Information Sciences 573 (2021) 171–181

Contents lists available at ScienceDirect

Information Sciences

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

Semi-supervised learning with mixed-order graph convolutional networks

Jie Wang, Jianqing Liang *, Junbiao Cui, Jiye Liang

Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Shanxi, Taiyuan 030006, China

ARTICLE INFO

Article history:
Received 12 November 2020
Received in revised form 19 May 2021
Accepted 26 May 2021
Available online 28 May 2021

Keywords:
Semi-supervised learning
Graph convolutional networks
Oversmoothing
Mixed-order

ABSTRACT

Recently, graph convolutional networks (GCN) have made substantial progress in semi-supervised learning (SSL). However, established GCN-based methods have two major limitations. First, GCN-based methods are restricted by the oversmoothing issue that limits their ability to extract knowledge from distant but informative nodes. Second, most available GCN-based methods exploit only the feature information of unlabeled nodes, and the pseudo-labels of unlabeled nodes, which contain important information about the data distribution, are not fully utilized. To address these issues, we propose a novel end-to-end ensemble framework, which is named mixed-order graph convolutional networks (MOGCN). MOGCN consists of two modules. (1) It constructs multiple simple GCN learners with multi-order adjacency matrices, which can directly capture the high-order connectivity among the nodes to alleviate the problem of oversmoothing. (2) To efficiently combine the results from multiple GCN learners, MOGCN employs a novel ensemble module, in which the pseudo-labels of unlabeled nodes from various GCN learners are used to augment the diversity among the learners. We conduct experiments on three public benchmark datasets to evaluate the performance of MOGCN on semi-supervised node classification tasks. The experimental results demonstrate that MOGCN consistently outperforms state-of-the-art methods.

1. Introduction

The success of machine learning algorithms typically depends on the data representation [4]. Deep learning [25], as one of the representation learning [7,17,18,27,28] methods, has made substantial breakthroughs in many fields, such as image vision, speech recognition, and natural language understanding. However, most deep learning models usually work under the supervised setting, in which the labels of training samples are assumed to be known. It is highly difficult and expensive to obtain labels for training samples in practical applications. Thus, deep learning will be impossible in scenarios in which labeled samples are extremely rare.

Fortunately, in many real-world tasks, unlabeled samples are readily available. Although unlabeled samples are unable to provide label information, they contain important information about the data distribution. Therefore, as one of the major paradigms for the exploitation of unlabeled samples, semi-supervised learning [38], which exploits unlabeled samples together with labeled samples to improve learning performance, has received widespread attention. Among established

* Corresponding author.
E-mail addresses: wangjie_reg@163.com (J. Wang), liangjq@sxu.edu.cn (J. Liang), CuiJunBiao163@163.com (J. Cui), ljy@sxu.edu.cn (J. Liang).

https://doi.org/10.1016/j.ins.2021.05.057
0020-0255/© 2021 Published by Elsevier Inc.

semi-supervised learning models, graph-based methods, especially graph convolutional networks (GCN) methods [23], have been demonstrated to be the most effective approaches. GCN-based methods use a message-passing scheme, in which each node aggregates features from its neighboring nodes to update its own feature vector. After K message-passing rounds or when the depth of GCN has increased to K layers, a node can aggregate information from nodes up to K hops away in the graph. Thus similar nodes have similar representations.

Despite their enormous success, the classification performance of GCN-based methods decreases rapidly when labeled training nodes are scarce. The decline in performance is due mainly to the following two reasons. First, according to the message-passing mechanism that is adopted by GCN-based methods, they explore different neighbor information of nodes by increasing the number of layers. As the depth increases, GCN-based methods mix the features of the nodes from different connections of neighbors and render them indistinguishable. This phenomenon is explained by [26] as an oversmoothing issue that can be observed even for small values of K (as low as $K = 3$). However, when the depth is too small, it is difficult to extract the knowledge from distant but informative nodes, i.e., remote hops of neighbors for the current node. Second, most available GCN-based methods exploit only the feature information of unlabeled nodes. However, pseudo-labels, which contain important information about the data distribution, are not fully utilized. In summary, these two drawbacks limit their ability to spread the node feature information to other nodes in the graph effectively, thereby leading to poor trainability and expressivity.

Based on the aforementioned analysis, we argue that a key direction of the construction of graph convolutional network models lies in the efficient exploration and combination of information from various connections of neighbors. In this paper, we propose a novel end-to-end ensemble framework, which is named mixed-order graph convolutional networks (MOGCN). First, instead of using the method of increasing the number of layers to explore the various neighbor information of nodes, we construct multi-order adjacency matrices in advance. Fig. 1 illustrates examples of 1st-order, 2nd-order, and 3rd-order relations in a graph. Different order neighbor relation graphs reflect different connections (nearest neighbors) of nodes. Then, on each special order adjacency matrix, we construct a simple GCN (e.g., $K = 2$ layers). Via this strategy, we can directly obtain the node representations based on various neighbor relations (or receptive fields) from multiple GCN learners and alleviate the problem of oversmoothing that is caused by stacking multiple message-passing layers. Second, it is natural to introduce ensemble learning to combine the results of GCN learners. The generalization error of an ensemble is related to the average generalization error of the base learners and the diversity among the base learners. Generally, the higher the average accuracy and the diversity of the base learners, the better the ensemble [24]. Therefore, to efficiently fuse the results from multi-GCN learners, an ensemble module that is based on negative correlation learning [30] is designed. Combined with the pseudo-labels of unlabeled nodes, the ensemble module can maximize the accuracy of GCN learners on labeled nodes, while maximizing the diversity among them on unlabeled nodes. Finally, we conduct extensive experiments to demonstrate the state-of-the-art performance of our approach. The main contributions of the paper can be summarized as follows:

- We propose a novel end-to-end ensemble framework, i.e., mixed-order graph convolutional networks (MOGCN), for graph-based semi-supervised learning. MOGCN combines the results of multiple GCN learners that are trained on adjacency matrices of various orders to boost the performance of semi-supervised node classification.
- MOGCN can directly obtain the node representations based on various neighbor relations and alleviate the problem of oversmoothing that is caused by stacking multiple message-passing layers.
- In the ensemble module, the pseudo-labels of unlabeled nodes are fully utilized to augment the diversity among the GCN learners.

