HONGAT: Graph Attention Networks in the Presence of High-Order Neighbors

Heng-Kai Zhang, Yi-Ge Zhang, Zhi Zhou, Yu-Feng Li*

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
{zhanghk,zhangyg,zhouz,liyf}@lamda.nju.edu.cn

Abstract

Graph Attention Networks (GATs) that compute node representation by its lower-order neighbors, are state-of-the-art architecture for representation learning with graphs. In practice, however, the high-order neighbors that turn out to be useful, remain largely unemployed in GATs. Efforts on this issue remain to be limited. This paper proposes a simple and effective **h**igh-**o**rder **n**eighbor GAT (HONGAT) model to both effectively exploit informative high-order neighbors and address over-smoothing at the decision boundary of nodes. Two tightly coupled novel technologies, namely *common neighbor similarity* and *new masking matrix*, are introduced. Specifically, high-order neighbors are fully explored by generic high-order common-neighbor-based similarity; in order to prevent severe over-smoothing, typical averaging range no longer works well and a new masking mechanism is employed without any extra hyperparameter. Extensive empirical evaluation on real-world datasets clearly shows the necessity of the new algorithm in the ability of exploring high-order neighbors, which promisingly achieves significant gains over previous state-of-the-art graph attention methods.

Introduction

Deep neural networks such as Convolutional Neural Networks (CNNs) have achieved great success in various tasks (LeCun, Bengio, and Hinton 2015; Zhou, Jin, and Li 2024; Ge et al. 2024; Zhu et al. 2024; Wei et al. 2022; Jia et al. 2024; Shi, Wei, and Li 2024). However, architectures in CNNs are typically designed for grid-like structures, which can not process graph-structured data that widely exist in real applications. For example, social networks are naturally graphs, where people are represented by nodes and friendships or interactions between them are represented by edges. Models that are able to exploit the rich information encoded in graph-structured data are highly desirable. Therefore, Graph Neural Networks (GNNs) (Gori, Monfardini, and Scarselli 2005; Scarselli et al. 2008) are introduced to provide powerful frameworks for encoding arbitrarily structured graphs by iteratively aggregating node representations. Nowadays, GNNs have been widely applied in various fields such as knowledge graphs (Hamaguchi et al. 2017), protein prediction (Fout et al. 2017), language processing (Yao, Mao, and Luo 2019), social networks (Wu et al. 2020), etc.

Recently, graph attention network (GAT) shows a promising framework by combining GNNs with attention mechanism in handling graphs with arbitrary structures (Veličković et al. 2018; Zhang et al. 2020). The attention mechanism allows dealing with variable sized input while focusing on the most relevant parts, and has been widely used in sequence modelling (Bahdanau, Cho, and Bengio 2015; Devlin et al. 2019; Vaswani et al. 2017), machine translation (Luong, Pham, and Manning 2015), and visual processing (Xu et al. 2015). The GAT model further introduces attention module into graphs, where the hidden representations of the nodes are computed by repeatedly attending over the features of their neighbors, and the weighting coefficients are calculated inductively based on a self-attention strategy. State-of-the-art performance has been obtained on tasks of node embedding and classification.

The attention in GAT is computed mainly based on the content of the nodes; the structures of the graph, on the other hand, are simply used to mask the attention, e.g., only one-hop neighbors will be attended. However, rich structural information revealed by *high-order neighbors* should provide a more valuable guidance on learning node representations. For example, in social networks or biological networks, a community or pathway is oftentimes composed of nodes that are densely inter-connected with each other but several hops away. Therefore, it can be quite beneficial if a node can attend to high-order neighbors from the same community, even if they show no direct connections. To achieve this, simply checking k-hop neighbors would seem insufficient; on the other hand, simply exploring high-order information with increased model layers would also cause performance degeneration (over-smoothing phenomenon (Li, Han, and Wu 2018; Oono and Suzuki 2020), where the increased GNN layers lead to an overbroad average of neighbor representations for each node (Xu et al. 2018)). A thorough exploration of structural landscapes of the graph becomes necessary.

In order to fully exploit rich, high-order structural details in graph attention networks, we propose a new model called HONGAT. The key idea is to first adaptively augment the high-order neighbor similarity calculation with a general framework, and then learn valuable high-order neighbors through masking in the aggregation stage, so as to increase

*Corresponding author

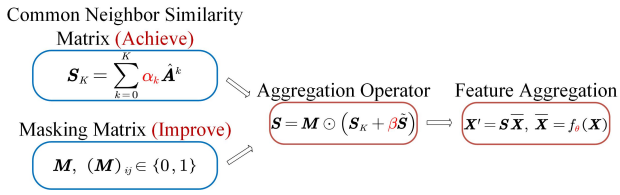


Figure 1: The framework of our proposed HONGAT model. High-order neighbor utilization is achieved by common neighbor similarity and improved by masking mechanism.

the information that is helpful to the generalization performance. To this end, two tightly coupled novel technologies, namely *common neighbor similarity* and *new masking matrix* are introduced. Specifically, high-order neighbors are fully explored by generic high-order common-neighbor-based similarity; in order to prevent potential feature over-smoothing, typical averaging range no longer works well and a new masking mechanism is employed without any extra hyperparameter. To our best knowledge, it is the first work to generalize and encode common neighbor topology that is critical for neighbor exploration into GATs, and we innovatively adjust averaging range to improve neighbor utilization. Extensive empirical evaluation on real-world datasets clearly shows the strength of our new algorithm in its ability to explore high-order neighbors, which leads to significant improvements over previous state-of-the-art graph attention methods.

Exploring High-Order Neighbors in GATs

Limitation of One-Hop Attention

We first briefly review the one-hop attention mechanism and discuss the limitation of it in handling with high-order neighbors, which sheds light on the motivation of our work.

