Artificial Intelligence 307 (2022) 103708

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Multi-view graph convolutional networks with attention mechanism

Kaixuan Yao [a], Jiye Liang [a,*], Jianqing Liang [a], Ming Li [b], Feilong Cao [c]

[a] *Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education and the School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China*
[b] *Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Jinhua 321004, Zhejiang, China*
[c] *Department of Applied Mathematics, College of Sciences, China Jiliang University, Hangzhou 310018, Zhejiang, China*

ARTICLE INFO

Article history:
Received 5 July 2020
Received in revised form 5 August 2021
Accepted 12 March 2022
Available online 18 March 2022

Keywords:
Graph neural networks
Multi-view learning
Attention mechanism
Semi-supervised learning

ABSTRACT

Recent advances in graph convolutional networks (GCNs), which mainly focus on how to exploit information from different hops of neighbors in an efficient way, have brought substantial improvement to many graph data modeling tasks. Most of the existing GCN-based models however are built on the basis of a fixed adjacency matrix, i.e., a single view topology of the underlying graph. That inherently limits the expressive power of the developed models especially when the raw graphs are often noisy or even incomplete due to the inevitably error-prone data measurement or collection. In this paper, we propose a novel framework, termed Multi-View Graph Convolutional Networks with Attention Mechanism (MAGCN), by incorporating multiple views of topology and an attention-based feature aggregation strategy into the computation of graph convolution. As an advanced variant of GCNs, MAGCN is fed with multiple "trustable" topologies, which already exist for a given task or are empirically generated by some classical graph construction methods, which has good potential to produce a better learning representation for downstream tasks. Furthermore, we present some theoretical analysis about the expressive power and flexibility of MAGCN, which provides a general explanation as to why multi-view based methods can potentially outperform those relying on a single view. Our experimental study demonstrates the state-of-the-art accuracies of MAGCN on Cora, Citeseer, and Pubmed datasets. Robustness analysis is also undertaken to show the advantage of MAGCN in handling some uncertainty issues in node classification tasks.

1. Introduction

Many scientific fields in artificial intelligence (AI) study graph structure data that is a non-Euclidean space, for example, an airline network connecting different areas, the transmission of a virus during an epidemic outbreak, social networks in computational social sciences [1], molecular structures, and so on. With the development and explosion of deep learning [2], AI has made major progress in key domains such as computer vision [3,4], natural language processing (NLP) [5], and urban computing [6,7]. This is mainly due to the underlying Euclidean or grid-like structure of data, and cases where the

[*] Corresponding author.
E-mail addresses: yaokx2@gmail.com (K. Yao), ljy@sxu.edu.cn (J. Liang), liangjq@sxu.edu.cn (J. Liang), mingli@zjnu.edu.cn (M. Li), icteam@163.com (F. Cao).

https://doi.org/10.1016/j.artint.2022.103708
0004-3702/© 2022 Elsevier B.V. All rights reserved.

invariances of these structures are built into deep neural networks (DNNs). However, the irregularity and complexity of graph data impose significant challenges on existing deep learning based models, largely because each graph has a different size of unordered nodes and each node has a different number of neighbors, causing the important discrete convolution operation, which is easy to compute in the image domain (Euclidean space), but is not directly applicable to the graph domain (non-Euclidean space).

Recently, graph neural networks [8,9], particularly graph convolutional networks (GCNs) [10] have received careful attention in light of their favorable performance on many graph data modeling tasks. So far, various methods are investigating how to define an effective convolution operation with the ability of feature extraction from different hops of neighbors. Technically, generalizing the convolution operator to the graph domain can be expressed typically as a neighborhood aggregation or message passing scheme [11]. Here, we omit a comprehensive review of the existing models. Interested readers can refer to survey papers [12,13] for more details.

Even though GCN and its variants/extensions have achieved great success on node classification tasks, almost all of these models have been developed based on a fixed adjacency matrix given in advance, in other words, a single view graph topology. Inherently, the expressive power of the resultant model may be limited due to the potential information discrepancy between the adjacency matrix and the (unknown) target one. Here we give a simple example as a reflection on this issue. The given adjacency matrix of Cora, one of the widely used benchmark datasets for node classification tasks, is simply formulated based on the practical citation outcome between two papers. In terms of the technical components, papers $\#i$ and $\#j$ are on completely different topics, for example computer sciences and chemistry, while the citation between them is barely because the algorithm/model developed in paper $\#i$ is used in $\#j$ for special application. This means, apart from the given adjacency matrix, there are other possible viewpoints to better represent the relation between papers $\#i$ and $\#j$. As such, it is logical to consider one practical question, that is, *how to carry out neighborhood aggregation or message passing when multi-view topologies of the graph are provided in advance?*

In this paper, we develop a novel model called Multi-View Graph Convolutional Networks with Attention Mechanism (MAGCN) to cope with the aforementioned question. Our initial idea is, for a given graph modeling task, the underlying graph topology is unknown in advance and the current adjacency matrix A is just one approximation of the 'ideal' one, which unavoidably leads to a certain degree of information loss (due to the inevitably error-prone data measurement or collection). Therefore, multiple adjacency matrices representing multi-view topologies of the graph can certainly work more favorably as they offer more reliable relation representations among the nodes. This means, in addition to A, it is necessary to explore different edge information (e.g. based on the feature similarity between two nodes) to obtain a set of adjacency matrices $A_1, A_2, \ldots, A_n$ for problem solving. It should be noted that our main purpose is to propose a general multi-view graph representation learning framework by incorporating multiple views of topology and an attention-based feature aggregation strategy into the computation of graph convolution, which can potentially obtain a better graph representation than a vanilla GCN with the given single-view. This means our work starts from the premise that multiple graph structures, which already exist for a given task or are empirically generated by some classical graph construction methods, are given in advance. Readers who are interested in how to construct the graph structure manually or (adaptively) learn the graph topology can refer to related works, such as kNN graph construction [14], b-Matching graph construction [15], low-rank graph learning [16], and references therein.

Recall that our primary focus is to develop a flexible framework that makes use of multiple graph topologies that are given in advance for problem-solving, we perform a similar way as the squeeze-and-excitation block for conventional CNN [17] to produce unified node-level feature extraction. First, given multiple adjacency matrices representing various views of the graph structure, a multi-GCN module is developed by employing the (vanilla) GCN convolution block for each view. Then, a multi-view graph attention module consisting of a graph global average pooling (graph GAP) block and a MLP block, is proposed to fuse the multiple node-level learning representations into a weighted form with attention coefficients as assigning different importance to different views. Finally, a merging module is introduced to obtain a unified (new) feature representation for each node. We also theoretically analyze the expressive power of MAGCN, indicating its flexibility (compared with the GCN model) gained by the view sufficiency and the attention mechanism. Our experiment results achieve state-of-the-art performance on three benchmark datasets for node classification. Interestingly, the advantage of MAGCN in dealing with problems with topology attacks is demonstrated in comparison with GCN [10] and GAT models [18]. The code is available on GitHub.[1]

In summary, our contribution is three-fold:

- A new framework for multi-view topologies of the graph is proposed to build a novel GCN model with an enhanced learning representation ability.
- The superiority of MAGCN is theoretically verified by bounding its expressive power with indicators reflecting the view insufficiency and the flexibility induced by attention coefficients among different views.
- Our proposed method shows good robustness for node classification problems with topology attacks, showing good potential to deal with uncertainty issues in other kinds of graph data modeling problems.

[1] https://sxu-yaokx.github.io/MAGCN/.

The rest of this paper is organized as follows. In Section 2, we start by reviewing some related work. In Section 3, we introduce the notations and preliminaries used in this paper. The framework and technical details of our proposed model are described in Section 4. In Section 5, we present a theoretical analysis of the expressive power and flexibility of the proposed MAGCN. In Section 6, we present extensive experimental results to demonstrate the effectiveness of the proposed model. Finally, we conclude our work in Section 7.