The remainder of this paper is organized as follows. Section 2 discusses the related work on semi-supervised learning and graph convolutional networks. We introduce our proposed method in Section 3 series of experiments for evaluating the performance of the proposed method are conducted in Section 4. Finally, Section 5 presents the conclusions of this study and discusses future work.

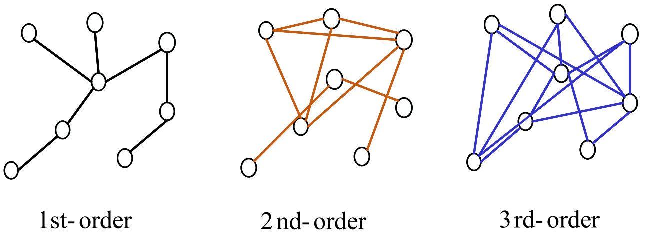


Fig. 1. Examples of high-order relations in a graph. Left is the 1st-order graph, middle is the 2nd-order graph and right is the 3rd-order graph. Different orders reflect different connections (nearest neighbors) of nodes. Self-loops are not shown in the figure for simplicity.

2. Related work

2.1. Semi-supervised learning

Semi-supervised learning (SSL) is widely adopted in many scenarios, in which the labeled samples are insufficient, while the unlabeled samples are extremely abundant. Various types of semi-supervised learning methods have been proposed, which include the following four main aspects: the generative semi-supervised learning method [9,34], the semi-supervised SVM method [6,29] the disagreement-based semi-supervised learning method [14,21,43] and the graph-based semi-supervised learning method (GSSL) [48]. Among established semi-supervised learning models, GSSL has received a large amount of attention, as it can map the dataset into a graph with the connection between the samples to realize satisfactory generalization performance. The study of GSSL methods can be divided into two aspects: graph construction [19,20] and graph-based label inference [11,35]. Interested readers can refer to survey papers [38] for additional details.

2.2. Graph convolutional networks

Graph convolutional networks (GCN) extend deep learning algorithms to graph-structured data by defining convolution operators on a graph and have been proven powerful when dealing with various downstream tasks [41,46]. GCN can be divided into two operations: spectral convolutions and spatial convolutions. Spectral convolutions are performed by transforming node representations into the spectral domain using the graph Fourier transform. Bruna et al. [5] first introduced convolution for graph data from the spectral domain. ChebNet [10] and GCN [23] were proposed to use a polynomial or a first-order spectral convolution function to solve the efficiency problem. Following that, HesGCN [13] obtained a more efficient convolution layer rule by optimizing the one-order spectral graph Hessian convolutions. Spatial convolutions are performed by considering node neighborhoods, such as Neural FPs [12], DCNN [3], MoNet [33], MPNNs [15], HAN [44] and DGI [40]. Several studies focused on improving the basic convolution operator, i.e., neighborhood aggregation schemes, such as attention mechanism [39], disentangled GCN [32], and making them more scalable on large graphs [8,16]. However, all these methods utilize the information of only a very limited neighborhood for each node. Such that it cannot effectively propagate the feature or label to the entire graph, especially for nodes in the periphery or in a sparsely labeled setting. This problem has been alleviated by directly increasing the number of labeled nodes, for example, [26] proposed Co-Training and Self-Training methods for enlarging the training dataset. After that, [36] proposed multi-stage self-training processing, which applies the deep clustering method on an embedding and relies on distance measures to align and extend the labeled dataset. However, these two types of methods enlarge the labeled training dataset by assigning pseudo-labels to unlabeled nodes, which inevitably introduces incorrect label information (label noise), especially when labeled nodes are scarce. In addition, similar to our model, MixHop [2] also attempted to use higher-order adjacency matrices for node feature aggregation. However, MixHop only simply splices the information from adjacency matrices of orders. The redundancy between them is not considered, which will cause performance degradation when combining that information. Moreover, node representations with different high-order connectivity among nodes cannot be obtained directly; thus, this approach is less flexible than approaches that combine multi-GCNs.

2.3. Negative correlation learning

Negative correlation learning (NCL) [30,42] is an ensemble learning algorithm that introduces a correlation penalty term into the cost function of each ensemble member. Each ensemble member minimizes its mean square error and its error correlation with the remainder of the ensemble. NCL has been shown to perform well on many applications, such as regression [47] and classification [31] problems.

3. Mixed-order graph convolutional networks

In this section, we introduce notations and definitions that will be used in the remainder of this paper. Then, we present the proposed mixed-order graph convolutional networks. Specifically, we first construct multiple GCN-based learners that are trained on adjacency matrices of various orders. In addition, we develop a novel module for ensembling the results from the base learners.

3.1. Notations

We consider a general undirected attribute graph $\mathscr{G} = (V, E, \mathbf{X})$, where $V = \{v_1, v_2, \ldots, v_N\}$ denotes the set of nodes, N is the number of nodes, and E denotes the set of edges with $e_{i,j} =< v_i, v_j >\in E$ denotes an edge between v_i and v_j. The structure of graph $\mathscr{G}$ can be represented by an adjacency matrix $\mathbf{A} = \{a_{i,j}\} \in \mathbb{R}^{N\times N}$, and $a_{i,j}$ denotes the entry of matrix $\mathbf{A}$ at the i-th row and the j-th column, $a_{i,j} = 1$ if $e_{i,j} \in E$, otherwise, $a_{i,j} = 0$. Additionally, we denote the node attribute matrix that is associated with the graph as $\mathbf{X} \in \mathbb{R}^{N\times F}$, in which F is the dimension of the features, and $\mathbf{x}_i \in \mathbb{R}^F$corresponds to the i-th row of matrix $\mathbf{X}$, which is the attribute vector of node v_i. $\mathbf{D} = diag(d_1, d_2, \cdots, d_N) \in \mathbb{R}^{N \times N}$ represents the degree matrix of $\mathbf{A}$, $d_i = \sum_{v_j \in V} a_{i,j}$ is the degree of node v_i.