Given a graph $G = (V, E)$ where V denotes the set of nodes and E denotes the set of edges. The input feature matrix to the convolutional layer can be written as $\mathbf{X} \in \mathbb{R}^{N \times d}$ where N indicates the node count and d is the feature dimension. The i-th row vector of $\mathbf{X}$, namely $\mathbf{x}_i = \mathbf{X}_{i:}$, represents the input feature of the i-th node. The attention mechanism allows nodes to aggregate the most relevant information by calculating a weighted average of their neighbors' feature representations. To obtain the averaging weights, we first compute an attention coefficient for every node pair (i, j), which indicates the importance of the node j to the node i:

$$e_{i,j} = LeakyReLU(\alpha^T[\overline{\mathbf{x}}_i^T \,||\, \overline{\mathbf{x}}_j^T]), \tag{1}$$

where $\alpha \in \mathbb{R}^{2d'}$ are learned. $\overline{\mathbf{x}}_i = \mathbf{x}_i\mathbf{W}$ is the hidden state feature of node i where $\mathbf{W} \in \mathbb{R}^{d \times d'}$ are parameters to transform input features into a new space of d' dimension. The operation $||$ indicates vector concatenation. The scores are then normalized by softmax across neighbors $j \in \mathcal{N}(i)$ to obtain the final weighting coefficient $a_{i,j}$:

$$a_{i,j} = softmax_j(e_{i,j}) = \frac{exp(e_{i,j})}{\sum_{j' \in \mathcal{N}(i)} exp(e_{i,j'})}. \tag{2}$$

Then, a weighted aggregation operation is taken (followed by a nonlinearity σ) to update the representation of node i:

$$\mathbf{x}'_i = \sigma\Big(\sum_{j \in \mathcal{N}(i)} a_{i,j}\overline{\mathbf{x}}_j\Big), \tag{3}$$

This attention is calculated mainly based on the content of the nodes and only one-hop neighbors will be attended. It can be quite beneficial if a node can attend to high-order neighbors from the same community, even if they show no direct connections. However, the high-order neighbors that turn out to be useful, remain largely unemployed. To achieve this, simply checking k-hop neighbors would seem insufficient; on the other hand, simply exploring high-order information with increased model layers would also cause performance degeneration, i.e., over-smoothing phenomenon (Li, Han, and Wu 2018; Oono and Suzuki 2020). Therefore, a thorough exploration of high-order neighbors of the graph becomes necessary.

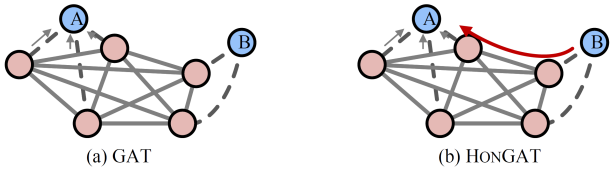
Inspired by this, we propose a HONGAT framework as shown in Figure 1 to exploit informative high-order neighbors by introducing two technologies: (a) neighbors are fully explored by generic high-order common neighbor similarity matrix $\mathbf{S}_K$, which is topology-based and ensures that HONGAT conducts a more comprehensive exploration of neighbor information compared to GAT, which only employs the content-based semantic similarity matrix $\tilde{\mathbf{S}}$; (b) the typical averaging range is further adjusted by a masking matrix $\mathbf{M}$ to prevent feature over-smoothing and learn most valuable high-order neighbors, so as to promote high-order neighbor exploration. Overall, high-order neighbor utilization is *achieved* by common neighbor similarity and *improved* by masking mechanism. Figure 1 and experimental results clearly illustrate the tight coupling and non-removability of these two components. In the following, we demonstrate the detailed construction of HONGAT framework.

Exploiting High-Order Neighbors with Common Neighbor Similarity

To fully exploit rich, high-order structural details in graph attention networks, we introduce *common neighbor similarity* to enable the exploration of high-order neighbors. The main idea is to adaptively adjust the importance of neighbor information of both low- and high-orders. In this way, the high-order neighbors can be explored while the advantages of low-order neighbors remain to be preserved. To achieve this, we introduce common-neighbor-based similarity to measure the importance of different orders of neighbors and allow them to be jointly attended within a single layer.

Specifically, we first define the similarity between nodes based on the neighbor distribution. One classic and popular way is common neighbors index (CN), which assumes that nodes with more common neighbors are more likely to be relevant. More formally, given two nodes i and j with neighbor sets $\mathcal{N}(i)$ and $\mathcal{N}(j)$, their similarity score can be computed by $S^{CN}_{i,j} = |\mathcal{N}(i) \cap \mathcal{N}(j)| = \sum_{n \in \mathcal{N}(i)} \mathbb{I}(n \in \mathcal{N}(j))$, where $\mathbb{I}(n \in \mathcal{N}(j))$ equals to 1 when $n \in \mathcal{N}(j)$ and 0 otherwise.

Obviously, the CN index reveals the importance of a node's second-order neighbors to it. To further explore high-order



(a) GAT (b) HONGAT

Figure 2: Comparison of attention employed by (a) GAT and (b) HONGAT. Conventional GATs only focus on one-hop neighbors each layer (revealed by gray arrows). As a result, nodes A and B, although highly inter-connected, fail to cooperate with each other. The proposed HONGAT, by contrast, exploits information of distant neighbors and structural details within a single step. By this, node B can be reached by node A through their high-order common neighbors (e.g., the red arrow).

information, we generalize the CN index to K-order common neighbors index (KCN), which aims to measure the importance of a node's Kth-order neighbors:

Definition 1 (K-order Common Neighbors)**.** *Given two nodes i and j with neighbor sets $\mathcal{N}(i)$ and $\mathcal{N}(j)$, their K-order common neighbors index ($K \geq 2$) is defined as:*

$$(S_{i,j}^{KCN})_K = \underbrace{\sum_{n_1 \in \mathcal{N}(i)} \frac{1}{\sqrt{d_i d_{n_1}}} \sum_{n_2 \in \mathcal{N}(n_1)} \frac{1}{\sqrt{d_{n_1} d_{n_2}}} \cdots}_{K-2} \sum_{n_{K-1} \in \mathcal{N}(n_{K-2})} \frac{\mathbb{I}(n_{K-1} \in \mathcal{N}(j))}{\sqrt{d_{n_{K-2}} d_j} d_{n_{K-1}}}, \tag{4}$$

where $\mathcal{N}(k)$ and d_k indicate the neighbor set and degree of node k. n_0 is defined as the node i itself.

