Future Generation Computer Systems 118 (2021) 170–179

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Graph-CAT: Graph Co-Attention Networks via local and global attribute augmentations

Liang Yang [a,c,d], Weixun Li [a,c], Yuanfang Guo [b,*], Junhua Gu [a,c]

[a] School of Artificial Intelligence, Hebei University of Technology, Tianjin, China
[b] SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
[c] Hebei Province Key Laboratory of Big Data Calculation, Hebei University of Technology, Tianjin, China
[d] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 14 June 2020
Received in revised form 8 December 2020
Accepted 30 December 2020
Available online 4 January 2021

Keywords:
Graph neural network
Attention mechanism
Attribute augmentation

ABSTRACT

Graph neural networks have achieved tremendous success in semi-supervised node classification. In this paper, we firstly analyse the propagation strategies in two milestone methods, Graph Convolutional Network (GCN) and Graph Attention Network (GAT), to reveal their underlying philosophies. According to our analysis, the propagations in GAT can be interpreted as learnable and asymmetric local attribute augmentations, while that of GCN can be interpreted as fixed and symmetric local attribute smoothing. Unfortunately, the local attribute augmentations in GAT is not adequate in certain circumstances, because the nodes tend to possess similar attributes in local neighbourhoods. With a toy experiment, we manage to demonstrate the necessity to incorporate global information. Therefore, we propose a novel Graph Co-ATtention Network (Graph-CAT), which performs both the local and global attribute augmentations based on two different yet complementary attention schemes. Extensive experiments in both the transductive and inductive tasks demonstrate the superiority of our Graph-CAT compared to the state-of-the-art methods.

1. Introduction

Networks are ubiquitous in real world and have been widely studied. Unfortunately, their irregular structures, which behave quite differently compared to the regular grid-like structures in image, video, audio, etc., prevent them from directly being processed by modern deep learning technologies, which have achieved significant performance improvements for many tasks on grid structured data, such as image classification, object detection, etc.

Recently, Graph Neural Networks (GNNs) [1–4] shed light on processing network data via deep learning. GNNs are developed by applying deep neural networks to graph Fourier domain, which has been investigated in spectral graph theory [5]. Unfortunately, the high complexity of eigen-decomposition hinders GNNs from being applied in practice, especially to the large scale networks. To alleviate this issue, many approximation approaches, such as a truncated expansion in terms of the Chebyshev polynomials, have been utilized. Graph Convolutional Network (GCN) [6], which is motivated from a first-order approximation of the spectral graph convolutions, bridges the gap between spectral graph convolutions and spatial attribute propagations, and achieves satisfactory results with decent scalability. The success of the simple yet well-behaved GCN attributes to the Laplacian smoothing in local neighbourhoods [7,8]. GCN has been widely used in semi-supervised node classification [6], network embedding [9,10], clustering [11] and link prediction [12], etc. (See Fig. 1.)

Inspired by the attention mechanism, which emphasizes the important inputs by assigning more weights to them, Graph Attention Network (GAT) [13] is proposed with an attention-based architecture, where node attends over its neighbours by specifying weights to the neighbours. The adoption of the self-attention strategy further promotes the performance. Unfortunately, different from the Laplacian smoothing and low-pass filtering interpretations of GCN, the underlying interpretations of GAT's performance gain still remain unrevealed.

In this paper, by carefully comparing propagation rules of GCN and GAT, which are the basis and cores of other recently proposed GNNs [14–19], it can be observed that their underlying philosophies are completely different, though they share similar propagation operations. In GCN, the propagation is fixed (task-independent) and the propagation weights of the two directions on an edge are symmetric. The propagation in GCN is equivalent to attribute smoothing in a local neighbourhood. On the contrary, the propagation in GAT is learnable (task-dependent) and

* Corresponding author.
E-mail addresses: yangliang@vip.qq.com (L. Yang), liweixun@hebut.edu.cn (W. Li), andyguo@buaa.edu.cn (Y. Guo), jhgu@hebut.edu.cn (J. Gu).

https://doi.org/10.1016/j.future.2020.12.026
0167-739X/

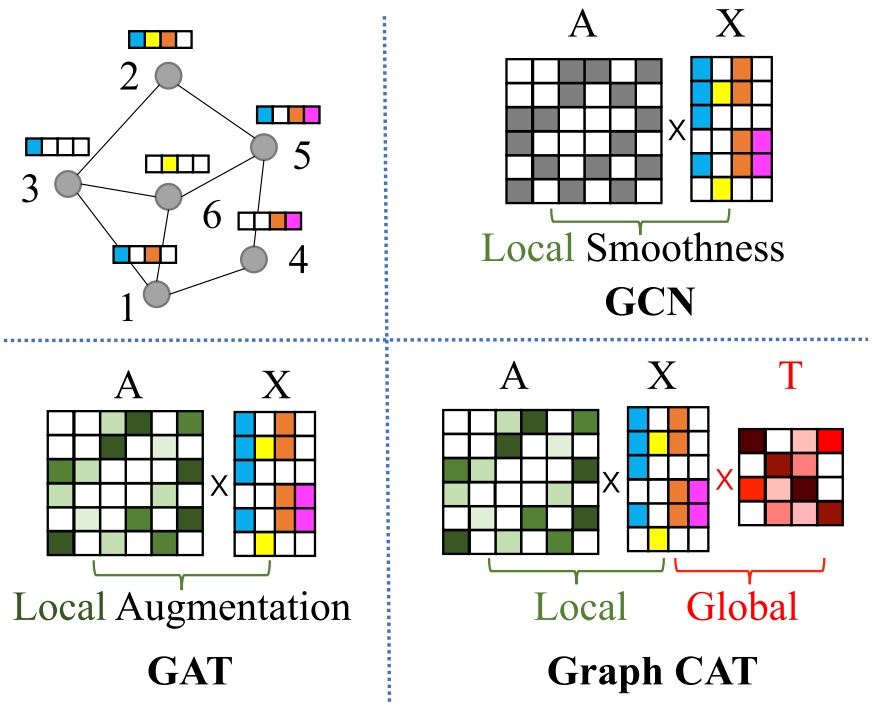


Fig. 1. Comparison of GCN, GAT and our proposed Graph-CAT. A and X are the (refined) adjacency and attribute matrices. T is the learned correlation matrix of attributes which possesses the global information about the correlation between attributes. GCN smoothes the node attribute in a local neighbourhood, while GAT locally augments the node attributes in the neighbourhood. To demonstrate their differences, grey and green adjacency matrices are employed to denote smoothness and augmentation, respectively. Elements with different shades of green are used to illustrate that they are learnable. On the contrary, our Graph-CAT augments the node attributes both locally and globally. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the propagation weights of the two directions on an edge are asymmetric. The propagation in GAT is equivalent to attribute augmentation in a local neighbourhood, instead of the simple smoothing. This attribute augmentation selects the neighbouring nodes with the most important attributes for correct classification and then propagates them.

