Pattern Recognition 118 (2021) 108039

Contents lists available at ScienceDirect

Pattern Recognition

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

Deep graph learning for semi-supervised classification

Guangfeng Lin*, Xiaobing Kang, Kaiyang Liao, Fan Zhao, Yajun Chen

Information Science Department, Xi'an University of Technology, 5 South Jinhua Road, Xi'an,Shaanxi Province 710048, PR China

ARTICLE INFO

Article history:
Received 17 June 2020
Revised 19 March 2021
Accepted 10 May 2021
Available online 19 May 2021

Keywords:
Graph learning
Graph convolutional networks
Semi-supervised classification

ABSTRACT

Graph learning (GL) can dynamically capture the distribution structure (graph structure) of data based on graph convolutional networks (GCN), and the learning quality of the graph structure directly influences GCN for semi-supervised classification. Most existing methods combine the computational layer and the related losses into GCN for exploring the global graph (measuring graph structure from all data samples) or local graph (measuring graph structure from local data samples). The global graph emphasizes the whole structure description of the inter-class data, while the local graph tends to the neighborhood structure representation of the intra-class data. However, it is difficult to simultaneously balance these learning process graphs for semi-supervised classification because of the interdependence of these graphs. To simulate the interdependence, deep graph learning (DGL) is proposed to find a better graph representation for semi-supervised classification. DGL can not only learn the global structure by the previous layer metric computation updating, but also mine the local structure by next layer local weight reassignment. Furthermore, DGL can fuse the different structures by dynamically encoding the interdependence of these structures, and deeply mine the relationship of the different structures by hierarchical progressive learning to improve the performance of semi-supervised classification. Experiments demonstrate that the DGL outperforms state-of-the-art methods on three benchmark datasets (Citeseer, Cora, and Pubmed) for citation networks and two benchmark datasets (MNIST and Cifar10) for images.

1. Introduction

Graph ($G(V, E)$, in which V is the vertex set for describing a dataset and E is the edge set for representing the relationship set between data) can capture the relationship of data distribution based on metric methods (for example, Euclidean distance, Cosine distance or Kullback-Leibler divergence). As a metric representation, graph plays a vital role in pattern recognition. In particular, recent graph convolutional networks (GCN) have achieved promising results for many applications, for example, long short-term memory networks and GCN can capture graph structures with time changes for human activities [1]; semantic GCN can learn node relationships to represent non-explicit graphs for human pose regression [2]; GCN can exploit proposal-proposal relations for temporal action localization [3] and model action unit relations for facial detection [4]; text GCN can build text graphs of many corpora on word co-occurrences and relations for classification [5]; and first-order spectral graph convolution [6], hyper-node aggregation and coarsened graph refinement [7], higher-order graph interactions [8], spectral-spatial graph convolutional networks [9] and multi-graph information mining [10] are used for node classification. However, the graph structure is fixed in the evolution process of the GCN methods, which limits GCN for applying the lacking graph structure. Furthermore, the fixed graph structure is usually measured by one metric method, which cannot better fit the data distribution. Therefore, graph learning (GL) based on the optimal graph structure [11], matching networks [12], multiple graph labeling [13], deep iterative and adaptive learning [14], and feature metric optimization [15] are presented for dynamically mining the graph structure of data.

Graph learning faces a key question, which is the structural relationship learning of data distributions. Existing methods focus on how to update the graph structure with the metric constraint to optimize the object function [16,17] or neural networks [11,12]. The metric constraint is usually defined in two ways. One way is similarity metric learning, which is a global graph structure that learns from all data samples. This method often focuses on the inter-class difference representation. The other way specifies different weights to different data in its neighborhood (for example, graph attention networks (GAT) [18]) for capturing the local graph structure, which tends to the intra-class difference description. In information cognition, the global and local structures can jointly enhance the data representation from the different perspectives. Global and local graphs complement each other for classification.

* Corresponding author
E-mail address: lgf78103@xaut.edu.cn (G. Lin).

https://doi.org/10.1016/j.patcog.2021.108039
0031-3203/

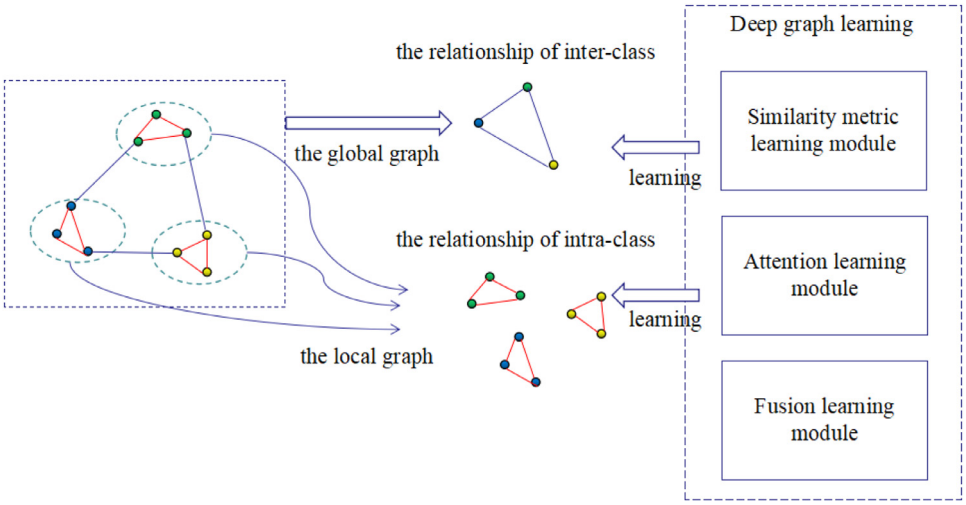

Fig. 1. Illustration of deep graph learning for mining global and local graphs.

However, existing methods ignore the joint effect of these graphs and the relationship between the global and local graphs based on GL for semi-supervised classification. Therefore, DGL is proposed to jointly consider these graph structures for semi-supervised classification.

Our main contributions include two points.

- One is constructing deep graph learning networks to dynamically capture the global graph by similarity metric learning and the local graph by attention learning. Compared with existing methods, the difference focuses on jointly considering the different graphs to further find the distribution structure of different data.
- Another is fusing the global and local graph by hierarchical progressive learning for semi-supervised classification. In contrast to existing methods, the difference is the dynamic mining of the relationship of these graphs to better balance the tendentious contradiction of the different graphs between inter-class and intra-class. Fig. 1 shows the difference between the global and local graphs and the modules of the DGL.

2. Related works

Graph learning attempts to automatically construct graph structures from data. Compared with fixed similarity metrics, the difference of GL can dynamically assign the neighbor of each data point and automatically compute the weight between data points. Therefore, GL can obtain better accuracy than the fixed graph description by similarity metrics [19].

According to the different learning frameworks, the recent GL methods can be divided into two categories: non-neural networks and neural networks.

Non-neural network methods attempt to build the optimization function based on the graph generation hypothesis. For example, in terms of the completeness hypothesis, similarity relationships are learned in kernel space [20], the subspace structure is recovered by low-rank representation [21], and self-expressiveness regards the linear coefficient matrix between data as the graph matrix for impressive performance [22] in clustering and semi-supervised learning. According to the Laplacian graph spectrum, graph learning based on spectral constraints [23] complements the relationship of data by incorporating prior structural knowledge. Structured graph learning with single kernel (SGSK) and structured graph learning with multiple kernel (SGMK) [24] simultaneously consider self-expressiveness to capture the global structure and adaptive neighbor approach to find the local structure. On the basis of sparse sampling theory, consistent similarity matrix learning with sparse structure from multiple views [25], the adaptive parameter adjustment strategy during sparse graph construction [26], and edge constrained sparse representation [27] can capture few graph connections by adjusting the sparsity parameter for improving the classification performance. In terms of increment generation, the traditional least squares method is used to construct a dynamic graph of the increment data [28]. The superiority of these methods focuses on the relevance between graph generation and constraints, and parameterizes graph generation processing to dynamically control graph learning. Because model construction is usually fixed by a specific function, graph structure information from raw data is difficult to mine for semi-supervised classification by iterative boosts and the end-to-end paradigm.