The K-order common neighbors is a generic high-order similarity metric with the same form as CN (remind that $S_{i,j}^{CN} = \sum_{n \in \mathcal{N}(i)} \mathbb{I}(n \in \mathcal{N}(j))$. Intuitively, rather than directly considering the common neighbors between node i and node j, we take the common high-order neighbors between node i and node j into account. Therefore, the node i is allowed to explore a "chain" of high-order neighbors when reaching node j, instead of being second-order limited. Additionally, node degree centrality (Salton and McGill 1984; Zhou, Lü, and Zhang 2009) is considered in denominators for numerical stability.

Let $(S_{i,j}^{KCN})_0 = \delta(i-j)$ where $\delta(\cdot)$ denotes an impulse function, which maps 0 to 1 and other values to 0; and let $(S_{i,j}^{KCN})_1 = \frac{\mathbf{A}_{ij}}{\sqrt{d_i}\sqrt{d_j}}$ where $\mathbf{A}$ indicates the adjacency matrix of nodes without self-loop. We then jointly attend to all the K-order neighbors within a single layer by simply ranging KCN indices from order 0 to K. Specifically, the representation of node i is updated as:

$$\mathbf{x}_i' = \sum_{j \in \mathcal{N}_K(i)} [\sum_{k=0}^{K} \alpha_k (S_{i,j}^{KCN})_k + \beta a_{i,j}] \overline{\mathbf{x}}_j, \tag{5}$$

where $\mathcal{N}_K(i)$ indicates all the K-order neighbors of node i and $\overline{\mathbf{x}}_j = \mathbf{x}_j \mathbf{W}$ is the hidden state feature of node j. $\{\alpha_k\}_{k=0}^K$, β and $\mathbf{W}$ are learned. Remind in GAT, $a_{i,j}$ indicates the feature-based importance score defined in Equation (2), which can be viewed as the semantic similarity. Different from GAT that only considers the semantic similarity, HONGAT introduces topological similarity revealed by high-order common neighbor indices $\{(S_{i,j}^{KCN})_k\}_{k=0}^K$. In this way, node importance is measured both topologically and semantically, as well as being adaptively learned by trainable parameters $\{\alpha_k\}_{k=0}^K$ and β to capture different high-order neighbors.

Figure 2 illustrates the benefit. Conventional GATs only focus on one-hop neighbors each layer. Thus, nodes A and B, although highly inter-connected, fail to cooperate with each other. The proposed HONGAT, by contrast, exploits information of distant neighbors and structural details within a single step. Thus, node B can be reached by node A through their high-order common neighbors.

Matrix Implementation. The K-order KCN index defined in Equation (4) can be expressed in the following matrix form (proof is covered in Appendix B):

$$\mathbf{S}_K^{KCN} = \hat{\mathbf{A}}^K = (\mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}})^K, \tag{6}$$

where matrix $\mathbf{S}_K^{KCN}$ is composed of K-order KCN indices, namely $(\mathbf{S}_K^{KCN})_{ij} = (S_{i,j}^{KCN})_K$. $\mathbf{A}$ indicates the graph adjacency matrix without self-loop and $\hat{\mathbf{A}}$ represents its symmetrically normalized form. $\mathbf{D}$ represents a diagonal degree matrix, namely $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. By this, the HONGAT layer can be formalized as:

$$\mathbf{X}' = (\Sigma_{k=0}^K \alpha_k \hat{\mathbf{A}}^k + \beta \tilde{\mathbf{S}}) \overline{\mathbf{X}}, \tag{7}$$

where $\overline{\mathbf{X}} = \mathbf{X}\mathbf{W}$ is the hidden feature matrix and $\mathbf{X}'$ is the aggregated feature matrix. $\mathbf{S}_K = \Sigma_{k=0}^K \alpha_k \hat{\mathbf{A}}^k$ denotes the proposed common neighbor similarity matrix which introduces topological information to enable high-order neighbor exploration. $\tilde{\mathbf{S}}$ denotes the feature similarity matrix used in GAT, where only values of K-order neighbors are non-zero and only node content is considered. By Equation (7), HONGAT layer is actually a polynomial-like filter, which therefore enjoys well-studied properties of polynomial filters (Shuman et al. 2013).

Optimizing High-Order Neighbor Exploitation with New Masking Matrix

Till now, we have achieved high-order neighbor utilization with generic common neighbor similarity. The exploration of high-order neighbors, on the other hand, leads to a broader range of averaged neighbors for each node (Xu et al. 2018), yielding the previously mentioned performance issues (over-smoothing phenomenon (Li, Han, and Wu 2018; Oono and Suzuki 2020), where the overbroad average of neighbors' feature representations for each node leads to harmful feature loss). To this end, we propose to improve the exploitation of high-order neighbors by adjusting the typical averaging range, so as to learn the most valuable high-order neighbors and increase the information that is helpful to the model generalization performance.

