Tensorized High-Order Hypergraph Convolutional Network for Hyperspectral Image Classification

Jin-Yu Yang, Jing-Hua Yang, Heng-Chao Li, *Senior Member, IEEE*, Shaohui Mei, *Senior Member, IEEE*, Qian Du, *Fellow, IEEE*, and Antonio Plaza, *Fellow, IEEE*

***Abstract*—In recent years, graph convolutional networks (GCNs) have gained increasing attention in hyperspectral image (HSI) classification due to their good ability to model the pairwise relationships between two pixels. However, it is difficult to effectively model more complex relationships among multiple pixels with simple graphs. To solve this problem, we propose a novel tensorized high-order hypergraph convolutional network (TH^{2}GCN) for HSI classification. Specifically, the hypergraph structure is employed to effectively model complex spatial relationships between pixels in HSIs, and we propose a new tensor-based algebraic representation of hypergraphs as a powerful strategy for describing the high-order interaction structures of the hypergraph. Besides, by extending the adjacency matrix-based GCN to the tensor domain and exploiting the tensor decomposition, the TH^{2}GCN method is designed to efficiently extract high-order discriminative information from the hypergraph at low complexity for improving HSI classification performance. Furthermore, the construction of the adjacency tensor on all the data requires a huge amount of memory, especially for large-scale remote sensing images. To this end, the TH^{2}GCN is trained and tested for HSI data in a minibatch fashion. Experimental results on three HSI datasets prove that the performance of the proposed method outperforms the comparison methods.**

***Index Terms*—Hypergraph convolutional networks, hyperspectral image (HSI) classification, tensor decomposition.**

I. Introduction

WITH the development of remote sensing technology, hyperspectral sensors capture reflected radiation in many narrow and contiguous the detailed spectral channels along the wavelength axis. Therefore, hyperspectral images (HSIs) contain plenty of spectral and spatial information due to their high spectral and spatial resolution. As a result, HSIs have been successfully applied to a wide range of applications, such as agriculture [1], [2], [3], [4], forestry [5], and geology [6]. Classification task assigning each pixel to a unique category with the spatial and spectral information has been an interesting and popular research topic [7], [8], [9], [10], [11]. HSI classification is greatly challenged by the complex spatial distribution and the spectral heterogeneity of objects.

Over the past few decades, a variety of conventional methods and deep learning (DL) methods have been successfully designed for HSI classification. Specifically, conventional methods, such as support vector machine (SVM) [12], [13], random forest [14], sparse-collaborative representation [15], [16], multinomial logistic regression [17], and so on, design handcrafted features to solve problems with small labeled samples, nonlinearity, and high dimensionality in HSI classification. However, the defects of these methods are the dependence on the prior information and the loss of robustness, leading to poor classification performance under different scenarios. Due to the ability of DL models to automatically extract complex and hierarchical features using multilayer network architectures, many classical DL methods including autoencoder networks [18], convolutional neural networks (CNNs) [19], [20], [21], [22], generative adversarial networks (GANs) [23], capsule networks [24], [25], and recurrent neural networks (RNNs) [26], transformer [27], [28], have become feasible options for HSI classification. CNNs are capable of extracting discriminative spectral–spatial features, which have predominantly served as the backbone for most DL-based HSI classification methods [29]. Subsequently, to mitigate the challenge of limited training samples, GANs have been exploited for improving HSI classifiers with additional generated hyperspectral samples [30]. Nevertheless, these methods adopt the max-pooling strategy to offer invariance, ignoring the spatial relationship of significant features. To the end, capsule networks implement a dynamic routing-by-agreement mechanism to explicitly model hierarchical relationships between capsules, often yielding superior performance compared to standard CNNs. Since HSIs can be considered as sequential data, several RNN-based methods have emerged that can effectively utilize sequential dependencies to extract features and classify HSIs [31]. More recently, transformer models have also been introduced to tackle the HSI classification problem

Received 2 March 2025; revised 5 August 2025; accepted 15 September 2025. Date of publication 23 September 2025; date of current version 9 October 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62271418 and Grant 12401605, in part by the Natural Science Foundation of Sichuan Province under Grant 2023NSFSC0030 and Grant 2024NSFSC1467, in part by the Postdoctoral Fellowship Program of China Postdoctoral Science Foundation (CPSF) under Grant GZC20232198, in part by CPSF under Grant 2024M752661, in part by the Fundamental Research Funds for the Central Universities under Grant 2682024CX017, and in part by the State Key Laboratory of Intelligent Geotechnics and Tunneling (FSDI). *(Corresponding author: Jing-Hua Yang.)*

Jin-Yu Yang, Jing-Hua Yang, and Heng-Chao Li are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: yangjinghua110@126.com; hcli@home.swjtu.edu.cn).

Shaohui Mei is with the School of Electronics and Information, Northwestern Polytechnical University, Xi’an 710129, China.

Qian Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762 USA.

Antonio Plaza is with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, 10003 Cáceres, Spain.

Digital Object Identifier 10.1109/TGRS.2025.3613347

by leveraging their powerful self-attention mechanisms [32]. The previous DL methods have achieved good performance in HSI classification, which, however, neglect the intrinsic geometric properties in HSIs.

In recent years, researchers have increasingly adopted graph neural networks (GNNs) for HSI classification by transforming HSI data into a non-Euclidean structure graph and then using the irregular convolution on the graph for learning a better representation [33], [34]. Currently, GNNs can be categorized into spectral-based methods and spatial-based methods according to graph theory and message propagation, respectively. Especially, the spectral-based GNN, called the graph convolutional network (GCN), defines graph convolution from the spectral perspective. The first attempt is to encode the HSI into a graph and then process it with the GCN to propagate information between adjacent pixels based on their spectral similarity and spatial distance [35]. The spectral–spatial GCN was proposed for HSI classification by combining the intrinsic spectral information and the local spatial window information [36]. However, the above GCN-based methods cannot construct a reliable initial graph. The proposed method considered the graph generation of the graph convolution as a variable to optimize, aiming to obtain a locally reliable initial graph structure [37]. Due to the large number of pixels in HSIs, treating each pixel as a graph node leads to a massive amount of computation and limits its applicability. Wan et al. [38] used superpixels instead of pixels as the nodes, significantly reducing the size of the graph and enhancing the practicality of GCN. Nevertheless, this method ignores the information of HSI pixels, which limits the performance of models in pixel-level classification tasks. Yang et al. [39] simultaneously updated the pixel- and region-level information to capture the information hidden in HSI pixels and spatial-contextual knowledge within HSI regions. In addition, Huang et al. [40] introduced an adaptive pixel-level and superpixel-level feature fusion transformer network, which integrates pixel-level and superpixel-level features with the transformer.

The spatial-based GNN, named graph attention network (GAT), is another type of GNN that defines the convolutional operation and typically involves updating the node representations based on the representations of its neighboring nodes. It utilizes the attention mechanism to dynamically adjust the weight between nodes during feature extraction. Dong et al. [41] proposed weighted feature fusion of CNNs to explore the complementary characteristics of superpixel-based GAT and pixel-based CNN for HSI classification. A multistage superpixel-guided sparse GAT was proposed to remove some task-irrelevant edges and assign a unique attention coefficient to each remaining edge by trimming the graph by spectral sparsity [42]. To fully utilize the strengths of GNN and CNN, Yu et al. [43] proposed a novel graph-polarized fusion network consisting of two branches: the fused GNN classifier in the GNN branch conducts feature learning on large, irregular target regions using both GCN and GAT as feature extraction operators. By discarding the redundant computing parts in GAT, the computation complexity of the model is reduced while maintaining its accuracy [44].

Despite the promising performance of GNN-based HSI classification methods, these methods can only capture the relationships between paired pixels (the second-order relationship), but cannot describe the relationships between multiple pixels (the high-order relationship). For this reason, some efforts have been made to generalize well-known graph signal processing techniques to more complex graphs such as hypergraphs, which enable the capture of higher order relationships among multiple pixels [45]. For example, Ma et al. [46] proposed an improved hypergraph neural network (HGNN) named feature fusion hypergraph neural network (F^2HNN) to input the HSI and the extracted hypergraph into the HGCNN for effective learning. The active HGNN method used the hypergraph to reveal the complex nonpairwise relationships between the HSI pixels and introduced an active learning strategy to alleviate the challenge of the limited labeled pixels for HSI classification [47]. Duan et al. [48] designed a structure-preserved HGNN to learn the structured semantic features of HSI from multiple pixel nodes by integrating local regular convolution and irregular hypergraph convolution. To learn the hidden and important relations represented in the HSI data, the dynamic HGNN was designed to dynamically update the hypergraph model and capture the global spatial information for HSI classification [49]. However, the above methods often use the incidence matrix to represent the structure of the hypergraph, which may ignore the symmetry of the relationships in the data, thereby losing structural information about the original hypergraph [50]. Therefore, these HSI classification methods still have potential for further improvement.

In this article, we propose a tensorized high-order hypergraph convolutional network (TH^{2}GCN) for HSI classification. The main contributions of this article are presented as follows.

1) To effectively capture the complex spatial relationships among pixels in HSIs, we innovatively employ the hypergraph structure, where hyperedges flexibly connect multiple pixel nodes to model complex high-order interactions. More importantly, we propose a novel tensor-based algebraic representation for hypergraphs that can fully describe the intrinsic high-order interaction structure embedded in the constructed hypergraph.
2) The proposed TH^{2}GCN is the high-order generalization of the traditional GCN, which enhances the model's ability to efficiently extract high-order discriminative information from the hypergraph, thereby substantially improving HSI classification performance. Moreover, to address the computational bottleneck caused by the exponential complexity of directly processing high-order outer product features, the canonical polyadic (CP) decomposition is used to reduce the model's computational complexity to a linear scale, thereby achieving the balance between high-order feature representation and computational efficiency.
3) The construction of the adjacency tensor on all the data requires a massive amount of memory, especially for large-scale remote sensing problems. To this end, the TH^{2}GCN is trained and tested for HSI data in a minibatch fashion to reduce the need for experimental

platforms. The experimental results on three public HSI datasets show that the proposed method outperforms several comparison methods by 0.61%, 2.07%, and 0.73% in terms of overall accuracy (OA) for the Indian Pines, Houston, and Liao River Estuary datasets, respectively.

The rest of this article is outlined as follows. Section II reviews the related work. Section III describes the TH^{2}GCN method. Section IV compares TH^{2}GCN with state-of-the-art methods on three HSI datasets. Section V concludes this article.

II. Related Work

In this section, we summarize some related GCN, hypergraph convolutional network, and CP decomposition.

A. Graph Convolution Network

GCNs have attracted increasing attention in recent years, which map the graph to the frequency domain space via the Fourier transform and realize the convolutional operation in the time domain by performing multiplication of the frequency domain space. Spectral graph theory, which focuses on the fundamental properties of graphs and uses algebraic methods to analyze the specification of the adjacency or Laplacian matrices of graphs, has laid the foundation for GCNs. Then, to avoid the high computational complexity arising from eigen decomposition, Chebyshev GCN utilizes the truncated expansion of Chebyshev polynomials to fit convolution kernels. Motivated by a first-order approximation of spectral graph convolutions, Kipf and Welling [51] further learned hidden layer representations that encode both the local graph structure and the features of the nodes.

In GCN, the graph is utilized to describe paired relationships in non-Euclidean spaces. An undirected graph is formally defined as $G = (V, E, A)$, where V is the set of nodes, E is the set of edges, and A is the adjacency matrix. The graph Laplacian matrix L is represented as $L = D - A$ where D is a diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. The symmetric normalized Laplacian matrix is $L = I - D^{-1/2}AD^{-1/2}$, where I is the identity matrix. As a result, the convolution operation of the GCN model can then be expressed as

$$X^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X^l W\right) \tag{1}$$

where X is the input feature matrix in the lth layer, W is the weight matrix, X^{l+1} is the output matrix in the lth layer, and $\sigma(\cdot)$ is a nonlinear activation function.