The attributes propagated from the neighbouring nodes tend to be similar to the current node and they may also lack the necessary attributes for correct classification. A straightforward approach to extend GAT by enlarging the local neighbourhood to contain all the nodes, unfortunately, tends to induce very high complexity and arithmetic underflowing, as the size of the input network increases. Most of the existing approaches, which take the global information into considerations, only explore the high-order topology information (path or random walking sequence) for attribute propagation [20–22], while ignoring the important interactions between the local/global attributes.

To incorporate the necessary global attributes, a novel Graph Co-ATtention Network (Graph-CAT) is proposed to augment the node attributes from both the local and global perspectives, as shown in Fig. 1. The local attribute augmentation is the same as that in GAT. The global (long-distanced) attribute information is incorporated by exploring the attribute correlations. To model the attribute correlations, a fully-connected attribute graph is constructed, where the edges represent the correlations among the attributes and the edge weights are learned via another attention module. This novel attention module constrains each attribute to attend others according to their categorical distributions. Note that the attention weights are determined by a normalized regression function, which accepts the categorical distributions of the two corresponding attributes. These two attention modules interactively augment the node attributes.

Our core contributions are summarized as below:

- We interpret Graph Attention Network (GAT) from the perspective of attribute augmentation, which reveals the insight of its success, compared to GCN.
- We analyse the limitations of the *local* attribute augmentation in GAT and introduce the challenge of augmenting the attributes *globally*.
- We propose a novel Graph Co-Attention Network (Graph-CAT) to locally and globally augment the node attributes with two interactive attention modules.
- Extensive experiments in transductive and inductive tasks demonstrate that our Graph-CAT can effectively alleviate the limitations of local attribute augmentation and yield superior performances compared to the state-of-the-art methods.

2. Related work

Recently, attention mechanism has been widely adopted to graph based semi-supervised node classification. Attention-based GNN (AGNN) [23] assigns the attention weights to each edge, according to the cosine similarity between the two correspondingly connected nodes without any learnable parameters. Graph Attention Network (GAT) [13] extends AGNN by estimating the attention weights with a learnable regression function and using multiple attention heads. Gated Attention Network (GaAN) [15] employs a self-attention mechanism, which calculates a specific weight for each head. Heterogeneous Graph Attention Network (HAN) [17] employs hierarchical attentions, including the node-level and metapath-level attentions. Motif Convolutional Networks (MCN) incorporates the attention mechanism by allowing each node to attend the most relevant motif-induced neighbourhood. Dual Attention Graph Convolutional Networks (DAGCN) [16] employs two attention modules to further improve the performance.

On the other hand, several literatures have explored to incorporate certain global information in GNNs. In GCN [6], the high-order (global) information is taken into consideration by stacking multiple convolutional layers. Unfortunately, its performance degrades for a high number of layers since its convolution are equivalent to Laplacian smoothing operations. To overcome the deteriorations caused by simply increasing the number of convolutional layers in GCN, some advanced strategies are developed. PageRank-GCN [20] establishes a relationship between the attribution propagations in GCN and random walk, and replaces the propagation with the personalized PageRank. DeepGCNs [24] adapt residual/dense connections and dilated convolutions to GCN to alleviate vanishing gradients.

Many recent methods further exploit the impact of different kinds of high-order topology information, such as path, motif and embedding, on attribute aggregation. GeniePath [21] introduces an LSTM-like gating mechanism to aggregate information over multiple graph convolutional layers. SPAGAN [22] extends the neighbourhood-based attention in GAT to path-based attention which can robustly and effectively explore more global topology information. Motif Convolutional Networks [25] employ a weighted multi-hop motif adjacency matrices to capture higher-order neighbourhoods and propose a novel attention mechanism to enable each individual node select the most relevant neighbourhood for propagation. Adaptive Structural Fingerprint (ADSF) [26] improve GAT [13] by designing a weighted, learnable receptive field, which encodes rich and diverse local graph structures, for each node. Geom-GCN [27] proposes a novel permutation-invariant geometric aggregation scheme, which augment neighbourhoods by taking the graph embedding into consideration.

Most of the existing approaches, which take the global information into consideration, only explore the high-order topology (path or random walking sequence) for attribute propagation, while ignoring the important interactions between local/global attributes.

Although our method is neither the pioneer work to construct two attention modules, nor the first work which considers global information, we perform the first attempt to interpret the success and limitations of the attention mechanism in GNN from the perspective of attribute augmentation, and thus propose our Graph-CAT by considering the important interactions between local/global attributes.

3. Notations

A network, whose nodes possess certain attributes, can be modelled as an attributed graph $G = (V, E, X)$. $V = \{v_i | i = 1, \ldots, N\}$ is a set of N vertices, each of which, v_i, is associated with an attribute $x_i \in \mathbb{R}^F$. $x_{i,(p)}$ represents the pth attribute of vertex v_i. Network topology is composed by a set of edges, $E = \{e_i | i = 1, \ldots, M\}$, each of which connects two vertices in V. $X = [x_{ij}] \in \mathbb{R}^{N \times F}$ represents the collection of the attribute features. Each row of X corresponds to the attributes of a node x_i'. For convenience, $x_i \in \mathbb{R}^F$ and $x_{\cdot j} \in \mathbb{R}^N$ are utilized to denote the ith row (all the attributes of vertex v_i) and jth column (the jth attribute of all the vertices) of X in vector form, respectively. Besides, the adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ represents the network topology, where $a_{ij} = 1$ if an edge connects the vertices v_i and v_j, and vice versa. $d_n = \sum_j a_{nj}$ stands for the degree of v_n and $D = \mathrm{diag}(d_1, d_2, \ldots, d_N)$ is the degree matrix of A. The graph Laplacian and its normalized form are defined as $L = D - A$ and $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, respectively.

The labels of a set of labelled nodes $V_l \subset V$ is represented as $Y = [y_{ik}] \in \mathbb{R}^{N \times K}$, where K denotes the number of classes. Note that $y_{ik} = 1$ if the vertex v_i belongs to V_l and the kth cluster. Otherwise, $y_{ik} = 0$. For simplicity, the first l nodes $\{v_i\}_{i=1}^{l}$ are assumed to be labelled. Semi-supervised node classification algorithm classifies other unlabelled nodes in $V - V_l$.

4. GCNN vs attribute augmentation

In this section, two milestone Graph Convolutional Neural Networks (GCNN), GCN and GAT, are reviewed with respect to the spectral graph theory. Then, GAT is interpreted from a novel perspective, i.e., attribute augmentation, which reveals the insight of its success.