2. Related work

The framework of graph convolutional neural networks is inspired by the formulation of graph convolution with graph Fourier transforms under the orthogonal basis of the graph Laplacian [9]. Basically, research in this direction proceeds along two directions, that is, spatial-based approaches with prominent examples like diffusion convolutional neural networks (DCNN) [19], Graph-SAGE [20], MoNet [21], MPNN [11], graph isomorphism networks (GIN) [22]; and spectral-based approaches like the widely cited representatives GCN [10] and ChebNet [23]. Since our work is inspired by GCN, we review the follow-up work, which can be summarized technically from several perspectives: (i) improvements focusing on sampling issues to accelerate the training process of GCN, such as variance reduction-based GCN [24], FastGCN with importance sampling [25], and Cluster-GCN [26]; (ii) extensions with a new formulation of spectral graph convolution with fast transforms, e.g. HANet [27] and graph wavelet neural networks [28], and/or generalization of the graph Laplacian to maximal entropy transition matrix derived from a path integral, e.g. PAN [29]; (iii) models using multi-scale information and a higher order adjacency matrix for node feature aggregation [30–32]; (iv) models that employ attention mechanisms to effectively learn the importance between nodes and their neighbors [18,33]; (v) variants with special concerns for advanced node representations such as disentangled GCN [34]; (vi) theoretical insights and explorations on several characteristics of GCN [35–38].

Clearly, our work is not the only one to consider the attention mechanism and multi-view topology information in the computation of graph convolution. Various such attempts have already appeared in the literature. Veličković et al. [18] introduced an attention-based architecture to perform node classification of graph-structured data (GAT), which computes the latent representations of each node in the graph by attending over its neighbors, following a self-attention strategy. Then Wang et al. [39] developed a variant of the GAT to learn the hidden representations of each node by attending over its neighbors (DAEGC) to combine the attribute values with the graph structure in the latent representation. However, both the GAT and DAEGC models utilize the self-attention strategy over each node and its neighbors on a single view topology, which inherently limits the expressive power of the developed models when the given adjacency matrix does not fully reflect the 'ideal' structure knowledge. Ma et al. [40] proposed a multi-dimensional GCN model that can effectively capture sufficient information in learning node-level representations for multi-dimensional graphs that are equipped with multiple types of relations among nodes. Their framework uses a weighted average to combine the representations from the within-dimension aggregation and across dimension aggregation, which however may require some prior knowledge to set the hyper-parameters (used to control the importance of each component) appropriately, showing less flexibility than a learning-based/attention-based combination approach. Khan et al. [41] introduced a Multi-GCN model by merging multiple topologies to a single topology and feeding the fused view to a vanilla GCN block. Their fusion process of multiple views, consisting of subspace merging and subsequent manifold ranking, can be viewed as a data-preprocessing step rather than an end-to-end framework. Generally, the multi-dimensional GCN [40] model and the Multi-GCN [41] model show empirically the potential advantages of considering multiple topologies in graph convolution. The theoretical analysis on the expressive power of multi-view graph learning, however, is missing in their works.

Another straightforward method is to construct or learn an appropriate topology, as attempted in [42,43]. Technically, they introduced a hybrid cost function ($L = L_1 + \lambda L_2$) that integrates node representation learning (driven by the graph convolution layer objective L_1) and graph structure learning (as a regularization term L_2) in a unified framework. This allows the graph structure to be refined adaptively during the whole learning process of the GCN model. From the perspective of algorithmic implementation, the final performance of a GCN model trained on this kind of regularized learning scheme relies heavily on the choice of regularization coefficient λ. On the other hand, the widely used alternative optimization manner to find a local minimum solution of L generally leads to higher computational cost, and importantly, the convergence property cannot be guaranteed theoretically.

3. Preliminaries

Definition. $G = \{V, E, A\}$ denotes an undirected graph, where V is the set of nodes with $|V| = N$, E is the set of edges, and A is the adjacency matrix indicating the unique topology of G. A multi-view graph is denoted as $G = \{V, (E_1, A_1), (E_2, A_2), \dots, (E_n, A_n)\}$, where $A_i \neq A_j$ when $i \neq j$, n is the number of views. Since the graph structure is mainly described with its adjacency matrix, for convenience, we simplify the formulation of a multi-view graph as $G = \{V, A_1, A_2, \dots, A_n\}$.

Graph convolutional networks. In the non-Euclidean domain, graph convolution is defined by the spectral graph theory [44] and convolution theorem. For a graph signal $x \in \mathbb{R}^N$, the graph Fourier transform (GFT) is defined as $\hat{x} = U^{\mathrm{T}} x$, and the inverse GFT is $x = U\hat{x}$ [45], where U denotes the matrix of eigenvectors of the normalized graph Laplacian matrix $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$,

D and A denote the degree matrix and adjacency matrix of the graph, respectively. Then, the graph convolution operator $*_{\mathcal{G}}$ is defined in the graph Fourier domain (see [9])

$$x \star_{\mathcal{G}} y = U((U^{\mathrm{T}}x) \odot (U^{\mathrm{T}}y)), \tag{1}$$

where $\odot$ denotes the Hadamard product. Further, the graph convolution can be formulated as a filter g_θ multiplying the signal x

$$g_\theta \star x = g_\theta(L)x = g_\theta(U\Lambda U^{\mathrm{T}})x = Ug_\theta(\Lambda)U^{\mathrm{T}}x, \tag{2}$$

where $\star$ is the convolution operator. To reduce the computation complexity, the Chebyshev expansion is utilized in graph signal processing to approximate graph kernels [46]. Defferrard et al. [23] use the Chebyshev polynomials to express the graph convolution

$$g_\theta \star x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x, \tag{3}$$

where $\tilde{L} = 2L/\lambda_{\max} - I$ denotes the scaled Laplacian matrix, and $\lambda_{\max}$ denotes the largest eigenvalue. $T_k \in \mathbb{R}^{N\times N}$ is the Chebyshev polynomial of order k, and $\theta \in \mathbb{R}^K$ is the Chebyshev coefficient. The introduction of the Chebyshev polynomials reduces the complexity of the graph convolution from $\mathcal{O}(N^2)$ (see Eq. (2)) to $\mathcal{O}(K|\varepsilon|)$ (Eq. (3)). Kipf et al. [10] further approximate the maximum eigenvalue $\lambda_{\max} \approx 2$ and introduce several tricks to generalize the graph convolution to a signal $X \in \mathbb{R}^{N\times M}$, where M is the dimension of the feature vector for each node, and F filters

$$\tilde{X} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW = \hat{A}XW, \tag{4}$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, $\tilde{A} = A + I$ (add self-loops in the adjacency matrix A) and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $W \in \mathbb{R}^{M\times F}$ denotes the trainable weight matrix of the filters. Most graph convolutional networks (GCNs) are based on Eq. (4).

4. Multi-view graph convolutional network with attention mechanism

Given a graph $G = \{V, A\}$, the set V comprises N nodes, and each node includes a feature vector $x \in \mathbb{R}^M$. Then let $X \in \mathbb{R}^{N\times M}$ denote the feature matrix of G. The GCN-based methods output a new representation for each node when given $G = \{V, X, A\}$. Actually, the pivotal idea in each layer of the general GCN is

$$Y = f(\hat{A}XW), \tag{5}$$

where $f(\cdot)$ denotes the activation function, e.g., $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$ [47]. The output $Y \in \mathbb{R}^{N\times F}$ is the new feature representation matrix.