B. Hypergraph Convolution Network

As discussed in the above GCN-based methods, pairwise modeling in a simple graph structure is inadequate. However, these existing GCN frameworks limit their applications in dealing with the high-order correlation of complex data in practice. To address this problem, hypergraphs are generalizations of simple graphs, where hyperedges can connect any number of vertices, thereby representing multiway relationships which are ubiquitous in many real-world networks including neuroscience [52], social networks [53], [54], [55], and bioinformatics [56], [57]. Furthermore, in order to deal with high-order data correlation during representation learning, Gao et al. [53] proposed HGNN for modeling and learning beyond-pairwise complex correlations, where the hypergraph Laplacian is first approximated and introduced into the deep hypergraph learning method to accelerate learning. Hao et al. [58] designed the HGNN to extract the local and global structure information via the hyperedge (i.e., nonphysical connection) constructions. Ma et al. [59] proposed a spatial, temporal, and spatio-temporal HGNN to learn high-order semantic information.

A hypergraph is defined as $G = (V, E, H)$, which includes a vertex set V, a hyperedge set E, and an incidence matrix H. The hypergraph convolution layer of HGNN can be formulated by

$$X^{l+1} = \sigma\left(D_{(v)}^{-\frac{1}{2}}HWD_{(e)}^{-1}H^T D_{(v)}^{-\frac{1}{2}}X^l P\right) \tag{2}$$

where P is a matrix to be learned during the training process, $D_{(e)jj} = \sum_{i=1}^{|v|} H_{ij}$ and $D_{(v)ii} = \sum_{j=1}^{|v|} W_{jj}H_{ij}$ are diagonal degree matrices of edges and vertices, respectively. X^l and X^{l+1} are the input and output of the lth layer, respectively. $\sigma(\cdot)$ is a nonlinear activation function like ReLU.

C. CP Decomposition

Vectors are 1-D, matrices are 2-D, and tensors then represent 3-D and beyond. Therefore, tensors can be conceptualized as higher order matrices [60], [61], [62]. Tensor analysis is developed based on tensor operations. CP decomposition [63] is the famous classical tensor decomposition, which factorizes a higher order tensor into a sum of several rank-1 tensor components. Given a k-order tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times \dots \times N_k}$, CP decomposition of $\mathcal{X}$ is denoted as

$$\begin{aligned}\mathcal{X} &\approx \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(k)} \\ &\approx \mathcal{I} \times {}_1 A^{(1)} \times {}_2 A^{(2)} \cdots \times {}_k A^{(k)}\end{aligned} \tag{3}$$

where R is the CP rank. One significant challenge associated with tensors is that their rank is not easily determined. In fact, establishing the rank of a given tensor is NP-hard. $\circ$ denotes the vector outer-product. $\mathcal{I} \in R^{R \times R \cdots \times R}$ is the identity tensor where all diagonal elements $\mathcal{I}_{i,\dots,i} = 1$. $A^{(1)} \in \mathbb{R}^{N_1 \times R}, \dots, A^{(k)} \in \mathbb{R}^{N_k \times R}$ denote a series of factor matrices.

Over the past years, CP decomposition has been progressively integrated with HGNN to reduce the temporal/spatial complexity from exponential to linear using partially symmetric CP decomposition. Recently, Hua et al. [64] proposed a tensorized GNN leveraging the symmetric CP decomposition to efficiently parameterize permutation-invariant multilinear maps for modeling nonlinear high-order multiplicative interactions among nodes. In order to improve computational efficiency for large hypergraphs, Wang et al. [65] localized the T-spectral convolution approach to formulate the T-spatial convolution and further devised a novel tensor message-passing algorithm for practical implementation by studying a compressed adjacency tensor representation. Considering the relationships among vertices within a hyperedge,

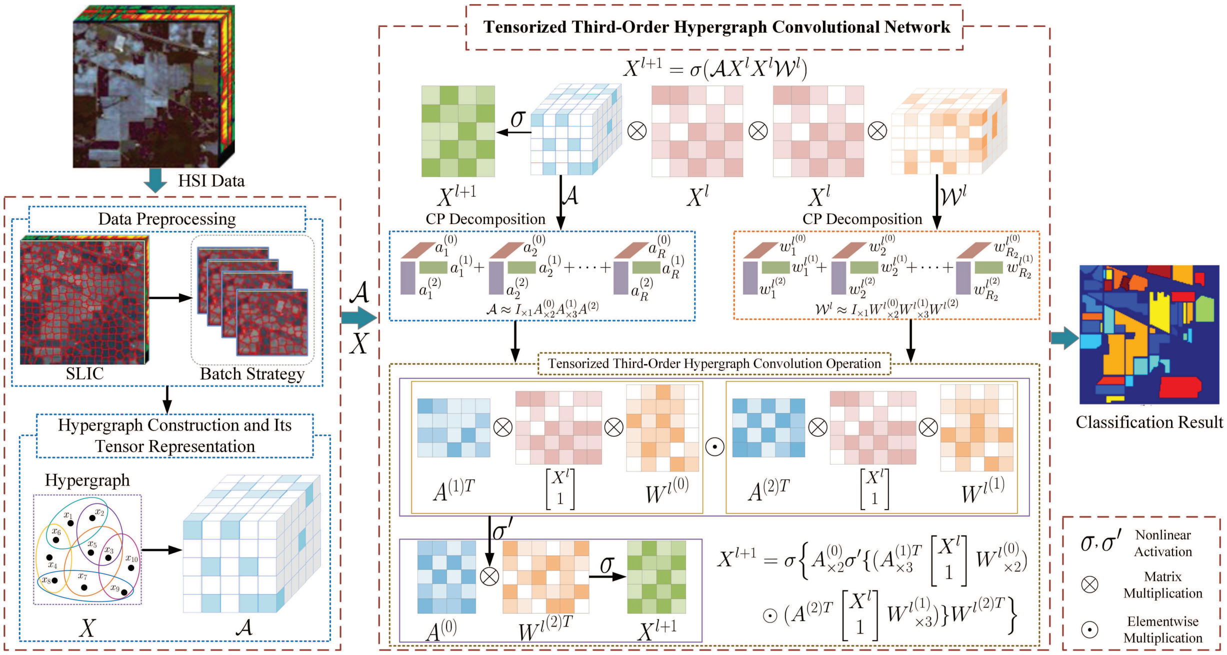

Tensorized Third-Order Hypergraph Convolutional Network
$X^{l+1} = \sigma(\mathcal{A}X^l X^l \mathcal{W}^l)$
σ
X^{l+1}
$\mathcal{A}$
X^l
X^l
$\mathcal{W}^l$
CP Decomposition
CP Decomposition
$\mathcal{A} \approx I_{\times1}A^{(0)}_{\times2}A^{(1)}_{\times3}A^{(2)}$
$\mathcal{W}^l \approx I_{\times1}W^{l(0)}_{\times2}W^{l(1)}_{\times3}W^{l(2)}$
Tensorized Third-Order Hypergraph Convolution Operation
$A^{(1)T}$
$\begin{bmatrix} X^l \\ 1 \end{bmatrix}$
$W^{l(0)}$
$A^{(2)T}$
$W^{l(1)}$
σ'
$A^{(0)}$
$W^{l(2)T}$
X^{l+1}
$X^{l+1} = \sigma\Big\{A^{(0)}_{\times2}\sigma'\{(A^{(1)T}_{\times3}\begin{bmatrix} X^l \\ 1 \end{bmatrix}W^{l(0)}_{\times2}) \odot (A^{(2)T}\begin{bmatrix} X^l \\ 1 \end{bmatrix}W^{l(1)}_{\times3})\}W^{l(2)T}\Big\}$
HSI Data
Data Preprocessing
SLIC
Batch Strategy
Hypergraph Construction and Its Tensor Representation
Hypergraph
X
$\mathcal{A}$
Classification Result
σ, σ' Nonlinear Activation
$\otimes$ Matrix Multiplication
$\odot$ Elementwise Multiplication

Fig. 1. Architecture of the tensorized third-order hypergraph convolutional network for HSI classification. First, the segmented HSI data is divided into several batches. Then, the hypergraph is constructed to represent the high-order correlation of multiple pixels in HSIs, which is described by an adjacency tensor. Furthermore, the adjacency tensor of the hypergraph is fed into the hypergraph convolution layer of a tensorized third-order hypergraph convolutional network to obtain discriminative features from multiple pixel nodes. Finally, the model employs the loss function to train this network and obtain the classification result.

Song et al. [52] converted the incidence matrix into a weighted adjacency tensor to describe the structure of the brain hypernetwork and employed the hypergraph signal processing tools to reveal high-order relationships in brain hypernetworks. Besides, Wang et al. [66] designed a novel HGNN to utilize partially symmetric CP decomposition to reduce time/space complexity from exponential to linear. However, when the high-order tensor is used to describe the hypergraph, a large number of samples will increase the dimensionality of the tensor, requiring enough storage space.

III. Proposed Method

A. Method Overview

The section gives detailed information about the TH^{2}GCN for HSI classification, aiming to establish a general framework for representation learning on the given HSI. The third-order TH^{2}GCN is illustrated in Fig. 1. First, we divide the segmented HSI data into several batches. Then, for each batch, the hypergraph is employed to model the complex spatial relationships among pixels in HSIs, and the adjacency tensor is used to describe the intrinsic high-order interaction structure of the constructed hypergraph. Furthermore, a novel TH^{2}GCN is designed to extract the high-order discriminative features from the correlations among pixels of HSIs with low complexity. Finally, the loss function is used to train the proposed model. The details of these modules are described below.

B. Hypergraph Construction and Its Tensor Representation

Given HSI data $X_o \in \mathbb{R}^{H\times W\times D}$, H, W, and D are the numbers of height, width, and spectral bands. To reduce computational complexity, the simple linear iterative clustering (SLIC) algorithm, which recursively grows the local clusters using k-means algorithm and provides fast and effective segmentation, segments the HSI into superpixels. After segmentation by SLIC, the HSI is partitioned into N superpixels, which are denoted as $X = \{x_1, x_2, \ldots, x_N\}$, where the feature of each superpixel is computed by the average of all pixels within that superpixel.

Previous GCN-based HSI classification methods directly convert the HSI into a graph in the preprocessing process and extract features based on this constructed graph [38]. However, the simple graph can only model the pairwise interactions between two nodes, which cannot capture high-order interactions among multiple pixels. In recent years, hypergraphs have emerged as high-dimensional generalizations of simple graph structures, which can express more complex relationships than pairwise relationships. To this end, the hypergraph is constructed to discover the complex structure of HSI data, where a hyperedge can connect an arbitrary number of pixels. Specifically, considering that the neighboring pixels in HSIs often belong to the same class, the hypergraph is constructed to model the neighborhood relationships between the central pixel and its neighboring pixels in terms of the spatial domain. The hypergraph effectively represents the local homogeneity of the HSI. As shown in Fig. 2(a), three pixels adjacent in position are connected by a hyperedge.

Furthermore, we employ tensor-based descriptors to represent the hypergraph $\mathcal{G} = (V, E, \mathcal{A})$. $V = \{v_1, \ldots, v_N\}$ is the set of vertexs composed of X, $E = \{e_1, \ldots, e_{k\}}$ is the set of hyperedges. Similar to an adjacency matrix whose 2-D entries indicate whether and how two nodes are pairwise connected

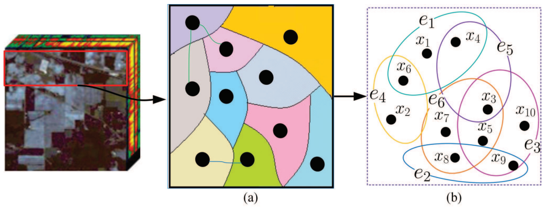


Fig. 2. Construction of a hypergraph. (a) SLIC algorithm segments the entire HSI into a small amount of compact superpixels, each of which is treated as a hypergraph node. (b) Hypergraph is constructed by modeling the neighboring information of the pixels, where the circles and lines represent the hypergraph nodes and hyperedges, respectively.