4.1. Graph Convolutional Network

Graph Convolutional Network (GCN) [6] is developed based on the first-order approximation of spectral graph convolution. The spectral convolution on graphs is equivalent to the multiplication of a signal $x \in \mathbb{R}^N$ (one scalar attribute for each node) and a filter $g_\theta = \mathrm{diag}(\theta)$, which is parameterized by θ, in the Fourier domain as

$$g_\theta \star x = U g_\theta U' x,$$

where U contains the eigenvectors of the normalized graph Laplacian $\hat{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U'$. Note that the diagonal matrix Λ contains its eigenvalues. To reduce the high complexity induced by the eigen-decomposition of $\hat{L}$, the spectral convolution can be well-approximated by exploiting a truncated expansion in terms of the Chebyshev polynomials [28] as

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

with $\tilde{L} = \frac{2}{\lambda_{max}} \hat{L} - I$. Note that λ_{max} denotes the largest eigenvalue of $\tilde{L}$ and $\theta' = [\theta_k'] \in \mathbb{R}^K$ is a vector of the Chebyshev coefficients. The Chebyshev polynomials are recursively defined as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$. By letting $K = 1$, $\theta = \theta_0' = -\theta_1'$ and $\lambda_{max} \approx 2$, Eq. (1) can be approximated as

$$g_\theta \star x \approx \theta (I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x.$$

Via generalizing the definition of scalar signal $x \in \mathbb{R}^N$ to a signal $X \in \mathbb{R}^{N \times F}$ with F input channels (columns), i.e., F scalar attributes for each node, and extending one filter θ to K filters $W \in \mathbb{R}^{K \times F}$, spectral convolutions on graph are approximated by the following propagation rule with a re-normalization trick as

$$h_i' = \sigma \left(W \sum_{j \in N_i \cup \{i\}} \frac{1}{\sqrt{(d_i + 1)(d_j + 1)}} h_j \right), \tag{2}$$

where $\sigma(.)$ denotes a nonlinear activation function, such as sigmoid and ReLU. h_i, which is the feature of node v_i, is initialized as x_i and gradually evolved by attribute propagations. The parameter W is learned by minimizing the cross-entropy between the given and predicted labels of the labelled nodes as

$$\mathcal{L} = -\sum_{v \in V_l} \sum_{k=1}^{K} Y_{lk} \ln h_{l,(k)}. \tag{3}$$

Li et al. interpret GCN from the perspective of Laplacian smoothing [7,8]. According to the interpretations, there are two fundamental characteristics of the propagations in GCN, i.e., fixed and symmetric propagation weights. Firstly, the propagation weights are not impacted by either the task or the given labels. They are completely determined by the degrees of the two corresponding nodes. Note that the degree of node is the local topology characteristic. Secondly, the propagation weights of two directions on an edge are the same, i.e., $\frac{1}{\sqrt{(d_i+1)(d_j+1)}}$. Because of these two characteristics, the attributes after propagations are more smooth than that before propagations, as shown in Fig. 2. Therefore, the propagations in GCN is equivalent to task independent smoothing.

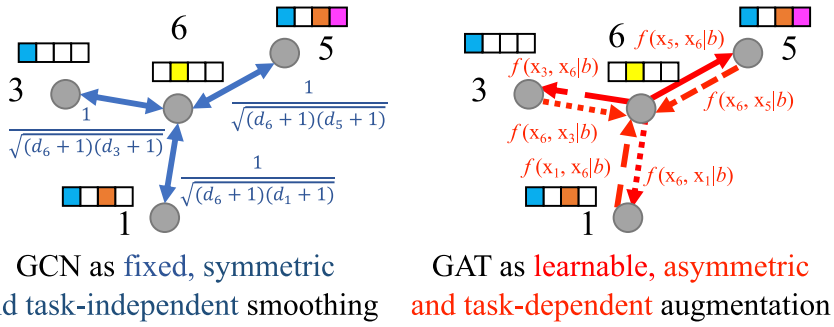


Fig. 2. Comparison of the attribute processing strategies between GCN and GAT.

4.2. Graph Attention Network

Inspired by the recent development of the attention mechanisms, Graph Attention Network (GAT) [13] is introduced to compute the hidden representations of each node by attending over its neighbours via a self-attention strategy. Different from GCN, where the edge weights are determined by the degrees of each two connected nodes, edge weights in GAT can be learned from the attributes of each two connected nodes via a normalized regression model. The regression model is defined as

$$e_{ij} = f(Wh_i, Wh_j) = \text{LeakyReLU}(b^T[Wh_i||Wh_j]), \tag{4}$$

where $||$ denotes the concatenation operation, and $f(.,.)$ stands for a single-layered feedforward neural network parameterized by a learnable weight vector $b \in \mathbb{R}^{2D'}$. The normalization is applied to the local neighbourhoods via a softmax function as

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{e_{ij}}{\sum_{k\in N_i} e_{ik}}. \tag{5}$$

Then, attribute propagation is conducted with respect to the learned edge weight α_{ij} as

$$h_i' = \sigma\left(W\sum_{j\in N_i}\alpha_{ij}h_j\right). \tag{6}$$

The parameters b and W can be obtained by minimizing Eq. (3) as in GCN.

4.3. Attribute augmentation

In GCN, the propagations are performed with *fixed, task-independent* weights which only vary according to the degrees of the nodes. The given labels are only used to train the classifier. Besides of the fixed weights, the propagation weights on an edge are *symmetric*, i.e., the weight of propagating from node v_i to node v_j is the same as that of propagating from node v_j to node v_i. Since the propagations tend to amend the connected nodes to be more similar to each other, they are equivalent to smoothing the node attributes in a local neighbourhood.

Comparing the propagation rules of GAT in Eq. (6) with GCN in Eq. (2), we can observe that their underlying philosophies are completely different in the following two aspects, though they possess similar forms, as shown in Fig. 2.

1. The propagation weights in GAT are *learnable*. The learned edge weights, as shown in Eq. (5), are impacted by both the node attributes and given labels. Thus, the learned weights are task-dependent. Therefore, GAT augments the node attributes by selecting the neighbouring attributes which may benefit the node classification, instead of directly smoothing the node attributes in the local neighbourhoods in GCN.
2. The propagation weights on an edge are *asymmetric* in GAT since $f(.,.)$ is asymmetric. It gives GAT more degrees of freedom to perform propagations, e.g., for two connected nodes v_i and v_j, v_i may accept a relatively large amount of attributes from v_j, while v_j may accept little attributes from v_i.

In general, the learnable and asymmetric propagation weights enable GAT to outperform GCN on both the transductive and inductive semi-supervised node classifications.

Remark. Due to the connection between spectral and spatial GNNs [29], both GCN and GAT can be seen as the low-pass filter, and their standard frequency profiles have almost the same low-pass filter shape corresponding to a function, which consists of a decreasing part on the first quarters of the eigenvalues range followed by an increasing part on the remaining range. The main difference is that variations on the frequency profile of GAT induce more variations on output signal when compared to GCN.

5. Methodology

In this section, a toy experiment is firstly given to show the powers of different attribute augmentation strategies (include smoothing), as well as the importance and difficulties of applying the global attribute augmentation. Then, a novel Graph Co-Attention Network is proposed to jointly perform the local and global attribute augmentations.

