Neurocomputing 461 (2021) 63–76

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Example-feature graph convolutional networks for semi-supervised classification

Sichao Fu [a,b,c], Weifeng Liu [c,b,*], Kai Zhang [d], Yicong Zhou [e]

[a] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China
[b] State Key Laboratory of Integrated Services Networks, Xidian University, Xian 710000, China
[c] College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266580, China
[d] School of Petroleum Engineering, China University of Petroleum (East China), Qingdao 266580, China
[e] Faculty of Science and Technology, University of Macau, Macau 999078, China

ARTICLE INFO

Article history:
Received 7 October 2020
Revised 19 June 2021
Accepted 15 July 2021
Available online 22 July 2021
Communicated by Zidong Wang

Keywords:
Data representation learning
Convolutional neural networks
Graph convolutional networks
Example-feature graph

ABSTRACT

Graph convolutional networks (GCNs) successfully generalize convolutional neural networks to handle the graphs with high-order arbitrary structures. However, most existing GCNs variants consider only the local geometry of row vectors of high-dimensional data via example graph Laplacian, while neglecting the manifold structure information of column vectors. To address this problem, we propose the example-feature graph convolutional networks (EFGCNs) for semi-supervised classification. Particularly, we introduce the definition of the spectral example-feature graph (EFG) convolution that simultaneously utilizes the example graph Laplacian and feature graph Laplacian to better preserve the local geometry distributions of data. After optimizing the spectral EFG convolution with the first-order approximation, a single-layer EFGCNs is obtained. It is then further extended to build a multi-layer EFGCNs. Extensive experiments on remote sensing and citation networks datasets demonstrate the proposed EFGCNs show superior performance in semi-supervised classification compared with state-of-the-art methods.

1. Introduction

With the rapid development of Internet technologies and computer hardware, massive high-dimensional data (e.g. images, videos and audio) can be easily generated and acquired by mobile devices. These data contain a huge amount of useful and valued information. How to effectively extract such information from those massive data and explore its inherent laws and essential structures has become a hot issue in the fields of machine learning, data mining, pattern recognition and data representation learning [1]. In recent years, many data representation learning models have been proposed. Examples include the auto-encoder [2,3], canonical correlation analysis [4] and convolutional neural networks (CNNs) [5,6]. These models play a significant role in many practical applications, such as human activity recognition and detection [7], remote sensing image recognition and annotation [8,9] and video retrieval [10]. CNNs denote a data representation learning method that combines artificial neural networks and deep learning theory. Different from traditional methods that manually extract sample features for specific tasks, CNNs simulate the human visual system to automatically extract significant information via the hierarchical abstraction of data. CNNs and CNN-based variants have already achieved great success in the areas of computer vision [11] and natural language processing [12]. These CNN-based models can efficiently handle the Euclidean data [13] with regular spatial structures and explore effective data representations.

However, in real life, there exist substantial graph-structured data with irregular grid structures, such as remote sensing images. Thus, the traditional convolution operation of CNNs cannot handle such data effectively because of the irregular spatial structures and high-order characteristics of data [14]. To learn better sample features of the graph-structured data in non-Euclidean data domains, graph neural networks (GNNs) [15] have caught widespread attention and exhibited great advantages in the representation learning of remote sensing images. They are also known as the typical models of graph representation learning [16]. Existing GNN-based models can be divided into spatial-domain-based models and spectral-domain-based models.

* Corresponding author at: College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266580, China.
E-mail address: liuwf@upc.edu.cn (W. Liu).

https://doi.org/10.1016/j.neucom.2021.07.048
0925-2312/

Utilizing the spatial structure information of original graph-structured data, they construct or capture the neighborhood node features of each sample on graphs. The spatial-domain-based models directly apply a feature aggregation operation to each sample and its neighborhood nodes. In GraphSAGE [17], the aggregation function was introduced to the graph convolution operation to build an inductive graph structure learning model. It can directly generate the embedding representation of unseen nodes when the graph structures change. In learnable graph convolutional networks [18], using the k-largest neighborhood node selection method on each node of graphs, graph-structured data were transformed into regular grid-like data, and then applied with the standard CNN convolution. The gated graph neural networks [19] continuously updated the representation of each node with its neighborhood nodes information on graphs via introducing gated recurrent units [20] and back-propagation through time to train its model. Veličković [21] assigned different weights for the neighborhoods of each central node via the attention mechanism [22].

Transforming the convolution in the time domain into the point multiplication in the frequency domain according to spectral graph theory, the spectral-domain-based models learn a mapping rule for graph-structured data, and obtain the feature representation of each node that fuses its neighborhood node information. As a classical graph representation learning model in the spectral domain, graph convolutional networks (GCNs) [23] extended the classical CNNs to non-Euclidean data utilizing the spectral filter with the first-order polynomial to acquire the direct neighborhood information of each node. To obtain richer sample structure relationships, Li et al. [24] proposed to learn the optimal residual graph Laplacian using the feature transform and distance metric of nodes. Yadati et al. [25] and Feng et al. [26] generalized GCNs into the hypergraph domain utilizing the hypergraph Laplacian matrix to express complex relationships between samples. Compared with the Laplacian matrix with the one-order derivative, HesGCNs [27] and GpLCNs [28] can utilize more abundant structure information because of the existence of Hessian's and p-Laplacian's high-order derivatives.

The above-mentioned spectral graph convolutional networks consider only the example graph-based structure relationships that represent the local geometry distributions of row vectors of high-dimensional data. However, they ignore the structure information of the feature graph that carries the geometric structures of column vectors of high-dimensional data. In other words, due to the shortcomings of structure information of the example graph, GCNs fail to learn more significant data features via the convolution operation of original input feature information.

To address this issue, in this paper, we propose the example-feature graph convolutional networks (EFGCNs) to simultaneously consider feature-graph and example-graph space structure information. In particular, we generalize the spectral example graph convolution into the spectral EFG graph convolution. A single-layer convolution layer rule of EFGCNs is then designed to optimize the first-order approximation of the spectral EFG graph convolution. A multi-layer EFGCNs model is further built to automatically extract more efficient data features. In comparison to GCNs, our proposed EFGCNs can better exploit the local geometry of feature distributions and data distributions during the training process. To demonstrate the classification performance of our proposed EFGCNs, we conduct extensive experiments on the RSSCN7 and SAT-6 datasets for remote sensing classification, on the Citesser, Cora and NELL datasets for citation networks classification.

In summary, the contributions of this paper can be written as follows:

(1) We develop the definition of the spectral EFG convolution from the spectral convolution on an example graph. Compared with the example graph or feature graph, it is able to simultaneously learn the local geometry of row and column vectors of high-dimensional data via the example-feature graph.
(2) We propose an efficient convolution layer rule of spectral EFG convolution with the first-order approximation. This forms the single-layer of our proposed example-feature graph convolutional networks (EFGCNs).
(3) Based on the single-layer convolution rule, we further build a multi-layer EFGCNs.
(4) To evaluate the proposed EFGCNs with the application of semi-supervised classification, we conduct extensive experiments on five databases for remote sensing classification and citation networks classification. Experiment results demonstrate the superior performance of EFGCNs in comparison with existing semi-supervised learning models.

The remainder of this paper is arranged as follows: We briefly summarize several related works in Section 2. Sections 3–5 present the theoretical analysis of our proposed spectral EFG convolution, the single-layer and multi-layer EFGCNs. The experimental results on five datasets are discussed and analyzed in Section 6. Finally, we conclude this paper in Section 7.

2. Related Works

Before introducing our proposed algorithm, this section briefly reviews the related works, such as graph convolutional networks and graph principal component analysis.

2.1. Graph Convolutional Networks

In principle, the convolution is defined as the linear operator diagonalized in the Fourier basis [29]. Bruna et al. [30] extended the classical CNNs to the irregular-structure data domains by using the eigenvectors of the graph Laplacian operator to represent the corresponding Fourier basis. To construct the spectral filter with spatial localization and small computational complexity, Henaff et al. [31] gave the definition of spectral convolution on the single graph (example graph). It is defined as the multiplication of frequency domain for a signal X and a non-parametric spectral filter $g_\theta(L_1)$ in the Fourier domain, i.e.

$$g_\theta(L_1)\star X = g_\theta\left(U\Lambda U^T\right)X = Ug_\theta(\Lambda)U^TX \tag{1}$$

where L_1 denotes the normalized graph Laplacian, i.e. $L_1 = I_N - D_D^{-\frac{1}{2}}A_D D_D^{-\frac{1}{2}} (L_1 = U\Lambda U^T)$. $\star$ denotes the convolution operation. U and Λ are the eigenvectors and eigenvalues of L_1 separately. In addition, $g_\theta(\Lambda) = diag(\theta)$. I_N denotes the identity matrix. A_D denotes the node's adjacency relationship matrix. D_D is the degree matrix about A_D. L_1 expresses the local geometry structures of example graph or data manifold. It is computed according to the row vectors of input data.

To further reduce the learning complexity of spectral filter, Defferrard [32] introduced the Chebyshev polynomials with K orders to achieve the polynomial parametrization of localized spectral filter $g_\theta(\Lambda)$, i.e. $g_\theta(\Lambda) = \sum_{K=0}^{K}\theta_K T_K(\Lambda)$. Kipf et al. [23] considered only direct neighborhoods of each node on the simple graph by using the spectral filter with the first order Chebyshev polynomials ($K = 1$). Finally, a linear layer-wise model of GCNs is proposed, i.e.

$$g_\theta(L_1)\star X = \overrightarrow{D_D}^{-\frac{1}{2}}(A_D + I_N)\overrightarrow{D_D}^{-\frac{1}{2}}X\theta \tag{2}$$

where $(\overrightarrow{D_D})_{ii} = \sum_j (A_D + I_N)_{ij}$. That is Eq. (2) denotes either the output signals after removing signal noise or the extracted sample

features after fusing the structure information $(A_D + I_N)$ and feature information X of original data.

