remove spatial information from  $\mathcal{F}^b$ , generating a matrix of size  $L \times C$ . Subsequently, we utilize two consecutive 1D convolutional layers with a kernel size of 3 to capture spectral information and increase the channel number to  $Ck^3$ . The resulting matrix is then reshaped into a fifth-order tensor, denoted as  $\mathcal{K}^b \in \mathbb{R}^{L \times C \times k \times k \times k}$ , which serves as the spectral kernels. In essence,  $\mathcal{K}^b$  can be interpreted as  $L$  sets of 3D kernels, each associated with a distinct spectral band.

In the combination section, we start by duplicating  $\mathcal{K}^a$  and  $\mathcal{K}^b$  to match the size of  $H \times W \times L \times C \times k \times k \times k$ . After that, we use a dot-product operation to sufficiently merge these two types of kernels, thereby creating adaptive 3D kernels represented as  $\mathcal{K} \in \mathbb{R}^{H \times W \times L \times C \times k \times k \times k}$ .

#### D. Bias Generator

The bias generator produces adaptive biases that enhance the convolutional output. As illustrated in Fig. 5, it consists of

four key components: a spatial bias generator, a spectral bias generator, a channel bias generator, and a combination section. Taking  $\mathcal{F}^a$  as input, the spatial bias generator creates spatial biases denoted as  $\mathcal{D}^a \in \mathbb{R}^{H \times W}$ . The spectral bias generator uses  $\mathcal{F}^b$  as input to produce spectral biases, represented as  $\mathcal{D}^b \in \mathbb{R}^L$ . In the channel bias generator,  $\mathcal{F}^a$  and  $\mathcal{F}^b$  are concatenated to form channel biases, denoted as  $\mathcal{D}^c \in \mathbb{R}^C$ . Finally, in the combination section,  $\mathcal{D}^a$ ,  $\mathcal{D}^b$ , and  $\mathcal{D}^c$  are integrated, similar to the process used in the kernel generator, resulting in adaptive biases represented as  $\mathcal{D} \in \mathbb{R}^{H \times W \times L \times C}$ .

#### E. Ada3D Block

The Ada3D block, shown in Fig. 5, is specifically designed to overcome the limitations of standard 3D convolution and achieve sufficient information integration. Given inputs  $\mathcal{F}^a$  and  $\mathcal{F}^b$ , the Ada3D block first produces  $\mathcal{K}$  and  $\mathcal{D}$  using kernel and bias generators. These are then applied to process  $\mathcal{F}^b$  through the Ada3D operation. Let  $\mathcal{D} \in \mathbb{R}^{C \times (HWL)}$ ,  $\mathcal{F} \in \mathbb{R}^{C \times (HWL)}$ , and  $\tilde{\mathcal{F}} \in \mathbb{R}^{C \times (HWL)}$  denote the mode-4 unfolding of  $\mathcal{D}$ ,  $\mathcal{F}^b$ , and the resulting feature map, respectively. The Ada3D operation is mathematically expressed as:

$$\tilde{\mathcal{F}}(\cdot, i) = \mathcal{D}(\cdot, i) + \sum_{j \in \Omega(i)} \mathcal{K}(p_i, \cdot, p_i - p_j) \odot \mathcal{F}(\cdot, j), \quad i = 1, 2, \dots, HWL. \quad (9)$$

Here,  $\Omega(i)$  is a collection of 1D positional indices that locate elements within the  $k \times k \times k$  convolution window centered around the  $i$ -th voxel of an  $H \times W \times L$  tensor. Additionally,

TABLE II

QUANTITATIVE EVALUATION RESULTS ON TESTING SAMPLES OF THE WDC, BOTSWANA, AND PAVIA DATASETS, WHICH BELONG TO THE HYPER-SPECTRAL PANSHARPENING TASK. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**, WHILE THE SECOND-BEST RESULTS ARE UNDERLINED. METHODS LISTED ABOVE THE DIVIDING LINE REPRESENT TRADITIONAL APPROACHES, WHEREAS THOSE BELOW IT ARE DL-BASED TECHNIQUES.

Methods	Params	WDC					Botswana					Pavia				
		PSNR	CC	SSIM	SAM	ERGAS	PSNR	CC	SSIM	SAM	ERGAS	PSNR	CC	SSIM	SAM	ERGAS
GLP [6]	—	27.946	0.934	0.761	6.546	5.110	32.559	0.951	0.837	1.383	1.207	31.944	0.935	0.749	6.099	4.909
CNMF [50]	—	24.604	0.890	0.678	8.441	6.682	30.220	0.917	0.788	1.934	1.718	31.184	0.894	0.659	6.953	6.263
Hysure [51]	—	25.598	0.913	0.718	7.254	5.834	30.610	0.928	0.796	1.747	1.595	32.208	0.921	0.730	6.240	5.474
HyperPNN [19]	0.13-0.14M	29.258	0.945	0.860	4.051	5.749	33.114	0.961	0.873	1.366	1.195	33.394	0.963	0.827	4.566	3.750
HSpeNet1 [52]	0.18-0.19M	29.634	0.960	0.870	4.039	4.266	31.746	0.942	0.844	1.456	1.663	33.612	0.964	0.824	4.690	3.721
HSpeNet2 [52]	0.11-0.13M	29.700	0.961	0.872	4.009	4.261	32.575	0.953	0.849	1.400	1.348	33.472	0.962	0.819	4.642	3.818
FusionNet [21]	0.21-0.26M	29.696	0.959	0.866	3.917	4.339	32.506	0.952	0.850	1.397	1.367	34.739	0.969	0.847	4.462	3.446
HyperDSNet [47]	0.18-0.31M	30.232	0.964	0.875	4.102	3.943	33.538	0.964	0.876	1.305	1.126	34.376	0.969	0.849	4.295	3.434
FPFNet [53]	3.00-3.06M	30.291	0.957	0.855	4.440	4.250	33.451	0.962	0.871	1.369	1.135	33.581	0.959	0.825	4.627	3.931
Pan-Mamba [46]	0.53-0.59M	30.794	0.968	0.884	3.632	3.693	33.901	0.965	0.882	1.286	1.089	34.057	0.967	0.857	4.508	3.689
ADWM [54]	0.50-0.62M	29.660	0.959	0.871	3.909	4.155	32.500	0.959	0.864	1.419	1.201	34.974	0.972	0.862	4.403	3.335
DFCFN [55]	0.50-0.57M	31.301	0.967	0.885	3.502	4.038	33.489	0.964	0.879	1.307	1.106	34.894	0.971	0.862	4.240	3.238
Ada3D (Ours)	0.55-0.58M	<b>31.998</b>	<b>0.971</b>	<b>0.893</b>	<b>3.461</b>	<b>3.414</b>	<b>34.290</b>	<b>0.969</b>	<b>0.889</b>	<b>1.230</b>	<b>1.034</b>	<b>35.752</b>	<b>0.973</b>	<b>0.873</b>	<b>4.038</b>	<b>3.142</b>
Ideal Values	—	+\infty	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	+\infty	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	+\infty	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>

$p_i$  and  $p_j$  denote the 3D positional indices of the  $i$ -th and  $j$ -th voxels.  $p_i - p_j$  is a 3D positional offset that locates a weight within the  $k \times k \times k$  kernel. Consequently,  $\mathcal{K}(p_i, \cdot, p_i - p_j) \in \mathbb{R}^C$  represents a weighting vector in  $\mathcal{K}$ . Furthermore,  $\odot$  defines the dot-product operation. Equation 9 illustrates that within an Ada3D block, each input voxel is convoluted with a distinct set of 3D kernels and refined by a particular set of biases.

#### F. Parameters & FLOPs Analysis

The learnable parameters of an Ada3D block primarily come from its spatial and spectral kernel generators. The former contributes  $9\alpha C^2(k^3 + 1)$  parameters, while the latter adds  $3\beta C^2(k^3 + 1)$  parameters. Consequently, the total parameter count of an Ada3D block is  $(9\alpha + 3\beta)C^2(k^3 + 1)$ . In terms of computational complexity, the spatial kernel generator, spectral kernel generator, combination section, and adaptive convolution operation require  $18\alpha HWC^2(k^3 + 1)$ ,  $6\beta LC^2(k^3 + 1)$ ,  $HWLCk^3$ , and  $2HWLCk^3$  FLOPs, respectively. Therefore, the total FLOP count is around  $3HWLCk^3 + 18\alpha HWC^2(k^3 + 1) + 6\beta LC^2(k^3 + 1)$ . Table I provides a comparison of Ada3D with standard 3D convolution, depth-wise 3D convolution, and PAC3D in terms of parameters and FLOPs. PAC3D, our proposed 3D extension of PAC [1], employs a standard  $3 \times 3 \times 3$  convolutional layer to generate adaptive 3D kernels, which leads to considerable computational cost. In practice, the ratio  $L/C$  is typically less than 5. Consequently, as demonstrated in the third row of Table I, Ada3D achieves lower FLOPs than other 3D convolutional paradigms with an equivalent number of parameters.

#### G. Loss function

The loss function for our method is crafted as a weighted sum of two distinct loss terms, as illustrated below:

$$\begin{cases} \mathcal{Loss} = \mathcal{L}_1 + \lambda_{ergas} \mathcal{L}_{ergas} \\ \mathcal{L}_{ergas} = \frac{1}{M} \sum_{m=1}^M \text{ERGAS}(\tilde{\mathcal{O}}^{\{m\}}, \mathcal{O}^{\{m\}}) \\ \text{ERGAS}(\tilde{\mathcal{O}}, \mathcal{O}) = \frac{100}{r} \sqrt{\frac{1}{L} \sum_{i=1}^L \frac{\text{MSE}(\tilde{\mathcal{O}}_i, \mathcal{O}_i)}{\mu_{\tilde{\mathcal{O}}_i}^2}} \end{cases} \quad (10)$$

Here,  $\mathcal{L}_1$  denotes the widely used sum of absolute differences (SAD).  $\lambda_{ergas}$  is a hyper-parameter associated with  $\mathcal{L}_{ergas}$ . ERGAS stands for the relative dimensionless global error in synthesis [56], while MSE represents the mean square error.  $M$  is the total number of training samples, and  $r$  is the up-sampling scale, which is set to 4 in this paper. Additionally,  $\tilde{\mathcal{O}}^{\{m\}}$  and  $\mathcal{O}^{\{m\}}$  refer to the  $m$ -th  $\tilde{\mathcal{O}}$  and  $\mathcal{O}$  in the training dataset.  $\tilde{\mathcal{O}}_i$  and  $\mathcal{O}_i$  denote the  $i$ -th spectral band in  $\tilde{\mathcal{O}}$  and  $\mathcal{O}$ . Furthermore,  $\mu^2$  represents the root mean square (RMS). In this paper, we utilize the ERGAS loss function to enhance the model's ability to capture and learn spectral information.

