Neural Networks 174 (2024) 106225

Contents lists available at ScienceDirect

Neural Networks

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

Full Length Article

Attributed Multi-Order Graph Convolutional Network for Heterogeneous Graphs

Zhaoliang Chen [a,b], Zhihao Wu [a,b], Luying Zhong [a,b], Claudia Plant [c,d], Shiping Wang [a,b], Wenzhong Guo [a,b,*]

[a] College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China
[b] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China
[c] Faculty of Computer Science, University of Vienna, Vienna 1090, Austria
[d] ds:UniVie, Vienna 1090, Austria

ARTICLE INFO

Keywords:
Heterogeneous graphs
Graph convolutional networks
Multi-order adjacency matrix
Semi-supervised classification

ABSTRACT

Heterogeneous graph neural networks play a crucial role in discovering discriminative node embeddings and relations from multi-relational networks. One of the key challenges in heterogeneous graph learning lies in designing learnable meta-paths, which significantly impact the quality of learned embeddings. In this paper, we propose an Attributed Multi-Order Graph Convolutional Network (AMOGCN), which automatically explores meta-paths that involve multi-hop neighbors by aggregating multi-order adjacency matrices. The proposed model first constructs different orders of adjacency matrices from manually designed node connections. Next, AMOGCN fuses these various orders of adjacency matrices to create an intact multi-order adjacency matrix. This process is supervised by the node semantic information, which is extracted from the node homophily evaluated by attributes. Eventually, we employ a one-layer simplifying graph convolutional network with the learned multi-order adjacency matrix, which is equivalent to the cross-hop node information propagation with multi-layer graph neural networks. Substantial experiments reveal that AMOGCN achieves superior semi-supervised classification performance compared with state-of-the-art competitors.

1. Introduction

Graphs are universally available in practical applications, such as links between websites, paper citations, user clicks and purchase data. As a powerful tool to process graph-structured data, graph neural networks have been extensively investigated in recent years (Chen, Wu, et al., 2023; Pan, Chen, & He, 2023; Wu, Zhang, & Fan, 2023; Xia, Wang, Yang, et al., 2022; Xu, Xia, Gao, Han, & Gao, 2021). For instance, Graph Convolutional Network (GCN) (Kipf & Welling, 2017) and its variants have been widely leveraged in a large number of models owing to its excellent ability to capture discriminative multi-hop node embeddings from neighbors (Chen, Fu, et al., 2023; Chen, Wu, Wang & Guo, 2023; Li, Xu, Chen, Zheng, & Liu, 2021; Wu et al., 2022; Xia, Wang, Gao, Zhang, & Gao, 2022). In real world, attributed to various connections among objects, learning intact node features from heterogeneous graphs has become a critical research problem. Accordingly, this has encouraged researchers to put more emphasis on the Heterogeneous Graph Neural Network (HGNN) for the dexterous manipulation of multi-graph-structured data (Liu et al., 2023; Xia, Gao, et al., 2022; Zhao et al., 2022).

Most HGNNs explored node embeddings on the basis of manually designed meta-paths (paths including different dotted lines in Fig. 1), and aggregated node information from neighbors discovered by meta-paths (Qian, Zhang, Wen, Ye, & Zhang, 2022; Wang et al., 2019, 2022). For instance, in citation networks, meta-paths w.r.t. authors can be distilled from relations between authors and their papers. Despite the great success of meta-path-based HGNNs, these models still face the following problems. On the one hand, the performance of these methods is tightly dependent on the quality of meta-paths, and models may encounter a decline in accuracy when some meta-paths are not adequately reliable. A simple experiment illustrating this issue is shown in Table 1, where meta-paths have different semantics in distinct scenarios and are precomputed in prior works. The performance of GCN varies significantly with differently defined meta-paths on these widely utilized heterogeneous graph datasets. It is obvious that GCN experiences unfavorable performance with some meta-paths, which results in the undesired classification accuracy when a weighted meta-path is adopted. This observation indicates that a manually defined meta-path extracted from one or two hops of neighbors is not always

* Corresponding author.
E-mail address: guowenzhong@fzu.edu.cn (W. Guo).

https://doi.org/10.1016/j.neunet.2024.106225
Received 17 April 2023; Received in revised form 17 January 2024; Accepted 2 March 2024
Available online 4 March 2024

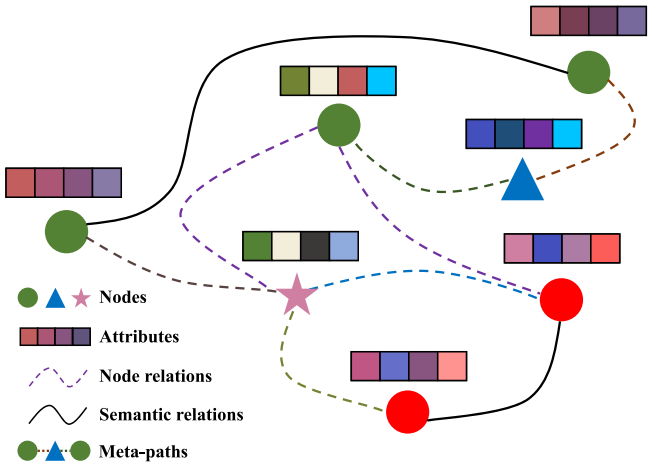

Fig. 1. Example of an attributed heterogeneous graph with multiple types of node connections, where different colors of nodes and dotted lines denote various classes of nodes and distinct node relations respectively. Distinct types of nodes are represented by various shapes. The black lines denote the semantic relations that indicate the attributed node homophily.

Table 1
Classification performance (Macro-F1) of GCN with different manually defined meta-paths, where weighted fusion indicates that GCN adopts a weighted aggregation of all meta-paths as the adjacency matrix.

Meta-path IDs/Datasets	ACM	DBLP	IMDB	YELP
Meta-path 1	0.874	0.841	0.467	0.589
Meta-path 2	0.681	0.905	0.502	0.385
Meta-path 3	–	0.725	0.201	0.556
Weighted fusion	0.778	0.860	0.241	0.544

satisfactory, and may influence the performance of HGNNs. To address this drawback, recent works have endeavored to design learnable multi-length meta-paths from short-length meta-paths (Li, Jin, et al., 2021; Wang, Song, Li, Zhang, & Han, 2018; Yu et al., 2022; Yun, Jeong, Kim, Kang, & Kim, 2019). Nevertheless, these methods generally considered a sequential combination of different meta-paths, or only studied refined node relations based on basic meta-paths. Hence, a more sophisticated exploration of multi-length meta-paths should be developed. On the other hand, most models ignored the natural homophily between attributes of nodes, which is pivotal for finding neighbors in many graph learning methods. In most cases, long-length meta-paths can connect highly correlated nodes that are remote in the topological space. Nevertheless, meta-path-based HGNNs often ignored such semantic connections extracted from node attributes, which also benefits the adaptive learning of multi-length meta-paths.

To better clarify our motivation, we refer to Fig. 1 again to depict an example of an attributed heterogeneous graph. There are three types of nodes with attributes in the graph, based on which we can propagate node information across different types of nodes via generating meta-paths. Although some similar nodes are not directly connected, their relations can be captured by the aggregation of meta-paths derived from other types of nodes and relations. Consequently, we assume that there exists consistency between the node homophily extracted from node attributes and long-length meta-paths. In other words, some nodes connected with multi-length meta-paths that are derived from the fusion of several manually defined short-length meta-paths sometimes share similar node attributes, which motivates us to make full use of node similarity to guide the design of new long-length meta-paths. In conclusion, semantic node similarities are generally consistent with meta-path-based node connections, attributed to which we aim to build a generalized HGNN framework to automatically learn multi-length meta-path fusion under the supervision of node feature homophily.

Therefore, in this paper, we propose an Attributed Multi-Order Graph Convolutional Network (AMOGCN) to learn adaptive multi-order (multi-length) meta-paths from heterogeneous graphs, which is supervised by the semantic node connections extracted from node attribute similarities. The proposed AMOGCN explores different orders of meta-paths to automatically seek multi-hop node neighbors as much as possible, which dexterously considers different permutations of first-order meta-paths via continuous coefficient matrices. After that, AMOGCN combines multi-order adjacency matrices with trainable weights. The aggregation process is guided by the semantic node homophily, which encourages the model to discover high-order neighbors that have similar attributes. Eventually, AMOGCN applies a one-layer simplifying GCN to learn high-order node embeddings from the adaptive multi-order meta-path, which is equivalent to the fusion of different multi-layer GCNs. In summary, the primary contributions of this paper are listed as follows:

1. We develop a multi-order GCN framework that addresses attributed heterogeneous graphs via constructing distinct lengths of meta-paths from several 1-length node relations, which is equivalent to the accumulation of multi-hop embeddings captured from various multi-layer GCNs.
2. The aggregation of multi-order meta-paths is supervised by the semantic information extracted from the homophily between attributed nodes, which contributes to the investigation of highly correlated nodes that are remote in the topological space.
3. We evaluate the proposed model on various heterogeneous graph datasets, and compare it with state-of-the-art works, which reveals that AMOGCN achieves competitive performance in terms of semi-supervised classification tasks.