2.2. Graph Principal Component Analysis

Principal component analysis (PCA) [33] is a linear data representation learning algorithm, and aims to find optimal Q-dimensional (low-dimensional) linear subspace. Thus, the information representation of high-dimensional data can be concentrated in a small number of data dimensions. PCA can achieve data dimensionality reduction. It can be expressed as the following optimization problem:

$$\min_{U,V} \|X^T - UV^T\|_F^2 \quad s.t. V^T V = I_N \tag{3}$$

where $X = (x_1, x_2, \cdots x_N) \in R^{N\times M}$($N$ samples with M-dimensional features) denotes original high-dimensional data, $X^T \in R^{M\times N}$ denotes the transposed matrix of X. $V^T \in R^{Q\times N}$ (N samples with Q-dimensional features) denotes output sample features after projection. $U \in R^{M\times Q}$ is the projection matrix. However, PCA is efficient for high-dimensional data with a linear geometry structure. Jiang et al. [34] extended PCA to non-linear data domain by introducing Laplacian Eigenmap (LE) [35] to preserve the local geometry distributions of the data manifold (local geometry of row vectors of high-dimensional data). The objective function can be written as:

$$\min_{U,V} \|X^T - UV^T\|_F^2 + \gamma Tr\left(V^T L_D V\right) \quad s.t. V^T V = I_N \tag{4}$$

where γ denotes the balance parameter of its objective function. $Tr()$ represents the matrix's trace. Graph Laplacian PCA (gLPCA) [34] used the non-normalized graph Laplacian matrix, i.e. $L_D = D_D - A_D$ with $(D_D)_{ii} = \sum_j (A_D)_{ij}$. $A_D \in R^{N\times N}$ denotes the adjacency relationship matrix between different samples and A_D can be acquired by the k-nearest neighboring method with Euclidean distance to data X.

To consider the local geometry distributions of the feature manifold (local geometry of column vectors of high-dimensional data) simultaneously on gLPCA, He et al. [36] and Liu et al. [37] proposed dual graph Laplacian PCA, i.e.

$$\min_{U,V} \|X^T - UV^T\|_F^2 + \gamma Tr\left(V^T L_D V\right) + \beta Tr\left(U^T L_F U\right) \quad s.t. V^T V = I_N \tag{5}$$

where γ and β are all regularization parameters to balance the reconstruction error of the first term, local geometry distributions of data manifold, and feature manifold in the objective function of dual graph Laplacian PCA. In addition, the computing method of L_F is similar to that of L_D, i.e. $L_F = D_F - B_F$ with $(D_F)_{ii} = \sum_j (B_F)_{ij}$. $B_F \in R^{M\times M}$ can be computed by using the k-nearest neighboring method with Euclidean distance to $X^T = \left((x_1)^T, (x_2)^T, \cdots (x_N)^T\right) \in R^{M\times N}$. L_D and L_F denote the Laplacian matrix about the adjacency matrix A_D and B_F respectively.

Wang et al. [38] generalized the dual graph Laplacian PCA from simple graph to complex graphs (hypergraph), and aimed to better utilize local grouping information between samples, i.e.

$$\min_{U,V} \|X^T - UV^T\|_F^2 + \gamma Tr\left(V^T L_{DH} V\right) + \beta Tr\left(U^T L_{FH} U\right) \quad s.t. V^T V = I_N \tag{6}$$

where L_{DH} and L_{FH} are non-normalized hypergraph Laplacian matrices. The detailed computation process can be found in [38,9]. Recently, many researchers proposed the dual graph regularized data representation learning models by utilizing the geometry structure of feature distributions and sample distributions. For example, Yin et al. [39] proposed the dual graph regularized latent low-rank representation for subspace clustering. Inspired with the ensemble manifold learning, Li et al. [40] introduced the nonnegative matrix trifactorization based relational multi-manifold co-clustering algorithm. This method aimed to better utilize the intrinsic manifold structures between the samples and features. Tong et al. [41] proposed the dual graph regularized nonnegative matrix factorization for hyperspectral unmixing. Wang et al. [42] proposed the dual hypergraph regularized supervised non-negative matrix factorization for the genes and tumor classification tasks by constructing the feature hypergraph and data hypergraph to learn richer data structure information.

3. Example-Feature Graph

Existing GCNs utilize only the example graph to capture the local structure relationships of row vectors of high-dimensional data. Due to the complexity of data structures, it may lead to the shortcoming of the acquired structure information. To address this issue, we propose an EFG to simultaneously consider the manifold structure of example graph and feature graph. Introducing the spectral convolution, our proposed EFG fuses the structure information of the sample graph and feature graph into one unified graph, called the example-feature graph that simultaneously expresses the geometry structure of row and column vectors of high-dimensional data. Compared with the sample graph or feature graph, EFG can fit the data exactly. Next, we describe how to generate the EFG by the fusion of input example graph and feature graph.

We first give the definition of EFG. It is the spectral convolution on example graph $L_1 = U\Lambda U^T$ and feature graph $L_2 = V\Lambda^1 V^T$, and can be expressed as the product of input signal X, a spectral filter $g_\theta(L_1)$ of example graph and a spectral filter $g_\theta(L_2)$ of feature graph in the frequency domain (Fourier domain).

GCNs can be regarded as the process of removing noise from input signals via $g_\theta(L_1)$. However, GCNs cannot remove multiple types of signal noise. Thus, EFG can learn momentous signal features via $g_\theta(L_1)$ and $g_\theta(L_2)$.

$$\begin{aligned} g_\theta(L_1)\bigstar\left(g_{\theta^1}(L_2)\bigstar X\right) &= g_\theta\left(U\Lambda U^T\right)\left(g_{\theta^1}\left(V\Lambda^1 V^T\right)X\right) \\ &= U g_\theta(\Lambda) U^T V g_{\theta^1}\left(\Lambda^1\right) V^T X \end{aligned} \tag{7}$$

where L_2 expresses the normalized feature graph Laplacian, i.e. $L_2 = I_N - D_F^{-\frac{1}{2}} B_F D_F^{-\frac{1}{2}}$. Λ^1 and V denote the eigenvalues and eigenvectors of L_2. θ^1 and θ are filter parameters of spectral filters $g_{\theta^1}(L_2)$ and $g_\theta(L_1)$. EFG is obtained fusing the space structure information of example graph and feature graph via the convolution operation in Eq. (7).

However, Eq. (7) needs to perform eigendecomposition for the feature graph Laplacian matrix and example graph Laplacian matrix on every forward propagation. Thus, it leads to high computation complexity. To overcome this issue, we use K-order Chebyshev polynomials [32] to obtain a K-localized filters for $g_\theta(\Lambda) = \sum_{K=0}^{K} \theta_K T_K(\Lambda)$ and $g_{\theta^1}\left(\Lambda^1\right) = \sum_{K1=0}^{K1} \theta_{K1}^1 T_{K1}^1\left(\Lambda^1\right)$. Finally, we can obtain a K-order localized spectral EFG convolution. Let $Z = g_\theta(L_1)\bigstar\left(g_{\theta^1}(L_2)\bigstar X\right)$ and Eq. (7) can be simplified into the following form:

$$Z = \sum_{K=0}^{K} \theta_K T_K\left(\overrightarrow{L_1}\right) \sum_{K1=0}^{K1} \theta_{K1}^1 T_{K1}^1\left(\overrightarrow{L_2}\right) X \tag{8}$$

where the computation complexity of each spectral filter range from $O\left(N^2\right)$ or $O\left(M^2\right)$ to $O(K|\varepsilon|)$ or $O(K_1|\varepsilon|)$. $\overrightarrow{L_1} = \frac{2}{\lambda_{max}} L_1 - I_N, \overrightarrow{L_2} = \frac{2}{\lambda_{max}^1} L_2 - I_N$ with the largest eigenvalue λ_{max} of L_1 and λ_{max}^1 of L_2. In addition, the Chebyshev polynomials $T_K(X)$

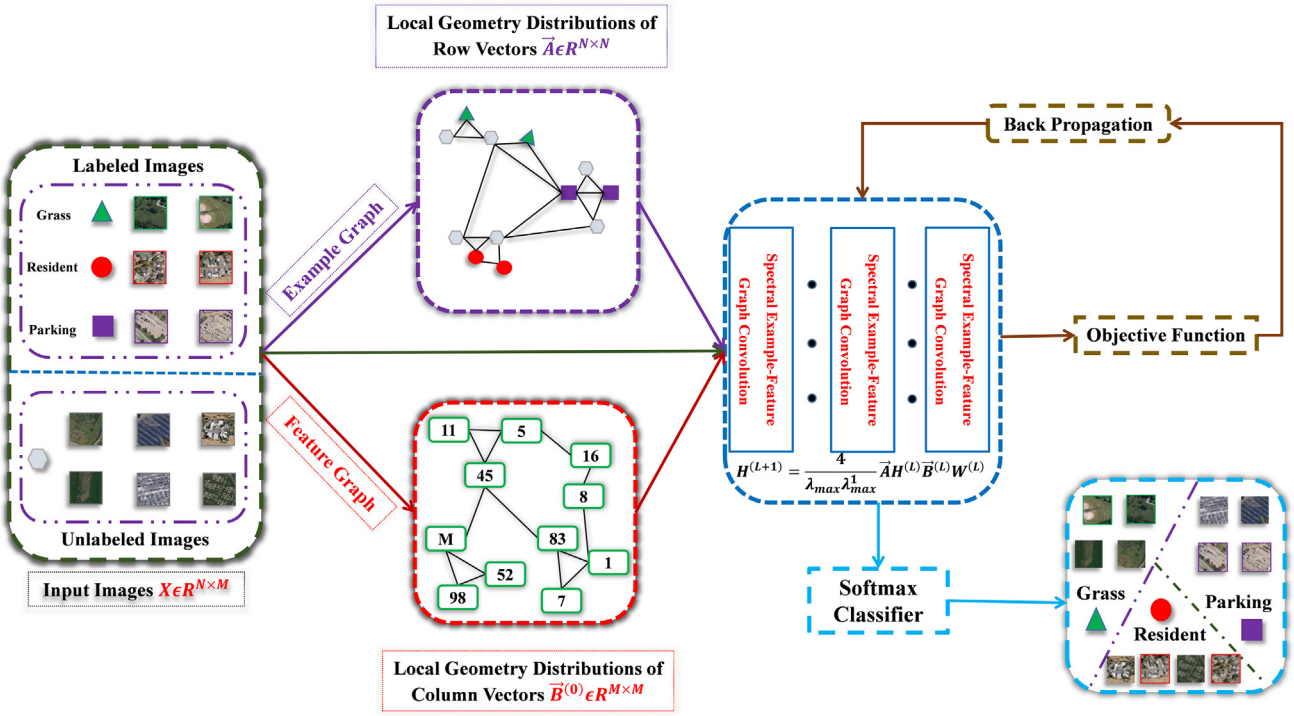

Fig. 1. The framework of the multi-layer EFGCNs.