## V. EXPERIMENTS

### A. Hyper-Spectral Pansharpening

1) *Datasets*: We conduct experiments using three widely recognized hyper-spectral pansharpening datasets [5]: Washington D.C. (WDC), Botswana, and Pavia. The WDC dataset consists of images captured by the HYper-spectral Digital Imagery Collection Experiment (HY-DICE) sensor, which collects data across 210 spectral bands, covering wavelengths from 0.4 to 2.4  $\mu\text{m}$ , with a spatial resolution of 1 meter. The Botswana dataset comprises images acquired by the Hyperion sensor aboard the Earth Observing-1 (EO-1) satellite, which is operated by the National Aeronautics and Space Administration (NASA). This sensor records data over 242 spectral bands, spanning wavelengths from 0.4 to 2.5  $\mu\text{m}$ , with a spatial resolution of 30 meters. The Pavia dataset contains images captured by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor, which gathers data across 115 spectral bands, covering wavelengths from 0.4 to 0.9  $\mu\text{m}$ , with a spatial resolution of 1.3 meters. The hyper-spectral pansharpening datasets utilized in this study are sourced from the HyperPanCollection repository<sup>1</sup>. These datasets are generated following Wald's protocol [56], with a detailed explanation of the procedure provided in [47]. Specifically, the WDC dataset consists of 1024 training samples, with 90% designated for direct training purposes and 10% for validation purposes.

<sup>1</sup><https://github.com/liangjiandeng/HyperPanCollection>

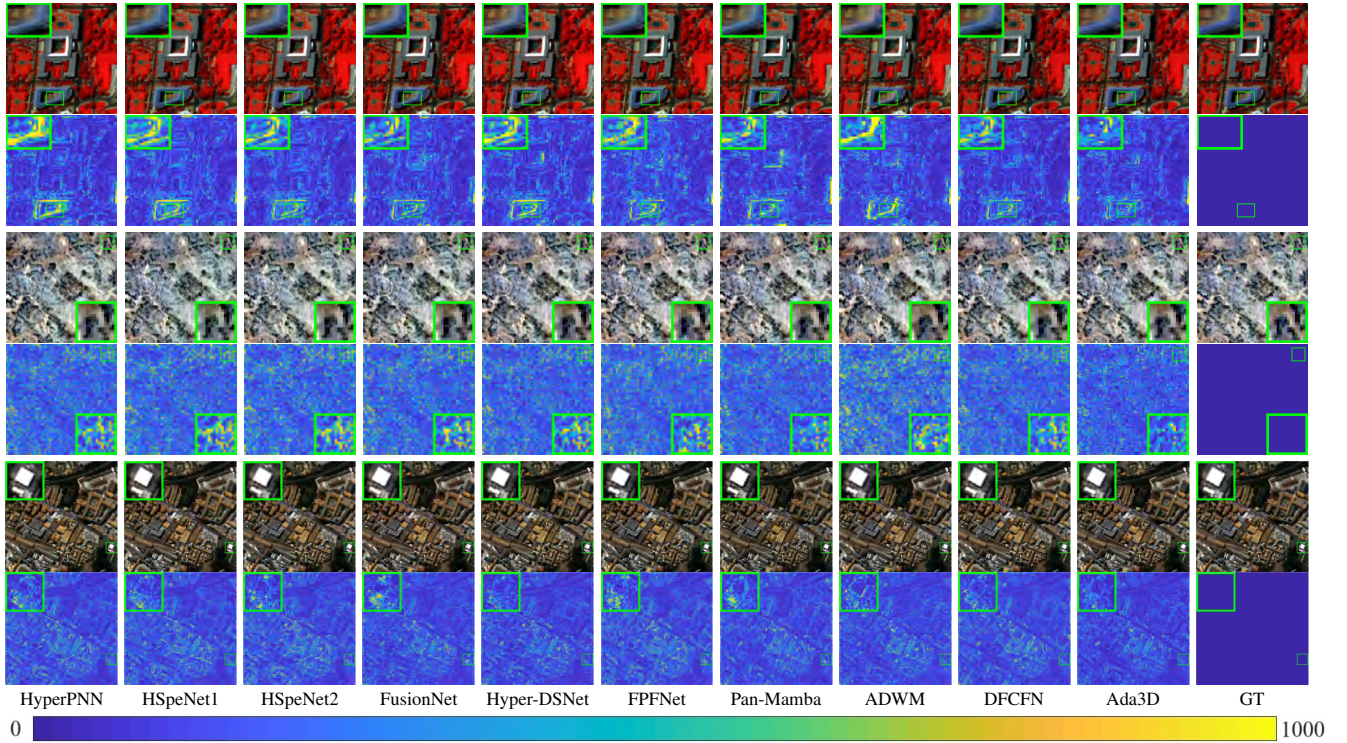


Fig. 6. Qualitative evaluation results for hyper-spectral pansharpening on the WDC, Botswana, and Pavia datasets. Row 1: Pseudo-color images of spectral bands 20, 50, and 80 for a testing sample from the WDC dataset. Row 2: Absolute error maps (AEMs) of spectral band 25 for the testing sample in Row 1. Row 3: Pseudo-color images of spectral bands 30, 50, and 70 for a testing sample from the Botswana dataset. Row 4: AEMs of spectral band 34 for the testing sample in row 3. Row 5: Pseudo-color images of spectral bands 20, 40, and 60 for a testing sample from the Pavia dataset. Row 6: AEMs of spectral band 68 for the testing sample in row 5. The values on both sides of the color bar represent the degree of errors. A darker AEM signifies a better result.

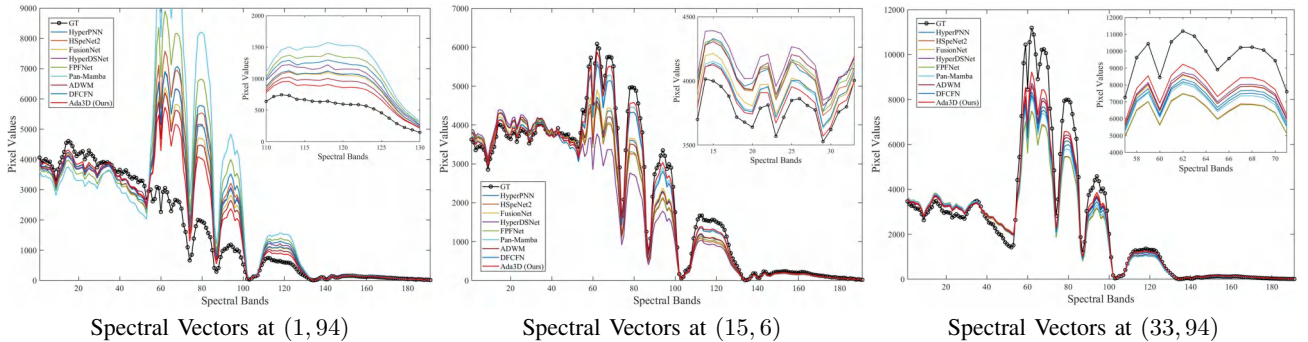


Fig. 7. Comparison of spectral vectors at three randomly selected spatial locations from a testing sample in the WDC dataset.

Additionally, it includes four testing samples. Each training sample is an image triplet in the PAN/LRHS/GT format, with sizes of  $64 \times 64$ ,  $16 \times 16 \times 191$ , and  $64 \times 64 \times 191$ , respectively. The testing samples are also in the PAN/LRHS/GT format, but with larger dimensions of  $128 \times 128$ ,  $32 \times 32 \times 191$ , and  $128 \times 128 \times 191$ , respectively. The Botswana dataset contains 967 training samples, with 83% allocated for direct training purposes and 17% for validation purposes. Additionally, there are four separate testing samples. Each training sample comprises a PAN/LRHS/GT image triplet with dimensions of  $64 \times 64$ ,  $16 \times 16 \times 145$ , and  $64 \times 64 \times 145$ , respectively. The testing samples consist of PAN/LRHS/GT image triplets sized  $128 \times 128$ ,  $32 \times 32 \times 145$ , and  $128 \times 128 \times 145$ . In the Pavia dataset, there are 1680 training samples, divided into 90% for direct training purposes and 10% for validation purposes. Additionally, this dataset includes four testing samples. Each

training sample contains a PAN/LRHS/GT image triplet of sizes  $64 \times 64$ ,  $16 \times 16 \times 102$ , and  $64 \times 64 \times 102$ , respectively. The testing samples comprise PAN/LRHS/GT image triplets with sizes of  $400 \times 400$ ,  $100 \times 100 \times 102$ , and  $400 \times 400 \times 102$ .

2) *Benchmarks*: Ada3D is compared against several representative techniques, including three traditional approaches: GLP [6], CNMF [50], and Hysure [51]; as well as eight DL-based methods: HyperPNN [19], the HSpeNet series [52], FusionNet [21], HyperDSNet [47], PPFNet [53], Pan-Mamba [46], ADWM [54], and DFCFN [55]. For a fair comparison, all DL-based methods are trained under identical conditions, utilizing the same Nvidia 3090 GPU and PyTorch framework.

3) *Quality Indices*: To align with the research standards for hyper-spectral pansharpening, we select five widely recognized quality indices for evaluation, namely PSNR [61], cross-correlation (CC), SSIM [61], SAM [62], and ERGAS [56].

