LKVHAN: multi-scale large kernel vertical-horizontal attention network for hyperspectral image classification.

LIU, X., NG, A.H.-M., LIAO, X., LEI, F., REN, J. and GE, L.

2025

LKVHAN: Multiscale Large Kernel Vertical-Horizontal Attention Network for Hyperspectral Image Classification

Xun Liu, Alex Hay-Man Ng, *Senior Member, IEEE*, Xuejiao Liao, Fangyuan Lei, *Member, IEEE*, Jinchang Ren, *Senior Member, IEEE*, and Linlin Ge, *Member, IEEE*

***Abstract*—Among deep learning-based hyperspectral image (HSI) classification models, convolutional neural networks (CNNs), transformers, Mamba, and large kernel CNNs (LKCNNs) models have been widely explored for HSI classification (HSIC). Nonetheless, these models suffer from several challenges: for example, 1) CNNs have a weak learning ability in capturing global information between land covers, due to their limited receptive field derived from small kernel convolutions; 2) transformers face quadratic computational complexity introduced by their self-attention mechanisms; and 3) LKCNNs require further enhancement in extracting global features, owing to the insufficient size of their receptive fields. To tackle these limitations, we propose a novel multiscale large kernel vertical–horizontal attention network (LKVHAN) for HSIC. The proposed LKVHAN consists of a 1×1 convolution module and a multiscale large kernel vertical–horizontal attention-based convolution (MSLKVHAC). The 1×1 convolution module is designed to facilitate band reduction, noise suppression, and spectral feature learning. Furthermore, the MSLKVHAC, leveraging a large vertical kernel size of 17×1 and a large horizontal kernel size of 1×17, extracts both local and global spatial features by incorporating a vertical attention-based convolution module and a horizontal attention-based convolution module. Extensive experimental results demonstrate that the proposed LKVHAN significantly outperforms ten state-of-the-art approaches across four widely used HSI datasets.**

***Index Terms*—Hyperspectral image (HSI) classification, multiscale large kernel convolution, vertical–horizontal attention.**

Received 2 March 2025; revised 24 April 2025; accepted 1 May 2025. Date of publication 7 May 2025; date of current version 22 May 2025. This work was supported in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2019ZT08L213, in part by the National Natural Science Foundation of China under Grant 42274016, and in part by the Key Discipline Research Capacity Improvement Project of Guangdong Province under Grant 2024ZDJS022. *(Corresponding author: Alex Hay-Man Ng.)*

Xun Liu is with the School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: liuxun.stf@gmail.com).

Alex Hay-Man Ng is with the Department of Surveying Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: hayman.ng@gmail.com).

Xuejiao Liao and Fangyuan Lei are with the Guangdong Provincial Key Laboratory of Intellectual Property Big Data, Guangdong Polytechnic Normal University, Guangzhou 510665, China (e-mail: liaoxuej@163.com; leify@gpnu.edu.cn).

Jinchang Ren is with National Subsea Centre, Robert Gordon University, AB21 0BH Aberdeen, U.K. (e-mail: jinchang.ren@ieee.org).

Linlin Ge is with the Geoscience and Earth Observing System Group (GEOS), School of Civil and Environmental Engineering, University of New South Wales (UNSW), Sydney, NSW 2052, Australia (e-mail: l.ge@unsw.edu.au).

Digital Object Identifier 10.1109/JSTARS.2025.3567742

I. Introduction

With the development of remote sensing techniques, hyperspectral image (HSI) can be extensively acquired [1], [2]. Unlike RGB images that contain only three channels, HSI typically comprises hundreds or even thousands of contiguous spectral bands [3], carrying substantially richer spectral–spatial information. This abundant information enables HSI to accurately identify land cover types. Benefiting from these advantages, HSI has been widely applied to various applications, such as mineral exploration [4], environmental monitoring [5], [6], and atmospheric sciences [7]. HSI classification (HSIC) provides the fundamental technical support for these applications.

Early HSIC methods can be categorized into two types: feature extraction methods and classifier methods. Feature extraction approaches, such as principal component analysis [8] and manifold learning [9], are used to capture spectral information. To extract spectral–spatial features, researchers propose spectral–spatial feature extraction models, including sparse representation [10], morphological profiles [11], and Gabor filters [12]. In addition, classifier models, such as linear regression [13] and support vector machines [14], have been utilized to classify HSI. Nonetheless, these models belong to the domain of traditional machine learning (ML) techniques and rely on manual and careful design, thereby limiting their learning capability in the extraction of high-level features [15].

Compared with traditional ML approaches, deep learning (DL) approaches are widely applied in HSIC tasks due to their ability to automatically extract more discriminative features. Typical DL-based approaches can be roughly divided into two types: spectral feature-based models and spatial–spectral feature-based models. Spectral feature-based models [16] learn vectorized spectral features along the 1-D spectral bands. However, these models overlook the significance of spatial information. To address the limitation, researchers focus on spatial–spectral feature-based models [17], [18], [19]. Among these spatial–spectral models, convolutional neural networks (CNNs) have gained more attention for HSIC. Makantasis et al. [20] and Ma et al. [21] introduced a 2D-CNN to capture local spatial–spectral features. Yang et al. [22] presented a two-branch CNN architecture to extract joint spatial–spectral information. Subsequently, Chen et al. [23] proposed a 3D-CNN framework to learn deep high-level features, overcoming the imbalance problem

between high dimensionality and limited training samples. To construct deeper model, Zhong et al. [24] incorporated residual connections [25] into their CNN to enhance the capability of learning discriminative features. Although these models excel at capturing local information, they face a challenge in modeling long-range and global information due to their inherent local connections and the restricted receptive field derived from small-kernel convolutions.

Compared to CNNs that tend to extract local features, transformers have demonstrated proficiency in learning long-range dependencies and capturing global spatial information. Owing to these advantages, transformer-based models have been used in HSIC. Hong et al. [26] introduced a transformer-based network termed SpectralFormer, which extracts sequence information from neighboring bands. To capture spatial and spectral features, Sun et al. [27] designed a spectral–spatial feature tokenization transformer (SSFTT). Subsequently, Peng et al. [28] proposed a spatial–spectral transformer with cross-attention, enhancing classification performance with efficient spatial–spectral feature utilization. Mei et al. [29] developed a group-aware hierarchical transformer, which introduces a grouped pixel embedding block to improve the learning ability of local relationships along spectral bands. HybridFormer [30], DBCTNet [31], and GSC-ViT [32] combine convolution with transformer to capture local and global information. Furthermore, Feng et al. [33] utilized superpixel sampling instead of square patch sampling to prevent the mix of heterogeneous pixels at category boundaries. Jiang et al. [34] introduced absolute positional encoding into a transformer model for obtaining the absolute positional sequences of pixels. Nevertheless, these transformer architectures introduce a quadratic computational complexity [35], thereby leading to a high computational overhead, especially on large HSI datasets.

Recently, state–space models [36] and structured state–space sequence models (S4) [37] have been proposed, achieving superior performance in long-sequence data analysis [37], [38] compared to CNNs and transformers. Mamba [39] develops the S4 via a selective mechanism, addressing the quadratic computational complexity of transformer frameworks. In addition, Mamba shows strong long-range modeling capabilities, and have been successfully applied in these fields, such as time series [40], [41], speech [42], [43], medical image segmentation [44], [45], and point clouds [46], [47]. Due to these advantages, researchers have explored the potential of Mamba for HSIC tasks [35], [48], [49], [50]. For instance, Yao et al. [48] first applied Mamba to classify land cover types, addressing the computational inefficiency of transformer-based models. Huang et al. [49] and He et al. [51] leveraged stacked spectral–spatial Mamba blocks to establish long-range dependencies with linear computational complexity. Li et al. [35] developed a novel pure-Mamba-based model termed MambaHSI, which extracts discriminative spatial and spectral features via a spatial and spectral Mamba block and residual learning. However, the aforementioned architectures primarily focus on global feature extraction, which may result in the neglect of important local feature learning.

In addition to transformers and Mamba, which are well-known for their capability to model long-range dependencies, large kernel CNNs (LKCNNs) [52], [53], [54], [55], [56] have also proven effective in capturing long-range information by enlarging the size of convolution kernel. The convolution in LKCNNs is characterized by large kernel convolution, which offers a larger receptive field compared to the small kernel convolution used in traditional CNNs, thereby demonstrating significant advantages in natural visual applications. These advantages have motivated a limited number of studies [57], [58], [59], [60], [61], [62] to explore the potential of LKCNNs for HSIC. Lu et al. [59] designed a new large kernel attention (LKA) combined with an enhanced transformer to extract global features. Wu et al. [60] presented spectral–spatial large kernel attention network (SSLKA), which builds upon the classical LKA [63]. The LKA decomposes a $k \times k$ large kernel convolution into three convolutions: a $(2d-1) \times (2d-1)$ depthwise convolution (DC) [64], a $\frac{k}{d} \times \frac{k}{d}$ depthwise dilation convolution (DDC) with a dilation factor of d, and a 1×1 convolution. Furthermore, Sun et al. [61] proposed a multiscale efficient attention with enhanced feature transformer to enhance spectral–spatial feature extraction. Liu et al. [62] presented a multiscale large kernel asymmetric CNN for capturing both local and global features. Although several studies such as SSLKA have demonstrated success in learning global features through the LKA, they suffer from two challenges: 1) the LKA struggles to capture sufficient long-range dependencies when the value of k is relatively small; 2) Conversely, when k is large, the LKA incurs a substantial increase in both the number of parameters and computational complexity. Therefore, their receptive fields are not sufficiently expansive, resulting in a need for further enhancement in global feature extraction capabilities.

To address these issues of CNN-, transformer-, Mamba-, LKCNN-based approaches, we introduce a multiscale large kernel vertical–horizontal attention network (LKVHAN) for HSIC. In LKVHAN, we scale up the large vertical kernel size to 17×1 and the large horizontal kernel size to 1×17, respectively, as depicted in Fig. 1. Specifically, we first construct a 1×1 convolution module for band reduction, noise suppression, and spectral feature learning. Subsequently, we propose a novel multiscale large kernel vertical–horizontal attention-based convolution (MSLKVHAC), which utilizes a vertical attention-based convolution module (VACM) and a horizontal attention-based convolution module (HACM) to capture local-to-global spatial features. The main contributions of this article are summarized as follows:

1) To capture both local and global spatial information along the vertical axis, we propose the VACM based on a multiscale vertical large kernel 1-D depthwise convolution (MSVLK1DDC) module. To learn both local and global spatial features along the horizontal axis, we present the HACM via a multiscale horizontal large kernel 1-D depthwise convolution (MSHLK1DDC) module.
2) By utilizing the proposed VACM and HACM, we introduce a novel MSLKVHAC, which enhances the both local and global feature extraction capabilities of the classical LKA, frequently applied in HSIC models.
3) Based on our MSLKVHAC, we propose the LKVHAN to extract both spectral and spatial local-to-global features,

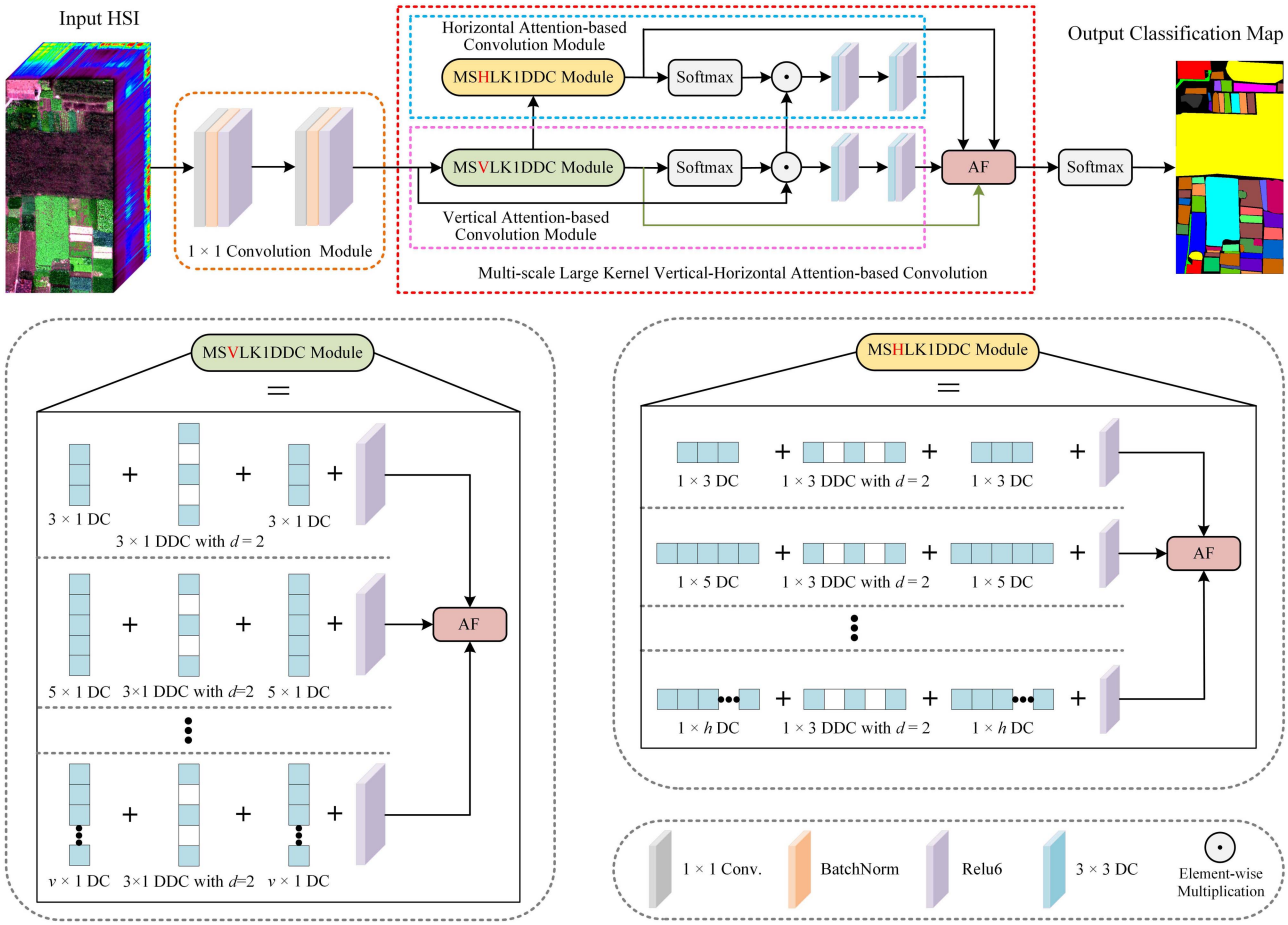

Fig. 1. Overview illustration of the proposed LKVHAN, which consists of two primary modules: the 1 × 1 convolution module and the MSLKVHAC.

tackling the challenges of traditional CNNs in modeling long-range dependencies and compensating for the shortcomings of transformer and Mamba models in capturing local features.