Fig. 2. Sample images of the RSSCN7 dataset. From left to right, each column represents a class, such as grass, field, parking, river lake, forest, resident and industry.

are computed by the following method: $T_0(X) = 1, T_1(X) = X$ with $T_K(X) = 2XT_{K-1}(X) - T_{K-2}(X)$. Finally, we can obtain a complementary EFG via a K-order localized spectral EFG convolution in Eq. (8). Next, we will introduce the proposed example-feature graph convolutional networks (EFGCNs).

4. Single-Layer EFGCNs

Using the spectral EFG convolution layer rule in Eq. (8), one can be built a deep-layer spectral CNNs model. However, its computation complexity is quite high because this model contains a large number of filter coefficients and the N-th power of matrices. Moreover, the model in Eq. (8) will lead to the overfitting problem for deep networks [22]. (When the K-order number of the model increases, the structure information between data will be dense. What's more, the classification accuracy of the model may not be effectively improved. It has been demonstrated in GCNs [23].) To obtain an optimized single-layer EFGCNs, we develop an optimization of the spectral EFG convolution layer rule, and then limit the orders K and $K1$ of Chebyshev polynomials. In this paper, we set $K = 1$ and $K1 = 1$ (The direct neighborhood structure information of each central node) because the first-order approximation of spectral EFG convolution can better preserve the intrinsic manifold structure of example graph and feature graph. Thus, Eq. (8) can be simplified into the following form:

$$Z = \left(\theta_0 + \theta_1\left(\tfrac{2}{\lambda_{max}}L_1 - I_N\right)\right)\left(\theta_0^1 + \theta_1^1\left(\tfrac{2}{\lambda_{max}^1}L_2 - I_N\right)\right)X \tag{9}$$

This definition has four filter parameters $\theta_0, \theta_1, \theta_0^1$ and θ_1^1 for each node of the example graph and feature graph. To better understanding the simplification processes, we make a further derivation for Eq. (9), i.e.

$$Z = \left(\theta_0 - \theta_1 + \frac{2}{\lambda_{max}}\theta_1 L_1\right)\left(\theta_0^1 - \theta_1^1 + \frac{2}{\lambda_{max}^1}\theta_1^1 L_2\right)X \tag{10}$$

For Eq. (10), it exists many model parameters. If we use Eq. (10) to build our single-layer EFGCNs, it will cause the overfitting problem of the model. Thus, how to reduce the overfitting problem and the computation complexity of the single-layer EFGCNs using a single filter parameter θ is very important. Eq. (10) can be further simplified via deduction from Eq. (11) to Eq. (14). To solve this problem, following, we detailed analyze the theoretical analysis process from Eq. (10) to Eq. (14).

Let $\theta_0 - \theta_1 = \theta_2$ ($\theta_2 \neq 0$) and $\theta_0^1 - \theta_1^1 = \theta_3$ ($\theta_3 \neq 0$). Thus, Eq. (10) can be further simplified:

$$\begin{aligned} Z &= \left(\theta_2 + \tfrac{2}{\lambda_{max}}\theta_1\left(I_N - D_D^{-\frac{1}{2}}A_D D_D^{-\frac{1}{2}}\right)\right)\left(\theta_3 + \tfrac{2}{\lambda_{max}^1}\theta_1^1\left(I_N - D_F^{-\frac{1}{2}}B_F D_F^{-\frac{1}{2}}\right)\right)X \\ &= \left(\tfrac{2}{\lambda_{max}}\left(\tfrac{\lambda_{max}}{2}\theta_2 + \theta_1\right) - \tfrac{2}{\lambda_{max}}\theta_1 D_D^{-\frac{1}{2}}A_D D_D^{-\frac{1}{2}}\right)\left(\tfrac{2}{\lambda_{max}^1}\left(\tfrac{\lambda_{max}^1}{2}\theta_3 + \theta_1^1\right) - \tfrac{2}{\lambda_{max}^1}\theta_1^1 D_F^{-\frac{1}{2}}B_F D_F^{-\frac{1}{2}}\right)X \end{aligned} \tag{11}$$

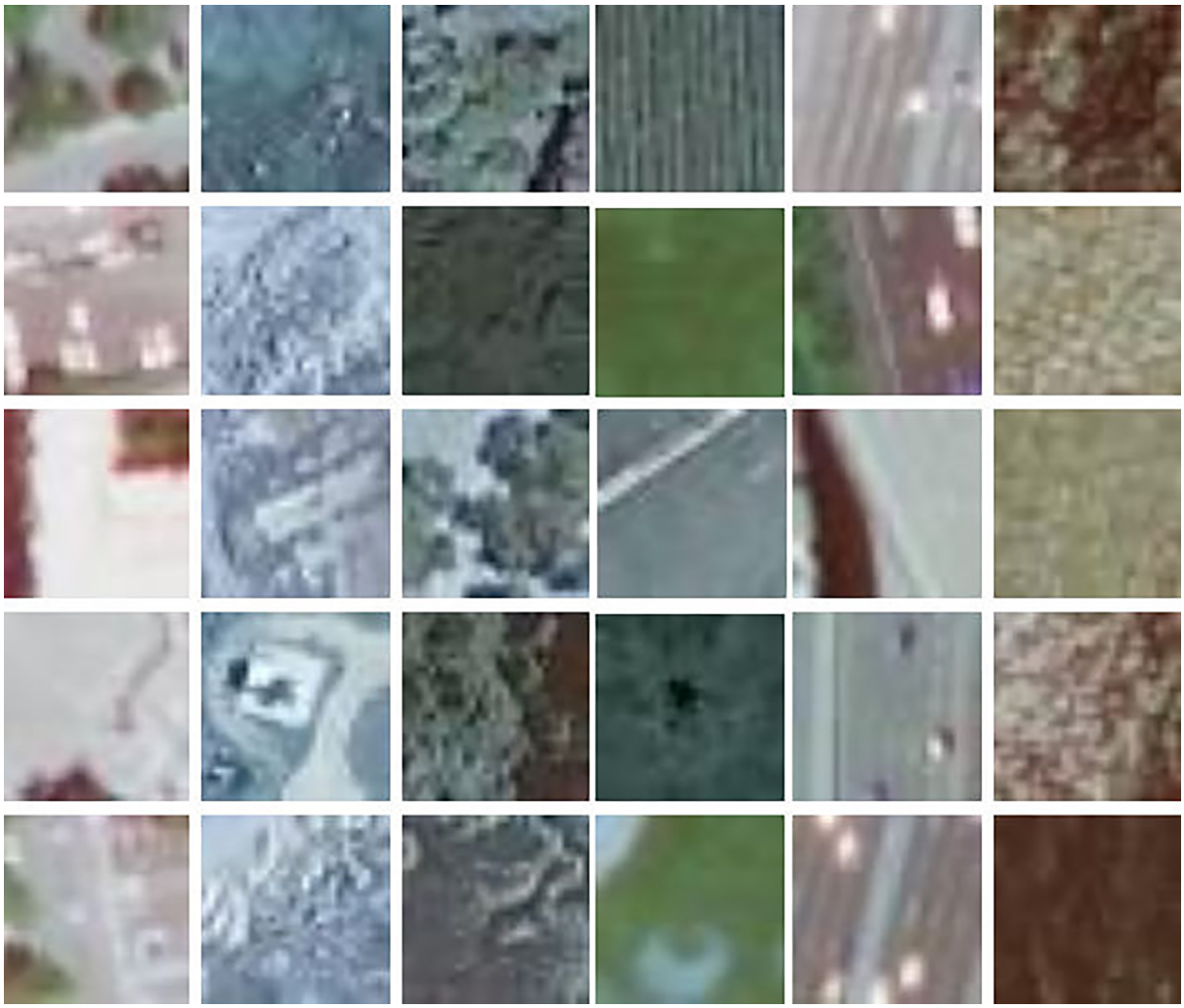

Fig. 3. Sample examples of the SAT-6 dataset. From left to right, each column represents a class, such as buildings, water bodies, trees, barren land, grassland, and roads.

Let $\frac{\lambda_{max}}{2}\theta_2 + \theta_1 = \theta_4$ and $\frac{\lambda^1_{max}}{2}\theta_3 + \theta^1_1 = \theta_5$.

$$Z = \frac{2}{\lambda_{max}}\left(\theta_4 - \theta_1 D_D^{-\frac{1}{2}} A_D D_D^{-\frac{1}{2}}\right)\frac{2}{\lambda^1_{max}}\left(\theta_5 - \theta^1_1 D_F^{-\frac{1}{2}} B_F D_F^{-\frac{1}{2}}\right)X \tag{12}$$

Let $\theta_4 = -\theta_1 = \theta_6$ and $\theta_5 = -\theta^1_1 = \theta_7$.

$$Z = \frac{2}{\lambda_{max}}\frac{2}{\lambda^1_{max}}\theta_6\left(I_N + D_D^{-\frac{1}{2}} A_D D_D^{-\frac{1}{2}}\right)\theta_7\left(I_N + D_F^{-\frac{1}{2}} B_F D_F^{-\frac{1}{2}}\right)X \tag{13}$$

Let $\theta = \theta_6\theta_7$. Finally, Eq. (11) can be simplified to the following form:

$$\begin{aligned} Z &= \tfrac{2}{\lambda_{max}}\tfrac{2}{\lambda^1_{max}}\left(I_N + D_D^{-\frac{1}{2}} A_D D_D^{-\frac{1}{2}}\right)X\left(I_N + D_F^{-\frac{1}{2}} B_F D_F^{-\frac{1}{2}}\right)\theta \\ &= \tfrac{4}{\lambda_{max}\lambda^1_{max}}\vec{A}X\vec{B}\theta \end{aligned} \tag{14}$$

$\vec{A} = I_N + D_D^{-\frac{1}{2}} A_D D_D^{-\frac{1}{2}}$ (example graph based structure information matrix) and $\vec{B} = I_N + D_F^{-\frac{1}{2}} B_F D_F^{-\frac{1}{2}}$ (structure information matrix based on feature graph). $\theta \in R^{M\times G}$ is the filter parameter learned from EFG. When a signal $X \in R^{N\times M}$ (N samples with M dimensional features) is regarded as the input, we can obtain the samples' feature matrix $Z \in R^{N\times G}$ after spectral EFG convolution.