TABLE III

QUANTITATIVE EVALUATION RESULTS ON 20 REDUCED-RESOLUTION AND 20 FULL-RESOLUTION SAMPLES FROM THE WV3 DATASET, WHICH BELONGS TO THE PANSHARPENING TASK. THE SYMBOL \* REPRESENTS DL-BASED METHODS THAT UTILIZE ADAPTIVE 2D CONVOLUTIONS.

Methods	Params	Reduced-Resolution				Full-Resolution (Real Data)		
		PSNR( $\pm$ std)	Q2n( $\pm$ std)	SAM( $\pm$ std)	ERGAS( $\pm$ std)	$D_\lambda$ ( $\pm$ std)	$D_s$ ( $\pm$ std)	QNR( $\pm$ std)
TV [12]	—	32.381 $\pm$ 2.328	0.795 $\pm$ 0.120	5.692 $\pm$ 1.808	4.855 $\pm$ 1.434	0.0234 $\pm$ 0.0061	0.0393 $\pm$ 0.0227	0.9383 $\pm$ 0.0269
GLP-HPM [10]	—	33.095 $\pm$ 2.800	0.835 $\pm$ 0.092	5.333 $\pm$ 1.761	4.616 $\pm$ 1.503	0.0206 $\pm$ 0.0082	0.0630 $\pm$ 0.0284	0.9180 $\pm$ 0.0346
BDS-PC [8]	—	32.970 $\pm$ 2.784	0.829 $\pm$ 0.097	5.428 $\pm$ 1.822	4.697 $\pm$ 1.617	0.0625 $\pm$ 0.0235	0.0730 $\pm$ 0.0356	0.8698 $\pm$ 0.0531
PNN [17]	0.10M	37.313 $\pm$ 2.646	0.893 $\pm$ 0.092	3.677 $\pm$ 0.762	2.681 $\pm$ 0.647	0.0213 $\pm$ 0.0080	0.0428 $\pm$ 0.0147	0.9369 $\pm$ 0.0212
MSDCNN [57]	0.23M	37.068 $\pm$ 2.686	0.890 $\pm$ 0.090	3.777 $\pm$ 0.803	2.760 $\pm$ 0.689	0.0230 $\pm$ 0.0091	0.0467 $\pm$ 0.0199	0.9316 $\pm$ 0.0271
BDPN [58]	1.49M	36.191 $\pm$ 2.702	0.871 $\pm$ 0.100	4.201 $\pm$ 0.857	3.046 $\pm$ 0.732	0.0364 $\pm$ 0.0142	0.0459 $\pm$ 0.0192	0.9196 $\pm$ 0.0308
FusionNet [21]	0.08M	38.047 $\pm$ 2.589	0.904 $\pm$ 0.090	3.324 $\pm$ 0.698	2.465 $\pm$ 0.644	0.0239 $\pm$ 0.0090	0.0364 $\pm$ 0.0137	0.9406 $\pm$ 0.0197
MUCNN [59]	2.32M	38.262 $\pm$ 2.703	0.911 $\pm$ 0.089	3.206 $\pm$ 0.681	2.400 $\pm$ 0.617	0.0258 $\pm$ 0.0111	0.0327 $\pm$ 0.0140	0.9424 $\pm$ 0.0205
LAGNet* [35]	0.15M	38.592 $\pm$ 2.778	0.910 $\pm$ 0.091	3.103 $\pm$ 0.558	2.292 $\pm$ 0.607	0.0368 $\pm$ 0.0148	0.0418 $\pm$ 0.0152	0.9230 $\pm$ 0.0247
PMACNet [60]	0.94M	38.595 $\pm$ 2.882	0.912 $\pm$ 0.092	3.073 $\pm$ 0.623	2.293 $\pm$ 0.532	0.0540 $\pm$ 0.0232	0.0336 $\pm$ 0.0115	0.9143 $\pm$ 0.0281
U2Net [25]	0.66M	39.117 $\pm$ 3.009	0.920 $\pm$ 0.085	2.888 $\pm$ 0.581	2.149 $\pm$ 0.525	0.0178 $\pm$ 0.0072	0.0313 $\pm$ 0.0075	0.9514 $\pm$ 0.0115
Pan-Mamba [46]	0.48M	39.012 $\pm$ 2.986	0.920 $\pm$ 0.085	2.914 $\pm$ 0.592	2.184 $\pm$ 0.521	0.0183 $\pm$ 0.0071	0.0307 $\pm$ 0.0108	0.9516 $\pm$ 0.0146
CANNet* [37]	0.78M	39.003 $\pm$ 2.900	0.919 $\pm$ 0.084	2.941 $\pm$ 0.590	2.175 $\pm$ 0.530	0.0196 $\pm$ 0.0083	0.0301 $\pm$ 0.0074	0.9510 $\pm$ 0.0126
ADWM [54]	0.40M	39.137 $\pm$ 2.894	0.919 $\pm$ 0.086	2.927 $\pm$ 0.589	2.154 $\pm$ 0.539	0.0261 $\pm$ 0.0105	0.0286 $\pm$ 0.0135	0.9462 $\pm$ 0.0191
Ada3D (Ours)	1.68M	<b>39.327<math>\pm</math>2.871</b>	<b>0.922<math>\pm</math>0.085</b>	<b>2.853<math>\pm</math>0.569</b>	<b>2.099<math>\pm</math>0.512</b>	<b>0.0168<math>\pm</math>0.0079</b>	<b>0.0282<math>\pm</math>0.0061</b>	<b>0.9554<math>\pm</math>0.0111</b>
Ideal Values	—	$+\infty$	1	0	0	0	0	1

The ideal values for these indices are  $+\infty$ , 1, 1, 0, and 0.

4) *Settings*: This paper initially assumes an equal number of channels for spatial and spectral feature maps to simplify the description. However, in practice, we allocate different channel counts to enhance the flexibility of network design. Specifically, for the hyper-spectral pansharpening task, we configure the spatial and spectral feature maps with 32 and 8 channels, respectively. The values of  $k$ ,  $\alpha$ , and  $\beta$  are set to 3, 1, and 1, respectively. Additionally, we use Bicubic interpolation for up-sampling. Furthermore, we apply three ResBlocks for spatial feature extraction and employ six Ada3D blocks to effectively integrate spatial and spectral information. All learnable weights are initialized using the Kaiming uniform distribution [63]. During the training of our networks on the WDC, Botswana, and Pavia datasets, we set the number of epochs to 1200, 2000, and 2000. Additionally, the batch size and initial learning rate are consistently set to 4 and  $5 \times 10^{-4}$  across all datasets. Furthermore, we utilize the Adam optimizer, with the learning rate halved every 250 epochs.

5) *Results*: The quantitative evaluation results for the three distinct datasets are summarized in Table II. Evidently, our method significantly outperforms other techniques across all quality indices. In addition to these results, the qualitative evaluation in Fig. 6 shows that Ada3D produces the darkest absolute error maps (AEMs). Moreover, Fig. 7 presents spectral vectors from three different spatial locations in a WDC testing sample, demonstrating the minimal spectral distortions achieved by our method. These findings highlight Ada3D’s superiority in the hyper-spectral pansharpening task.

## B. Pansharpening

1) *Datasets*: For the pansharpening task [4], we perform experiments using the popular WorldView-3 (WV3) dataset, which contains imagery captured by the sensor aboard the WV3 satellite. This sensor collects data across eight spectral bands, spanning wavelengths from 0.4 to 1  $\mu$ m, with a spatial resolution of 1.2 meters. The dataset employed in this study

is sourced from the PanCollection<sup>2</sup>, and the data generation process strictly follows Wald’s protocol [56]. A comprehensive description of this process can be found in [64]. Specifically, the WV3 dataset consists of 10000 training samples, with 90% designated for direct training purposes and 10% for validation purposes. It also includes 20 reduced-resolution and 20 full-resolution testing samples. Each training sample comprises an image triplet in the PAN/LRMS/GT format, with sizes of  $64 \times 64$ ,  $16 \times 16 \times 8$ , and  $64 \times 64 \times 8$ , respectively. The reduced-resolution testing samples consist of PAN/LRMS/GT image triplets sized  $256 \times 256$ ,  $64 \times 64 \times 8$ , and  $256 \times 256 \times 8$ , respectively. In addition, the full-resolution testing samples contain image pairs in the PAN/LRMS format, with dimensions of  $512 \times 512$  and  $128 \times 128 \times 8$ , respectively.

2) *Benchmarks*: We compare Ada3D with several representative pansharpening techniques, including three traditional approaches: TV [12], GLP-HPM [10], and BDS-PC [8]; and eleven DL-based methods: PNN [17], MSDCNN [57], BDPN [58], FusionNet [21], MUCNN [59], LAGNet [35], PMACNet [60], U2Net [25], Pan-Mamba [46], CANNet [37], and ADWM [54]. For fairness, all DL-based methods are trained in the same Nvidia 3090 GPU and PyTorch environment.

3) *Quality Indices*: In line with the research standards for pansharpening, we utilize four quality indices to assess the results on reduced-resolution samples: PSNR, Q2n [65], SAM, and ERGAS. The ideal values for these indices are  $+\infty$ , 1, 0, and 0, respectively. For full-resolution samples, we employ  $D_\lambda$ ,  $D_s$ , and QNR [48] as evaluation metrics, with ideal values of 0, 0, and 1. Notably, QNR, which integrates  $D_\lambda$  and  $D_s$ , offers a comprehensive measure of overall fusion quality.

4) *Settings*: For the pansharpening task, we configure both the spatial and spectral feature maps with 48 channels. The values of  $k$ ,  $\alpha$ , and  $\beta$  are set to 3, 0.25, and 0.25, respectively. Additionally, we select PixelShuffle for up-sampling. Furthermore, we utilize four ResBlocks to extract spatial features and stack eight Ada3D blocks for information fusion. All learnable weights are initialized using the Kaiming uniform distribution.