For semi-supervised learning (SSL), let $X_L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$ denote the L labeled samples and $X_U = \{\mathbf{x}_j\}_{j=L+1}^{L+U}$ denote the U unlabeled samples, $N = L + U$ and $y_i \in \{1, 2, \cdots, C\}$ is the class label of the i-th labeled sample, in which C is the number of categories. In graph-based semi-supervised learning (GSSL), only a subset of nodes $V_L \subset V$ are labeled. In general, $|V_L| \ll |V|$. The objective of GSSL is to recover labels for all unlabeled nodes $V_U = V - V_L$, using the feature matrix $\mathbf{X}$, the known labels for nodes in V_L, and the graph structure $\mathbf{A}$.

For convenience of discussion, the class labels of the nodes are represented in the form of a matrix. Let $\mathbf{Z} = \{z_{i,c}\} \in \{0, 1\}^{N \times C}$ be the label matrix,

$$z_{i,c} = \begin{cases} 1, & i = 1, 2, \cdots, L,\ y_i = c \\ 0, & otherwise \end{cases}$$

Let $\tilde{\mathbf{Z}} = \{\tilde{z}_{i,c}\} \in \mathbb{R}^{N \times C}$ be the predicting label matrix, $\tilde{z}_{i,c}$ represents the corresponding degree of the i-th node to the c-th category.

3.2. Overall framework

The architecture of our proposed MOGCN is illustrated in Fig. 2. It is composed of two main modules: a GCN-based learners module and an ensemble module:

- **GCN-based learners module**: MOGCN constructs multiple simple graph neural network (e.g., two-layer GCN) learners with multi-order adjacency matrices. Via this strategy, we can directly obtain node representations based on various neighbor relations from multiple GCN learners.
- **Ensemble module**: The proposed ensemble module is a fusion component that aggregates the results from the base learners to obtain the final label in a way that maximizes the accuracies of the base learners on the labeled nodes and their diversity on the unlabeled nodes.

3.3. GCN-based Learners module

The crucial part of MOGCN is to explore the knowledge from various relations of neighbors. To enlarge the learning field of each node, in this section, we construct multiple adjacency matrices of various orders, which directly represent the relations between nodes. Inside those graphs, each node can directly connect with farther neighbors. Then, we construct multiple GCN-based learners on those adjacency matrices, which enables each node to learn the node representations from various neighbor relations directly.

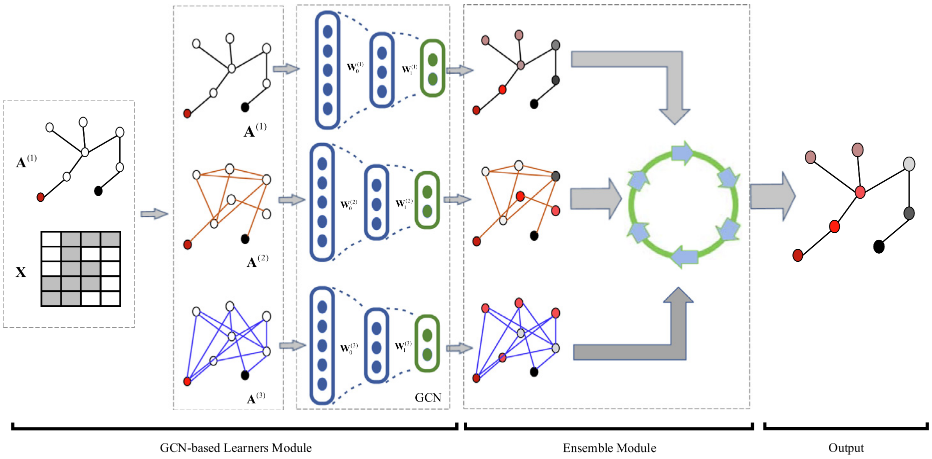


Fig. 2. Architecture of the proposed MOGCN. The model takes the adjacency matrix multiple $\mathbf{A}$ and the node attribute matrix $\mathbf{X}$ as input. Then, we construct adjacency matrices of m orders $\{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(m)}\}$ ($m = 3$ in the figure for simplicity). After that, MOGCN constructs multiple simple GCN-based learners with those matrices to obtain the initial result in each branch. Finally, the ensemble module is exploited to combine them and obtain the final label. In the figure, the red and black solid points represent labeled nodes of two categories, the white hollow points represent unlabeled nodes in the graph, and the darkness of a node's color indicates the degree to which the node belongs to corresponding category.

3.3.1. Multi-order adjacency matrices construction

The definition of the 2nd-order adjacency matrix in [37] is as follow. Denote $\mathbf{a}_j$ as the j-th row of the 1st-order adjacency matrix $\mathbf{A}^{(1)}$, and $a_{i,j}^{(2)}$ as the entry of the 2nd-order adjacency matrix $\mathbf{A}^{(2)}$ at the i-th row and the j-th column; the mathematical definition of $a_{i,j}^{(2)}$ is:

$$a_{i,j}^{(2)} = \mathbf{a}_i^{\top}\mathbf{a}_j, \forall i,j \in \{1,2,\cdots,N\}.$$

According to this definition, we can readily calculate an mth-order adjacency matrix $\mathbf{A}^{(m)}$ via

$$\mathbf{A}^{(m)} = \mathbf{A}^{(m-1)}\mathbf{A}^{(1)}. \tag{1}$$

By extending the original adjacency matrix to a sequence of adjacency matrices, we obtain a family of multi-order adjacency matrices $\left\{\mathbf{A}^{(1)},\mathbf{A}^{(2)},\ldots,\mathbf{A}^{(m)}\right\}$. Thus, we utilize those adjacency matrices to directly represent the various relations between nodes.