The rest of this article is organized as follows. In Section II, we introduce the proposed LKVHAN. The classification results of LKVHAN are evaluated and analyzed in Section III. Finally, Section IV concludes this article.

II. Proposed Method

The flowchart of the proposed LKVHAN framework is presented in Fig. 1. LKVHAN primarily comprises a 1 × 1 convolution module and the MSLKVHAC. The 1 × 1 convolution module is first used to suppress noise, reduce the number of bands, and capture spectral information. Subsequently, MSLKVHAC employs the VACM, followed by the HACM, to extract both local and global features along the vertical and horizontal axes of the HSI.

A. 1 × 1 Convolution Module

The raw HSI contains redundant spectral information and noise. To address the issues, we introduce the 1 × 1 convolution module using two consecutive 1 × 1 convolution blocks, offering three advantages: band reduction, noise suppression, and the ability to learn spectral features. Each block consists of a 1 × 1 convolution with a limited number of filters, followed by batch normalization (BN) and a ReLU6 activation function.

Let $X_i^l(\cdot)$ denote the input feature map of the lth 1 × 1 convolutional layer in the ith spectral channel. The output feature map of this convolutional layer, denoted as $X_i^{l+1}(p_1)$, can be formulated as follows:

$$X_i^{l+1}(p_1) = \text{ReLU6}(\text{BN}((W_i^l \cdot X_i^l(p_1) + b_i^l))) \quad (1)$$

where $p_1 = (x, y)$ represents the spatial location of the pixel in HSI, W_i^l is the trainable weight of the ith kernel in the lth convolutional layer, and b_i^l denotes the bias of the ith kernel in the lth convolutional layer.

B. MSLKVHAC

Compared to standard convolutions, DCs achieve a substantial reduction in the number of parameters and computational cost. Leveraging these advantages, many studies such as MSRN [65] explore DCs for HSIC. To capture long-range dependencies, Sun et al. [58] and Wu et al. [60] applied the LKA [63], which is commonly used in HSIC models based on large kernel convolution. As shown in Fig. 2(a), the LKA decomposes a $k \times k$ large kernel convolution into three convolutions: a $(2d-1) \times (2d-1)$ DC,

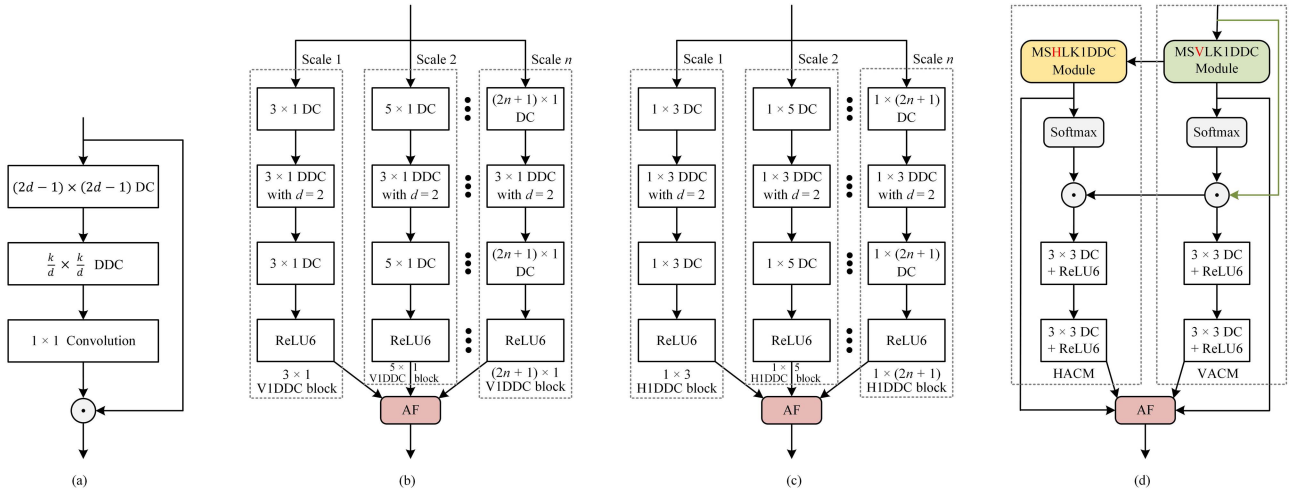

Fig. 2. Various large kernel convolutions. (a) LKA. (b) Proposed MSVLK1DDC module. (c) Proposed MSHLK1DDC. (d) Proposed MSLKVHAC.

a $\frac{k}{d} \times \frac{k}{d}$ DDC with dilation d, and a 1×1 convolution. This decomposition enables LKA to effectively capture long-range relationships while mitigating the quadratic increase in parameters and computational complexity that would otherwise result from applying a large kernel DC alone. However, LKA suffers from a significant number of parameters and computations when the kernel size k becomes large. To tackle the limitations, we replace the standard square kernels in LKA with vertical and horizontal kernels, and design the MSVLK1DDC module [see Fig. 2(b)] and the MSHLK1DDC module [see Fig. 2(c)] based on these respective kernels. By leveraging the proposed MSVLK1DDC and MSHLK1DDC modules, we present the novel MSLKVHAC, as illustrated in Fig. 2(d). The MSLKVHAC consists of two attention-based convolution modules and a fusion operation as follows:

1) The VACM, which learns local-to-global spatial features across pixels along the vertical axis.
2) The HACM, which extracts local-to-global spatial information along the horizontal axis.
3) An average fusion (AF) that integrates the features learned by the two attention-based convolution modules.

1) VACM: To extract local–global spatial features along the horizontal axis, we introduce the VACM by developing the MSVLK1DDC module. As illustrated in Fig. 2(b), the MSVLK1DDC module consists of n parallel V1DDC blocks, ranging from a vertical 1-D DC (V1DDC) block equipped with a 3×1 kernel to a V1DDC block with a $(2n+1) \times 1$ kernel. Each V1DDC block first extracts spatial features along the vertical axis through a sequence of three convolutions: a DC with a vertical kernel, a 3×1 DDC with dilation 2, and another DC with the same kernel size as the first DC. Then, a ReLU6 activation function is applied to model nonlinear features. In these V1DDC blocks, we maintain a consistent topology and follow two rules: 1) the hyperparameters, such as filter numbers and strides, remain uniform across all V1DDC blocks, except for their kernel sizes and 2) the kernel sizes follow an arithmetic progression with a common difference of 2. Based on these two rules, we only need to design the first V1DDC block and set the scale numbers, and hence the design complexity is significantly reduced. This streamlined approach allows us to focus on fine-tuning a small number of hyperparameters, simplifying the design process while maintaining performance efficiency.

For the proposed VACM, we take the feature map X_c captured by the 1×1 convolution module as input, then the output feature map H_{VACM} of the VACM can be expressed as follows:

$$H_{\text{MSVLK1DDC}} = \text{MSVLK1DDC}(X_c) = \text{AF}(V_3, V_5, \ldots, V_{2n+1}) = \frac{1}{n}(V_3 + V_5 + \cdots + V_{2n+1}) \quad (2)$$

$$H_{\text{va}} = \text{Softmax}(H_{\text{MSVLK1DDC}}) \quad (3)$$

$$\overline{H}_{\text{va}} = H_{\text{va}} \odot X_c \quad (4)$$

$$H_{\text{VACM}} = \text{ReLU6}(\text{DC}_{3\times3}(\text{ReLU6}(\text{DC}_{3\times3}(\overline{H}_{\text{va}})))) \quad (5)$$

where V_{2n+1} denotes the extracted features from X_c through a $(2n+1) \times 1$ V1DDC block. AF and $\odot$ denote the operators of AF and elementwise multiplication, respectively. n is the number of scales in the proposed MSVLK1DDC module. $H_{\text{MSVLK1DDC}}$ in (2) represents the output feature map of the MSVLK1DDC module which learns the local-to-global spatial information along the vertical axis. H_{va} in (3) is the vertical attention map obtained by applying the Softmax function. The output features $\overline{H}_{\text{va}}$ in (4) denotes elementwise multiplication of the H_{va} and X_c. $\text{DC}_{3\times3}$ represents a DC with a 3×3 kernel. As shown in (5), we utilize two identical $\text{DC}_{3\times3}$ blocks (including $\text{DC}_{3\times3}$ and ReLU6) to perform feature transformation, enhancing the model's expressive capability.

2) HACM: To capture local–global spatial features along the horizontal axis, we propose the HACM through designing the MSHLK1DDC module. As shown in Fig. 2(c), the MSHLK1DDC module is equipped with n parallel H1DDC blocks, ranging from a horizontal 1-D DC (H1DDC) block with a 1×3 kernel to an H1DDC block with a $1 \times (2n+1)$ kernel. Each H1DDC block first learns spatial information along the

horizontal axis using a sequence of three convolutions: a DC with a vertical kernel, a 1×3 DDC with dilation factor 2, and another DC with the same kernel size as the first DC. Subsequently, it applies a ReLU6 activation function to establish the nonlinear information. For these H1DDC blocks, we preserve a uniform structure and adhere to two rules: 1) apart from varying kernel sizes, the other hyperparameters are set to be the same and 2) the kernel sizes are arranged in an arithmetic progression with a common difference of 2. Analogous to the MSVLK1DDC module, following these two rules, the MSHLK1DDC module can avoid complicated design.

In the proposed HACM, the output feature map H_{HACM} can be defined as follows:

$$\begin{aligned} H_{\text{MSHLK1DDC}} &= \text{MSHLK1DDC}(H_{\text{MSVLK1DDC}}) \\ &= \text{AF}(H_3, H_5, \ldots, H_{2n+1}) \\ &= \frac{1}{n}(H_3 + H_5 + \cdots + H_{2n+1}) \end{aligned} \tag{6}$$

$$H_{\text{ha}} = \text{Softmax}(H_{\text{MSHLK1DDC}}) \tag{7}$$

$$\overline{H}_{\text{ha}} = H_{\text{ha}} \odot \overline{H}_{\text{va}} \tag{8}$$

$$H_{\text{HACM}} = \text{ReLU6}(\text{DC}_{3\times3}(\text{ReLU6}(\text{DC}_{3\times3}(\overline{H}_{\text{ha}})))) \tag{9}$$

where H_{2n+1} represents the learned features from $H_{\text{MSVLK1DDC}}$ using a $1 \times (2n+1)$ H1DDC block that consists of a sequence of a DC with a $1 \times (2n+1)$ kernel, a 1×3 DDC with dilation factor 2, another DC with a $1 \times (2n+1)$ kernel, and a ReLU6 activation function. $H_{\text{MSHLK1DDC}}$ in (6) is the output feature map of the MSHLK1DDC module, extracting the local-to-global spatial features along the horizontal axis. H_{ha} in (7) denotes the horizontal attention map. The output features $\overline{H}_{\text{ha}}$ in (8) is the elementwise multiplication of H_{ha} and $\overline{H}_{\text{va}}$. Similar to (5), two identical $\text{DC}_{3\times3}$ blocks are used to improve the model's learning ability.

3) AF: In the proposed MSLKVHAC, we employ the VACM and HACM to extract local-to-global spatial features along the vertical and horizontal axes, respectively. We introduce the AF to combine these features extracted by the VACM, HACM, MSVLK1DDC module, and MSHLK1DDC module. The process can be defined as follows:

$$\begin{aligned} H_{\text{MSLKVHAC}} &= \text{AF}(H_{\text{VACM}}; H_{\text{HACM}}; H_{\text{MSVLK1DDC}}; H_{\text{MSHLK1DDC}}) \\ &= \frac{1}{4}(H_{\text{VACM}} + H_{\text{HACM}} \\ &\quad + H_{\text{MSVLK1DDC}} + H_{\text{MSHLK1DDC}} \end{aligned} \tag{10}$$

where H_{MSLKVHAC} is the output features of the MSLKVHAC.