The rest of this paper is organized as follows. We introduce related works w.r.t. GCNs and heterogeneous graph learning in Section 2. In Section 3, some basic concepts of heterogeneous graphs are clarified. We elaborate on the proposed model in Section 4, and conduct substantial experiments to verify it in Section 5. Eventually, we conclude our work in Section 6.

2. Related works

In this section, we review and discuss some related works w.r.t. GCN and heterogeneous graph learning, including computations of high-order adjacency matrices and learnable meta-paths, which are the primary topics of this paper.

2.1. Graph convolutional networks

GCN has significantly improved the learning performance of graph neural networks, each layer of which is defined as

$$\mathbf{H}^{(l)} = \sigma(\mathbf{A}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}), \tag{1}$$

where $\mathbf{A}$ is the renormalized adjacency matrix and $\mathbf{W}^{(l)}$ is the layer-specific weight matrix. $\sigma(\cdot)$ is the non-linear activation function. Wu et al. further simplified GCN by removing the activation function and precomputing a high-order adjacency matrix, which was termed as Simplifying Graph Convolution (SGC) (Wu et al., 2019). SGC applied the high-order adjacency matrix in a one-layer GCN to approximate the information propagation of a multi-layer GCN, i.e.,

$$\mathbf{H}^{(l)} = \mathbf{A}^{l}\mathbf{X}\mathbf{W}, \tag{2}$$

where $\mathbf{A}^{l}$ is the lth power of $\mathbf{A}$ and $\mathbf{W}$ is a trainable weight. Actually, SGC aimed to explore high-order neighbors efficiently with a precomputed adjacency matrix. A large number of studies have demonstrated the encouraging performance of GCN and SGC. Feng et al. designed a cross-feature graph convolution to model the arbitrary-order across node features (Feng, He, Zhang, & Chua, 2023). Min et al. proposed

geometric scattering transformations and residual convolutions to enhance the conventional GCN (Min, Wenkel, & Wolf, 2020). Xu et al. presented a GCN-based deep feature aggregation model to perform the high spatial resolution scene classification task (Xu, Huang, Deng, & Li, 2021). Wu et al. introduced a robust tensor GCN framework to promote the robustness of the model with the multi-view augmentation (Wu et al., 2022). Zhu et al. aggregated k-step diffusion matrices and proposed a simple spectral graph convolution on the basis of SGC (Zhu & Koniusz, 2021). Wang et al. learned an aggregation of node embeddings from topology space and feature space via a multi-channel GCN (Wang et al., 2020). Zhang et al. also leveraged the schema of SGC for attributed graph clustering (Zhang, Liu, Li, & Wu, 2019). Inspired by the PageRank techniques on graphs (Jeh & Widom, 2003), Gasteiger et al. proposed the personalized propagation of neural predictions, which utilizes the PageRank scores to encode the local neighborhood nodes of each root node (Klicpera, Bojchevski, & Günnemann, 2019). Moreover, Bojchevski et al. presented a new GNN model for large-scale graphs by utilizing an adapted propagation schema of approximated personalized PageRank (Bojchevski et al., 2020). Nevertheless, these methods were generally developed for homogeneous graphs, which could not handle more complex multi-relational networks.

2.2. Heterogeneous graph learning

Owing to the complex relations between objects in the real world, numerous works have paid attention to heterogeneous graph learning with graph neural networks. A key idea of HGNNs is to discover the consistency and complementarity of different graphs via a joint framework or a fusion strategy. Yang et al. presented a GCN-based multi-graph fusion method with pseudo-label supervision (Yang et al., 2023). Sadikaj et al. designed a joint dimensionality reduction algorithm for multi-relational graphs with node attributes, and applied it to the spectral clustering (Sadikaj, Velaj, Behzadi, & Plant, 2021). Park et al. proposed an unsupervised node embedding method for attributed multiplex networks, which jointly integrates node features from distinct graphs (Park, Kim, Han, & Yu, 2020). Zhao et al. presented a heterogeneous graph structure learning framework that fuses heterogeneous graphs with a graph neural network (Zhao et al., 2021). A critical factor to achieve satisfactory performance with HGNNs is the design of meta-paths, which significantly affects the quality of learned node embeddings. Some state-of-the-art works have focused on the automatic learning of meta-paths. A classical method for meta-path learning is the random walk algorithm (Dong, Chawla, & Swami, 2017; Grover & Leskovec, 2016; Shi, Hu, Zhao, & Philip, 2018). Yun et al. built graph transformer layers to explore a soft selection of edge types and generate complex multi-hop neighborhood connections (Yun et al., 2019). Several works have attempted to conduct automatic meta-path selections with HGNNs, which select and maintain critical meta-paths between nodes (Li, Jin, et al., 2021; Wang et al., 2018). Yu et al. proposed a multiplex heterogeneous GCN which adaptively learns multi-length meta-paths via fusing outputs from different depths of graph convolutional layers (Yu et al., 2022). Nonetheless, these automatic graph learning methods seldom considered the permutations of different basic node relations, which are helpful to explore better complementary information from different graphs. Moreover, existing methods for constructing multi-length meta-paths generally lacked suitable supervision. In light of this, we propose a multi-order GCN framework that learns adaptive multi-length meta-paths, which is under the supervision of node homophily discovered from node attributes.

3. Preliminary

In this section, we first introduce some important definitions w.r.t. heterogeneous graphs in this paper.

Table 2
A summary of primary notations in this paper.

Notations	Explanations
$\mathbf{X}$	Node attributes.
$\mathbf{R}_i$	The ith node relation matrix.
$\mathbf{n}_i, \mathbf{m}_i$	The numbers of different types of nodes.
$\mathbf{A}_i$	The ith first-order adjacency matrix.
$\mathbf{A}_S$	Semantic adjacency matrix.
$\mathbf{A}^{(l)}$	The arbitrary lth order adjacency matrix.
$\mathbf{A}^{(t,l)}$	The tth lth order adjacency matrix.
$\boldsymbol{\alpha}^{(t,l)}$	Coefficient matrix for the tth lth order meta-path.
L	Maximum order of high-order adjacency matrices.
$\mathbf{W}$	Trainable weight matrix of SGC.
$\mathcal{A}$	Multi-order adjacency matrix.
$\mathbf{H}^{(t,l)}$	The tth lth order node embeddings.
$\mathbf{Z}$	Multi-order node embeddings.
$\beta^{(t,l)}$	Weight for the tth lth order adjacency matrix.
$\mathbf{Y}$	Ground truth.

Definition 1 (*Heterogeneous Graph Wang et al., 2019*)**.** A heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which includes a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$. With the vertex type mapping function $\phi : \mathcal{V} \to \mathcal{O}$ and the edge type mapping function $\phi : \mathcal{E} \to \mathcal{R}$, the vertex types and edge types in a heterogeneous graph should satisfy $|\mathcal{O}| + |\mathcal{R}| > 2$.

Definition 2 (*Meta-path Wang et al., 2019*)**.** A meta-path is defined as the path in the form of $O_1 \xrightarrow{R_1} O_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{h-1}} O_h$, which depicts the composite relation $A = R_1 \circ R_2 \cdots R_{h-1}$ from vertex types O_1 to O_h, where $\circ$ is the composition operator on relations.

Given a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with distinct types of nodes and relations, we can generate the attributed multi-graph data with various node features $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{|\mathcal{O}|}\}$ and basic node relations $\mathcal{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_{|\mathcal{R}|}\}$. In this paper, we define the 1-length meta-path as the connection between the same type of nodes, which is manually constructed from basic node relations among different types of nodes, which corresponds to a first-order adjacency matrix $\mathbf{A}$. In detail, assume that there is a meta-path manually defined as $O_1 \xrightarrow{R_1} O_2 \xrightarrow{R_2} O_1$, which includes two types of relations R_1 and R_2, and two types of nodes O_1 and O_2. This 1-length directed meta-path can be formulated as a first-order adjacency matrix $\mathbf{A} = \mathbf{R}_1\mathbf{R}_2$, where adjacency matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ store the node relations R_1 and R_2. To illustrate, in an online shopping scenario, users $\mathcal{U}$ have two actions for various items $\mathcal{I}$: purchase and collection, which establish connections between users and items with two different relations. In this context, a 1-length undirected meta-path $\mathbf{A}$ can be built by $\mathcal{U} \overset{purchase}{\longleftrightarrow} \mathcal{I} \overset{collection}{\longleftrightarrow} \mathcal{U}$. Based on various first-order adjacency matrices, we aim to explore multi-order neighbors for nodes and extract node embeddings automatically via an attributed multi-order GCN. The primary mathematical notations used in this paper and their explanations are listed in Table 2.