To achieve this, we introduce a new masking mechanism. The key idea is that, by eliminating neighbor weights in the

aggregation stage, part of the neighbors are "masked" and the averaging range is adjusted. We start by introducing a novel learnable masking matrix as following:

$$\mathbf{M} = \Sigma_{k=0}^{K} \gamma_k \mathbf{M}_k, \tag{8}$$

In this equation, $\mathbf{M}_k \in \{0,1\}^{N\times N}$ where $(\mathbf{M}_k)_{ij} = 1$ if $j \in \mathcal{N}_k(i)$ and $(\mathbf{M}_k)_{ij} = 0$ otherwise. This is achieved by calculating the k-order polynomial of adjacency matrix and binarizing it:

$$(\mathbf{M}_k)_{ij} = \begin{cases} 1, & (\tilde{\mathbf{M}}_k)_{ij} \neq 0; \\ 0, & (\tilde{\mathbf{M}}_k)_{ij} = 0. \end{cases} \tag{9}$$

where $\tilde{\mathbf{M}}_k = \Sigma_{l=0}^{k} \mathbf{A}^l$. Specifically, in $\mathbf{M}_k$, elements corresponding to neighbors other than the ones of order 0 to k are set to 0. Therefore, by applying $\mathbf{M}_k$ to the aggregation operator with an element-wise Hadamard product, the weights corresponding to neighbors of order $(k+1)$ to K are eliminated. In this way, the aggregation is restricted to include only k-order neighbors and the averaging range is implicitly adjusted.

Based on this, trainable parameters $\{\gamma_k\}_{k=0}^{K}$ aim to adjust the ratio of different discrete averaging ranges defined by $\{\mathbf{M}_k\}_{k=0}^{K}$. In this way, the averaging range can be continuously tuned. For example, when γ_k with a larger k is increased, the averaging range is encouraged to include more nodes and otherwise, more concentrated nodes. The model learns the most suitable averaging range for high-order neighbor exploration according to the property of graphs and therefore increases the information that is helpful to the generalization performance by this, which is quite desirable. The following experiments clearly shows the improvement of high-order neighbor utilization and prediction performance with averaging range adjusted by new masking.

Finally, we deploy this masking matrix in HONGAT layer defined in Equation (7):

$$\mathbf{X}' = [\mathbf{M} \odot (\mathbf{S}_K + \beta\tilde{\mathbf{S}})]\overline{\mathbf{X}}, \tag{10}$$

where $\odot$ indicates the element-wise Hadamard product, $\overline{\mathbf{X}} = \mathbf{X}\mathbf{W}$ denotes hidden feature matrix and $\mathbf{S}_K = \Sigma_{k=0}^{K} \alpha_k \hat{\mathbf{A}}^k$. Equation (10) defines the full layer of HONGAT . In practice, we employ a single HONGAT layer and obtain hidden features with $\overline{\mathbf{X}} = f_\theta(\mathbf{X})$, where f_θ denotes a neural network with parameter set $\{\theta\}$. We train $\{\theta\}$ and other parameters together in an end-to-end fashion.

Overall Algorithm Description of HONGAT

The framework figure and pseudocode of the proposed HONGAT method are shown in Figure 1 and Algorithm 1, where $\mathbf{S}_K$ represents the common neighbor similarity matrix and $\mathbf{M}$ the masking matrix, $\tilde{\mathbf{S}}$ is the feature similarity matrix following the work of GAT. Overall, in HONGAT framework, common neighbor indices are generalized to explore high-order neighbors and the masking matrix aims to further improve information utilization. In this way, high-order neighbors are well exploited.

Algorithm 1: Training Phase of HONGAT

Input: Normalized adjacency matrix $\hat{\mathbf{A}}$, feature matrix $\mathbf{X}$, maximum order of common neighbors K
Output: Neural network $f_\theta(\cdot)$, aggregation operator $\mathbf{S}$

while not convergence **do**
  Generate the common neighbor similarity matrix via $\mathbf{S}_K = \Sigma_{k=0}^{K} \alpha_k \hat{\mathbf{A}}^k$
  Compute the feature similarity matrix $\tilde{\mathbf{S}}$ via Eq. (2).
  Generate the masking matrix $\mathbf{M}$ via Eq. (8).
  Obtain aggregation operator: $\mathbf{S} = \mathbf{M} \odot (\mathbf{S}_K + \beta\tilde{\mathbf{S}})$
  Obtain the hidden features: $\overline{\mathbf{X}} = f_\theta(\mathbf{X})$
  Compute $Loss$ via aggregated features $\mathbf{X}' = \mathbf{S}\overline{\mathbf{X}}$
  Optimize parameters according to $Loss$.
end while
return $f_\theta(\cdot)$ and $\mathbf{S}$ =0

Calculation Details. To generate polynomials of adjacency matrix in common neighbor similarity and masking, K iterations: $\mathbf{S}^{(k)} = \mathbf{S}^{(k-1)}\hat{\mathbf{A}} + \alpha_{K-k}\mathbf{I}$ are employed where $\mathbf{S}^{(0)} = \alpha_K \mathbf{I}$. The computational cost is $\mathcal{O}(K|E|n + Kn)$ due to sparsity of $\hat{\mathbf{A}}$ and $\mathbf{I}$. Then, computational cost of binarization and Hadamard product is $\mathcal{O}(n^2)$. Although all operations can be further accelerated by distributed computing infrastructures such as Apache Spark, HONGAT is available to be recommended for large graphs where operators are pre-calculated. A detailed analysis of this is covered in Appendix D.

Experiments

In this section, we conduct experiments on diverse real-world datasets to validate the performance of the proposed HONGAT. We try to give answers to the following three questions.

RQ 1. Does the proposed method, with high-order neighbors, outperform GATs and other baseline models?

RQ 2. Does over-smoothing hinder the exploration of high-order neighbors in GAT and can HONGAT alleviate it?

RQ 3. Does the adjusted averaging range benefit generalization performance and the high-order graph details?

