Dynamic Graph Learning Convolutional Networks for Semi-supervised Classification

SICHAO FU and WEIFENG LIU, College of Control Science and Engineering, China University of Petroleum (East China), China
WEILI GUAN, Faculty of Information Technology, Monash University Clayton Campus, Australia
YICONG ZHOU, Faculty of Science and Technology, University of Macau, China
DAPENG TAO, School of Information Science and Engineering, Yunnan University, China
CHANGSHENG XU, Institute of Automation, Chinese Academy of Sciences, China

Over the past few years, graph representation learning (GRL) has received widespread attention on the feature representations of the non-Euclidean data. As a typical model of GRL, graph convolutional networks (GCN) fuse the graph Laplacian-based static sample structural information. GCN thus generalizes convolutional neural networks to acquire the sample representations with the variously high-order structures. However, most of existing GCN-based variants depend on the static data structural relationships. It will result in the extracted data features lacking of representativeness during the convolution process. To solve this problem, dynamic graph learning convolutional networks (DGLCN) on the application of semi-supervised classification are proposed. First, we introduce a definition of dynamic spectral graph convolution operation. It constantly optimizes the high-order structural relationships between data points according to the loss values of the loss function, and then fits the local geometry information of data exactly. After optimizing our proposed definition with the one-order Chebyshev polynomial, we can obtain a single-layer convolution rule of DGLCN. Due to the fusion of the optimized structural information in the learning process, multi-layer DGLCN can extract richer sample features to improve classification performance. Substantial experiments are conducted on citation network datasets to prove the effectiveness of DGLCN. Experiment results demonstrate that the proposed DGLCN obtains a superior classification performance compared to several existing semi-supervised classification models.

CCS Concepts: • **Computing methodologies** → *Semi-supervised learning settings*; • **Theory of computation** → *Semi-supervised learning*;

Additional Key Words and Phrases: Graph representation learning, graph convolutional networks, semi-supervised classification

This work was supported in part by the Major Scientific and Technological Projects of CNPC under Grant no. ZD2019-183-008; in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (Grant no. 202000009); in part by the Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3).
Authors' addresses: S. Fu and W. Liu (corresponding author), College of Control Science and Engineering, China University of Petroleum (East China), Qingdao, 266580, China; emails: fusichao_upc@163.com, liuwf@upc.edu.cn; W. Guan, Faculty of Information Technology, Monash University Clayton Campus, Australia; email: honeyguan@gmail.com; Y. Zhou, Faculty of Science and Technology, University of Macau, Macau, 999078, China; email: yicongzhou@um.edu.mo; D. Tao, School of Information Science and Engineering, Yunnan University, Kunming, 650091, China; email: dapeng.tao@gmail.com; C. Xu, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China; email: csxu@nlpr.ia.ac.cn.

1551-6857/2021/03-ART4 $15.00
https://doi.org/10.1145/3412846

ACM Reference format:
Sichao Fu, Weifeng Liu, Weili Guan, Yicong Zhou, Dapeng Tao, and Changsheng Xu. 2021. Dynamic Graph Learning Convolutional Networks for Semi-supervised Classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 1s, Article 4 (March 2021), 13 pages.
https://doi.org/10.1145/3412846

1 INTRODUCTION

With the popularity of cloud computing technologies and the internet of things and the booming of the social networks, global data are growing in an explosive way that has never been seen before. In these massive data, there exist massive valuable information and knowledge [31], which may play a vital role in business competition, government management, and scientific research. At the same time, data types and structures have become increasingly complex. How to effectively manage these massive data and quickly obtain valuable information have become the research focus of the current data fields. As a kind of universal data structure in computer science, graphs have strong data expression ability, which are more suitable for describing the data objects with complex structures. Many areas, such as medicinal chemistry [8, 16], geographic information system [19, 27], and social networks [26, 43], can be modeled with graphs. In recent years, many GRL models [34, 38] have shown their superiorities to extract the sample features of graphs or high-order data and have been applied in many areas including classification [1, 4, 36] and retrieval [11, 44, 47].

To adequately utilize the sample structural relationships, many researchers have made some attempts [33, 48] to optimize these existing models. The manifold regularization [5] is one prominent work of the GRL. By regarding the manifold structures as a loss function's regularization term, models can make full use of the local geometry probability distribution of data to improve their generalization ability. Luo et al. [24] exploited the Laplacian regularization [13] to make the functions of the discriminative shared subspace change smoothly along the underlying manifold for semi-supervised multilabel image classification. Zhao et al. [49] proposed a robust dual clustering method with adaptive manifold regularization to address the high-dimension data clustering by utilizing the learned affinity matrix-based manifold regularization term to better the intrinsic local geometrical data structures. By combining the Hessian regularization [23] and sample features that are constructed from different views, Liu et al. [21] proposed a multiview Hessian regularized logistic regression to solve the problems of the multi-modal content and different representations for action recognition. Liu et al. [22] proposed a sparse coding model with p-Laplacian regularization by utilizing the high-order manifold structural information of the p-Laplacian [2] for human activity recognition. Ma et al. [25] proposed a logistic regression classifier with hypergraph p-Laplacian regularization by utilizing the combination of the hypergraph theory [12] and p-Laplacian to preserve the richer sample structural information of data for remotely sensed image classification. Zhang et al. [46] proposed a sparse low-rank approximation with ensemble manifold regularization to solve the multi-view feature dimensionality reduction problem by introducing the group sparsity constraint and ensemble manifold regularization to the basic least-squares component analysis.