5.1. Motivations

To reveal the drawbacks of the local attribute augmentations and the necessity of the global attribute augmentations, a toy experiment is conducted on three attributed networks, Cora, Citeseer and PubMed, to compare the performance of 4 different attribute augmentation strategies.

- **FCN** only utilizes the node attributes without any topology and label information.
- **GCN** smoothes the node attributes in the local neighbourhoods based on the network topology, yet without considering the labels.
- **GAT** augments the node attributes by leveraging the given labels and the attributes from the neighbours.
- **GAT-Global** augments the node attributes by *globally* leveraging all the given labels and nodes from the same ground-truth cluster.

As can be observed from Table 1, the performances are gradually improved as more information is adopted to augment the node attributes. GCN outperforms FCN because the topology information is employed to smooth the node attributes. GAT outperforms GCN, because it exploits the label information and selectively augments the node attributes according to the nodes in the

Table 1
Classification results with different attribute augmentation strategies (include smoothing).

Dataset	FCN	GCN	GAT	GAT-Global
Citeseer	57.1%	70.3%	72.5%	100%
Cora	56.2%	81.3%	83.0%	100%
PubMed	70.7%	79.0%	79.0%	100%

local neighbourhoods. GAT-Global further outperforms GAT by incorporating the global information from long-distanced nodes in the same cluster. The great performances of GAT-Global indicate that it is inadequate to only propagate attributes in the local neighbourhoods, because the attributes propagated from the neighbouring nodes tend to be similar to the current node and they may also lack the necessary attributes for correct classification. Therefore, it is necessary to augment the node attributes with global attribute information.

Naturally, two straightforward extensions of GAT can be constructed. The first approach extends GAT in Eqs. (5) and (6) to allow each node interacting with all the other nodes. This approach possesses two main drawbacks: (1) The computational complexity of interacting with all the nodes is very high; (2) The normalization over all the nodes, especially for large networks, may cause arithmetic underflowing. The second approach considers all the nodes in the same cluster as GAT-Global. Unfortunately, our task is node classification, thus it is impossible to know the membership of each node before processing the input network. Therefore, it is challenging to directly incorporate the global information into attribute propagations.

Since direct integration is difficult, we consider to exploit the global attribute information by taking the attribute co-occurrences into consideration. Intuitively, certain attributes usually appear together, e.g., in a citation network, "learning" and "loss" usually appear simultaneously in the same paper. If one of them is missing in one paper (node), it can be added according to the knowledge of their co-occurrence. Therefore, attribute correlations (co-occurrences), which capture the global attribute information, can be utilized to augment the node attributes.

5.2. Graph Co-ATtention Network

Graph Co-ATtention Network (Graph-CAT) is proposed to exploit both the local and global attributes to augment the node attributes. In Graph-CAT, Co-Attention means that the attention mechanism is concurrently applied to the given graph and the global attribute correlation mining.

Local attribute augmentations are performed by adopting the regular attention mechanism in GAT. Then, we only need to design a global attribute correlation mining module, which is flexible and can be efficiently integrated with our local correlation model. The global attribute correlation mining is achieved by applying the attention mechanism to the attribute graph. This attribute graph is constructed by making each node corresponding to one attribute, as shown in Fig. 3(b). This constructed attribute graph is fully-connected with a learnable adjacency matrix $T = [t_{pq}] \in \mathbb{R}^{P\times P}$, where each learnable edge weight t_{pq} denotes the correlation between the two corresponding attributes. Different from GAT, where each row of the attribute matrix is considered as the signal in each node of the given graph, as shown in Fig. 3(a), each column of the attribute matrix can be regarded as the signal in each node of the constructed attribute graph. Since the dimension of attributes is fixed and usually much smaller than the number of nodes in practice, computing the attentions among any pairs of attributes is feasible while that among any pairs of nodes is not. Thus, our global attribute correlation mining is more efficient and practical compared to GAT-Global.

However, the attention mechanism in GAT (Eq. (4) and Fig. 3(a)), which takes two corresponding rows of the attribute matrix as input, is not suitable for directly processing the attribute graph due to the following two problems. (1) The dimension of the signals in the constructed attribute graph, i.e., the dimension of the column of attribute matrix X, is the same as the number of nodes in the given network. Unfortunately, this dimension is very high and the signals are quite sparse. It requires vector b possessing many parameters, which may induce serious overfittings. (2) The input signals have noises, which may cause errors in estimating the attribute correlations. The co-occurrence of attributes in each nodes may not possess statistical regularity. On the contrary, the categorical distribution of attributes, which reflects the global attribute correlations, is more meaningful.

To alleviate the above problems, the categorical distributions of the attributes are employed to compute the attention weights, as shown in Fig. 3(b). Thus, a matrix $M \in \mathbb{R}^{F\times K}$, each row of which represents the categorical distribution of an attribute, is defined as

$$M = X^T Y, \tag{7}$$

where Y is the label matrix. Then, the attention between attributes is estimated based on the rows of M instead of the columns of X as

$$t_{pq} = \frac{\text{LeakyReLU}(c^T[m_p||m_q])}{\sum_r \text{LeakyReLU}(c^T[m_p||m_r])}, \tag{8}$$

where $c \in \mathbb{R}^{2K}$ with K being the number of clusters, t_{pq} is the element of T, i.e., $T = [t_{pq}] \in \mathbb{R}^{K\times K}$, and m_p stands for the pth row of matrix M. In Eq. (8), the global correlation between two attributes can be computed by a function, which is parameterized by c and takes the categorical distributions of two attributes as inputs. Then, the attendings are carried out at the attribute level as

$$h'_{i,(p)} = \sigma\left(\sum_q t_{pq} h_{i,(q)}\right), \tag{9}$$

where $h_{i,(p)}$ is the pth element (attribute) of node v_i, and its matrix form is

$$h'_i = \sigma(Th_i). \tag{10}$$

By combining Eqs. (6) and (10), which respectively formulates the local and global augmentations, the final output feature of Graph-CAT can be obtained as

$$h'_i = \sigma\left(WT\sum_{j\in N_i}\alpha_{ij}h_j\right), \tag{11}$$

where $T = [t_{pq}]$ and α_{ij} are given in Eqs. (5) and (8) with parameters b and c, respectively. To combine the global attribute augmentation with the multiple attention heads [30] in local attribute augmentation, different global attribute augmentations with different parameter c are assigned to their corresponding attention heads as

$$h'_i = \Big\Vert_{d=1}^{D} \sigma\left(W^d T^d \sum_{j\in N_i}\alpha_{ij}^d h_j\right), \tag{12}$$

where D independent attention estimations are conducted with different local augmentations, α_{ij}^d, global augmentation, T^d, and mapping W^d. Note that $||$ represents the concatenation of the results from different attention heads.