Neural network approaches include two aspects: optimal graph structure mining with node representation evolution based on the fixed vertexes for modeling the vertex structure [11] and evolution model construction between the fixed graph structures based on the varied vertexes for capturing the sequence structure of the vertex group [1]. Our work focuses on the former aspect, which contains two types of methods for mining graph structures. The first type of method is the aggregation of nodes and edge information for updating the weight between nodes layer by layer. For example, hierarchical graph convolutional network (H-GCN) [7] repeatedly aggregates similar nodes to hyper-nodes, and combines one- or two-hop neighborhood information to enlarge the receptive field of each node to encode graph structure information; edge-labeling graph neural network (EGNN) [29,30] updates the graph weight by iteratively aggregating the node representation and the edge labels with direct exploitation of both intra-cluster similarity and inter-cluster dissimilarity. The second type of method is the similarity metric of pairwise nodes in some layers. For instance, graph learning-convolutional network (GLCN) [11] optimizes graph structure by learning the transformation relationship of feature differences; dimension-wise separable graph convolution (DSGC) [31] uses the relationship among node attributes to complement node relations for representation learning by the covariance metric; graph learning neural networks (GLNNs) [32] iteratively explore the optimization of graphs from both data and tasks by a graph Laplacian regularizer; relation-guided representation learning (RGRL) [33] can construct deep auto-encoders network model by preserving the local structure relationship between sam-

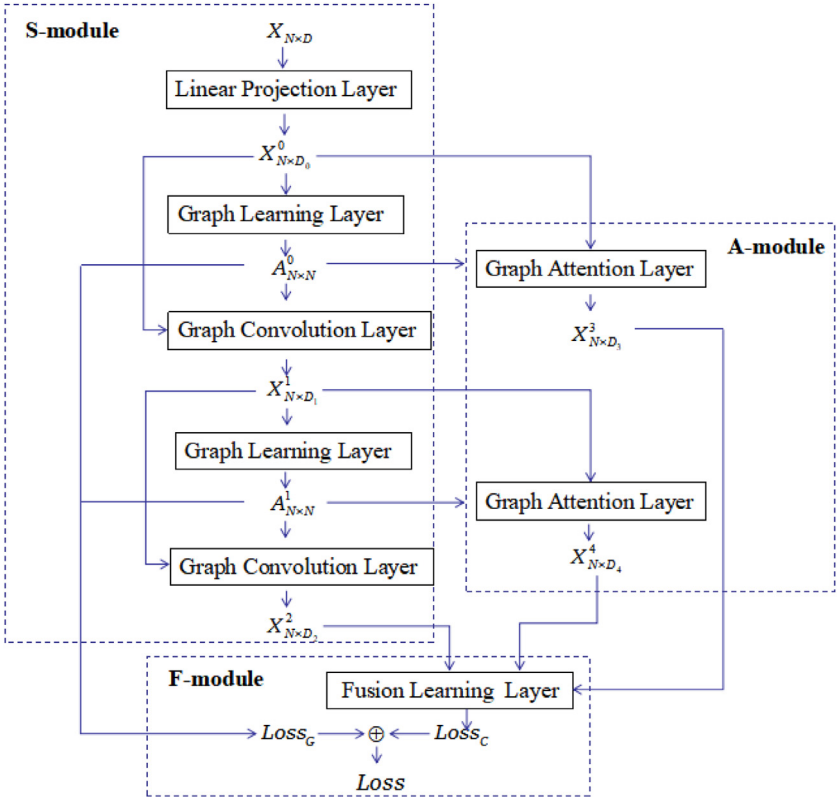

Fig. 2. The network frameworks of deep graph learning contain the similarity metric learning module (S-module), attention learning module (A-module) and fusion learning module (F-module). $X_{N\times D}$, $X^0_{N\times D_0}$, $X^1_{N\times D_1}$, $X^2_{N\times D_2}$, $X^3_{N\times D_3}$ and $X^4_{N\times D_4}$ are node representations of each layer (the superscript of the node representation is the serial number of layers, and the subscript of the node representation shows the dimension space of the node representation.); $A^0_{N\times N}$ and $A^1_{N\times N}$ are the adjacent matrixes of the different layers (the superscript of the adjacent matrix is the serial number of layers, and the subscript of the adjacent matrix shows the dimension space of the adjacent matrix.); $Loss_G$ is the graph learning loss; $Loss_C$ is the classification loss; *Loss* is the total loss of the whole network.

ple; deep iterative and adaptive learning for graph neural networks (DIAL-GNN) [14] deal with the graph structure learning problem as a dynamic cosine similarity metric learning problem. These methods mostly consider the global structure from all data samples in the second type of method or the local structure from neighboring data in the first type of method. However, the hierarchical progressive relationship between the global and local graphs is ignored.

From the above mentions, the methods based on non-neural networks show a better causal relationship between the graph structure and the specific optimization function, while the methods based on neural networks demonstrate a stronger learning ability between the graph structure and the uncertain optimization networks. This makes the latter more suitable for further mining the graph structure. Moreover, the similarity metric of pairwise nodes in a graph is usually directly connected with raw data to easily fit its distribution. Therefore, our proposed method focuses on graph learning based on GCN to find the hierarchical progressive relationship between the global and local graphs.

3. Deep graph learning

Deep graph learning (DGL) includes three modules, which are the similarity metric learning module (S-module), attention learning module (A-module) and fusion learning module (F-module) in Fig. 2. The similarity metric learning module implements graph structure computation for dynamically updating global structure relationships based on raw data or transformed data. The attention learning module reassigns the weight of the neighbor of each data point for finding the significant local structure based on the global structure. The fusion learning module integrates node representation based on the different graph structures for semi-supervised classification. Although the role of the graph learning module in S-module and the graph attention module is very similar from the macroscopic perspective. However, from the microscopic perspective, there are two different points between the graph learning module and graph attention model. Specially, one is that the graph learning module hierarchically learns the global graph structure from the whole data, while the graph attention model trends to hierarchically capture the local graph structure based on the global graph structure by confirming the graph node neighborhood. Another is that the graph learning module computes graph structure by differential modulus metric, whereas the graph attention model further combine graph node aggregation information into local graph structure learning. Therefore, these modules in the new network architecture can interactively be considered for constructing the discriminate representation.

3.1. Similarity metric learning module

In this section, we can construct similarity metric learning global structure relationship by computing the weight of the pairwise nodes. There are two points for this purpose. One is that the global structure relationship can be calculated by similarity metric between any pair-wise node representations, so the structure relationship between any node and other nodes reflects the global distribution of all nodes. Two is that the global structure is changing with the different node representations in the different layers, so the different scale structures in the various layers also can complement the global distribution of all nodes each other.

Given data matrix $X \in R^{N\times D}$ (N is the sample number of data, and D is the dimension of each data), let X be the node representation of graph G. We expect to learn G from X for semi-supervised

classification. In this module, there are three types of layers for the stacking network structure.

The first type of layer is a linear projection layer for reducing the dimension of the raw data feature. Because the dimension of raw data often leads to higher computational complexity, the linear transformation of the reduction dimension is implemented in this layer.

$$X^0_{N\times D_0} = XP, \tag{1}$$

where $P \in R^{D\times D_0}$ is the linear transformation matrix, and $X^0_{N\times D_0}$ represents the output of the linear projection layer.

The second type of layer is the graph learning layer for computing the weight of the pairwise nodes. The adjacent relationship $A^l_{N\times N}(i,j)$(i and j are the subscripts of the different node representations in Graph G; l represents the serial number of the layer) can describe this relationship weight and is defined as follows.

$$A^l_{N\times N}(i,j) = \frac{A(i,j)\exp(ReLU((\alpha^l)^T|x^l_i - x^l_j|))}{\sum_j^N A(i,j)\exp(ReLU((\alpha^l)^T|x^l_i - x^l_j|))}, \tag{2}$$

where A is the normalized adjacent matrix from the initial data source. If A is not available, $A(i,j)=1$. $ReLU(f)=\max(0,f)$ (f is any variable or matrix) can assure the nonnegativity of $A^l_{N\times N}(i,j)$. $x^l_i \in R^{D_l\times 1}$ and $x^l_j \in R^{D_l\times 1}$ are the different row transposes of input $X^l_{N\times D_l}$ in the current layer. Eq. 2 normalizes $A^l_{N\times N}$ corresponding to its row. $\alpha^l \in R^{D_l\times 1}$ is the weight parameter vector for measuring the significance of the relationship between nodes. Graph learning mainly trains the network for learning $\alpha^l(l=\{0,1\})$.