Analysis of parameters and complexity: In this section, we compute the number of parameters and floating point operations per second (FLOPS) of the LKA, the proposed MSVLK1DDC module, MSHLK1DDC module, and MSLKVHAC, as illustrated in Fig. 2. We assume that the number of input channels and output channels of the four modules are C. Then, the parameters of the four modules can be calculated as

$$\text{Param}_{\text{LKA}} = \left((2d-1)^2 \times C + \left(\frac{k}{d}\right)^2\right) \times C + C \times C \tag{11}$$

$$\text{Param}_{\text{MSVLK1DDC}} = C \times \sum_{i=1}^{n}(4i+5) = C \times (2n^2 + 7n) \tag{12}$$

$$\text{Param}_{\text{MSHLK1DDC}} = C \times \sum_{i=1}^{n}(4i+5) = C \times (2n^2 + 7n) \tag{13}$$

$$\begin{aligned} \text{Param}_{\text{MSLKVHAC}} &= 2C \times \sum_{i=1}^{n}(4i+5) + 3 \times 3 \times 4 \times C \\ &= C \times (4n^2 + 14n + 36) \end{aligned} \tag{14}$$

where $\text{Param}_{\text{LKA}}$, $\text{Param}_{\text{MSVLK1DDC}}$, $\text{Param}_{\text{MSHLK1DDC}}$, and $\text{Param}_{\text{MSLKVHAC}}$ denote the number of parameters of the LKA, the proposed MSVLK1DDC module, MSHLK1DDC module, and MSLKVHAC, respectively. The FLOPS are directly proportional to the number of parameters in their respective modules. In our experiments, we set n to 8 and C to 64. With these settings, the proposed MSLKVHAC has 25 858 parameters and proportional FLOPS. For (11), we set k to 41 and d to 2 to match the receptive field of the proposed MSLKVHAC. Based on these settings, the LKA has 32 896 parameters and proportional FLOPS. These results demonstrate the advantages of the proposed MSLKVHAC in terms of parameters and computational complexity.

C. Softmax Classification

In this section, we apply a fully connected layer followed by a Softmax function to classify the fused feature map H_{MSLKVHAC}, which can be expressed as

$$Y = \frac{e^{(W_i H_{\text{MSLKVHAC}} + b_i)}}{\sum_{i=1}^{c} e^{(W_i H_{\text{MSLKVHAC}} + b_i)}} \tag{15}$$

where c denotes the number of land cover categories, and W_i and b_i are the trainable parameter and bias, respectively.

The cross-entropy loss function is adopted to train the proposed model, namely

$$\zeta = -\sum_{z \in O_{\text{label}}} \sum_{j=1}^{c} O_{zj} \ln Y_{zj} \tag{16}$$

where O represents the label matrix, and Y_{zj} denotes the probability of the zth pixel belonging to the jth category.

III. Experiment

In this study, considering the high acquisition costs of labeled HSI samples, we focus on classification under small training sample conditions.

TABLE I
SUMMARY OF INDIAN PINES AND LONGKOU DATASETSSUMMARY OF INDIAN PINES AND LONGKOU DATASETS

Dataset	Indian Pines				LongKou			
Data size	145 × 145 × 200				550 × 400 × 270			
Wavelength	0.4 – 2.5 um				0.4 – 1.0 um			
Time	1992				2018			
Class no.	Class name	Train.	Val.	Test.	Class Name	Train.	Val.	Test.
C1	Alfalfa	4	5	37	Corn	4	5	34502
C2	Corn–notill	4	5	1419	Cotton	4	5	8365
C3	Corn–mintill	4	5	821	Sesame	4	5	3022
C4	Corn	4	5	228	Broad-leaf soybean	4	5	63203
C5	Grass–pasture	4	5	474	Narrow-leaf soybean	4	5	4142
C6	Grass–trees	4	5	721	Rice	4	5	11845
C7	Grass–pasture–mowed	4	5	19	Water	4	5	67047
C8	Hay–windrowed	4	5	469	Roads and houses	4	5	7115
C9	Oats	4	5	11	Mixed weed	4	5	5220
C10	Soybean–notill	4	5	963	–	-	–	–
C11	Soybean–mintill	4	5	2446	–	-	–	–
C12	Soybean–clean	4	5	587	–	-	–	–
C13	Wheat	4	5	196	–	-	–	–
C14	Woods	4	5	1256	–	-	–	–
C15	Buildings–grass–trees–drives	4	5	377	–	-	–	–
C16	Stone–steel–towers	4	5	84	–	-	–	–
Total	–	64	80	10105	–	36	45	204461

TABLE II
SUMMARY OF HONGHU AND HANCHUAN DATASETS

Dataset	HongHu				HanChuan			
Data size	940 × 475 × 270				1217 × 303 × 274			
Wavelength	0.4 – 1.0 um				0.4 – 1.0 um			
Time	2017				2016			
Class No.	Class name	Train.	Val.	Test.	Class name	Train.	Val.	Test.
C1	Red roof	4	5	14032	Strawberry	4	5	44726
C2	Road	4	5	3503	Cowpea	4	5	22744
C3	Bare soil	4	5	21812	Soybean	4	5	10278
C4	Cotton	4	5	163276	Sorghum	4	5	5344
C5	Cotton firewood	4	5	6209	Water spinach	4	5	1191
C6	Rape	4	5	44548	Watermelon	4	5	4524
C7	Chinese cabbage	4	5	24094	Greens	4	5	5894
C8	Pakchoi	4	5	4045	Trees	4	5	17969
C9	Cabbage	4	5	10810	Grass	4	5	9460
C10	Tuber mustard	4	5	12385	Red roof	4	5	10507
C11	Brassica parachinensis	4	5	11006	Gray roof	4	5	16902
C12	Brassica chinensis	4	5	8945	Plastic	4	5	3670
C13	Small brassica chinensis	4	5	22498	Bare soil	4	5	9107
C14	Lactuca sativa	4	5	7347	Road	4	5	18551
C15	Celtuce	4	5	993	Bright object	4	5	1127
C16	Film covered lettuce	4	5	7253	Water	4	5	75392
C17	Romaine lettuce	4	5	3001	–	–	–	–
C18	Carrot	4	5	3208	–	–	–	–
C19	White radish	4	5	8703	–	–	–	–
C20	Garlic sprout	4	5	3477	–	–	–	–
C21	Broad bean	4	5	1319	–	–	–	–
C22	Tree	4	5	4031	–	–	–	–
Total	–	88	110	386495	–	64	80	257386

A. Dataset Description

To evaluate the proposed LKVHAN, we adopted four benchmark HSI datasets, including Indian Pines, WHU-Hi-LongKou (LongKou), WHU-Hi-HongHu (HongHu), and WHU-Hi-HanChuan (HanChuan). Tables I and II summarize the details of these four datasets.

1) Indian Pines: The Indian Pines dataset was captured by using the airborne visible infrared imaging spectrometer (AVIRIS) sensor. The whole image comprises 145 × 145 pixels in the wavelength range from 0.4 to 2.5 μm, with 16 land cover categories and 10 249 labeled pixels. We retain 200 spectral bands by removing these noisy and water absorption bands of 104–108, 150–163, and 220.

TABLE III
QUANTITATIVE COMPARISON OF ALL METHODS ON THE INDIAN PINES DATASET USING FOUR LABELLED SAMPLES PER CLASS FOR TRAINING

Class	CNNs			Transformers				Mamba	LKCNNs		
	SSRN	DBDA	A^2S^2K-Res	SSFTT	morphFormer	GSC-ViT	DBCTNet	MambaHSI	ETLKA	SSLKA	LKVHAN
	TGRS 2018	RS 2020	TGRS 2021	TGRS 2022	TGRS 2023	TGRS 2024	TGRS 2024	TGRS 2024	RS 2024	TGRS 2024	Ours
1	95.68 ± 3.67	97.84 ± 2.02	98.38 ± 2.16	**100.0 ± 0.00**	92.16 ± 11.11	95.14 ± 10.24	98.11 ± 4.02	97.57 ± 2.25	98.92 ± 1.32	**100.0 ± 0.00**	99.73 ± 0.81
2	41.52 ± 17.88	53.64 ± 12.95	56.12 ± 16.28	47.17 ± 4.39	42.64 ± 10.75	45.53 ± 8.98	36.71 ± 12.98	48.68 ± 7.79	50.97 ± 5.70	47.74 ± 8.57	**69.89 ± 7.35**
3	51.53 ± 11.04	50.52 ± 13.62	46.18 ± 12.78	54.62 ± 5.44	44.54 ± 9.96	54.63 ± 8.41	45.48 ± 7.86	47.39 ± 10.54	57.25 ± 7.09	63.62 ± 8.16	**72.03 ± 11.31**
4	74.08 ± 14.98	96.89 ± 7.10	93.03 ± 6.80	**97.85 ± 1.99**	77.15 ± 16.51	74.91 ± 11.92	80.83 ± 15.99	87.68 ± 7.33	65.75 ± 7.41	89.30 ± 6.63	93.07 ± 8.20
5	78.14 ± 8.04	77.32 ± 19.64	82.34 ± 12.94	**92.74 ± 1.76**	73.73 ± 6.50	75.61 ± 8.57	69.51 ± 8.40	78.19 ± 6.06	75.40 ± 7.78	69.11 ± 6.79	79.77 ± 8.25
6	91.76 ± 9.14	95.80 ± 2.67	96.82 ± 2.52	95.38 ± 2.27	81.37 ± 13.16	93.18 ± 3.79	87.88 ± 4.29	88.25 ± 5.49	95.02 ± 1.79	97.98 ± 1.96	**99.25 ± 0.64**
7	99.47 ± 1.58	**100.0 ± 0.00**	**100.0 ± 0.00**	**100.0 ± 0.00**	98.95 ± 3.16	99.47 ± 1.58	99.47 ± 1.58	**100.0 ± 0.00**	98.42 ± 3.37	**100.0 ± 0.00**	**100.0 ± 0.00**
8	78.46 ± 14.45	92.75 ± 14.83	86.35 ± 7.55	97.91 ± 1.63	94.80 ± 5.09	93.48 ± 11.33	76.29 ± 14.53	92.69 ± 10.03	83.65 ± 8.62	99.25 ± 1.13	**99.77 ± 0.53**
9	**100.0 ± 0.00**	**100.0 ± 0.00**	99.09 ± 2.73	**100.0 ± 0.00**	95.45 ± 9.32	99.09 ± 2.73	**100.0 ± 0.00**	**100.0 ± 0.00**	85.45 ± 24.80	**100.0 ± 0.00**	**100.0 ± 0.00**
10	56.16 ± 7.24	68.85 ± 9.07	74.41 ± 10.15	67.03 ± 5.02	57.63 ± 10.98	66.63 ± 14.27	59.40 ± 11.34	59.66 ± 13.10	69.62 ± 5.84	71.41 ± 2.63	**80.34 ± 9.59**
11	50.99 ± 18.43	**61.05 ± 11.89**	43.71 ± 16.20	55.22 ± 6.21	51.91 ± 10.07	50.81 ± 13.91	48.21 ± 16.48	50.98 ± 8.43	59.57 ± 9.95	46.40 ± 8.50	60.94 ± 9.13
12	52.79 ± 12.42	60.99 ± 14.40	68.99 ± 12.59	39.01 ± 6.11	41.85 ± 6.94	46.11 ± 11.23	49.67 ± 14.86	49.85 ± 9.40	60.03 ± 4.63	70.55 ± 9.29	**72.38 ± 11.46**
13	98.88 ± 1.04	**100.0 ± 0.00**	99.39 ± 1.25	97.24 ± 2.02	99.29 ± 1.12	99.34 ± 0.89	94.59 ± 7.73	99.64 ± 0.61	**100.0 ± 0.00**	99.95 ± 0.15	99.69 ± 0.41
14	81.74 ± 10.84	91.47 ± 5.01	82.92 ± 6.11	80.94 ± 4.40	77.11 ± 8.71	82.24 ± 6.96	85.98 ± 8.92	87.14 ± 7.75	**95.09 ± 2.05**	91.54 ± 9.56	90.80 ± 6.87
15	76.79 ± 10.84	92.33 ± 6.69	84.72 ± 7.10	84.32 ± 5.02	76.92 ± 13.79	79.18 ± 12.06	73.21 ± 11.71	87.53 ± 13.27	55.81 ± 5.42	65.25 ± 6.32	**94.35 ± 4.80**
16	**100.0 ± 0.00**	99.88 ± 0.36	99.40 ± 1.79	**100.0 ± 0.00**	98.33 ± 3.54	**100.0 ± 0.00**	98.21 ± 2.57	98.93 ± 1.24	89.76 ± 5.30	**100.0 ± 0.00**	99.76 ± 0.71
OA	62.71 ± 6.27	71.69 ± 5.27	67.05 ± 5.41	67.57 ± 2.05	61.31 ± 3.35	64.95 ± 3.64	60.67 ± 4.92	65.35 ± 3.99	69.24 ± 3.21	68.09 ± 1.94	**78.04 ± 3.23**
AA	76.75 ± 2.82	83.71 ± 3.12	81.99 ± 2.28	81.84 ± 1.08	75.24 ± 2.49	78.46 ± 2.52	75.22 ± 2.78	79.64 ± 2.27	77.54 ± 2.99	82.01 ± 0.79	**88.24 ± 1.84**
KAPPA	58.18 ± 6.40	68.15 ± 5.82	63.37 ± 5.72	63.65 ± 2.15	56.74 ± 3.57	60.71 ± 3.94	56.09 ± 5.12	**76.99 ± 2.71**	65.17 ± 3.50	64.52 ± 1.99	75.33 ± 3.56

The bold values indicate the best performance.