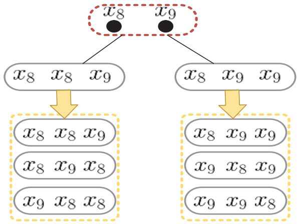


Fig. 3. Interpretation of generalizing e_2 in Fig. 2(b) to hyperedges with $c = 3$ nodes.

by a simple edge, we adopt an adjacency tensor whose entries indicate whether and how corresponding subsets of M nodes are connected by hyperedges to describe hypergraphs. Thus, the adjacency tensor $\mathcal{A}$ of the hypergraph $\mathcal{G}$ is a high-order extension of the adjacency matrix, which is characterized as an Mth-order N-dimensional tensor. The entries of the above adjacency tensor $\mathcal{A} \in \mathbb{R}^{\underbrace{N\times,\ldots,\times N}_{M}}$ of the hypergraph $\mathcal{G}$ are defined as

$$\mathcal{A} = \left(a_{i_1,\ldots,i_M}\right), \quad \mathcal{A}_{i_1,\ldots,i_M} \in \mathbb{R}$$
$$1 \le i_1,\ldots,i_M \le N \tag{4}$$

where the entry $a_{i_1,\ldots,i_M}$ indicates whether the nodes $x_{i_1},\ldots,x_{i_M}$ are in the same hyperedge, i.e., whether a hyperedge $e_i = \{x_{i_1},\ldots,x_{i_c}\} \in E$ exists. Let c be the number of vertices contained in one hyperedge. Therefore, if the weight is nonzero, the hyperedge exists; otherwise, the hyperedge does not exist. In an undirected simple graph, $a_{ij} = a_{ij} \neq 0$ implies the connection of the node v_j to the node v_i. Therefore, the indices $\{i_1,\ldots,i_M\}$ for adjacency entries are chosen from all possible permutations of $\{i_1,\ldots,i_M\}$ with at least one appearance for each element of the hyperedge set.

To understand the physical meaning of the adjacency tensor, we use a third-order adjacency tensor to represent the hypergraph in Fig. 2(b) as an example. Notably, in Fig. 2(b), different hyperedges may contain different numbers of nodes. How to use the adjacency tensor to represent the hyperedges with different numbers of nodes may become an issue. To solve this problem, these hyperedges are divided into three cases and processed as follows.

The Number of Nodes in the Hyperedges Is Smaller Than the Order Number of the Adjacency Tensor ($c < M$): To represent such hyperedge $e_l = \{x_{l_1}, x_{l_2},\ldots,x_{l_c}\} \in E$ with the number of vertices $c < M$ in an Mth-order adjacency tensor, the node exists in a node set would be added many times in one hyperedge until the number of nodes in the hyperedge is equal to the order number of the adjacency tensor. As shown in Fig. 2(b), the hyperedge $e_2 = \{x_8, x_9\}$ only contains two nodes. To this end, we construct two sets of nodes $\{x_8, x_9, x_8\}$ and $\{x_8, x_9, x_9\}$ by adding the node in the hyperedge, which could be interpreted as generalizing the original hyperedge with $c = 2$ to hyperedges with $c = 3$ nodes. Next, the entries in the third-order adjacency tensor are calculated by all possible permutations of $\{x_8, x_9, x_8\}$ and $\{x_8, x_9, x_9\}$, which represent the correlation of x_8 and x_9. Therefore, the hyperedge $e_2 = \{x_8, x_9\}$ can be represented by the entries $a_{889}, a_{898}, a_{988}, a_{899}, a_{989}, a_{998}$ in the third-order tensor $\mathcal{A}$, which is illustrated in Fig. 3.

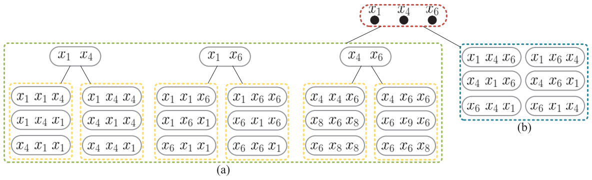


Fig. 4. Using a third-order adjacency tensor to represent the hyperedge with three nodes.

The Number of Nodes in the Hyperedges Equals the Order Number of the Adjacency Tensor ($c = M$): In order to use the Mth-order adjacency tensor to represent the hyperedges $e_l = \{x_{l_1}, x_{l_2},\ldots,x_{l_c}\} \in E$ with the number of vertices $c = M$, the possible combination of c nodes in the adjacency tensor can represent the relationship of c nodes and various possible subsets of these nodes are also used to represent the relationship of nodes in these subsets. We take the hyperedge e_1 containing x_1, x_4, x_6 in Fig. 2(b) as an example. The number of nodes in e_1 is 3, such that $c = 3$. When $c = M$, the intuitive idea is that the hyperedge e_1 is characterized by $a_{146}, a_{164}, a_{461}, a_{416}, a_{614}, a_{641}$ in the adjacency tensor. The third-order adjacency tensor only represents the relationship between three different nodes in a hyperedge, resulting in the loss of information. Thus, in the hyperedge e_1, there is a relationship between three nodes, which can also indicate that there is a relationship between two nodes in the e_1. This situation is similar to that of $c < M$. For example, in order to represent the relationship between two nodes $\{x_1, x_4\}$ in the third-order adjacency tensor, such that, x_1 or x_4 is added to $\{x_1, x_4\}$ to make the number of nodes reach 3. The above operation is to preserve richer hypergraph information, which is shown in Fig. 4.

The Number of Nodes in the Hyperedges Is Greater Than the Order Number of the Adjacency Tensor ($c > M$): To represent such hyperedge $e_l = \{x_{l_1}, x_{l_2},\ldots,x_{l_c}\} \in E$ with the number of vertices $c > M$ in an Mth-order adjacency tensor, these node subsets, where the number of nodes in subsets is less than or equal to the order number of the adjacency tensor, from the node set of hyperedge are used to represent this hyperedge. For example, as shown in Fig. 2(b), the hyperedge $e_3 = \{x_3, x_5, x_9, x_{10}\}$ contains four nodes. On the one hand, the hyperedge $e_3 = \{x_3, x_5, x_9, x_{10}\}$ can be represented by $\{x_3, x_5, x_9\}, \{x_3, x_5, x_{10}\}, \{x_5, x_9, x_{10}\}$. On the other hand, these subsets $\{x_3, x_5\}, \{x_3, x_9\}, \{x_3, x_{10}\}, \{x_5, x_9\}, \{x_5, x_{10}\}, \{x_9, x_{10}\}$ are obtained to represent the relation of two nodes. These node subsets can be described in the adjacency tensor by the above

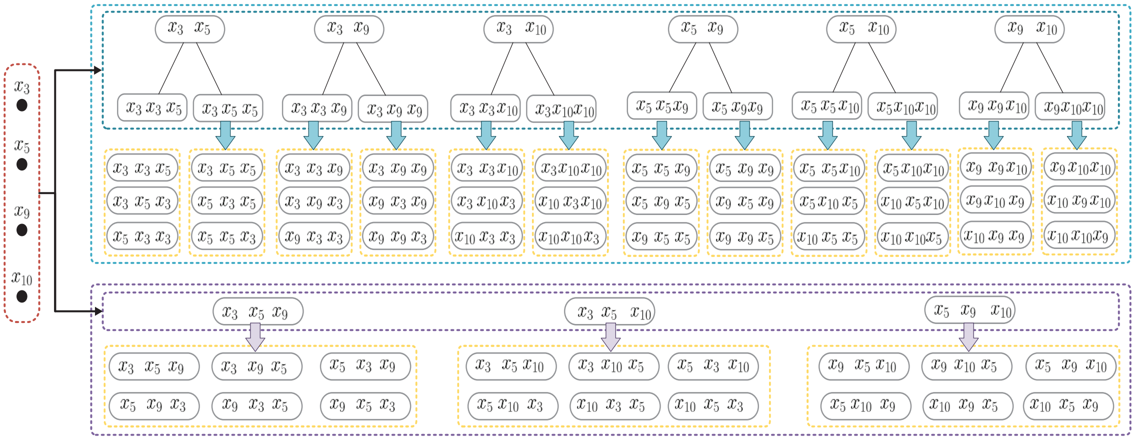


Fig. 5. Using the third-order adjacency tensor to represent the hyperedge with four nodes.

methods. Fig. 5 shows the entries of the adjacency tensor for the hyperedge e_3 in Fig. 2(b).

Furthermore, GCN-based methods without feature normalization can result in numerical instabilities because direct application of the convolutional layer changes the scale of the feature vectors. As a consequence of this, similar to the degree-normalized adjacency matrix in GCN, we adopt the famous normalized adjacency tensor extension. The value of $\mathcal{A}_{i_1,\ldots,i_M}$ indicates the weight of the hyperedge, which can be calculated as

$$\mathcal{A}_{i_1,\ldots,i_M} = \begin{cases} e_i \dfrac{c}{(M-1)!}, & e_i = \left\{v_{i_1}, \ldots, v_{i_M}\right\} \in E \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $(M-1)!$ is the total number of permutations for length-M hyperedge e^M. c is the exact number of vertices contained in one hyperedge. Equation (5) is a structural definition of hyperedge weights, which ensures that the weight of the hyperedge is larger when it contains more vertices.

C. Tensorized Third-Order Hypergraph Convolutional Network

Most GCN-based methods focus on the simple pixel-to-pixel structure between two pixels to learn representations for HSI classification. The graph convolution operation is defined by

$$X^{l+1} = \sigma\left(AX^l W^l\right) \tag{6}$$

where $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix, which is constructed to model the pairwise relationships of two pixels. $X^l \in \mathbb{R}^{N\times D}$ is the input feature matrix in the lth layer. $W^l \in \mathbb{R}^{D\times D'}$ denotes the weight matrix in the lth layer. $X^{l+1} \in \mathbb{R}^{N\times D'}$ is the output feature matrix in the lth layer.

However, the graph convolution operation of (6) can only extract features from the simple graph. Recently, the hypergraph has been used to model the complex spatial relationships among multiple pixels in HSIs. And the adjacency tensor is introduced for describing the intrinsic high-order interaction structure of a hypergraph, which is shown in Section III-B. Thus, how to develop the hypergraph convolution operation to learn discriminative features from the adjacency tensor is a key issue. Therefore, we design the TH^{2}GCN via the CP tensor decomposition of the adjacency and weight tensors to learn more effective features from the relationship among multiple vertices in the hypergraph at low complexity for gaining better performance in the HSI classification task.

The proposed TH^{2}GCN framework is the high-order generalization of traditional GCN, enabling the processing of signals in a high-order hypergraph to capture the polyadic relationships among the pixels. Specifically, similar to the representation equivalence between the aggregation scheme and the adjacency matrix formulation for GCN, as the representation learning method, the proposed TH^{2}GCN comes with a message-passing mechanism capturing high-order interactions in the hypergraph by generalizing graph convolution to a hypergraph structure to learn more discriminative features from multiple pixel nodes. The hypergraph convolution operation can be defined by the generalization of the graph convolution operation. The proposed TH^{2}GCN is a more general framework than traditional GCN. For example, a third-order adjacency tensor is required to represent the interaction of three nodes in the hypergraph. Therefore, the hypergraph convolution operation based on the third-order adjacency tensor is defined as

$$X^{l+1} = \sigma\left(\mathcal{A}X^l X^l \mathcal{W}^l\right) \tag{7}$$

where $\mathcal{A} \in \mathbb{R}^{N\times N\times N}$ represents the normalized adjacency tensor. $X^l \in \mathbb{R}^{N\times D}$ is the input feature of the lth layer. $\mathcal{W}^l \in \mathbb{R}^{D\times D\times D'}$ is the learnable weight tensor. $X^{l+1} \in \mathbb{R}^{N\times D'}$ is the output feature of the lth layer. $\sigma(\cdot)$ represents an activation function, such as the ReLU$(\cdot)$.

In addition to the feature extraction capability, the number of parameters and the time complexity are also important evaluation indicators for the proposed TH^{2}GCN method. For the third-order hypergraph convolutional operation in (7), the parameter space complexity is $O(D^2D')$ and the computational time complexity is $O(N^3D + N^2D^2 + ND^2D')$. It is very clear that the size of the hypergraph adjacency tensor grows exponentially with the order number, and such extremely large storage and computational complexity is unacceptable. This phenomenon is known as the curse of dimensionality, and a proper tensor decomposition format can effectively solve this problem. Thus, the hypergraph convolution operation in TH^{2}GCN is defined via the CP tensor decomposition of the adjacency tensor, resulting in a more expressive high-order node interaction scheme. In particular, similar to the eigenvalue decomposition for a matrix, CP decomposition analyzes tensors via factorization, which decomposes the tensor representing the hypergraph adjacency tensor. Therefore, the third-order adjacency tensor $\mathcal{A}$ in (7) can be decomposed into a set of component rank-one tensors, which are represented as

$$\mathcal{A} \approx I_{\times 1} A^{(0)}_{\times 2} A^{(1)}_{\times 3} A^{(2)} \tag{8}$$

where $A^{(0)}, A^{(1)}, A^{(2)} \in \mathbb{R}^{N \times R_1}$ are a series of factor matrices. Generally, we have the rank $R_1 \leq N$ in an adjacency tensor. $I \in \mathbb{R}^{R_1 \times R_1 \times R_1}$ is an identity tensor.