The third type of layer is the graph convolution layer for propagating information based on the graph. According to GCN [6], we can define the graph convolution layer as follows:

$$X^{l+1}_{N\times D_{l+1}} = ReLU(\hat{D}^{l\,-1/2}\hat{A}^l\hat{D}^{l\,-1/2}X^l_{N\times D_l}W^l), \tag{3}$$

where $\hat{A}^l = I_{N\times N} + A^l_{N\times N}$($I_{N\times N} \in R^{N\times N}$ is the identity matrix); $\hat{D}^l(i,i) = \sum_j A^l_{N\times N}(i,j)$; $W^l \in R^{D^l\times D^{l+1}}$ is the trainable weight matrix of the current layer.

The similarity metric learning module based on the three types of layers includes one linear projection layer, two graph learning layers and a graph convolution layer from input to output. In particular, two times stack of the graph learning layer and graph convolution layer can construct a deep network for mining the global graph structure of the different scale node representations.

3.2. Attention learning module

In the whole network construction, the specific computation of the global graph structure is based on the similarity relationship between any pair-wise node representations and includes the local structure information of the different node neighborhoods. However, this local structure information only comes from the pair-wise relevance between the current node and all other nodes but weakens the importance discrimination of the node in the neighborhood of the current node. Therefore, we construct an attention learning module by aggregating the neighbor information to further capture the local structure based on the sparse constraint neighborhood of the global structure (we call this process hierarchical progressive learning). The original GAT [18] can only process the binary weight of pair-wise node representation. For example, an attention mechanism is built based on a node's neighborhood weighted by a binary value. However, the weight of the learned graph is real-value, which helps to confirm the node's neighborhood by incorporating the sparse constraints of the global graph structure. Therefore, the operation of the attention mechanism is defined as follows.

$$X^{l+1}_{N\times D_{l+1}} = ReLU(\beta^l X^l_{N\times D_l}W^l), \tag{4}$$

where $\beta^l \in R^{N\times N}$ is the attention coefficient matrix, in which any entry $\beta^l(i,j)$ is directly relevant to $X^l_{N\times D_l}(i,:)$, $X^l_{N\times D_l}(j,:)$ and $A^l_{N\times N}(i,j)$. Therefore, we define $\beta^l(i,j)$ by information aggregation based on the graph as follows.

$$\hat{\beta}^l(i,j) = \exp(ReLU(\gamma^T[X^l_{N\times D_l}(i,:)W^l \| X^l_{N\times D_l}(j,:)W^l]))A^l_{N\times N}(i,j), \tag{5}$$

$$\beta^l(i,j) = \hat{\beta}^l(i,j)/\textstyle\sum_k^N \hat{\beta}^l(i,k), \tag{6}$$

where $\|$ is the concatenation operator for transforming into the column vector; and $\gamma \in R^{2D_{l+1}\times 1}$ is the aggregation weight, which is shared by the dimension of all pair-wise node aggregations.

Attention mechanism is a visual processing method according to human vision characteristic. Under the suitable guide, this mechanism can capture the discriminate information for the specific visual task. Therefore, we construct attention module to deal with the updating attention coefficient (local graph structure relationship representation) based on the pair-wise nodes aggregation of the normalized adjacent matrix (global graph structure relationship representation). The attention module mainly considers two aspects. One is that attention coefficient can be projected into node representation by Eq. (4). Another is that attention coefficient is learned by the pair-wise nodes and their structure relationship by Eq. (5). Therefore, the attention module not only can demonstrate layer by layer progressive idea from global to local graph structure learning, but also can autonomously adapt node neighborhood changing pattern based on the sparse constraint loss (Eq. (9)) of the global structure.

In the attention learning module, we handle the different scale information from the global graph structure by two graph attention layers to further mine the local graph structure, which is a credible basis for the intra-class description.

3.3. Fusion learning module

The fusion learning module includes two parts: the fusion learning layer for the different node representations and the loss function for network training propagation.

The first part is the fusion learning layer for processing the different dimension questions of the node representation or the weight balance issue from the different modules (similarity metric learning module or attention learning module). In Fig. 2, the inputs of this module have $X^2_{N\times D_2}$ graph convolution layer output and $X^3_{N\times D_3}$ and $X^4_{N\times D_4}$ different graph attention layer output. Because this network deals with classification, we make the output dimension of the different modules uniform($D_2=D_3=D_4=C$, C is the class number). Therefore, we define the fusion learning layer as follows:

$$Z = Softmax(\eta_1 X^2_{N\times D_2} + \eta_2 X^3_{N\times D_3} + \eta_3 X^4_{N\times D_4}), \tag{7}$$

where $\eta = [\eta_1, \eta_2, \eta_3]$ is the fusion coefficient vector, which encodes the importance of the different node representations.

The second part is the loss function definition, which determines the tendency of network learning. The total loss $Loss$ contains the classification loss $Loss_C$ and the graph loss $Loss_G$.

In semi-supervised classification, we construct classification loss based on the labeled data by cross-entropy loss for evaluating the error between the predicted label Z and the real label Y. Therefore, $Loss_C$ is defined as follows:

$$Loss_C = -\textstyle\sum_{k\in S}\sum_{c=1}^{C} Y_{kc} lnZ_{kc}, \tag{8}$$

where S is the labeled data set, Y_{kc} represents the kth label data belonging to the cth class, and Z_{kc} shows the kth label data predicted as the cth class.

Table 1
Datasets statistics and the extracted features in experiments.

Datasets	Classes number	Training Number	Validating Number	Testing Number	Total number of images	Feature dimension	Initial graph
Generated data	4	4 ∼ 16	400	3596 ∼ 3584	4000	200	No
Cora	7	140	500	1000	2708	1433	Yes
Citeseer	6	120	500	1000	3327	3703	Yes
Pubmed	3	59	500	1000	19717	500	Yes
Cifar10	10	1000 ∼ 8000	1000	8000 ∼ 1000	50000	128	No
MNIST	10	1000 ∼ 8000	1000	8000 ∼ 1000	60000	784	No

In graph learning, we compute the adjacent matrix $A^0_{N\times N}$ and $A^1_{N\times N}$ for describing the graph of the different scales. To constrain the properties (sparsity and consistency) of these adjacent matrixes, we define the graph loss $Loss_G$ as follows.

$$\begin{aligned} Loss_G = \quad & \lambda_1(X^T_{N\times D}(I - A^0_{N\times N})X_{N\times D} + X^T_{N\times D}(I - A^1_{N\times N})X_{N\times D}) \\ & +\lambda_2(\|A^0_{N\times N}\|^2_F + \|A^1_{N\times N}\|^2_F) + \lambda_3\|A^0_{N\times N} - A^1_{N\times N}\|^2_F, \end{aligned} \tag{9}$$

where the first term can enforce $X_{N\times D}$ matching with the topology of the graph by the graph Laplacian regularizer, the second term can guarantee the sparsity of these adjacent matrixes, and the third term can assure the consistency between these adjacent matrixes.

Therefore, the total loss $Loss$ is the sum of $Loss_C$ and $Loss_G$.

$$Loss = Loss_C + Loss_G, \tag{10}$$

Although classification loss and graph loss seem to be processed by equal way in Eq. (10), graph loss is computed by the weight way in Eq. (9). Therefore, the total loss is calculated by the weight way. These weights can be decided with cross validation by prior. Local or global structures are also dealed with the weights, which can be learned by attention mechanism or graph learning. These weights are the different parameters of networks. The proposed DGL includes many parameters, which have two types. One kind can learned by end to end paradigm, for instance parameters in Eqs. (1)-(8). Another kind can be decided by cross validation, for example hyper-parameters in Eq. (9). The key point is how to construct deep network architecture to capture the complementary information between local or global structures (these structures can be described by weights between nodes) for representation and classification. Therefore, the proposed DGL designs the deep networks to mine the local or global structures information by the different weights or parameters learning for improving the classification.