Using the definition in Eq. (14) of the spectral EFG convolution with one-order Chebyshev polynomial, we can obtain the single-layer EFGCNs $f\left(X, \vec{A}, \vec{B}, W\right)$ in the following form:

$$H^{(L+1)} = \sigma\left(\frac{4}{\lambda_{max}\lambda^1_{max}}\vec{A}H^{(L)}\vec{B}^{(L)}W^{(L)}\right) \tag{15}$$

where, $\vec{B}^{(L)} = I_N + D_F^{-\frac{1}{2}} B_F^{(L)} D_F^{-\frac{1}{2}}$. $H^{(L+1)}$ denotes the extracted sample features of each layer, $H^{(0)} = X$. Due to the change of the dimension for column vectors of $H^{(L+1)}$, the adjacency matrix $B_F^{(L)}$ of each layer will be recalculated. $W^{(L)}$ expresses the weight matrix trained during the training process. σ denotes the nonlinear activation function.

5. Multi-Layer EFGCNs

Stacking Eq. (15), we can further build the multi-layer EFGCNs for semi-supervised classification. Fig.1 describes the framework of the multi-layer example-feature graph convolutional networks (EFGCNs). The multi-layer EFGCNs can be written in Eq. (16).

$$H^{(L+1)} = classifier\left(\frac{4}{\lambda_{max}\lambda^1_{max}}\vec{A}\left(\sigma\left(\frac{4}{\lambda_{max}\lambda^1_{max}}\vec{A}\left(\cdots\sigma\left(\frac{4}{\lambda_{max}\lambda^1_{max}}\vec{A}X\vec{B}^{(0)}W^{(0)}\right)\cdots\right)\vec{B}^{(L-1)}W^{(L-1)}\right)\right)\vec{B}^{(L)}W^{(L)}\right) \tag{16}$$

For multi-layer EFGCNs, the initial graph-Laplacian-based example-graph structure information $\vec{A}$ and feature-graph structure information $\vec{B}^{(0)}$ are constructed from original data X and X^T, respectively. The detailed computation processes of A_D and B_F can be found in Section 2.2. Due to the change of output feature column vector dimensions on each convolution layer, the structure information of the feature graph (except for the first layer) should be recomputed according to output features of the last convolution layer. Algorithm 1 briefly illustrates the implementation processes of multi-layer EFGCNs for semi-supervised classification.

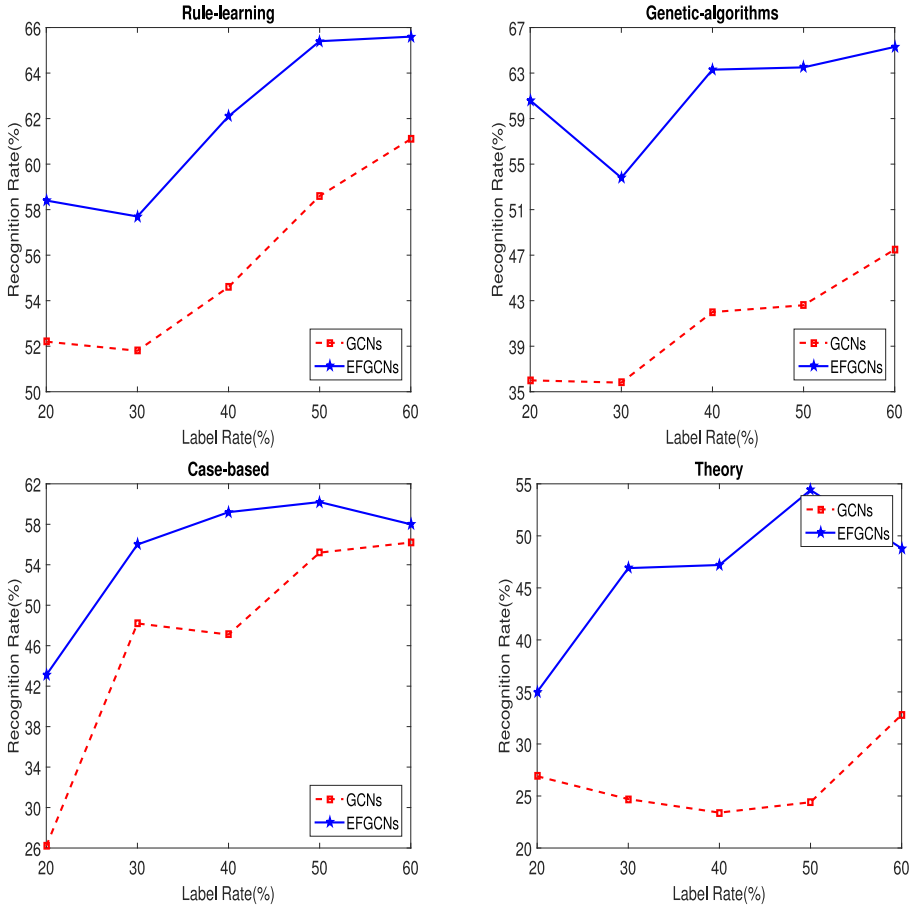

Fig. 4. Mean recognition rates of each class on the Cora dataset. Each subfigure corresponds to a single class.

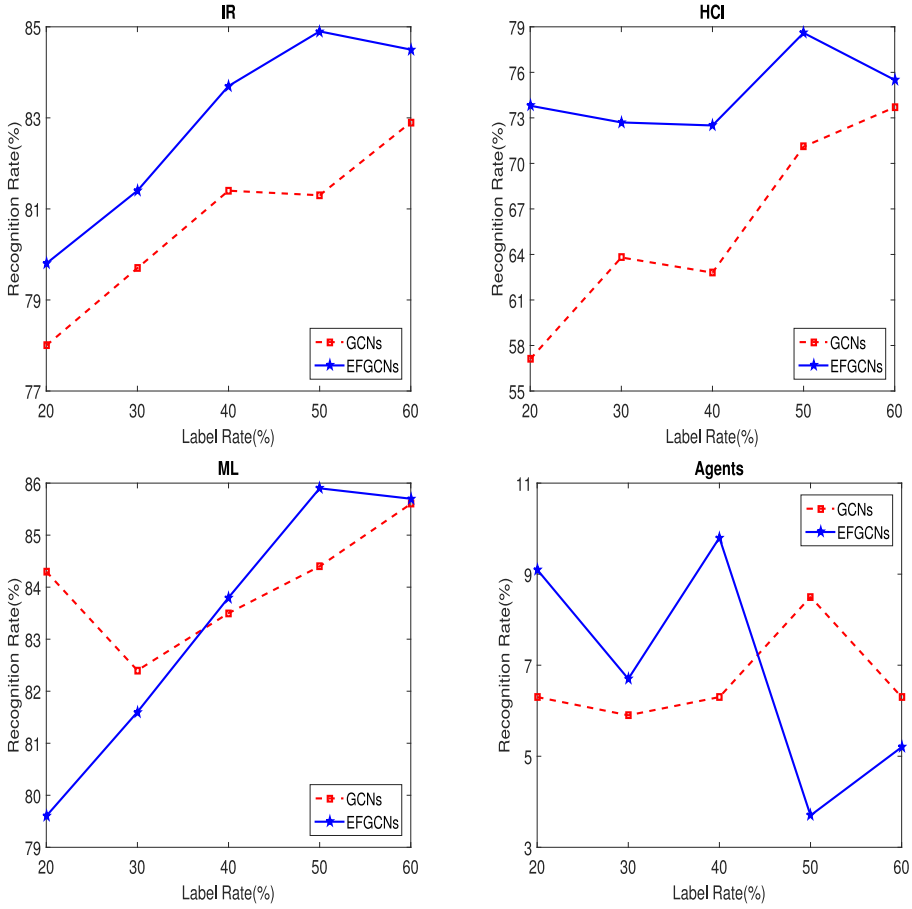

Fig. 5. Mean recognition rates of each class on the Citeseer dataset. Each subfigure corresponds to a single class.

Here, we build a two-layer EFGCNs based on Eq. (15) for our experiments to evaluate the classification performance of EFGCNs, i.e.

$$H^{(2)} = \frac{4}{\lambda_{max}\lambda^{1}_{max}}\overrightarrow{A}\left(RELU\left(\frac{4}{\lambda_{max}\lambda^{1}_{max}}\overrightarrow{A}X\overrightarrow{B}^{(0)}W^{(0)}\right)\right)\overrightarrow{B}^{(1)}W^{(1)} \quad (17)$$

where $RELU$ is the Rectified Linear Unit, i.e. $f(x) = \max(0, x)$. $W^{(0)} \in R^{M\times G1}$ is the filter coefficient matrix of the first layer. The example-graph structure information $\overrightarrow{A}$ and feature-graph structure information $\overrightarrow{B}^{(0)}$ of the first layer are computed from original data X or X^T. After the spectral convolution of the first layer, we can obtain the sample features $H^{(1)} \in R^{N\times G1}$, i.e.

$$H^{(1)} = RELU\left(\frac{4}{\lambda_{max}\lambda^{1}_{max}}\overrightarrow{A}X\overrightarrow{B}^{(0)}W^{(0)}\right) \quad (18)$$

Then, we recalculate $\overrightarrow{B}^{(1)} \in R^{G1\times G1}$ according to $H^{(1)}$. In addition, the output sample features of the first layer are regarded as the input of the second layer. Finally, the two-layer EFGCNs obtain the final data features $H^{(2)} \in R^{N\times G2}$ with $W^{(1)} \in R^{G1\times G2}$ (the dimension of $G2$ is equal to the number of categories for different datasets), i.e.

$$H^{(2)} = \frac{4}{\lambda_{max}\lambda^{1}_{max}}\overrightarrow{A}H^{(1)}\overrightarrow{B}^{(1)}W^{(1)} \quad (19)$$

Table 1
Descriptions of experimental datasets.