Experimental Datasets

We conduct experiments on three real-world benchmarks tested in the work of GAT (Veličković et al. 2018) — Cora, Citeseer and Pubmed (Sen et al. 2008). For all datasets, 20 nodes per class are used for training, 500 nodes are used for validation and 1000 nodes are used for testing. We follow the transductive setup in (Yang, Cohen, and Salakhudinov 2016) and use random splits. The characteristics of all datasets are summarized in Appendix C. Additional results on large and heterophilic datasets are also reported there.

Implementation Details

We employ Pytorch (Paszke et al. 2017) to implement HONGAT. Following GAT (Veličković et al. 2018), we adopt

Adam optimizer (Kingma and Ba 2015) with learning rate 0.005 and L_2 regularization with $\lambda = 0.0005$ for Cora and Citeseer. For Pubmed, we strengthen the learning rate to 0.01 and the L_2 regularization with $\lambda = 0.001$. For all datasets, we use early stopping with a window size of 100 and report mean $\pm$ std accuracy over 10 runs.

To construct HONGAT, we choose $K = 10$. For all datasets, we preprocess the input features by a 2-layer MLP with 64 hidden units and employ a single HONGAT layer, followed by a softmax activation. Dropout (Srivastava et al. 2014) with $p = 0.6$ is applied to each layer's input. For weights in Equation (5), we initialize $\{\alpha_k\}_{k=0}^{K}$ with random initialization in Pytorch and initialize β to 0. For weights in Equation (8), we initialize γ_K to 1 and others to 0. The above setting is equivalent to beginning training without using the masking matrix.

Compared Methods

HONGAT is firstly compared with state-of-the-art GAT and its variants, including SPAGAN (Yang et al. 2019), ADSF (Zhang et al. 2020), GAT^3 and GAT^{10}. SPAGAN and ADSF are state-of-the-art GAT variants, which are also designed for full exploration of graph information. Specifically, SPAGAN introduces a path-based attention when updating node features. ADSF encodes structural details into GAT layers with an adaptive fingerprint. GAT^3 and GAT^{10} are respectively a 3-layer and 10-layer GAT. In particular, GAT^{10} has the same receptive field size as we apply in HONGAT, which ensures a fair comparison.

We also compare HONGAT with strong baseline models, including i) MLP, which uses the attribute information of nodes; ii) DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), a graph embedding method based on random walk; iii) Chebyshev (Defferrard, Bresson, and Vandergheynst 2016), a spectral method which defines graph convolution using Chebyshev polynomials; iv) JKNet (Xu et al. 2018), which incorporates the outputs of different layers to preserve the locality of node features; v) GCN (Kipf and Welling 2017), which employs a predefined propagation matrix to approximate the first-order Chebyshev and can be thought of as a special case of attention, where the attention score for each neighbor mainly depends on the fixed adjacency matrix; vi) SGC (Wu et al. 2019), a simplified version of graph convolution architecture which removes all the nonlinearities between GCN layers; vii) APPNP (Klicpera, Bojchevski, and Günnemann 2019) and GPR-GNN (Chien et al. 2021), which combine GNNs with PageRank techniques. Moreover, S^{2}GC (Zhu and Koniusz 2021) which aggregates diffusion matrices over K steps and SIGN (Frasca et al. 2020) which employs multi-sized convolutional operators, are also compared.

Comparison Results

In this section, we answer the three questions we raise via the experimental comparisons.

RQ 1. Does the proposed method, with high-order neighbors, outperform GATs and other baseline models?

Table 1 summarizes the comparison of HONGAT and other compared methods. From the results, we observe that HONGAT consistently outperforms all the baseline models and achieves an average accuracy improvement of 2.2% compared to GAT, which is non-marginal. Specifically, HONGAT achieves a 2.3%, 1.6% and 2.7% accuracy improvement on Cora, Citeseer and Pubmed. Besides, baseline models generally do not work as good as GATs and HONGAT. This confirms the necessity and promising results of exploring informative high-order neighbors.

Method	Cora	Citeseer	Pubmed
MLP	56.7±2.1%	55.4±2.2%	73.9±0.8%
DeepWalk	65.4±2.0%	52.0±1.8%	67.7±0.7%
Chebyshev	76.3±1.6%	66.7±1.6%	77.5±0.3%
JKNet	76.4±2.5%	62.6±3.4%	77.3±0.6%
SGC	78.3±1.0%	69.0±0.9%	75.4±1.7%
GCN	80.0±1.7%	68.1±1.8%	78.5±0.6%
APPNP	82.1±1.4%	69.2±1.3%	79.6±1.7%
GPR-GNN	82.1±1.0%	68.7±1.8%	79.4±2.1%
S^{2}GC	82.1±0.4%	68.9±0.9%	79.8±0.6%
SIGN	81.7±1.0%	69.1±0.8%	79.6±1.5%
GAT	81.2±0.7%	68.4±1.4%	78.9±0.6%
GAT^3	80.1±1.7%	67.3±2.6%	76.9±2.0%
GAT^{10}	66.3±2.5%	20.8±5.0%	42.3±3.4%
SPAGAN	82.0±0.7%	69.0±1.5%	79.5±0.5%
ADSF	82.3±0.8%	69.1±1.8%	80.0±0.7%
HONGAT	**83.1±1.0%**	**69.5±1.2%**	**81.1±0.8%**

Table 1: Comparison of classification (mean accuracy $\pm$ std (%)) on real datasets. The best results are shown in bold.

RQ 2. Does over-smoothing hinder the exploration of high-order neighbors in GAT and can HONGAT alleviate it?