Next, by plugging (8) into (7), the hypergraph convolution operation using the CP decomposition can be reformulated as

$$X^{l+1} = \sigma\left(I_{\times 1} A^{(0)}_{\times 2} A^{(1)}_{\times 3} A^{(2)} X^l X^l \mathcal{W}^l\right). \tag{9}$$

Equation (9) can be simplified as

$$X^{l+1} = \sigma\left(A^{(0)}_{\times 2} A^{(1)}_{\times 3} A^{(2)} X^l X^l \mathcal{W}^l\right). \tag{10}$$

Another third-order tensor is the learnable weight tensor $\mathcal{W}^l$. So, we also apply the CP decomposition structure with the rank R_2 to decompose the weight tensor $\mathcal{W}^l$, which can be expressed as

$$\mathcal{W}^l \approx I_{\times 1} W^{l(0)}_{\times 2} W^{l(1)}_{\times 3} W^{l(2)} \tag{11}$$

where $W^{l(0)}, W^{l(1)} \in \mathbb{R}^{D \times R_2}$, and $W^{l(2)} \in \mathbb{R}^{D' \times R_2}$ are a set of factor matrices. $I \in \mathbb{R}^{R_2 \times R_2 \times R_2}$ is an identity tensor. Thus, (10) can be expressed as

$$X^{l+1} = \sigma\left(A^{(0)}_{\times 2} A^{(1)}_{\times 3} A^{(2)} X^l X^l W^{l(0)}_{\times 2} W^{l(1)}_{\times 3} W^{l(2)}\right) \tag{12}$$

where $W^{l(0)}$, $W^{l(1)}$, and $W^{l(2)}$ are learnable weight matrices in the lth layer.

We could also rewrite (12) with the element-wise dot aggregation function form as

$$\begin{aligned} &X^{l+1} \\ &= \sigma\left\{A^{(0)}_{\times 2}\left\{\left(A^{(1)T}_{\times 3} X^l W^{l(0)}_{\times 2}\right) \odot \left(A^{(2)T} X^l W^{l(1)}_{\times 3}\right)\right\} W^{l(2)T}\right\} \end{aligned} \tag{13}$$

where $\odot$ denotes the componentwise product.

Low-order information can also be very important in some cases. However, (13) considers more on the second-order interactions and ignores some first-order information. We concatenate the original feature vector with a scalar 1 to help TH^{2}GCN with low-order information modeling. By using homogeneous coordinates, i.e., appending an entry equal to one to each of the input feature X^l, (13) can be written as

$$\begin{aligned} X^{l+1} = \sigma\Bigg\{A^{(0)}_{\times 2}\Bigg\{&\left(A^{(1)T}_{\times 3}\begin{bmatrix} X^l \\ 1 \end{bmatrix} W^{l(0)}_{\times 2}\right) \\ &\odot \left(A^{(2)T}\begin{bmatrix} X^l \\ 1 \end{bmatrix} W^{l(1)}_{\times 3}\right)\Bigg\} W^{l(2)T}\Bigg\}. \end{aligned} \tag{14}$$

Since the dot product of many vectors would lead to the numerical insataibility empirically, a new activation function σ' is added in (14). In experiments, the commonly used activation function Tanh($\cdot$) is used. The final expression of the third-order hypergraph convolution operation with the CP decomposition is represented as

$$\begin{aligned} X^{l+1} = \sigma\Bigg\{A^{(0)}_{\times 2}\sigma'\Bigg\{&\left(A^{(1)T}_{\times 3}\begin{bmatrix} X^l \\ 1 \end{bmatrix} W^{l(0)}_{\times 2}\right) \\ &\odot \left(A^{(2)T}\begin{bmatrix} X^l \\ 1 \end{bmatrix} W^{l(1)}_{\times 3}\right)\Bigg\} W^{l(2)T}\Bigg\}. \end{aligned} \tag{15}$$

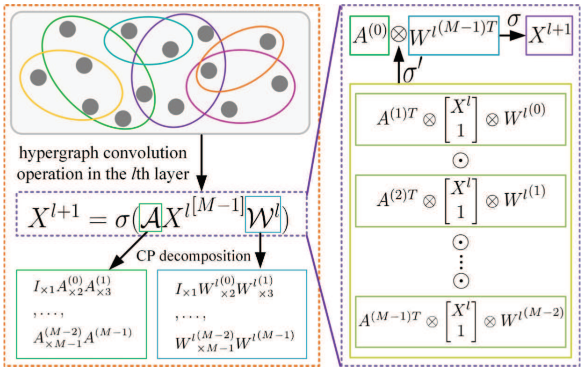

hypergraph convolution
operation in the lth layer
$X^{l+1} = \sigma(\mathcal{A}X^{l^{[M-1]}}\mathcal{W}^l)$
CP decomposition
$I_{\times 1}A^{(0)}_{\times 2}A^{(1)}_{\times 3}$
$,\ldots,$
$A^{(M-2)}_{\times M-1}A^{(M-1)}$
$I_{\times 1}W^{l(0)}_{\times 2}W^{l(1)}_{\times 3}$
$,\ldots,$
$W^{l(M-2)}_{\times M-1}W^{l(M-1)}$
$A^{(0)} \otimes W^{l(M-1)T}$
σ
X^{l+1}
σ'
$A^{(1)T} \otimes \begin{bmatrix} X^l \\ 1 \end{bmatrix} \otimes W^{l(0)}$
$\odot$
$A^{(2)T} \otimes \begin{bmatrix} X^l \\ 1 \end{bmatrix} \otimes W^{l(1)}$
$\odot$
$\odot$
$A^{(M-1)T} \otimes \begin{bmatrix} X^l \\ 1 \end{bmatrix} \otimes W^{l(M-2)}$

Fig. 6. Hypergraph convolution operation based on the Mth-order adjacency tensor.

After exploiting the CP decomposition of the adjacency and weight tensors as shown in (15), the parameter space complexity and computational time complexity can be reduced into $O((2D + D')R_2)$ and $O(2R_1D(N + R_2) + NR_2(R_1 + D'))$, respectively. In summary, the proposed TH^{2}GCN method can efficiently extract high-order discriminative information from the hypergraph with low complexity, thereby improving HSI classification performance.

D. Tensorized Mth-Order Hypergraph Convolutional Network

In Section III-C, the hypergraph convolution operation of the tensorized third-order HGCN is introduced in detail to facilitate understanding. In addition, when many data are constructed as hypergraphs in practical applications, the hyperedges often contain more than three nodes. Therefore, the higher order tensors will be utilized to represent these hypergraphs more accurately. The hypergraph convolution operation of TH^{2}GCN based on the Mth-order adjacency tensor is defined as

$$X^{l+1} = \sigma\left(\mathcal{A}X^{l^{[M-1]}}\mathcal{W}^l\right) \tag{16}$$

where $\mathcal{A} \in \mathbb{R}^{\underbrace{N\times,\ldots,\times N}_{M}}$ represents the normalized adjacency tensor. $X^{l^{[M-1]}}$ is $M-1$ times outer product of the input feature $X^l \in \mathbb{R}^{N \times D}$ in the lth layer. $\mathcal{W}^l \in \mathbb{R}^{D \times \ldots \times D \times D'}$ is a learnable weight tensor in the lth layer. The output features are defined as $X^{l+1} \in \mathbb{R}^{N \times D'}$. The CP decomposition is used to decompose $\mathcal{A}$ and $\mathcal{W}^l$. The hypergraph convolution operation based on the Mth-order adjacency tensor is shown in Fig. 6, which is written as

$$\begin{aligned} X^{l+1} = \sigma\Bigg\{A^{(0)}_{\times 2}\sigma'\Bigg\{&\left(A^{(1)T}_{\times 3}\begin{bmatrix} X^l \\ 1 \end{bmatrix} W^{l(0)}_{\times 2}\right) \\ &\odot \ldots \odot \left(A^{(M-1)T}\begin{bmatrix} X^l \\ 1 \end{bmatrix} W^{l(M-2)}_{\times M}\right)\Bigg\} W^{l(M-1)T}\Bigg\}. \end{aligned} \tag{17}$$

Algorithm 1 Training TH^{2}GCN for HSI Classification

```
Input: HSI data $X$, Ground truth $Y$;
Segment the whole image into superpixels via SLIC algorithm;
Split hyperpixels into multiple batches;
Construct a set of adjacency tensors $\{\mathcal{A}_s | s = 1, \ldots, \lceil (N/K) \rceil\}$;
These adjacency tensors $\mathcal{A}_s$ can be decomposed into a set of component rank-one tensors according to Eq. (8);
Parameter setting and weights initialization;
for epoch$\leq$ 3000 do
    for s=1 to $\lceil (N/K) \rceil$ do
        Calculate the features $X^1$ of the $1st$ layer by Eq. (15);
        Calculate the features $X^2$ of the $2nd$ layer by Eq. (15);
    Minimize the loss function by Eq. (19);
    Back propagation and update the parameters by Adam optimizer;
Output: Classification map $O$.
```

E. Batch Strategy

Most GCN-based networks are trained and tested by using full-batch gradient descent, where all samples are utilized to perform gradient descent in each iteration. If the proposed method directly applies the same full-batch strategy, sufficient storage space is required. Since the reduction of the order number breaks the structure of the hypergraph, we adopt the dimension reduction of each order. A feasible solution to address this issue is to use batch processing.

Specifically, given a segmented HSI data, superpixels are denoted as $X = \{x_1, x_2, \ldots, x_N\}$. We construct a node sampler with a budget $K(K \ll N)$ and repeatedly apply the sampler to HSI data until each node is sampled. Next, we input these sampled batches to construct a set of hypergraphs $\mathcal{G}_s = \{(V_s, E_s, \mathcal{A}_s) | s = 1, \ldots, \lceil (N/K) \rceil\}$, where $\lceil\rceil$ denotes the ceiling operation. Our TH^{2}GCN can perform the hypergraph convolution operation in batches. By collecting the outputs of all batches, the final output in the $(l+1)$th layer can be reformulated as

$$H = \left[X_1^{1+1}, \ldots, X_{N/K}^{1+1}\right]. \tag{18}$$

Finally, the output classification results of TH^{2}GCN can be learned by $O = \text{Softmax}(H)$. To train this network, the commonly used cross-entropy loss function is employed to minimize the discrepancy between the predicted labels generated by the network and the ground-truth labels of the training samples. Hence, the objective function of TH^{2}GCN can be formulated as

$$\text{Loss} = -\sum_{g \in y_G} \sum_{f=1}^{C} Y_{gf} lnO \tag{19}$$

where y_G is the set of indices corresponding to the labeled examples, C denotes the number of classes, and Y denotes the true label. The implementation detail of our TH^{2}GCN is shown in Algorithm 1.

IV. Experimental Results

In this section, we conduct numerical experiments on three different HSI datasets (i.e., Indian Pines, Houston, and Liao River Estuary datasets) to evaluate the effectiveness of the proposed TH^{2}GCN. In particular, the detailed information of these HSI datasets and several baseline models are described in Sections IV-A and IV-B, respectively. In addition, hyperparameter experiments are conducted in Section IV-C to gain a better knowledge of the proposed model. Next, Section IV-D presents the experimental results to verify the effectiveness of the proposed TH^{2}GCN method. Section IV-E provides the impact of different numbers of training samples on the proposed method. Finally, a number of ablation experiments are carried out in Section IV-F to confirm the impact of each component of our method.

A. Experimental Data

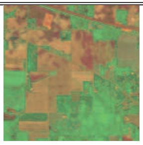
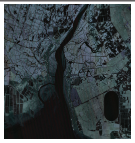
In our experiments, three HSI datasets are adopted to evaluate the performance of our model, whose detailed information are described as follows.