Dataset	Nodes	Classes	Dimensions
RSSCN7	2800	7	4096
SAT-6	405000	6	784
Citeseer	3327	6	3703
Cora	2708	7	1433
NELL	65755	210	5414

Algorithm 1: Multi-Layer Example-Feature Graph Convolutional Networks (EFGCNs)

Input: Data X
Parameter: Dropout rate, learning rate, hidden units, L2 regularization.
Output: Mean classification accuracy
1: Construct adjacency matrix A_D and B_F of the example graph L_1 and feature graph L_2 (see Section 2.2)
2: Compute the initial structure information $\overrightarrow{A}$ and $\overrightarrow{B}^{(0)}$
3: Initialize the hyperparameters
4: **for** $j = 0 \rightarrow T - 1$ (T denote the iteration numbers)
5: $H^{(1)} = RELU\left(\frac{4}{\lambda_{max}\lambda^{1}_{max}}\overrightarrow{A}X\overrightarrow{B}^{(0)}W^{(0)}\right)$.
6: Recalculate adjacency matrix B_F, update structure information $\overrightarrow{B}^{(1)}$

(*continued on next page*)

a (*continued*)

Algorithm 1: Multi-Layer Example-Feature Graph Convolutional Networks (EFGCNs)

7: $H^{(2)} = RELU\left(\frac{4}{\lambda_{max}\lambda_{max}^{1}}\overrightarrow{A}H^{(1)}\overrightarrow{B}^{(1)}W^{(1)}\right)$.
8: Recalculate adjacency matrix B_F, update structure information $\overrightarrow{B}^{(2)}$
9: $\cdots$
10: Recalculate adjacency matrix B_F, update structure information $\overrightarrow{B}^{(L)}$
11: $H^{(L+1)} = \frac{4}{\lambda_{max}\lambda_{max}^{1}}\overrightarrow{A}H^{(L)}\overrightarrow{B}^{(L)}W^{(L)}$.
12: **until convergence**
(lines 6 to 11 denote the multi-layer convolution operations of data X)
13: Obtain the final structure information $\overrightarrow{B}^{(1)}, \cdots, \overrightarrow{B}^{(L)}$ and optimal weight information $W^{(0)}, \cdots, W^{(L)}$.
14: Send the extracted features H(L+1) to Softmax classifier.
15: Return the mean classification accuracy of data.

After two convolution layers, we feed the extracted data features $H^{(2)}$ into the classifier and obtain the classification accuracy. In this paper, we use the Softmax classifier [43]. In the back propagation of EFGCNs, we use the cross entropy function [44] to evaluate our proposed model. If the value of the objective function cannot reach a specific threshold, we will repeat the training processes (Eq. (17)) until $\overrightarrow{B}^{(1)}$, $W^{(0)}$ and $W^{(1)}$ reach the optimal. We use the gradient descent method [45] to update the weight matrix of each layer. Compared with two-layer GCNs [23] that stack Eq. (2), our EFGCNs simultaneously consider the local distributions of row vectors and column vectors of high-dimensional data in each convolution layer to form the complementary EFG structure information.

6. Experiments

In this section, we test our proposed EFGCNs and several existing semi-supervised learning models including HyperGCNs [25], GAT [21], GCNs [23], Chebyshev (K = 2) [32], Chebyshev (K = 3) [32], Semi-supervised Embedding [46], Manifold Regularization [47], HesGCNs [27], GpLCNs [28] and Multi-layer Perception [48] using the RSSCN7 [49] and SAT-6 [50] datasets for remote sensing scene classification, and the Citeseer [51], Cora [52] and NELL [53] datasets for citation networks classification.

RSSCN7 dataset [49] is composed of 2800 images collected from seven categories, including grass, field, parking, river lake, forest, resident and industry. Each class contains 400 images. We resize the original RSSCN7 images from 400*400 to 64*64 pixels, and then extract their wavelet transform features by the Coiflets orthogonal wavelet transform [54,55]. Several images in the RSSCN7 are exhibited in Fig. 2.

SAT-6 dataset [50] consists of totally 405,000 RGB images with 28*28 pixels. All images are divided into six classes, such as buildings, water bodies, trees, barren land, grassland, and roads. In addition, we utilize the edge feature method to perform the preprocessing of experimental images. Fig. 3 exhibits some images of the SAT-6 dataset.

Citeseer [51] and Cora [52] are citation networks datasets. Citeseer contains a total of 3327 documents collected from HCI (Human Computer Interaction), AI (Artificial Intelligence), ML (Machine Language), DB (Database), IR (Information Retrieval) and Agents. Cora contains seven classes, such as case-based, neural networks, probabilistic-methods, genetic-algorithms, reinforcement-learning, rule-learning and theory. The dataset is composed of 2708 samples. Each document has many words. The NELL dataset [53] is composed of a total of 65755 samples collected from 210 classes. The dimension of each sample is 5414. It exists 266144 link relationships between samples. Table 1 briefly describes these five public datasets.

6.1. Experiment Settings

For RSSCN7, SAT-6, Citeseer, Cora and NELL, 1000 samples are selected to form the testing set, 500 samples for the validation set and the rest of the samples are employed for the training set. (Our experiments use all samples of the Citeseer, Cora and RSSCN7, and select 5000 data of the NELL and 3000 samples of the SAT-6 to evaluate our proposed EFGCNs.) In our experiments, we randomly

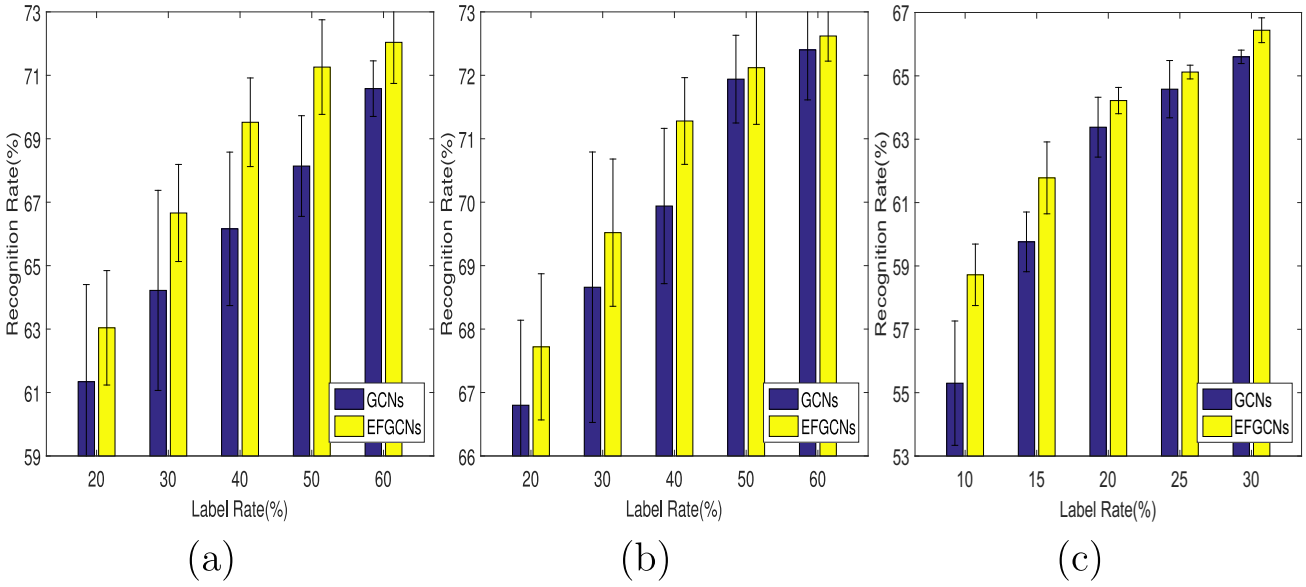

Fig. 6. Mean recognition rates of all categories on the (a) Cora, (b) Citeseer and (c) NELL datasets.

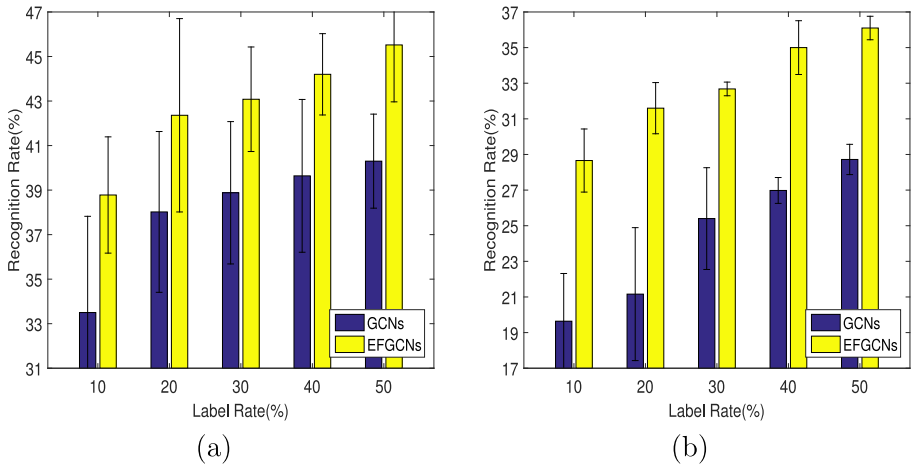

Fig. 7. Mean recognition rates of all categories on the (a) SAT-6 and (b) RSSCN7 datasets.

select 10%, 20%, 30%, 40% and 50% images from the training sets of RSSCN7 and SAT-6 as labeled images, the rest of the images are used for unlabeled images. For Citeseer and Cora, we randomly assign a specific label rate (20%, 30%, 40%, 50% and 60%) to their training samples. For NELL, a specific percentage of data on training set as labeled data, such as 10%, 15%, 20%, 25% and 30%.