<sup>2</sup><https://github.com/liangjiandeng/PanCollection>

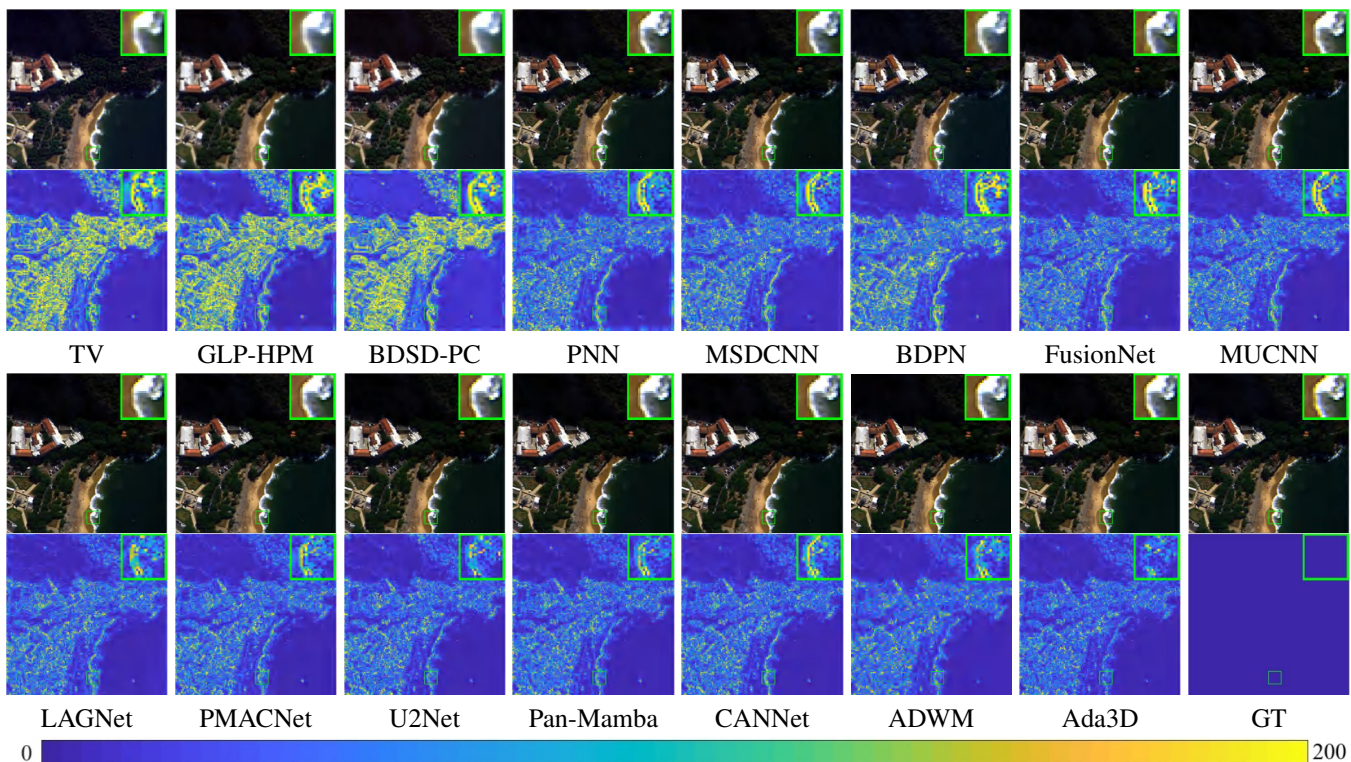


Fig. 8. Qualitative evaluation results on a reduced-resolution example from the WV3 dataset, which belongs to the pansharpening task. Rows 1 and 3: Pseudo-color images representing spectral bands 1, 3, and 5. Rows 2 and 4: The corresponding AEMs for spectral band 7.

TABLE IV  
ABLATION STUDY OF OUR NETWORK DESIGN USING TESTING SAMPLES FROM THE WDC DATASET FOR HYPER-SPECTRAL PANSHARPENING.

Methods	Params	FLOPs	PSNR	CC	SSIM	SAM	ERGAS
w/o ResBlocks	0.52M	2.41G	31.842	0.970	0.892	3.543	3.466
Ada3D-tiny	0.18M	0.82G	31.382	<b>0.971</b>	0.891	3.529	3.434
Ada3D-small	0.37M	1.66G	31.641	0.970	0.891	3.465	3.427
Ada3D-large	1.26M	5.44G	31.823	0.969	0.889	3.592	3.562
The Proposed	0.58M	2.64G	<b>31.998</b>	<b>0.971</b>	<b>0.893</b>	<b>3.461</b>	<b>3.414</b>

During training, we configure the number of epochs, batch size, initial learning rate, and  $\lambda_{ergas}$  as 500, 8,  $1 \times 10^{-4}$ , and  $1 \times 10^{-4}$ . In addition, we employ the Adam optimizer and reduce the learning rate by half at the 300-th epoch.

5) *Results*: The quantitative evaluation results for the WV3 dataset are summarized in Table III. It is evident that the proposed Ada3D outperforms all other techniques on both reduced-resolution and full-resolution testing samples. With indicator values nearing their limits, our method demonstrates substantial improvements over other approaches. Additionally, the qualitative evaluation outcomes, as shown in Fig. 8, illustrate that Ada3D produces the darkest AEMs. These findings demonstrate that Ada3D excels in the pansharpening task.

### C. Ablation Studies

1) *Network Design*: We evaluate the effectiveness of our network design using testing samples from the WDC dataset. To assess the impact of ResBlocks on spatial feature extraction, we conduct experiments without them (w/o ResBlocks). Additionally, we develop three variants of Ada3D that share

TABLE V  
ABLATION STUDY OF THE ADA3D BLOCK ON THE WDC DATASET. ALL COMPARED METHODS HAVE AN IDENTICAL NUMBER OF PARAMETERS.

Methods	FLOPs	PSNR	CC	SSIM	SAM	ERGAS
Ada3D-rev	0.69G	30.069	0.961	0.874	4.128	3.971
w/o Spatial Kernels	2.58G	30.852	0.963	0.876	3.912	3.878
w/o Spectral Kernels	2.73G	30.691	0.961	0.870	4.135	3.976
w/o Spatial Biases	2.63G	<b>31.907</b>	<b>0.971</b>	<b>0.891</b>	<b>3.518</b>	<b>3.448</b>
w/o Spectral Biases	2.60G	31.829	0.970	0.889	3.682	3.555
w/o Channel Biases	2.63G	31.846	0.969	<b>0.891</b>	3.611	3.539
w/o Adaptive Biases	2.59G	31.742	0.969	0.890	3.617	3.585
w/o Group Convolution	5.40G	31.623	0.970	0.889	3.556	3.527
The Proposed	2.64G	<b>31.998</b>	<b>0.971</b>	<b>0.893</b>	<b>3.461</b>	<b>3.414</b>

the same architectural design but vary in model size (Ada3D-tiny, Ada3D-small, and Ada3D-large). This helps verify the correctness of our network configuration. As shown in Table IV, the proposed method achieves the best results across all quality indices, demonstrating the validity of our design.

2) *Ada3D Block*: To validate the effectiveness of the Ada3D block, we develop eight variants and evaluate them using testing samples from the WDC dataset. For fairness, all compared methods are designed with the same number of network parameters. First, we create spatial kernels from the spectral feature map and spectral kernels from the spatial feature map (Ada3D-rev). This variant helps verify the correctness of our kernel generation strategy. Next, we examine the efficacy of the spatial and spectral kernels. Since directly removing spatial kernels would eliminate spatial information, we develop a new pipeline that integrates spatial and spectral characteristics through a dot-product operation. Then, the spectral kernels are duplicated to match the size of the adaptive

TABLE VI

COMPARISON WITH OTHER CONVOLUTIONAL PARADIGMS USING TESTING SAMPLES FROM THE WDC DATASET. ALL COMPARED METHODS ARE CONFIGURED WITH THE SAME NUMBER OF NETWORK PARAMETERS. THE TECHNIQUES ABOVE THE DIVIDING LINE REPRESENT 2D METHODS, WHILE THOSE BELOW CORRESPOND TO 3D APPROACHES.

Methods	FLOPs	PSNR	CC	SSIM	SAM	ERGAS
<b>Conv2D</b>	2.42G	30.918	0.964	0.875	3.842	3.874
<b>Ada2D</b>	1.89G	30.890	0.966	0.886	3.630	3.786
<b>Conv3D</b>	327G	31.696	0.963	0.876	<b>3.425</b>	4.183
<b>DW3D</b>	81.9G	31.261	0.964	0.877	3.475	4.110
<b>GPAC3D</b>	342G	31.874	0.969	0.891	3.576	3.524
<b>D2Conv3D [2]</b>	658G	31.302	0.969	0.888	3.530	3.536
<b>ST-A3DNet [3]</b>	14.3G	31.000	0.970	0.886	3.465	5.817
<b>The Proposed</b>	2.64G	<b>31.998</b>	<b>0.971</b>	<b>0.893</b>	3.461	<b>3.414</b>

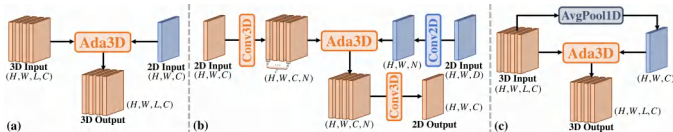


Fig. 9. Graphical illustration of Ada3D’s applications. (a) The original Ada3D block. (b) Ada3D used as an information integration module, merging two 2D inputs of sizes  $H \times W \times C$  and  $H \times W \times D$ . Here,  $D$  and  $N$  represent the number of channels in the 2D and 3D feature maps, respectively. (c) Ada3D functioning as a single-input 3D feature extraction block.

3D kernels (w/o Spatial Kernels). Additionally, we directly remove the spectral kernels to evaluate their effectiveness (w/o Spectral Kernels). Furthermore, we investigate the effect of excluding various biases, including spatial biases (w/o Spatial Biases), spectral biases (w/o Spectral Biases), channel biases (w/o Channel Biases), and all adaptive biases (w/o Adaptive Biases). Lastly, we examine the role of group convolution by removing it from the design (w/o Group Convolution). The quantitative results, as presented in Table V, strongly affirm the validity of our structural design for the Ada3D block.