4. Experiment

4.1. Datasets

To evaluate the proposed DGL method, we carry out experiments on one generated dataset and six benchmark datasets, which include three paper-citation network datasets (Cora, Citeseer and Pubmed [34]) and two image datasets(MNIST [35] and Cifar10 [36]).

The synthesized dataset contains 4 classes, each of which has 1000 samples, and includes 4000 samples. These data are randomly synthesized. In the experiment, each class of samples is divided into four groups, which are 1/100/899, 2/100/898, 3/100/897 and 4/100/896 for the training/validation/testing sets. Table 1 shows its details.

The Cora dataset includes 7 classes that have 2708 grouped publications as nodes represented by a one-hot vector in terms of the presence or absence state of a word in the learned directory and their link relationship graph. The Citeseer dataset contains 6 classes that involve 3327 scientific papers described in the same way as the Cora dataset and their undirected graph. The Pubmed dataset has 3 classes that include 19,717 diabetes-related publications indicated by the term frequency-inverse document frequency (TF-IDF) [37] and their relevance graph. In these datasets, experiments follow the configuration of the previous work [6]. We selected 500 samples for validation and 1000 samples for testing. Table 1 shows the specific information of these datasets.

The Cifar10 dataset has 10 classes that consist of 50,000 natural images [36]. The size of each RGB image is 32×32. We select 10,000 images (1,000 images for each class) for evaluating the proposed DGL. To represent each image, we use ResNet-20 [38] to extract features. The MNIST dataset contains 10 classes of handwritten digits. We also select 10,000 images (1,000 images for each class) for assessing the proposed DGL. Each image feature is a 784-dimensional vector generated by a gray image. Table 1 demonstrates the statistics of these datasets.

4.2. Experimental configuration

In the experiments, we set $D_0 = 70$, $D_1 = 30$ and $D_2 = D_3 = D_4$, which is equal to the number of classes. The maximum training episodes of the proposed DGL are 200. The parameters λ_1, λ_2 and λ_3 are set as 0.1, 0.01 and 0.001, respectively. In the Cifar10 and MNIST datasets, we select 8 groups of data for the different training-validating-testing sets (1,000-1,000-8,000, 2,000-1,000-7,000, 3,000-1,000-6,000, 4,000-1,000-5,000, 5,000-1,000-4,000, 6,000-1,000-3,000, 7,000-1,000-2,000 and 8,000-1,000-1,000). In the different datasets, the validation set is mainly used for optimizing hyper-parameters, which include the dropout rate for all layers, the number of hidden units and the learning rate.

4.3. Generated data experiment

To observe the generated data, we reduce multi-dimensional data to two dimensions for visualizing data by t-SNE [39]. Fig. 3 shows the distribution of the generated data in two dimensions and the experimental results of four methods: the proposed DGL, GLSGCN, GLGCN [11] and GLGAT. GLSGCN and GLGAT are constructed to extend the graph learning method in Section 4.6. Although few data are labeled, DGL can still learn the structure distribution of the data to obtain promising results. Therefore, we conduct the following experiments to further evaluate the proposed DGL in real datasets.

4.4. Comparison with baseline approaches

In this section, we implement the proposed DGL and the baseline methods, which are GCN [6], GAT [18], simplifying graph convolutional networks (SGCN) [40] and GLGCN [11]. GCN can construct the basic architecture of graph representations and classification models by the localized first-order approximation of spectral graph convolutions. GAT can learn the different weights to different nodes in a neighborhood to find the attention mechanism of local data. SGCN can eliminate the redundant complexity and

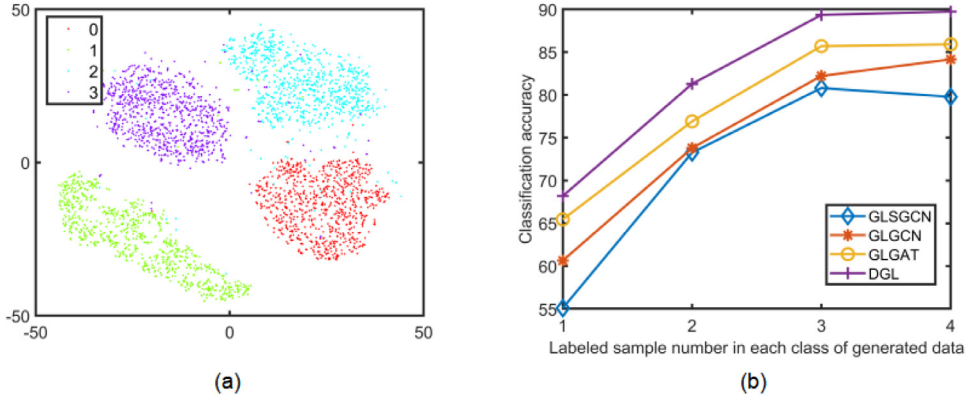

Fig. 3. The structure distribution of the two-dimensional generated data in (a) and the contrast experiment of the graph learning method in (b).

Table 2
Comparison of the proposed DGL method with baseline methods (GCN, GAT, SGCN and GLGCN) for semi-supervised classification, average per-class accuracy (%) is reported based on the same data configurations in the different datasets. The results between parentheses come from the different studies. All methods use the initial graph for the computing model.

Method	Cora	Citeseer	Pubmed
GCN [6]	81.1 ± 0.4(82.9)	71.0 ± 0.2(70.9)	78.9 ± 0.5(77.9)
GAT [18]	81.4 ± 0.8(83.2)	71.8 ± 0.3(71.0)	78.1 ± 0.4(78.0)
SGCN [40]	82.3 ± 0.5(81.0)	71.4 ± 0.3(71.9)	78.3 ± 0.2(78.9)
GLGCN [11]	82.2 ± 0.7(85.5)	72.0 ± 0.2(72.0)	78.3 ± 0.1(78.3)
DGL	**84.8** ± 0.7	**74.2** ± 0.5	**80.2** ± 0.2

Table 3
Comparison of the proposed DGL method with state-of-the-art methods (H-GCN, GLNNs, DIAL-GNN and GLGCN) for semi-supervised classification; average per-class accuracy (%) is reported based on the same data configurations in the different datasets. The results in parentheses are from the different literatures. All methods use the initial graph for the computing model.

Method	Cora	Citeseer	Pubmed
H-GCN [7]	(84.5 ± 0.5)	(72.8 ± 0.5)	(79.8 ± 0.4)
GLNNs [32]	(83.4)	(72.4)	(76.7)
DIAL-GNN [14]	(84.5 ± 0.3)	(74.1 ± 0.2)	*Null*
GLGCN [11]	**(85.5)**	(72.0)	(78.3)
DGL	84.8 ± 0.7	**74.2** ± 0.5	**80.2** ± 0.2

Table 4
Comparison of the proposed DGL method with the related graph learning methods (GLGCN, GLSGCN, GLGAT and DGL) for semi-supervised classification, average per-class accuracy (%) is reported based on the same data configurations in the citation datasets (Cora, Citeseer and Pubmed). All methods do not use the initial graph for the computing model.