With the continuous development of deep learning [39, 45], other relatively significant works are graph neural networks (GNN) [7, 40, 50]. By using the structural information within the sample's k-hop neighbors from the spectral or spatial [41] domain, each sample of the graphs recursively aggregates the sample features of its k-hop neighbors. The most representative model of spectral methods is GCN [18], which learn the sample features by utilizing a graph Laplacian-based simple linear function to define the convolution operations, which aimed to use the structural relationships of each sample. Yadati et al. [42] utilized the hypergraph structure to encode the beyond pairwise connection relationships of each sample. Atwood and Towsley [3] utilized a transition

matrix with degree-normalized to express the spatial structural relationships from each node to its neighbors. Velickovic et al. [30] applied the attention mechanism to the graph convolution for citation network classification task, which aimed to assign different weight values for the edge of arbitrary nodes. Zhou et al. [51] presented the graph convolutional network-based hashing model to learn an affinity graph, which can effectively learn the similarity-preserving binary embeddings of the massive image retrieval. Yin et al. [15] proposed a relation-aware attention mechanism based graph convolutional networks for jointly extracting entities and relations. It can better utilize the adjacent node features and edge information when encoding node representation. Wu et al. [37] proposed a deep Siamese attention network that incorporates spatiotemporal information into deep convolutional networks to jointly learn their feature representations and similarity metrics for video-based person re-identification tasks.

In fact, once the structural information of the sample is constructed according to the original data, these above-mentioned works never change it in the training process of the model, i.e., static high-order data representation learning structure. It is difficult to make sure that the computed static data structural information is optimal or most representative for all applications.

To build a dynamic graph structure learning method, we introduce the dynamic graph learning convolutional networks (DGLCN) with semi-supervised learning. The aims of our proposed method are to constantly optimize the sample structural information during the training process of DGLCN. Particularly, we generalize the original spectral graph convolution definition into the spectral convolution on dynamic graph. And then we get a single-layer propagation rule of DGLCN by optimizing the spectral dynamic graph convolution with the one-order approximation. After k epoch iteration until model fitting, multi-layer DGLCN can get the optimal structural information, and then it can further get the most representative data features. In addition, the initial sample structural information is computed through the graph Laplacian formula. In each training iteration process, the sample structural information of each convolution layer is restructured based on the extracted features of the final layer of the last iteration.

The contributions of this article are as follows:

- We propose the definition of the spectral dynamic graph convolution to acquire richer high-order structural relationships. It is able to preserve the high-order manifold structure information of data properly.
- We further optimize the one-approximation of the spectral dynamic graph convolution to build a single-layer dynamic graph learning convolutional networks (DGLCN).
- Multi-layer DGLCN can be built by stacking the above-mentioned single-layer DGLCN.
- To validate the classification performance of DGLCN, we have conducted massive comparison experiments on the Cora [6], Citeseer [32], and Pubmed [28] datasets for the classification task. Experiment data on the three datasets have shown the superior classification performance of DGLCN than several existing models.

The rest of this article is formed as follows: In Section 2, we briefly describe the related work about the spectral graph convolution. Section 3 describes our proposed DGLCN framework. A large number of experimental results on the three datasets are given in Section 4. Finally, we give the conclusion of this article in Section 5.

2 SPECTRAL CONVOLUTION ON STATIC GRAPH

The original spectral graph convolution can be denoted as the multiplication of a signal X and a filter g_θ, i.e.,

$$\begin{aligned} g_\theta(Lap) \star X &= U((U^T g_\theta) \odot (U^T X)) \\ &= U g_\theta(\Lambda) U^T X. \end{aligned} \tag{1}$$

In this formulation, θ denotes the filter coefficient. The normalized graph Laplacian Lap is computed by $Lap = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, which is a positive definite matrix. In addition, A represents the adjacency relationship matrix of a signal X or high-order data, $D_{ii} = \sum_j A_{ij}$. We can get a diagonal orthonormal eigenvectors matrix U and a diagonal non-negative eigenvalues matrix Λ according to the eigen decomposition of Lap. The Fourier transform of a signal X is denoted as the $U^T X$. The $\odot$ denotes the element-wise Hadamard.

Because of the expensive computation cost for the eigen decomposition of Lap, this method is not applicable for a large graph. To avoid this problem, Defferrard et al. [9] utilized the truncated Chebyshev expansion with K-order polynomials to approximate the $g_\theta(\Lambda)$. Thus, the $g_\theta(\Lambda)$ can be denoted as the following form:

$$g_\theta(\Lambda) = \sum_{k=0}^{k} \theta_k T_k(\Lambda), \tag{2}$$

where $T_k(\widetilde{Lap})$ can be written as $\widetilde{Lap} = \frac{2}{\lambda_{max}} Lap - I_N$. The Chebyshev $T_k(X)$ describes the following form recursively: $T_0(X) = 1, T_1(X) = X$ and $T_k(X) = 2XT_{k-1}(X) - T_{k-2}(X)$. λ_{max} presents the maximum eigenvalue of graph Laplacian Lap. To build a linear model and limit the convolution's order, Kipf and Welling [18] used the Chebyshev with one-order polynomial, i.e., $K = 1$. Following, it introduced the optimized graph convolutional networks (GCN) model, i.e.,

$$\begin{aligned} g_\theta(Lap) \star X &= \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X\theta \\ &= BH^{(L)} W^{(L)}. \end{aligned} \tag{3}$$

For Equation (3), $\widetilde{A} = A$ (adjacency matrix) $+I_N$. This method can be interpreted as the fusion process of the feature information $H^{(L)}$ and structural information B of samples in the learning process. $B = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$ represents the structural information. $W^{(L)}$ denotes the weight parameters learned on each layer, $H^{(L)}$ can be regarded as the output data features of the last layer with $H^{(0)} = X$. The detailed proof process can be found in Reference [18].

3 DYNAMIC GRAPH LEARNING CONVOLUTIONAL NETWORKS

In this section, we introduce the proposed dynamic graph learning convolutional networks (DGLCN) framework. Specifically, the spectral dynamic graph convolution with one-order Chebyshev polynomial is presented first. It can form a single-layer convolution rule of DGLCN. And then we analyze the implementation process of multi-layer DGLCN model in detail.

3.1 Spectral Convolution on Dynamic Graph

Compared to convolutional neural networks [29], GCN can learn more representative data features by utilizing the static structural relationships of samples. However, it is difficult to guarantee that the computed structural information is optimal or the extracted sample features are richer for all tasks.