Graph-CAT employs the cross-entropy loss, similar to GCN and GAT. Since the two attention modules are coupled, direct minimization of Eq. (11) can hardly be achieved. Therefore, T is

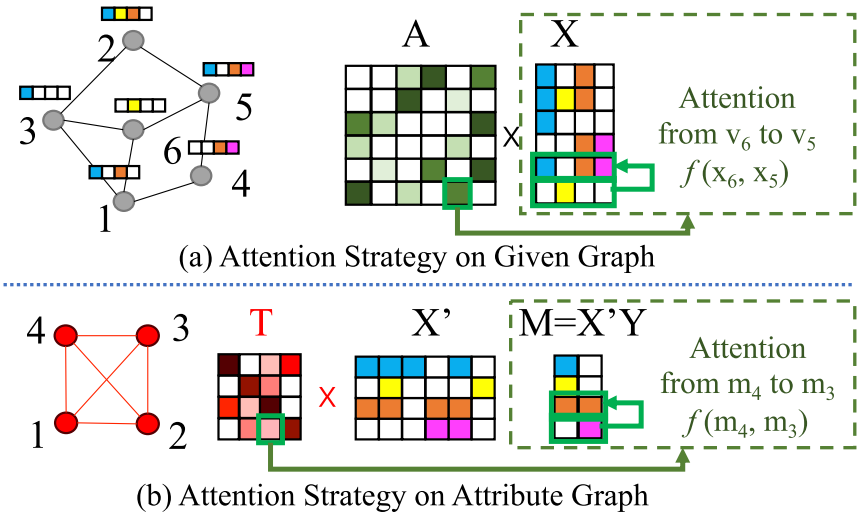


Fig. 3. Comparisons of different attention strategies on given graph and constructed attribute graph.

firstly initialized as the identity matrix, and parameters b and W are optimized. Then, $\{W, b\}$ and c are alternatively updated via the Adam optimizer.

After the formulation of Graph-CAT for the transductive semi-supervised node classification task, we extend it to the inductive task. Similar to GAT, the learned attribute correlations in Graph-CAT can be directly applied to the testing graphs if the attribute distributions in nodes are identical. Therefore, Graph-CAT can be extended to the inductive semi-supervised node classification task, where the testing graphs are not provided in the training stage, by directly employing the learned classifier W, local attention parameter b and global attention parameter c from the training stage.

Complexity. The number of parameters and the computational complexity of one-layer one-head GAT are $(F + 2) * K$ and $O(|V|FK + |E|K)$, respectively, where F and K are the numbers of input and output features, respectively, and $|V|$ and $|E|$ are the numbers of nodes and edges. Meanwhile, those of our Graph-CAT are $(F + 4) * K$ and $O(|V|(F + K)K + |E|K)$, respectively. Since the dimension of the output feature is less than that of the input feature, i.e., $K < F$, the complexity of Graph-CAT is the same as that of GAT. Due to the back-propagation scheme adopted to train the GNNs, it is difficult to estimate the complexity of training GNNs. In fact, the training time of Graph-CAT is 1.2–1.5 times that of CAT, and the additional training time is caused by the additional attention module.

Remark 1. There exists an interesting question "Is T redundant?", because W and T are adjacent in Eq. (11) and they are both learnable. If T is trained with no constraints, the answer to the question is "Yes". However, since the structure of T is highly correlated to the input and the estimated attentions, the answer is "No". The structure (attention mechanism) of T reflects the knowledge that the global attribute correlation is the function of the categorical distributions of the two corresponding attributes. Then, T is meaningful and can constrain the learning of W. Therefore, T is not redundant. This answer will also be verified by the experiments.

Remark 2. Oversmoothing is a serious issue which prevents most GNNs from leveraging deep structure of neural networks. Recently, [31] investigates the loss of expressive power of GNNs via their asymptotic behaviours by generalizing the forward propagation of a GCN as a specific dynamical system. The oversmoothing is the results that the attribute augmentation is too dependent on the local propagation. Existing methods constrain the propagation [20,32–34]. PageRank-GCN [20] integrates personalized PageRank to GCN to constrain the propagation. JKNet [32] employs dense connections for multi-hop message passing. PairNorm [33] proposes a normalization layer to prevents all the node embeddings from becoming too similar. DropEdge [34] randomly removes a certain number of edges from the input graph at each training epoch to act like a data augmenter and to reduce the adverse effect of message passing. In this paper, the dependence of GNNs on local propagation is weakened by enhancing the attribute via both local and global augmentation. Thus, the local propagation only plays a complementary role to global augmentation, and the over-smoothing issue can be alleviated. Recently, Wang et al. [35] observes that GAT may cause the overfitting issue induced by the learning propagation weight. Since our proposed Graph-CAT consists of two adversarial attention modules, the role of propagation weight learning is not as important as in GAT. Thus, the overfitting issue can be alleviated to some extent.

Table 2
Datasets.

Dataset	#Nodes	#Edges	#Classes	#Features
CiteSeer	3,327	4,732	6	3,703
Cora	2,708	5,429	7	1,433
PubMed	19,717	44,338	3	500
NELL	65,755	266,144	210	5,414
PPI	56,944	818,716	121	50

6. Evaluations

To validate the effectiveness of the proposed Graph-CAT, we empirically evaluate its performances in both the transductive and inductive semi-supervised node classification tasks. All the results of the baseline methods are either from their original papers or generated by running the codes from the authors with their default settings.

6.1. Transductive learning

Datasets: To evaluate the performance for the transductive learning task, three commonly utilized citation networks (Cora, CiteSeer and PubMed) [36] and a bi-partite large network

Table 3
Results of transductive learning in terms of accuracies.

Methods	Cora	Citeseer	Pubmed	NELL
MLP	55.1%	46.5%	71.4%	22.9%
ManiReg	59.5%	60.1%	70.7%	21.8%
SemiEmb	59.0%	59.6%	71.7%	26.7%
LP	68.0%	45.3%	63.0%	26.5%
DeepWalk	67.2%	43.2%	65.3%	58.1%
ICA	75.1%	69.1%	73.9%	23.2%
Planetoid	75.7%	64.7%	77.2%	61.9%
Chebyshev	81.2%	69.8%	74.4%	–
MoNet	81.7%	69.9%	78.8%	64.2%
GCN	81.5%	70.3%	79.0%	66.0%
GAT	83.0%	72.5%	79.0%	64.5%
LGCN	83.3%	73.0%	79.5%	–
GWNN	82.8%	71.7%	79.1%	–
DGI	82.3%	71.8%	76.8%	–
SPAGAN	83.6%	73.0%	79.6%	–
Graph-CAT	**84.8%**	**73.6%**	**80.5%**	**68.7%**

(NELL) [37] are adopted, as shown in Table 2. We follow the transductive experimental setups of [38]. In each citation network, nodes are the research papers and the words extracted from the documents will serve as the content. The papers are connected via undirected citations, which are represented as edges, and categorized into various classes according to their research disciplines. In each citation network, 20 nodes per class, 500 nodes and 1000 nodes are employed for training, validation and testing, respectively. The bi-partite network is constructed from a knowledge graph. Two corresponding relationship nodes (e_i, r) and (e_j, r) will be extracted from each entity pair (e_i, r, e_j) along with the entity nodes in the original knowledge graph. The edges are constructed between each entity e_i and all of its relationship nodes (e_i, r).