We store $\mathbf{A}^{(m)}$ as a sparse matrix with M nonzero entries. Under the realistic assumptions of $m \ll M$ and $M \ll N$, it is efficient to obtain multiple orders of the adjacency matrix.

3.3.2. Multi-order graph convolutional networks

Given the attribute matrix $\mathbf{X}$ and the family of multi-order adjacency matrices $\left\{\mathbf{A}^{(1)},\mathbf{A}^{(2)},\ldots,\mathbf{A}^{(m)}\right\}$, MOGCN constructs multiple branches of GCN-based learners $\{f_1,f_2, \ldots, f_m\}$, respectively. The k-th learner f_k $(1 \leqslant k \leqslant m)$ in MOGCN that is trained on $\mathbf{A}^{(k)}$is a simple two-layer GCN learner:

$$f_k = softmax\left(\tilde{\mathbf{A}}_s^{(k)} ReLU\left(\tilde{\mathbf{A}}_s^{(k)}\mathbf{X}\mathbf{W}_0^{(k)}\right)\mathbf{W}_1^{(k)}\right), \tag{2}$$

where $\tilde{\mathbf{A}}_s^{(k)} = \tilde{\mathbf{D}}^{(k)-\frac{1}{2}}\tilde{\mathbf{A}}^{(k)}\tilde{\mathbf{D}}^{(k)-\frac{1}{2}}$, and $\tilde{\mathbf{A}}^{(k)} = \mathbf{A}^{(k)} + \mathbf{I}$, and the associated degree matrix is $\tilde{\mathbf{D}}^{(k)} = \mathbf{D}^{(k)} + \mathbf{I}$. $\mathbf{W}_0^{(k)}$ and $\mathbf{W}_1^{(k)}$ are the trainable weight matrices in the k-th learner f_k, and $\mathrm{ReLU}(\cdot) = \max(0,\cdot)$.

The outputs of f_k is a initial predicting label matrix $\tilde{\mathbf{Z}}^{(k)} = \left\{\tilde{z}_{i,c}^{(k)}\right\} \in \mathbb{R}^{N\times C}$, where $\tilde{z}_{i,c}^{(k)}$ represents the correspondence degree of the i-th node to the c-th category in the associated learner f_k.

In this GCN-based learners module, we can directly obtain the node representations based on various neighbor relations from multiple GCN learners and alleviate the problem of oversmoothing that is caused by stacking multiple message-passing layers.

3.4. Ensemble module

Through the above module, we obtain the initial results $\left\{\tilde{\mathbf{Z}}^{(1)},\tilde{\mathbf{Z}}^{(2)},\ldots,\tilde{\mathbf{Z}}^{(m)}\right\}$ from the multiple GCN-based learners, where $\tilde{\mathbf{Z}}^{(k)}$ $(1 \leqslant k \leqslant m)$ denotes the representation that is obtained by propagating information from nodes that are k-hops away. As the depth k increases, information that is far away from the node will be included in $\tilde{\mathbf{Z}}^{(k)}$. Therefore, the initial results contain information from local and distant neighborhoods. Since in graph-based semi-supervised learning, there are few labeled nodes, but many unlabeled nodes. These unlabeled nodes can obtain initial pseudo-labels through GCN based modules. Inspired by negative correlation learning (NCL) [30], we propose a novel ensemble module for combining them. In this module, we maximize the accuracies of base learners on labeled nodes and their diversity on unlabeled nodes, via this strategy, we adaptively adjust and combine the information from local and distant neighborhoods to generate suitable labels for unlabeled nodes.

Formally, the proposed ensemble module in MOGCN combines m graph convolutional networks learners $\{f_1,f_2,\ldots,f_m\}$ to form an ensemble:

$$f_{\mathrm{ens}}(\mathbf{x}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{x}). \tag{3}$$

The objective of the ensemble module is to maximize the fit of the base learners on labeled nodes while maximizing the diversity of the learners on unlabeled nodes. Therefore, the proposed novel ensemble model needs to minimize the following global loss function:

$$L = (1-\eta)L_{emp} + \eta L_{div}, \tag{4}$$

where the first term L_{emp} on the right-hand side of Eq. (4) is the empirical loss of the labeled dataset V_L; the second term L_{div} is the diversity loss of the base learners on unlabeled dataset V_U and $0 \leqslant \eta \leqslant 1$ is a hyper-parameter that controls the correlation between the empirical loss and the diversity loss.

The first term L_{emp} in Eq. (4) is the cross-entropy error over all labeled nodes:

$$L_{emp} = \frac{1}{m}\sum_{k=1}^{m}\cdot\frac{1}{L}\sum_{i=1}^{L}\sum_{c=1}^{C}\left(-z_{i,c}\ln\tilde{z}_{i,c}^{(k)}\right). \tag{5}$$

The second term L_{div} is a diversity loss function. It minimizes each network's mean square error with errors for the remainder of the ensemble calculated by unlabeled nodes:

$$L_{div} = \exp\left\{\frac{1}{m}\sum_{k=1}^{m}\left(\frac{1}{U}\sum_{i=L+1}^{L+U}\|f_k(x_i) - f_{ens}(x_i)\|_2\sum_{k'\neq k}\|f_{k'}(x_i) - f_{ens}(x_i)\|_2\right)\right\} = \exp\left(-\frac{1}{mU}\sum_{k=1}^{m}\sum_{i=L+1}^{L+U}\|f_k(x_i) - f_{ens}(x_i)\|_2^2\right). \tag{6}$$

Through the above ensemble module, we combine the outputs of all m GCN learners, which enables us to jointly train all GCN learners and the ensemble network via backpropagation. Thus, MOGCN can aggregate the information of each node that is learned from multiple branches via an end-to-end way.

Finally, the prediction function is:

$$\tilde{y}_i = \underset{c\in\{1,2,\cdots,C\}}{\arg\max}\tilde{z}_{i,c}, \tag{7}$$

where, $\tilde{y}_i$ is the final prediction result of the i-th unlabeled node, and $\tilde{z}_{i,c}$ is the entry of the prediction matrix $\tilde{\mathbf{Z}}$ that was obtained via Eq. (3) after all training steps.