Unfortunately, in practical terms, a single view (topology) is insufficient to describe one problem. Let P_n denote a view generated from the latent complete space $\mathcal{S}$. According to the information theory, the view insufficiency of view P_n is formulated as conditional mutual information [48]

$$I(S; P_n|P_{n-1}, \ldots, P_1) \geq \varepsilon_{\text{info}}, \tag{6}$$

where S denotes the information contained in $\mathcal{S}$. $\varepsilon_{\text{info}}$ is a positive parameter. Eq. (6) measures how much co-occurrence information is shared between S and P_n when $P_{n-1}, \ldots, P_1$ are already known. From Eq. (6), it can be seen that each view contains distinctive information about S. It is natural to use multiple topologies to learn graph representation. However, once the single view graph $G = \{V, X, A\}$ becomes a multi-view type $G^* = \{V, X, A_1, A_2, \cdots, A_n\}$, the traditional GCN-based methods can not handle this situation.

To overcome the above conundrum, we propose a Multi-View Graph Convolutional Network with Attention Mechanism (MAGCN). As shown in Fig. 1, the proposed MAGCN consists of two multi-GCN blocks (unfold and merge) and a multiview attention block.

Multi-GCN (unfold) block. The multi-view graph $G^* = \{V, X, A_1, A_2, \cdots, A_n\}$ is fed to the multi-GCN (unfold) block to achieve the new representations of all views

$$\begin{aligned} \hat{X}_i &= f_{\text{GCN}}(X, A_i) = \mathrm{ReLU}(\hat{A}_i X W_i), \\ \mathcal{X} &= \{\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_n\}, \end{aligned} \tag{7}$$

where $\hat{X}_i \in \mathbb{R}^{N\times F}$ is the new representation of the i-th view, N is the number of nodes, F is the dimension of the feature vector. $\mathcal{X} \in \mathbb{R}^{n\times N\times F}$ denotes the new representation of the multi-view graph G^*, and n denotes the number of views. As Fig. 1 shows, the raw feature matrix $X \in \mathbb{R}^{N\times M}$ of multi-view graph G^* is represented as a feature tensor $\mathcal{X} \in \mathbb{R}^{n\times N\times F}$.

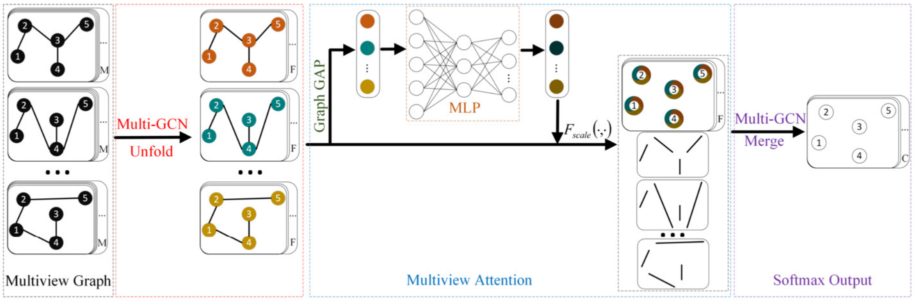

Fig. 1. The overall structure of our MAGCN. The multi-view graph G^* with 5 nodes, n topologies and a feature matrix $X \in \mathbb{R}^{5\times M}$, is first expressed by the multi-GCN (unfold) block to obtain a multi-view representation $\mathcal{X} = \{\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_n\} \in \mathbb{R}^{n\times 5\times F}$. Then a multi-view attention is utilized to fuse the $\mathcal{X}$ to a complete representation $\bar{X} \in \mathbb{R}^{5\times F}$. Finally, a multi-GCN (merge) block with softmax is introduced to obtain the final classification expressing matrix $X^* \in \mathbb{R}^{5\times C}$.

To explain further, for the graph signal G^* with n views, the raw feature in M-dimensional space is embedded to n F-dimensional spaces. In particular, each F-dimensional space carries some unique information.

Attention block. The attention block aggregates all new representations from n F-dimensional spaces (views), which comprises two stages: the identity stage and the attention distribution learning stage. The identity stage directly maps the new representation tensor $\mathcal{X}$ to the $F_{scale}(\cdot,\cdot)$, see Eq. (10). The attention distribution learning stage comprises a graph global average pooling (graph GAP) module and a multilayer perceptron (MLP) module. The global average pooling (GAP) [49] is widely used in convolutional neural networks to capture the global feature from a feature map (see the left panel of Fig. 2), which is formulated as

$$\mathrm{f}_i = f_{\mathrm{GAP}}(\mathrm{F}_i) = \frac{1}{h\times w}\sum_{j=1}^{h}\sum_{k=1}^{w}\mathrm{F}_{i,jk}, \tag{8}$$

where $\mathrm{F}_i \in \mathbb{R}^{h\times w}$ denotes the feature map of i-th channel, and $h \times w$ denotes the spatial dimension of F_i. $\mathrm{f}_i \in \mathbb{R}$ is a statistic, which is the global feature generated by the GAP from the input feature map F_i. From Eq. (8) we can see that the traditional GAP is a simple arithmetic mean operation, which means each pixel $\mathrm{F}_{i,jk}$ makes an equal contribution to the global feature. However, different nodes in a graph have different significance. To consider the differences between different nodes, we propose a graph GAP (see the right panel of Fig. 2)

$$\hat{x}_i = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{|\mathcal{N}_{i,j}|}\sum_{k=1}^{|\mathcal{N}_{i,j}|}(I+A_i)_{jk}\hat{X}_{i,j,k}, \tag{9}$$

where $\mathcal{N}_{i,j}$ denotes the neighbor set of j-th node on i-th view. N denotes the number of nodes of graph G^*. $I + A_i$ is the adjacency matrix with self-loops of the i-th view and $\sum_{k=1}^{|\mathcal{N}_{i,j}|}(I+A_i)_{jk}\hat{X}_{i,j,k}$ is a graph aggregation operation. $\hat{x}_i$ is a statistic which denotes the global feature of the i-th view generated by the graph GAP. From Eq. (9), it can be seen that the graph GAP is no longer just a mean data aggregation as a traditional GAP. The biggest improvement of the proposed graph GAP is that the introduction of the graph aggregation operation $\sum_{k=1}^{|\mathcal{N}_{i,j}|}(I+A_i)_{jk}\hat{X}_{i,j,k}$. Each node is first aggregated by its neighbor nodes and itself, then all re-expressed nodes are aggregated by an arithmetic mean operation.

After the graph GAP, the representation of each view is aggregated as a statistic, which means the representation $\mathcal{X} = \{\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_n\} \in \mathbb{R}^{n\times N\times F}$ turns into $\dot{X} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n\} \in \mathbb{R}^n$. Then, the aggregated $\dot{X}$ is fed to an MLP module (see Fig. 1) to learn the view-wise weights vector $C = \{c_1, c_2, \cdots, c_n\} \in \mathbb{R}^n$ (equal to attention distribution coefficients), which can explicitly measure the correlation and importance distribution among different views. Importantly, the number of neurons of the last layer in MLP must match the number of views. Finally, the new representation $\mathcal{X}$ is aggregated by the $F_{scale}(\cdot,\cdot)$ with the attention distribution coefficients C

$$\bar{X} = F_{scale}(\mathcal{X}, C) = \sum_{i=1}^{n} c_i\hat{X}_i, \tag{10}$$

where $\bar{X} \in \mathbb{R}^{N\times F}$ stands for the representation aggregated by the attention block and n stands for the number of views.