4. The proposed model

In this section, we illustrate the proposed AMOGCN which copes with attributed heterogeneous graphs. Fig. 2 elaborates on the proposed method. For the sake of sustained explorations of different lengths of meta-paths and their fusion, we calculate adaptive high-order adjacency matrices from extracted first-order meta-paths. In addition, we measure the node similarities from attributes and estimate the semantic node connections from node homophily, which are adopted to supervise the fusion of distinct orders of adjacency matrices. Given that the intact multi-order adjacency matrix involves multi-hop neighbor information, we apply a simplifying graph convolutional network that computes node embeddings from high-order neighborhood structures rather than building multiple graph convolutional layers, which saves the training time and corresponds to the neighborhood propagation of the multi-layer GCN. In the following contents, we introduce

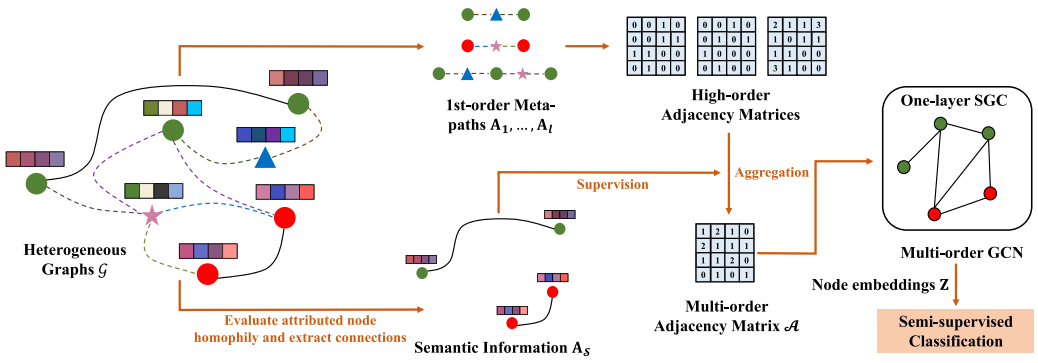

Fig. 2. The framework of the proposed AMOGCN. The model first builds first-order meta-paths from heterogeneous graphs, based on which different orders of adjacency matrices are constructed. After that, the model learns a fused multi-order adjacency matrix from high-order meta-paths, supervised by semantic information extracted from node attributes. Eventually, the proposed model conducts simplifying graph convolution with the learned multi-order adjacency matrix.

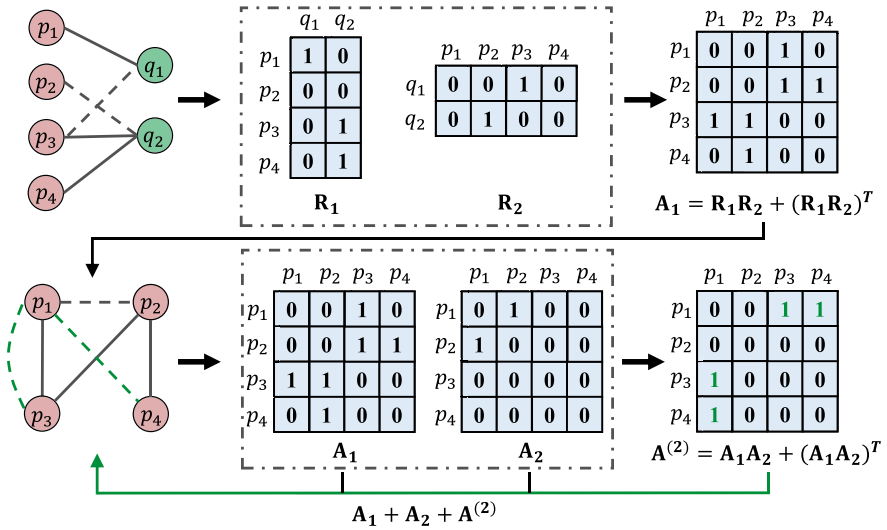

Fig. 3. Example of the multi-order meta-path. In this figure, a first-order meta-path $\mathbf{A}_1$ is constructed from basic node relations. On the basis of 2 different first-order meta-paths $\mathbf{A}_1$ and $\mathbf{A}_2$, a new second-order meta-path $\mathbf{A}^{(2)}$ is calculated, which generates two new relations between nodes (green dotted lines). Fusing all first-order and second-order meta-paths, the new multi-order meta-path is achieved.

and analyze the proposed model via solving the following research problems:

(RP1). How to construct multi-length meta-paths dexterously via learning multiple high-order adjacency matrices and their fusion?

(RP2). How to build a multi-order GCN that efficiently propagates node information over learned multi-length meta-paths?

(RP3). How to adopt node attributes to supervise the fusion of high-order adjacency matrices?

4.1. Multi-order graph convolutional network

In pursuit of exploring different lengths of meta-paths based on the existing node relations or manually defined meta-paths, we design a multi-order GCN to flexibly study node embeddings with various orders of adjacency matrices, which correspond to multi-length meta-paths. First, we provide a concrete example to illustrate the meaning of multi-order adjacency matrices.

Example. Fig. 3 provides a heterogeneous graph which contains two types of nodes p and q, and two types of node relations. Different node relations can be converted to two adjacency matrices $\mathbf{R}_1$ and $\mathbf{R}_2$. For instance, $[\mathbf{R}_1]_{ij}$ records the connection from node p_i to node q_j. We can get an undirected first-order meta-path via $\mathbf{A}_1 = \mathbf{R}_1\mathbf{R}_2 + (\mathbf{R}_1\mathbf{R}_2)^T$. Namely, $\mathbf{A}_1$ stores the undirected meta-path $p \overset{R_1}{\longleftrightarrow} q \overset{R_2}{\longleftrightarrow} p$. Here we temporarily ignore the self-connections of nodes. Assuming that there exists another first-order meta-path $\mathbf{A}_2$ yielded from other node relations, we can derive a new undirected second-order meta-path via $\mathbf{A}^{(2)} = \mathbf{A}_1\mathbf{A}_2 + (\mathbf{A}_1\mathbf{A}_2)^T$, which produces new second-order neighborhood connections. Accordingly, we can construct a new multi-order graph with $\mathcal{A} = \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}^{(2)}$, which consists of the connections to the first-order and the second-order neighbors. If more other first-order meta-paths are considered, we can generate more different high-order meta-paths. Thus, we should design a more generalized framework to fully consider different permutations of low-order meta-paths, which explores potential multi-hop neighbors more completely.

More generally, an l-length (lth order) meta-path can be formulated as a combination of distinct first-order adjacency matrices. For example, a simple l-length meta-path can be represented by a sequential adjacency matrix multiplication, i.e.,

$$\mathbf{A}^{(l)} = \mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_l, \tag{3}$$

where $\mathbf{A}_1 \in \mathbb{R}^{n_1 \times n_2}$, $\mathbf{A}_2 \in \mathbb{R}^{n_2 \times n_3}$, $\cdots$, $\mathbf{A}_l \in \mathbb{R}^{n_l \times n_{l+1}}$. Eq. (3) considers the connections to the l-hop neighbors. Actually, $\mathbf{A}^{(l)}$ can be obtained by arbitrary permutations of the first-order adjacency matrix multiplications, as long as any $\mathbf{A}_i \in \mathbb{R}^{n_i \times m_i}$ and $\mathbf{A}_{i+1} \in \mathbb{R}^{n_{i+1} \times m_{i+1}}$ satisfy $m_i = n_{i+1}$. In order to consider a more flexible fusion of various manually defined

meta-paths, we transform the lth order adjacency matrix in Eq. (3) to

$$\begin{aligned}\mathbf{A}^{(l)} = &\left(\alpha_{11}^{(l)}\mathbf{A}_1 + \alpha_{12}^{(l)}\mathbf{A}_2 + \cdots + \alpha_{1l}^{(l)}\mathbf{A}_l\right)\cdot\\ &\left(\alpha_{21}^{(l)}\mathbf{A}_1 + \alpha_{22}^{(l)}\mathbf{A}_2 + \cdots + \alpha_{2l}^{(l)}\mathbf{A}_l\right)\cdot\\ &\cdots\\ &\left(\alpha_{l1}^{(l)}\mathbf{A}_1 + \alpha_{l2}^{(l)}\mathbf{A}_2 + \cdots + \alpha_{ll}^{(l)}\mathbf{A}_l\right),\end{aligned} \tag{4}$$

where $\mathbf{A}_1, \ldots, \mathbf{A}_l$ are different manually defined 1-length meta-paths describing the relationships between the same type of nodes. Thus, we have $n_1 = n_2 = \cdots = n_{l+1}$. When α_{ij} is a binary coefficient and $\sum_{j=1}^{l}\alpha_{ij} = 1$ for any $i = 1, 2, \ldots, l$, Eq. (4) formulates a high-order adjacency matrix with various permutations of meta-paths with varying α_{ij}. Here, we rewrite Eq. (4) as

$$\mathbf{A}^{(l)} = \text{Product}\left(\boldsymbol{\alpha}^{(l)}\left[\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_l\right]^T\right), \tag{5}$$

where $\text{Product}(\cdot)$ is the cumulative multiplication of vector elements and

$$\boldsymbol{\alpha}^{(l)} = \begin{bmatrix}\alpha_{11}^{(l)} & \cdots & \alpha_{1l}^{(l)}\\ \vdots & \ddots & \vdots\\ \alpha_{l1}^{(l)} & \cdots & \alpha_{ll}^{(l)}\end{bmatrix} \tag{6}$$

is the binary coefficient matrix that the sum of each row is 1. In particular, when $\boldsymbol{\alpha}^{(l)} = \mathbf{I}$, Eq. (5) is exactly a sequential cumulative multiplication of adjacency matrices as Eq. (3). Herein, we relax the discrete coefficient matrix into a continuous one, and renormalize it by a softmax function, i.e.,

$$\boldsymbol{\alpha}^{(l)} = \text{Softmax}\left(\boldsymbol{\alpha}^{(l)}, \text{dim} = 1\right), \tag{7}$$

which ensures that $\sum_{j=1}^{l}\alpha_{ij}^{(l)} = 1$. With a directed $\mathbf{A}^{(l)}$, we can further obtain an lth order undirected adjacency matrix via

$$\mathbf{A}^{(l)} = \mathbf{A}^{(l)} + \mathbf{A}^{(l)^T} - \text{diag}\left(\mathbf{A}^{(l)}\right). \tag{8}$$