Overall, the framework of the MOGCN algorithm is described in Algorithm 1.

Algorithm 1: MOGCN (Mixed-Order Graph Convolutional Networks)

```
Input: Attribute matrix $\mathbf{X}$, 1st-order adjacency matrix $\mathbf{A}^{(1)}$, label matrix
       $\mathbf{Z}$, number of orders $m$, parameter $\eta$, and maximum number of
       iterations $max\_iter$.
Output: Final prediction result of the unlabeled nodes with Eq. (7).
Construct high-order adjacency matrix sets $\{\mathbf{A}^{(2)}, \mathbf{A}^{(3)}, \ldots, \mathbf{A}^{(m)}\}$ via Eq.
(1) ;
for $k = 1$ to $m$ do
    Initialize parameters $\{\mathbf{W}_0^{(k)}, \mathbf{W}_1^{(k)}\}$ of GCN learner $f_k$;
end
for $iter = 1$ to $max\_iter$ do
    for $k = 1$ to $m$ do
        Update the $k$-th GCN learner $f_k$ via Eq. (2);
    end
    Calculate the ensemble learner $f_{ens}$ via Eq. (3);
    Calculate the empirical loss $L_{emp}$ of the labeled nodes via Eq. (5);
    Calculate the diversity loss $L_{div}$ of the base learners via Eq. (6);
    Train the MOGCN with the global loss $L$ in Eq. (4);
end
```

4. Experiments

In this section, we evaluate the proposed model against state-of-the-art semi-supervised node classification models. Then, we conduct auxiliary experiments to valuate the performances of the components of MOGCN.

4.1. Datasets

We adopt three citation networks (Cora, Citeseer, and Pubmed.)[1] Every network dataset has a node attribute matrix **X** and a graph structure matrix **A**. The datasets are summarized in Table 1, where the nodes represent the publications and the edges

[1] https://linqs.soe.ucsc.edu/data.

Table 1
Statistics of the citation network datasets.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3

represent the citation links, the features of each node are bag-of-words representations of the corresponding publications, and the classes are the number of clusters.

4.2. Comparison with state-of-the-art algorithms

4.2.1. Baseline methods

To evaluate the performance of the proposed MOGCN, we compare it with the following methods: (1) LP [45]: label propagation is a traditional graph-based semi-supervised learning method; (2) GCN [23]: classic graph convolutional neural networks; (3) Co-Training: a method that trains GCN with a random walk model to extend the labeled dataset; (4) Self-Training: GCN with simple self-training to extend the labeled dataset; (5) Union: a method that expands the label set with the most confident predictions that are found in the random walk and self-training; (6) Intersection: a method that adds the most confident predictions that are found in both the random walk and the self-training to the labeled set. Note that methods (3)–(6) are proposed in [26]. (7) MultiStage [36] and its a variant (8) M3S: methods that are based on aligning mechanism on the embedding space and then extend the labeled dataset; (9) MixHop [2]: a method that uses higher-order adjacency matrices for node feature aggregation with a simply splicing operator.

4.2.2. Parameter settings

In the proposed MOGCN, we set the maximum iteration number *max_iter* to 500. Similar to [23], the GCN learner in our method with a 16-neuron hidden layer. We train our model by using the full batch in each training epoch and implement our algorithm in TensorFlow [1], and we optimize it with the Adam [22] algorithm. Moreover, we set the learning rate as 0.001, and the dropout rate as 0.5×10^{-4}. Following [26,36], we conduct experiments on the following label rates: 0.5%, 1%, 2%, 3%, and 4% on Citeseer and Cora datasets, and 0.03%, 0.05%, and 0.1% on the Pubmed dataset. We report the mean classification accuracy (ACC) on the 1000 test nodes of our method after 20 runs over the dataset splits that are specified above. The hyperparameter η in Eq. (4) is selected in the search range $\{0, 0.05, 0.1, \ldots, 1\}$ and the number of mixed-orders m is selected among $\{2, 3, \ldots, 6\}$.

4.2.3. Results

The semi-supervised node classification results of the baseline methods and MOGCN under optimal hyperparameters (η and m) are reported in Tables 2–4. We obtain the following observations: (i) On the Cora and Pubmed datasets, GCN is outperformed by label propagation (LP) when there are realtively few labeled training nodes. Since GCN is restricted by an oversmoothing issue that limits its ability to propagate the label information to distant nodes. (ii) Our model outperforms methods that expand the labeled training set with unlabeled nodes, such as Co-Training, Self-Training, Union, Intersection, M3S, and MultiStage. Since the pseudo-labels of unlabeled nodes will inevitably add the label noise to the labeled training dataset, especially when labeled nodes are scarce or in the initial training iterations. In contrast, our model ensures the accuracy of the originally labeled nodes and utilizes the pseudo-labels to increase the diversity of the learners, which renders our method effective. (iii) Compared with the MixHop model, which also utilizes high-order matrices, our proposed model is more effective. Because our model can directly learn representations of a node from the various receptive fields, and the

Table 2
Experimental results of SSL on the Cora dataset.

Label Rate	0.5%	1%	2%	3%	4%
LP	56.4	62.3	65.4	67.5	69.0
GCN	50.9	62.3	72.2	76.5	78.4
Co-Training	56.6	66.4	73.5	75.9	78.9
Self-Training	53.7	66.1	73.8	77.2	79.4
Union	58.5	69.6	75.9	78.5	80.4
Intersection	49.7	65.0	72.9	77.1	79.4
MultiStage	61.1	63.7	74.4	76.1	77.2
M3S	61.5	67.5	75.6	77.8	78.0
MixHop	51.2	62.9	73.3	77.3	79.8
MOGCN	**62.8**	**71.5**	**76.1**	**79.8**	**82.4**

Table 3
Experimental results of SSL on the Citeseer dataset.