2) WHU-Hi-LongKou (LongKou): The LongKou dataset was provided by using an 8 mm focal length Headwall Nano-Hyperspec imaging sensor over the town of LongKou, Hubei Province, China, in 2018 [66]. It contains 550 × 400 pixels with nine ground-truth classes, ranging from 0.4 to 1.0 μm.

3) WHU-Hi-HongHu (HongHu): The HongHu dataset was acquired by the 17 mm focal length Headwall Nano-Hyperspec imaging sensor over the town of Honghu, Hubei Province, China in 2017 [66]. The dataset contains 22 land cover categories, comprising 940 × 475 pixels with 270 spectral bands ranging from 0.4 to 1.0 μm.

4) WHU-Hi-HanChuan (HanChuan): The HanChuan dataset was captured by using the same imaging sensor as that used for the LongKou dataset. The dataset was collected over the town of HanChuan, Hubei Province, China in 2016 [66]. It comprises 1217 × 303 pixels with 16 land cover classes and 274 spectral bands in the wavelength range from 0.4 to 1.0 μm.

B. Experimental Setup

1) Comparison Methods: To demonstrate the effectiveness of the proposed LKVHAN, we compared LKVHAN with ten baseline methods as follows:

1) Three CNN-based methods: the spectral–spatial residual network (SSRN) [24], the double-branch dual-attention (DBDA) network [67], and the attention-based adaptive spectral–spatial kernel ResNet (A^2S^2 K-Res) [68].
2) Four transformer-based methods: the SSFTT [27], the morphological transformer (morphFormer) [69], the groupwise separable convolutional vision transformer (GSC-ViT) [32], and the double branch convolution-transformer network (DBCTNet) [31].
3) Mamba-based method: the spatial–spectral Mamba (MambaHSI) [35].
4) Two LKCNN-based methods: the enhanced transformer with large kernel attention (ETLKA) [59] and the SSLKA [60].

2) Evaluation Metrics: To evaluate the classification performance of various methods, we introduced four evaluation metrics: per-class accuracy, overall accuracy (OA), average accuracy (AA), and kappa coefficient (KAPPA).

3) Implementation Details: All experiments were implemented on a Silver 4210 CPU, Python 3.10, and a GTX-3090 GPU. We trained our model using Adam optimizer with a learning rate of 0.001 on Pytorch platform. In the proposed LKVHAN, we set the number of filters for each convolution layer to 64, the scale number n–8, and the number of training epochs to 250, for all datasets. For different methods on each dataset, we reported the average performance across these four evaluation metrics by running the experiments twenty times with diverse random initializations.

C. Comparison With State-of-The-Art Methods

Tables III–VI summarize the evaluation metrics for different methods on the Indian Pines, LongKou, HongHu, and HanChuan datasets. The corresponding visualization maps are presented in Figs. 3–6.

1) Results on Indian Pines: Table III reports the quantitative results of the proposed LKVHAN and ten comparison methods on the Indian Pines dataset. From these results, we can observe that DBDA outperforms other baselines in terms of OA and AA, demonstrating the superiority of DBDA in enhancing local feature extraction through its double-branch and dual-attention mechanisms. It is worth noting that all methods achieve low and unstable accuracy for several challenging classes, such as classes 2 and 3. This may be attributed to overfitting caused by the limited number of training samples, as well as the significant impact of randomly selected training samples on these classes. In addition, our LKVHAN utilizes the VACM and HACM to extract both local and global features, which surpasses almost all comparison methods in terms of OA, AA, and KAPPA, as well as achieving higher accuracy in nine out of sixteen classes, with the exception of MambaHSI in KAPPA. These results demonstrate the effectiveness of LKVHAN.

Fig. 3 presents a qualitative evaluation by visualizing classification maps generated by diverse methods on the Indian Pines dataset. As evidenced by the figure, the proposed LKVHAN achieves the clearest and smoothest classifications across most classes, such as "Corn-notill," "Corn-mintill," and "Soybean-notill."

2) Results on LongKou: The comparative results of different methods on the LongKou dataset are showed in Table IV. As

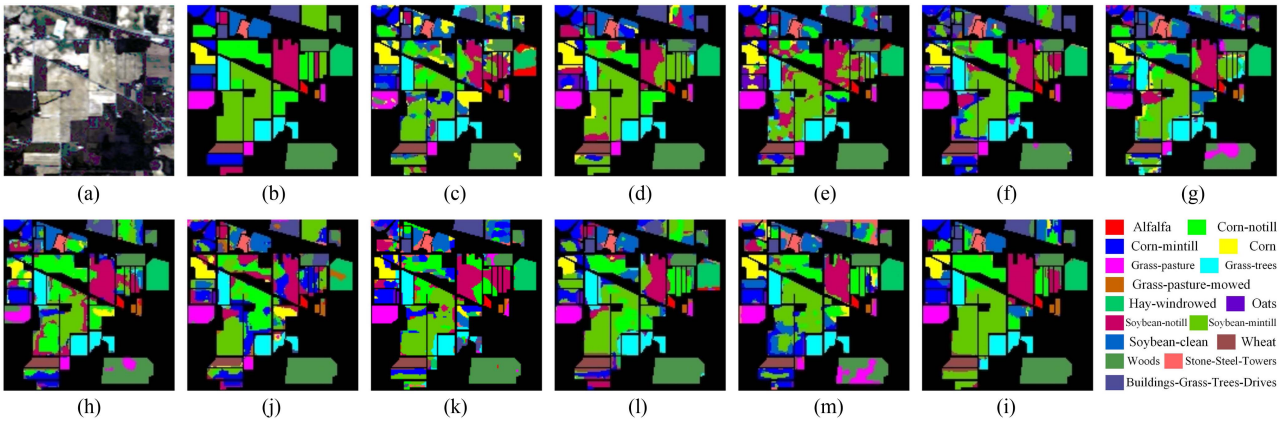

Fig. 3. False-color image, ground truth, and classification maps on the Indian Pines dataset. (a) False-color image. (b) Ground truth. (c) SSRN (OA = 62.71%). (d) DBDA (OA = 71.69%). (e) A^2S^2K-Res (OA = 67.05%). (f) SSFTT (OA = 67.57%). (g) morphFormer (OA = 61.31%). (h) GSC-ViT (OA = 64.95%). (j) DBCTNet (OA = 60.67%). (k) MambaHSI (OA = 65.35%). (l) ETLKA (OA = 69.24%). (m) SSLKA (OA = 68.09%). (i) LKVHAN (OA = 78.04%).

TABLE IV
QUANTITATIVE COMPARISON OF ALL METHODS ON THE LONGKOU DATASET USING FOUR LABELLED SAMPLES PER CLASS FOR TRAINING

Class	CNNs			Transformers				Mamba	LKCNNs		
	SSRN TGRS 2018	DBDA RS 2020	A^2S^2K-Res TGRS 2021	SSFTT TGRS 2022	morphFormer TGRS 2023	GSC-ViT TGRS 2024	DBCTNet TGRS 2024	MambaHSI TGRS 2024	ETLKA RS 2024	SSLKA TGRS 2024	LKVHAN Ours
1	93.76 ± 3.95	**98.96 ± 1.19**	93.32 ± 6.11	97.94 ± 1.49	93.30 ± 4.30	97.43 ± 1.94	95.35 ± 4.13	98.79 ± 1.01	90.76 ± 4.81	96.32 ± 1.78	97.67 ± 2.32
2	72.91 ± 17.51	87.95 ± 14.31	95.56 ± 5.29	**96.02 ± 1.44**	82.27 ± 13.08	84.22 ± 9.52	78.04 ± 13.10	85.98 ± 9.94	92.91 ± 2.77	86.13 ± 5.71	92.90 ± 5.62
3	81.40 ± 12.78	94.37 ± 3.76	95.84 ± 5.39	**99.86 ± 0.21**	86.30 ± 10.82	90.32 ± 6.18	92.91 ± 6.29	92.39 ± 5.27	99.70 ± 0.39	95.11 ± 2.00	98.10 ± 3.17
4	67.28 ± 28.46	81.98 ± 7.71	85.59 ± 6.28	74.87 ± 6.90	70.81 ± 12.70	78.77 ± 10.28	74.25 ± 18.12	64.41 ± 16.81	73.71 ± 5.39	81.28 ± 2.58	**88.12 ± 7.60**
5	73.06 ± 29.40	78.30 ± 16.24	89.24 ± 14.78	77.11 ± 4.00	86.39 ± 11.76	90.61 ± 9.67	91.26 ± 6.94	75.00 ± 19.43	**99.57 ± 0.37**	95.88 ± 2.28	90.62 ± 12.78
6	93.95 ± 2.75	91.38 ± 7.42	74.17 ± 10.10	94.69 ± 3.85	90.32 ± 6.35	93.40 ± 4.36	96.08 ± 3.09	93.82 ± 7.08	89.31 ± 0.97	**97.84 ± 1.23**	94.02 ± 7.14
7	**99.91 ± 0.12**	99.80 ± 0.34	98.40 ± 2.35	96.59 ± 0.78	99.16 ± 0.48	99.77 ± 0.17	99.38 ± 0.44	99.72 ± 0.39	96.65 ± 0.64	99.00 ± 0.45	99.20 ± 0.90
8	60.75 ± 20.82	**88.27 ± 13.44**	76.79 ± 19.98	50.82 ± 3.95	73.49 ± 12.89	82.06 ± 10.93	83.37 ± 14.26	73.60 ± 19.56	65.58 ± 7.39	82.03 ± 5.96	84.91 ± 15.88
9	61.92 ± 17.55	54.22 ± 18.48	76.29 ± 13.39	**93.32 ± 1.49**	68.70 ± 12.04	62.69 ± 13.70	70.36 ± 13.47	87.57 ± 8.15	84.35 ± 4.18	88.00 ± 2.94	85.69 ± 10.42
OA	84.19 ± 8.92	91.10 ± 3.22	90.52 ± 1.98	87.95 ± 2.28	86.09 ± 3.63	89.99 ± 3.22	88.31 ± 5.80	85.92 ± 5.06	86.70 ± 1.87	91.48 ± 0.83	**93.93 ± 2.62**
AA	78.33 ± 6.22	86.14 ± 4.74	87.25 ± 3.41	86.80 ± 1.31	83.42 ± 1.77	86.59 ± 2.05	86.78 ± 3.27	85.70 ± 2.97	88.06 ± 1.49	91.29 ± 1.00	**92.36 ± 2.47**
KAPPA	80.25 ± 10.22	88.53 ± 4.04	87.75 ± 2.50	84.54 ± 2.83	82.37 ± 4.37	87.19 ± 3.97	85.18 ± 7.10	82.25 ± 7.02	83.05 ± 2.29	89.06 ± 1.04	**92.15 ± 3.33**

The bold values indicate the best performance.

can be observed, the proposed LKVHAN achieves the first-best performance in terms of OA, AA, and KAPPA across all methods. Specifically, LKVHAN improves over CNN-based approaches by at least 3.11%, 5.86%, and 4.09%, improves over Transformer-based approaches by at least 4.38%, 6.41%, and 5.69%, improves over MambaHSI by 9.32%, 7.77%, and 12.04%, and improves over LKCNN-based approaches by 2.68%, 1.17%, and 3.47% in terms of OA, AA, and KAPPA, respectively. We observe that most methods exhibit instability under small training sample sizes for several challenging classes, such as class 8, indicating that more samples are needed for these classes to achieve stable results. Furthermore, the corresponding classification maps of various approaches are visualized in Fig. 4. Along these all approaches, our LKVHAN exhibits a superior classification map across most regions. These results demonstrate the advantages of LKVHAN.

3) Results on HongHu: Table V summarizes the classification accuracies achieved by diverse methods on the HongHu dataset. From the table, we can observe that the proposed LKVHAN consistently outperforms all CNN-based, transformer-based, Mamba-based, and LKCNN-based methods. Specifically, LKVHAN gains 6.46%, 6.73%, and 7.64% improvements over the best CNN-based method (DBDA), gains 8.89%, 8.96%, and 10.25% improvements over the best transformer-based method (DBCTNet), gains 12.45%, 10.32%, and 12.11% improvements over MambaHSI, and gains 10.62%, 13.50%, and 12.44% improvements over the best LKCNN-based method (SSLKA) in terms of OA, AA, and KAPPA, respectively. For certain challenging classes such as classes 7 and 8, all methods achieve limited performance in terms of accuracy and stability with small training data sizes, suggesting that these methods are prone to overfitting and instability when dealing with small training samples. In addition, the visualization maps for all methods on the HongHu dataset are shown in Fig. 5. Based on the figure, we can observe that the proposed LKVHAN has less misclassification across all methods. These significant improvements highlight the effectiveness of LKVHAN in capturing both local and global information through the proposed large kernel convolution.