3) *Comparison with Other Convolutional Paradigms*: We compare Ada3D with several convolutional paradigms, including standard 2D convolution (Conv2D), adaptive 2D convolution (Ada2D), standard 3D convolution (Conv3D), depth-wise 3D convolution (DW3D), GPAC3D, D2Conv3D [2], and ST-A3DNet [3]. Ada2D, based on the design in [36], serves as the 2D counterpart to Ada3D. Given the high computational cost associated with the PAC3D block, we adopt a more efficient group convolution version, which we call GPAC3D. In terms of network architecture, for 2D methods (Conv2D and Ada2D), we incorporate a spectral branch specifically designed for the 2D domain. For 3D approaches (all others), we replace the original Ada3D blocks in Fig. 4 with the corresponding 3D convolutional paradigms. Except for Ada2D, we apply a concatenation operation before the convolution block to integrate spatial and spectral information. To ensure fairness, all methods are configured to have an equivalent number of parameters. The quantitative evaluation outcomes on testing samples of the WDC dataset, as presented in Table VI, highlight two key findings: (i) 3D approaches generally outperform 2D methods but require significantly more computational resources; (ii) the proposed Ada3D achieves the best overall results, with FLOPs comparable to those of 2D methods. Consequently, our method combines the strengths of 2D and 3D convolutional paradigms.

TABLE VII

APPLICATIONS OF ADA3D IN VARIOUS PANSHARPENING FRAMEWORKS, EVALUATED ON REDUCED-RESOLUTION WV3 TESTING SAMPLES.

Methods	Params	PSNR	Q2n	SAM	ERGAS
<b>PNN [17]</b>	0.10M	37.313	0.893	3.677	2.681
<b>PNN + Ada3D</b>	0.12M	<b>37.914</b>	<b>0.902</b>	<b>3.331</b>	<b>2.479</b>
<b>FusionNet [21]</b>	0.08M	38.047	0.904	3.324	2.465
<b>FusionNet + Ada3D</b>	0.10M	<b>38.569</b>	<b>0.912</b>	<b>3.108</b>	<b>2.302</b>
<b>U2Net [25]</b>	0.66M	39.117	<b>0.920</b>	2.888	2.149
<b>U2Net + Ada3D</b>	0.67M	<b>39.183</b>	<b>0.920</b>	<b>2.885</b>	<b>2.129</b>

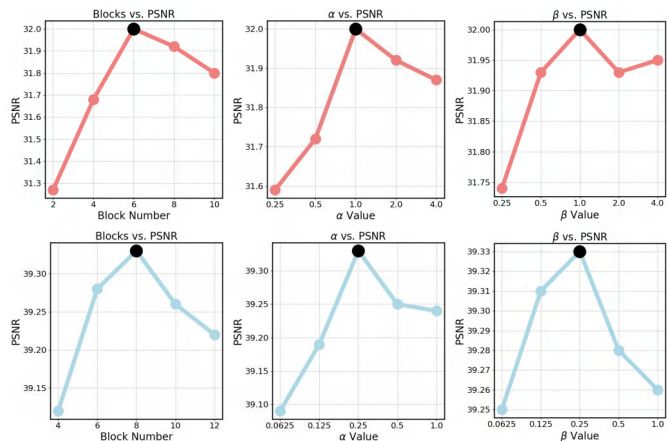


Fig. 10. Ablation study on three hyper-parameters: block number,  $\alpha$ , and  $\beta$ . The first row corresponds to the hyper-spectral pansharpening task, evaluated on the testing samples of the WDC dataset, while the second row pertains to the pansharpening task, evaluated on the reduced-resolution testing samples of the WV3 dataset. The black dots represent our chosen settings.

4) *The Applications of Ada3D*: The original Ada3D block is designed to fuse 2D and 3D inputs, which is relatively uncommon in practice. To expand its practical utility, we explore two additional applications of Ada3D, as illustrated in Fig. 9. First, the Ada3D block can serve as a plug-and-play module for integrating two 2D inputs, even with differing dimensions, by incorporating extra convolutional layers. Second, it can be repurposed for 3D feature extraction from a single input by applying average pooling across the third dimension of the 3D input. We examine the first additional application of Ada3D by embedding it into several representative pansharpening frameworks: PNN [17], FusionNet [21], and U2Net [25]. Specifically, we replace the concatenation operation in PNN and FusionNet, as well as the S2Block in U2Net, with the enhanced Ada3D block. Table VII presents the quantitative evaluation results using reduced-resolution samples from the WV3 dataset. Our method significantly improves performance, especially when baseline metrics are low. Even when the baseline performance is high, our approach exceeds the existing benchmarks. These findings indicate that the Ada3D block is an effective plug-and-play module for various architectures.

5) *Hyper-parameters*: We perform a comprehensive set of ablation studies to validate our hyper-parameter choices, focusing on three key parameters: the number of Ada3D blocks, the value of  $\alpha$ , and the value of  $\beta$ . As depicted in Fig. 10, our selected settings consistently produce the best performance, underscoring the soundness of our design decisions.

TABLE VIII

QUANTITATIVE EVALUATION RESULTS ON TESTING SAMPLES FROM THE CAVE DATASET, WHICH BELONGS TO THE HISR TASK. THE SYMBOL \* REPRESENTS METHODS UTILIZING 3D MODELING.

Methods	Params	PSNR	SSIM	SAM	ERGAS
ResTFNet [66]	2.39M	45.584	0.994	2.764	2.313
SSRNet [67]	0.03M	48.620	0.995	2.542	1.636
MoG-DCN* [68]	6.84M	51.688	0.997	1.971	1.104
Fusformer [69]	0.50M	49.983	0.994	2.203	2.534
3DT-Net [70]	3.16M	51.471	0.997	2.117	1.119
U2Net [25]	2.65M	50.433	0.997	2.187	1.277
Ada3D (Ours)	2.30M	<b>52.018</b>	<b>0.998</b>	<b>1.875</b>	<b>1.035</b>
Ideal Values	—	$+\infty$	1	0	0

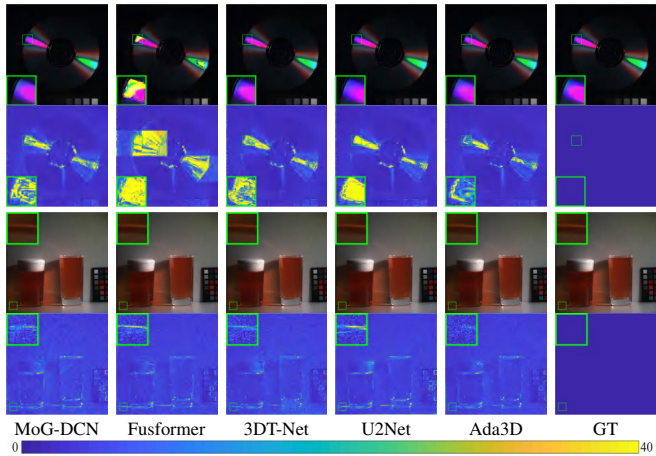


Fig. 11. Qualitative evaluation results of SOTA HISR methods on two testing samples from the CAVE dataset. Row 1: Pseudo-color images of spectral bands 6, 10, and 26 for the testing example *cd*. Row 2: AEMs of spectral band 26 for *cd*. Row 3: Pseudo-color images of spectral bands 6, 13, and 26 for the testing example *bears*. Row 4: AEMs of spectral band 20 for *bears*.

## VI. DISCUSSIONS

### A. Experiments for HISR

1) *Dataset*: To assess the effectiveness of Ada3D for image fusion tasks outside the realm of remote sensing, we conduct experiments using the CAVE dataset<sup>3</sup>, commonly employed for the HISR task [71]. The CAVE dataset, initially introduced in [72], contains 32 pairs of RGB and HRHS images, each with dimensions of  $512 \times 512 \times 3$  and  $512 \times 512 \times 31$ , respectively. These images are not directly suitable for training and testing purposes. Following the data processing approach described in [25], we transform this dataset into 3920 RGB/LRHS/GT training samples with sizes of  $64 \times 64 \times 3$ ,  $16 \times 16 \times 31$ , and  $64 \times 64 \times 31$ , respectively. Additionally, we create 11 RGB/LRHS/GT testing samples with dimensions of  $512 \times 512 \times 3$ ,  $128 \times 128 \times 31$ , and  $512 \times 512 \times 31$ , respectively.

2) *Benchmarks, Quality Indices, & Settings*: We compare Ada3D with six DL-based methods for HISR: ResTFNet [66], SSRNet [67], MoG-DCN [68], Fusformer [69], 3DT-Net [70], and U2Net [25]. To align with the research standards of HISR, we select four quality indices for evaluation, namely PSNR, SSIM, SAM, and ERGAS. The ideal values for these indices are  $+\infty$ , 1, 0, and 0. For detailed network configurations and training settings, please refer to the supplementary material.

<sup>3</sup><https://www.cs.columbia.edu/CAVE/databases/multispectral/>

TABLE IX

COMPARISON OF STRENGTHS ACROSS DIFFERENT CONVOLUTIONS.

Convolution	Spectral Preservation	Content Awareness	High Efficiency
Standard 2D	✗	✗	✓
Adaptive 2D	✗	✓	✓
Standard 3D	✓	✗	✗
PAC3D	✓	✓	✗
Ada3D (Ours)	✓	✓	✓

3) *Results*: The quantitative evaluation results in Table VIII demonstrate that our method consistently surpasses other approaches across all quality indices. Additionally, the qualitative assessment shown in Fig. 11 illustrates that Ada3D produces fusion outputs that most closely align with the GT. These findings provide compelling evidence of Ada3D’s effectiveness in image fusion tasks beyond the domain of remote sensing.

### B. Strengths & Limitations

1) *Strengths*: As shown in Table IX, our method surpasses other convolutional paradigms by simultaneously achieving spectral preservation, content awareness, and high efficiency. Ada3D preserves spectral information by treating image fusion as a 3D problem. Additionally, its content awareness is enhanced through the use of adaptive convolution. Moreover, Ada3D achieves high efficiency with its carefully designed kernel generator and the implementation of group convolution.