Multi-GCN (merge) block. As Fig. 1 shows, the raw feature $X \in \mathbb{R}^{N\times M}$ is first expressed by the multi-GCN (unfold) block to obtain a new multi-view representation $\mathcal{X} = \{\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_n\} \in \mathbb{R}^{n\times N\times F}$. Then, the attention block considers the correlation among different views to obtain a more comprehensive semantic representation $\bar{X} \in \mathbb{R}^{N\times F}$. Finally, the semantic representation $\bar{X}$ is embedded to the C-dimensional categorical distribution space through the multi-GCN (merge) block

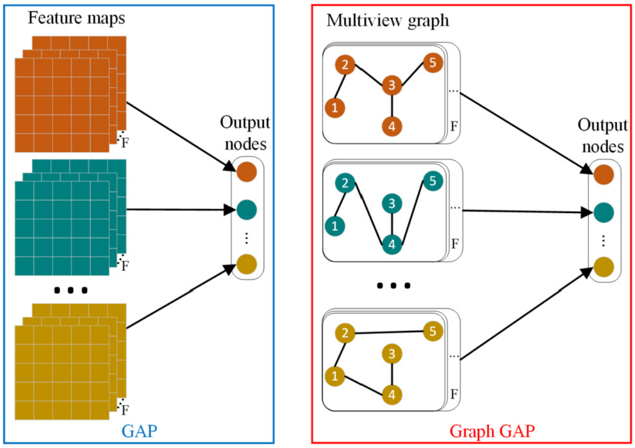

Fig. 2. The architectures of GAP and the proposed graph GAP.

$$X^* = \sum_{i=1}^{n} f_{\mathrm{GCN}}(\bar{X}, A_i) = \mathrm{softmax}(\sum_{i=1}^{n} \hat{A}_i \bar{X} W_i),$$

where $X^* \in \mathbb{R}^{N \times C}$ denotes the final category distribution. The softmax activation function is defined as

$$\mathrm{softmax}(X^*_{ij}) = \frac{\exp(X^*_{ij})}{\sum_{j=1}^{C} \exp(X^*_{ij})},$$

where C denotes the number of classes. For semi-supervised classification tasks, we choose the cross-entropy error as the loss function

$$\mathcal{L} = -\sum_{k \in V_L} \sum_{j=1}^{C} Y_{kj} \ln X^*_{kj},$$

where V_L denotes the set of labeled nodes. $Y \in \mathbb{R}^{|V_L| \times C}$ denotes the label indicator matrix.

Complexity analysis. For a single layer of the graph convolutional network (GCN) [10], with an F-dimensional input feature and a C-dimensional output feature, the time complexity is $\mathcal{O}(|E|FC)$ where E is the set of edges in the graph. The time complexity of a single graph attention network (GAT) [18] is $\mathcal{O}(|V|FC + |E|C)$ where V denotes the set of nodes in a graph. Likewise, for a layer with F-dimensional input and C-dimensional output of the proposed MAGCN, the complexity is $\mathcal{O}(n|E|FC + KC)$, where n is the number of views, $\mathcal{O}(KC)$ is the cost of computing multi-view attention and K denotes the number of neurons in the multilayer perceptron (MLP) in the multi-view attention block. Compared with GCN, while the introduction of multiple views multiplies the storage and parameter requirements by a factor of n, the individual view computations are fully independent and can be parallelized. Overall, the computational complexity is on par with the baseline methods GCN and GAT.

5. Theoretical analysis

Here we characterize the expressive power of these two variants, aiming to investigate why our proposed method can outperform the existing GCN model. For a formal function analysis, we assume that there is a possibly infinite graph $\bar{G}$ with the node set $\bar{V}$ and a probability space with measure $\tilde{P}$, such that the given graph G is viewed as a subgraph of $\bar{G}$ and its nodes are iid samples of $\bar{V}$ based on $\tilde{P}$. Let $f : V \to R$ be a bounded and continuous function defined on a node set $V \subset \bar{V}$, of which each node has m features. Let g be a bounded, differentiable function on R, satisfying $|g(x)| \le 1$ and $\int_K g'(t)dt < \infty$ (K is a compact set). Denote $\hat{M}_{\mu,\nu} := \hat{M}(\mu, \nu)$ as the kernel corresponding to the (μ, ν) element of a given matrix $\hat{M}$. x_μ, x_ν stand for the feature vector of node μ and node ν, respectively. Based on the integral representation framework [50], the unknown function (on $\bar{G}$) can be formulated as

$$f(x_\nu) = \int_{R^m} \alpha(w) g\left(w^{\mathrm{T}} \int_{V} \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu)\right) dw.$$

It can be approximated as

$$f(x_\nu) = \lim_{\theta \longrightarrow +\infty} \int_{-\theta}^{\theta} \cdots \int_{-\theta}^{\theta} \alpha(w_1, w_2, \ldots, w_m) \cdot g\left(w^{\mathrm{T}} \int_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu) \right) dw_1 dw_2 \ldots dw_m.$$

Similar to the previous discussion, the analysis of MLP's universal approximation capability can start with the integral representation of f (see Murata et al. [50]), i.e.,

$$f(x_\nu) = T \int_{R^m} sign[\alpha(w)] g\left(w^{\mathrm{T}} \int_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu) \right) P^*(w) dw,$$

where

$$T := \int_{R^m} |\alpha(w)| dw, \;\; P^*(w) := \frac{1}{T} |\alpha(w)|.$$

Likewise, this integral can be approximated with the Monte-Carlo method to get an approximation defined as

$$\hat{f}_L^*(x_\nu) := \sum_{j=1}^{L} \beta_j g\left(w_j^{\mathrm{T}} \int_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu) \right), \tag{11}$$

where $\beta_j = \frac{T}{L} sign[\alpha(w_j)]$ and the $\{w_1, w_2, \ldots, w_L\}$ are independently chosen subject to the probability distribution P^*, which is unknown in advance. As such, all the parameters $\{\beta_1, \beta_2, \ldots, \beta_L\}$ and $\{w_1, w_2, \ldots, w_L\}$ need to be optimized algorithmically (with the given training samples), as the philosophy of BP training.

Denote the distance between f and $\hat{f}_L^*$ by

$$d_V(f, \hat{f}_L^*) := \sqrt{\frac{1}{|V|}} E\left[\int_V (f(x_\nu) - \hat{f}_L^*(x_\nu))^2 d\tilde{P}(\nu) \right],$$

where $E[\cdot]$ denotes the expectation value with respect to the probability distribution $\tilde{P}$, and $|V|$ denotes the volume of the node set V.

Theorem 1. *There exists a certain appropriate augmented adjacency matrix $\hat{M}$, weights $\{\beta_1, \beta_2, \ldots, \beta_L\}$ and $\{w_1, w_2, \ldots, w_L\}$, for a GCN model with sufficiently large L and trained by a gradient descent-based learning algorithm, its expressive power is bounded in the probability sense that*

$$d_V(f, \hat{f}_L^*) \le \frac{C}{\sqrt{L}}, \;\; \textit{where} \;\; C = \int_{R^m} |\alpha(w)| dw < +\infty.$$

Proof. First, it is easy to verify that $E(\hat{f}_L^*(x_\nu)) = f(x_\nu)$. By Eq. (11) one can obtain that

$$E[\hat{f}_L^*(x_\nu)] = E\left[\frac{T}{L} \sum_{j=1}^{L} sign[\alpha(w_j)] g\left(w_j^{\mathrm{T}} \int_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu) \right) \right].$$

Since the w_j $(j = 1, 2, \ldots, L)$ is drawn independently from the probability distribution $P^*(w) := \frac{1}{T}|\alpha(w)|$, this gives

$$\begin{aligned} E[\hat{f}_L^*(x_\nu)] &= E\left[T sign[\alpha(w)] g\left(w^{\mathrm{T}} \int_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu) \right) \right] \\ &= T \int_{R^m} sign[\alpha(w)] g\left(w^{\mathrm{T}} \int_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu) \right) P^*(w) dw \\ &= f(x_\nu). \end{aligned}$$