4) Results on HanChuan: Table VI reports the classification results of various approaches on the HanChuan dataset. From these results, the proposed LKVHAN still achieves promising performance, exceeding other approaches by a substantial margin. Among the sixteen categories, LKVHAN obtains the state-of-the-art performance in seven of them. Similar to the Indian Pines and HongHu datasets, the limited number of training samples and their random selection lead to unstable and suboptimal classification results for all methods in certain classes, such as classes 13 and 14. In addition, the corresponding classification maps generated by different approaches are illustrated in Fig. 6. According to the classification maps, we can observe that in most classes, our LKVHAN achieves the most precise prediction

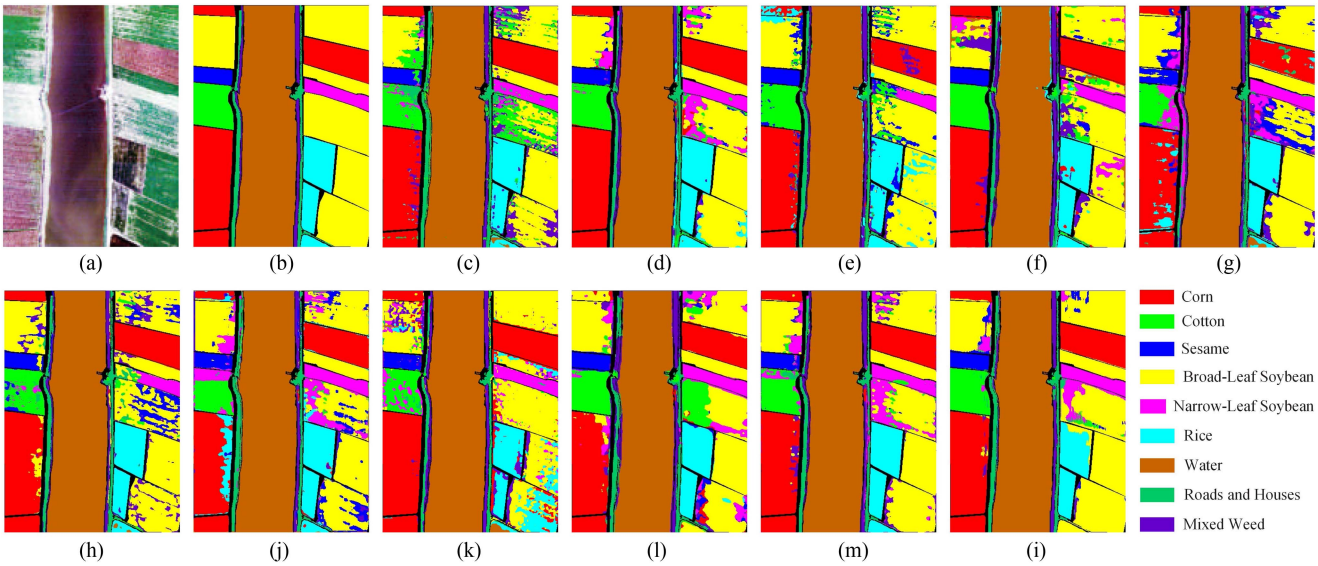

Fig. 4. False-color image, ground truth, and classification maps on the LongKou dataset. (a) False-color image. (b) Ground truth. (c) SSRN (OA = 84.19%). (d) DBDA (OA = 91.10%). (e) A^2S^2K-Res (OA = 90.52%). (f) SSFTT (OA = 87.95%). (g) morphFormer (OA = 86.09%). (h) GSC-ViT (OA = 89.99%). (j) DBCTNet (OA = 88.31%). (k) MambaHSI (OA = 85.92%). (l) ETLKA (OA = 86.70%). (m) SSLKA (OA = 91.48%). (i) LKVHAN (OA = 93.93%).

TABLE V
QUANTITATIVE COMPARISON OF ALL METHODS ON THE HONGHU DATASET USING FOUR LABELED SAMPLES PER CLASS FOR TRAINING

Class	CNNs			Transformers				Mamba	LKCNNs		
	SSRN	DBDA	A^2S^2K-Res	SSFTT	morphFormer	GSC-ViT	DBCTNet	MambaHSI	ETLKA	SSLKA	LKVHAN
	TGRS 2018	RS 2020	TGRS 2021	TGRS 2022	TGRS 2023	TGRS 2024	TGRS 2024	TGRS 2024	RS 2024	TGRS 2024	Ours
1	72.77 ± 7.98	75.92 ± 8.32	73.12 ± 10.67	**87.70 ± 4.08**	70.98 ± 11.33	80.29 ± 12.10	75.81 ± 9.88	83.16 ± 8.81	68.76 ± 4.44	83.14 ± 4.68	83.00 ± 8.47
2	76.36 ± 11.33	**87.27 ± 6.59**	61.12 ± 10.91	61.49 ± 6.37	61.33 ± 11.73	70.52 ± 10.41	73.12 ± 6.73	74.22 ± 9.36	61.22 ± 11.87	81.36 ± 7.22	76.08 ± 17.42
3	61.31 ± 25.04	81.16 ± 6.77	79.46 ± 8.06	**85.30 ± 2.36**	70.19 ± 7.07	76.83 ± 2.27	76.58 ± 8.66	76.63 ± 12.34	84.10 ± 3.57	83.11 ± 3.24	81.79 ± 4.77
4	59.49 ± 16.65	84.95 ± 6.96	68.85 ± 15.58	80.44 ± 8.05	80.90 ± 7.36	81.73 ± 9.02	81.63 ± 11.74	77.55 ± 10.03	83.92 ± 9.95	81.29 ± 3.80	**91.72 ± 2.84**
5	51.52 ± 20.82	74.47 ± 11.43	57.48 ± 6.97	84.47 ± 11.70	62.44 ± 8.71	68.70 ± 12.04	78.45 ± 7.69	75.50 ± 14.43	74.60 ± 12.76	72.74 ± 8.56	**90.71 ± 5.42**
6	83.72 ± 7.10	91.45 ± 3.73	86.69 ± 7.26	85.39 ± 2.82	86.14 ± 4.75	83.36 ± 5.33	87.52 ± 4.23	79.56 ± 11.01	81.28 ± 3.55	85.40 ± 9.18	**91.74 ± 4.12**
7	35.12 ± 15.72	52.09 ± 14.21	45.54 ± 9.11	30.17 ± 4.49	48.88 ± 8.33	58.95 ± 7.24	54.24 ± 10.27	49.32 ± 11.08	45.88 ± 22.57	**60.16 ± 8.55**	59.81 ± 14.64
8	21.33 ± 11.65	34.86 ± 14.61	30.87 ± 14.69	52.89 ± 8.15	29.06 ± 7.66	32.27 ± 10.65	45.33 ± 9.57	**62.23 ± 10.08**	56.93 ± 7.50	28.36 ± 7.31	54.87 ± 11.81
9	90.81 ± 5.92	92.92 ± 3.23	90.40 ± 4.30	82.19 ± 3.98	87.65 ± 6.43	90.68 ± 6.89	86.89 ± 5.87	82.49 ± 6.74	90.17 ± 3.87	**94.38 ± 1.05**	93.37 ± 2.03
10	37.45 ± 15.89	59.68 ± 16.90	33.85 ± 13.43	49.73 ± 10.04	66.08 ± 7.11	59.43 ± 8.58	63.55 ± 13.17	55.01 ± 11.53	51.10 ± 16.42	51.24 ± 12.72	**82.89 ± 5.00**
11	25.57 ± 12.97	50.30 ± 8.19	24.72 ± 13.59	41.13 ± 5.85	42.76 ± 12.42	43.87 ± 9.24	59.15 ± 16.68	47.38 ± 11.19	49.28 ± 13.76	41.29 ± 11.68	**63.01 ± 12.78**
12	41.95 ± 19.98	58.13 ± 17.76	33.33 ± 17.73	44.35 ± 6.39	57.59 ± 13.46	57.92 ± 10.75	54.41 ± 13.38	55.62 ± 8.76	36.30 ± 21.84	51.46 ± 17.74	**63.66 ± 6.64**
13	52.71 ± 20.55	50.16 ± 12.00	32.07 ± 13.96	49.51 ± 7.58	47.79 ± 15.98	49.31 ± 11.89	49.21 ± 13.06	48.90 ± 13.86	26.85 ± 11.19	40.62 ± 10.80	**62.80 ± 6.46**
14	57.56 ± 12.39	72.54 ± 8.31	60.43 ± 8.60	76.92 ± 7.38	65.06 ± 15.27	62.95 ± 13.15	65.88 ± 6.91	67.70 ± 9.12	77.55 ± 13.52	43.83 ± 12.43	**81.72 ± 9.76**
15	78.42 ± 18.21	95.32 ± 3.40	93.20 ± 4.12	**98.80 ± 0.94**	88.24 ± 6.73	94.58 ± 2.01	92.40 ± 3.91	92.79 ± 4.13	86.98 ± 7.17	92.95 ± 3.81	98.14 ± 1.09
16	61.63 ± 23.89	86.02 ± 8.50	80.00 ± 8.56	80.02 ± 4.80	69.05 ± 10.57	80.23 ± 9.18	85.44 ± 15.00	79.42 ± 9.72	82.32 ± 7.26	**89.28 ± 6.04**	85.03 ± 6.40
17	69.83 ± 14.19	77.34 ± 9.19	73.28 ± 12.79	74.26 ± 1.73	78.47 ± 7.04	75.48 ± 10.35	82.23 ± 11.48	66.28 ± 11.69	**96.90 ± 2.28**	74.08 ± 21.59	87.99 ± 11.85
18	71.95 ± 14.97	88.63 ± 14.58	73.88 ± 18.88	73.83 ± 12.49	69.92 ± 12.04	77.88 ± 9.41	78.85 ± 8.67	83.99 ± 7.41	93.78 ± 6.84	66.43 ± 16.83	**95.34 ± 5.54**
19	59.00 ± 18.28	83.49 ± 6.08	73.59 ± 9.41	72.31 ± 4.42	53.70 ± 12.51	72.28 ± 9.78	77.53 ± 12.47	62.41 ± 15.38	80.41 ± 6.47	61.87 ± 12.09	**86.90 ± 4.98**
20	59.33 ± 18.05	**89.68 ± 7.45**	48.84 ± 18.33	45.09 ± 4.01	71.74 ± 17.98	82.95 ± 14.72	78.21 ± 17.37	88.23 ± 16.07	63.44 ± 15.87	71.89 ± 11.13	87.47 ± 18.78
21	64.61 ± 22.87	82.84 ± 20.06	50.15 ± 20.32	65.78 ± 13.76	86.53 ± 9.59	81.77 ± 9.95	88.22 ± 10.77	84.89 ± 12.81	75.66 ± 22.30	70.77 ± 19.99	**95.78 ± 4.10**
22	67.70 ± 20.75	93.28 ± 3.85	59.12 ± 10.68	74.12 ± 9.55	84.45 ± 7.59	76.77 ± 11.04	78.81 ± 12.92	90.29 ± 4.52	70.12 ± 13.47	87.92 ± 6.00	**96.89 ± 2.39**
OA	59.12 ± 6.16	78.09 ± 2.15	64.77 ± 6.99	72.58 ± 4.00	72.43 ± 3.94	74.67 ± 3.69	75.66 ± 5.25	72.10 ± 5.10	73.39 ± 4.84	73.93 ± 1.80	**84.55 ± 1.63**
AA	58.64 ± 5.03	75.57 ± 2.94	60.45 ± 1.39	68.00 ± 1.98	67.22 ± 3.18	70.85 ± 2.28	73.34 ± 1.98	71.98 ± 2.85	69.89 ± 2.27	68.80 ± 2.75	**82.30 ± 1.88**
KAPPA	52.69 ± 5.65	73.23 ± 2.30	58.44 ± 6.58	66.83 ± 4.27	66.71 ± 4.19	69.35 ± 3.96	70.62 ± 5.53	68.76 ± 3.76	67.66 ± 5.22	68.43 ± 2.06	**80.87 ± 1.92**

The bold values indicate the best performance.