In this paper, we define the maximum order of meta-paths as L, where L is the number of the first-order meta-paths. When $l < L$, we explore different selections of basic meta-paths to generate $\mathbf{A}^{(l)}$. For example, when $L = 3$, we can compute $\mathbf{A}^{(2)}$ with subsets $\{\mathbf{A}_1, \mathbf{A}_2\}$, $\{\mathbf{A}_1, \mathbf{A}_3\}$ and $\{\mathbf{A}_2, \mathbf{A}_3\}$. Notice that Eq. (5) has considered various permutations of the given first-order adjacency matrices implicitly. To simplify, we denote the set of all lth order adjacency matrices as $\boldsymbol{\Psi}^{(l)} = \{\mathbf{A}^{(t,l)}\}_{t=1}^{|\boldsymbol{\Psi}^{(l)}|}$, where $|\boldsymbol{\Psi}^{(l)}| = \frac{L!}{l!(L-l)!}$.

Since we have obtained high-order adjacency matrices that contain multi-hop neighborhood information, we only need to adopt a one-layer GCN to propagate node attributes. Inspired by SGC (Wu et al., 2019), we remove the non-linear activation function and formulate the tth lth order GCN as

$$\begin{aligned}\mathbf{H}^{(t,l)} = &\left(\alpha_{11}^{(t,l)}\mathbf{A}_1 + \alpha_{12}^{(t,l)}\mathbf{A}_2 + \cdots + \alpha_{1l}^{(t,l)}\mathbf{A}_l\right)\\ &\left(\alpha_{21}^{(t,l)}\mathbf{A}_1 + \alpha_{22}^{(t,l)}\mathbf{A}_2 + \cdots + \alpha_{2l}^{(t,l)}\mathbf{A}_l\right)\\ &\cdots\\ &\left(\alpha_{l1}^{(t,l)}\mathbf{A}_1 + \alpha_{l2}^{(t,l)}\mathbf{A}_2 + \cdots + \alpha_{ll}^{(t,l)}\mathbf{A}_l\right)\\ &\mathbf{X}\mathbf{W}^{(t,1)}\mathbf{W}^{(t,2)}\cdots\mathbf{W}^{(t,l)},\end{aligned} \tag{9}$$

where $\mathbf{A}_1, \ldots, \mathbf{A}_l$ are selected from L precomputed first-order meta-paths. Herein, $\alpha_{ij}^{(t,l)}$ is the trainable probability. Particularly, given an exact j, when $\alpha_{ij}^{(t,l)} = 1$ for all $i = 1, \ldots, l$ and 0 otherwise, we obtain an lth order adjacency matrix $\mathbf{A}^{(t,l)} = \mathbf{A}_j^l$ based on the jth first-order meta-path, i.e., the lth power of $\mathbf{A}_j$. In this case, Eq. (9) becomes a standard SGC without non-linear activation functions, i.e.,

$$\mathbf{H}^{(t,l)} = \mathbf{A}_j^l\mathbf{X}\mathbf{W}^{(t,1)}\mathbf{W}^{(t,2)}\cdots\mathbf{W}^{(t,l)}. \tag{10}$$

It can be regarded as the l-hop graph information propagation over the topology network denoted by $\mathbf{A}_j^l$.

Replacing $\mathbf{W}^{(t,1)}\mathbf{W}^{(t,2)}\cdots\mathbf{W}^{(t,l)}$ with a shared trainable matrix $\mathbf{W}^{(t,l)}$, we can simplify Eq. (9) into

$$\mathbf{H}^{(t,l)} = \mathbf{A}^{(t,l)}\mathbf{X}\mathbf{W}^{(t,l)}, \tag{11}$$

where $\mathbf{A}^{(t,l)}$ denotes the tth lth order adjacency matrix defined in Eq. (9). Moreover, if different lengths of meta-paths are considered, we have

$$\begin{aligned}\mathbf{Z} &= \sum_{l=1}^{L}\sum_{t=1}^{|\boldsymbol{\Psi}^{(l)}|}\beta^{(t,l)}\mathbf{H}^{(t,l)}\\ &= \sum_{l=1}^{L}\sum_{t=1}^{|\boldsymbol{\Psi}^{(l)}|}\beta^{(t,l)}\left(\mathbf{A}^{(t,l)}\mathbf{X}\mathbf{W}^{(t,l)}\right),\end{aligned} \tag{12}$$

where $\beta^{(t,l)}$ is a trainable weight and $\sum_{l=1}^{L}\sum_{t=1}^{|\boldsymbol{\Psi}^{(l)}|}\beta^{(t,l)} = 1$, indicating the importance of the tth l-length meta-path. We adopt the softmax function to project $\beta^{(t,l)}$ onto the feasible space. Considering a shared $\mathbf{W}$ for all trainable weight matrices, Eq. (12) leads to

$$\mathbf{Z} = \mathcal{A}\mathbf{X}\mathbf{W}, \tag{13}$$

where $\mathcal{A}$ is the multi-order adjacency matrix defined as

$$\mathcal{A} = \sum_{l=1}^{L}\sum_{t=1}^{|\boldsymbol{\Psi}^{(l)}|}\beta^{(t,l)}\mathbf{A}^{(t,l)}. \tag{14}$$

In summary, Eqs. (5), (7), (8) and (14) explain the construction of multi-order adjacency matrix (**RP1**), on the basis of which we establish the multi-order GCN with Eq. (13) (**RP2**).

4.2. Semantic information supervision

Owing to the complexity of heterogeneous graphs, some highly similar nodes may be disconnected in the topology space. Because the aggregation of multi-order meta-paths can describe the node relations more comprehensively, it should also be correlated with node homophily information. Thus, apart from serving as the input of GCN, node attributes are utilized to build a semantic adjacency matrix that describes the node homophily to further instruct the learning of multi-order meta-path aggregation (**RP3**). The semantic adjacency matrix can be evaluated by some pair-wise distance measurements that compute the spatial distance based on node feature vectors. In this paper, we evaluate the node similarity via the cosine similarity, i.e., $\text{sim}(x_i, x_j) = \frac{<x_i, x_j>}{\|x_i\|\|x_j\|}$. We select the top-$k$ similar neighbors for each node to generate the semantic adjacency matrix. Formulaically, we construct the semantic adjacency matrix via

$$\mathbf{A}_S = \begin{cases}1, & x_i \in TopK(x_j)\ or\ x_j \in TopK(x_i),\\ 0, & \text{otherwise.}\end{cases} \tag{15}$$

With the precomputed $\mathbf{A}_S$, we require that the multi-order adjacency matrix fusion should involve the critical connections estimated by node homophily. This is because that a high-order neighborhood relation should tend to connect nodes that are remote but similar. Accordingly, the supervision of semantic information can be measured by the loss function

$$\mathcal{L}_{rec}\left(\mathcal{A}, \mathbf{A}_S\right) = -\frac{1}{\kappa}\sum_{i,j}\left[\mathbf{A}_S\right]_{ij} log\left(\mathcal{A}_{ij}\right), \tag{16}$$

where κ is the number of non-zero entries in $\mathbf{A}_S$. It is noted that the semantic node homophily information only indicates the potential connections between similar attributed nodes, and we should not ignore other types of node relations. Consequently, we only consider the supervision from existing semantic connections.

Integrating both label and semantic information supervision, the loss function of AMOGCN for the semi-supervision classification is

$$\mathcal{L}\left(\mathbf{Z}, \mathbf{Y}, \mathcal{A}, \mathbf{A}_S\right) = \mathcal{L}_{ce}\left(\mathbf{Z}, \mathbf{Y}\right) + \gamma\mathcal{L}_{rec}\left(\mathcal{A}, \mathbf{A}_S\right), \tag{17}$$

Algorithm 1: Attributed Multi-Order Graph Convolutional Network (AMOGCN)

Input: Node attributes $\mathbf{X} \in \mathbb{R}^{n \times f}$, first-order adjacency matrices $\{\mathbf{A}_i\}_{i=1}^{L}$, ground truth $\mathbf{Y} \in \mathbb{R}^{n \times c}$, hyperparameters γ and k.

Output: Node embedding $\mathbf{Z}$.