1) *Indian Pines Dataset:* The dataset was recorded by the airborne visible-infrared imaging spectrometer (AVIRIS) over the Indian Pines test site in North-Western Indiana in 1992. It contains 145 × 145 pixels with a spatial resolution of 20 m/pixel and 220 spectral bands in the wavelength range from 0.4 to 2.5 μm. By removing 20 water absorption and noisy bands, 200 bands are used for all experiments. This image contains two-thirds of agricultural lands, one-third of forests, and other natural perennial vegetation, for a total of 16 land-cover classes.
2) *Houston Dataset:* The data was gathered with the compact airborne spectrographic imager (CASI) sensor over the University of Houston, TX, USA, and the surrounding area in June 2012, which were introduced in the 2013 IEEE Geoscience and Remote Sensing Society (GRSS) Data Fusion contest. Both hyperspectral and LiDAR data have a spatial size of 349 × 1905 pixels with a spatial resolution of 2.5 m. For the hyperspectral data, there exist 144 spectral bands with the wavelength ranging from 0.38 to 1.05 μm. The Houston dataset contains 15 land cover classes, such as "Tree," "Water," and "Commercial."
3) *Liao River Estuary Dataset:* The dataset was recorded by GF5-AHSI on the Liao River Estuary wetland, Panjin City, Liaoning Province. This image contains 680 × 640 pixels with the spatial resolution of 30m and 330 spectral bands. The image is comprised of nine different land-cover classes, such as "Phragmites australis, "Paddy fields," "Suaeda salsa," and "Intertidal muds." The number of labeled samples in the Liao River Estuary dataset is small and discrete.

For the three datasets described above, we randomly select 30 labeled pixels in each class for training. Classes containing fewer than 30 samples are limited to 15 training samples. Meanwhile, the remaining samples are used as the test set to evaluate the classification performance of the proposed TH^{2}GCN method. Besides, the number of training samples

TABLE I

NUMBER OF TRAINING SAMPLES AND TOTAL SAMPLES FOR INDIAN PINES, HOUSTON, AND LIAO RIVER ESTUARY DATASETS

Indian Pines					Houston					Liao River Estuary				
														
		(a)					(b)					(c)		
NO.	Color	Class	Training	Total	NO.	Color	Class	Training	Total	NO.	Color	Class	Training	Total
1	🟦	Alfalfa	30	46	1	🟦	Healthy grass	198	1251	1	🟦	Paddy field	30	1589
2	🟦	Corn-notill	30	1428	2	🟦	Stressed grass	190	1254	2	🟦	Reed	30	1575
3	🟦	Corn-mintill	30	830	3	🟦	Synthetic grass	192	697	3	🟦	Architecture	30	416
4	🟦	Corn	30	237	4	🟦	Tree	188	1244	4	🟩	Pond	30	869
5	🟦	Grass-pasture	30	483	5	🟦	Soil	186	1242	5	🟩	Suaeda salsa	30	1426
6	🟦	Grass-trees	30	730	6	🟦	Water	182	325	6	🟧	Naked Beach	30	636
7	🟩	Grass-pasture-mowed	15	28	7	🟩	Residential	196	1268	7	🟥	Rivers	30	1424
8	🟩	Hay-windrowed	30	478	8	🟩	Commercial	191	1244	8	🟥	Breeding pond	30	1478
9	🟩	Oats	15	20	9	🟩	Road	193	1252	9	🟫	Chao Gou	30	329
10	🟨	Soybean-notill	30	972	10	🟨	Highway	191	1227					
11	🟧	Soybean-mintill	30	2455	11	🟧	Railway	181	1235					
12	🟧	Soybean-clean	30	593	12	🟧	Parking lot 1	192	1233					
13	🟥	Wheat	30	205	13	🟥	Parking lot 2	184	469					
14	🟥	Woods	30	1265	14	🟥	Tennis court	181	428					
15	🟥	Buildings-Grass-Trees-Drives	30	386	15	🟥	Running track	187	660					
16	🟫	Stone-Steel-Towers	30	93										
Total			450	10249	Total			2832	15029	Total			270	9742

and all pixels of different classes in the Houston dataset are represented in Table I.

B. Baseline Models

To verify the effectiveness and classification accuracy of the proposed method, some classical classification methods are adopted as comparison methods. We compare our model with the following methods.

1) SVM [13]: SVM is the most classical classification method, which is performed on the well-known libsvm toolbox3 in our case.

2) CNN [19]: The model is composed of three 2-D convolutional blocks and a softmax layer. Each convolutional block involves a 2-D conventional layer, a BN layer, a max-pooling layer, and a ReLU activation layer.

3) GCN [51]: GCN applies a localized first-order spectral filter on graphs, which can be considered a convolutional neural network for processing graph-structured data.

4) FuNet [33]: The method combines features from CNNs and GCNs by additive fusion, element-wise multiplicative fusion, and concatenation fusion to improve classification results in HSIs.

5) MDGCN [38]: The MDGCN constructs graphs with different neighborhood scales to leverage the hidden spatial context at various scales. In addition, the method gradually refines the graphs during the convolution process to reduce the influence of a poorly predefined graph.

6) HCGN [67]: The HCGN method captures pixel-level fine information and superpixel-level long-distance structural information for HSI classification.

7) F^{2}HNN [46]: The F^{2}HNN network generates hyperedges from features of different modalities to construct a hypergraph representing multimodal features in HSIs. Subsequently, three feature fusion strategies are employed to integrate these multimodal features for HSI classification.

8) MF-RHCN [45]: The MF-RHCN method automatically updates the hyperedge weights to construct a hypergraph by fusing spectral and spatial features for HSI classification.

9) HGCN-MHF [68]: The HGCN-MHF proposes a CNN branch for capturing spatial and spectral pixel-level features, designs an HGCN branch to extract the superpixel-level feature, and develops a score-weighted feature fusion strategy to enhance the integration of features from both branches for HSI classification.

10) CSFGNet [69]: CSGNet utilizes an HGCN branch for superpixel-level high-order feature extraction and a CNN branch for pixel-level local feature enhancement.

In addition, four mainstream evaluation indexes [70], i.e., per-class accuracy, OA, average accuracy (AA), and kappa coefficient (κ), are adopted for the sake of evaluating and analyzing the performance of the proposed method presented in this article and other related works.

C. Experimental Settings

In the section, the effects of several crucial parameters, such as the order number of the adjacency tensor M, the batch size K, the rank R_1, the rank R_2, and the training epoch in the proposed TH^{2}GCN method, are evaluated in the following.

1) Analysis of the Order Number M: The adjacency tensor is the high-order extension of the adjacency matrix. The order number of a tensor M is the number of its dimensions. The comparison results of the proposed method using different orders are shown in Table II, from which the classification accuracies of the proposed method increase with the larger value of M. The results demonstrate that the proposed method benefits from higher order information modeling. Considering that both the number of network parameters and the time

TABLE II

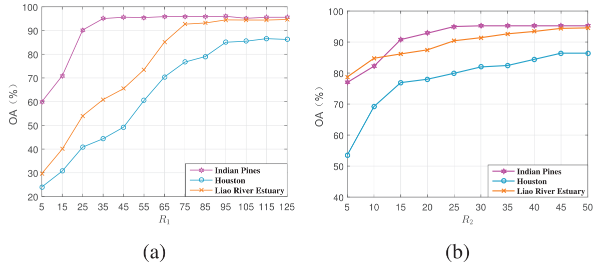
OA (%) Results of the Proposed Method Using Different Order Numbers M on Three Datasets

M	Indian Pines	Houston	Liao River Estuary
3	95.21	85.03	95.19
4	**95.68**	**85.53**	**95.44**

TABLE III

OA (%) Results of the Proposed Method Using Different Batch Sizes on Three Datasets

K	Indian Pines	Houston	Liao River Estuary
32	92.25	84.37	94.65
64	93.26	85.15	95.17
128	95.73	85.45	**95.19**
256	**96.49**	**85.28**	94.89
512	96.48	84.88	94.35



(a)

(b)

Fig. 7. OA (%) results of the proposed TH^{2}GCN method under different ranks. (a) OA (%) results of the proposed TH^{2}GCN method under different ranks of adjacency tensors. (b) OA (%) results of the proposed TH^{2}GCN method under different ranks of weight tensors.

complexity increase as the value of M increases, the third-order adjacency tensor is chosen to represent the hypergraph structure ($M = 3$) in our experiments.

2) Analysis of Batch Size K: The batch size refers to the number of samples in each batch, which also affects the construction of the hypergraph. Table III shows the classification accuracies of the proposed model under different batch sizes in the range of {32, 64, 128, 256, 512}. As can be seen, OA results initially increase and then plateau as the batch size increases on three datasets. This improvement is attributed to the fact that more samples result in more complex correlations, emphasizing the effectiveness of high-order relations. In order to strike a balance between experimental accuracy and time complexity, 128 is selected for all datasets in the following experiments.

3) Analysis of Rank R_1: R_1 is the rank from the CP decomposition of the adjacency tensor, which controls the tradeoff between parameter efficiency and expressiveness. We evaluate the influences of rank R_1 on model performance by setting its value of R_1 to between 5 and 125 in increments of 10 for the Indian Pines, Houston, and Liao River Estuary datasets when the dimension of the adjacency tensor is 128 × 128 × 128. These results of TH^{2}GCN using different ranks are shown in Fig. 7(a), where the classification result initially improves and then gradually becomes more consistent as the value of R_1 increases. In the Indian Pines dataset, similar features are distributed continuously over a large area in space, and the pixel values in the spatial neighborhood change smoothly. Therefore, when the optimal CP decomposition rank is small, it can effectively capture the main features of the constructed adjacency tensor and obtain the optimal classification result. In contrast, the Houston and Liao River Estuary datasets exhibit significant spatial heterogeneity, small target sizes, and complex feature distributions, with more complex data structures. Therefore, it is necessary to choose a higher rank in CP decomposition for accurate representation of the constructed adjacency tensor and satisfactory classification results. In subsequent experiments, we set R_1 to 35, 115, and 95 for the Indian Pines, Houston, and Liao River Estuary datasets, respectively.

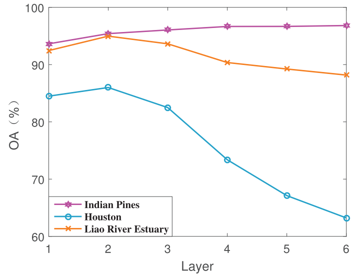


Fig. 8. OA (%) results of the proposed TH^{2}GCN method with different network layers on three datasets.

4) Analysis of Rank R_2: In the proposed method, the network depth is fixed at two layers, with each layer containing a weight tensor for feature transformation. For computational convenience, we set the CP decomposition ranks of both weight tensors to the same value. To evaluate the impact of the CP decomposition rank R_2 on model performance, experiments are conducted by varying R_2 within the set {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}. The OA results of TH^{2}GCN using different ranks R_2 are shown in Fig. 7(b). The classification performance of TH^{2}GCN initially improves with increasing rank before plateauing beyond a certain point. The Houston and Liao River Estuary datasets exhibit stronger spatial heterogeneity than the Indian Pines dataset, requiring more parameters to learn effective features. Therefore, R_2 is set to 25 for the Indian Pines. In addition, R_2 is set to 45 for the Houston and the Liao River Estuary datasets.

5) Analysis of Network Layer: To investigate the effect of layer depth on the classification performance of the proposed TH^{2}GCN method, experiments are performed across the Indian Pines, Houston, and Liaohe Estuary datasets with depths ranging from 1 to 6 layers. As depicted in Fig. 8, classification accuracy initially increases with greater layer depth, but subsequently declines beyond a certain threshold. For the Indian Pines dataset, the highest accuracy is achieved with five layers. In contrast, the best performance on the Houston and Liaohe Estuary datasets is achieved with a depth of two layers. To balance performance and computational efficiency, a network depth of two layers is selected for subsequent experiments on three datasets.