To surmount this problem, we develop an effective definition of spectral dynamic graph convolution with K-order Chebyshev polynomials. It dynamically updates the structural information of samples during the convolution process. It can be written as the following definition, i.e.,

$$g_\theta(Lap^{(a)}) \star X^{(a)} = \sum_{k=0}^{k} \theta_k T_k(\widetilde{Lap^{(a)}}) X^{(a)}, \tag{4}$$

where $\widetilde{Lap^{(a)}} = \frac{2}{\lambda_{max}} Lap^{(a)} - I_N$. $Lap^{(a)}$ corresponds to the adjacency relationship matrix of the arbitrarily signal $X^{(a)}$.

3.2 A Single-layer Convolution Rule of DGLCN

We also let $K = 1$ that suggested in Reference [18] to reduce its convolution operation. It has two advantages: (1) It further reduces the computation complexity of the spectral dynamic graph convolution (from $O(N^2)$ to $O(K|\varepsilon|)$). (2) The spectral convolution on dynamic graph with one-order approximation can better describe the complex and high-order structural relationships between data. Thus, this definition can be simplified as:

$$g_\theta(Lap^{(a)}) \star X^{(a)} = \theta_0 X^{(a)} + \theta_1 (Lap^{(a)} - I_N) X^{(a)}. \tag{5}$$

To further reduce the computation parameters, we use a single parameter $\theta = \theta_0 = -\theta_1$ and a renormalization trick [18]: $I_N + D^{-\frac{1}{2}} A^{(a)} D^{-\frac{1}{2}} \rightarrow \widetilde{D}^{-\frac{1}{2}} (A^{(a)} + I_N) \widetilde{D}^{-\frac{1}{2}}$. Thus, we can make a further simplification on Equation (5), i.e.,

$$\begin{aligned} g_\theta(Lap^{(a)}) \star X^{(a)} &= \theta (I_N + D^{-\frac{1}{2}} A^{(a)} D^{-\frac{1}{2}}) X^{(a)} \\ &= \widetilde{D}^{-\frac{1}{2}} (A^{(a)} + I_N) \widetilde{D}^{-\frac{1}{2}} X^{(a)} \theta. \end{aligned} \tag{6}$$

Based on Equation (6), we can get a single-layer dynamic propagation rule $f(X^{(a)}, W, A^{(a)})$, i.e.,

$$H^{(L+1)} = \sigma(B^{(L)} H^{(L)} W^{(L)}). \tag{7}$$

For Equation (7), $B^{(L)} = \widetilde{D}^{-\frac{1}{2}} (A^{(L)} + I_N) \widetilde{D}^{-\frac{1}{2}}$ denotes the graph Laplacian-based dynamic structural information. σ denotes the activation function. $A^{(L)}$ denotes the adjacency relationship matrix of each layer. In addition, we use the same calculation method that described in Reference [10] to get the structural information $A^{(L)}$.

3.3 Multi-layer DGLCN Model

Based on a single-layer DGLCN Equation (7), we can build a deeper DGLCN. Figure 1 describes the basic thought of the proposed multi-layer DGLCN:

$$H^{(L+1)} = B^{(L)} (\sigma(B^{(L-1)} \cdots \sigma(B^{(0)} X W^{(0)}) \cdots W^{(L-1)})) W^{(L)}. \tag{8}$$

A deeper DGLCN will cause the over-smoothing issue of the extracted features [20]. Following, we use a widely used two-layer DGLCN Equation (9) to describe the updated process of the $B^{(L)}$ in detail:

$$H^{(L+1)} = B^{(1)} (\sigma(B^{(0)} X W^{(0)})) W^{(1)}. \tag{9}$$

Two-layer DGLCN can be built by stacking the proposed dynamic propagation rule. First, we introduce the convolution operation of the first layer, i.e.,

$$H^{(1)} = \sigma(B^{(0)} X W^{(0)}). \tag{10}$$

For each convolution layer, we utilize the Rectified Linear Unit nonlinear activation function. X denotes the original input data matrix. $W^{(0)}$ represents the first layer's weight parameter. We first calculate the adjacency matrix $A^{(0)}$ according to the original features X by the k-Nearest Neighbor with the Euclidean, and then we can get the $B^{(0)}$. After the first convolution, we can get the learned data features $H^{(1)}$ of the first convolution layer by fusing the $B^{(0)}$ and X.

The second layer has a similar convolution process, but the structural information will constantly update until model fitting. First, we give the convolution formulation of the second layer, i.e.,

$$H^{(2)} = B^{(1)} H^{(1)} W^{(1)}. \tag{11}$$

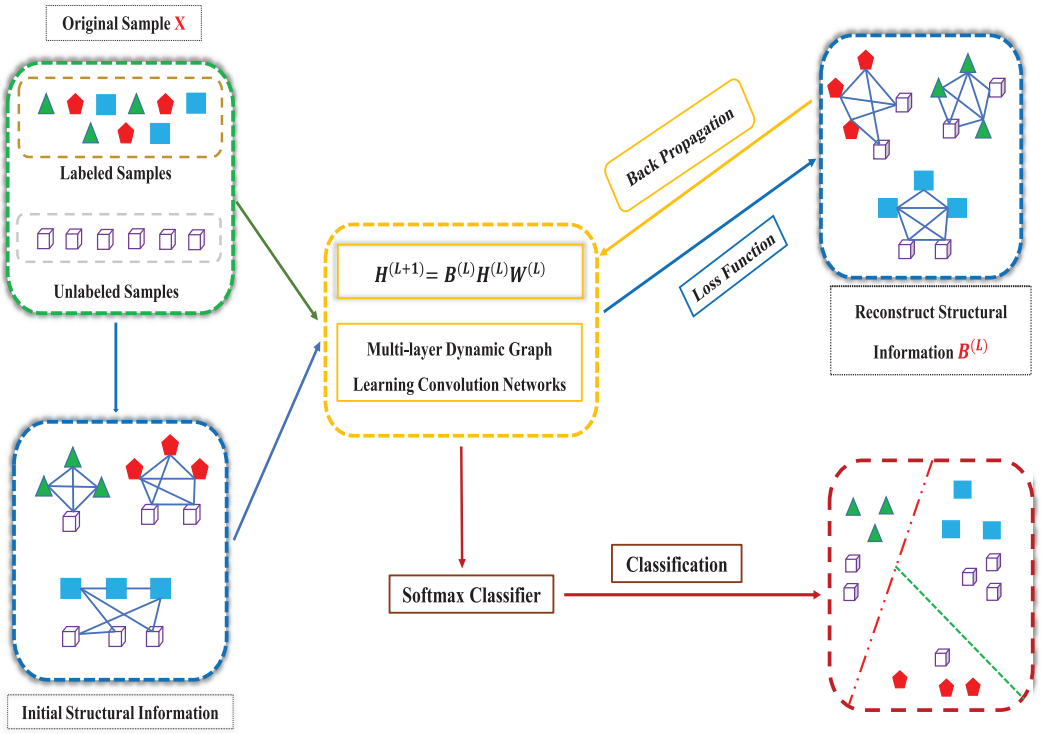

Fig. 1. The basic framework of multi-layer DGLCN model.