1 Obtain the semantic adjacency matrix $\mathbf{A}_S$ for supervision;
2 Initialize all coefficient matrices $\boldsymbol{\alpha}^{(t,l)}$ randomly;
3 **while** $\mathcal{L}\left(\mathbf{Z}, \mathbf{Y}, \mathcal{A}, \mathbf{A}_S\right)$ *does not converge* **do**
  `/* Adaptive high-order meta-paths learning. */`
4 Obtain different orders of adjacency matrices $\mathbf{A}^{(t,l)}$ that contain various lengths of meta-paths with Eqs. (5), (7) and (8);
  `/* Multi-order meta-path aggregation. */`
5 Compute the multi-order adjacency matrix $\mathcal{A}$ that fuses distinct lengths of meta-paths with Eq. (14);
  `/* Multi-order feature learning with SGC. */`
6 Compute outputs $\mathbf{Z}$ of the multi-order GCN with Eq. (13);
7 Calculate the loss $\mathcal{L}\left(\mathbf{Z}, \mathbf{Y}, \mathcal{A}, \mathbf{A}_S\right)$ with Eq. (17);
8 Update trainable GCN weight $\mathbf{W}$, coefficient matrices $\boldsymbol{\alpha}^{(t,l)}$ and all weights $\beta^{(t,l)}$ with back propagation;
9 **return** *Multi-order node embedding* $\mathbf{Z}$.

where γ is the trade-off hyperparameter and $\mathcal{L}_{ce}$ is the cross-entropy loss function, i.e.,

$$\mathcal{L}_{ce}\left(\mathbf{Z}, \mathbf{Y}\right) = -\sum_{i \in \Omega} \sum_{j=1}^{c} \mathbf{Y}_{ij} \ln \mathbf{Z}_{ij}, \tag{18}$$

where $\mathbf{Y}$ is the matrix containing the label information from the training set Ω.

4.3. Training algorithm

We elaborate on the detailed training procedure of AMOGCN in Algorithm 1. Overall, the forward propagation of the network consists of three steps. (1) Calculate different orders of meta-paths adaptively with trainable coefficients (Line 4). (2) Obtain a multi-order meta-path fusion with learnable weights (Line 5). (3) Conduct SGC with the precomputed multi-order meta-path (Line 6). Notice that we may get some repeated or closed-loop paths if the lengths of meta-paths are long. Thus, the fusion of excessive lengths of meta-paths often yields redundant node relations and massive self-connections, which result in the meaningless high-order adjacency matrix. On the other hand, because the proposed AMOGCN corresponds to the fusion of multi-layer GCNs with various first-order adjacency matrices, the multi-length meta-paths with repeated types of node connections in AMOGCN are equivalent to the homogeneous deep-layer GCN. This often leads to notorious performance owing to the oversmoothing issue when a GCN is too deep. Hence, we only consider the maximum order of adjacency matrices as the number of the first-order meta-paths L to reduce unnecessary computations. Because L is generally small, the computational complexity of the first step is approximately $\mathcal{O}(n^2 + n^3)$. The second step takes about $\mathcal{O}(n^2)$, and the third step takes about $\mathcal{O}(n^2 m + nmc)$, where n is the number of target nodes, m is the dimension of target nodes and c is the class number of target nodes. Consequently, the overall computational complexity is approximately $\mathcal{O}(n^2(n+m))$ if $c \ll \min(m, n)$. Actually, because we adopt SGC to propagate node information over the multi-order adjacency matrix, the time cost is much lower than a multi-layer GCN.

4.4. Differences to existing works

AMOGCN considers the automatic learning of the multi-order adjacency matrix, which is also investigated by some prior studies as we discussed in related works. Our model differs from existing multi-order or multi-layer methods (e.g., SGC (Wu et al., 2019), MHGCN (Yu et al., 2022)) in the following aspects:

(1) ***AMOGCN is a more generalized framework exploring multi-order or multi-length meta-paths.*** For example, MHGCN and SGC can be regarded as special cases of AMOGCN. It is exactly SGC when a high-order adjacency matrix generated from a specific first-order meta-path is adopted. MHGCN is a multi-layer network that performs the weighted sum of outputs from distinct layers, which is equivalent to the one-layer AMOGCN that only considers sequential permutations of the first-order meta-paths.

(2) ***The aggregation of different orders of adjacency matrices is considered, which attempts to cover remote and nearby neighborhood connections at the same time.*** The adaptive fusion of different meta-paths enables the final multi-order adjacency matrix to capture more consistency and complementary information from different types of node relations.

(3) ***We make full use of node attributes in heterogeneous graphs.*** Most prior methods only utilized node attributes as the input of GCNs. In our work, apart from serving as initial embeddings of graph convolutions, node attributes are applied to estimate the semantic node relations containing node homophily, which instruct the learning of the multi-order adjacency matrix aggregation.

5. Experimental analysis

In this section, we evaluate the proposed AMOGCN with comprehensive experiments. The proposed model is implemented with PyTorch[1] and runs on a machine with AMD R9-5900HS CPU, RTX 3080 16G GPU and 32G RAM.

5.1. Experimental setup

Compared Baselines. In this paper, we compare the proposed model with both classical and state-of-the-art baselines, including GCN (Kipf & Welling, 2017), SGC (Wu et al., 2019), HAN (Wang et al., 2019), DGI (Velickovic et al., 2019), DMGI (Park et al., 2020), IGNN (Gu, Chang, Zhu, Sojoudi, & El Ghaoui, 2020), SSDCM (Mitra et al., 2021) and MHGCN (Yu et al., 2022). Table 3 provides an overview of these models. These methods can be categorized into two types of models: homogeneous-network-based frameworks (GCN, SGC and DGI) and heterogeneous-network-based frameworks (HAN, DMGI, IGNN, SSDCM and MHGCN). For homogeneous-network-based models, we evaluate the model with the average weighted meta-paths. Important related works exploring learnable meta-paths or multi-order meta-paths are compared in our experiments (SGC and MHGCN). In particular, GCN, SGC, MHGCN are typical multi-layer or multi-order GNN models, among which SGC, MHGCN and the proposed AMOGCN all require precomputation of the multi-order adjacency matrix before training. However, GCN-based methods may suffer from the severe over-smoothing issue that leads to poor and unacceptable performance. Thus, we generally adopt a 2-layer architecture for these models. In detail, the description and code links of these methods are given below.

- **GCN**[2] (Kipf & Welling, 2017) is a semi-supervised homogeneous graph convolutional network which obtains node embeddings by aggregating message from local neighborhood structures.
- **SGC**[3] (Wu et al., 2019) is a simplified version of GCN framework, which only employs the product of high-order adjacency matrices and attribute matrix, removing non-linear transformation for the semi-supervised classification tasks.

[1] https://github.com/chenzl23/AMOGCN
[2] https://github.com/tkipf/gcn
[3] https://github.com/Tiiiger/SGC

Table 3
Properties of different models (Attr.: Using attributes, Heter.: For heterogeneous graphs, Learnable.: Learnable meta-paths, Multi-order.: Multi-order meta-paths).

Methods	Attr.	Heter.	Learnable.	Multi-order.
GCN (Kipf & Welling, 2017)	✓	×	×	×
SGC (Wu et al., 2019)	✓	×	×	✓
DGI (Velickovic et al., 2019)	✓	×	×	×
HAN (Wang et al., 2019)	✓	✓	×	×
DMGI (Park et al., 2020)	✓	✓	×	×
IGNN (Gu et al., 2020)	✓	✓	×	×
SSDCM (Mitra et al., 2021)	✓	✓	×	×
MHGCN (Yu et al., 2022)	✓	✓	✓	✓
AMOGCN (Ours)	✓	✓	✓	✓

Table 4
Statistics of tested heterogeneous graph datasets.

Datasets	# Nodes	# Attributes	Node types	Manually defined meta-paths	# Classes
ACM	8,916	1,870	Paper (P)/Author (A)/Subject (S)	PAP, PSP	3
DBLP	27,194	334	Paper (P)/Author (A)/Conference (C)/Term (T)	APA, APCPA, APTPA	4
IMDB	12,722	1,232	Movie (M)/Actor (A)/Director (D)/Year (Y)	MAM, MDM, MYM	3
YELP	3,913	82	Business (B)/User (U)/Service (S)/Level (L)	BUB, BLB, BSB	3

- **HAN**[4] (Wang et al., 2019) explores the node-level and semantic-level attention on multiplex networks to learn the importance of nodes and meta-paths, thereby generating node representations in a hierarchical manner.
- **DGI**[5] (Velickovic et al., 2019) is an unsupervised graph learning representation approach which maximizes mutual information between the graph-level summary embeddings and the local patches to capture global graph structures.
- **DMGI**[6] (Park et al., 2020) is an unsupervised attributed multiplex network which jointly integrates the node embeddings from multiple relations to learn high-quality representations through a consensus regularization framework and a universal discriminator for downstream tasks.
- **IGNN**[7] (Gu et al., 2020) is a graph learning framework which employs a fixed-point equilibrium equation and the Perron–Frobenius theory to iterate graph convolutional aggregation until converging for node classification tasks.
- **SSDCM**[8] (Mitra et al., 2021) is a semi-supervised framework for representation learning which aims to maximize the mutual information between local and contextualized global graph summaries and employs the cross-layer links to impose the regularization of the node embeddings.
- **MHGCN**[9] (Yu et al., 2022) automatically learns the useful relation-aware topological structural signals by the multiplex relation aggregation and a multi-layer graph convolution for graph representation learning tasks.