When focusing on the results of GAT^3 and GAT^{10} in Table 1, we find that the increased attention layers always lead to worse performance. This indicates GAT's failure to explore high-order neighbors caused by the over-smoothing issue. To further study the over-smoothing effect and the validity of our method in alleviating it, we provide a spectral analysis as shown in Figure 3. Specifically, the first row shows the spectrums of the graph signals output by GAT, HONGAT and GAT^{10} on the Cora, Citeseer and Pubmed datasets. The second row shows the corresponding frequency responses $h(\tilde{\lambda})$ recovered from the output signals and the original ones. We employ symmetrically normalized Laplacian $\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$. From Figure 3, we find that the differences between the components of the graph signals output by GAT^{10} are significantly eliminated, which reflects the over-smoothing issue caused by multi-layer stacking and explains the degraded performance of GAT^{10}. Nevertheless, the outputs of HONGAT remain distinguishable, which validates its ability to escape from the over-smoothing risk and explore high-order-neighbor information.

Another observation is that, when compared with GAT, HONGAT suppresses high-frequency noises better as well as preserves more informative low-frequency components. Specifically, when focusing on the spectrums of output sig-

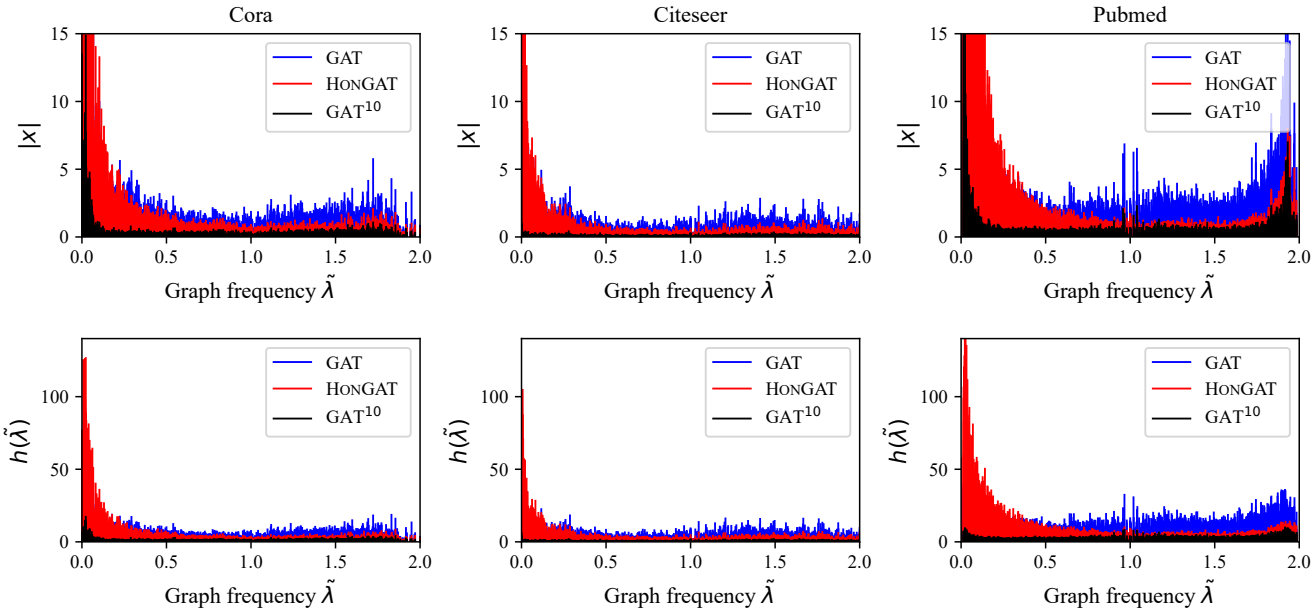


Figure 3: Spectral analysis for GAT, HONGAT and GAT^{10}. The first row shows the spectrums of graph signals output by the three models. The second row shows the corresponding frequency responses $h(\tilde{\lambda})$ recovered from the output signals and the original ones.

Dataset	Model	Acc	$Ratio_1$ / $Ratio_2$ / $Ratio_3$
Cora	HONGAT†	82.1±1.5%	8.9% / 18.0% / 27.2%
	HONGAT	**83.1±1.0%**	**9.3% / 18.5% / 27.9%**
Citeseer	HONGAT†	69.3±1.4%	9.0% / **18.0%** / 27.0%
	HONGAT	**69.5±1.2%**	**9.5%** / 17.9% / **27.4%**
Pubmed	HONGAT†	80.7±1.3%	9.5% / 18.7% / 27.3%
	HONGAT	**81.1±0.8%**	**9.8% / 19.1% / 28.0%**

Table 2: Comparison of classification (mean accuracy ± std (%)) and high-order KCN index ratios ($Ratio_1$ /$Ratio_2$/$Ratio_3$ (%)) obtained by HONGAT and HONGAT with the removed mask. The best results and highest ratios are shown in bold.

nals and the recovered frequency responses, the red lines corresponding to HONGAT are always higher than the blue ones corresponding to GAT in the range of low frequency, which means, by leveraging high-order neighbors, HONGAT further amplifies the useful low-frequency components. While in high frequency range, the red lines are always lower than the blue ones, which means the harmful high-frequency noises have been well attenuated by our method.

RQ 3. Does the adjusted averaging range benefit generalization performance and the high-order graph details?

To validate the benefits of the proposed new masking matrix, we compare the classification performance of HONGAT and HONGAT with the removed mask. We use HONGAT† to describe the latter. The results are shown in Table 2, where the full HONGAT model consistently achieves better performance. This implies that HONGAT, by learning a suitable averaging range, is more powerful in identifying valuable high-order neighbors. In addition, we compare the ratios of the highest order, the highest two order and the highest three order KCN indices employed ($Ratio_1$, $Ratio_2$ and $Ratio_3$ for short) to examine the effect of new masking in exploring high-order graph details. A higher ratio of the high-order KCN indices used implies a more global utilization of graph information because a longer "chain" of high-order neighbors is explored when computing averaging weights. The results are shown in Table 2, where we find that $Ratio_1$, $Ratio_2$ and $Ratio_3$ are clearly improved, which shows the effectiveness of proposed masking mechanism in exploiting high-order graph information.