$W^{(1)}$ represents the weight parameter of the second convolution layer. In the first training iteration process, the $B^{(1)}$ is equal to $B^{(0)}$. After two-layer convolution, the extracted sample features $H^{(2)}$ are the input of the classifier and the loss function, and then we can get the recognition accuracy and loss value of the first training iteration. If the output values of the loss function do not satisfy the specific conditions, DGLCN will implement the second training iteration process. Before the second training iteration, our proposed model will recalculate $B^{(1)}$. The $B^{(1)}$ is calculated according to the $H^{(2)}$. The calculated method is the same as the first layer.

Due to the richer extracted features, i.e., $H^{(2)}$, thus the updated $B^{(1)}$ is optimal compared to the $B^{(1)}$ of the first training iteration. This training iteration process will carry out until model fitting. Finally, we can get the best structural information matrix, and then DGLCN can get the most representative sample features. In the output layer, we utilize the Softmax classifier and cross entropy loss function to deal with the extracted sample features $H^{(2)}$. The cross entropy loss function can be written as $C = -\sum_k y_k \log Z_k$. Z_k represents the probability matrix of the Softmax function handled. y_k denotes the true label information. The Softmax function can be represented as the following definition: i.e., $f(Z_j) = \frac{e^{Z_j}}{\sum_{i=1}^{n} e^{Z_i}}$. In addition, the input values of the Softmax function $f(Z_j)$ is $H^{(2)}$. We summarize a two-layer dynamic graph learning convolutional network in Algorithm 1.

4 EXPERIMENTS

In this section, we implement substantial experiments on the Cora [6], Citeseer [32], and Pubmed [28] datasets for the semi-supervised classification task. And then, we compare our proposed

ALGORITHM 1: A two-layer dynamic graph learning convolutional networks

Input: Datasets X
Hyperparameter: Dropout rate, hidden units, iteration epoch k etc.
Output: Mean recognition accuracy
Construct adjacency matrix $A^{(0)}$ of the input datasets
Let $B^{(0)} = B^{(1)}$
Initialize the hyperparameters
for $i = 0 \rightarrow k - 1$
$H^{(1)} = RELU(B^{(0)}XW^{(0)})$.
$H^{(2)} = Softmax(B^{(1)}H^{(1)}W^{(1)})$.
Compute loss values of the loss function.
If the output values of the cross entropy loss function satisfy the specific conditions, DGLCN will stop training. Otherwise, DGLCN will optimize $B^{(1)}$ according to $H^{(2)}$.
$H^{(1)} = RELU(B^{(0)}XW^{(0)})$.
$H^{(2)} = Softmax(B^{(1)}H^{(1)}W^{(1)})$.
Compute loss values of the loss function.
If the output values of the cross entropy loss function satisfy the specific conditions, DGLCN will stop training. Otherwise, DGLCN will update $W^{(0)}$ and $W^{(1)}$, and then carry out the next training iteration.
until convergence
Get the optimal $B^{(1)}$, $W^{(0)}$ and $W^{(1)}$.
Send the acquired data features $H^{(2)}$ to classifier.
Give the mean recognition accuracy.

DGLCN model with the several existing semi-supervised classification models to show the superior performance of DGLCN. In addition, we describe the experimental datasets, experimental parameters, and experimental results analysis that we used in the experimental process.

4.1 Experimental Datasets

The Cora [6] dataset contains 2,708 samples collected from seven classes, including neural networks, case-based, theory, probabilistic-methods, genetic algorithms, rule-learning, and reinforcement learning. The feature dimensional of its are 1433 columns.

The Citeseer [32] dataset is composed of 3,327 samples, totally six categories including HCI (Human-Computer Interaction), AI (Artificial Intelligence), DB (Database), Agents, ML (Machine Learning), and IR (Information Retrieval) scientific publications. The dimensional of each scientific publication are 3703 columns. For Cora and Citeseer dataset, the data features of each sample are described by the bag-of-words method.

In Pubmed [28] dataset, there are totally three categories including diabetes mellitus type 2, diabetes mellitus experimental, and diabetes mellitus type 1, which is collected from diabetes patients and consists of 19,717 samples. The feature dimensional are 500 columns for the diabetes mellitus books.

4.2 Experimental Parameters

In our experiment, we utilize the total data of Cora and Citeseer, and 5,000 data of the Pubmed dataset to carry out the semi-supervised classification task. We choose 1,000 samples to form the testing dataset, the 500 data are chosen for the validation dataset, the remaining data of each dataset as the training dataset. For Cora and Citeseer, we randomly choose 20%, 30%, 40%, 50%, and 60% samples of the training dataset as the labeled samples, the rest data as the unlabeled samples. For Pubmed dataset, the 5%, 10%, 15%, 20%, and 25% samples are used for labeled samples. The testing