Method	Cora	Citeseer	Pubmed
GLSGCN	55.9 ± 0.6	49.6 ± 0.3	74.8 ± 0.5
GLGCN [11]	60.1 ± 0.3	64.6 ± 0.2	73.3 ± 0.5
GLGAT	63.1 ± 0.4	65.5 ± 0.2	75.3 ± 0.2
DGL	**65.3** ± 0.3	**68.9** ± 0.4	**76.9** ± 0.5

computations of GCN by removing nonlinear units and collapsing operations between the different layers. GLGCN can combine graph learning and graph convolution to optimize the global graph structure. Compared with these methods, DGL can not only mine the global graph structure by the different scale graph learning layers but also capture the local graph structure by the different scale graph attention layers. Furthermore, DGL can integrate the node representation from the different graph structures by a fusion learning layer. Table 2 shows that DGL has the best performance of these methods. The experimental results of GCN, GAT and GLGCN in parentheses are from the literature [11], while the results of SGCN stem from the literature [40]. Specifically, DGL outperforms the existing GCN, GAT and GLGCN with the least margins 1.2% on the Cora dataset, 1.6% on the Citeseer dataset, and 0.6% on the Pubmed dataset. This shows that DGL can further optimize the known graph structure for semi-supervised classification by deep graph learning.

4.5. Compared with state-of-the-arts

Graph learning with neural networks shows promising results for semi-supervised classification. In Section 2, we summarize the graph learning methods based on neural networks and find the bias of the global graph structure or the local graph structure in existing methods. Therefore, we try to construct a new graph learning method based on a neural network to further mine the graph structure and balance the bias of these methods. We compare the proposed DGL with H-GCN [7], GLNNs [32], DIAL-GNN [14] and GLGCN [11]. The difference between these methods is detailed in Section 2. Table 3 shows the best performance of the different methods; for example, GLGCN can improve by 0.7% on average at least on the Cora dataset, and DGL can improve by 0.1% on average on the Citeseer dataset and 0.4% on the Pubmed dataset. Therefore, these methods can obtain the approximate performance in these datasets. To further contrast the difference between GLGCN and the proposed DGL, we carry out the graph learning experiments in the following section.

4.6. Compared with the extended graph learning methods

In this section, we involve four methods: GLGCN [11], the proposed DGL and two extended methods (graph learning based on SGCN (GLSGCN) and graph learning based on GAT (GLGAT)). We use the basic idea of GLGCN to construct GLSGCN and GLGAT. **GLSGCN** includes a linear projection layer, which reduces the dimension of the original data to 70, a graph learning layer and the following layers that are the same as SGCN [40]. **GLGAT** also adds a linear projection layer to reduce the dimension of the data, a graph learning layer and the other layers that have the same configuration as GAT [18]. In these experiments, all citation datasets do not use the initial graph, and the graph structure can be learned from the original data by the different methods. For instance, GLSGCN and GLGCN tend to capture the global structure, GLGAT shallowly mines the global and local structure, and the proposed DGL can deeply consider these structures for semi-supervised classification.

Table 5
Comparison of the proposed DGL method with the related graph learning methods (GLGCN, GLSGCN, GLGAT and DGL) for semi-supervised classification, average per-class accuracy (%) is reported based on the different data training/validation/testing in the MNIST image datasets. The initial graph for computing model is not available.

Method	MNIST 1000/1000/8000	MNIST 2000/1000/7000	MNIST 3000/1000/6000	MNIST 4000/1000/5000
GLSGCN	37.7 ± 0.2	38.7 ± 0.4	39.5 ± 0.1	39.6 ± 0.2
GLGCN [11]	84.9 ± 0.4	85.9 ± 0.2	85.2 ± 0.3	88.0 ± 0.2
GLGAT	86.3 ± 0.5	89.9 ± 0.2	89.7 ± 0.4	89.2 ± 0.6
DGL	**89.1** ± 0.6	**91.4** ± 0.2	**91.1** ± 0.3	**92.4** ± 0.5
Method	MNIST 5000/1000/4000	MNIST 6000/1000/3000	MNIST 7000/1000/2000	MNIST 8000/1000/1000
GLSGCN	39.4 ± 0.3	39.3 ± 0.4	38.9 ± 0.3	42.7 ± 0.5
GLGCN [11]	87.9 ± 0.4	86.4 ± 0.2	88.0 ± 0.5	88.9 ± 0.7
GLGAT	89.7 ± 0.3	89.1 ± 0.7	89.6 ± 0.4	90.2 ± 0.5
DGL	**91.1** ± 0.5	**91.3** ± 0.2	**91.6** ± 0.6	**92.4** ± 0.4

Table 6
Comparison of the proposed DGL method with the related graph learning methods (GLGCN, GLSGCN, GLGAT and DGL) for semi-supervised classification, average per-class accuracy (%) is reported based on the different data training/validation/testing in the Cifar10 image datasets. The initial graph for the computing model is not available.

Method	Cifar10 1000/1000/8000	Cifar10 2000/1000/7000	Cifar10 3000/1000/6000	Cifar10 4000/1000/5000
GLSGCN	63.5 ± 0.4	66.4 ± 0.3	71.5 ± 0.5	72.6 ± 0.2
GLGCN [11]	84.2 ± 0.2	79.7 ± 0.5	81.1 ± 0.8	86.8 ± 0.4
GLGAT	86.5 ± 0.8	87.4 ± 0.5	87.5 ± 0.6	88.0 ± 0.3
DGL	**87.5** ± 0.5	**88.8** ± 0.3	**88.8** ± 0.6	**88.8** ± 0.4
Method	Cifar10 5000/1000/4000	Cifar10 6000/1000/3000	Cifar10 7000/1000/2000	Cifar10 8000/1000/1000
GLSGCN	63.7 ± 0.5	73.3 ± 0.3	75.5 ± 0.6	71.0 ± 0.3
GLGCN [11]	83.7 ± 0.9	80.0 ± 0.5	84.5 ± 0.7	80.0 ± 0.7
GLGAT	85.2 ± 0.5	86.3 ± 0.4	87.5 ± 0.6	87.0 ± 0.3
DGL	**87.0** ± 0.2	**88.6** ± 0.5	**89.0** ± 0.4	**89.0** ± 0.3

Table 4 demonstrates that the performance of the proposed DGL is better than that of other graph learning methods with the least margins 1.5% on the Cora dataset, 2.8% on the Citeseer dataset, and 0.9% on the Pubmed dataset. This indicates that deep mining and fusion of the different structures can significantly improve the performance of semi-supervised classification. GLSGCN shows worse results than other methods in the Cora and Citeseer datasets, while this method has the approximate result of other methods in the Pubmed datasets. The main reason is that the simplifying structure of GLSGCN has a negative influence on graph structure learning in more categories.

Table 5 shows the experimental results in the MINIST image datasets. In the different training sets, DGL can, on average, outperform other graph learning methods with the least margins 1.6% on the MINIST dataset. The proposed GLGAT and DGL are, on average, superior to GLSGCN with the least margins 2.3% on the MINIST dataset. The same situation occurs in the Cifar10 dataset of Table 6. DGL is, on average, better than other graph learning methods with the least margins 1.5% on the Cifar10 dataset. The proposed GLGAT and DGL are, on average, superior to GLSGCN with the least margins 4.4% on the Cifar10 dataset. In all methods, increasing the training data is not a necessary and sufficient condition for better performance because of random data selection.

4.7. Ablation experiments

In this section, we expect to delete some parts from the DGL to analyze the functions of the different components. In the proposed DGL, 'deep' has two meanings. One meaning of 'deep' is the information mining from the global structure to local structure (from the S-module of the DGL to the A-module of the DGL in Fig. 2). Therefore, for capture the more global structure information by the different scale structures learning in the various layers, we delete A-module for simulating the situation (nonlocal structure), which is called **DGL-nonlocal. DGL-nonlocal** method is the deep layer extend of graph learning method GLGCN [11]. Therefore, the performance of **DGL-nonlocal** method can further evaluate the benefit of the graph learning module. Another meaning of 'deep' is the metric learning of the different scale convolution information (two graph learning layers of DGL in 2). Consequently, we delete the second graph learning layer to imitate shallow metric learning, which is called **DGL-shallow-metric**. If DGL does not consider the local graph structure and only considers the metric learning of the single-layer information, DGL will degrade to GLGCN. Therefore, the intrinsic difference between DGL and GLGCN is deep graph structure information mining and learning.

Table 7
Comparison of the proposed DGL method with GLGCN, and the ablated methods (DGL-nonlocal and DGL-shallow-metric) for semi-supervised classification, average per-class accuracy (%) is reported based on the different datasets. The initial graph for the computing model is not available.