Datasets. Four publicly available datasets are adopted to evaluate the performance of compared methods, i.e., ACM, DBLP, IMDB and YELP. All of these datasets contain heterogeneous graphs with attributed nodes, and some manually defined meta-paths are precomputed from multiple node relations. Detailed descriptions are given as follows.

- **ACM** is a citation network dataset which contains 3,025 nodes divided into three types of nodes i.e., paper, author and subject. All nodes are leveraged to construct citation networks, paper content, and other data integration studies. We employ the meta-path set {PAP, PSP} for experiments.
- **DBLP** is extracted from the DBLP citation network website with each node having 334 attributes. All the nodes are classified into four categories, i.e., author, paper, term and conference. The meta-path set {APA, APCPA, APTPA} are employed to conduct experiments.
- **IMDB** is a movie dataset containing four types of nodes i.e., movie, actor, director and year. Nodes are divided into three classes i.e., action, comedy, drama according to the movie genre. Movie features correspond to elements of a bag-of-words representation of plots. We employ the meta-path set {MAM, MDM, MYM} to perform experiments.
- **YELP** is a subset derived from the merchant review website with four types of nodes, i.e., business, user, service and level. We generate the meta-path set {BUB, BLB, BSB} to conduct experiments.

More statistics of these datasets are summarized in Table 4.

Performance Evaluation. We evaluate the classification performance via the widely used Macro-F1 score and Micro-F1 score. All methods are repeated 5 times and we record the average performance. We randomly split all datasets into different ratios of the training sets (20%/40%/60%), validation sets (10%) and test sets (10%) for classification performance evaluation.

5.2. Experimental results

Classification Results. In this paper, we run AMOGCN with a learning rate fixed as 0.01, and the Adam optimizer is adopted. Table 5 exhibits the classification performance with varying training ratios on different datasets. In most cases, HGNNs are able to gain remarkable performance with a few node labels. The experimental comparisons reveal that AMOGCN achieves competitive classification performance compared with state-of-the-art baselines. In general, methods for homogeneous graphs, especially SGC and DGI, perform poorly on node embedding learning. It is noted that SGC precomputes a high-order adjacency matrix from a weighted sum of existing meta-paths. The undesired classification accuracy indicates that the learnable permutations of first-order meta-paths are necessary when computing the high-order adjacency matrix. Among all heterogeneous models, MHGCN and the proposed AMOGCN behave the best, revealing that adaptive multi-order meta-paths are beneficial to the graph embedding learning. In summary, SGC is a multi-order homogeneous model and MHGCN is a multi-layer heterogeneous model, and the proposed AMOGCN that adopts multi-order adjacency matrices and their permutations to approximate a multi-layer network performs satisfactorily compared with them. We also visualize the classification results of

[4] https://github.com/Jhy1993/HAN
[5] https://github.com/PetarV-/DGI
[6] https://github.com/pcy1302/DMGI
[7] https://github.com/SwiftieH/IGNN
[8] https://github.com/anasuamitra/ssdcm
[9] https://github.com/NSSSJSS/MHGCN

Table 5
Node classification performance with various percentages of training samples.

Datasets	Training	Metrics	GCN	SGC	DGI	HAN	DMGI	IGNN	SSDCM	MHGCN	AMOGCN
ACM	20%	Macro-F1	0.786	0.675	0.224	0.915	0.867	0.809	0.877	0.889	**0.924**
		Micro-F1	0.788	0.702	0.369	0.914	0.868	0.795	0.876	0.891	**0.925**
	40%	Macro-F1	0.751	0.672	0.227	0.910	0.871	0.872	0.881	0.911	**0.943**
		Micro-F1	0.758	0.699	0.377	0.911	0.868	0.871	0.883	0.918	**0.944**
	60%	Macro-F1	0.764	0.689	0.241	0.892	0.912	0.903	0.886	0.937	**0.948**
		Micro-F1	0.755	0.649	0.375	0.891	0.909	0.904	0.888	0.924	**0.947**
DBLP	20%	Macro-F1	0.901	0.873	0.243	0.893	0.657	0.891	0.894	0.909	**0.924**
		Micro-F1	0.916	0.904	0.376	0.904	0.711	0.902	0.899	0.921	**0.943**
	40%	Macro-F1	0.895	0.816	0.241	0.900	0.714	0.890	0.902	0.895	**0.916**
		Micro-F1	0.906	0.850	0.366	0.904	0.773	0.884	0.906	0.904	**0.921**
	60%	Macro-F1	0.902	0.850	0.368	0.907	0.721	0.895	0.909	0.922	**0.929**
		Micro-F1	0.909	0.870	0.386	0.911	0.787	0.904	0.911	**0.936**	0.933
IMDB	20%	Macro-F1	0.243	0.270	0.263	0.498	0.353	0.403	0.494	**0.505**	0.502
		Micro-F1	0.554	0.548	0.552	0.549	0.573	0.500	0.591	0.642	**0.651**
	40%	Macro-F1	0.247	0.285	0.264	0.520	0.382	0.503	0.521	0.523	**0.533**
		Micro-F1	0.544	0.550	0.541	0.540	0.590	0.575	0.592	0.612	**0.628**
	60%	Macro-F1	0.287	0.301	0.271	0.542	0.397	0.516	0.549	0.531	**0.551**
		Micro-F1	0.504	0.563	0.565	0.569	0.610	0.588	0.601	0.615	**0.678**
YELP	20%	Macro-F1	0.520	0.519	0.503	0.483	0.516	0.642	0.527	0.546	**0.655**
		Micro-F1	0.674	0.674	0.683	0.489	0.699	0.712	0.702	0.707	**0.724**
	40%	Macro-F1	0.530	0.535	0.542	0.458	0.534	0.645	0.542	0.553	**0.660**
		Micro-F1	0.713	0.720	0.719	0.552	0.709	0.711	0.707	0.697	**0.724**
	60%	Macro-F1	0.580	0.567	0.543	0.439	0.546	0.671	0.587	0.598	**0.692**
		Micro-F1	0.736	0.782	0.723	0.529	0.721	0.624	0.722	0.739	**0.751**

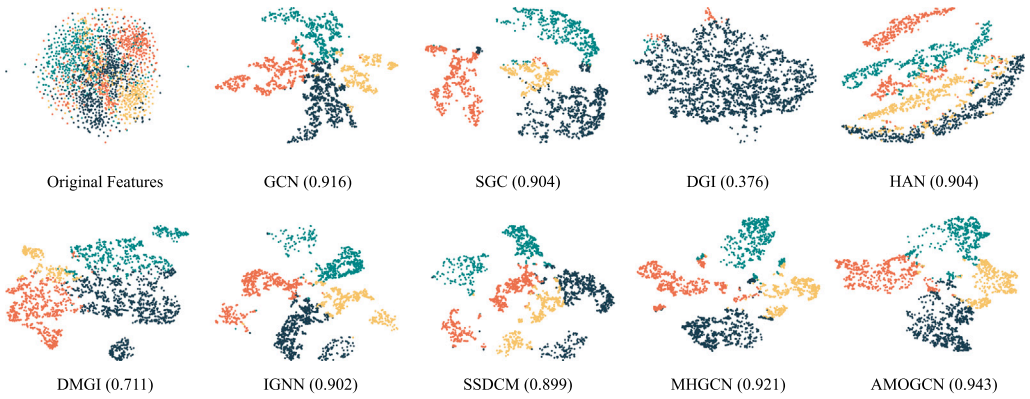

Fig. 4. T-SNE Visualization and Micro-F1 values of compared methods on DBLP dataset with 20% training samples.

various models on DBLP dataset, as shown in Fig. 4. It can be observed that most compared methods generally succeed in learning separable node features, while several compared approaches have some significant mixed nodes belonging to different classes. IGNN, SSDCM, MHGCN and the proposed AMOGCN obtain higher classification accuracy with a stronger ability to get more distinguishable node clusters. In general, these methods gain competitive performance with few training samples (e.g., 20% training data). AMOGCN performs even better and the intra-class correlations in each cluster are closer, which may be attributed to the supervision of semantic information extracted from node homophily. These observations verify the superior node representation learning ability of the proposed AMOGCN.

Parameter Sensitivity. We analyze the impact of hyperparameters k and γ for semantic information supervision in Fig. 5, from which we can obtain the following discoveries. It is obvious that a higher γ promotes the classification performance on all datasets, and the model reaches the best accuracy when γ is large enough. The minimal optimal γ varies on different datasets, but generally it is larger than 0.05. With a fixed γ, a larger k also often leads to higher accuracy, and the model generally cannot obtain more performance benefits when $k > 50$. This is because that an attributed node only has a limited number of semantic neighbors in most cases. These observations indicate that the semantic information extracted from adequate numbers of neighborhood nodes facilitates the model to learn more tailored fusion of multi-order adjacency matrices, which endows the model with a more powerful ability to study node embeddings from suitable high-order neighbors.