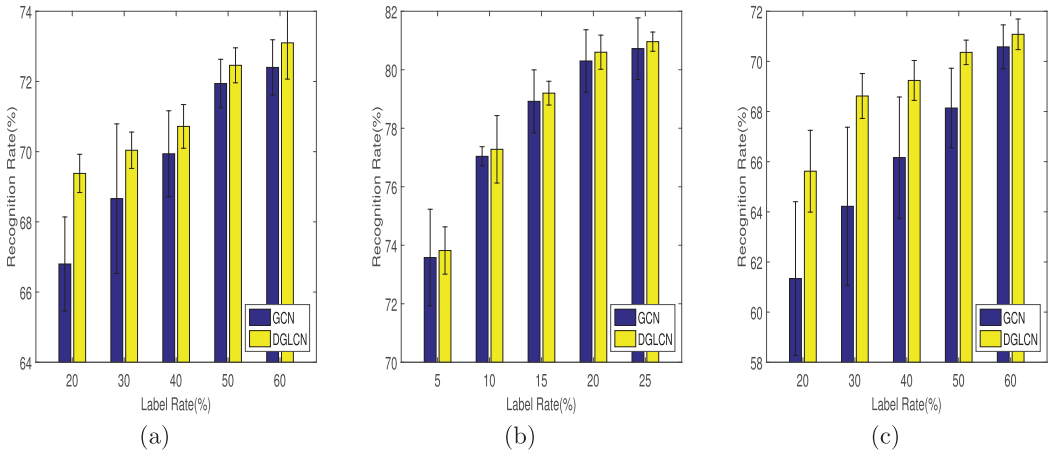

Fig. 2. Mean classification accuracy of all classes for (a) Citeseer, (b) Pubmed, and (c) Cora datasets.

Table 1. More Comparsion Data Analysis on the Cora and Citeseer Datasets

Method	Citeseer (120)	Cora(140)
Chebyshev (K=2)	53.6	49.8
ManiReg	60.1	59.5
SemiEmb	59.6	59
MLP	46.5	55.1
Chebyshev (K=3)	53.7	50.5
GCN	52.8	57.2
HyperGCN	55	59.4
HpLapGCN	**62.5**	59.8
GAT	59.8	57
DGLCN	61.4	**61.8**

set and validation set of each dataset are all labeled samples. In addition, we repeat this process independently five times, which aims to avoid any experimental contingency introduced by the random partitioning of samples.

We utilize the adaptive moment estimation [17] method with a 0.01 learning rate to optimize the error values of the loss function. In the training process, DGLCN randomly initializes the weight matrix by the Xavier [14] method. The maximum training iteration values of the proposed DGLCN are 200 times. Once the loss values of loss function remains unchanged for 10 epochs, our model will stop training. In addition, the dropout rate and L2 regularization are utilized to avoid the overfitting issue. We also set the output dimension of the hidden layer. The experimental hyperparameters are shown as follows: Cora: 0.3 (dropout rate), 32 (dimension of hidden units), and 5×10^{-4} (L2 coefficient); Citeseer: 0.4 (dropout rate), 32 (dimension of hidden units), and 5×10^{-4} (L2 coefficient); and Pubmed: 0.4 (dropout rate), 16 (dimension of hidden units), and 5×10^{-4} (L2 coefficient).

4.3 Experimental Analysis

In these experiments, we first show the experimental data of our proposed DGLCN and GCN model on three datasets. Figure 2 shows the mean classification accuracy of overall classes for