TABLE VI
QUANTITATIVE COMPARISON OF ALL METHODS ON THE HANCHUAN DATASET USING FOUR LABELLED SAMPLES PER CLASS FOR TRAINING

Class	CNNs			Transformers				Mamba	LKCNNs		
	SSRN	DBDA	A^2S^2K-Res	SSFTT	morphFormer	GSC-ViT	DBCTNet	MambaHSI	ETLKA	SSLKA	LKVHAN
	TGRS 2018	RS 2020	TGRS 2021	TGRS 2022	TGRS 2023	TGRS 2024	TGRS 2024	TGRS 2024	RS 2024	TGRS 2024	Ours
1	62.48 ± 20.29	67.04 ± 17.56	52.56 ± 21.15	59.53 ± 8.12	59.65 ± 21.10	62.42 ± 14.97	68.27 ± 3.99	74.55 ± 14.04	72.74 ± 9.90	**88.88 ± 2.25**	82.21 ± 5.68
2	30.99 ± 19.27	54.95 ± 8.87	54.42 ± 14.66	**71.75 ± 4.89**	54.27 ± 9.74	42.00 ± 8.63	54.81 ± 12.71	39.22 ± 7.83	63.02 ± 7.17	45.32 ± 9.37	58.91 ± 9.57
3	67.97 ± 12.75	77.92 ± 12.63	61.92 ± 26.26	81.03 ± 6.24	49.59 ± 12.44	64.21 ± 18.03	63.91 ± 16.97	69.09 ± 15.71	70.76 ± 9.44	80.77 ± 4.35	**88.34 ± 5.20**
4	83.48 ± 16.91	98.12 ± 1.05	97.77 ± 1.48	95.12 ± 2.95	79.75 ± 10.48	90.86 ± 7.40	95.69 ± 3.55	88.69 ± 8.64	**98.50 ± 0.90**	91.49 ± 2.59	97.81 ± 1.06
5	73.64 ± 26.31	99.1 1± 1.90	91.97 ± 6.61	86.12 ± 7.66	88.93 ± 10.84	90.55 ± 7.69	90.76 ± 8.45	97.30 ± 2.62	96.64 ± 7.47	74.78 ± 8.46	**99.69 ± 0.48**
6	26.54 ± 18.69	37.49 ± 13.80	44.52 ± 13.77	41.83 ± 4.72	26.74 ± 4.52	36.90 ± 8.87	38.02 ± 11.91	33.16 ± 10.39	46.60 ± 5.38	45.84 ± 6.42	**64.39 ± 13.24**
7	68.57 ± 16.63	80.09 ± 13.08	79.46 ± 7.00	86.45 ± 4.42	78.15 ± 14.24	80.73 ± 9.07	87.39 ± 7.98	74.51 ± 19.82	83.21 ± 4.38	75.99 ± 5.07	**89.63 ± 7.97**
8	36.95 ± 12.00	39.92 ± 9.83	42.40 ± 12.35	33.70 ± 7.46	35.61 ± 12.98	44.65 ± 12.60	44.17 ± 8.31	33.27 ± 12.99	56.30 ± 14.34	36.61 ± 3.15	**60.79 ± 6.91**
9	28.44 ± 16.31	36.46 ± 13.14	30.77 ± 13.43	53.29 ± 3.67	25.07 ± 7.80	39.30 ± 11.14	42.93 ± 15.03	45.09 ± 9.07	49.57 ± 3.42	**62.26 ± 5.16**	54.09 ± 11.77
10	59.71 ± 27.26	75.22 ± 17.81	69.45 ± 22.84	76.00 ± 5.91	70.89 ± 9.44	86.15 ± 13.04	65.00 ± 18.75	72.12 ± 14.32	**94.75 ± 2.96**	79.73 ± 2.51	88.08 ± 10.77
11	30.06 ± 23.82	36.52 ± 30.31	63.71 ± 16.98	59.08 ± 3.84	70.38 ± 16.65	66.02 ± 15.86	72.38 ± 12.99	67.86 ± 16.94	45.69 ± 5.49	44.89 ± 4.68	**87.95 ± 11.88**
12	32.18 ± 16.45	57.66 ± 13.82	62.37 ± 13.49	60.85 ± 5.23	57.23 ± 14.08	56.38 ± 18.54	61.54 ± 18.63	67.68 ± 14.39	**88.07 ± 5.52**	67.12 ± 11.72	70.93 ± 11.19
13	46.63 ± 14.04	49.09 ± 21.92	43.06 ± 17.69	58.09 ± 6.27	40.68 ± 7.26	45.32 ± 7.35	38.69 ± 16.42	42.20 ± 15.28	37.49 ± 6.62	45.13 ± 9.61	**61.96 ± 10.27**
14	42.95 ± 13.90	41.42 ± 16.34	45.86 ± 14.16	39.94 ± 4.54	51.35 ± 10.63	53.67 ± 12.40	52.40 ± 17.63	**55.28 ± 8.03**	54.44 ± 3.73	44.67 ± 10.75	49.65 ± 11.74
15	69.36 ± 11.36	79.27 ± 7.50	68.50 ± 11.78	**91.33 ± 1.11**	71.68 ± 13.52	75.31 ± 6.49	71.97 ± 12.83	76.69 ± 9.47	86.19 ± 4.46	85.79 ± 4.85	78.28 ± 9.28
16	89.90 ± 7.86	89.50 ± 9.72	84.81 ± 10.86	97.20 ± 1.00	90.76 ± 9.37	91.05 ± 8.33	92.30 ± 7.98	87.01 ± 9.35	**99.15 ± 0.59**	96.37 ± 0.55	93.49 ± 9.19
OA	60.30 ± 5.66	66.31 ± 4.96	63.44 ± 6.47	70.97 ± 2.43	65.25 ± 3.83	67.62 ± 2.44	69.78 ± 4.14	67.40 ± 3.61	74.75 ± 2.09	72.89 ± 1.24	**78.81 ± 3.32**
AA	53.11 ± 5.28	63.74 ± 4.58	62.10 ± 4.41	68.21 ± 2.01	59.42 ± 2.03	64.10 ± 2.02	65.01 ± 2.52	63.98 ± 2.78	70.82 ± 1.75	66.60 ± 1.22	**76.64 ± 1.70**
KAPPA	54.31 ± 6.42	61.53 ± 5.32	58.37 ± 7.07	66.52 ± 2.74	60.15 ± 4.15	62.85 ± 2.67	65.18 ± 4.51	60.67 ± 4.47	70.61 ± 2.34	68.59 ± 1.40	**75.58 ± 3.59**

The bold values indicate the best performance.

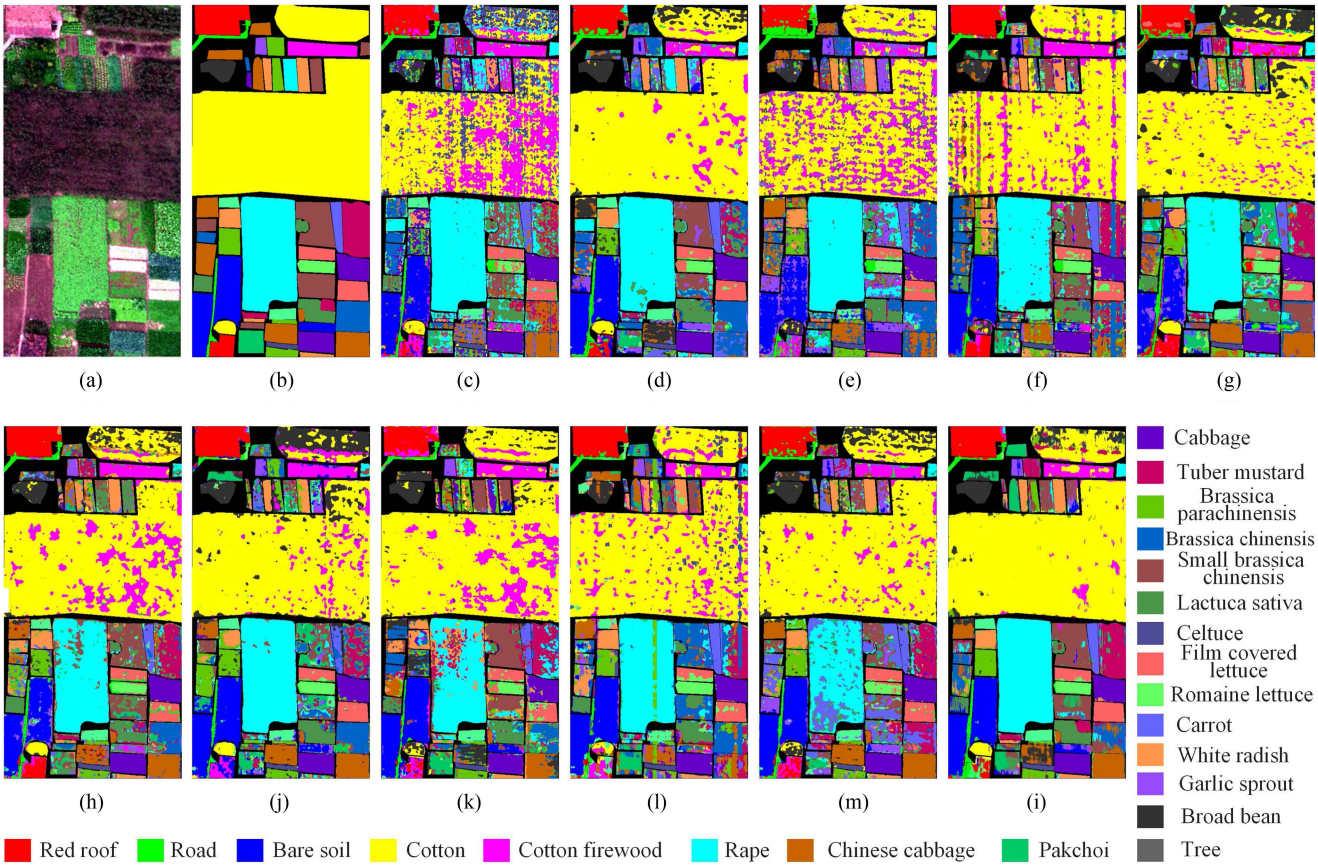

Fig. 5. False-color image, ground truth, and classification maps on the HongHu dataset. (a) False-color image. (b) Ground truth. (c) SSRN (OA = 59.12%). (d) DBDA (OA = 78.09%). (e) A^2S^2 K-Res (OA = 64.77%). (f) SSFTT (OA = 72.58%). (g) morphFormer (OA = 72.43%). (h) GSC-ViT (OA = 74.67%). (j) DBCTNet (OA = 75.66%). (k) MambaHSI (OA = 72.10%). (l) ETLKA (OA = 73.39%). (m) SSLKA (OA = 73.93%). (i) LKVHAN (OA = 84.55%).

details. These significant improvements further demonstrate the potential of LKVHAN for HSIC.

D. Analysis of Different Methods Under Various Training Sample Sizes

To evaluate the robustness of the proposed LKVHAN, we compared the OA results obtained by different methods using training sample sizes of 2, 4, 6, 8, and 10 for each category across the Indian Pines, LongKou, HongHu, and HanChuan datasets. The results are depicted in Fig. 7. From these results, we can observe that the OA results of most approaches show a steady enhancement trend as the number of training samples increases. However, for a few comparison approaches, such as GSC-ViT and SSLKA, there are instances where the OA results decrease unexpectedly with an increase in training sample size. These anomalous results may be attributed to the extra noise caused by the increased training sample size. Notably, our proposed LKVHAN employs its 1×1 convolution module for noise suppression, exhibiting a notable increase in OA as the number of training samples increases. In addition, the proposed LKVHAN significantly outperforms other competitive methods across all datasets, which further demonstrates its robustness and strength.

E. Analysis of LKA and Our MSLKVHAC

As illustrated in Fig. 2, we introduced two large kernel convolutions: the classical LKA and the proposed MSLKVHAC. For the proposed LKVHAN, we replaced the MSLKVHAC with the LKA, followed by the two identical $DC_{3\times3}$ blocks, as introduced in (5). The resulting model is termed $LKVHAN_{LKA}$. To evaluate our MSLKVHAC, we compared the classification results of the $LKVHAN_{LKA}$ and LKVHAN in terms of OA, AA, and KAPPA coefficient. As presented in Table VII, our LKVHAN significantly outperforms the $LKVHAN_{LKA}$ across the four datasets. These significant improvements validate the superiority of our MSLKVHAC in extracting both local and global features.

F. Analysis of Computational Complexity

Table VIII presents the performance metrics, including parameters, FLOPS, training time, and testing time, generated by various methods across all datasets. From the table, we made the following observations.