Label Rate	0.5%	1%	2%	3%	4%
LP	34.8	40.2	43.6	45.3	46.4
GCN	43.6	55.3	64.9	67.5	68.7
Co-Training	47.3	55.7	62.1	62.5	64.5
Self-Training	43.3	58.1	68.2	69.8	70.4
Union	46.3	59.1	66.7	66.7	67.6
Intersection	42.9	59.1	68.6	70.1	70.8
MultiStage	53.0	57.8	63.8	68.0	69.0
M3S	56.1	62.1	66.4	70.3	70.5
MixHop	44.2	56.7	66.1	68.8	70.2
MOGCN	**58.9**	**62.8**	**68.6**	**71.6**	**72.4**

Table 4
Experimental results of SSL on the Pubmed dataset.

Label Rate	0.03%	0.05%	0.1%
LP	61.4	66.4	65.4
GCN	60.5	57.5	65.9
Co-Training	62.2	68.3	72.7
Self-Training	51.9	58.7	66.8
Union	58.4	64.0	70.7
Intersection	52.0	59.3	69.4
MultiStage	57.4	64.3	70.2
M3S	59.2	64.4	70.6
MixHop	61.6	59.1	67.2
MOGCN	**63.2**	**68.5**	**76.7**

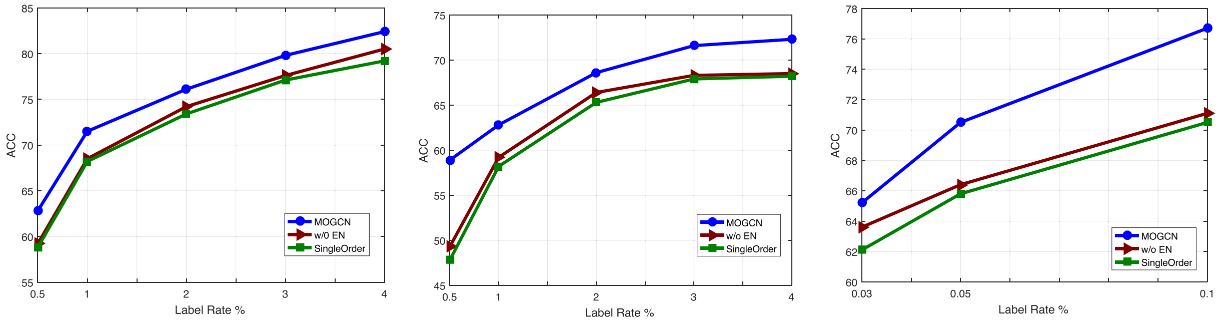
novel ensemble module can reduce the redundancy among learners. The above observations demonstrate that the effectiveness of our proposed method.

4.3. Ablation study

In this section, we examine the performance of each component of MOGCN, namely, the GCN-based learners module and the ensemble module, via an ablation study. Here, we consider two variants of our model:

- **(1) SingleOrder**: *SingleOrder* represents the GCN that is trained on the single order adjacency matrix; and we report the best result under the single order (among $\{1, 2, 3, \ldots, 6\}$) adjacency matrix.
- **(2) w/o EN**: w/oEN represents the method that combines the results of the GCN-based learners via a simple voting strategy without using the proposed ensemble module, and we report the experimental results of the optimal combination.

We compare the two variants with our proposed model MOGCN, which consists of both important components. The comparison result is presented in Fig. 3. According to this result, The w/oEN method is slightly outperforms *SingleOrder* in clas-



(a) Cora dataset. (b) Citeseer dataset. (c) Pubmed dataset.

Fig. 3. Illustration of the performances of variants of MOGCN at various label rates.

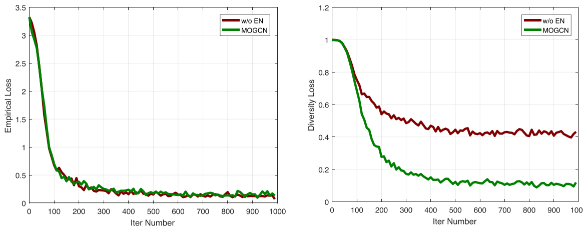
sification. This is because there is redundant information among the GCN learners that are learned on the adjacency matrices of various orders in w/oEN, which affects the performance improvement. Hence, simply combining the results of multiple GCN learners is not a satisfactory strategy. In contrast, the proposed MOGCN model can realize substantially increased classification performance because it not only directly considers multiple high-order connections among nodes by constructing multiple simple GCNs, but also augments the diversity among the learners.

Then, to explore the ensemble module in more detail, we examine the empirical loss and the diversity loss of w/oEN and MOGCN on the Cora dataset in Fig. 4a and 4b, respectively. We report the experimental results under a 1% of label rate, with 3 GCN learners, and $\eta = 0.2$. For w/oEN, we calculate only the value of the diversity loss, and the diversity loss term does not participate in optimization. According to this result, both methods realize similar accuracy on labeled nodes, while MOGCN realizes a lower diversity loss than w/oEN. The lower the diversity loss is, the larger the differences among learners, which shows that the proposed method can indeed maximize the accuracy of the base learners on the labeled nodes while maximizing the diversity among them on the unlabeled nodes.

We also conduct an experiment to explore the impact of the rate of unlabeled nodes in the ensemble module. Here, we fix the rate of labeled nodes and set the rates of the remaining unlabeled nodes to range from 0% to 100%. As shown in Fig. 5, the use of many unlabeled nodes tends to prodece higher accuracy than the use of few unlabeled nodes for a fixed number of labeled nodes. This result demonstrates that the pseudo-labels of unlabeled nodes are helpful for realizing higher performance.