2) *Limitations*: Our method exhibits two primary limitations. First, Ada3D utilizes the group convolution technique, which constrains the output feature map to have the same number of channels as one of the input feature maps. Second, Ada3D requires a 3D input, making it unsuitable for applications involving feature extraction from a single 2D input.

## VII. CONCLUSION

In this paper, we mathematically demonstrate that treating image fusion as a 2D problem can lead to significant spectral distortions. We further show that adaptive convolution is superior to standard convolution in data processing. Based on these insights, we propose a novel convolutional paradigm called Ada3D for remote sensing image fusion. Ada3D assigns a unique set of 3D kernels to each input voxel, enabling the capture of fine-grained details. These adaptive kernels are generated through a two-step process that integrates both spatial and spectral information. In parallel, we produce adaptive biases to refine the convolutional outcome. Additionally, group convolution is employed to reduce computational complexity. We assess the performance of Ada3D on five datasets related to three image fusion tasks. Our method achieves SOTA results, highlighting the superiority of Ada3D for image fusion.

## ACKNOWLEDGMENTS

This research is supported by National Natural Science Foundation of China (12271083), and the Project of the Department of Science and Technology of Sichuan Province (Grant No. 2025YFNH0001).

## REFERENCES

- [1] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz, "Pixel-adaptive convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 158–11 167.
- [2] C. Schmidt, A. Athar, S. Mahadevan, and B. Leibe, "D2conv3d: Dynamic dilated convolutions for object segmentation in videos," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 1200–1209.
- [3] H. Li, X. Li, L. Su, D. Jin, J. Huang, and D. Huang, "Deep spatio-temporal adaptive 3d convolutional neural networks for traffic flow prediction," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 2, pp. 1–21, 2022.
- [4] G. Vivone, M. Dalla Mura, A. Garzelli, R. Restaino, G. Scarpa, M. O. Ulfarsson, L. Alparone, and J. Chanussot, "A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 1, pp. 53–81, 2021.
- [5] L. Loncan, L. B. de Almeida, J. M. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. A. Licciardi, M. Simões, J.-Y. Tourneret, M. A. Veganzones, G. Vivone, Q. Wei, and N. Yokoya, "Hyperspectral pansharpening: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 3, no. 3, pp. 27–46, 2015.
- [6] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, "Mtf-tailored multiscale fusion of high-resolution ms and pan imagery," *Photogrammetric Engineering & Remote Sensing*, vol. 72, no. 5, pp. 591–596, 2006.
- [7] J. Choi, K. Yu, and Y. Kim, "A new adaptive component-substitution-based satellite image fusion by using partial replacement," *IEEE Transactions on Geoscience and Remote Sensing*, 2010.
- [8] G. Vivone, "Robust band-dependent spatial-detail approaches for panchromatic sharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6421–6433, 2019.
- [9] J. Liu, "Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details," *International Journal of Remote Sensing*, vol. 21, no. 18, pp. 3461–3472, 2000.
- [10] G. Vivone, R. Restaino, M. Dalla Mura, G. Licciardi, and J. Chanussot, "Contrast and error-based fusion schemes for multispectral image pansharpening," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 5, pp. 930–934, 2014.
- [11] G. Vivone, R. Restaino, and J. Chanussot, "Full scale regression-based injection coefficients for panchromatic sharpening," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3418–3431, 2018.
- [12] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "A new pansharpening algorithm based on total variation," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 318–322, 2013.
- [13] G. Zhang, F. Fang, A. Zhou, and F. Li, "Pan-sharpening of multi-spectral images using a new variational model," *International Journal of Remote Sensing*, vol. 36, no. 5, pp. 1484–1508, 2015.
- [14] T. Wang, F. Fang, F. Li, and G. Zhang, "High-quality bayesian pansharpening," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 227–239, 2019.
- [15] H. Lu, Y. Yang, S. Huang, and W. Tu, "An efficient pansharpening approach based on texture correction and detail refinement," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [16] Y. Yang, H. Lu, S. Huang, W. Wan, and L. Li, "Pansharpening based on variational fractional-order geometry model and optimized injection gains," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 2128–2141, 2022.
- [17] M. Giuseppe, C. Davide, V. Luisa, and S. Giuseppe, "Pansharpening by convolutional neural networks," *Remote Sensing*, vol. 8, no. 7, p. 594, 2016.
- [18] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley, "Pannet: A deep network architecture for pan-sharpening," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1753–1761.
- [19] L. He, J. Zhu, J. Li, A. Plaza, J. Chanussot, and B. Li, "Hyperpnn: Hyperspectral pansharpening via spectrally predictive convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 3092–3100, 2019.
- [20] X. Fu, W. Wang, Y. Huang, X. Ding, and J. Paisley, "Deep multiscale detail networks for multiband spectral image sharpening," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2090–2104, 2021.
- [21] L.-J. Deng, G. Vivone, C. Jin, and J. Chanussot, "Detail injection-based deep convolutional neural networks for pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6995–7010, 2021.
- [22] M. Gong, J. Ma, H. Xu, X. Tian, and X.-P. Zhang, "D2tnet: A convlstm network with dual-direction transfer for pan-sharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
- [23] R. Dian, A. Guo, and S. Li, "Zero-shot hyperspectral sharpening," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 12 650–12 666, 2023.
- [24] Y. Xing, Y. Zhang, H. He, X. Zhang, and Y. Zhang, "Pansharpening via frequency-aware fusion network with explicit similarity constraints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, 2023.
- [25] S. Peng, C. Guo, X. Wu, and L.-J. Deng, "U2net: A general framework with spatial-spectral-integrated double u-net for image fusion," in *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM)*. New York, NY, USA: Association for Computing Machinery, 2023, p. 3219–3227.
- [26] H. Gao, S. Li, J. Li, and R. Dian, "Multispectral image pan-sharpening guided by component substitution model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.
- [27] Y. Jia, Q. Hu, R. Dian, J. Ma, and X. Guo, "Paps: Progressive attention-based pan-sharpening," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 391–404, 2024.
- [28] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 184–199.
- [29] H. Lin, Y. Li, X. Fu, X. Ding, Y. Huang, and J. Paisley, "Rain o'er me: Synthesizing real rain to derain with data distillation," *IEEE Transactions on Image Processing*, vol. 29, pp. 7668–7680, 2020.
- [30] Q. Zhang, Q. Yuan, M. Song, H. Yu, and L. Zhang, "Cooperated spectral low-rankness prior and deep spatial prior for hsi unsupervised denoising," *IEEE Transactions on Image Processing*, vol. 31, pp. 6356–6368, 2022.
- [31] Y. Xiao, Q. Yuan, K. Jiang, J. He, C.-W. Lin, and L. Zhang, "Ttst: A top-k token selective transformer for remote sensing image super-resolution," *IEEE Transactions on Image Processing*, vol. 33, pp. 738–752, 2024.
- [32] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Advances in neural information processing systems (NeurIPS)*, vol. 29, 2016, pp. 667–675.
- [33] J. Zamora Esquivel, A. Cruz Vargas, P. Lopez Meyer, and O. Tickoo, "Adaptive convolutional kernels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019, pp. 1998–2005.
- [34] J. Zhou, V. Jampani, Z. Pi, Q. Liu, and M.-H. Yang, "Decoupled dynamic filter networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6643–6652.
- [35] Z.-R. Jin, T.-J. Zhang, T.-X. Jiang, G. Vivone, and L.-J. Deng, "Lagconv: Local-context adaptive convolution kernels with global harmonic bias for pansharpening," in *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, vol. 36, 2022, pp. 1113–1121.
- [36] S. Peng, L.-J. Deng, J.-F. Hu, and Y. Zhuo, "Source-adaptive discriminative kernels based network for remote sensing pansharpening," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, 2022, pp. 1283–1289.
- [37] Y. Duan, X. Wu, H. Deng, and L.-J. Deng, "Content-adaptive non-local convolution for remote sensing pansharpening," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 27 738–27 747.
- [38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [39] H. Firat, H. Çiğ, M. T. Güllüoğlu, M. E. Asker, and D. Hanbay, "Multiscale feature fusion for hyperspectral image classification using hybrid 3d-2d depthwise separable convolution networks," *Traitement du Signal*, vol. 40, no. 5, p. 1921, 2023.
- [40] H. Firat, M. E. Asker, M. İ. Bayındır, and D. Hanbay, "Hybrid 3d/2d complete inception module and convolutional neural network for hyperspectral remote sensing image classification," *Neural Processing Letters*, vol. 55, no. 2, pp. 1087–1130, 2023.
- [41] H. Firat, M. E. Asker, M. İ. Bayındır, and D. Hanbay, "3d residual spatial-spectral convolution network for hyperspectral remote sensing image classification," *Neural Computing and Applications*, vol. 35, no. 6, pp. 4479–4497, 2023.