1) In terms of parameters, LKVHAN, along with DBCTNet and MambaHSI, achieves the best performance, significantly surpassing other methods.
2) LKVHAN and MambaHSI exhibit the highest FLOPS across all methods, which is attributed to the fact that

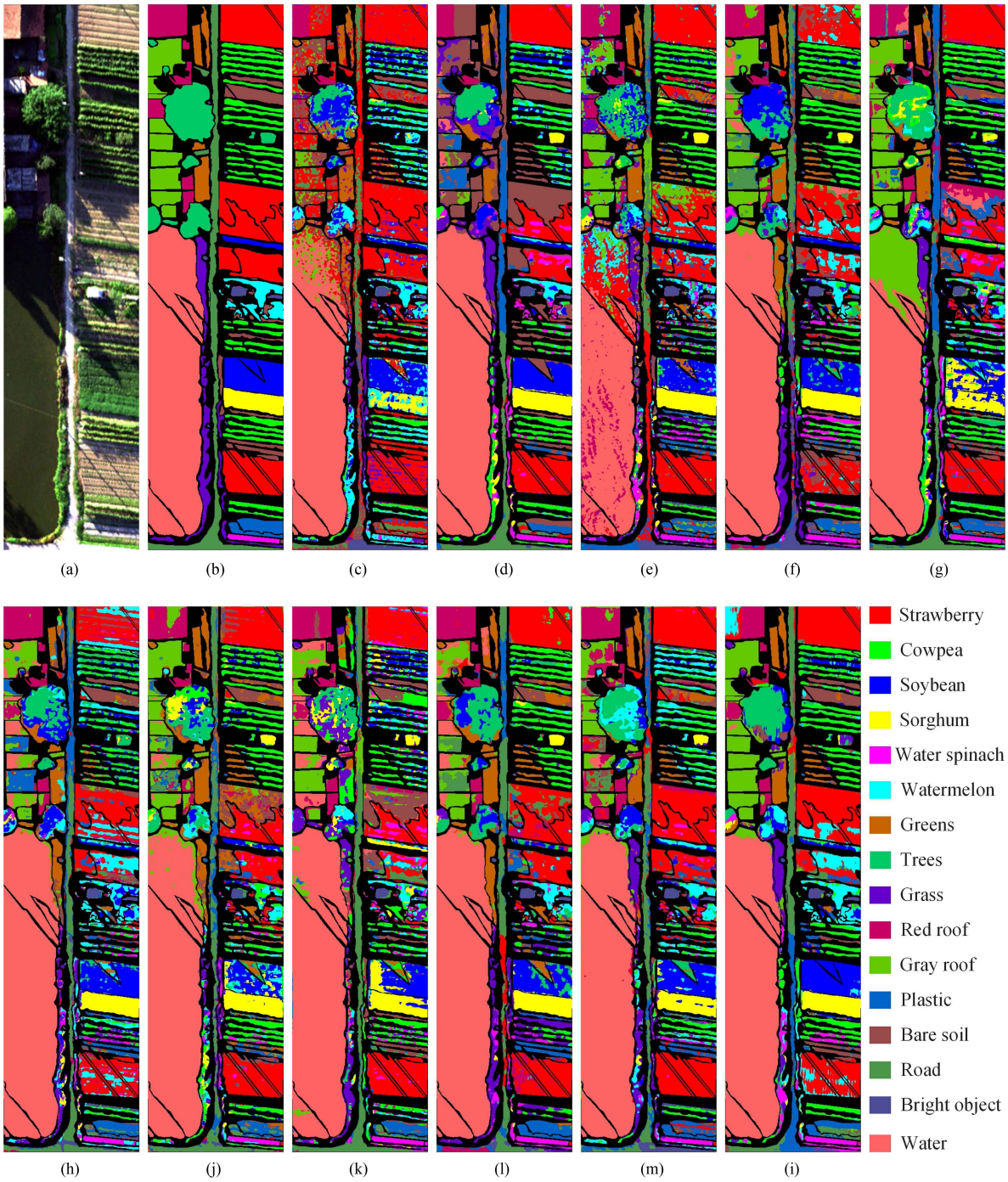

Fig. 6. False-color image, ground truth, and classification maps on the HanChuan dataset. (a) False-color image. (b) Ground truth. (c) SSRN (OA = 60.30%). (d) DBDA (OA = 66.31%). (e) A^2S^2 K-Res (OA = 63.44%). (f) SSFTT (OA = 70.97%). (g) morphFormer (OA = 65.25%). (h) GSC-ViT (OA = 67.62%). (j) DBCTNet (OA = 69.78%). (k) MambaHSI (OA = 67.40%). (l) ETLKA (OA = 74.75%). (m) SSLKA (OA = 72.89%). (i) LKVHAN (OA = 78.81%).

TABLE VII
CLASSIFICATION RESULTS OF LKVHAN$_{\text{LKA}}$ AND LKVHAN

Method	Indian Pines			LongKou			HongHu			HanChuan		
	OA	AA	KAPPA	OA	AA	KAPPA	OA	AA	KAPPA	OA	AA	KAPPA
LKVHAN$_{\text{LKA}}$	54.03	69.68	48.77	90.88	88.06	88.27	64.73	58.30	57.64	66.15	59.66	61.03
LKVHAN	**78.04**	**88.24**	**75.33**	**93.93**	**92.36**	**92.15**	**84.55**	**82.30**	**80.87**	**78.81**	**76.64**	**75.58**

The bold values indicate the best performance.

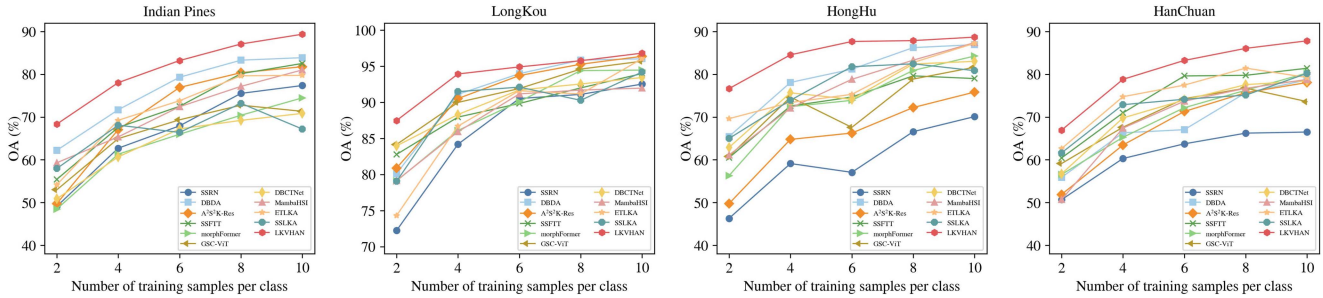

Fig. 7. OA results of different methods using various training sample sizes per class across the Indian Pines, LongKou, HongHu, and HanChuan datasets.

TABLE VIII
ANALYSIS OF DIFFERENT METHODS IN TERMS OF PARAMETERS, FLOPS, TRAIN TIME, AND TEST TIME ON THE INDIAN PINES, LONGKOU, HONGHU, AND HANCHUAN DATASETS

Dataset	Metrics	CNNs			Transformers				Mamba	LKCNNs		
		SSRN TGRS 2018	DBDA RS 2020	A^2S^2K-Res TGRS 2021	SSFTT TGRS 2022	morphFormer TGRS 2023	GSC-ViT TGRS 2024	DBCTNet TGRS 2024	MambaHSI TGRS 2024	ETLKA RS 2024	SSLKA TGRS 2024	LKVHAN Ours
Indian Pines	Parameters (K)	364.2	382.3	370.8	148.5	199.6	563.7	**30.3**	44.3	203.1	136.9	44.2
	FLOPS (G)	0.158	0.108	0.104	0.011	0.036	0.021	0.012	0.563	0.032	**0.007**	0.931
	Train time (s)	3.82	9.97	4.16	**1.83**	50.35	10.05	43.86	32.79	3.69	9.50	3.47
	Test time (s)	2.17	6.30	2.02	0.47	6.10	1.86	1.13	6.65 ms	0.51	1.04	**2.79 ms**
LongKou	Parameters (K)	471.5	509.2	74.0	148.0	255.1	173.1	**40.3**	52.4	202.7	140.0	48.2
	FLOPS (G)	0.215	0.146	**0.003**	0.011	0.048	0.010	0.017	7.852	0.032	0.007	10.630
	Train time (s)	2.62	7.03	3.03	**1.49**	40.45	8.88	42.26	272.33	2.57	6.82	33.51
	Test time (s)	57.37	167.15	35.93	9.13	69.14	20.52	32.08	22.47 ms	11.11	18.74	**5.60 ms**
HongHu	Parameters (K)	471.8	510.7	93.1	148.9	256.0	173.9	**40.5**	54.0	203.5	140.6	49.0
	FLOPS (G)	0.215	0.146	0.010	0.011	0.048	0.010	0.017	15.981	0.032	**0.007**	21.946
	Train time (s)	5.68	24.12	6.13	**2.62**	60.69	12.14	133.33	543.84	6.25	15.12	66.60
	Test time (s)	108.91	287.76	83.05	18.19	85.43	44.77	60.11	28.20 ms	20.26	38.61	**25.03 ms**
HanChuan	Parameters (K)	477.8	517.3	83.6	148.5	258.8	122.1	**41.0**	53.8	203.1	140.5	48.9
	FLOPS (G)	0.218	0.149	**0.007**	0.011	0.049	0.008	0.017	13.368	0.032	**0.007**	18.078
	Train time (s)	4.09	12.00	4.42	**1.83**	47.18	10.70	152.14	458.25	3.08	8.37	67.02
	Test time (s)	69.21	191.59	53.29	12.04	71.78	38.35	36.57	**22.69 ms**	12.78	25.07	26.25 ms

K, G, S, and Ms denote kilo, giga, second, and millisecond, respectively.
The bold values indicate the best performance.

TABLE IX
CLASSIFICATION RESULTS OF DIFFERENT MODULES IN LKVHAN

1×1 conv. module	MSLKVHAC	Indian Pines			LongKou			HongHu			HanChuan		
		OA	AA	KAPPA	OA	AA	KAPPA	OA	AA	KAPPA	OA	AA	KAPPA
×	✓	72.07	84.75	68.75	93.48	91.85	91.58	83.33	78.87	79.30	77.95	76.03	74.56
✓	×	40.21	54.14	34.06	79.25	74.89	73.93	32.68	41.52	27.93	47.63	42.05	41.07
✓	✓	**78.04**	**88.24**	**75.33**	**93.93**	**92.36**	**92.15**	**84.55**	**82.30**	**80.87**	**78.81**	**76.64**	**75.58**

The bold values indicate the best performance.

these two methods take the entire HSI as input, while other methods use small HSI cubes for input.

3) SSFTT exhibits the fastest training speeds across each method on all datasets, due to its limited number of convolutional layers.
4) MambaHSI and the proposed LKVHAN replace small HSI cubes with the whole HSI for input, thereby performing better on each dataset. Notably, LKVHAN outperforms baseline methods by substantial margins. These observations demonstrate the significant advantages of our LKVHAN in balancing computational efficiency and classification performance.

G. Ablation Study

1) Effects of Different Modules: Our LKVHAN consists of two core modules: the 1×1 convolution module and the MSLKVHAC. To assess the individual effects of the two modules, we compared the classification results generated by the full LKVHAN with those modified versions of LKVHAN that remove one of the two modules. The results are presented in Table IX. To maintain consistency between the number of bands in the original HSI and the number of filters in the convolutional layers, we retained a 1×1 convolution block after removing the 1×1 convolution module. As shown in Table IX, LKVHAN without the MSLKVHAC module exhibits a significant inferiority to other approaches on the four datasets, implying that the MSLKVHAC plays an extremely important role on enhancing performance. Furthermore, LKVHAN outperforms its modified versions across each dataset. These observations show the effectiveness of the two modules to performance improvement.

2) Effects of Various Scales: As shown in Fig. 2, we utilize the scale number n of the proposed MSLKVHAC to control its kernel size. Therefore, the optimal scale number directly determines the optimal kernel size. To evaluate the effects of the number of scales, we compared the OA results obtained by the proposed LKVHAN using various scales across the four

TABLE X
COMPARISON RESULTS OF NUMBER OF VARIOUS SCALES IN MSLKVHAC IN TERMS OF PARAMETERS, FLOPS, AND OA ON THE HONGHU DATASET

Scale (n)	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$
Parameters (K)	**26.6**	28.3	30.5	33.2	36.4	40.1	44.3	49.0	54.3
FLOPS (G)	**11.9**	12.7	13.7	14.9	16.3	17.9	19.8	21.9	24.3
OA	56.85	72.31	73.83	78.91	81.02	82.15	82.72	**84.55**	83.45

K and G denote kilo and giga, respectively.
The bold values indicate the best performance.

TABLE XI
OA COMPARISON OF LKVHAN AND THE REDUCED LKVHAN METHODS THAT EXCLUDE ONE OF THE TWO ATTENTION MODULES ACROSS THE INDIAN PINES, LONGKOU, HONGHU, AND HANCHUAN DATASETS

Method	Indian Pines			LongKou			HongHu			HanChuan		
	OA	AA	KAPPA	OA	AA	KAPPA	OA	AA	KAPPA	OA	AA	KAPPA
LKVHAN without VACM	70.28	81.41	66.64	86.73	86.77	83.23	75.02	75.43	69.87	71.25	67.87	67.01
LKVHAN without HACM	72.15	83.25	68.76	90.62	89.26	87.95	68.65	68.57	62.87	70.41	69.43	66.23
LKVHAN	**78.04**	**88.24**	**75.33**	**93.93**	**92.36**	**92.15**	**84.55**	**82.30**	**80.87**	**78.81**	**76.64**	**75.58**

The bold values indicate the best performance.

TABLE XII
COMPARISON RESULTS OF NUMBER OF DIFFERENT CHANNELS IN TERMS OF PARAMETERS, FLOPS, AND OA ACROSS THE INDIAN PINES, LONGKOU, HONGHU, AND HANCHUAN DATASETS

Metrics	Indian Pines			LongKou			HongHu			HanChuan		
	$C=32$	$C=64$	$C=128$	$C=32$	$C=64$	$C=128$	$C=32$	$C=64$	$C=128$	$C=32$	$C=64$	$C=128$
Parameters (K)	**21.1**	44.2	96.5	**23.1**	48.2	104.6	**23.5**	49.0	106.3	**23.4**	48.9	106.0
FLOPS (G)	**0.4**	0.9	2.0	**5.1**	10.6	23.1	**10.5**	21.9	47.6	**8.7**	18.1	39.2
OA	75.47	78.04	**78.22**	93.40	**93.93**	92.26	82.46	**84.55**	83.91	75.08	**78.81**	76.35

K and G denote kilo and giga, respectively.
The bold values indicate the best performance.