In the same way, the variance is obtained as

$$\begin{aligned}
&Var[\hat{f}_L^*(x_\nu)]\\
&= Var\left[\frac{T}{L}\sum_{j=1}^{L} sign[\alpha(w_j)]g\left(w_j^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)\right]\\
&= \frac{1}{L}Var\left[Tsign[\alpha(w)]g\left(w^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)\right]\\
&= \frac{1}{L}E\left[\left(Tsign[\alpha(w)]g\left(w^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)\right)^2\right]\\
&\quad -\frac{1}{L}E\left[Tsign[\alpha(w)]g\left(w^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)\right]^2\\
&= \frac{1}{L}E\left[\left(Tsign[\alpha(w)]g\left(w^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)\right)^2\right] - \frac{1}{L}(f(x_\nu))^2\\
&= \frac{1}{L}\int\left(Tsign[\alpha(w)]g\left(w^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)\right)^2 P^*(w)dw - \frac{1}{L}(f_\theta(x_\nu))^2\\
&= \frac{T}{L}\int|\alpha(w)|g^2\left(w^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)dw - \frac{1}{L}(f_\theta(x_\nu))^2.
\end{aligned}\tag{12}$$

So far we can have

$$\begin{aligned}
d_V^2(f,\hat{f}_L) &= \frac{1}{|V|}E\left[\int\limits_V (f(x_\nu)-\hat{f}_L^*(x_\nu))^2 d\tilde{P}(\nu)\right]\\
&= \frac{1}{|V|}\int\limits_V E\left[(f(x_\nu)-\hat{f}_L^*(x_\nu))^2\right]d\tilde{P}(\nu)\\
&= \frac{1}{|V|}\int\limits_V E\left[(E[\hat{f}_L^*]-\hat{f}_L^*)^2\right]d\tilde{P}(\nu)\\
&= \frac{1}{|V|}\int\limits_V Var[\hat{f}_L^*]d\tilde{P}(\nu).
\end{aligned}$$

We can now substitute the expression for the variance, i.e. Eq. (12), to obtain

$$d_V^2(f,\hat{f}_L) = \frac{1}{L|V|}\int\limits_V\left(T\int|\alpha(w)|g^2\left(w^{\mathrm{T}}\int\limits_V \hat{M}_{\mu,\nu}x_\mu d\tilde{P}(\mu)\right)dw - (f(x_\nu))^2\right)d\tilde{P}(\nu).$$

Since $|g(x)| \le 1$, the above can be bounded by

$$d_V^2(f,\hat{f}_L) \le \frac{1}{L|V|}\int\limits_V\left(T\int|\alpha(w)|dw\right)d\tilde{P}(\nu) = \frac{1}{L}\left(\int|\alpha(w)|dw\right)^2.$$

This gives

$$d_K(f_\theta,\hat{f}_L) \le \frac{C}{\sqrt{L}},$$

where

$$C := \int\limits_{R^m} |\alpha(w)| dw < +\infty.$$

Sridharan et al. [51] proposed an information theoretic framework for multi-view learning, then Xu et al. [48] analyzed the view insufficiency and provided a strict proof based on this framework. Based on the conclusion and theorem presented by Xu et al. [48], the effectiveness of the proposed MAGCN is easily certified by the following theorem.

Theorem 2. *For a graph signal G with an ideal latent complete view set $\mathcal{A}_M$ and given the bounded loss function $\mathcal{L}(\cdot)$, the expected losses of the proposed MAGCN with the view set $\mathcal{A}_M$ and its subset $\mathcal{A}_m$ are expressed as $\mathcal{L}(X_G; \mathcal{A}_M)$ and $\mathcal{L}(X_G; \mathcal{A}_m)$, respectively. Then, the difference between the $\mathcal{L}(X_G; \mathcal{A}_M)$ and $\mathcal{L}(X_G; \mathcal{A}_m)$ is bounded by*

$$|\mathcal{L}(X_G; \mathcal{A}_M) - \mathcal{L}(X_G; \mathcal{A}_m)| \leq \sqrt{I(X_G; (\mathcal{A}_M \setminus \mathcal{A}_m)|\mathcal{A}_m)},$$

and the difference decreases with the increase of m.

Theorem 1 characterizes the universal approximation property of GCN-based learning models. In practical implementations, however, the existing GCN model uses the augmented adjacency matrix $\hat{A}$ as an approximation for the 'ideal' augmented adjacency matrix $\hat{M}$. By evaluating the integral $\int_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu)$ in the Monte Carlo manner via uniform sampling in the node space, the widely-used GCN model can be expressed concisely as

$$\hat{f}_L^{GCN}(x_\nu) := \sum_{j=1}^{L} \beta_j g\left(w_j^{\mathrm{T}} \frac{1}{N} \sum_{i=1}^{N} \hat{A}_{\mu_i,\nu} x_{\mu_i}\right).$$

In the same manner, our model can be formulated as

$$\hat{f}_L^{MAGCN}(x_\nu) := \sum_{j=1}^{L} \beta_j g\left(\alpha_1 w_j^{\mathrm{T}} \frac{1}{N} \sum_{i=1}^{N} \hat{A}_{\mu_i,\nu} x_{\mu_i} + \alpha_2 \tilde{w}_j^{\mathrm{T}} \frac{1}{N} \sum_{i=1}^{N} \hat{B}_{\mu_i,\nu} x_{\mu_i}\right),$$

where $\hat{B}$ stands for the augmented adjacency matrix representing a different view on the graph, α_1 and α_2 are the attention coefficients to be learned.

Based on Theorem 2, it is logical to suppose that GCN has view insufficiency measured by I_1 that partially implies the differences between $\hat{A}$ and $\hat{M}$, MAGCN by I_2 that partially implies the differences between $\{\hat{A}, \hat{B}\}$ and $\hat{M}$, with $I_1 \geq I_2$.

Theorem 3. *In the context of real implementations in discrete case, the expressive powers for GCN and MAGCN, respectively, can be bounded by*

$$d_V(f, \hat{f}_L^{GCN}) \leq \frac{C}{\sqrt{L}} + \|\beta\| \|w\| |g'(0)| \mathcal{O}(I_1),$$

$$d_V(f, \hat{f}_L^{MAGCN}) \leq \frac{C}{\sqrt{L}} + \|\beta\| \|w\| |g'(0)| \mathcal{O}(\alpha_1 I_1 + \alpha_2 I_2),$$

indicating that MAGCN can perform more favorably than GCN with certain appropriate attention coefficients, that is, approaching the theoretical approximation error $d_V(f, \hat{f}_L^*)$ with higher probability.

Proof. It roughly admits that

$$\|\frac{1}{N} \sum_{i=1}^{N} \hat{A}_{\mu_i,\nu} x_{\mu_i} - \int\limits_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu)\| \propto \mathcal{O}(I_1),$$

and

$$\|\alpha_1 \frac{1}{N} \sum_{i=1}^{N} \hat{A}_{\mu_i,\nu} x_{\mu_i} + \alpha_2 \frac{1}{N} \sum_{i=1}^{N} \hat{B}_{\mu_i,\nu} x_{\mu_i} - \int\limits_V \hat{M}_{\mu,\nu} x_\mu d\tilde{P}(\mu)\| \propto \mathcal{O}(\alpha_1 I_1 + \alpha_2 I_2).$$

Based on triangular inequality, we have $d_V(f, \hat{f}_L^{GCN}) \leq d_V(f, \hat{f}_L^*) + d_V(\hat{f}_L^*, \hat{f}_L^{GCN})$. After a careful deduction based on Taylor series for the activation $g(\cdot)$, the latter can be bounded by $\|\beta\| \|w\| |g'(0)| \mathcal{O}(I_1)$, completing the proof of $d_V(f, \hat{f}_L^{GCN})$. Similar tricks can be used to prove $d_V(f, \hat{f}_L^{MAGCN})$.

Table 1
Statistics of the datasets.