Baselines: Thirteen semi-supervised node classification algorithms are employed as baselines for the evaluation of the transductive learning task, including multilayer perceptron (MLP), label propagation (LP) [39], semi-supervised embedding (SemiEmb) [40], manifold regularization (ManiReg) [41], graph embedding (DeepWalk) [42], iterative classification algorithm (ICA) [43], attribute-graph based semi-supervised learning framework (Planetoid) [38], graph convolution with higher-order Chebyshev filters (Chebyshev) [28], graph convolutional network (GCN) [6], mixture model networks (MoNet) [44], graph attention networks (GAT) [13], learnable graph convolutional networks [45], graph wavelet neural network (GWNN) [46], deep graph infomax (DGI) [47] and shortest path graph attention network (SPAGAN) [22].

Setups: For the transductive learning tasks, we employ a two-layered Graph-CAT model. The first layer adopts only 3 attention heads, each of which is followed by a global attribute correlation mining module. Note that the exponential linear unit (ELU) [48] is employed as the nonlinear activation function. The second layer, which is designed for performing the classification, consists of only one attention head and it employs the softmax function for nonlinear mapping. During the training process, the L2 regularization with λ ranging from 0.0001 to 0.001 and the dropout with p ranging from 0.3 to 0.7, is applied to the inputs of both layers and normalized attention coefficients. The model is initialized by Glorot and optimized via Adam SGD whose initial learning rate ranges from 0.001 to 0.01.

Results and Analysis: The results of transductive learning in terms of classification accuracies are shown in Table 3. The performance of our proposed Graph-CAT outperforms the state-of-the-art methods including GCN and GAT on all the networks. Our results on two large networks, Pubmed and NELL, are even more remarkable. Besides, we also observe two interesting phenomenons. (1) The results on Cora and Citeseer networks indicate that the performance of local attribute augmentation in GAT is better than that of local smoothing in GCN. However, the superiority of local attribute augmentation in GAT, compared to the attribute smoothing in GCN, is not significant on Pubmed. These unstable performance gains may appear because the information may be limited in the local neighbourhoods. On the contrary, our Graph-CAT, which leverages both the local and global attribute information, consistently outperforms GCN and GAT. (2) We observe that the performance of GAT is lower than that of GCN on NELL, which may be induced by the error accumulations of the local attention weights on the large networks. Since we propose a global attribute augmentation to compensate the local attribute augmentation, our Graph-CAT can successfully reduce the error accumulations. The outstanding performances of our method, not only demonstrate the importance of global attribute information, but also indicate the effectiveness of our method on incorporating the global information.

6.2. Inductive learning

Datasets: The protein–protein interaction (PPI) dataset [49] and Reddit dataset [50] are employed for inductive learning task. We follow the inductive experimental setups of [50]. PPI dataset contains of 24 attributed graphs. Each graph corresponds to a human tissue which consists of 2373 nodes in average. Each node possesses 50 attributes which represents the positional gene sets, motif gene sets and immunological signatures. The nodes are classified into 121 classes according to cellular functions, which are collected from the Molecular Signatures Database [51]. Among the 24 graphs, 20 graphs and 2 graphs are employed to train and validate the algorithms, respectively, while another 2 unseen graphs are employed for testing. The training and validation graphs are both fully labelled, while the testing graphs are not given during the training and validation processes and they are completely unlabelled.

Baselines Ten state-of-the-art algorithms are employed for the evaluation of the inductive learning task, including random classifier (Random), logistic regression based on node features without the network topology smoothing (Logistic Regression), inductive setting of GCN where only the classifier learned from the training graph is adopted on the testing graphs (Inductive GCN) [6], graph attention network (GAT) [13] and its const version (GAT-Const) and three variants of GraphSAGE [50] with different aggregation functions, i.e., GraphSAGE-mean, GraphSAGE-LSTM and GraphSAGE-pool. Note that GAT-Const possesses the same architecture as GAT, but it assigns the same weight to every neighbour. GraphSAGE, which aggregates the representations in local neighbourhoods and concatenates the aggregations with the corresponding node representations, can also be considered as a kind of local attribute augmentations. GraphSAGE-mean utilizes the element-wise means of the local neighbourhoods as the representations. GraphSAGE-LSTM obtains the representation by feeding the representations from local neighbourhoods into an LSTM, which refines the input features. The symmetric and trainable GraphSAGE-pool inputs the representations obtained from local neighbourhoods into a fully-connected neural network and then performs the max-pooling operation to the outcomes.

Setups For the inductive learning task, a two-layered Graph-CAT model is constructed. The first layer consists of 4 attention heads, each of which is followed by a global attribute correlation mining module. The nonlinear mapping functions are ELU and Softmax, for the first and second layers, respectively. Since 20 fully labelled graphs are employed for training, L2 regularization and dropout are both removed. For reducing memory usage and improving the training efficiency, every 2 graphs are trained as a batch. The initialization and optimization parameters are identical to the transductive learning task.

Table 4
Results of inductive learning in terms of micro-F1 scores.

Methods	PPI
Random	0.396
Logistic Regression	0.422
GraphSAGE-mean	0.598
GraphSAGE-LSTM	0.612
GraphSAGE-pool	0.600
Inductive GCN	0.500
LGCN	0.772
GAT-Const	0.934
DGI	0.638
GAT	0.973
Graph-CAT	**0.981**

Results and Analysis The results of inductive learning in terms of classification accuracies are shown in Table 4. All the results of the baseline methods are from [13]. It can be observed that the number of layers employed in Graph-CAT (2 layers) is less than that in GAT (3 layers), because our proposed global augmentation reduces the demand of large local perception fields, which are usually achieved with more layers. The performance gain of GAT compared to GAT-Const, which only uses a constant attention weight, reveals the superiority of augmentation over smoothing. The global attribute correlations, which are learned from the training graphs, can be successfully transferred to the testing graphs in inductive learning task, according to the satisfactory performance gains of Graph-CAT compared to GAT. It can be observed that the improvements of our proposed Graph-CAT in inductive learning task is more significant than that in transductive learning task. It attributes to the following reasons. Firstly, the label information is adequately provided in inductive learning task compared to that in transductive learning task. Secondly, the ratio of the number of attributes to the number of nodes in inductive learning task is smaller than that in transductive learning task. Both of them tend to improve the estimation of categorical distributions of the attributes. Thus, the estimation of attribute correlations will be more robust compared to the transductive learning task.