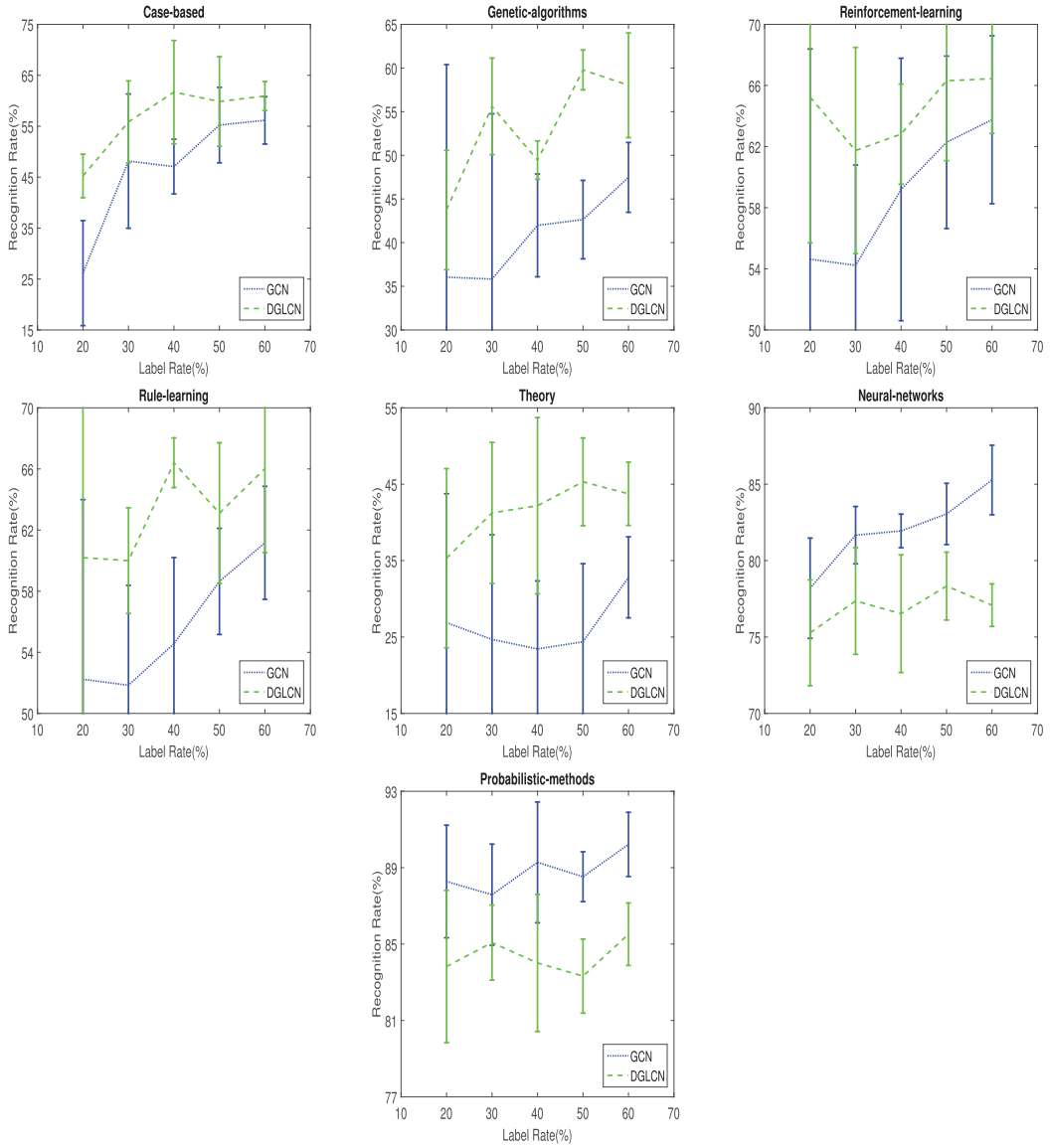

Fig. 3. Mean classification accuracy of single category for Cora dataset, including Reinforcement-learning, Theory, Genetic-algorithms, Probabilistic-methods, Rule-learning class, Neural-networks, Case-based. Each subfigure corresponds to a single category.

each model in the form of the errorbar histogram. The errorbar line chart is selected to describe the mean classification accuracy of single class. In Figure 2, the y-axis represents the mean classification accuracy (in percentages) over all categories, the x-axis denotes the different label rate of the training set. In Table 1, we make more comparative experiment results, such as Chebyshev (K=3) [9], Chebyshev (K=2) [9], ManiReg [5], SemiEmb [35], HyperGCN [42], HpLapGCN [10], GAT [30], and MLP. In Table 1, we follow the experiment setting in Reference [18]. We report the mean recognition accuracy (in percentages) with 100 random trains on the 120 labeled data and

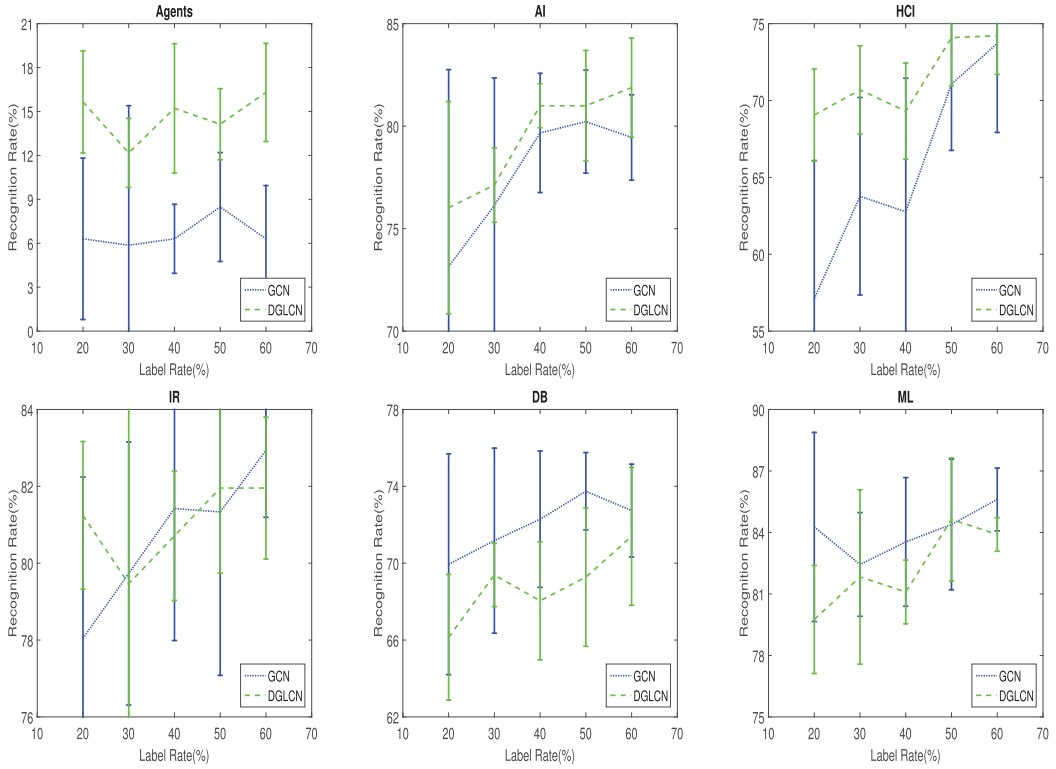

Fig. 4. Mean classification accuracy of single category for Citeseer dataset, including AI, Agents, ML, IR, HCI, DB. Each subfigure corresponds to a single category.

140 labeled data of Citeseer and Cora. From the experimental results, we can find that DGLCN outperforms more state-of-the-art methods especially when the fewer label rates are used. We see that DGLCN model acquires the superior classification performance for Citeseer and Cora. With the increase of the label rate, the mean classification accuracy of DGLCN and GCN all increases. It also proves that DGLCN can exploit the optimal structural information or the effectiveness of optimized graph structure learning method by our proposed method.

To evaluate the performance of DGLCN on a single class, we compare the proposed DGLCN with GCN in Figures 3–5. Figure 3 describes the mean classification accuracy (in percentages) of the Cora on the Reinforcement-learning, Theory, Genetic-algorithms, Probabilistic-methods, Rule-learning class, Neural-networks, Case-based. Figure 4 describes the mean classification accuracy (in percentages) on the Citeseer about six classes, such as AI, Agents, ML, IR, HCI, DB. Figure 5 is the average classification accuracy (in percentages) of Pubmed about three categories including the diabetes mellitus type 2, diabetes mellitus experimental, and diabetes mellitus type 1. The y-axis represents the average classification accuracy (in percentages) of each single class. From the mean classification accuracy of three datasets, we can know that DGLCN achieves a better performance than GCN in the majority of cases.