We further confirm this result in Figure 4. Specifically, for each dataset, we set $K = 10$ and increase order of averaged neighbors (described by K_a) from 1 to 10. We then record how $Ratio_1$, $Ratio_2$ and $Ratio_3$ change with K_a. The values of $Ratio_2$ and $Ratio_3$ are scaled and maximum values are marked with stars. As can be seen, when K_a is close to or equal to K, $Ratio_1$, $Ratio_2$ and $Ratio_3$ are usually lower than those in the cases of $K_a < K$. This means by adjusting the averaging range with the proposed new masking matrix, the exploration of high-order graph details can be improved.

Ablation Study

An ablation study is also conducted to validate the significance of the proposed common-neighbor-based topological similarity. Specifically, the classification performance of HONGAT with only semantic similarity considered, namely $\{\alpha_k\}_{k=0}^{K} = 0$ (described by HONGAT w/o TOPO^1), and the

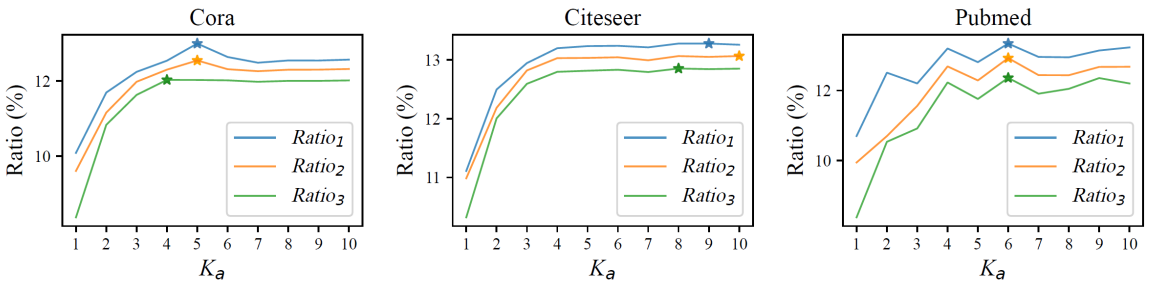


Figure 4: Verification of improved graph exploration with adjusted averaging range. The curves in different colors reveal that when $K = 10$, how the ratios of the highest order, the highest two order and the highest three order KCN indices employed change with order of aggregated neighbors K_a.

Method	Cora	Citeseer	Pubmed
HONGAT w/o TOPO^1	20.5±4.9%	27.9±6.1%	43.6±5.4%
HONGAT w/o TOPO^2	18.0±6.5%	18.6±1.8%	30.1±4.1%
HONGAT w/o SEM	82.7±0.9%	69.2±1.1%	80.7±0.9%
HONGAT	**83.1±1.0%**	**69.5±1.2%**	**81.1±0.8%**

Table 3: Comparison of classification (mean accuracy ± std (%)) of HONGAT with only semantic or topological similarity and full HONGAT. The best results are shown in bold.

full HONGAT model is reported. In this case, masking matrix provides adaptive weights for different orders of neighbors. Another way to utilize semantic information is also compared where $\{\alpha_k\}_{k=0}^{K} = 0$ and the masking matrix is removed (described by HONGAT w/o TOPO^2). The results are shown in Table 3. It can be seen that although having the same receptive field, HONGAT with only semantic similarity fail to maintain the effectiveness of HONGAT. This suggests the necessity of using the topological similarity revealed by common neighbors.

Furthermore, it is natural to think that whether the semantic similarity can be ignored via letting $\beta = 0$ in Equation (5). Therefore, the results of the HONGAT model that only contains topological similarity are also compared, which is described by HONGAT w/o SEM. As shown in Table 3, HONGAT without semantic similarity considered usually performs worse than the full HONGAT model, which confirms the positive impact of semantic information.

Relationship with Existing Methods

Our algorithm is discussed and compared with existing GNN methods (mainly divided into attention-based and spectral-based methods).

Attention-Based Methods. By attending to most relevant neighbors, GAT methods have shown great success in various tasks (Song et al. 2019; Huang and Carley 2019; Wang et al. 2019a; Park et al. 2020; Rong et al. 2020; Wang et al. 2020; Dong et al. 2022). Some works adopt attention mechanisms other than the standard one used in GAT (Zhang et al. 2018; Busbridge et al. 2019; Wang et al. 2019b; Zeng et al. 2021). However, high-order neighbors are still largely unemployed in GATs. Recently, there are works trying to incorporate global graph details into graph attention networks. For example, SPAGAN (Yang et al. 2019) conducts path-based attention and ADSF (Zhang et al. 2020) encodes structural information into an adaptive fingerprint. Our method is different in two ways. First, common-neighbor-based similarity is employed as a new way to introduce graph topology and enable the exploration of high-order neighbors, which ensures a polynomial-like frequency response for HONGAT. Second, the averaging range is explicitly adjusted in HONGAT to further optimize the high-order neighbor utilization. Experiments demonstrate the performance gains of HONGAT over these methods.

Spectral-Based Methods. Let $\beta = 0$ in Equation (5) and remove the mask, the proposed HONGAT layer will degenerate to a polynomial filter with free coefficients, which implies popular spectral methods like SGC (Wu et al. 2019), APPNP (Klicpera, Bojchevski, and Günnemann 2019) and GPR-GNN (Chien et al. 2021) can be seen as special cases of HONGAT. As demonstrated by the above empirical results, HONGAT conducts a more informative exploration of graphs and achieves improved performance on all benchmarks. Besides, S^{2}GC (Zhu and Koniusz 2021) aggregates diffusion matrices over K steps and SIGN (Frasca et al. 2020) employs multi-sized convolutional operators for extending the neighborhood size. HONGAT is notably different in that it is derived from GATs and it can exploit similarity information embedded in feature space. In addition, we find that, under the common-neighbor-based attention framework, other polynomial filters can also be interpreted as attention networks without consideration of the semantic similarity $a_{i,j}$, which is further discussed in Appendix E.