6.3. Importance of the categorial distribution

To demonstrate the importance of adopting the categorial distribution, the impact of the training sets with various sizes on performance is investigated. If the label matrix is omitted in Eq. (7), the proposed Graph-CAT fails (too low to be included), due to the overfitting caused by the large number of parameter. The classification results versus the different sizes of the training set are shown in Fig. 4. It can be observed that the improvement becomes remarkable as the size of training set increases, because the attention between attributes are obtained based on the categorial distribution of attributes, as shown in Eq. (8). If the size of training set becomes larger, the estimation of the categorial distribution becomes more robust. Then the obtained attention between attributes tend to be more accurate. Therefore, adapting the categorial distribution of attributes serves more effectively and efficiently.

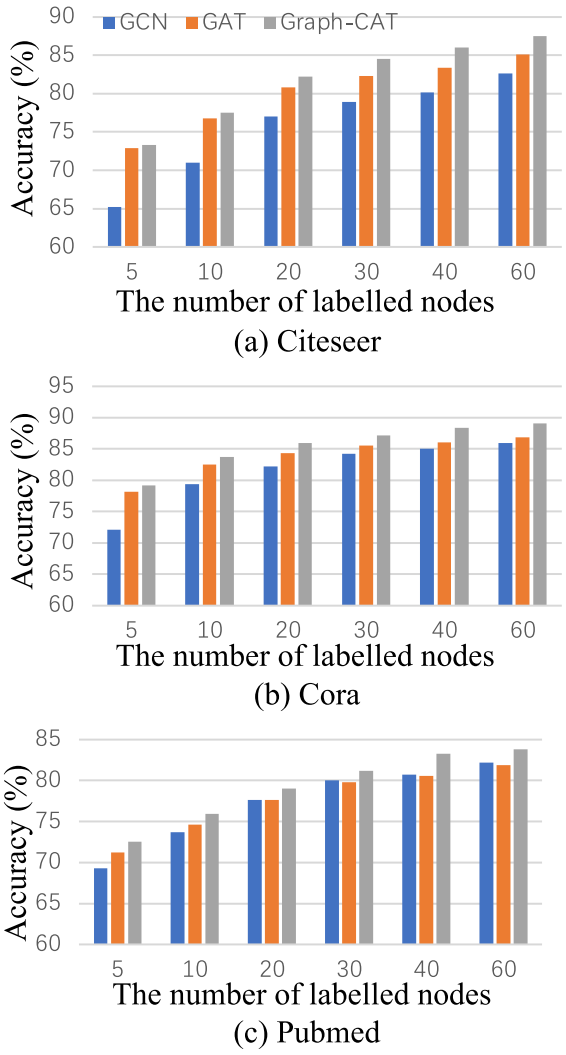


Fig. 4. The impact of training set size on accuracy. As the size increases, the improvement becomes remarkable.

7. Conclusions and future work

In this paper, we firstly analyse the propagation strategies in GCN and GAT. According to our analysis, the learnable and asymmetric local attribute augmentation in GAT is superior compared to the fixed and symmetric local attribute smoothing in GCN. Then, the limitations of local attribute augmentation is introduced. To resolve these limitations, we propose a novel Graph Co-ATtention Network (Graph-CAT) to augment the node attributes both locally and globally. Unfortunately, global attribute information cannot be directly incorporated. Thus, we model the global attribute information with attributes co-occurrence in a constructed attribute graph and perform the local and global augmentations with two interactive attention modules. Experimental results indicate that our Graph-CAT can effectively compensate the limitations of local attribute augmentation and yield superior performances compared to the state-of-the-art methods. In the future, we will employ our proposed Graph-CAT to unsupervised network embedding [52] and knowledge graph [53] to show its potentials.

CRediT authorship contribution statement

Liang Yang: Conceptualization, Methodology, Formal analysis. **Weixun Li:** Software, Validation. **Yuanfang Guo:** Conceptualization, Methodology, Writing - review & editing. **Junhua Gu:** Supervision, Funding acquisition.

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 in part by the National Key R&D Program of China under Grant 2017YFC0820106, in part by the National Natural Science Foundation of China under Grant 61972442, Grant 61802282 and Grant 61802391, in part by the Key Research and Development Project of Hebei Province of China under Grant 20350802D; in part by the Natural Science Foundation of Hebei Province of China under Grant F2020202040, in part by the Hebei Province Innovation Capacity Enhancement Project, China under Grant 199676146H, in part by the Fundamental Research Funds for Central Universities, China, and in part by the Natural Science Foundation of Tianjin of China under Grant 20JCYBJC00650.

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2020) 1–21.

[2] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 2224–2232.

[3] 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, http://dx.doi.org/10.1109/TNN.2008.2005605.

[4] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks? in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, La, USA, May 6-9, 2019, 2019.

[5] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, in: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[6] T.N. Kipf, M. Welling, Semi-Supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[7] 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, AAAI 2018, New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 3538–3545.

[8] Q. Li, X. Wu, H. Liu, X. Zhang, Z. Guan, Label efficient semi-supervised learning via graph filtering, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 2019, pp. 9582–9591, http://dx.doi.org/10.1109/CVPR.2019.00981.

[9] H. Chen, H. Yin, T. Chen, Q.V.H. Nguyen, W. Peng, X. Li, Exploiting centrality information with graph convolutions for network representation learning, in: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019, 2019, pp. 590–601.

[10] Z. Tao, H. Liu, J. Li, Z. Wang, Y. Fu, Adversarial graph embedding for ensemble clustering, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 3562–3568.

[11] J. Cheng, Q. Wang, Z. Tao, D. Xie, Q. Gao, Multi-View attribute graph convolution networks for clustering, in: Proceedings of the Twenty-Nineth International Joint Conference on Artificial Intelligence, IJCAI 2020, Yokohama, Japan, January 1-2, 2020, 2020, pp. 2973–2979.

[12] H. Chen, H. Yin, X. Sun, T. Chen, B. Gabrys, K. Musial, Multi-level graph convolutional networks for cross-platform anchor link prediction, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2020, San Diego, California USA, August 22-27, 2020, 2020.

[13] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.

[14] H. Gao, S. Ji, Graph U-nets, in: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, 2019, pp. 2083–2092.

[15] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, D. Yeung, GaAN: Gated attention networks for learning on large and spatiotemporal graphs, in: proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018, 2018, pp. 339–349.

[16] F. Chen, S. Pan, J. Jiang, H. Huo, G. Long, DAGCN: Dual attention graph convolutional networks, in: International Joint Conference on Neural Networks, IJCNN 2019, Budapest, Hungary, July 14-19, 2019, 2019, pp. 1–8, http://dx.doi.org/10.1109/IJCNN.2019.8851698.

[17] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, 2019, pp. 2022–2032, http://dx.doi.org/10.1145/3308558.3313562.