As described in Table 1 and Figure 2, we can acquire the following observations:

(1) Compared with GCN, DGLCN has shown its superior classification performance for Cora, Pubmed, and Citeseer datasets. As shown in Figure 2(a) and Figure 2(c), when 20%, 30%, 40%, 50%, and 60% labeled data are selected for training, our proposed DGLCN can obtain

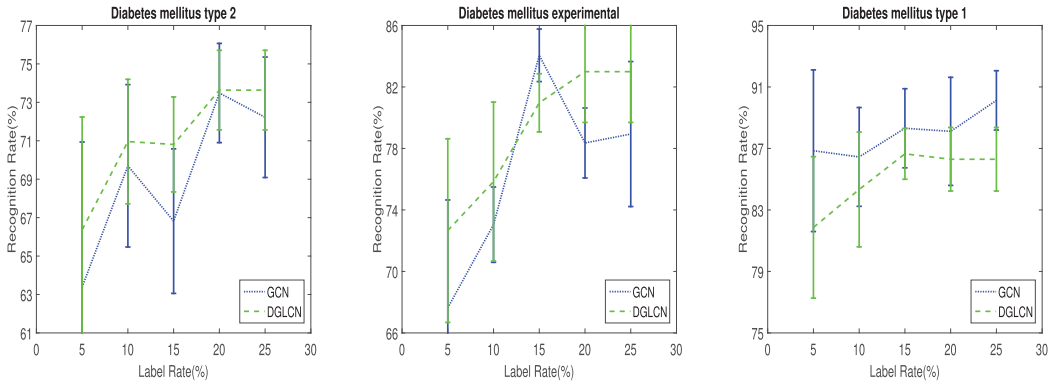

Fig. 5. Mean classification accuracy of single category for Pubmed dataset, including diabetes mellitus type 2, diabetes mellitus experimental, diabetes mellitus type 1. Each subfigure corresponds to a single category.

gains of 2.58%, 1.38%, 0.78%, 0.52%, and 0.7% on the Citeseer dataset, 4.28%, 4.4%, 3.08%, 2.22%, and 0.5% on the Cora dataset. For Pubmed dataset, DGLCN only has a slight gains than GCN. The main reason is that the optimized structural relations is quite similar to the constructed adjacent matrix according to the original data.

(2) In Table 1, DGLCN also outperforms several recent semi-supervised models especially when only tiny amounts of labeled samples are used. For example, compared with Chebyshev (K=2), Chebyshev (K=3), ManiReg, SemiEmb, MLP, GCN, HyperGCN, HpLapGCN, and GAT with 140 labeled training sample of the Cora dataset, DGLCN achieves 12%, 11.3%, 2.3%, 2.8%, 6.7%, 4.6%, 2.4%, 2%, and 4.8%.

5 CONCLUSION

In recent years, graph representation learning has become a hot research topic. In real practice, how to effectively encode the correlation of high-order data in graph structures is still a challenging problem. The most representative graph convolutional networks (GCN) extend convolutional neural networks to irregular graph structure data. Nevertheless, GCN only fuses the static structural information of original data during the learning process, which with little effort focuses on the optimization of sample structural information. To acquire richer structural information, we present a dynamic graph learning convolutional network (DGLCN) to tackle the above problem. DGLCN dynamically updates structural information apart from the first layer on each training iteration process until the model can get richer features or an outstanding classification performance. Extensive experimental results on citation network show that DGLCN model gets the higher classification performance compared to many state-of-the-art models.

REFERENCES

[1] Andrie Alexandrescu and Katrin Kirchhoff. 2007. Graph-based learning for phonetic classification. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*. 359–364.

[2] Walter Allegretto and Huang Yin Xi. 1998. A Picone's identity for the p-Laplacian and applications. *Nonlin. Anal.: Theor., Meth. Applic.* 32, 7 (1998), 819–830.

[3] James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 1993–2001.

[4] Jun Bai, Shiming Xiang, and Chunhong Pan. 2013. A graph-based classification method for hyperspectral images. *IEEE Trans. Geosci. Rem. Sens.* 51, 2 (2013), 803–817.

[5] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* 7, Nov. (2006), 2399–2434.

[6] Cecile Cabanes, A. Grouazel, K. von Schuckmann, M. Hamon, Victor Turpin, C. Coatanoan, F. Paris, S. Guinehut, C. Boone, N. Ferry, et al. 2013. The CORA dataset: Validation and diagnostics of in-situ ocean temperature and salinity measurements. *Ocean Sci.* 9, 1 (2013), 1–18.

[7] Yu Cao, Meng Fang, and Dacheng Tao. 2019. BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 357–362.

[8] Diane J. Cook and Lawrence B. Holder. 1993. Substructure discovery using minimum description length and background knowledge. *J. Artif. Intell. Res.* 1 (1993), 231–255.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 3844–3852.

[10] Sichao Fu, Weifeng Liu, Yicong Zhou, and Liqiang Nie. 2019. HpLapGCN: Hypergraph p-Laplacian graph convolutional networks. *Neurocomputing* 362 (2019), 166–174.

[11] Yue Gao and Qionghai Dai. 2014. Efficient view-based 3-D object retrieval via hypergraph learning. *Tsinghua Sci. Technol.* 19, 3 (2014), 250–256.

[12] Yue Gao, Meng Wang, Dacheng Tao, Rongrong Ji, and Qionghai Dai. 2012. 3-D object retrieval and recognition with hypergraph analysis. *IEEE Trans. Image Proc.* 21, 9 (2012), 4290–4303.

[13] Bo Geng, Dacheng Tao, Chao Xu, Linjun Yang, and Xian-Sheng Hua. 2012. Ensemble manifold regularization. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 6 (2012), 1227–1233.

[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 249–256.

[15] Yin Hong, Yanxia Liu, Suizhu Yang, Kaiwen Zhang, Aiqing Wen, and Jianjun Hu. 2020. Improving graph convolutional networks based on relation-aware attention for end-to-end relation extraction. *IEEE Access* 8 (2020), 51315–51323.

[16] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. 2016. Molecular graph convolutions: Moving beyond fingerprints. *J. Comput.-aided Molec. Des.* 30, 8 (2016), 595–608.

[17] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference of Learning Representation.*

[18] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 6th International Conference of Learning Representation.*

[19] Frank Lambert. 2010. Online community information seeking: The queries of three communities in Southwestern Ontario. *Inf. Proc. Manag.* 46, 3 (2010), 343–361.