Method	Cora	Citeseer	Pubmed
GLGCN [11]	60.1 ± 0.3	64.6 ± 0.2	73.3 ± 0.5
DGL-nonlocal	62.5 ± 0.5	65.9 ± 0.2	75.4 ± 0.3
DGL-shallow-metric	63.7 ± 0.2	66.2 ± 0.5	75.8 ± 0.4
DGL	**65.3** ± 0.3	**68.9** ± 0.4	**76.9** ± 0.5
Method	MNIST 1000/1000/8000	Cifar10 1000/1000/8000	
GLGCN [11]	84.9 ± 0.4	84.2 ± 0.2	
DGL-nonlocal	85.7 ± 0.3	85.2 ± 0.5	
DGL-shallow-metric	87.6 ± 0.6	86.9 ± 0.3	
DGL	**89.1** ± 0.6	**87.5** ± 0.5	

In Table 7, the performance of DGL is, on average, superior to that of other ablation methods with the least margins 1.5%

Table 8
Comparison of the proposed DGL method with the different metric methods (DGL-Euclidean and DGL-Cosine) for semi-supervised classification, average per-class accuracy (%) is reported based on the different datasets. The initial graph for the computing model is not available.

Method	Cora	Citeseer	Pubmed
DGL-Euclidean	63.5 ± 0.7	65.4 ± 0.8	73.6 ± 0.6
DGL-Cosine	63.7 ± 0.6	65.1 ± 0.9	74.2 ± 0.7
DGL	**65.3** ± 0.3	**68.9** ± 0.4	**76.9** ± 0.5
Method	MNIST 1000/1000/8000	Cifar10 1000/1000/8000	
DGL-Euclidean	83.4 ± 0.5	83.8 ± 0.8	
DGL-Cosine	85.7 ± 0.4	84.6 ± 0.6	
DGL	**89.1** ± 0.6	**87.5** ± 0.5	

on the Cora dataset, 2.7% on the Citeseer dataset, 1.1% on the Pubmed dataset, 1.5% on the MNIST dataset, and 0.6% on the Cifar10 dataset. Specifically, the proposed DGL is better, on average, than GLSGCN with the least margins 5.2% on the Cora dataset, 4.3% on the Citeseer dataset, 3.6% on the Pubmed dataset, 4.2% on the MNIST dataset, and 3.3% on the Cifar10 dataset, which shows that the local graph structure mined by the attention mechanism can complement the global structure captured by metric learning, so the performance of DGL-shallow-metric is better than that of GLGCN. Deep metric learning can obtain more abundant structural information from the different scale node representations, hence the classification accuracy of DGL-nonlocal outperforms that of GLGCN. The performance of DGL-shallow-metric is obviously better than that of DGL-nonlocal, and it demonstrates that hierarchical progressive learning from the global structure to the local structure can obtain a more positive effect than metric learning from the different scale node representation. Furthermore, both factors can be considered for constructing DGL, and DGL can obtain promising results for semi-supervised classification.

4.8. Metric experiments

The adjacency matrix is a graph representation for describing the distribution structure of all nodes. In some cases, the adjacency matrix is not available for graph convolutional networks. Although the adjacency matrix can be calculated by some metrics (Euclidean distance or cosine distance), these metrics are not the optimal method for constructing graph structure in semi-supervised classification because of the fixed computing style. To validate this point, we carry out metric experiments using different metrics based on the DGL network architecture. When we replace the similarity metric with the radial basis function of Euclidean distance for the adjacency matrix computation, this method is called DGL-Euclidean. If we substitute cosine similarity for similarity learning to calculate the adjacency matrix, this method is called DGL-Cosine.

Table 8 demonstrates that the performance of DGL is, on average, better than that of other metric methods with the least margins 1.6% on the Cora dataset, 3.5% on the Citeseer dataset, 2.7% on the Pubmed dataset, 3.4% on the MNIST dataset, and 2.9% on the Cifar10 dataset. Specifically, DGL-Euclidean and DGL-Cosine have similar performances, but these methods cannot adjust the metric parameters to adapt the evolution process of the graph. The DGL shows that the similarity metric can be dynamically learned for diversity data by global and local structure mining at different scales.

4.9. Graph learning visualization and efficiency evaluation

To directly observe the graph learning process, we reduce multi-dimensional node data to two dimensions to visualize data by t-SNE [39]. We show the node data distribution of the different episodes (1, 50, 100, 150) in the Cifar10 image datasets, in which the training/validation/testing data number is set to $1,000/1,000/8,000$. Fig. 4 shows the various structure distributions in the different learning stages. In episode 1, the data distribution presents the hybrid state of the class; in episode 50, fewer categories can be separated from all classes; in episode 100, more categories subsequently can be parted from all classes; in episode 150, most categories can be separated from each other. We observe that the global and local structure distributions gradually show the aggregation state of the class.

The proposed DGL includes many parameters, which have two types. One kind can learned by end to end paradigm, for instance parameters in Eqs. (1)-(8). Another kind can be decided by cross validation, for example hyper-parameters in Eq. (9). The number of parameter is heavily increasing with the network depth extending. It is difficult to avoid that the computation complex is heavy load. The main computation load of the proposed method comes from adjacent matrix calculation of graph learning in Eq. (2). The adjacent matrix calculation of graph learning is a key point to mine the nonlinear graph structure of data, and may be updated by increment computation to reduce the computation complex. Now that the number of data is the main fact for the computation complex, we can divide data into many tasks for reducing the number of data in each model training epoch by meta learning as increment computation method like our other work [41]. This idea will be considered in our future work. The main reason is that this paper mainly focuses on the global and local graphs information mining. The whole network can be efficiently trained by Adam algorithm. In Fig. 5, we evaluate the efficiency of the whole training and testing process between GLGCN and DGL in Cifar10 datasets. DGL model can efficiently be trained and tested for classification. Fig. 5 indicates that the loss changes with increasing episodes in DGL and GLGCN. The training or testing loss of DGL is obviously less than that of GLGCN, which shows that the DGL model can obtain better performance than the GLGCN model in training and testing for semi-supervised classification.

4.10. Experimental results analysis

In the experiments, eleven methods are utilized to evaluate the different aspects of the proposed DGL. These methods can be divided into four groups for different purposes. The first group includes four baseline methods (GCN [6], GAT [18], SGCN [40] and GLGCN [11] in Section 4.4) for understanding the motivation of the proposed DGL. The second group contains four state-of-art methods (H-GCN [7], GLNNs [32], DIAL-GNN [14] and GLGCN [11] in Section 4.5) for analyzing the advantages and disadvantages of these graph learning methods and the proposed DGL. The third group explores two methods (GLSGCN and GLGAT in Section 4.6) based on the main idea of GLGCN [11] for extending the graph learning method based on GCN [6]. The fourth group exploits two methods (DGL-nonlocal and DGL-shallow-metric in Section 4.7) for finding the function of the different components in the proposed DGL. According to the above experiments, we can make the following observations.