Conclusion

In this paper, we tackle a crucial issue of GAT, that is, the failure to explore high-order neighbors. We propose a simple and effective HONGAT model to explore high-order neighbors for GAT, which adopts two tightly coupled novel technologies: *common neighbor similarity* and *new masking matrix*. Empirical results on real-world benchmark datasets show that, by utilizing high-order neighbors, HONGAT always performs better than GAT and other baseline methods.

Acknowledgments

This research was supported by the National Science Foundation of China (62176118, 62306133).

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations*.

Busbridge, D.; Sherburn, D.; Cavallo, P.; and Hammerla, N. Y. 2019. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive universal generalized pagerank graph neural network. In *Proceedings of the International Conference on Learning Representations*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*, 3844–3852.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Dong, Y.; Liu, Q.; Du, B.; and Zhang, L. 2022. Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification. *IEEE Transactions on Image Processing*, 31: 1559–1572.

Fout, A.; Byrd, J.; Shariat, B.; and Ben-Hur, A. 2017. Protein interface prediction using graph convolutional networks. In *Advances in Neural Information Processing Systems*, 6530–6539.

Frasca, F.; Rossi, E.; Eynard, D.; Chamberlain, B.; Bronstein, M.; and Monti, F. 2020. Sign: Scalable inception graph neural networks. In *Graph Representation Learning and Beyond (GRL+) Workshop at the 37th International Conference on Machine Learning*.

Ge, Y.; Huang, C.; Liu, Y.; Zhang, S.; and Kong, W. 2024. Unsupervised social network embedding via adaptive specific mappings. *Frontiers of Computer Science*, 18(3): 183310.

Gori, M.; Monfardini, G.; and Scarselli, F. 2005. A new model for learning in graph domains. In *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, 729–734.

Hamaguchi, T.; Oiwa, H.; Shimbo, M.; and Matsumoto, Y. 2017. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1802–1808.

Huang, B.; and Carley, K. M. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 5472–5480.

Jia, L.; Guo, L.; Zhou, Z.; and Li, Y. 2024. LAMDA-SSL: a comprehensive semi-supervised learning toolkit. *Science China Information Sciences*, 67(1): 117101–117102.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then propagate: Graph neural networks meet personalized pagerank. In *Proceedings of the International Conference on Learning Representations*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature*, 521(7553): 436–444.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3538–3545.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1412–1421.

Oono, K.; and Suzuki, T. 2020. Graph neural networks exponentially lose expressive power for node classification. In *Proceedings of the International Conference on Learning Representations*.

Park, C.; Lee, C.; Bahng, H.; Tae, Y.; Jin, S.; Kim, K.; Ko, S.; and Choo, J. 2020. ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 1215–1224.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshop on Autodiff*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710.

Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; and Huang, J. 2020. Self-supervised graph transformer on large-scale molecular data. In *Advances in Neural Information Processing Systems*, 12559–12571.

Salton, G.; and McGill, M. 1984. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI Magazine*, 29(3): 93–93.

Shi, J.; Wei, T.; and Li, Y. 2024. Residual diverse ensemble for long-tailed multi-label text classification. *Science China Information Sciences*.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3): 83–98.

Song, W.; Xiao, Z.; Wang, Y.; Charlin, L.; Zhang, M.; and Tang, J. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 555–563.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations*.

Wang, K.; Shen, W.; Yang, Y.; Quan, X.; and Wang, R. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3229–3238.

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019a. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 950–958.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019b. Heterogeneous graph attention network. In *Proceedings of the The World Wide Web Conference*, 2022–2032.

Wei, T.; Wang, H.; Tu, W.; and Li, Y. 2022. Robust model selection for positive and unlabeled learning with constraints. *Science China Information Sciences*, 65(11): 212101.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *Proceedings of the International Conference on Machine Learning*, 6861–6871.

Wu, Y.; Lian, D.; Xu, Y.; Wu, L.; and Chen, E. 2020. Graph convolutional networks with markov random field reasoning for social spammer detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1054–1061.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, 2048–2057.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the International Conference on Machine Learning*, 5453–5462.

Yang, Y.; Wang, X.; Song, M.; Yuan, J.; and Tao, D. 2019. Spagan: Shortest path graph attention network. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 4099–4105.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the International Conference on Machine Learning*, 40–48.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7370–7377.

Zeng, J.; Liu, T.; Jia, W.; and Zhou, J. 2021. Fine-grained Question-Answer sentiment classification with hierarchical graph attention network. *Neurocomputing*, 457: 214–224.

Zhang, J.; Shi, X.; Xie, J.; Ma, H.; King, I.; and Yeung, D.-Y. 2018. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, 339–349.

Zhang, K.; Zhu, Y.; Wang, J.; and Zhang, J. 2020. Adaptive structural fingerprints for graph attention networks. In *Proceedings of the International Conference on Learning Representations*.

Zhou, T.; Lü, L.; and Zhang, Y.-C. 2009. Predicting missing links via local information. *The European Physical Journal B*, 71(4): 623–630.

Zhou, Z.; Jin, Y.; and Li, Y. 2024. Rts: learning robustly from time series data with noisy label. *Frontiers of Computer Science*, 18(6): 186332.

Zhu, H.; and Koniusz, P. 2021. Simple spectral graph convolution. In *Proceedings of the International Conference on Learning Representations*.

Zhu, Y.; Geng, Y.; Li, Y.; Qiang, J.; and Wu, X. 2024. Representation learning: serial-autoencoder for personalized recommendation. *Frontiers of Computer Science*, 18(4): 184316.