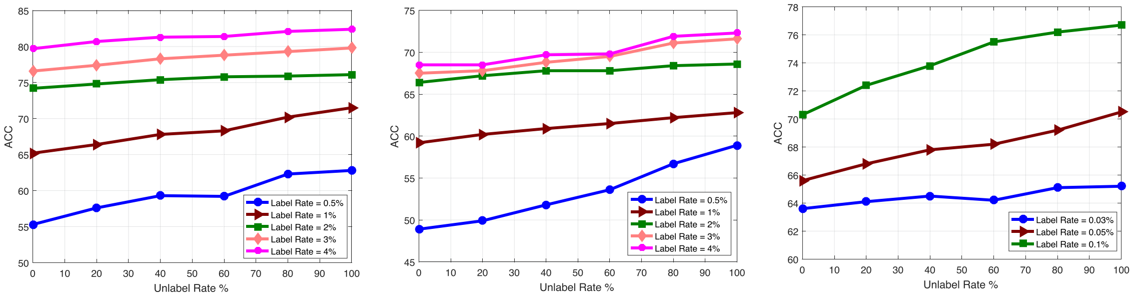
4.4. Parameter sensitivity

Parameter m controls the number of adjacency matrices of different orders in MOGCN. In this part, the 2nd-order, 3rd-order, 4th-order, 5th-order, and 6th-order MOGCN are compared. The results are presented in Table 5, according to which, the 3rd-order and 4th-order MOGCN models provide comparably satisfactory performance. However, as the order continues to increase, the range of neighborhoods also increases, and the classification performance of the algorithm starts to decrease. This phenomenon is consistent with our intuition, because the higher the order is, the farther the node's direct neighbors will be, thereby increasing the risk of mixing node features from other categories. Consequently, to realize satisfactory classification performance and computational efficiency, we suggest that the number of orders m of our proposed method be set to 3 or 4.



(a) Empirical Loss (b) Diversity Loss

Fig. 4. Illustration of the empirical loss and the diversity loss of the methods with and without the ensemble module on the Cora dataset.



(a) Cora dataset. (b) Citeseer dataset. (c) Pubmed dataset.

Fig. 5. Illustration of the performance of MOGCN at various unlabeled rates in the ensemble module.

Table 5
Experimental results of the order-number.

Method	Cora					Citeseer					Pubmed		
	0.5%	1%	2%	3%	4%	0.5%	1%	2%	3%	4%	0.03%	0.05%	0.1%
2nd-order	61.9	70.8	75.7	79.2	82.0	57.2	62.0	67.8	70.4	72.1	62.9	67.5	76.1
3rd-order	**62.8**	71.3	**76.1**	**79.8**	**82.4**	58.3	**62.8**	68.4	**71.6**	**72.4**	63.1	68.1	**76.7**
4th-order	61.2	**71.5**	75.8	79.1	82.2	**58.9**	62.2	**68.6**	71.3	72.1	**63.2**	**68.5**	75.9
5th-order	61.1	70.7	75.3	78.2	82.0	56.2	60.5	67.4	70.5	71.4	63.0	67.4	75.4
6th-order	60.2	68.9	74.8	77.4	81.2	55.9	58.3	66.7	69.5	71.2	62.4	67.2	74.2

Parameter η controls the correlation between the empirical loss and the diversity loss. When setting $\eta = 0$, our model is equivalent to training each GCN learner independently, and when the value of η increases, increasing emphasis is placed on minimizing the diversity loss; thus, the differences among the base learners will increase. However, employing a larger value for η will overemphasize the effect of diversity and lead to poor performance. We empirically find that setting η to a smaller value, i.e., $\eta \in [0.1, 0.3]$ usually leads to satisfactory results.

5. Conclusions and future work

In this paper, we propose mixed-order graph convolutional networks (MOGCN), which is a novel end-to-end ensemble framework that has two advantages: (1) The proposed framework constructs multiple simple GCN learners with adjacency matrices of various orders and ensembles the results, which can directly capture various high-order connectivities among nodes and alleviate the problem of oversmoothing. (2) In the ensemble module, the pseudo-labels of unlabeled nodes are exploited to help augment the diversity of the base learners; via this strategy, unlabeled nodes are fully utilized. Our model achieves state-of-the-art results and enables us to balance the accuracy of the labeled nodes and the diversity of the base learners that are obtained by the unlabeled nodes. Moreover, our method is general. Thus, we can combine more sophisticated models, such as the recently proposed GAT [39] or disentangled GCN [32], with our ensemble module.

CRediT authorship contribution statement

Jie Wang: Conceptualization, Methodology, Software, Visualization, Writing - original draft. **Jianqing Liang:** Methodology, Project administration, Writing - review & editing. **Junbiao Cui:** Writing - review & editing, Software. **Jiye Liang:** Supervision, Methodology.

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.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (2020AAA0106100), the National Natural Science Foundation of China (Nos. 62006147, 61876103).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., Tensorflow: a system for large-scale machine learning, in: Proceedings of the Symposium on Operating Systems Design and Implementation, 2016, pp. 265–283..
[2] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan, Mixhop: higher-order graph convolutional architectures via sparsified neighborhood mixing, in: Proceedings of the International Conference on Machine Learning, 2019, pp. 21–29..
[3] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, pages 1993–2001, 2016..
[4] Yoshua Bengio, Aaron C. Courville, Pascal Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun, Spectral networks and locally connected networks on graphs, in: International Conference on Learning Representations, 2014..
[6] Adrian Calma, Tobias Reitmaier, Bernhard Sick, Semi-supervised active learning for support vector machines: a novel approach that exploits structure information in data, Inf. Sci. 456 (2018) 13–33.
[7] Yan Chen, Keyu Liu, Jingjing Song, Hamido Fujita, Xibei Yang, Yuhua Qian, Attribute group for attribute reduction, Inf. Sci. 535 (2020) 64–80.
[8] Wei Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho Jui Hsieh, Cluster-gcn: an efficient algorithm for training deep and large graph convolutional networks, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, 2019, pp. 257–266.
[9] Fabio G. Cozman, Ira Cohen, Unlabeled data can degrade classification performance of generative classifiers, in: Proceedings of the International Florida Artificial Intelligence Research Society Conference, 2002, pp. 327–331.

[10] Michal Defferrard, Xavier Bresson, Pierre Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 3837–3845.

[11] F. Dornaika, A. Baradaaji, Y. El Traboulsi, Semi-supervised classification via simultaneous label and discriminant embedding estimation, Inf. Sci. 546 (2021) 146–165.