6) Analysis of Epoch: To analyze the empirical convergence of the proposed method, Fig. 9 presents the cross-entropy loss values over training samples at different epochs on three HSI datasets. For both the Indian Pines and

TABLE IV
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE INDIAN PINES DATASET

Class	SVM	CNN	GCN	FuNet	MDGCN	HCGN	F^{2}HNN	MF-RHCN	HGCN-MHF	CSGNet	TH^{2}GCN
1	**100.00**	**100.00**	97.82	**100.00**	**100.00**	99.43	**100.00**	**100.00**	91.30	**100.00**	**100.00**
2	55.22	58.58	48.67	60.52	78.68	85.67	76.03	89.05	**92.90**	90.79	91.47
3	80.50	85.00	64.10	66.75	94.88	94.06	**97.37**	91.62	**97.37**	94.05	99.33
4	99.52	**100.00**	91.98	90.34	96.14	99.42	98.55	99.51	**100.00**	96.62	**100.00**
5	77.04	85.43	89.86	91.83	90.73	94.49	93.37	93.81	82.30	**96.42**	**100.00**
6	76.86	71.57	95.75	98.29	97.86	97.45	**99.71**	99.71	98.86	97.51	98.71
7	**100.00**	**100.00**	92.85	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
8	95.76	96.42	89.96	99.78	**100.00**	99.71	98.43	99.10	99.78	99.20	**100.00**
9	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
10	56.47	59.76	69.75	71.23	94.16	93.25	67.62	87.15	**98.62**	94.63	95.99
11	59.09	64.12	61.22	56.53	**96.58**	90.92	93.77	92.57	91.29	91.27	90.69
12	74.78	76.38	53.29	70.52	90.76	96.35	81.17	90.23	91.82	**96.66**	92.62
13	98.86	98.86	97.56	**100.00**	97.71	99.26	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
14	87.61	86.48	85.30	84.21	99.68	99.39	95.78	**100.00**	97.88	99.06	99.09
15	94.94	93.54	46.63	77.81	99.44	99.28	**100.00**	94.66	99.44	98.65	**100.00**
16	**100.00**	**100.00**	96.77	**100.00**	**100.00**	98.85	**100.00**	**100.00**	**100.00**	99.68	**100.00**
OA (%)	71.57	73.94	69.71	72.93	93.84	93.78	89.69	93.57	94.80	94.71	**95.41**
AA (%)	84.79	86.00	80.10	85.49	96.04	96.78	93.86	96.09	96.34	97.16	**97.99**
κ (%)	67.97	70.50	65.86	69.45	92.94	92.90	88.13	92.62	94.07	93.95	**94.76**

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE HOUSTON DATASET

Class	SVM	CNN	GCN	FuNet	MDGCN	HCGN	F^{2}HNN	MF-RHCN	HGCN-MHF	CSGNet	TH^{2}GCN
1	81.01	80.05	82.52	81.95	81.10	81.97	76.54	83.09	**86.69**	80.47	86.57
2	82.24	84.02	80.73	83.27	83.45	77.82	79.79	83.83	80.44	86.84	**93.36**
3	82.97	82.17	95.04	74.65	97.02	99.73	99.60	**100.00**	95.92	99.92	**100.00**
4	90.81	82.95	91.76	91.00	92.89	88.51	90.43	89.48	73.12	**91.82**	89.86
5	97.63	99.33	96.49	99.52	98.57	**100.00**	**100.00**	**100.00**	**100.00**	95.47	**100.00**
6	79.72	81.81	78.32	82.51	83.80	96.03	79.02	81.12	88.73	**95.80**	94.43
7	76.12	**89.08**	77.61	83.11	68.56	73.04	66.69	68.65	76.82	71.74	77.23
8	43.40	67.61	50.71	64.95	60.39	67.67	66.66	64.10	68.04	60.36	**74.54**
9	79.41	72.14	75.73	77.33	78.47	**87.79**	66.00	72.99	76.84	84.59	59.63
10	**90.15**	45.46	88.03	54.34	45.36	60.54	64.96	67.47	68.43	58.01	83.94
11	63.00	67.36	67.93	61.19	83.87	89.52	86.24	**99.71**	93.48	93.62	92.32
12	84.15	85.39	74.35	81.36	65.99	93.68	87.31	92.41	90.41	**95.81**	88.74
13	**89.82**	81.05	74.03	88.77	58.24	75.37	54.38	81.05	81.40	83.23	83.18
14	80.97	76.51	91.90	79.75	99.19	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.41
15	66.60	81.18	93.02	93.86	97.66	**100.00**	**100.00**	**100.00**	98.61	**100.00**	96.12
OA (%)	78.79	77.87	79.98	78.69	77.80	83.86	80.05	83.92	83.16	83.90	**85.99**
AA (%)	79.20	78.41	81.21	79.84	79.64	86.11	81.17	85.59	85.26	86.51	**87.95**
κ (%)	77.15	76.03	78.36	77.03	76.07	82.55	78.32	82.53	81.74	82.52	**84.79**

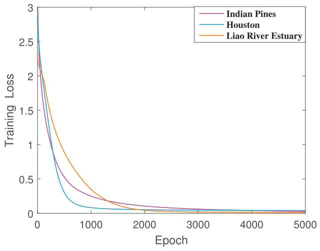


Fig. 9. Empirical convergence analysis of the proposed method on three HSI datasets.

Liao River Estuary datasets, it is evident that the training loss rapidly decreases before iteration 3000, then the rate of decrease slows noticeably thereafter. For the Houston dataset, the training loss significantly reduces during the first 1000 epochs and then remains stable with negligible variation from 1000 to 5000 epochs. We set the training epoch as 3000 to ensure that the model has a sufficient chance to converge.

D. Classification Results

To evaluate the effectiveness of the proposed TH^{2}GCN method, we compare the proposed method with the baseline methods on the Indian Pines, Houston, and Liao River Estuary datasets. Tables IV–VI record the per-class accuracy, OA, AA, and κ of our method and all baseline methods. Furthermore, the visualizations of all methods on the Indian Pines, Houston, and Liao River Estuary datasets are displayed in Figs. 10–12. The experimental analysis is detailed as follows.

1) Results on the Indian Pines Dataset: The classification results of all baseline methods and TH^{2}GCN on the Indian Pines dataset are shown in Table IV, where the best results in each category (i.e., each row) are marked in bold. As shown in Table IV, we can find that compared with the second best method, the proposed TH^{2}GCN method achieves gains of 0.61%, 1.65% and 0.69% in OA, AA, and κ, respectively. Besides, TH^{2}GCN maintains the best performance in the

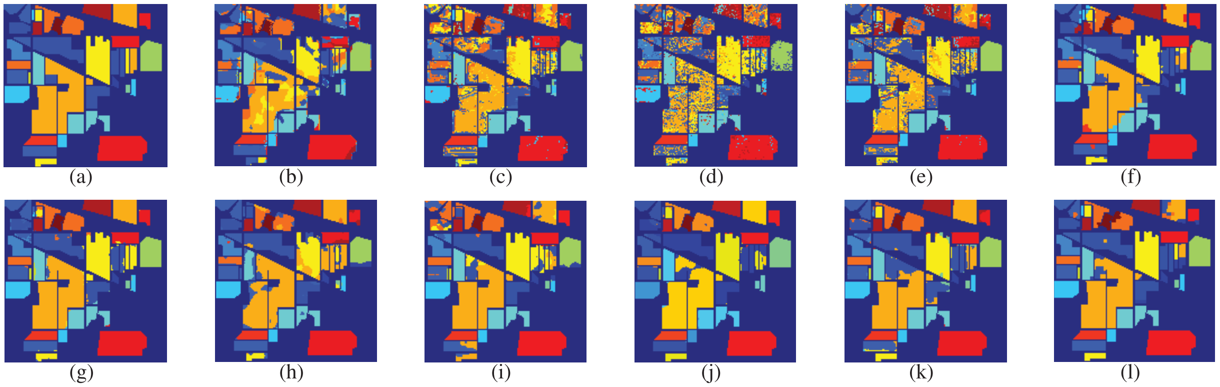


Fig. 10. Classification maps for the Indian Pines dataset. (a) Ground-truth map. (b) SVM. (c) CNN. (d) GCN. (e) FuNet. (f) MDGCN. (g) HCGN. (h) F^{2}HNN. (i) MF-RHCN. (j) HGCN-MHF. (k) CSGNet. (l) TH^{2}GCN.

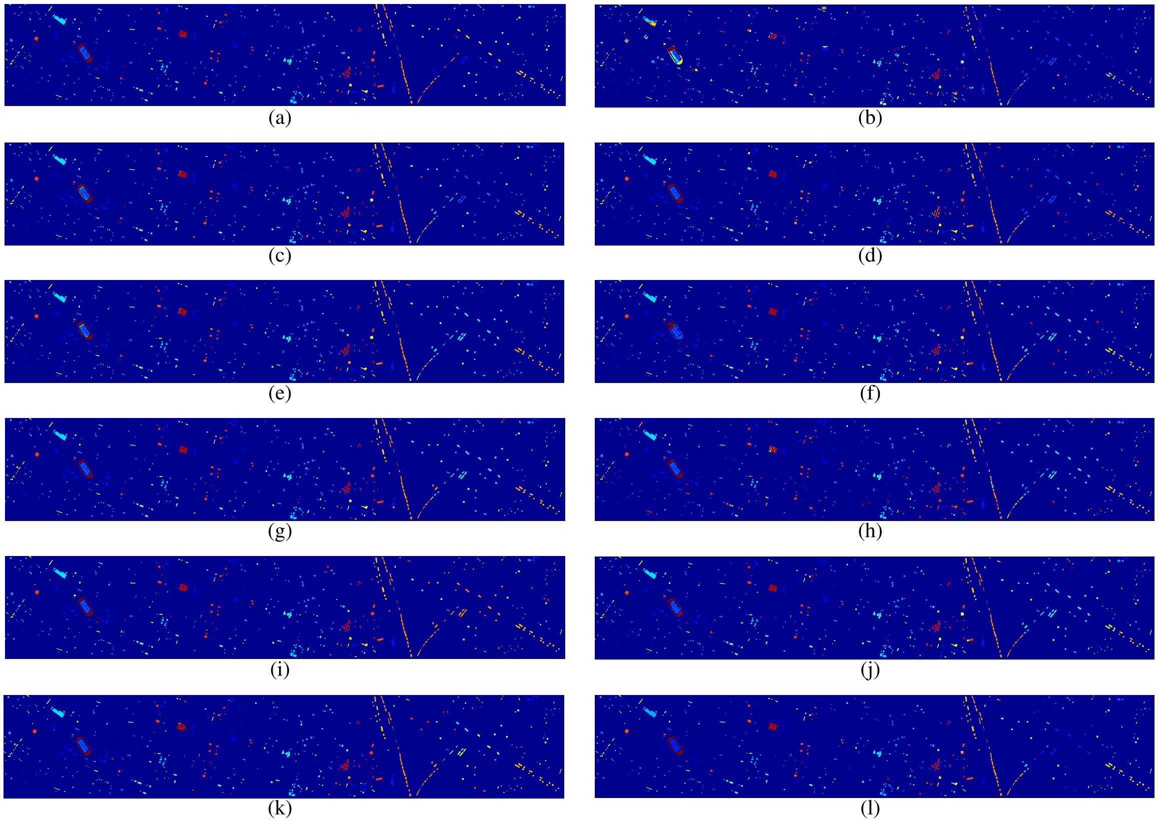


Fig. 11. Classification maps for the Houston dataset. (a) Ground-truth map. (b) SVM. (c) CNN. (d) GCN. (e) FuNet. (f) MDGCN. (g) HCGN. (h) F^{2}HNN. (i) MF-RHCN. (j) HGCN-MHF. (k) CSGNet. (l) TH^{2}GCN.

majority of cases, such as class 1, class 4, class 5, class 7, class 8, class 9, class 13, class 15, and class 16. This superior performance is due to the effectiveness of our method in extracting high-order information from the hypergraph. Specifically, the OA of F^{2}HNN is 89.69%, while the accuracy of GCN is 69.71%. The GCN model performs worse than the hypergraph GCN because the hypergraph structure can convey complicated high-order correlations among data. Besides, TH^{2}GCN offers an absolute improvement of 5.72% in OA over F^{2}HNN, which proves that the tensor-based representation of the hypergraph can better model the high-order interaction between pixels compared to the incidence matrix.