[18] S. Pan, R. Hu, S. Fung, G. Long, J. Jiang, C. Zhang, Learning graph embedding with adversarial training methods, IEEE Trans. Cybern. 50 (6) (2020) 2475–2487.

[19] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, J. Yang, Hyperspectral image classification with context-aware dynamic graph convolutional network, IEEE Trans. Geosci. Remote Sens. (2020) 1–16.

[20] J. Klicpera, A. Bojchevski, S. Günnemann, Predict then propagate: Graph neural networks meet personalized PageRank, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, La, USA, May 6-9, 2019, 2019.

[21] Z. Liu, C. Chen, L. Li, J. Zhou, X. Li, L. Song, Y. Qi, GeniePath: Graph neural networks with adaptive receptive paths, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, 2019, pp. 4424–4431.

[22] Y. Yang, X. Wang, M. Song, J. Yuan, D. Tao, SPAGAN: Shortest path graph attention network, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, 2019, pp. 4099–4105, http://dx.doi.org/10.24963/ijcai.2019/569.

[23] K.K. Thekumparampil, C. Wang, S. Oh, L. Li, Attention-based graph neural network for semi-supervised learning, CoRR abs/1803.03735 (2018) arXiv:1803.03735.

[24] G. Li, M. Müller, A.K. Thabet, B. Ghanem, DeepGCNs: Can GCNs go as deep as CNNs? in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, 2019, pp. 9266–9275.

[25] J.B. Lee, R.A. Rossi, X. Kong, S. Kim, E. Koh, A. Rao, Graph convolutional networks with motif-based attention, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019, 2019, pp. 499–508.

[26] K. Zhang, Y. Zhu, J. Wang, J. Zhang, Adaptive structural fingerprints for graph attention networks, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.

[27] H. Pei, B. Wei, K.C. Chang, Y. Lei, B. Yang, Geom-GCN: Geometric graph convolutional networks, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.

[28] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 3837–3845.

[29] M. Balcilar, G. Renton, P. Heroux, B. Gauzere, S. Adam, P. Honeine, Bridging the gap between spectral and spatial domains in graph neural networks, 2020, arXiv:2003.11702.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 5998–6008.

[31] K. Oono, T. Suzuki, Graph neural networks exponentially lose expressive power for node classification, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.

[32] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, S. Jegelka, Representation learning on graphs with jumping knowledge networks, in: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, StockholmsmäSsan, Stockholm, Sweden, July 10-15, 2018, 2018, pp. 5449–5458.

[33] L. Zhao, L. Akoglu, PairNorm: Tackling oversmoothing in GNNs, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.

[34] Y. Rong, W. Huang, T. Xu, J. Huang, DropEdge: Towards deep graph convolutional networks on node classification, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.

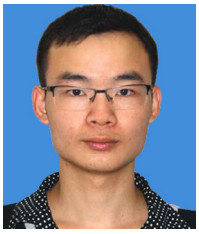
[35] G. Wang, R. Ying, J. Huang, J. Leskovec, Improving graph attention networks with large margin-based constraints, 2019, arXiv preprint arXiv:1910.11945.

[36] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93–106.

[37] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., T.M. Mitchell, Toward an architecture for never-ending language learning, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010, 2010.

[38] Z. Yang, W.W. Cohen, R. Salakhutdinov, Revisiting semi-supervised learning with graph embeddings, in: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, 2016, pp. 40–48.
[39] X. Zhu, Z. Ghahramani, J.D. Lafferty, Semi-Supervised learning using gaussian fields and harmonic functions, in: Proceedings of the 20th International Conference on Machine Learning, ICML 2003, Washington, DC, USA, August 21-24, 2003, 2003, pp. 912–919.
[40] J. Weston, F. Ratle, R. Collobert, Deep learning via semi-supervised embedding, in: Proceedings of the Twenty-Fifth International Conference on Machine Learning, ICML 2008, Helsinki, Finland, June 5-9, 2008, 2008, pp. 1168–1175, http://dx.doi.org/10.1145/1390156.1390303.
[41] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[42] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: Online learning of social representations, in: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, 2014, pp. 701–710, http://dx.doi.org/10.1145/2623330.2623732.
[43] Q. Lu, L. Getoor, Link-based classification, in: Machine Learning, Proceedings of the Twentieth International Conference, ICML 2003, August 21-24, 2003, Washington, DC, USA, 2003, pp. 496–503.
[44] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, M.M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model CNNs, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017, pp. 5425–5434, http://dx.doi.org/10.1109/CVPR.2017.576.
[45] H. Gao, Z. Wang, S. Ji, Large-scale learnable graph convolutional networks, in: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '18, London, UK, August 19-23, 2018, 2018, pp. 1416–1424, http://dx.doi.org/10.1145/3219819.3219947.
[46] B. Xu, H. Shen, Q. Cao, Y. Qiu, X. Cheng, Graph wavelet neural network, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, La, USA, May 6-9, 2019, 2019.
[47] P. Velickovic, W. Fedus, W.L. Hamilton, P. Liò, Y. Bengio, R.D. Hjelm, Deep graph infomax, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, La, USA, May 6-9, 2019, 2019.
[48] D. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), in: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.
[49] M. Zitnik, J. Leskovec, Predicting multicellular function through multi-layer tissue networks, Bioinformatics 33 (14) (2017) i190–i198, http://dx.doi.org/10.1093/bioinformatics/btx252.
[50] W.L. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 1024–1034.
[51] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, E.S. Lander, J.P. Mesirov, Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles, Proc. Natl. Acad. Sci. 102 (43) (2005) 15545–15550, http://dx.doi.org/10.1073/pnas.0506580102.
[52] T. Guo, S. Pan, X. Zhu, C. Zhang, CFOND: Consensus factorization for co-clustering networked data, IEEE Trans. Knowl. Data Eng. 31 (4) (2019) 706–719.
[53] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, CoRR abs/2002.00388 (2020) arXiv:2002.00388.

Liang Yang received the B.E. and M.E. degrees in computational mathematics from Nankai University, Tianjin, China, and the Ph.D. degree in computer science from the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China. His current research interests include community detection, graph neural networks, low-rank modelling, and data mining.



Weixun Li received the B.S. degree in software engineering from the North China Institute of Aerospace Engineering, Hebei, China. He is currently pursuing the M.S. degree in computer science and technology with the Hebei University of Technology, Tianjin. His current research interests include graph neural network and data mining.

Yuanfang Guo (Senior Member, IEEE) received his B.E. degree in computer engineering and his Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2009 and 2015, respectively. Then, he served as an Assistant Professor with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, for three years. He is currently an Assistant Professor with the Laboratory of Intelligent Recognition and Image Processing, School of Computer Science and Engineering, Beihang University, Beijing, China. His research interests include image/video security, compression, processing, and data analysis.

Junhua Gu was born in 1966. He is currently working at the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China. His main research interests include data mining, intelligent information processing, information acquisition and integration, intelligent computing and optimization, function and information display, software engineering and project management.