- DGL outperforms the baseline approaches, GCN [6], GAT [18], SGCN [40] and GLGCN [11] in Section 4.4. GCN [6] can reveal the node information propagation based on the statically global graph structure for capturing the data distribution relationship and node representation. GAT [18] can assign the weight of the neighborhood in each data node to learn the local graph structure. SGCN [40] can simplify the network architecture based on the statically global graph structure to reach the approximating results of GCN. GLGCN [11] can extract the global graph struc-

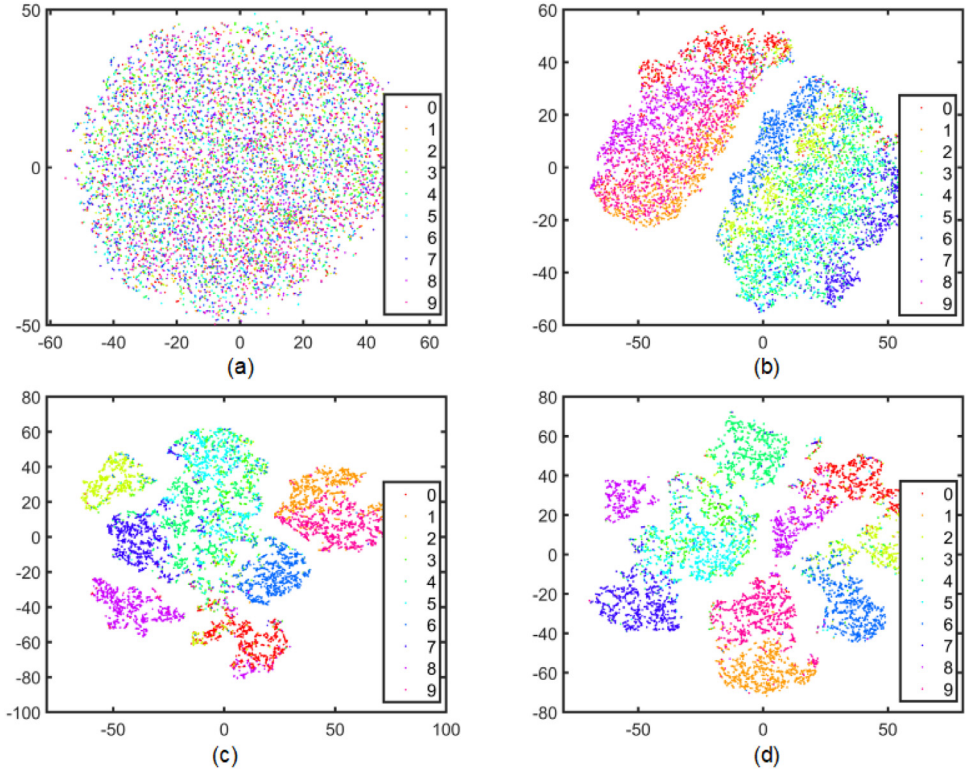

Fig. 4. The various structure distributions of the different learning stages of DGL in the Cifar10 dataset. (a) is the structure distribution of episode 1, (b) for that of episode 50, (c) for that of episode 100 and (d) for that of episode 150. The horizontal and vertical axes represent the different dimensions of the data.

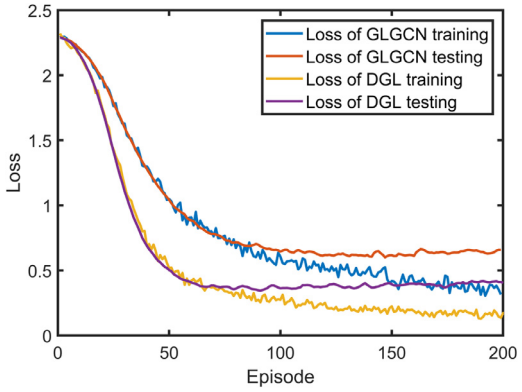

Fig. 5. The loss of DGL and GLGCN in training and testing in the Cifar10 dataset.

ture from the original data in network learning to construct the basic frameworks of graph learning based on GCN. DGL can not only dynamically mine the global and local graph structure to balance their effect on information propagation but also simultaneously encode the node representation of the different scale outputs to improve the performance of semi-supervised classification.

- The graph learning method-based GCN (GLGCN [11] and the proposed DGL) has an obvious performance improvement over the non-graph learning methods (GCN [6], GAT [18] and SGCN [40]). The main reason is that the graph learning methods can dynamically generate graph structure by the parameterized interaction computation, while non-graph learning methods only depend on the static graph structure in the whole network learning regardless of the change in each layer. Therefore, the graph learning methods can better fit the distribution of the transforming data in each layer to enhance the performance of semi-supervised classification.
- In the state-of-the-art graph learning methods based on neural networks (H-GCN [7], GLNNs [32], DIAL-GNN [14] and GLGCN [11] in Section 4.5), the global or local graph structure can be described and mined by hierarchical aggregation or metric learning. The proposed DGL can comprehensively consider the global and local graph structure and encode their propagation relationship to improve the performance of the network model. Therefore, DGL can obtain the best performance of the Citeseer and Pubmed datasets and the approximated best performance of the Cora dataset in these state-of-the-art methods.
- The extended graph learning methods (GLSGCN and GLGAT in Section 4.6) combine the main idea of GLGCN [11] with GAT [18] or SGCN [40] to find the adaptation of the graph learning method. GLSGCN achieves worse performance than GLGCN [11], while GLGAT can achieve better performance than GLGCN [11]. This shows that the nonlinear unit layer has a stronger learning ability for dynamically generating the graph structure. DGL outperforms GLSGCN and GLGAT, and it demonstrates that the different scale metric learning (from the global to the local graph structure and from the different layers) can contribute to the construction of the graph learning model.
- The proposed DGL method can delete the different components to formulate the different ablation methods (DGL-nonlocal and DGL-shallow-metric in Section 4.7). The DGL-nonlocal method emphasizes global graph structure learning from the different scale node representations, while the DGL-shallow-metric focuses on balance learning between the global and local graph structures in a single layer. The performance of DGL-shallow-metric is superior to that of DGL-nonlocal, which indicates that the depth mining from the global graph structure to the local graph structure has a more obvious effect than deep metric learning from the different scale outputs. However, two factors

are simultaneously considered to build a DGL that can obtain promising results for semi-supervised classification.

- Adjacency matrix (the representation of the graph) not only greatly influences the final performance, but also gradually approximates to the intrinsic data structure with hierarchical progressive learning. In experiment, Table 2 is the results with the initial graph, while Table 4 is the results without the initial graph in Cora, Citeseer and Pubmed datasets. It shown that different initial graphs can provider the richer information form the different views. However, the initial graph is often non-available, and needs gradually be computed and optimized form data. Therefore, we present DGL to capture the more approximate to the intrinsic structure for classification.
- In the extended graph learning experiment, the different graph learning methods show approximate results with training/validation/testing changes. This reveals that the graph learning process can complement the insufficient number of training samples to improve the generalization of the model. Therefore, in Tables 5 and 6, this situation occurs in the experimental results of the different graph learning methods.
- The computation load of the proposed method mainly involves the computation complex of two graph learning layer unit, two graph convolution layer unit, and two graph attention layer unit. Supposed, n is the number of data as the nodes of the graph, d is the dimension of the data and m is the number of edges between the nodes of the graph. Therefore, the computation complex of one graph learning layer unit is $O(n^2d)$, the computation complex of one graph convolution layer unit is $O(md)$, and the computation complex of one graph attention layer unit is $O(md)$. The computation load of the proposed method is about $O(4md + 2n^2d)$.

5. Conclusion

We presented a deep graph learning (DGL) method to address global and local graph integration learning for improving semi-supervised classification. The proposed DGL can not only use the graph learning layer and graph attention layer for hierarchical progressive graph structure mining but also adopt two graph learning layers for deeply capturing the global graph structure information from the different scale node representations. Furthermore, DGL can balance the difference between the global and local graph structure to find the abundant data relationship and fuse the node representation of the different layers to enhance semi-supervised classification. Finally, DGL can automatically generate graph structures in network learning and dynamically encode the various information of the different layers. The experimental results and analysis show that the proposed DGL method is promising for node classification on the Citeseer, Cora, Pubmed, MNIST and Cifar10 datasets.

The proposed DGL can provide an alternative metric learning method, which is always a key question in pattern recognition. This method attempts to optimize the graph structure by the hierarchical progressive from the global to the local graph for semi-supervised classification. The optimized graph can be used to find potential commerce value based on the recommender system, capture latent social relationships based on social networks, and distinguish invisible representation patterns based on gene analysis.

However, the limitation of the proposed DGL, as with most methods based on GCN, involves the information input of all nodes, which may include considerable data. This situation causes high computational loading. To find the increment graph structure mining method, we need to construct a deep graph structure transfer mechanism for increasing data. Therefore, the DGL framework based on transfer learning may be a research direction for solving this issue.

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

The authors would like to thank the anonymous reviewers for their insightful comments that helped improve the quality of this paper. This work was supported by the NSFC (Program no. 61771386, Program no. 61671376 and Program no. 61671374) and Key Research and Development Program of Shaanxi (Program no. 2020SF-359).