During the training process of EFGCNs, we utilize the Adam optimizer [56] with the learning rate of 0.01 to optimize hyperparameters. This aims to reduce the loss value of the objective function. The training process of EFGCNs will stop when the maximum training iteration is 200 epochs or the loss value of the validation set remains unchanged continuously for 10 epochs. To avoid the overfitting problem, we also use the following hyperparameters, such as dropout rate, the dimensions of hidden layer and L2 regularization. The detailed (initial) experiment parameters are set as follows (We make a manual fine-tuning for EFGCNs according to

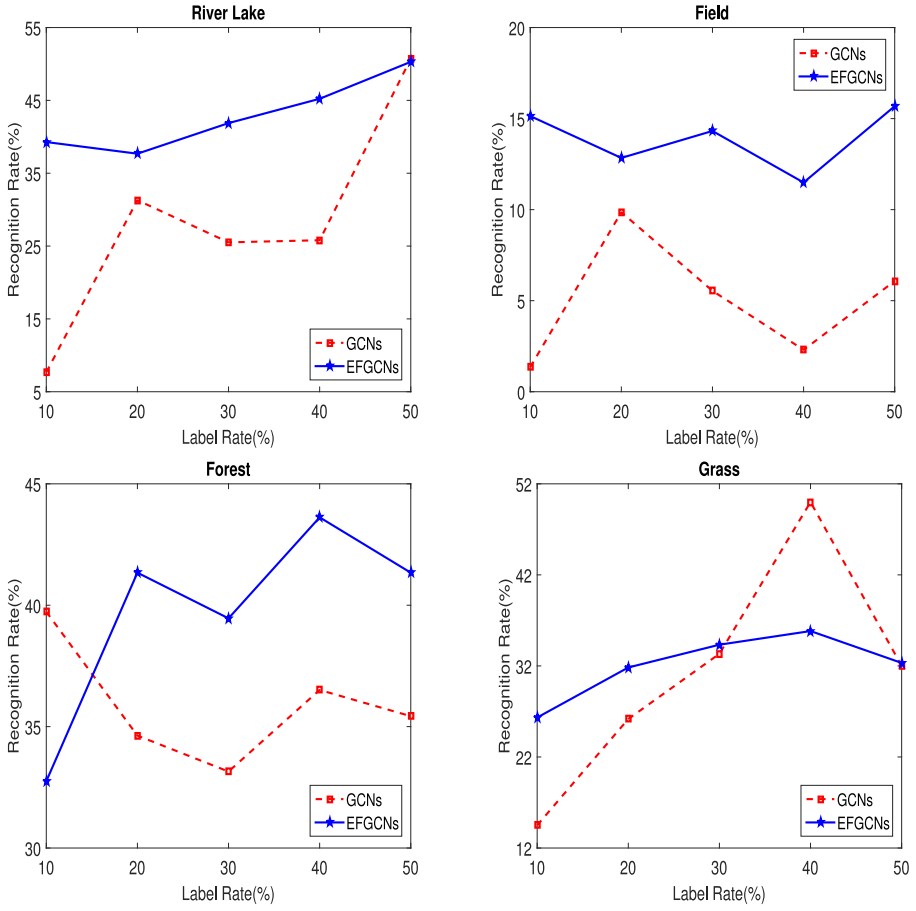

Fig. 8. Mean recognition rates of each class on the RSSCN7 dataset. Each subfigure corresponds to a single class.

the default hyperparameters of the baseline model GCNs [23], and then we select the most effective initial hyperparameters which can let EFGCNs obtain the best classification performance.): (1) For RSSCN7: 0.5 as the dropout rate, 64 as the dimensions of hidden layer and 5×10^{-7} as the L2 regularization; (2) For SAT-6: 0.5 as the dropout rate, 64 as the dimensions of hidden layer and 5×10^{-6} as the L2 regularization; (3) For Citeseer: 0.4 as the dropout rate, 32 as the dimensions of hidden layer and 5×10^{-4} as the L2 regularization; (4) For Cora: 0.5 as the dropout rate, 512 as the dimensions of hidden layer and 5×10^{-4} as the L2 regularization; (5) For NELL: 0.4 as the dropout rate, 512 as the dimensions of hidden layer and 5×10^{-6} as the L2 regularization.

6.2. Citation Networks Classification

In the existing GCNs and their variants, Citeseer, Cora and NELL are commonly-used datasets. In this section, we first compare our proposed EFGCNs with several existing semi-supervised learning models, such as HyperGCNs [25] and GCNs [23] on the Citeseer, Cora and NELL datasets. In Figs. 6 and 7, the x-axis denotes the number of labeled images in the training set and the y-axis represents the average recognition rates of GCNs and EFGCNs. In Figs. 4, 5, 8 and 9, the y-axis denotes the mean recognition rates of the single class (To better show our proposed EFGCNs on the single class's classification performance for the readers' understanding, the Appendix section in the form of tables (Tables 8–11) detailed describes the mean recognition rates with standard deviations of the single class.). In Table 2, the reported numbers express the average recognition rates with 100 random runs under 120 (Citeseer) and 140 (Cora) labeled training samples. We follow the experimental settings in [23], and the experimental results of the comparison models (except GCNs) can be obtained from [23,28].

Figs. 4 and 5 illustrate the mean recognition rates of several selected categories in the Citeseer and Cora datasets. From the experiment results of Figs. 4–6 and Table 2, we can see that, EFGCNs perform the best among all the competing methods. The reason is that, compared with GCNs, EFGCNs can automatically extract more important data information from the high-dimensional input data. Essentially, considering the geometric structures of row and column vectors of high-dimensional data at the same time, our proposed EFGCNs can learn richer data features to improve the classification of semi-supervised classification while taking advantage of the example graph and feature graph based structure relationships during the training process.

From these results in Figs. 4–6 and Table 2, we can find the following observations:

(1) In Figs. 6, our proposed EFGCNs obtain better performance than the basic GCNs models on the Citeseer, Cora and NELL datasets. Moreover, EFGCNs improve GCNs 1.7%, 2.44%, 3.36%, 3.12% and 1.46% on the Cora dataset, 0.92%, 0.86%, 1.34%, 0.18% and 0.22% on the Citeseer dataset, 3.42%, 2.02%, 0.84%, 0.54% and 0.84% on the NELL dataset, respectively, when randomly choosing 20%, 30%, 40%, 50% and 60% (10%, 15%, 20%, 25% and 30%) labeled images from the training sets.
(2) As seen in Table 2, EFGCNs obtain a significant improvement compared with several existing semi-supervised learning models. For example, when 120 training samples of the Citeseer dataset are employed as labeled samples, EFGCNs improve 15.1%, 1.5%, 2%, 8%, 7.9%, 8.8%, 1.8%, 6.6%, 1% and 0.1% over Multi-layer Perception, Manifold Regularization, Semi-supervised Embedding, Chebyshev (K = 2), Chebyshev (K = 3), GCNs, GAT, HyperGCNs, HesGCNs and GpLCNs respectively. EFGCNs improve 0.8% over the state-of-the-art model HyperGCNs when using 140 labeled training images from Cora datasets.
(3) In summary, these results indicate that EFGCNs can acquire richer data space structure information effectively even when there are few labeled training samples available.

6.3. Remote Sensing Image Classification

In this section, we report the average recognition rates of all categories in the RSSCN7 and SAT-6 datasets. To compare existing semi-supervised learning models, Table 3 compares the classification performance of our proposed EFGCNs and that of other methods on the RSSCN7 and SAT-6. In Table 3, we give the average accuracy with 100 random weight initialization of all competing methods under 650 and 150 labeled training samples of RSSCN7 and SAT-6 datasets respectively. We also follow the detailed experimental setting in [23].

From Fig. 7 and Table 3, we can see that, with the increasing number of labeled training samples, the mean recognition rates

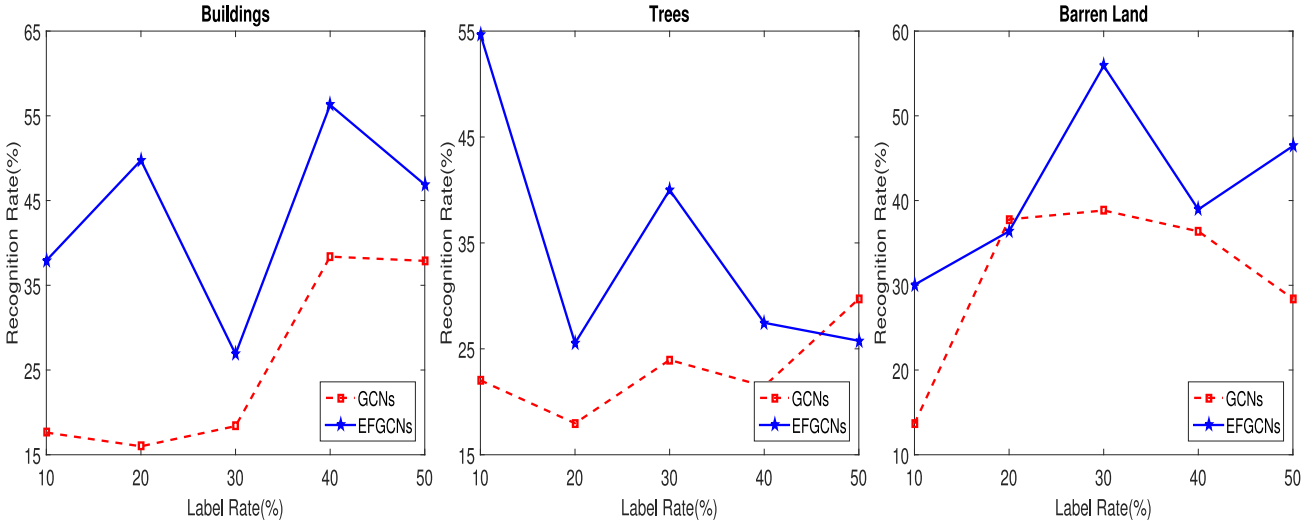

Fig. 9. Mean recognition rates of each class on the SAT-6 dataset. Each subfigure corresponds to a single class.

Table 2
Average recognition rates performance comparison of EFGCNs and different semi-supervised learning methods on the Citeseer and Cora.

Method	Citeseer (120)	Cora (140)
Multi-layer Perception	46.5	55.1
Manifold Regularization	60.1	59.5
Semi-supervised Embedding	59.6	59
Chebyshev ($K = 2$)	53.6	49.8
Chebyshev ($K = 3$)	53.7	50.5
GCNs	52.8 ± 3.3	57.2 ± 2.6
GAT	59.8	57
HyperGCNs	55	59.4
HesGCNs	60.6	59.7
GpLCNs	61.5	–
EFGCNs	61.6 ± 1.2	60.3 ± 2.5

Table 3
Average recognition rates performance comparison of EFGCNs and different semi-supervised learning methods on the RSSCN7 and SAT-6.

Method	RSSCN7 (650)	SAT-6 (150)
Chebyshev ($K = 2$)	27.1	38.5
Chebyshev ($K = 3$)	27.4	40.1
GCNs	28.5	37.8
HyperGCNs	30.2	40.8
EFGCNs	36.7	41.3

Table 4
Mean Micro-F1 with multi-times experiments of all classes on the Cora database.