2) Results on the Houston Dataset: Table V shows the classification accuracies of the baseline methods and TH^{2}GCN on the Houston dataset. The OA, AA, and κ of our proposed method are 85.99%, 87.95%, and 84.79%, respectively. The classification performance of TH^{2}GCN outperforms all baseline methods. In particular, the SVM and CNN methods lag behind TH^{2}GCN by margins of 7.2% and 8.12%, respectively, in OA. This is attributed to their predominant focus on local information, which leads to a reduced capacity for balancing sample processing by disregarding global information. Significantly, the OA of TH^{2}GCN surpasses that of four GCN-based methods, such as GCN, FuNet, MDGCN, and HCGN, and four

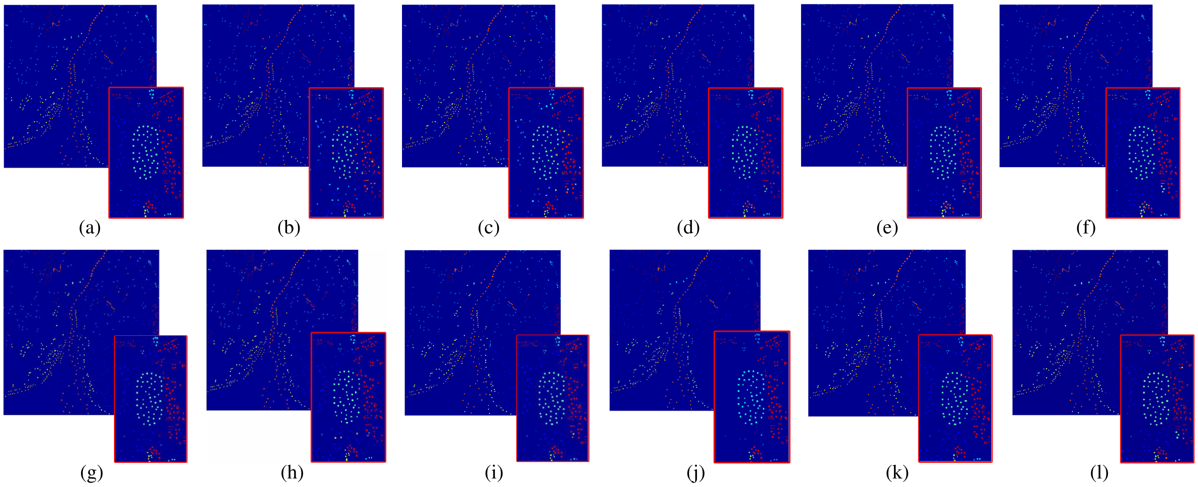


Fig. 12. Classification maps for the Liao River Estuary dataset. (a) Ground-truth map. (b) SVM. (c) CNN. (d) GCN. (e) FuNet. (f) MDGCN. (g) HCGN. (h) F^2HNN. (i) MF-RHCN. (j) HGCN-MHF. (k) CSGNet. (l) TH^2GCN.

TABLE VI

Classification Results of Different Methods on the Liao River Estuary Dataset

Class	SVM	CNN	GCN	FuNet	MDGCN	HCGN	F^2HNN	MF-RHCN	HGCN-MHF	CSGNet	TH^2GCN
1	68.44	65.23	96.66	93.45	97.31	97.12	91.46	97.18	94.80	96.24	**99.85**
2	68.16	52.75	90.28	90.81	86.92	91.67	86.02	90.10	88.30	91.03	**94.53**
3	60.10	69.43	90.38	85.49	90.93	96.66	92.49	94.04	**97.04**	94.30	95.85
4	70.32	94.99	96.55	96.31	90.46	76.15	82.95	79.74	**99.88**	94.92	96.18
5	67.19	66.40	95.79	92.62	94.69	88.57	**100.00**	94.99	85.45	92.28	90.90
6	97.19	93.89	99.68	**100.00**	**100.00**	**100.00**	96.86	99.17	99.04	94.92	**100.00**
7	65.27	80.63	91.01	90.60	97.70	**99.49**	93.47	96.34	97.73	93.66	96.77
8	82.04	80.46	90.26	88.19	86.95	**99.86**	99.45	98.27	97.34	98.23	91.19
9	82.94	94.31	69.60	79.93	86.96	94.72	**95.99**	91.63	95.92	95.18	91.63
OA (%)	71.94	73.52	92.71	91.59	92.68	93.97	93.13	94.02	94.05	94.42	**95.15**
AA (%)	73.52	77.56	91.13	90.82	92.44	93.81	93.19	93.49	95.05	94.53	**95.21**
κ (%)	67.82	69.84	91.59	90.30	91.56	93.03	92.08	93.10	93.15	93.55	**94.41**

HGCN-based methods, such as F^2HNN, MF-RHCN, HGCN-MHF, and CSGNet, by margins ranging from 2.07% to 8.19%. This underscores the efficacy of the hypergraph convolution operation based on the CP decomposition in enabling the model to learn complex discriminative features from high-order correlations between multiple pixels, thereby obtaining better classification results.

3) Results on Liao River Estuary Dataset: The experimental results on the Liao River Estuary dataset are presented in Table VI. The proposed model yields highly competitive classification accuracies of 95.15%, 95.21%, and 94.41%, which also achieves superior performance than the baseline methods. In particular, it can be found that the classification performance of CNN, GCN, FuNet, MDGCN, HCGN, F^2HNN, MF-RHCN, HGCN-MHF, and CSGNet is superior to SVM in terms of OA, AA, and κ in most cases. The above experimental results show that for traditional models, such as SVM, the way of converting the HSI data into vectors leads to the loss of the spatial and geometric structure information. Besides, it can be observed that TH^2GCN achieves the best OA, AA, and κ values, outperforming superpixel-based methods, MDGCN, HCGN, HGCN-MHF, and CSGNet. The reason is that the proposed TH^2GCN via the CP tensor decomposition of the hypergraph adjacency tensor achieves a message-passing mechanism from high-order interactions in the adjacency tensor, which extracts more discriminative features from multiple pixel nodes in HSI and yields higher classification accuracy.

The classification maps of different methods on the Indian Pines dataset are shown in Fig. 10. We can see that the classification map of TH^2GCN is closer to the ground-truth than other baseline methods. Specifically, the classification maps obtained by CNN, GCN, and FuNet suffer from noise mistakes within certain regions due to the lack of spatial context. Comparatively, the results of the proposed TH^2GCN method, MDGCN, HCGN, HGCN-MHF, and CSGNet yield smoother visual effects and show fewer misclassifications than other compared methods, which can obtain satisfactory performance in these small regions. This is because SLIC, as a preprocessing tool, can better capture contextual information. Furthermore, the proposed method can produce better classification results, demonstrating its effectiveness. In addition, Figs. 11 and 12 show the visual and qualitative comparison of different methods on the Houston and Liao River Estuary datasets. The classification map of TH^2GCN is evidently less noisy and smoother, demonstrating the superiority of TH^2GCN.

E. Impact of the Number of Labeled Examples

When given sufficient training samples, most HSI classification methods can achieve satisfactory performance. However, in most cases, the labeled samples are very limited

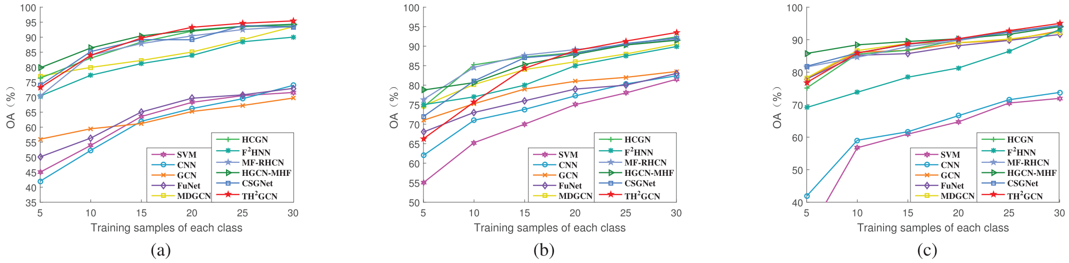


Fig. 13. OA (%) of all methods with different numbers of training samples for three HSI datasets. (a) Indian Pines. (b) Houston. (c) Liao River Estuary.

TABLE VII

ABLATION RESULTS WITH/WITHOUT THE SCALAR 1

Method	Indian Pines			Houston			Liao River Estuary		
	OA (%)	AA (%)	κ (%)	OA (%)	AA (%)	κ (%)	OA (%)	AA (%)	κ (%)
w/o scalar 1	94.69	97.11	93.93	79.23	80.05	77.43	92.67	93.22	91.55
TH^{2}GCN	**95.41**	**97.99**	**94.76**	**85.99**	**87.95**	**84.79**	**95.15**	**95.21**	**94.41**

TABLE VIII

EXPERIMENTAL RESULTS OF THE PROPOSED METHOD WITH DIFFERENT ACTIVATION FUNCTIONS ON THREE DATASETS

Method	Indian Pines			Houston			Liao River Estuary		
	OA (%)	AA (%)	κ (%)	OA (%)	AA (%)	κ (%)	OA (%)	AA (%)	κ (%)
w/o AF	95.09	97.75	94.40	72.79	72.51	70.43	94.04	94.54	93.14
ReLU	91.82	96.47	90.71	75.36	76.78	73.25	92.55	92.79	91.42
Softplus	91.17	96.11	89.97	73.15	76.57	70.87	92.14	92.81	90.95
Sigmoid	92.25	96.39	91.16	83.82	86.38	82.44	93.96	94.49	93.04
Tanh	**95.41**	**97.99**	**94.76**	**85.99**	**87.95**	**84.79**	**95.15**	**95.21**	**94.41**

due to expensive human and material resources. Therefore, small-sample learning capability, i.e., the performance of HSI classification methods with very few training samples, becomes another important evaluation index. In order to evaluate the classification performance of TH^{2}GCN and other HSI classification methods in the case of different training conditions, we perform experiments varying the number of training samples per class from 5 to 30 with an interval of 5. Notably, specifically for the Indian Pines dataset, when the number of pixels in certain classes does not reach the number of training samples, 15 samples per class are chosen as the training set. The OA results of some HSI classification methods on three datasets are reported in Fig. 13. Compared to other methods, too few samples, such as 5 or 10 training samples, result in poor performance in HSI classification. This reason is that the semisupervised TH^{2}GCN model is difficult to train well due to the presence of fewer training samples in some batches which leads to performance degradation. Besides, we can observe that the proposed method generally achieves higher classification results than the other baseline methods when the training samples are between 20 and 30 per class. These results benefit from the valuable high-order correlation information between multiple samples exploited in a hypergraph and the discriminative information extracted by the proposed hypergraph convolution operation, which makes up for the deficiency of prior information in the small training set and enhances its small-sample learning ability.

F. Ablation Study

Each component in the proposed method plays a pivotal role. In this section, a series of ablation studies is conducted to evaluate the performance of each module of the proposed model.

1) Influence of Connecting Scalar 1: The proposed TH^{2}GCN employs the technique of concatenating the scalar 1 to highlight low-order information. To test its effectiveness, an ablation model, called w/o scalar 1, is designed by removing the scalar 1. Table VII reports the classification results of the ablation model and the proposed model for the Indian Pines, Houston, and Liao River Estuary datasets. We can observe that the proposed model achieves gains of 0.72%, 6.76%, and 2.48% on three datasets compared to the model without the scalar 1 in terms of OA, respectively. These results demonstrate that the technique of concatenating the scalar 1 effectively improves the expressiveness of the model.

2) Influence of the Activation Function: The activation function Tanh(·) in (15) is used to avoid numerical instabilities during training caused by the repeated products. To evaluate its contribution, an ablation model is designed by removing the activation function, called w/o AF. Furthermore, we also replace tanh with another common activation function, such as ReLU, Softplus, and Sigmoid, while maintaining all other settings the same. Table VIII reports the classification results of the proposed method with different activation functions. The proposed method with Tanh outperforms the w/o AF model by margins of 0.32%, 13.2%, and 1.11% in OA for three datasets. In addition, the proposed method, employing either Tanh or Sigmoid activation functions, demonstrates significantly superior performance compared to the variant utilizing ReLU or Softplus. These results indicate that bounded activation functions, such as Tanh and Sigmoid, are more effective than the unbounded activation functions, such as ReLU and Softplus, in improving the classification performance of the proposed method on three datasets.