Dataset	Cora	Citeseer	Pubmed
Nodes	2,708	3,327	19,717
Edges	5,429	4,732	44,338
Features	1,433	3,703	500
Classes	7	6	3
Label rate	0.052	0.036	0.001

We provide a rigorous mathematical analysis on the expressive power of MAGCN (in terms of universal approximation theory), as well as the reason why the proposed framework can effectively avoid suffering from view insufficiency by virtue of the information theory result for multi-view learning [48]. The derived upper bound for MAGCN (Theorem 3) indicates that, by adjusting the attention coefficients, one can potentially enhance the expressive power of the combination of GCN models that are built on various views.

6. Experiments

In this section, we compare MAGCN with several state-of-the-art methods on three benchmark node classification datasets (Cora, Citeseer, Pubmed) to verify the effectiveness and advantages of our proposed method. To demonstrate the robustness of MAGCN, in comparison with GCN and GAT, we conduct attack simulations with different levels of topology perturbations. The experiment results show that MAGCN consistently and significantly outperforms the state-of-the-art models in terms of both performance and stability.

Baselines. To demonstrate the performance of the proposed MAGCN, we compare it against some baseline methods, including some classic ones like DeepWalk [52], Planetoid [53], some well-known GNNs such as Graph-SAGE [20], ChebNet [23], MPNN [11], GCN [10], GAT [18], DGI [54], AdaLNet [30]. We also compare a multi-view learning based method, i.e., Multi-GCN [41].

Datasets. Three standard citation network datasets - Cora, Citeseer, and Pubmed are utilized in our experiments. For these datasets, the nodes and edges represent the articles and citation relation, respectively. The node feature vector is a vector of a bag-of-words representation of the corresponding article and each article belongs to a research field of its class. Table 1 shows the statistics. We note that a public split (see Yang et al. [53]) for training/validation/testing sets is used in the simulations.

Experiment setup. For the semi-supervised classification task, we follow GCN [10] and build a two-layer MAGCN as Fig. 1 shows. The output dimension F of multi-GCN (unfold) is set to 16. The MLP module in the multi-view attention block contains three fully connected layers, and the number of neurons in the first and second layers is set to 6 and 3, respectively. In particular, the number of neurons in the last layer in MLP is equal to the number of views. The optimizer used in our experiments is Adam [55] with a learning rate of 0.01 and weight decay of 0.0005. The weights of all layers are initialized by the Glorot uniform initializer [56] and the dropout operation [57] with a 0.5 rate is utilized for all layers. It is important to point out that we use three views for all three datasets, the first view (topology) is provided by these datasets, the second view is generated by the feature similarity between different nodes, and the third topology is constructed by a classic graph construction method, as described in the following.

In practice, for these citation network datasets, the nodes denote the documents and the fixed topology (adjacency matrix) corresponds to the citation relation between different papers. Each node (paper) in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the co-occurrence dictionary. It is natural to construct another topology from the text similarity of these word vectors. Specifically, cosine similarity is utilized to measure text similarity. Concretely, there will be an edge between two nodes if the text similarity is greater than the similarity threshold. For more multiple topology views, many classic graph construction methods, e.g., kNN graph construction [14], b-Matching graph construction [15], and low-rank graph learning [16], can easily generate more graph structures. Here, we choose the b-Matching method to generate the third views for the Cora, Citeseer, and Pubmed datasets.

Results and discussion. The quantitative results of our comparative experiments are presented in Table 2. We report the mean and standard error of prediction accuracy of our model after 10 runs. It is worth pointing out that the results of the compared methods are obtained from corresponding papers. Table 2 only lists the results of Cora and Citeseer for Multi-GCN [41] provided by the corresponding paper. As we can see from Table 2, the proposed MAGCN outperforms the other methods on the public split sets. In addition, the performance of GAT is very close to MAGCN because the GAT assigns different weights to nodes of the same neighborhood, which learns a more suitable topology indirectly. It should be noted that Multi-GCN also achieves excellent performance, because Multi-GCN also takes into account multiple topologies. However, the Multi-GCN merges multiple topologies (adjacency matrices) to one adjacency matrix based on the subspace analysis, which leads to a loss of multiple topology information to some extent.

Ablation study. As discussed above, the superiority of the proposed MAGCN has been validated by comparison experiments and proved by theoretical analysis. To further verify the validity of multi-view learning with the attention mechanism in our model, we perform the following ablation studies on the three datasets:

Table 2
Summary of the semi-supervised classification accuracy (%) on Cora, Citeseer, and Pubmed datasets.

Method	Datasets		
	Cora	Citeseer	Pubmed
DeepWalk	67.2	43.2	65.3
Planetoid	75.7	64.7	77.2
ChebNet	81.2	69.8	74.4
MPNN	79.1	65.9	76.6
Graph-SAGE	75.3	68.2	77.4
GCN	81.5	70.3	79.0
GAT	83.0	72.5	79.0
DGI	82.3	71.8	76.8
AdaLNet	81.4	69.7	78.1
Multi-GCN	82.5	71.3	-
MAGCN-2Views (Ours)	83.0 ± 0.1	72.5 ± 0.2	80.1 ± 0.3
MAGCN-3Views (Ours)	**84.5 ± 0.2**	**73.5 ± 0.3**	**80.6 ± 0.2**

Table 3
Ablation studies on three datasets.

Method	Datasets		
	Cora	Citeseer	Pubmed
GCN+View 1	81.5	70.3	79.0
GCN+View 2	59.2	69.9	72.1
GCN+View 3	62.7	63.5	73.5
MLP+GCN+View 1, 2, 3	82.2	71.2	79.3
MAGCN+View 1, 2, 3 (Ours)	**84.6**	**73.8**	**80.8**

- GCN+View 1: GCN with view 1 (the given adjacency matrix);
- GCN+View 2: GCN with view 2 (the similarity-based graph);
- GCN+View 3: GCN with view 3 (the b-matching graph);
- MLP+GCN+View 1,2,3: GCN with three views via a standard MLP;
- MAGCN+View 1,2,3: Our MAGCN with three views.

The ablation results for the three datasets are listed in Table 3. As we can see, the accuracy of GCN with a single topology view is much lower than our MAGCN with three topology views, i.e., the performance of the proposed MAGCN with three views is better than GCN with only a single view. This shows the advantage of multi-view learning, which is consistent with our theoretical and empirical studies discussed above.

To further verify the advantage of our multi-view graph network architecture, we apply the normal GCN on each topology and concatenate them in a straightforward way, i.e., a standard MLP for final classification, which can be regarded as a kind of ensemble learning strategy. The results show that the accuracy of GCN with three views via a standard MLP is also lower than our MAGCN. In addition, we can theoretically explain this empirical finding by referring to Theorem 3, i.e., the generalization capability of MAGCN, which in a broad sense also reflects the model's stability, is implicitly (but not completely) affected by the coefficients α_1, α_2 (which can be viewed as attention coefficients) and the view insufficiency measures I_1, I_2 (corresponding to the two graph topologies considered in the model).

We propose a novel graph GAP to implement the attention mechanism for graph data modeling. To show how much the graph GAP improves the performance compared to a normal GAP, we conduct an ablation study comparing a MAGCN model equipped with a graph GAP with the one with a normal GAP. The accuracies of the models with the graph GAP and the normal GAP (on Cora) are 84.6% and 82.8%, respectively. This shows the proposed graph GAP can improve the performance significantly.

Visualization. A recognized visualization tool t-SNE [58] is utilized to illustrate the effectiveness of the representations of different methods. Feature representations are embedded to 2D projections by the t-SNE and the visualization results are shown in Fig. 3. Obviously, the 2D projections show a discernible clustering phenomenon in the 2D embedding space, and the clusters precisely correspond to the seven classes of Cora. In particular, compared with GCN, the distribution of the node representations in the same cluster is more concentrated whereas different clusters are more separated. As shown in Fig. 3, the red cluster and blue cluster in the GCN are commingled, while the two clusters in our MAGCN visualization still have an obvious segregation character. This illustrates that the MAGCN has a better performance for feature representation.