Ablation Study. We also conduct the ablation study to verify the effectiveness of semantic supervision, as recorded in Table 6. The experimental results show that the supervision of semantic information works on different datasets, which validate that the aggregation of distinct orders of adjacency matrices can be effectively guided by the node homophily. It is clear that the node relations extracted from node similarities promote the classification accuracy, which is more significant on IMDB and YELP datasets. Therefore, this also validates that some attributed nodes connected by long-length meta-paths are similar in the feature space.

Impact of Multi-order Paths. Furthermore, we examine the adaptive weights of distinct orders of meta-paths in Fig. 6. In this paper, we set the highest degree of meta-paths as the number of manually defined first-order meta-paths. It can be observed that high-order meta-paths play critical roles in the multi-order GCN. On ACM dataset, the first-order adjacency matrices only take a small percentage of the multi-order meta-paths. These experimental results indicate that multi-order

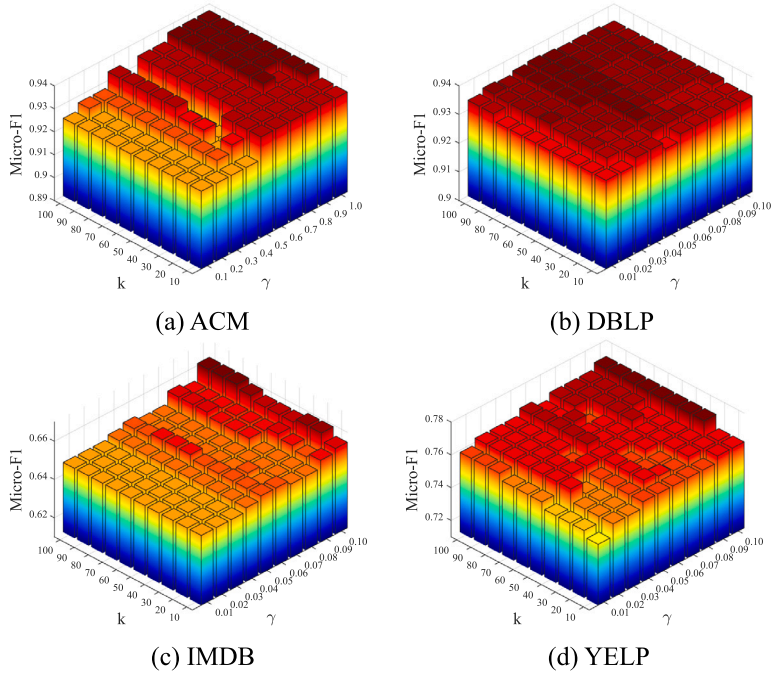

Fig. 5. Parameter sensitivity w.r.t. hyperparameters k and γ in AMOGCN.

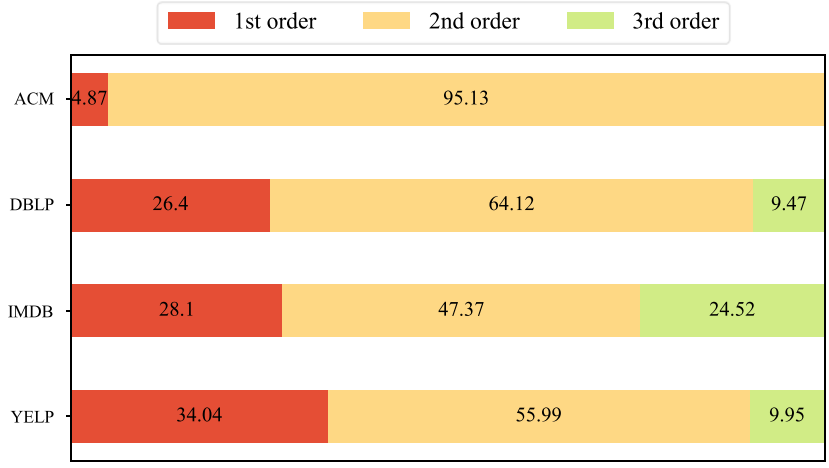

Fig. 6. Weights (%) of different lengths of meta-paths in the multi-order adjacency matrix learned in different datasets.

Table 6
Ablation study of AMOGCN with 20% training samples, evaluated by Micro-F1 values (AMOGCN w/o Sem.: AMOGCN without semantic information supervision).

Methods	ACM	DBLP	IMDB	YELP
AMOGCN w/o Sema.	0.912	0.933	0.644	0.709
AMOGCN	**0.925**	**0.943**	**0.651**	**0.724**

meta-paths are helpful for the model to capture the information from high-order neighbors, which promote high-quality node embedding learning.

Training Time. Fig. 7 shows the runtime of the proposed AMOGCN. In general, the runtime of AMOGCN is acceptable compared with HAN and DMGI, and it is more efficient compared with SSDCM and MHGCN. AMOGCN gains more robust node embeddings with faster speed compared with MHGCN which also adopts learnable high-order meta-paths. A primary reason for the speed boosting is that we adopt a one-layer graph convolution, rather than using a multi-layer GCN to capture the aggregation of varying lengths of meta-paths as MHGCN.

Convergence Curves. Eventually, we record the curves of loss values and accuracy on the training set and the validation set during network learning, as exhibited in Fig. 8. It can be observed that loss values decrease rapidly during training and eventually converge on all datasets. The accuracy of training and validation sets also grows up as the loss dwindles. In this paper, we select the optimal model with the highest validation accuracy to get classification predictions on test sets.

6. Conclusion

In this paper, we proposed an AMOGCN framework to cope with heterogeneous networks, which explored the fusion of different lengths of meta-paths via learning an adaptive multi-order adjacency matrix. For the purpose of exploring multi-length meta-paths, we first designed the formulation of the adaptive high-order adjacency matrix, which corresponded to a long-length meta-path involving various first-order meta-paths. AMOGCN further introduced semantic information that considered node homophily as a new type of node relations, which was regarded as supervision signals to instruct the automatic aggregation

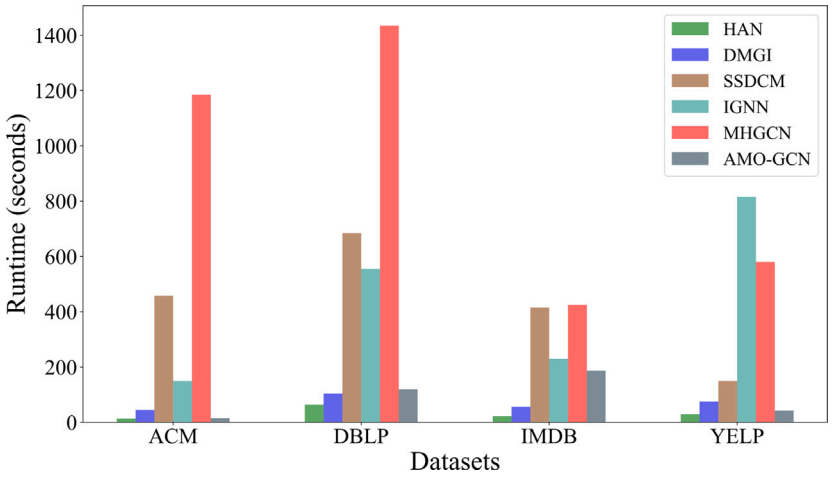

Fig. 7. Running time (seconds) of compared HGNNs with 500 training iterations.

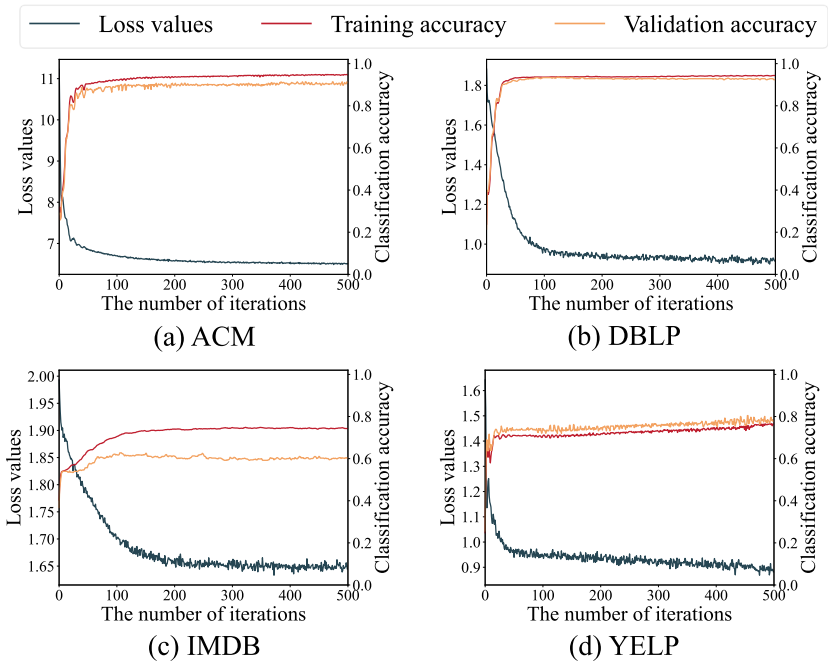

Fig. 8. Convergence and training/validation Macro-F1 curves of AMOGCN.

of multi-order adjacency matrices. Substantial experimental results pointed out that the proposed model succeeded in learning multi-length meta-paths with a multi-order adjacency matrix, and gained promising performance improvement on heterogeneous graph datasets compared with state-of-the-art competitors. In our future work, we will devote ourselves to research on joint HGNNs with learnable meta-paths, such as adaptive meta-path refining with more complex heterogeneous graphs.