- [42] Z. Li, G. Chen, G. Li, L. Zhou, X. Pan, W. Zhao, and W. Zhang, "Dbanet: Dual-branch attention network for hyperspectral remote sensing image classification," *Computers and Electrical Engineering*, vol. 118, p. 109269, 2024.
- [43] Y. Zheng, S. Liu, H. Chen, and L. Bruzzone, "Hybrid fusionnet: A hybrid feature fusion framework for multisource high-resolution remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–14, 2024.
- [44] H. Su, H. Jin, and C. Sun, "Deep pansharpening via 3d spectral super-resolution network and discrepancy-based gradient transfer," *Remote Sensing*, vol. 14, no. 17, 2022.
- [45] E. Zhang, Y. Fu, J. Wang, L. Liu, K. Yu, and J. Peng, "Msac-net: 3d multi-scale attention convolutional network for multi-spectral imagery pansharpening," *Remote Sensing*, vol. 14, no. 12, p. 2761, 2022.
- [46] X. He, K. Cao, K. Yan, R. Li, C. Xie, J. Zhang, and M. Zhou, "Pan-mamba: Effective pan-sharpening with state space model," *arXiv preprint arXiv:2402.12192*, 2024.
- [47] Y.-W. Zhuo, T.-J. Zhang, J.-F. Hu, H.-X. Dou, T.-Z. Huang, and L.-J. Deng, "A deep-shallow fusion network with multidetail extractor and spectral attention for hyperspectral pansharpening," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 7539–7555, 2022.
- [48] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. A. Licciardi, R. Restaino, and L. Wald, "A critical comparison among pansharpening algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2565–2586, 2015.
- [49] J. Wu, D. Li, Y. Yang, C. Bajaj, and X. Ji, "Dynamic filtering with large sampling field for convnets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018, pp. 185–200.
- [50] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled non-negative matrix factorization (cnmf) for hyperspectral and multispectral data fusion: Application to pasture classification," in *2011 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2011, pp. 1779–1782.
- [51] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3373–3388, 2015.
- [52] L. He, J. Zhu, J. Li, D. Meng, J. Chanussot, and A. Plaza, "Spectral-fidelity convolutional neural networks for hyperspectral pansharpening," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5898–5914, 2020.
- [53] W. Dong, Y. Yang, J. Qu, Y. Li, Y. Yang, and X. Jia, "Feature pyramid fusion network for hyperspectral pansharpening," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2023.
- [54] J. Huang, H. Chen, J. Ren, S. Peng, and L. Deng, "A general adaptive dual-level weighting mechanism for remote sensing pansharpening," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 7447–7456.
- [55] X. Wang, Y. Yang, S. Huang, W. Wan, Z. Liu, L. Zhang, and A. Zhao, "Dfcfn: Dual-stage feature correction fusion network for hyperspectral pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–14, 2025.
- [56] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogrammetric Engineering and Remote Sensing*, vol. 63, pp. 691–699, 1997.
- [57] Y. Wei, Q. Yuan, X. Meng, H. Shen, L. Zhang, and M. Ng, "Multi-scale-and-depth convolutional neural network for remote sensed imagery pan-sharpening," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 3413–3416.
- [58] Y. Zhang, C. Liu, M. Sun, and Y. Ou, "Pan-sharpening using an efficient bidirectional pyramid network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5549–5563, 2019.
- [59] Y. Wang, L.-J. Deng, T.-J. Zhang, and X. Wu, "Ssconv: Explicit spectral-to-spatial convolution for pansharpening," in *Proceedings of the 29th ACM International Conference on Multimedia (ACM MM)*. New York, NY, USA: Association for Computing Machinery, 2021, p. 4472–4480.
- [60] Y. Liang, P. Zhang, Y. Mei, and T. Wang, "Pmacnet: Parallel multiscale attention constraint network for pan-sharpening," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [61] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [62] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm," in *JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, 1992.
- [63] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [64] L.-J. Deng, G. Vivone, M. E. Paoletti, G. Scarpa, J. He, Y. Zhang, J. Chanussot, and A. Plaza, "Machine learning in pansharpening: A benchmark, from shallow to deep networks," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 3, pp. 279–315, 2022.
- [65] A. Garzelli and F. Nencini, "Hypercomplex quality assessment of multi/hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 4, pp. 662–665, 2009.
- [66] X. Liu, Q. Liu, and Y. Wang, "Remote sensing image fusion based on two-stream fusion network," *Information Fusion*, vol. 55, pp. 1–15, 2020.
- [67] X. Zhang, W. Huang, Q. Wang, and X. Li, "Ssr-net: Spatial-spectral reconstruction network for hyperspectral and multispectral image fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5953–5965, 2021.
- [68] W. Dong, C. Zhou, F. Wu, J. Wu, G. Shi, and X. Li, "Model-guided deep hyperspectral image super-resolution," *IEEE Transactions on Image Processing*, vol. 30, pp. 5754–5768, 2021.
- [69] J.-F. Hu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, D. Hong, and G. Vivone, "Fusformer: A transformer-based fusion network for hyperspectral image super-resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [70] Q. Ma, J. Jiang, X. Liu, and J. Ma, "Learning a 3d-cnn and transformer prior for hyperspectral image super-resolution," *Information Fusion*, vol. 100, p. 101907, 2023.
- [71] X. Wang, Q. Hu, Y. Cheng, and J. Ma, "Hyperspectral image super-resolution meets deep learning: A survey and perspective," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 8, pp. 1668–1691, 2023.
- [72] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2241–2253, 2010.

# Supplementary Material

## I. SUPPLEMENTARY EXPLANATION FOR REMARK 1

First, we extend Definition 2 into matrix space as follows.

**Lemma 1.** *Let  $\varphi$  be a linear map from an  $n$ -dimensional matrix space  $V$  to an  $m$ -dimensional matrix space  $U$ . Suppose  $(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n)$  is a basis for  $V$  and  $(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_m)$  is a basis for  $U$ . For each basis matrix  $\mathbf{A}_i$  in  $V$ , its image under  $\varphi$ , denoted by  $\varphi(\mathbf{A}_i)$ , can be written as a unique linear combination of the basis matrices in  $U$ . Explicitly, we have:*

$$(\varphi(\mathbf{A}_1), \varphi(\mathbf{A}_2), \dots, \varphi(\mathbf{A}_n)) = (\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_m) \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ t_{21} & t_{22} & \dots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \dots & t_{mn} \end{pmatrix}. \quad (11)$$

We define the  $m \times n$  matrix on the right as  $\mathbf{T}$ , representing  $\varphi$  in terms of two given bases. Let  $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_n)^\top$  be a coordinate vector based on  $(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n)$ . The corresponding coordinate vector  $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_m)^\top$  based on  $(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_m)$  after the linear map is expressed as:

$$\boldsymbol{\mu} = \mathbf{T}\boldsymbol{\lambda}. \quad (12)$$

*Proof.* To start, we rewrite Equation 11 in the following form:

$$\begin{cases} \varphi(\mathbf{A}_1) = t_{11}\mathbf{B}_1 + t_{21}\mathbf{B}_2 + \dots + t_{m1}\mathbf{B}_m = \sum_{i=1}^m t_{i1}\mathbf{B}_i \\ \varphi(\mathbf{A}_2) = t_{12}\mathbf{B}_1 + t_{22}\mathbf{B}_2 + \dots + t_{m2}\mathbf{B}_m = \sum_{i=1}^m t_{i2}\mathbf{B}_i \\ \dots \\ \varphi(\mathbf{A}_n) = t_{1n}\mathbf{B}_1 + t_{2n}\mathbf{B}_2 + \dots + t_{mn}\mathbf{B}_m = \sum_{i=1}^m t_{in}\mathbf{B}_i \end{cases}. \quad (13)$$

Therefore, the image of the matrix  $\mathbf{A} = \lambda_1\mathbf{A}_1 + \lambda_2\mathbf{A}_2 + \dots + \lambda_n\mathbf{A}_n$  can be expressed as follows:

$$\begin{aligned} \varphi(\mathbf{A}) &= \lambda_1\varphi(\mathbf{A}_1) + \lambda_2\varphi(\mathbf{A}_2) + \dots + \lambda_n\varphi(\mathbf{A}_n) \\ &= \lambda_1\left(\sum_{i=1}^m t_{i1}\mathbf{B}_i\right) + \lambda_2\left(\sum_{i=1}^m t_{i2}\mathbf{B}_i\right) + \dots + \lambda_n\left(\sum_{i=1}^m t_{in}\mathbf{B}_i\right) \\ &= \left(\sum_{j=1}^n \lambda_j t_{1j}\right)\mathbf{B}_1 + \left(\sum_{j=1}^n \lambda_j t_{2j}\right)\mathbf{B}_2 + \dots + \left(\sum_{j=1}^n \lambda_j t_{mj}\right)\mathbf{B}_m. \end{aligned} \quad (14)$$

Consequently, the corresponding coordinate vector  $\boldsymbol{\mu}$  with respect to the basis  $(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_m)$  is given by:

$$\begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ t_{21} & t_{22} & \dots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \dots & t_{mn} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}. \quad (15)$$

This equation represents the expanded form of Equation 12. Therefore, Lemma 1 is mathematically verified.

Recall that we denote the spectral data and feature map as  $\mathcal{X} \in \mathbb{R}^{H \times W \times L}$  and  $\mathcal{Y} \in \mathbb{R}^{H \times W \times C}$ , respectively. Let the kernel size of  $f(\cdot)$  be  $k \times k$ . To formulate the function  $f(\cdot)$  mathematically, we undertake the following transformation. According to the properties of 2D convolution,  $\mathcal{Y}(i, j, \cdot)$  is determined by the vectors within a  $k \times k$  spatial window

centered at  $\mathcal{X}(i, j, \cdot)$ . For each window, we first organize these vectors into a  $k \times k \times L$  tensor. We then apply mode-3 unfolding to this tensor, which produces an  $L \times k^2$  matrix. This matrix is further flattened row-wise into an  $Lk^2$  vector. Collecting these vectors from all windows results in an  $H \times W \times (Lk^2)$  tensor, denoted as  $\hat{\mathcal{X}}$ . Finally, we conduct mode-3 unfolding on both  $\hat{\mathcal{X}}$  and  $\mathcal{Y}$ , generating  $\mathbf{X} \in \mathbb{R}^{(Lk^2) \times (HW)}$  and  $\mathbf{Y} \in \mathbb{R}^{C \times (HW)}$ , respectively. The columns of  $\mathbf{X}$  can be viewed as coordinate vectors with respect to a basis  $(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{Lk^2})$ , where each  $\mathbf{A}_i \in \mathbb{R}^{L \times k^2}$ . This basis spans a matrix space of dimension  $Lk^2$ . Besides, the columns of  $\mathbf{Y}$  can be seen as coordinate vectors related to the standard basis. Here, we represent this basis in the form of column matrices as  $(\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_C)$ , where  $\mathbf{E}_i \in \mathbb{R}^{C \times 1}$ . Let  $\mathcal{T} \in \mathbb{R}^{C \times L \times k \times k}$  denote the weights of  $f(\cdot)$ . We perform mode-1 unfolding on  $\mathcal{T}$ , resulting in a matrix represented as  $\mathbf{T} \in \mathbb{R}^{C \times (Lk^2)}$ . Therefore, Equation 5 in the main text can be reformulated as follows:

$$\mathbf{Y} = \mathbf{T}\mathbf{X}. \quad (16)$$

According to Lemma 1,  $\mathbf{T}$  can be interpreted as the matrix representation of the linear map  $\psi$ , which is the abstract form of  $f(\cdot)$ .  $\psi$  maps spectral information into feature map channels with respect to  $(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{Lk^2})$  and  $(\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_C)$ . Therefore,  $\psi(\mathbf{A}_i)$  is the column matrix that corresponds to the  $i$ -th column of  $\mathbf{T}$ . According to Definition 3, the dimension of the matrix space spanned by  $(\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_C)$  is equal to  $\text{rank}(\mathbf{T})$ . To accurately capture spectral information without distortion, it is necessary to use at least  $L$  base matrices from  $(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{Lk^2})$ . Therefore, if  $\text{rank}(\mathbf{T})$  is less than  $L$ , a loss of spectral information is unavoidable.