V. CONCLUSION

This article proposes a novel tensorized high-order hypergraph convolutional network, called TH^{2}GCN, which generalizes the traditional GCN to achieve high-order discriminative information extraction from hypergraphs for HSI classification. Specifically, we use the hypergraph to effectively model complex spatial relationships between pixels in HSIs, and apply the adjacency tensor to better describe the intrinsic high-order interaction information of the hypergraph structure. In addition, the hypergraph convolution operation based on adjacency tensors can effectively extract more discriminative features encoded in hypergraphs to improve HSI classification performance while reducing computational complexity by using tensor decomposition. Furthermore, the HSI data are divided into multiple batches to train and test the proposed TH^{2}GCN in a minibatch fashion. Extensive experimental results on three HSI datasets show that the proposed TH^{2}GCN method is superior to other comparison methods. In future research, we will develop a distributed framework for efficiently training the proposed network with a particular focus on addressing ultralarge HSI classification tasks. Considering that the construction of hypergraphs and the CP decomposition of adjacency tensors are unsupervised processes, we will further integrate these components into networks and use attention mechanisms for dynamic hypergraph construction to boost the feature extraction.

References

[1] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[2] Y. Xu, Z. Wu, Z. Wei, H. Liu, and X. Xu, "A novel hyperspectral image anomaly detection method based on low rank representation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Milan, Italy, Jul. 2015, pp. 4444–4447.

[3] Y. Niu and B. Wang, "Hyperspectral anomaly detection based on low-rank representation and learned dictionary," *Remote Sens.*, vol. 8, no. 4, p. 289, Mar. 2016.

[4] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[5] A. Ghiyamat and H. Z. M. Shafri, "A review on hyperspectral remote sensing for homogeneous and heterogeneous forest biodiversity assessment," *Int. J. Remote Sens.*, vol. 31, no. 7, pp. 1837–1856, Apr. 2010.

[6] Y. Tan, L. Lu, L. Bruzzone, R. Guan, Z. Chang, and C. Yang, "Hyperspectral band selection for lithologic discrimination and geological mapping," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 471–486, 2020.

[7] K. Y. Ma and C.-I. Chang, "Kernel-based constrained energy minimization for hyperspectral mixed pixel classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5510723.

[8] W.-S. Hu, H.-C. Li, Y.-J. Deng, X. Sun, Q. Du, and A. Plaza, "Lightweight tensor attention-driven ConvLSTM neural network for hyperspectral image classification," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 734–745, Apr. 2021.

[9] J. Feng, G. Bai, D. Li, X. Zhang, R. Shang, and L. Jiao, "MR-selection: A meta-reinforcement learning approach for zero-shot hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5500320.

[10] J. Feng et al., "S^{4}DL: Shift-sensitive spatial–spectral disentangling learning for hyperspectral image unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 9, pp. 16894–16908, Sep. 2025.

[11] M. Ding, X.-L. Zhao, J.-H. Yang, Z. Zhou, and M. K. Ng, "Bilateral tensor low-rank representation for insufficient observed samples in multidimensional image clustering and recovery," *SIAM J. Imag. Sci.*, vol. 18, no. 1, pp. 20–59, Mar. 2025.

[12] L. Pan, H.-C. Li, W. Li, X.-D. Chen, G.-N. Wu, and Q. Du, "Discriminant analysis of hyperspectral imagery using fast kernel sparse and low-rank graph," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6085–6098, Nov. 2017.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[14] Z. Liu, B. Tang, X. He, Q. Qiu, and F. Liu, "Class-specific random forest with cross-correlation constraints for spectral–spatial hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 2, pp. 257–261, Feb. 2017.

[15] J. Peng and Q. Du, "Robust joint sparse representation based on maximum correntropy criterion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7152–7164, Dec. 2017.

[16] W. Li and Q. Du, "Joint within-class collaborative representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2200–2208, Jun. 2014.

[17] X. Wang, "Kronecker factorization-based multinomial logistic regression for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, Art. no. 5508005.

[18] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[19] 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.

[20] H. Sun, X. Zheng, and X. Lu, "A supervised segmentation network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 30, pp. 2810–2825, 2021.

[21] H. Yu, H. Yang, L. Gao, J. Hu, A. Plaza, and B. Zhang, "Hyperspectral image change detection based on gated spectral–spatial–temporal attention network with spectral similarity filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5511313.

[22] C. Chen, Y. Ma, and G. Ren, "Hyperspectral classification using deep belief networks based on conjugate gradient update and pixel-centric spectral block features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4060–4069, 2020.

[23] W. Wei, L. Tong, B. Guo, J. Zhou, and C. Xiao, "Few-shot hyperspectral image classification using relational generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5539016.

[24] W.-Y. Wang, H.-C. Li, Y.-J. Deng, L.-Y. Shao, X.-Q. Lu, and Q. Du, "Generative adversarial capsule network with ConvLSTM for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 523–527, Mar. 2021.

[25] M. E. Paoletti, S. Moreno-Álvarez, and J. M. Haut, "Multiple attention-guided capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5520420.

[26] S. Hao, W. Wang, and M. Salzmann, "Geometry-aware deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2448–2460, Mar. 2021.

[27] C. Yu, Y. Zhu, Y. Wang, E. Zhao, Q. Zhang, and X. Lu, "Concern with center-pixel labeling: Center-specific perception transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 5514614.

[28] H. Yang, H. Yu, K. Zheng, J. Hu, T. Tao, and Q. Zhang, "Hyperspectral image classification based on interactive transformer and CNN with multilevel feature fusion network," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023, Art. no. 5507905.

[29] H. Yu, Z. Ling, K. Zheng, L. Gao, J. Li, and J. Chanussot, "Unsupervised hyperspectral and multispectral image fusion with deep spectral–spatial collaborative constraint," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5534114.

[30] J. Feng, Z. Gao, R. Shang, X. Zhang, and L. Jiao, "Multi-complementary generative adversarial networks with contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5520018.

[31] C. Shi and C.-M. Pun, "Multiscale superpixel-based hyperspectral image classification using recurrent neural networks with stacked autoencoders," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 487–501, Feb. 2020.

[32] Y. Wang, L. Liu, J. Xiao, D. Yu, Y. Tao, and W. Zhang, "MambaHSI+: Multidirectional state propagation for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 4411414.

[33] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[34] J.-Y. Yang, H.-C. Li, W.-S. Hu, L. Pan, and Q. Du, "Adaptive cross-attention-driven spatial–spectral graph convolutional network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, Art. no. 6004705.

[35] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.

[36] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8246–8257, Dec. 2020.

[37] Y. Ding, Y. Guo, Y. Chong, S. Pan, and J. Feng, "Global consistent graph convolutional network for hyperspectral image classification," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–16, 2021.

[38] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3162–3177, May 2020.

[39] Y. Yang et al., "Semi-supervised multiscale dynamic graph convolution network for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 6806–6820, May 2024.

[40] W. Huang, D. Zhou, L. Sun, Q. Chen, and J. Yin, "Adaptive pixel-level and superpixel-level feature fusion transformer for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 16876–16889, 2024.

[41] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 1559–1572, 2022.

[42] W. Li, Q. Liu, S. Fan, H. Bai, and M. Xin, "Multistage superpixel-guided hyperspectral image classification with sparse graph attention networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5519718.

[43] Q. Yu, W. Wei, Z. Pan, J. He, S. Wang, and D. Hong, "GPF-Net: Graph-polarized fusion network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5519622.

[44] K. Qu, C. Wang, Z. Li, and F. Luo, "Spatial–spectral attention graph U-Nets for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5528317.

[45] T. Zhu, Q. Liu, and L. Zhang, "Hyperspectral image classification using multi-feature fusion residual hypergraph convolution network," in *Proc. 8th Int. Conf. Signal Image Process. (ICSIP)*, Wuxi, Wuxi, China, Jul. 2023, pp. 343–349.

[46] Z. Ma, Z. Jiang, and H. Zhang, "Hyperspectral image classification using feature fusion hypergraph convolution neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5517314.

[47] Y. Sun, A. Qin, Y. Bandoh, C. Gao, and Y. Hiwasaki, "Active learning for hyperspectral image classification via hypergraph neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 2576–2580.

[48] Y. Duan, F. Luo, M. Fu, Y. Niu, and X. Gong, "Classification via structure-preserved hypergraph convolution network for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5507113.

[49] Q. Xu, S. Xu, J. Liu, and L. Huang, "Dynamic hypergraph convolution and recursive gated convolution fusion network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[50] C. Yang, R. Wang, S. Yao, and T. F. Abdelzaher, "Hypergraph learning with line expansion," 2020, *arXiv:2005.04843*.

[51] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.

[52] X. Song, K. Wu, and L. Chai, "Brain network analysis of schizophrenia patients based on hypergraph signal processing," *IEEE Trans. Image Process.*, vol. 32, pp. 4964–4976, 2023.

[53] Y. Gao, Y. Feng, S. Ji, and R. Ji, "HGNN+: General hypergraph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3181–3199, Mar. 2023.

[54] M. Wang, Y. Zhang, X. Zhao, Y. Hu, and B. Yin, "Traffic origin-destination demand prediction via multichannel hypergraph convolutional networks," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 4, pp. 5496–5509, Aug. 2024.

[55] N. Wang, D. Liu, J. Zeng, L. Mu, and J. Li, "HGRec: Group recommendation with hypergraph convolutional networks," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 3, pp. 4214–4225, Jun. 2024.

[56] D. A. Nguyen, C. H. Nguyen, and H. Mamitsuka, "Central-smoothing hypergraph neural networks for predicting drug–drug interactions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 11620–11625, Aug. 2024.

[57] W. Hou, C. Lin, L. Yu, J. Qin, R. Yu, and L. Wang, "Hybrid graph convolutional network with online masked autoencoder for robust multimodal cancer survival prediction," *IEEE Trans. Med. Imag.*, vol. 42, no. 8, pp. 2462–2473, Aug. 2023.

[58] X. Hao, J. Li, Y. Guo, T. Jiang, and M. Yu, "Hypergraph neural network for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 2263–2275, 2021.

[59] N. Ma, Z. Wu, Y. Feng, C. Wang, and Y. Gao, "Multi-view time-series hypergraph neural network for action recognition," *IEEE Trans. Image Process.*, vol. 33, pp. 3301–3313, 2024.

[60] M. Ding, J.-H. Yang, X.-L. Zhao, J. Zhang, and M. K. Ng, "Nonconvex low-rank tensor representation for multi-view subspace clustering with insufficient observed samples," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 6, pp. 3583–3597, Jun. 2025.

[61] J.-H. Yang, C. Chen, H.-N. Dai, M. Ding, Z.-B. Wu, and Z. Zheng, "Robust corrupted data recovery and clustering via generalized transformed tensor low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8839–8853, Jul. 2024.

[62] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Tensor completion via complementary global, local, and nonlocal priors," *IEEE Trans. Image Process.*, vol. 31, pp. 984–999, 2022.

[63] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sep. 2009.

[64] C. Hua, G. Rabusseau, and J. Tang, "High-order pooling for graph neural networks with tensor decomposition," in *Proc. Neur. Inf. Process. Syst.*, vol. 35, 2022, pp. 6021–6033.

[65] F. Wang, K. Pena-Pena, W. Qian, and G. R. Arce, "T-HyperGNNs: Hypergraph neural networks via tensor representations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 5044–5058, Mar. 2025.

[66] M. Wang et al., "Tensorized hypergraph neural networks," *IEEE Trans. Image Process.*, vol. 32, pp. 4964–4976, 2024.

[67] C. Shi, D. Liao, and L. Wang, "Hybrid CNN-GCN network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 10530–10546, 2025.

[68] Y. Wang, Z. Xue, M. Jia, Z. Liu, and H. Su, "Hypergraph convolutional network with multiple hyperedges fusion for hyperspectral image classification under limited samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5526318.

[69] Q. Wang, J. Huang, S. Wang, Z. Zhang, T. Shen, and Y. Gu, "Community structure guided network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 4404115.

[70] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.