CRediT authorship contribution statement

Zhaoliang Chen: Conceptualization, Data curation, Formal analysis, Writing – original draft, Writing – review & editing. **Zhihao Wu:** Conceptualization, Data curation, Writing – review & editing. **Luying Zhong:** Visualization, Writing – review & editing. **Claudia Plant:** Supervision, Writing – review & editing. **Shiping Wang:** Funding acquisition, Writing – review & editing. **Wenzhong Guo:** Funding acquisition, Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China under Grant U21A20472 and 62276065, and the National Key Research and Development Plan of China under Grant 2021YFB3600503.

References

Bojchevski, A., Gasteiger, J., Perozzi, B., Kapoor, A., Blais, M., Rózemberczki, B., et al. (2020). Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2464–2473).

Chen, Z., Fu, L., Yao, J., Guo, W., Plant, C., & Wang, S. (2023). Learnable graph convolutional network and feature fusion for multi-view learning. *Information Fusion*, *95*, 109–119.

Chen, Z., Wu, Z., Lin, Z., Wang, S., Plant, C., & Guo, W. (2023). AGNN: Alternating graph-regularized neural networks to alleviate over-smoothing. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13. http://dx.doi.org/10.1109/TNNLS.2023.3271623.

Chen, Z., Wu, Z., Wang, S., & Guo, W. (2023). Dual low-rank graph autoencoder for semantic and topological networks. In *Proceedings of the 37th AAAI conference on artificial intelligence, vol. 37, no. 4* (pp. 4191–4198).

Dong, Y., Chawla, N. V., & Swami, A. (2017). Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD conference on knowledge discovery and data mining* (pp. 135–144).

Feng, F., He, X., Zhang, H., & Chua, T. (2023). Cross-GCN: Enhancing graph convolutional network with k-order feature interactions. *IEEE Transactions on Knowledge and Data Engineering*, *35*(1), 225–236.

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD conference on knowledge discovery and data mining* (pp. 855–864).

Gu, F., Chang, H., Zhu, W., Sojoudi, S., & El Ghaoui, L. (2020). Implicit graph neural networks. In *Advances in neural information processing systems*: *vol. 33*, (pp. 11984–11995).

Jeh, G., & Widom, J. (2003). Scaling personalized web search. In *Proceedings of the 12th international conference on world wide web* (pp. 271–279).

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th international conference on learning representations*.

Klicpera, J., Bojchevski, A., & Günnemann, S. (2019). Predict then propagate: Graph neural networks meet personalized PageRank. In *Proceedings of the 7th international conference on learning representations*.

Li, Y., Jin, Y., Song, G., Zhu, Z., Shi, C., & Wang, Y. (2021). Graphmse: Efficient meta-path selection in semantically aligned feature space for graph neural networks. In *Proceedings of the 35th AAAI conference on artificial intelligence, vol. 35, no. 5* (pp. 4206–4214).

Li, J., Xu, K., Chen, L., Zheng, Z., & Liu, X. (2021). GraphGallery: A platform for fast benchmarking and easy development of graph neural networks based intelligent software. In *Proceedings of the 43rd IEEE/ACM international conference on software engineering* (pp. 13–16).

Liu, Y., Fan, L., Wang, X., Xiao, Z., Ma, S., Pang, Y., et al. (2023). HGBER: Heterogeneous graph neural network with bidirectional encoding representation. *IEEE Transactions on Neural Networks and Learning Systems*, http://dx.doi.org/10.1109/TNNLS.2022.3232709.

Min, Y., Wenkel, F., & Wolf, G. (2020). Scattering GCN: overcoming oversmoothness in graph convolutional networks. In *Advances in neural information processing systems*: *vol. 33*, (pp. 14498–14508).

Mitra, A., Vijayan, P., Singh, S. R., Goswami, D., Parthasarathy, S., & Ravindran, B. (2021). Semi-supervised deep learning for multiplex networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1234–1244).

Pan, H., Chen, Y., & He, Z. (2023). Multi-granularity graph pooling for video-based person re-identification. *Neural Networks*, *160*, 22–33.

Park, C., Kim, D., Han, J., & Yu, H. (2020). Unsupervised attributed multiplex network embedding. In *Proceedings of the 34th AAAI conference on artificial intelligence* (pp. 5371–5378).

Qian, Y., Zhang, Y., Wen, Q., Ye, Y., & Zhang, C. (2022). Rep2vec: Repository embedding via heterogeneous graph adversarial contrastive learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1390–1400).

Sadikaj, Y., Velaj, Y., Behzadi, S., & Plant, C. (2021). Spectral clustering of attributed multi-relational graphs. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1431–1440).

Shi, C., Hu, B., Zhao, W. X., & Philip, S. Y. (2018). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), 357–370.

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2019). Deep graph infomax. In *Proceedings of the 7th international conference on learning representations*.

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., et al. (2019). Heterogeneous graph attention network. In *Proceedings of the world wide web conference* (pp. 2022–2032).

Wang, C., Song, Y., Li, H., Zhang, M., & Han, J. (2018). Unsupervised meta-path selection for text similarity measure based on heterogeneous information networks. *Data Mining and Knowledge Discovery*, *32*(6), 1735–1767.

Wang, C., Zhou, S., Yu, K., Chen, D., Li, B., Feng, Y., et al. (2022). Collaborative knowledge distillation for heterogeneous information network embedding. In *Proceedings of the ACM web conference* (pp. 1631–1639).

Wang, X., Zhu, M., Bo, D., Cui, P., Shi, C., & Pei, J. (2020). AM-GCN: adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1243–1253).

Wu, Z., Shu, L., Xu, Z., Chang, Y., Chen, C., & Zheng, Z. (2022). Robust tensor graph convolutional networks via T-SVD based graph augmentation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2090–2099).

Wu, F., Souza, A. H., Jr., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. Q. (2019). Simplifying graph convolutional networks. In *Proceedings of the 36th international conference on machine learning, vol. 97* (pp. 6861–6871).

Wu, Z., Zhang, Z., & Fan, J. (2023). Graph convolutional kernel versus graph convolutional networks. *36*, In *Advances in neural information processing systems* (pp. 19650–19672).

Xia, W., Gao, Q., Wang, Q., Gao, X., Ding, C., & Tao, D. (2022). Tensorized bipartite graph learning for multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, http://dx.doi.org/10.1109/TPAMI.2022.3187976.

Xia, W., Wang, Q., Gao, Q., Zhang, X., & Gao, X. (2022). Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, *24*, 3182–3192.

Xia, W., Wang, S., Yang, M., Gao, Q., Han, J., & Gao, X. (2022). Multi-view graph embedding clustering network: Joint self-supervision and block diagonal representation. *Neural Networks*, *145*, 1–9.

Xu, K., Huang, H., Deng, P., & Li, Y. (2021). Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing. *IEEE Transactions on Neural Networks and Learning Systems*, http://dx.doi.org/10.1109/TNNLS.2021.3071369.

Xu, H., Xia, W., Gao, Q., Han, J., & Gao, X. (2021). Graph embedding clustering: Graph attention auto-encoder with cluster-specificity distribution. *Neural Networks*, *142*, 221–230.

Yang, Y., Sun, Y., Ju, F., Wang, S., Gao, J., & Yin, B. (2023). Multi-graph fusion graph convolutional networks with pseudo-label supervision. *Neural Networks*, *158*, 305–317.

Yu, P., Fu, C., Yu, Y., Huang, C., Zhao, Z., & Dong, J. (2022). Multiplex heterogeneous graph convolutional network. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2377–2387).

Yun, S., Jeong, M., Kim, R., Kang, J., & Kim, H. J. (2019). Graph transformer networks. In *Advances in neural information processing systems* (pp. 11960–11970).

Zhang, X., Liu, H., Li, Q., & Wu, X. (2019). Attributed graph clustering via adaptive graph convolution. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 4327–4333).

Zhao, J., Wang, X., Shi, C., Hu, B., Song, G., & Ye, Y. (2021). Heterogeneous graph structure learning for graph neural networks. In *Proceedings of the 35th AAAI conference on artificial intelligence* (pp. 4697–4705).

Zhao, Y., Wang, L., Wang, C., Du, H., Wei, S., Feng, H., et al. (2022). Multi-granularity heterogeneous graph attention networks for extractive document summarization. *Neural Networks*, *155*, 340–347.

Zhu, H., & Koniusz, P. (2021). Simple spectral graph convolution. In *Proceedings of the 9th international conference on learning representations*.