Robustness analysis. To further demonstrate the advantage of our proposed method, we test the performance of MAGCN, GCN [10] and GAT [18] when dealing with some uncertainty issues in the node classification tasks. Here we only use the Cora dataset and consider two types of uncertainty issues: random topology attack (RTA) and low label rates (LLR), which can lead to potential perturbations and affect the classification performance. Specifically for RTA, we randomly delete some

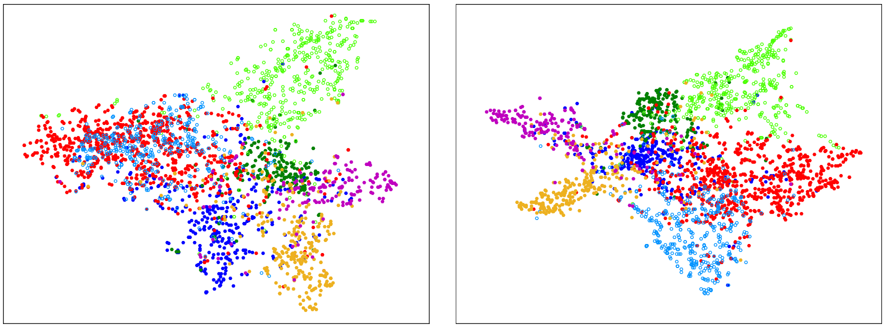

Fig. 3. t-SNE visualization for the computed feature representations of a pre-trained model's first hidden layer on the Cora dataset: GCN (left) and our MAGCN (right). Node colors denote classes. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

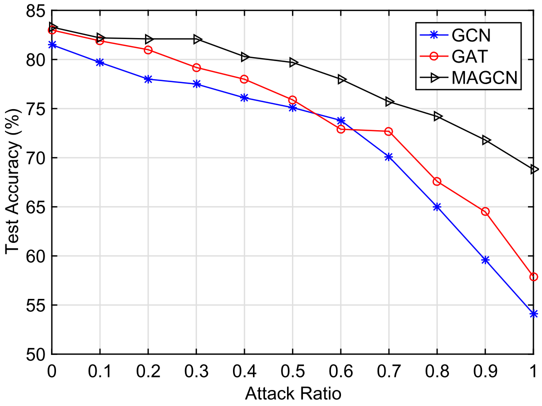

Fig. 4. Test performance comparison of GCN, GAT, and MAGCN on Cora with different levels of random topology attack.

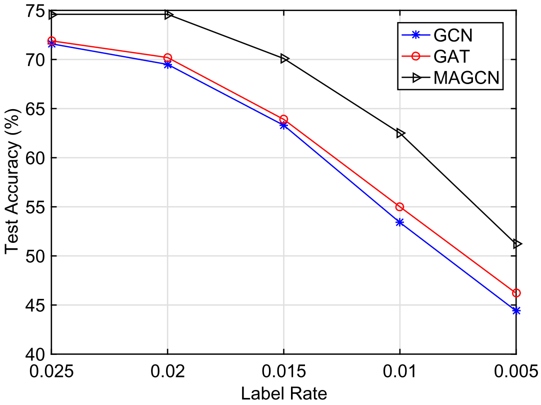

Fig. 5. Test performance comparison for GCN, GAT, and MAGCN on Cora with different low label rates.

edges with a given ratio to get the modified graph and evaluate the classification accuracy on the (clean) test set. We vary the attack ratio (ratio of deleting edges) from 0.1 to 1 and report the experiment results in Fig. 4. It is clear that MAGCN consistently outperforms GCN and GAT. On the whole, the performance of all methods decays rapidly with respect to the random attack ratio. For LLR, we adopt five different label rates $\{0.025, 0.02, 0.015, 0.01, 0.005\}$ to train these models. The test accuracies are plotted in Fig. 5. While the performance of the baselines drops quickly with a decreasing label rate, our MAGCN performs robustly, even when the label rate is extremely low. Overall, the robustness of MAGCN due to its superiority in dealing with real-world applications where the underlying graph has few labels.

Further study. Technically, an end-to-end graph learning framework that can jointly and iteratively adjust the graph structure (as 'component 1') and learn graph embedding (as 'component 2') also has a good potential for problem-solving. Although we have placed particular emphasis on the 'component 2' in the main part of this work, some initial exploration for an

Table 4
Simulation results of further study for MAGCN on Cora, Citeseer, Pubmed.

Method	Datasets		
	Cora	Citeseer	Pubmed
GCN+View 1	81.5	70.3	79.0
GCN+View 2	59.2	69.9	72.1
GCN+View 2*	64.5	70.5	78.6
MAGCN+View 1, 2 (Ours)	83.0	72.5	80.1
MAGCN+View 1, 2* (Ours*)	**83.5**	**72.7**	**80.4**

end-to-end framework considering both the 'component 1' and 'component 2' is expected to motivate further improvement or extension of our proposal. With this purpose and without altering the architecture of MAGCN, we conduct a further study to show that it is possible and flexible to implant 'component 1' in the framework of MAGCN. In particular, we have demonstrated that the modified MAGCN with 'component 1' produces better results than the vanilla MAGCN (that only has 'component 2'). For simplicity, we use a refined cosine function to evaluate the similarity, that is,

$$s_{ij} = \cos(\boldsymbol{w} \odot \boldsymbol{v_i}, \boldsymbol{w} \odot \boldsymbol{v_j}), \tag{13}$$

where s_{ij} denotes the similarity between the i-th node and j-th node, which ranges between $[-1, 1]$.[2] $\boldsymbol{v_i}$ is the feature vector of i-th node and $\boldsymbol{w}$ is a trainable weight vector which has the same dimension as $\boldsymbol{v_i}$ (which is expected to distinguish the contribution of different dimensions of the feature vectors), $\odot$ denotes the Hadamard product.

Unlike the vanilla MAGCN as detailed in the main section, we use Eq. (13) to build the second view, therefore, both the graph structure and graph representation can be learned jointly and iteratively along with solving the downstream tasks. In particular, we perform the following empirical studies on Cora, Citeseer and Pubmed to verify the effectiveness and good potential of this enhancement:

- GCN+View 1: GCN with view 1, i.e., the given adjacency matrix;
- GCN+View 2: GCN with view 2, i.e., the similarity-based graph;
- GCN+View 2*: GCN with view 2*, i.e., the weighted trainable similarity-based graph based on Eq. (13);
- MAGCN+View 1,2: MAGCN with the view 1 and view 2;
- MAGCN+View 1,2*: MAGCN with the view 1 and view 2*.

Experiment results for this further study are summarized in Table 4. Obviously, the enhanced MAGCN (with view 1 and view 2*) outperforms the vanilla MAGCN (with view 1 and view 2). Also, it is clear that GCN with view 2* leads to better performance than GCN with view 2. These results demonstrate that 'component 1' is of great significance and has a good potential to further improve our MAGCN. That also indicates the potential merits of an end-to-end framework for building a multi-view GCN model. Since our main target lies in 'component 2', we only provide a preliminary empirical study in this part. More investigation and in-depth analysis/discussion for the end-to-end methodology are highly expected in our future work.