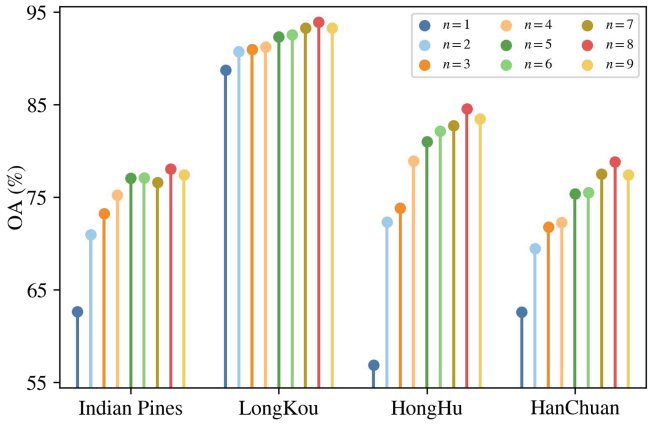

Fig. 8. OA Results of number of various scales in MSLKVHAC across the Indian Pines, LongKou, HongHu, and HanChuan datasets.

datasets. The results are depicted in Fig. 8. We can observe that the OA results exhibit an increasing trend with the increase of number of scales (n), reaching a peak at $n=8$. Nevertheless, further increasing n leads to a decrease in OA results. In addition, taking the HongHu dataset as an example, we compared the parameters, FLOPS, and OA of our proposed LKVHAN at different scales. The results are reported in Table X. From the table, it can be observed that when $n=8$, the OA reaches its highest value while maintaining acceptable parameters and computational complexity. Based on these comparisons, we determine the optimal scale number to be 8, corresponding to an optimal vertical kernel size of 17×1 and an optimal horizontal kernel size of 1×17.

3) Effects of LKA Module: For the proposed MSLKVHAC, we designed two LKA modules: the VACM and HACM. To verify the effects of the two LKA modules, we compared the OA results of LKVHAN with those of LKVHAN removed versions that exclude one of the two attention modules across each dataset. These results are reported in Table XI. From these results, we can observe that LKVHAN exhibits superior performance compared to its removed versions, which supports the design for the two LKA modules.

4) Effects of Different Channels: To evaluate the effects of the number of channels (C), we compared the parameters, FLOPS, and OA of the proposed LKVHAN across different channel configurations ($C=32$ to 128) on all four datasets. As shown in Table XII, both the parameters and FLOPS show a monotonic increase with increasing C values, while the OA in most datasets peaks at $C=64$ before declining. Considering the balance between computational complexity and classification performance, we determined $C=64$ to be the optimal hyperparameter configuration.

IV. CONCLUSION

In this article, we presented a novel attention architecture, termed LKVHAN, for HSIC. This architecture explores two new mechanisms: the large kernel vertical attention mechanism and the large kernel horizontal attention mechanism. The LKVHAN architecture comprises two modules: 1) the 1×1 convolution

module is responsible for band reduction, noise suppression, and spectral feature learning and 2) the MSLKVHAC module is designed to capture short-range, middle-range, and long-range spatial features along vertical and horizontal axes through the VACM and the HACM, respectively. The experimental results demonstrate that LKVHAN outperforms CNN-, transformer-, Mamba-, and LKCNN-based approaches by a significant margin, highlighting its effectiveness in HSIC. Future research will focus on integrating the LKVHAN with a Mamba model for further enhancing global feature extraction capabilities.

REFERENCES

[1] Q. Zhu et al., "A spectral-spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 11709–11723, Nov. 2022.

[2] P. Ma et al., "Multiscale superpixelwise prophet model for noise-robust feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, Mar. 2023.

[3] Y. Li et al., "CBANet: An end-to-end cross-band 2-D attention network for hyperspectral change detection in remote sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–11, May 2023.

[4] R. J. Murphy, S. Schneider, and S. T. Monteiro, "Consistency of measurements of wavelength position from hyperspectral imagery: Use of the ferric iron crystal field absorption at ∼900 nm as an indicator of mineralogy," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2843–2857, May 2014.

[5] J. P. Ryan, C. O. Davis, N. B. Tufillaro, R. M. Kudela, and B.-C. Gao, "Application of the hyperspectral imager for the coastal ocean to phytoplankton ecology studies in Monterey Bay, CA, USA," *Remote Sens.*, vol. 6, no. 2, pp. 1007–1025, Jan. 2014.

[6] A. Brook and E. B. Dor, "Quantitative detection of settled dust over green canopy using sparse unmixing of airborne hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 884–897, Feb. 2016.

[7] A. D. Vibhute, K. Kale, R. K. Dhumal, and S. Mehrotra, "Hyperspectral imaging data atmospheric correction challenges and solutions using QUAC and FLAASH algorithms," in *Proc. Int. Conf. Man Mach. Interfacing (MAMI)*, Apr. 2016, pp. 1–6.

[8] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7140–7151, Dec. 2017.

[9] D. Lunga, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *IEEE Signal Proc. Mag.*, vol. 31, no. 1, pp. 55–66, Jan. 2014.

[10] X. Sun, Q. Qu, N. M. Nasrabadi, and T. D. Tran, "Structured priors for sparse-representation-based hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 7, pp. 1235–1239, Jul. 2014.

[11] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A new approach to mixed pixel classification of hyperspectral imagery based on extended morphological profiles," *Pattern Recogn.*, vol. 37, no. 6, pp. 1097–1116, Jun. 2004.

[12] R. Cai, C. Liu, and J. Li, "Efficient phase-induced Gabor cube selection and weighted fusion for hyperspectral image classification," *Sci China Technol. Sc.*, vol. 65, no. 4, pp. 778–792, Mar. 2022.

[13] H. Yuan and Y. Y. Tang, "Spectral–spatial shared linear regression for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 934–945, Apr. 2017.

[14] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[15] H. Yu, Z. Xu, K. Zheng, D. Hong, H. Yang, and M. Song, "MSTNet: A multilevel spectral–spatial transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, Jun. 2022.

[16] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jul. 2015.

[17] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[18] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[19] X. Liu, A. H.-M. Ng, L. Ge, F. Lei, and X. Liao, "Multibranch fusion: A multibranch attention framework by combining graph convolutional network and CNN for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–17, Aug. 2024.

[20] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Nov. 2015, pp. 4959–4962.

[21] S. Ma, L. Liang, and S. Teng, "A lightweight classification method for hyperspectral remote sensing images," *J. Guangdong Univ. Technol.*, vol. 38, no. 3, pp. 29–35, May 2021, (in Chinese).

[22] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[23] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[24] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[26] D. Hong et al., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, Nov. 2022.

[27] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, Jan. 2022.

[28] Y. Peng, Y. Zhang, B. Tu, Q. Li, and W. Li, "Spatial–spectral transformer with cross-attention for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, Sep. 2022.

[29] S. Mei, C. Song, M. Ma, and F. Xu, "Hyperspectral image classification using group-aware hierarchical transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, Sep. 2022.

[30] E. Ouyang, B. Li, W. Hu, G. Zhang, L. Zhao, and J. Wu, "When multigranularity meets spatial–spectral attention: A hybrid transformer for hyperspectral image classificationwhen multigranularity meets spatial–spectral attention: A hybrid transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–18, Feb. 2023.

[31] R. Xu, X.-M. Dong, W. Li, J. Peng, W. Sun, and Y. Xu, "DBCTNet: Double branch convolution-transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–15, Feb. 2024.

[32] Z. Zhao, X. Xu, S. Li, and A. Plaza, "Hyperspectral image classification using groupwise separable convolutional vision transformer network," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–17, Mar. 2024.

[33] J. Feng, Q. Wang, G. Zhang, X. Jia, and J. Yin, "CAT: Center attention transformer with stratified spatial-spectral token for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–15, Mar. 2024.

[34] M. Jiang et al., "GraphGST: Graph generative structure-aware transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–16, Jan. 2024.

[35] Y. Li, Y. Luo, L. Zhang, Z. Wang, and B. Du, "MambaHSI: Spatial-spectral mamba for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–16, Jul. 2024.

[36] A. Gu et al., "Combining recurrent, convolutional, and continuous-time models with linear state space layers," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, Dec. 2021, pp. 572–585.

[37] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2022, pp. 1–12.

[38] K. Goel, A. Gu, C. Donahue, and C. Ré, "It's raw! audio generation with state-space models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 162, Jul. 2022, pp. 7616–7633.

[39] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *Proc. 1st Conf. Lang. Model.*, Aug. 2024, pp. 1–16.

[40] M. A. Ahamed and Q. Cheng, "Timemachine: A time series is worth 4 mambas for long-term forecasting," *Proc. ECAI*, vol. 392, Oct. 2024, pp. 1688–1695.

[41] M. A. Ahamed and Q. Cheng, "TSCMamba: Mamba meets multi-view learning for time series classification," *Inform. Fusion*, vol. 120, pp. 1–18, Aug. 2025.

[42] X. Jiang, C. Han, and N. Mesgarani, "Dual-path Mamba: Short and long-term bidirectional selective structured state space models for speech separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Mar. 2025, pp. 1–6,.

[43] K. Li, G. Chen, R. Yang, and X. Hu, "SPMamba: State-space model is all you need in speech separation," 2024, *arXiv:2404.02063*.

[44] J. Ma, F. Li, and B. Wang, "U-Mamba: Enhancing long-range dependency for biomedical image segmentation," 2024, *arXiv:2401.04722*.

[45] J. Ruan and S. Xiang, "VM-UNet: Vision Mamba UNet for medical image segmentation," 2024, *arXiv:2402.02491*.

[46] D. Liang et al., "PointMamba: A simple state space model for point cloud analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2024, pp. 32653–32677.

[47] T. Zhang, X. Li, H. Yuan, S. Ji, and S. Yan, "Point could Mamba: Point cloud learning via state space model," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, Apr. 2025, pp. 10121–10130.

[48] J. Yao, D. Hong, C. Li, and J. Chanussot, "SpectralMamba: Efficient Mamba for hyperspectral image classification," 2024, *arXiv:2404.08489*.

[49] L. Huang, Y. Chen, and X. He, "Spectral-spatial Mamba for hyperspectral image classification," *Remote Sens.*, vol. 16, no. 13, Jul. 2024, Art. no. 2449.

[50] W. Zhou, S. ichiro Kamata, H. Wang, M. S. Wong, and H. C. Hou, "Mamba-in-Mamba: Centralized Mamba-cross-scan in tokenized Mamba model for hyperspectral image classification," *Neurocomputing*, vol. 613, Jan. 2025, Art. no. 128751.

[51] Y. He, B. Tu, B. Liu, J. Li, and A. Plaza, "3DSS-Mamba: 3D-spectral-spatial Mamba for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–16, Oct. 2024.

[52] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11953–11965.

[53] S. Liu et al., "More convnets in the 2020 s: Scaling up kernels beyond 51×51 using sparsity," in *Proc. Int. Conf. Learn. Representations*, May 2023, pp. 1–16.

[54] X. Ding et al., "UniRepLKNet: A universal perception large-kernel convnet for audio video point cloud time-series and image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 5513–5524.

[55] K. W. Lau, L.-M. Po, and Y. A. U. Rehman, "Large separable kernel attention: Rethinking the large kernel attention design in CNN," *Expert Syst. Appl.*, vol. 236, Feb. 2024, Art. no. 121352.

[56] H. Chen, X. Chu, Y. Ren, X. Zhao, and K. Huang, "PeLK: Parameter-efficient large kernel convnets with peripheral convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 5557–5567.

[57] C. Zhong, N. Gong, Z. Zhang, Y. Jiang, and K. Zhang, "LiteCCLKNet: A lightweight criss-cross large Kernel convolutional neural network for hyperspectral image classification," *IET Comput. Vis.*, vol. 17, no. 7, pp. 763–776, Jul. 2023.

[58] G. Sun et al., "Large kernel spectral and spatial attention networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, Jul. 2023.

[59] W. Lu, X. Wang, L. Sun, and Y. Zheng, "Spectral–spatial feature extraction for hyperspectral image classification using enhanced transformer with large-kernel attention," *Remote Sens.*, vol. 16, no. 1, p. 67, 2024.

[60] C. Wu, L. Tong, J. Zhou, and C. Xiao, "Spectral-spatial large kernel attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–14, Feb. 2024.

[61] Q. Sun, G. Zhao, Y. Fang, C. Fang, L. Sun, and X. Li, "MEA-EFFormer: Multiscale efficient attention with enhanced feature transformer for hyperspectral image classification," *Remote Sens.*, vol. 16, no. 9, Apr. 2024, Art. no. 1560.

[62] X. Liu et al., "Hyperspectral image classification using a multi-scale CNN architecture with asymmetric convolutions from small to large kernels," *Remote Sens.*, vol. 17, no. 8, Apr. 2025, Art. no. 1461.

[63] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *Comput. Vis. Media*, vol. 9, no. 4, pp. 733–752, Jul. 2023.

[64] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1–9.

[65] H. Gao, Y. Yang, C. Li, L. Gao, and B. Zhang, "Multiscale residual network with mixed depthwise convolution for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3396–3408, Apr. 2021.

[66] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H^2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sens. Environ.*, vol. 250, Dec. 2020, Art. no. 112012.

[67] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double–branch dual–attention mechanism network," *Remote Sens.*, vol. 12, no. 3, pp. 582:1–582:25, Feb. 2020.

[68] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-based adaptive spectral–spatial kernel resnet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7831–7843, Sep. 2021.

[69] S. K. Roy, A. Deria, C. Shah, J. M. Haut, Q. Du, and A. Plaza, "Spectral–spatial morphological attention transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, Feb. 2023.