[20] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence.*

[21] Weifeng Liu, Hongli Liu, Dapeng Tao, Yanjiang Wang, and Ke Lu. 2015. Multiview Hessian regularized logistic regression for action recognition. *Sig. Proc.* 110 (2015), 101–107.

[22] Weifeng Liu, Zheng-Jun Zha, Yanjiang Wang, Ke Lu, and Dacheng Tao. 2016. p-Laplacian regularized sparse coding for human activity recognition. *IEEE Trans. Industr. Electron.* 63, 8 (2016), 5120–5129.

[23] Xiao Liu, Jun Shi, and Congzhi Wang. 2015. Hessian regularization based non-negative matrix factorization for gene expression data clustering. In *Proceedings of the 37th International Conference of the IEEE Engineering in Medicine and Biology Society*. 4130–4133.

[24] Yong Luo, Dacheng Tao, Bo Geng, Chao Xu, and Stephen J. Maybank. 2013. Manifold regularized multitask learning for semi-supervised multilabel image classification. *IEEE Trans. Image Proc.* 22, 2 (2013), 523–536.

[25] Xueqi Ma, Weifeng Liu, Shuying Li, Dapeng Tao, and Yicong Zhou. 2019. Hypergraph p-Laplacian regularization for remotely sensed image recognition. *IEEE Trans. Geosci. Rem. Sens.* 57, 3 (2019), 1585–1595.

[26] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018. DeepInf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2110–2119.

[27] Rafael Rodriguez-Puente and Manuel S. Lazo-Cortes. 2012. Graph-reduction algorithm for finding shortest path in geographic information systems. *IEEE Lat. Amer. Trans.* 10, 6 (2012), 2201–2208.

[28] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Mag.* 29, 3 (2008), 93–93.

[29] Andrea Vedaldi and Karel Lenc. 2015. MatConvNet: Convolutional neural networks for MATLAB. In *Proceedings of the 23rd International Conference on Multimedia*. 689–692.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 7th International Conference of Learning Representation.*

[31] Yang Wang. 2020. Survey on deep multi-modal data analytics: Collaboration, rivalry and fusion. *ACM Trans. Multim. Comput. Commun. Applic.* (2020). DOI: 10.1145/1122445.1122456

[32] Yan Wang, Hao Zhang, Yaxin Li, Deyun Wang, YanLin Ma, Tong Zhou, and Jianguo Lu. 2016. A data cleaning method for Citeseer dataset. In *Proceedings of the International Conference on Web Information Systems Engineering*. 35–49.
[33] Yang Wang, Wenjie Zhang, Lin Wu, Xuemin Lin, Meng Fang, and Shirui Pan. 2016. Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2153–2159.
[34] Yang Wang, Wenjie Zhang, Lin Wu, Xuemin Lin, and Xiang Zhao. 2017. Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 1 (2017), 57–70.
[35] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. 2012. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*. Springer, 639–655.
[36] Jia Wu, Shirui Pan, Xingquan Zhu, Chengqi Zhang, and S. Yu Philip. 2018. Multiple structure-view learning for graph classification. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 7 (2018), 3236–3251.
[37] Lin Wu, Yang Wang, Junbin Gao, and Xue Li. 2019. Where-and-when to look: Deep Siamese attention networks for video-based person re-identification. *IEEE Trans. Multim.* 21, 6 (2019), 1412–1424.
[38] Lin Wu, Yang Wang, Junbin Gao, and Dacheng Tao. 2021. Deep co-attention based comparators for relative representation learning in person re-identification. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 2 (2021), 722–735.
[39] Lin Wu, Yang Wang, and Ling Shao. 2019. Cycle-consistent deep generative hashing for cross-modal retrieval. *IEEE Trans. Image Proc.* 28, 4 (2019), 1602–1612.
[40] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).
[41] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the International Conference on Machine Learning*.
[42] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Anand Louis, and Partha Talukdar. 2018. HyperGCN: Hypergraph convolutional networks for semi-supervised classification. In *Proceedings of the International Conference on Multimedia and Expo*.
[43] Yunyun Yang and Gang Xie. 2016. Efficient identification of node importance in social networks. *Inf. Proc. Manag.* 52, 5 (2016), 911–922.
[44] Sławomir Zadrożny, Janusz Kacprzyk, and Guy De Tré. 2012. Bipolar queries in textual information retrieval: A new perspective. *Inf. Proc. Manag.* 48, 3 (2012), 390–398.
[45] Wenjie Zhang, Lin Wu, Yang Wang, and Xuemin Lin. 2017. Effective multi-query expansions: Collaborative deep networks for robust landmark retrieval. *IEEE Trans. Image Proc.* 26, 3 (2017), 1393–1404.
[46] Lefei Zhang, Qian Zhang, Liangpei Zhang, Dacheng Tao, Xin Huang, and Bo Du. 2015. Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding. *Pattern Recog.* 48, 10 (2015), 3102–3112.
[47] Yongsheng Zhang, Tsuyoshi Yamamoto, and Yoshinori Dobashi. 2016. Multi-scale object retrieval via learning on graph from multimodal data. *Neurocomputing* 207 (2016), 684–692.
[48] Zhao Zhang, Sun Yulin, Yang Wang, Zheng Zhang, and Meng Wang. 2020. Twin-incoherent self-expressive locality-adaptive latent dictionary pair learning for classification. *IEEE Trans. Neural Netw. Learn. Syst.* (2020). DOI: 10.1109/TNNLS.2020.2979748
[49] Nengwen Zhao, Lefei Zhang, Bo Du, Qian Zhang, Jane You, and Dacheng Tao. 2017. Robust dual clustering with adaptive manifold regularization. *IEEE Trans. Knowl. Data Eng.* 29, 11 (2017), 2498–2509.
[50] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* (2018).
[51] Xiang Zhou, Fumin Shen, Li Liu, Wei Liu, Liqiang Nie, Yang Yang, and Heng Tao Shen. 2020. Graph convolutional network hashing. *IEEE Trans. Syst., Man Cyber.* 50, 4 (2020), 1460–1472.

Received February 2020; revised July 2020; accepted July 2020