7. Conclusion

In this paper, we propose a novel graph convolutional network model called MAGCN, which allows to aggregate node features from different hops of neighbors using the multi-view topology of the graph and attention mechanism. Theoretical analysis on the expressive power and flexibility is provided with rigorous mathematical proofs, showing the good potential of MAGCN over the vanilla GCN model in producing a better node-level learning representation. We test MAGCN on several benchmarks Cora, Citeseer, and Pubmed, and demonstrate that it yields results which are superior to the state-of-the-art on the node classification task. Our work paves the way to exploit different adjacency matrices representing a distinguished graph structure to build graph convolution. Jointly and iteratively learning graph structure and graph embedding is an alternative solution for problem-solving. We have made an initial attempt in our experimental study, in which the empirical results illustrate that a trainable graph construction module can be implanted in our proposed MAGCN framework, producing effectively a more advanced MAGCN model that can jointly and iteratively learn the graph structure and embeddings optimized for the downstream node-classification task. More exploration on how to design an end-to-end framework for building a multi-view GCN model, with specific concerns in reducing the complexity burden in graph construction (either for learnable or non-learnable case) and using more advanced GNNs as backbones for more challenging scenarios, are desirable in our future work.

[2] As a widely-used trick for extracting a symmetric sparse non-negative adjacency matrix, we set some s_{ij} values that are smaller than a non-negative threshold to zero.

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 Natural Science Foundation of China (No. U21A20473, 62172370, 62006147 and 62176244), the National Key Research and Development Program of China (2020AAA0106100).

References

[1] L. Sless, N. Hazon, S. Kraus, M. Wooldridge, Forming k coalitions and facilitating relationships in social networks, Artif. Intell. 259 (2018) 217–245.

[2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[3] P. Baldi, P. Sadowski, Z. Lu, Learning in the machine: random backpropagation and the deep learning channel, Artif. Intell. 260 (2018) 1–35.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[5] T. Lüddecke, A. Agostini, M. Fauth, M. Tamosiunaite, F. Wörgötter, Distributional semantics of objects in visual scenes in comparison to text, Artif. Intell. 274 (2019) 44–65.

[6] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, T. Li, Predicting citywide crowd flows using deep spatio-temporal residual networks, Artif. Intell. 259 (2018) 147–166.

[7] A. Rosenfeld, O. Maksimov, Optimal cruiser-drone traffic enforcement under energy limitation, Artif. Intell. 277 (2019) 103166.

[8] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80.

[9] J.B. Estrach, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and deep locally connected networks on graphs, in: International Conference on Learning Representations, 2014.

[10] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.

[11] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: Proceedings of International Conference on Machine Learning, 2017, pp. 1263–1272.

[12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2020) 1–21, https://doi.org/10.1109/TNNLS.2020.2978386.

[13] Z. Zhang, P. Cui, W. Zhu, Deep learning on graphs: a survey, IEEE Trans. Knowl. Data Eng. (2020) 1–21, https://doi.org/10.1109/TKDE.2020.2981333.

[14] J. Chen, H. ren Fang, Y. Saad, Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection, J. Mach. Learn. Res. 10 (69) (2009) 1989–2012.

[15] T. Jebara, J. Wang, S.-F. Chang, Graph construction and b-matching for semi-supervised learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 441–448.

[16] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, N. Yu, Non-negative low rank and sparse graph for semi-supervised learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2328–2335.

[17] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.

[19] J. Atwood, D. Towsley, Diffusion-convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, 2016, pp. 1993–2001.

[20] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Proceedings of Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.

[21] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, M.M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5115–5124.

[22] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: International Conference on Learning Representations, 2019.

[23] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proceedings of Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.

[24] J. Chen, J. Zhu, L. Song, Stochastic training of graph convolutional networks with variance reduction, in: Proceedings of International Conference on Machine Learning, 2018, pp. 941–949.

[25] J. Chen, T. Ma, C. Xiao, FastGCN: fast learning with graph convolutional networks via importance sampling, in: International Conference on Learning Representations, 2018.

[26] W. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, C.-J. Hsieh, Cluster-gcn: an efficient algorithm for training deep and large graph convolutional networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 257–266.

[27] M. Li, Z. Ma, Y.G. Wang, X. Zhuang, Fast Haar transforms for graph neural networks, Neural Netw. 128 (2020) 188–198.

[28] B. Xu, H. Shen, Q. Cao, Y. Qiu, X. Cheng, Graph wavelet neural network, in: International Conference on Learning Representations, 2019.

[29] Z. Ma, M. Li, Y.G. Wang, PAN: path integral based convolution for deep graph neural networks, in: Workshop on Learning and Reasoning with Graph-Structured Representation, ICML, 2019.

[30] R. Liao, Z. Zhao, R. Urtasun, R.S. Zemel, Lanczosnet: multi-scale deep graph convolutional networks, in: International Conference on Learning Representations, 2019.

[31] F. Wu, T. Zhang, A.H.d. Souza Jr, C. Fifty, T. Yu, K.Q. Weinberger, Simplifying graph convolutional networks, in: Proceedings of International Conference on Machine Learning, 2019, pp. 6861–6871.

[32] S. Abu-El-Haija, A. Kapoor, B. Perozzi, J. Lee N-GCN, Multi-scale graph convolution for semi-supervised node classification, in: MLG KDD Workshop, 2018.

[33] Y. Yang, X. Wang, M. Song, J. Yuan, D. Tao, SPAGAN: shortest path graph attention network, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 4099–4105.

[34] J. Ma, P. Cui, K. Kuang, X. Wang, W. Zhu, Disentangled graph convolutional networks, in: Proceedings of International Conference on Machine Learning, 2019, pp. 4212–4221.

[35] Q. Li, Z. Han, X. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 3538–3545.

[36] Q. Li, X. Wu, H. Liu, X. Zhang, Z. Guan, Label efficient semi-supervised learning via graph filtering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9582–9591.

[37] S. Verma, Z. Zhang, Stability and generalization of graph convolutional neural networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1539–1548.

[38] N. Dehmamy, A.-L. Barabási, R. Yu, Understanding the representation power of graph neural networks in learning graph topology, in: Proceedings of Advances in Neural Information Processing Systems, 2019, pp. 15413–15423.

[39] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, C. Zhang, Attributed graph clustering: a deep attentional embedding approach, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 3670–3676.

[40] Y. Ma, S. Wang, C.C. Aggarwal, D. Yin, J. Tang, Multi-dimensional graph convolutional networks, in: Proceedings of the 2019 SIAM International Conference on Data Mining, 2019, pp. 657–665.

[41] M.R. Khan, J.E. Blumenstock, Multi-GCN: graph convolutional networks for multi-view networks, with applications to global poverty, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 606–613.

[42] 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.

[43] J. Liang, J. Cui, J. Wang, W. Wei, Graph-based semi-supervised learning via improving the quality of the graph dynamically, Mach. Learn. 110 (2021) 1345–1388.

[44] F.R. Chung, F.C. Graham, Spectral Graph Theory, American Mathematical Society, 1997.

[45] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Process. Mag. 30 (3) (2013) 83–98.

[46] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2) (2011) 129–150.

[47] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of International Conference on Machine Learning, 2010, pp. 807–814.

[48] C. Xu, D. Tao, C. Xu, Multi-view intact space learning, IEEE Trans. Pattern Anal. Mach. Intell. 37 (12) (2015) 2531–2544.

[49] M. Lin, Q. Chen, S. Yan, Network in network, in: International Conference on Learning Representations, 2014.

[50] N. Murata, An integral representation of functions using three-layered networks and their approximation bounds, Neural Netw. 9 (6) (1996) 947–956.

[51] K. Sridharan, S.M. Kakade, An information theoretic framework for multi-view learning, in: Proceedings of Annual Conference on Learning Theory, 2008, pp. 403–414.

[52] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.

[53] Z. Yang, W. Cohen, R. Salakhudinov, Revisiting semi-supervised learning with graph embeddings, in: Proceedings of International Conference on Machine Learning, 2016, pp. 40–48.

[54] P. Veličković, W. Fedus, W.L. Hamilton, P. Liò, Y. Bengio, R.D. Hjelm, Deep graph infomax, in: International Conference on Learning Representations, 2019.

[55] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, in: International Conference on Learning Representations, 2015.

[56] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[58] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.