To further support Remark 1, we consider a more practical scenario by introducing a normalization layer and a non-linear activation function. Thus, Equation 16 can be rewritten as:

$$\mathbf{Y}_{final} = \sigma(\text{Norm}(\mathbf{Y})) = \sigma(\text{Norm}(\mathbf{T}\mathbf{X})), \quad (17)$$

where  $\text{Norm}(\cdot)$  denotes the normalization layer, and  $\sigma(\cdot)$  represents the non-linear activation function. Here,  $\mathbf{Y}_{final} \in \mathbb{R}^{C \times (HW)}$  is the final output. Importantly,  $\text{Norm}(\cdot)$  is an invertible affine transformation that rescales and centers  $\mathbf{Y}$  without changing its size. As a result, it does not recover any spectral information lost during the convolution process. Additionally,  $\sigma(\cdot)$  operates element-wise and introduces non-linear distortions or stretching to the input data. Consequently, it not only fails to recover the spectral information lost by the convolutional layer, but it may also exacerbate spectral distortions. Thus, neither the normalization layer nor the non-linear activation function can recover the spectral information lost by the convolution operation, and therefore, these components do not alter the conclusion drawn in Remark 1.

## II. SUPPLEMENTARY EXPLANATIONS FOR REMARK 2

Here, we provide detailed explanations of Remark 2 under two additional conditions: (i) when the input data and target

output have sizes  $H \times W \times C$  and  $H \times W \times C'$ , respectively, where  $C'$  denotes the number of output channels; and (ii) when the input data and target output have dimensions of  $H \times W \times L \times C$  and  $H \times W \times L \times C'$ , respectively.

In the first condition, we denote the input data and target output as  $\mathcal{A} \in \mathbb{R}^{H \times W \times C}$  and  $\mathcal{B} \in \mathbb{R}^{H \times W \times C'}$ , respectively. Similar to the supplementary explanation for Remark 1, we first generate a third-order tensor, denoted as  $\mathcal{A}' \in \mathbb{R}^{H \times W \times (Ck^2)}$ , which encompasses all convolution windows in  $\mathcal{A}$ . Next, we perform mode-3 unfolding and transposition on  $\mathcal{A}'$  and  $\mathcal{B}$ , resulting in matrices  $\mathbf{A} \in \mathbb{R}^{(HW) \times (Ck^2)}$  and  $\mathbf{B} \in \mathbb{R}^{(HW) \times C'}$ , respectively. With the kernel weights directly represented as  $\mathbf{X} \in \mathbb{R}^{(Ck^2) \times C'}$ , the standard 2D convolution can be mathematically expressed as follows:

$$\mathbf{A}\mathbf{X}(\cdot, i) = \mathbf{B}(\cdot, i), \quad i = 1, 2, \dots, C'. \quad (18)$$

According to Definition 4, Equation 18 can be interpreted as  $C'$  non-homogeneous systems. Let  $\overline{\mathbf{A}}_i = (\mathbf{A}|\mathbf{B}(\cdot, i)) \in \mathbb{R}^{(HW) \times (Ck^2+1)}$ , where  $i = 1, 2, \dots, C'$ , be a sequence of augmented matrices. Since  $HW$  significantly exceeds  $Ck^2+1$ , the standard convolution fails to yield the desired output whenever there exists an  $i$  such that  $\text{rank}(\mathbf{A}) \neq \text{rank}(\overline{\mathbf{A}}_i)$ . With the kernel weights directly represented as  $\mathcal{X} \in \mathbb{R}^{(Ck^2) \times (HW) \times C'}$ , the adaptive 2D convolution can be expressed as follows:

$$\mathbf{A}(j, \cdot)\mathcal{X}(\cdot, j, i) = \mathbf{B}(\cdot, i), \quad i = 1, 2, \dots, C', \quad j = 1, 2, \dots, HW. \quad (19)$$

If we consider  $\mathbf{A}(j, \cdot)$  and  $\overline{\mathbf{A}}_i(j, \cdot)$  as row matrices, then  $\text{rank}(\mathbf{A}(j, \cdot)) = \text{rank}(\overline{\mathbf{A}}_i(j, \cdot)) \leq Ck^2$ , which means the adaptive convolution can generate the intended output.

In the second condition, we denote the input data and target output as  $\mathcal{C} \in \mathbb{R}^{H \times W \times L \times C}$  and  $\mathcal{D} \in \mathbb{R}^{H \times W \times L \times C'}$ . We begin by creating a fourth-order tensor, represented as  $\mathcal{C}' \in \mathbb{R}^{H \times W \times L \times (Ck^3)}$ , that contains all  $k \times k \times k$  windows in  $\mathcal{C}$ . Next, we perform mode-4 unfolding and transposition on  $\mathcal{C}'$  and  $\mathcal{D}$ , generating  $\mathbf{C} \in \mathbb{R}^{(HWL) \times (Ck^3)}$  and  $\mathbf{D} \in \mathbb{R}^{(HWL) \times C'}$ . With the kernel weights defined as  $\mathbf{W} \in \mathbb{R}^{(Ck^3) \times C'}$ , the standard 3D convolution can be expressed as follows:

$$\mathbf{C}\mathbf{W}(\cdot, i) = \mathbf{D}(\cdot, i), \quad i = 1, 2, \dots, C'. \quad (20)$$

According to Definition 4, Equation 20 can be also seen as  $C'$  non-homogeneous systems. Let  $\overline{\mathbf{C}}_i = (\mathbf{C}|\mathbf{D}(\cdot, i)) \in \mathbb{R}^{(HWL) \times (Ck^3+1)}$ , where  $i = 1, 2, \dots, C'$ , be a sequence of augmented matrices. Since  $HWL$  is much larger than  $Ck^3+1$ , the standard convolution cannot produce the desired output whenever there exists an  $i$  such that  $\text{rank}(\mathbf{C}) \neq \text{rank}(\overline{\mathbf{C}}_i)$ . With the kernel weights denoted as  $\mathcal{W} \in \mathbb{R}^{(Ck^3) \times (HWL) \times C'}$ , the adaptive 3D convolution can be expressed as follows:

$$\mathbf{C}(j, \cdot)\mathcal{W}(\cdot, j, i) = \mathbf{D}(\cdot, i), \quad i = 1, 2, \dots, C', \quad j = 1, 2, \dots, HWL. \quad (21)$$

If  $\mathbf{C}(j, \cdot)$  and  $\overline{\mathbf{C}}_i(j, \cdot)$  are considered row matrices, then  $\text{rank}(\mathbf{C}(j, \cdot)) = \text{rank}(\overline{\mathbf{C}}_i(j, \cdot)) \leq Ck^3$ , which suggests that the adaptive convolution can produce the desired output.

### III. PRACTICAL EFFICIENCY ANALYSIS

To measure the practical efficiency of our method, we first compare Ada3D with other 3D convolutional paradigms in terms of inference time and energy consumption. As shown in

TABLE X  
PRACTICAL EFFICIENCY EVALUATION OF VARIOUS 3D CONVOLUTIONAL PARADIGMS. OUR METHOD YIELDS THE BEST PERFORMANCE.

Methods	Time (ms)	Energy (J)
Conv3D	36.3	8.97
DW3D	85.6	21.0
GPAC3D	71.2	17.6
D2Conv3D	289	31.9
ST-A3DNet	43.1	10.4
<b>The Proposed</b>	<b>17.8</b>	<b>4.42</b>

TABLE XI  
INFERENCE TIME FOR DIFFERENT COMPONENTS IN THE ADA3D BLOCK.

Modules	Time (ms)
Spatial Kernel Generator	0.34
Spectral Kernel Generator	0.37
Combination Section	0.03
Bias Generator	0.54
Ada3D Operation	0.12
<b>Ada3D Block</b>	<b>1.40</b>

Table X, the proposed method attains the lowest inference time and energy consumption, demonstrating the superior efficiency of Ada3D. Next, we analyze computational bottlenecks within the Ada3D block by measuring per-component inference time. As summarized in Table XI, the spatial kernel generator, spectral kernel generator, and bias generator dominate the runtime. All tests are run on an Nvidia A100 GPU.

### IV. NETWORK & TRAINING SETTINGS FOR HISR

For the HISR task, we configure the spatial and spectral feature maps with 64 and 32 channels. The values of  $k$ ,  $\alpha$ , and  $\beta$  are set to 3, 0.375, and 0.5, respectively. Additionally, we apply Bicubic for up-sampling. Furthermore, we utilize four ResBlocks to capture spatial details and leverage eight Ada3D blocks to integrate different types of information. All learnable weights are initialized using the Kaiming uniform distribution. During training, we set the number of epochs, batch size, initial learning rate, and  $\lambda_{ergas}$  to 500, 4,  $1 \times 10^{-4}$ , and  $1 \times 10^{-4}$ . In addition, we use the Adam optimizer and reduce the learning rate by half at the 300-th epoch.