Methods	20%	30%	40%	50%	60%
GCNs	61.3 ± 2.7	64.2 ± 2.9	66.2 ± 2.2	68.1 ± 1.4	70.6 ± 7.5
EFGCNs	63.1 ± 1.6	66.7 ± 1.3	69.5 ± 1.3	71.3 ± 1.3	72.1 ± 1.2

Table 5
Mean Macro-F1 with multi-times experiments of all classes on the Cora database.

Methods	20%	30%	40%	50%	60%
GCN	53.5 ± 5.4	57.4 ±5.3	60.1 ± 3.1	62.1± 2.4	65.8 ± 1.4
EFGCNs	58.6 ± 1.8	63.2 ± 1.6	66.1 ± 1.8	68.5 ± 1.5	68.6 ± 1.4

Table 6
Mean Micro-F1 with multi-times experiments of all classes on the RSSCN7 database.

Methods	10%	20%	30%	40%	50%
GCNs	19.7 ± 2.3	21.2 ± 3.3	25.4 ± 2.6	27 ± 0.6	28.7 ± 0.8
EFGCNs	28.7 ± 1.6	31.2 ± 1.3	32.7 ± 3.9	35 ± 1.4	36 ± 0.6

Table 7
Mean Macro-F1 with multi-times experiments of all classes on the RSSCN7 database.

Methods	10%	20%	30%	40%	50%
GCN	10.8 ± 2.2	13.5 ± 3.2	18.2 ± 4.7	20 ± 1	23 ± 1.2
EFGCNs	25.3 ± 2.2	28.6 ± 2	30.2 ± 0.7	32.8 ± 1.2	34.5 ± 0.7

of GCNs and EFGCNs increase. Our EFGCNs obtain the best performance compared with other competing methods. Especially when used only a small number of labeled samples, the superior performance of EFGCNs is even more obvious. This also suggests that our EFGCNs outperform GCNs in extracting the sample features of graph-structured data because EFGCNs consider the local geometry distributions of example graph and feature graph simultaneously.

As seen in these results of Fig. 7 and Table 3, we can obtain the following observations:

(1) Utilizing the spectral convolution to fuse the sample graph and feature graph into one unified EFG, the proposed EFGCNs can obtain around 4.8% improvement on the SAT-6 dataset, 8.5% improvement on the RSSCN7 dataset compared with GCNs. These demonstrate the effectiveness of EFGCNs on remote sensing image recognition.
(2) When EFG is used to describe the spatial structure information of data with few labeled training samples, the proposed EFGCNs 6.5% improvement in comparison with HyperGCNs on the RSSCN7 dataset when using 650 labeled training samples. Compared with HyperGCNs, EFGCNs have a slight improvement with 0.5% on the SAT-6 dataset when using 150 labeled training samples. The reason may be that the generated EFG using spectral EFG convolution is insufficient in this case.

To further demonstrate the performance of EFGCNs in each class, Fig. 8 shows the mean classification accuracy with multiple experiments of several chosen classes in the RSSCN7 database, such as grass, river lake, forest and field. Fig. 9 shows the average recognition rates of several classes in the SAT-6 dataset including trees, buildings and barren land. From these results, we can find that, under most conditions, our EFGCNs perform better than GCNs.

To better validate the effectiveness of our proposed EFGCNs model, thus we introduce two standard numerical analysis criteria (Micro-F1 and Macro-F1) on the Cora and RSSCN7 datasets under the different label rates. From these experimental results in Tables 4–7, we can observe that EFGCNs all obtain superior classification performances even if a small number of labeled training data were used.

7. Conclusion

With the diversification of data structure, traditional data representation learning models, such as CCA, PCA, CNNs and recurrent neural networks, cannot effectively handle the graph-structured data to extract more representative information. Recently, graph convolutional networks (GCNs) have attracted increasing attention of researchers in the field of machine learning. However, how to better construct the high-order space structure information of data is still a challenging problem while exploring the geometry structure of data for GCNs. In this paper, we have proposed a graph

Table 8
Mean recognition rates of each class on the Cora dataset.

Category	Methods	20%	30%	40%	50%	60%
Rule-	GCNs	52.2 ± 11.8	51.8 ± 6.5	54.6 ± 5.6	58.6 ± 3.5	61.2 ± 3.7
	EFGCNs	58.5 ± 7.3	57.7 ± 4.7	62.1 ± 5.6	65.4 ± 4	65.6 ± 3.3
Genetic-	GCNs	36 ± 24.3	35.8 ± 18.9	42 ± 5.9	42.6 ± 4.5	47.5 ± 4
	EFGCNs	60.7 ± 19.7	53.9 ± 12.5	63.6 ± 6.4	63.5 ± 6.7	65.3 ± 3.9
Case-	GCNs	26.2 ± 10.3	48.2 ± 13.2	47.1 ± 5.4	55.2 ± 7.5	56.2 ± 4.6
	EFGCNs	43.1 ± 12.4	56 ± 11.4	59.3 ± 12.8	60.2 ± 9.4	58 ± 7
Theory	GCNs	26.9 ± 16.9	24.7 ± 13.7	23.4 ± 9	24.4 ± 10.2	32.8 ± 5.3
	EFGCNs	35 ± 16.9	46.9 ± 4.4	47.2 ± 3.4	54.4 ± 7.3	48.8 ± 9.1

Table 9
Mean recognition rates of each class on the Citeseer dataset.

Category	Methods	20%	30%	40%	50%	60%
IR	GCNs	78 ± 4.2	79.7 ± 3.4	81.4 ± 3.4	81.3 ± 4.3	82.9 ± 1.7
	EFGCNs	79.8 ± 3.4	81.4 ± 1.9	83.7 ± 1.4	84.9 ± 2.4	84.5 ± 1.8
HCI	GCNs	57.1 ± 9	63.8 ± 6.4	62.8 ± 8.7	71.1 ± 4.3	73.7 ± 5.8
	EFGCNs	73.8 ± 7.7	72.7 ± 5.9	72.5 ± 4.7	78.6 ± 3.5	75.5 ± 4.3
ML	GCNs	84.3 ± 4.6	82.4 ± 2.5	83.5 ± 3.1	84.4 ± 3.2	85.6 ± 1.5
	EFGCNs	79.6 ± 3.5	81.6 ± 4.2	83.8 ± 4.1	85.8 ± 2	85.7 ± 2
Agents	GCNs	6.3 ± 5.5	5.9 ± 1	6.3 ± 2.4	8.5 ± 3.7	6.3 ± 3.6
	EFGCNs	9.1 ± 6.7	6.7 ± 3.2	9.7 ± 6.5	3.7 ± 0.4	5.2 ± 2.8

Table 10
Mean recognition rates of each class on the RSSCN7 dataset.

Category	Methods	10%	20%	30%	40%	50%
River	GCNs	7.7 ± 1.5	31.3 ± 4.6	25.5 ± 2.7	25.8 ± 3.6	50.7 ± 22
	EFGCNs	39.3 ± 10	37.7 ± 7.2	41.9 ± 6.3	45.2 ± 10.3	50.3 ± 4.9
Field	GCNs	1.3 ± 0.3	9.9 ± 3	5.5 ± 0.5	2.3 ± 0.3	6.1 ± 4.9
	EFGCNs	15.1 ± 9.6	12.8 ± 5	14.3 ± 4.7	11.5 ± 4.8	15.7 ± 3.1
Forest	GCNs	39.9 ± 20.4	34.6 ± 1.9	33.2 ± 6	36.5 ± 3.2	35.4 ± 3.3
	EFGCNs	32.8 ± 15.9	41.3 ± 2.8	39.5 ± 0.4	43.6 ± 6.1	41.3 ± 2.4
Grass	GCNs	14.5 ± 3.2	26.2 ± 4.1	33.3 ± 4	50 ± 28.6	32 ± 17.2
	EFGCNs	26.3 ± 1.6	31.8 ± 6.5	34.3 ± 6.7	35.8 ± 10.1	32.6 ± 6.3

Table 11
Mean recognition rates of each class on the SAT-6 dataset.

Category	Methods	10%	20%	30%	40%	50%
Buildings	GCNs	17.6 ± 17.4	16 ± 14.9	18.4 ± 17.8	38.4 ± 28.6	37.9 ± 28
	EFGCNs	37.4 ± 21.7	49.8 ± 26.9	26.9 ± 13.8	56.3 ± 32.3	46.9 ± 22.1
Trees	GCNs	22 ± 21.9	18 ± 17.2	24 ± 20.3	21.5 ± 21.1	29.7 ± 7.3
	EFGCNs	54.7 ± 31.1	25.6 ± 24.4	40 ± 32.1	27.5 ± 26.7	25.8 ± 20.7
Barren	GCNs	13.7 ± 13.3	37.8 ± 25.7	38.8 ± 23.3	36.4 ± 26.8	28.4 ± 27.1
	EFGCNs	30.1 ± 28.2	36.4 ± 22.4	55.9 ± 26.8	39 ± 19.5	46.4 ± 10.3

structure learning model, i.e. example-feature graph convolutional networks (EFGCNs). We have not only considered the geometry structure of data space (the local geometry distributions of row vectors of high-dimensional data), but also utilized the local geometry of feature space (the local geometry distributions of column vectors of high-dimensional data) during the training process of EFGCNs. Compared with GCNs, EFGCNs can capture more accurate space structure information that described the geometric distributions of data. Building a multi-layer EFGCNs allow us to extract effective data features from original sample features. To verify the performance of EFGCNs, we conduct extensive experiments on four public datasets. for remote sensing and citation networks classification. The experiment results show the superiority of our EFGCNs.

CRediT authorship contribution statement

Sichao Fu: Conceptualization, Methodology, Formal analysis, Writing - review & editing. **Weifeng Liu:** Supervision, Funding acquisition, Writing - review & editing. **Kai Zhang:** Supervision, Funding acquisition. **Yicong Zhou:** Supervision, Funding acquisition, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No.61671480), in part by the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183–008, in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (Grant No.202000009).