[12] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 2224–2232..

[13] Fu. Sichao, Weifeng Liu, Dapeng Tao, Yicong Zhou, Liqiang Nie, Hesgcn: Hessian graph convolutional networks for semi-supervised classification, Inf. Sci. 514 (2020) 484–498.

[14] Can Gao, Jie Zhou, Duoqian Miao, Jiajun Wen, Xiaodong Yue, Three-way decision with co-training for partially labeled data, Inf. Sci. 544 (2021) 500–518.

[15] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, George E. Dahl, Neural message passing for quantum chemistry, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 1263–1272.

[16] Will Hamilton, Zhitao Ying, Jure Leskovec, Inductive representation learning on large graphs, in: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.

[17] Zehua Jiang, Keyu Liu, Xibei Yang, Hualong Yu, Hamido Fujita, Yuhua Qian, Accelerator for supervised neighborhood based attribute reduction, Int. J. Approx. Reasoning 119 (2020) 122–150.

[18] Zhao Kang, Lu. Xiao, Jian Liang, Kun Bai, Xu. Zenglin, Relation-guided representation learning, Neural Netw. 131 (2020) 93–102.

[19] Zhao Kang, Haiqi Pan, Steven C.H. Hoi, Xu. Zenglin, Robust graph learning from noisy data, IEEE Trans. Cyber. 50 (5) (2020) 1833–1843.

[20] Zhao Kang, Chong Peng, Qiang Cheng, Xinwang Liu, Xi Peng, Xu. Zenglin, Ling Tian, Structured graph learning for clustering and semi-supervised classification, Pattern Recogn. 110 (2021) 107627.

[21] Donghwa Kim, Deokseong Seo, Suhyoun Cho, Pilsung Kang, Multi-co-training for document classification using various document representations: tf-idf, lda, and doc2vec, Inf. Sci. 477 (2019) 15–29.

[22] Diederik P. Kingma, Jimmy Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations, 2015.

[23] Thomas N. Kipf, Max Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.

[24] Anders Krogh, Jesper Vedelsby, Neural network ensembles, cross validation, and active learning, in: Proceedings of the Advances in Neural Information Processing Systems, 1995, pp. 231–238.

[25] Yann Lecun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[26] Qimai Li, Zhichao Han, and Xiao Ming Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, pp. 3538–3545..

[27] Keyu Liu, Xibei Yang, Hamido Fujita, Dun Liu, Xin Yang, Yuhua Qian, An efficient selector for multi-granularity attribute reduction, Inf. Sci. 505 (2019) 457–472.

[28] Keyu Liu, Xibei Yang, Yu. Hualong, Hamido Fujita, Xiangjian Chen, Dun Liu, Supervised information granulation strategy for attribute reduction, Inter. J. Mac. Learn. Cybern. 11 (9) (2020) 2149–2163.

[29] Ying Liu, Xu. Zhen, Chunguang Li, Distributed online semi-supervised support vector machine, Inf. Sci. 466 (2018) 236–257.

[30] Yong Liu, Xin Yao, Ensemble learning via negative correlation, Neural Netw. 12 (10) (1999) 1399–1404.

[31] Yong Liu, Xin Yao, Tetsuya Higuchi, Evolutionary ensembles with negative correlation learning, IEEE Trans. Evol. Comput. 4 (4) (2000) 380–387.

[32] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu, Disentangled graph convolutional networks, in: Proceedings of the International Conference on Machine Learning, 2019, pp. 4212–4221..

[33] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein, Geometric deep learning on graphs and manifolds using mixture model CNNs, in: Proceedings of the Internaltional Conference on Computer Vision and Pattern Recogintion, 2017, pp. 5425–5434..

[34] B.M. Shahshahani, D.A. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon, IEEE Trans. Geosci. Remote Sensing 32 (5) (1994) 1087–1095.

[35] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, Mohammad Ali Zare Chahooki, A robust graph-based semi-supervised sparse feature selection method, Inform. Sci. 531 (2020) 13–30.

[36] Ke Sun, Zhouchen Lin, Zhanxing Zhu, Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 5892–5899.

[37] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In Proceedings of the International Conference on World Wide Web, pages 1067–1077, 2015..

[38] Jesper E Van Engelen, Holger H Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (2020) 373–440.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018..

[40] Petar Velikovi, William Fedus, William L Hamilton, Pietro Li, Yoshua Bengio, and R Devon Hjelm, Deep graph infomax, in: International Conference on Learning Representations, 2019..

[41] Jingjuan Wang, Qingkui Chen, Huilin Gong, Stmag: a spatial-temporal mixed attention graph-based convolution model for multi-data flow safety prediction, Inf. Sci. 525 (2020) 16–36.

[42] Shuo Wang, Huanhuan Chen, Xin Yao, Negative correlation learning for classification ensembles, in: Proceedings of the International Joint Conference on Neural Networks, 2010, pp. 1–8.

[43] Wei Wang and Zhi Hua Zhou. A new analysis of co-training. In Proceedings of the International Conference on Machine Learning, pages 1135–1142, 2010..

[44] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, P Yu, and Yanfang Ye, Heterogeneous graph attention network, in: Proceedings of the International Conference on World Wide Web, 2019, pp. 2022–2032..

[45] Xiao Ming Wu, Zhenguo Li, Anthony M So, John Wright, and Shih Fu Chang, Learning with partially absorbing random walks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 3077–3085..

[46] Liang Yao, Chengsheng Mao, Yuan Luo, Graph convolutional networks for text classification, in: Proceedings of The AAAI Conference on Artificial Intelligence, 2019, pp. 7370–7377.

[47] Le Zhang, Zenglin Shi, Ming Ming Cheng, Yun Liu, Jia Wang Bian, Joey Tianyi Zhou, Guoyan Zheng, Zeng Zeng, Nonlinear regression via deep negative correlation learning, IEEE Trans. Pattern Anal. Mach. Intell. 43 (3) (2021) 982–998.

[48] Xiaojin Zhu, Zoubin Ghahramani, John D Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 912–919.