References

[1] F. Manessi, A. Rozza, M. Manzo, Dynamic graph convolutional networks, Pattern Recognit. 97 (2020) 107000.
[2] L. Zhao, X. Peng, Y. Tian, M. Kapadia, D.N. Metaxas, Semantic graph convolutional networks for 3d human pose regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3425–3435.
[3] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, C. Gan, Graph convolutional networks for temporal action localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 7094–7103.
[4] Z. Liu, J. Dong, C. Zhang, L. Wang, J. Dang, Relation modeling with graph convolutional networks for facial action unit detection, in: International Conference on Multimedia Modeling, Springer, 2020, pp. 489–501.
[5] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 7370–7377.
[6] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings, OpenReview.net, 2017.
[7] F. Hu, Y. Zhu, S. Wu, L. Wang, T. Tan, Hierarchical graph convolutional networks for semi-supervised node classification, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10–16, 2019, ijcai.org, 2019, pp. 4532–4539.
[8] J.B. Lee, R.A. Rossi, X. Kong, S. Kim, E. Koh, A. Rao, Higher-order graph convolutional networks, arXiv preprint: 1809.07697(2018).
[9] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, Y.Y. Tang, Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification, IEEE Geosci. Remote Sens. Lett. 16 (2) (2018) 241–245.
[10] G. Lin, J. Wang, K. Liao, F. Zhao, W. Chen, Structure fusion based on graph convolutional networks for node classification in citation networks, Electronics 9 (3) (2020) 432.
[11] B. Jiang, Z. Zhang, D. Lin, J. Tang, B. Luo, Semi-supervised learning with graph learning-convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11313–11320.
[12] B. Jiang, P. Sun, J. Tang, B. Luo, Glmnet: Graph learning-matching networks for feature matching, arXiv preprint: 1911.07681(2019b).
[13] B. Jiang, X. Jiang, A. Zhou, J. Tang, B. Luo, A unified multiple graph learning and convolutional network model for co-saliency estimation, in: Proceedings of the 27th ACM International Conference on Multimedia, ACM, 2019, pp. 1375–1382.
[14] Y. Chen, L. Wu, M.J. Zaki, Deep iterative and adaptive learning for graph neural networks, arXiv preprint: 1912.07832(2019).
[15] W. Hu, X. Gao, G. Cheung, Z. Guo, Feature graph learning for 3d point cloud denoising, IEEE Trans. Signal Process. 68 (2020) 2841–2856.
[16] H. Du, L. Ma, G. Li, S. Wang, Low-rank graph preserving discriminative dictionary learning for image recognition, Knowl Based Syst 187 (2020) 104823.
[17] G. Lin, K. Liao, B. Sun, Y. Chen, F. Zhao, Dynamic graph fusion label propagation for semi-supervised multi-modality classification, Pattern Recognit. 68 (2017) 14–23.
[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30, - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018.
[19] Z. Kang, H. Pan, S.C.H. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE Trans. Cybern. 50 (5) (2020) 1833–1843.
[20] S. Huang, Z. Kang, I.W. Tsang, Z. Xu, Auto-weighted multi-view clustering via kernelized graph learning, Pattern Recognit. 88 (2019) 174–184.
[21] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 171–184.
[22] Z. Kang, X. Lu, J. Yi, Z. Xu, Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 2312–2318.
[23] S. Kumar, J. Ying, C.J.V. de M., D.P. Palomar, A unified framework for structured graph learning via spectral constraints, J. Mach. Learn. Res. 21 (22) (2020) 1–60.

[24] Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, L. Tian, Structured graph learning for clustering and semi-supervised classification, Pattern Recognit. 110 (2020) 107627.
[25] Z. Hu, F. Nie, W. Chang, S. Hao, R. Wang, X. Li, Multi-view spectral clustering via sparse graph learning, Neurocomputing 384 (2020) 1–10.
[26] P. Chen, L. Jiao, F. Liu, Z. Zhao, J. Zhao, Adaptive sparse graph learning based dimensionality reduction for classification, Appl. Soft Comput. 82 (2019) 105459.
[27] X. Pei, J. Zou, W. Chen, Graph learning via edge constrained sparse representation for image analysis, IEEE Access 7 (2019) 42408–42417.
[28] F. Dornaika, R. Dahbi, A. Bosaghzadeh, Y. Ruichek, Efficient dynamic graph construction for inductive semi-supervised learning, Neural Netw. 94 (2017) 192–203.
[29] J. Kim, T. Kim, S. Kim, C.D. Yoo, Edge-labeling graph neural network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11–20.
[30] L. Gong, Q. Cheng, Exploiting edge features for graph neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9211–9219.
[31] Q. Li, X. Zhang, H. Liu, X.-M. Wu, Attributed graph learning with 2-d graph convolution, arXiv preprint: 1909.12038(2019).
[32] X. Gao, W. Hu, Z. Guo, Exploring structure-adaptive graph learning for robust semi-supervised classification, arXiv preprint: 1904.10146(2019).
[33] Z. Kang, X. Lu, J. Liang, K. Bai, Z. Xu, Relation-guided representation learning, Neural Netw. 131 (2020) 93–102.
[34] S. Prithviraj, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008). 93–93
[35] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[36] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, Technical Report, Citeseer, 2009.
[37] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2020) 1–21.
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[39] M.L. van der, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.
[40] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph convolutional networks, in: International Conference on Machine Learning, 2019, pp. 6861–6871.
[41] G. Lin, Y. Yang, Y. Fan, X. Kang, K. Liao, F. Zhao, High-order structure preserving graph neural network for few-shot learning, arXiv preprint: 2005.14415(2020).

Guangfeng Lin received the B.S. degree in mechatronic engineering from Xi'an Institute of Technology in 2001, the M.S. degree in traffic information engineering and control from Chang'an University in 2005 and the Ph.D. degree in control theory and control engineering from Xi'an University of Technology in 2013. From September 2014 to September 2015, he have completed postdoctoral research about heterogenous structure fusion for classification in Visual Computing and Image Processing Lab (VCIPL) at Oklahoma State University. He is an associate professor in the Department of Information Science at Xi'an University of Technology. His research interests focus on digital image processing and pattern recognition. He is CCF professional member and IEEE SPS member.

Xiaobing Kang is currently working as an associate professor in Department of Information Science, Faculty of Printing, Packaging Engineering and Digital Media Technology, Xi'an University of Technology, Xi'an, China. He received his B.E. Degree in University of Science and Technology Beijing, China, his M.E. and Ph.D. Degrees in Northwestern Polytechnical University, Xi'an, China, respectively. He is a member of IEEE and CCF. His main research interests include Signal and Image Processing, Multimedia Forensics and Security, Machine Learning.

Kaiyang Liao received the B.S. degree in computer science from the XIDIAN University, Xi'an, China, in 2004, the M.S. degree in computer science from the University of Science and Technology LiaoNing, Anshan, China, and the Ph.D. degree in Information and Communication Engineering from the Xi'an Jiaotong University, Xi'an, China, in 2013. He is currently a Full lecturer with the School of Printing and Packaging Engineering, Xi'an University of Technology, Xi'an, China. His research interests include data mining, pattern recognition, video analysis and retrieval.

Fan Zhao received a Ph.D. in Information and Communication Engineering from Xi'an Jiaotong University, Xi'an, China, in 2009. She worked as a postdoctoral fellow in the Department of Computer Science and Engineering, Xi'an Jiaotong University from July 2010 to April 2012. She is an associate professor in the Department of Information Science at Xi'an University of Technology., Xi'an, China, Her research interests include image processing, object tracking, and pattern recognition.

Yajun Chen received his B.S. degree in mechanical engineering, M.S. degree in signal and information processing and Ph.D. degree in control theory and control engineering, all from Xi'an University of Technology, Xi'an, China, in 2002, 2006 and 2015, respectively. He is an associate professor in the Department of Information Science at Xi'an University of Technology. He is CCF professional member. His research interests cover digital image processing and machine vision.