Appendix A

References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.
[2] H. Wang, Q.M.J. Wu, J. Wang, W. Wu, K. Yu, Optimizing simple deterministically constructed cycle reservoir network with a redundant unit pruning auto-encoder algorithm, Neurocomputing 356 (2019) 184–194.
[3] W. Liu, T. Ma, Q. Xie, D. Tao, J. Cheng, Lmae: a large margin auto-encoders for classification, Signal Process. 141 (2017) 137–143.
[4] L. Han, X.Y. Jing, F. Wu, Multi-view local discrimination and canonical correlation analysis for image classification, Neurocomputing 275 (2018) 1087–1098.
[5] X. He, W. Zhang, Emotion recognition by assisted learning with convolutional neural networks, Neurocomputing 291 (2018) 187–194.
[6] O. Jun, L. Yujian, Vector-kernel convolutional neural networks, Neurocomputing 330 (2019) 253–258.
[7] I. Ariav, I. Cohen, An end-to-end multimodal voice activity detection using wavenet encoder and residual networks, IEEE J. Sel. Topics Signal Process. 13 (2019) 265–274.
[8] H.G. A, W.Z. A, L.X. A, H.J. B, W.Z. B, Z.Z. C, Ship detection based on squeeze excitation skip-connection path networks for optical remote sensing images, Neurocomputing 332 (2019) 215–223..

[9] X. Ma, W. Liu, S. Li, D. Tao, Y. Zhou, Hypergraph p-laplacian regularization for remotely sensed image recognition, IEEE Trans. Geosci. Remote Sens. 57 (2018) 1585–1595.
[10] Y. Feng, P. Zhou, J. Xu, S. Ji, D.O. Wu, Video big data retrieval over media cloud: A context-aware online learning approach, IEEE Trans. Multimedia (2018), https://doi.org/10.1109/TMM.2018.2885237.
[11] M. Amin-Naji, A. Aghagolzadeh, M. Ezoji, Ensemble of cnn for multi-focus image fusion, Inform. Fusion 51 (2019) 201–214.
[12] M. Lan, Z. Zhang, Y. Lu, J. Wu, Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis, in: Proc. Int. Joint Conf. Neural Networks, 2016, pp. 3172–3179..
[13] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, in: Proc. Int. Conf. Mach. Learn., 2016, pp. 2014–2023..
[14] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Process. Mag. 30 (2013) 83–98.
[15] U. Chaudhuri, B. Banerjee, A. Bhattacharya, Siamese graph convolutional network for content based remote sensing image retrieval, Compu. Vis. Image Und. 184 (2019) 22–30.
[16] R. Xie, Z. Liu, J. Jia, H. Luan, M. Sun, Representation learning of knowledge graphs with entity descriptions, in: Proc. AAAI Conf. Artificial Intelligence, 2016..
[17] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 1024–1034..
[18] H. Gao, Z. Wang, S. Ji, Large-scale learnable graph convolutional networks, in: Proc. Int. Conf. Knowledge Discovery and Data Mining, 2018, pp. 1416–1424..
[19] Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, in: Proc. Int. Conf. Learning Representations, 2015..
[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, in: Proc. Conf. Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734..
[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: Proc. Int. Conf. Learning Representations, 2017..
[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778..
[23] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proc. Int. Conf. Learning Representations, 2017..
[24] R. Li, S. Wang, F. Zhu, J. Huang, Adaptive graph convolutional neural networks, in: Proc. AAAI Conf. Artificial Intelligence, 2018..
[25] N. Yadati, M. Nimishakavi, P. Yadav, A. Louis, P. Talukdar, Hypergcn: Hypergraph convolutional networks for semi-supervised classification, in: Proc. Int. Conf. Multimedia and Expo, 2019..
[26] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proc. AAAI Conf. Artificial Intelligence, 2019..
[27] S. Fu, W. Liu, D. Tao, Y. Zhou, L. Nie, Hesgcn: Hessian graph convolutional networks for semi-supervised classification, Inform. Sci. 514 (2020) 484–498.
[28] S. Fu, W. Liu, K. Zhang, Y. Zhou, D. Tao, Semi-supervised classification by graph p-laplacian convolutional networks, Inform. Sci. 560 (2021) 92–106.
[29] S. Mallat, A wavelet tour of signal processing, Leiden (1999).
[30] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, in: Proc. Int. Conf. Learning Representations, 2013..
[31] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, in: Proc. Adv. Neural Inf. Process. Syst., 2015..
[32] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3844–3852..
[33] M. Collins, S. Dasgupta, R.E. Schapire, A generalization of principal components analysis to the exponential family, in: Proc. Adv. Neural Inf. Process. Syst., 2002, pp. 617–624..
[34] B. Jiang, C. Ding, J. Tang, Graph-laplacian pca: Closed-form solution and robustness, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2013, pp. 3492–3498..
[35] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (2003) 1373–1396.
[36] J. He, Y. Bi, B. Liu, Z. Zeng, Graph-dual laplacian principal component analysis, J. Amb. Intel. Hum. Comp. (2018) 1–14.
[37] J.-X. Liu, C.-M. Feng, X.-Z. Kong, Y. Xu, Dual graph-laplacian pca: A closed-form solution for bi-clustering to find checkerboard structures on gene expression data, IEEE Trans. Knowl. Data Eng. (2019).
[38] X. Wang, J. Liu, Y. Cheng, A. Liu, E. Chen, Dual hypergraph regularized pca for biclustering of tumor gene expression data, IEEE Trans. Knowl. Data Eng. (2018).
[39] M. Yin, J. Gao, Z. Lin, Q. Shi, Y. Guo, Dual graph regularized latent low-rank representation for subspace clustering, IEEE Trans. Image Process. 24 (2015) 4918–4933.
[40] P. Li, J. Bu, C. Chen, Z. He, D. Cai, Relational multimanifold coclustering, IEEE Trans. Cybernetics 43 (2013) 1871–1881.
[41] L. Tong, J. Zhou, X. Bai, Y. Gao, Dual graph regularized nmf for hyperspectral unmixing, in: Proc. Int. Conf. Digital Image Computing: Techniques and Applications, 2014, pp. 1–8..
[42] C. Wang, N. Yu, M.-J. Wu, Y.-L. Gao, J.-X. Liu, J. Wang, Dual hyper-graph regularized supervised nmf for selecting differentially expressed genes and tumor classification, IEEE/ACM Trans. Comput. Bi. (2020), https://doi.org/10.1109/TCBB.2020.2975173.
[43] R. Zeng, J. Wu, Z. Shao, L. Senhadji, H. Shu, Quaternion softmax classifier, Electron. Lett. 50 (2014) 1929–1931.
[44] P.-T. De Boer, D.P. Kroese, S. Mannor, R.Y. Rubinstein, A tutorial on the cross-entropy method, Ann. Oper. Res. 134 (2005) 19–67.
[45] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G.N. Hullender, Learning to rank using gradient descent, in: Proc. Int. Conf. Mach. Learn., 2005, pp. 89–96..
[46] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep learning via semi-supervised embedding, in: Neural Networks: Tricks of the Trade, Springer, Berlin, Heidelberg, 2012, pp. 639–655.
[47] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[48] C. Bai, J. Guo, L. Guo, J. Song, Deep multi-layer perception based terrain classification for planetary exploration rovers, Sensors 19 (2019) 3102.
[49] Q. Zou, L. Ni, T. Zhang, Q. Wang, Deep learning based feature selection for remote sensing scene classification, IEEE Geosci. Remote Sens. Lett. 12 (2015) 2321–2325.
[50] X. Gong, Z. Xie, Y. Liu, X. Shi, Z. Zheng, Deep salient feature based anti-noise transfer network for scene classification of remote sensing imagery, Remote Sens. 10 (2018) 410.
[51] G. Bisson, C. Grimal, Co-clustering of multi-view datasets: a parallelizable approach, in: Proc. Int. Conf. Data Mining, 2012, pp. 828–833..
[52] C. Zhu, D. Miao, Entropy-based multi-view matrix completion for clustering with side information, Pattern Anal. Appl. 1–12 (2019).
[53] H. Poon, P. Domingos, Joint inference in information extraction, in: Proc. AAAI Conf. Artificial Intelligence, volume 7, 2007, pp. 913–918..
[54] S.-J. Huang, C.-T. Hsieh, Coiflet wavelet transform applied to inspect power system disturbance-generated signals, IEEE Trans. Aero. Elec. Sys. 38 (2002) 204–210.
[55] P. Rieder, J. Gotze, J. Nossek, C.S. Burrus, Parameterization of orthogonal wavelet transforms and their implementation, IEEE Trans. Circuits Systems II: Analog Digital Signal Processing 45 (1998) 217–226.
[56] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proc. Int. Conf. Learning Representations, 2014..

Sichao Fu received his master's degree in electronics and communication engineering from the China University of Petroleum (East China), in 2020. Currently, he is pursuing the Ph.D. degree at the Huazhong University of Science and Technology. His research interests include pattern recognition and deep manifold learning.

Weifeng Liu (M'12-SM'17) received the double B.S. degrees in automation and business administration and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2002 and 2007, respectively. He was a Visiting Scholar with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia, from 2011 to 2012. He is currently a Full Professor with the College of Information and Control Engineering, China University of Petroleum, Qingdao, China. He has authored or co-authored a dozen papers in top journals and prestigious conferences, including four Essential Science Indicators (ESI) highly cited papers and two ESI hot papers. His research interests include computer vision, pattern recognition, and machine learning. Prof. Liu serves as an Associate Editor for the Neural Processing Letters, the Co-Chair for the IEEE SMC Technical Committee on Cognitive Computing, and a Guest Editor for the special issue of the Signal Processing, the IET Computer Vision, the Neurocomputing, and the Remote Sensing. He also serves over 20 journals and over 40 conferences.

Zhang Kai received the Ph.D. degree in petroleum engineering from the China University of Petroleum (East China), Qingdao, China, in 2008. From June 2007 to May 2008, he studied with the University of Tulsa, Tulsa, OK, USA. He has been a Teacher with the China University of Petroleum (East China) since 2008. He teaches courses, including fluid flow in porous media and reservoir engineering. As a Project Leader, he has been in charge of three projects supported by the Natural Science Foundation of China, one project supported by the National Natural Science Foundation of Shandong Province, and 20 projects supported by SINOPEC, CNOOC, and CNPC. He has already published more than 60 papers. His research focuses on reservoir simulation, production optimization, history matching, and the development of the nonconventional reservoir.

Yicong Zhou (M'07-SM'14) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently an Associate Professor and the Director with the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau, Macau, China. His research interests include chaotic systems, multimedia security, image processing and understanding, and machine learning. Dr. Zhou was a recipient of the Third Price of Macau Natural Science Award in 2014. He served as an Associate Editor for the Neurocomputing, the Journal of Visual Communication and Image Representation, and the Signal Processing: Image Communication. He is a Co-Chair of the Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society.