SHNE: Semantics and Homophily Preserving Network Embedding

Ziyang Zhang, Chuan Chen, *Member, IEEE* , Yaomin Chang, Weibo Hu, Xingxing Xing, Yuren Zhou, and Zibin Zheng, *Senior Member, IEEE*

***Abstract*— Graph convolutional networks (GCNs) have achieved great success in many applications and have caught significant attention in both academic and industrial domains. However, repeatedly employing graph convolutional layers would render the node embeddings indistinguishable. For the sake of avoiding oversmoothing, most GCN-based models are restricted in a shallow architecture. Therefore, the expressive power of these models is insufficient since they ignore information beyond local neighborhoods. Furthermore, existing methods either do not consider the semantics from high-order local structures or neglect the node homophily (i.e., node similarity), which severely limits the performance of the model. In this article, we take above problems into consideration and propose a novel Semantics and Homophily preserving Network Embedding (SHNE) model. In particular, SHNE leverages higher order connectivity patterns to capture structural semantics. To exploit node homophily, SHNE utilizes both structural and feature similarity to discover potential correlated neighbors for each node from the whole graph; thus, distant but informative nodes can also contribute to the model. Moreover, with the proposed dual-attention mechanisms, SHNE learns comprehensive embeddings with additional information from various semantic spaces. Furthermore, we also design a semantic regularizer to improve the quality of the combined representation. Extensive experiments demonstrate that SHNE outperforms state-of-the-art methods on benchmark datasets.**

***Index Terms*— Graph convolutional network (GCN), network embedding, node homophily, structural semantics.**

I. Introduction

GRAPH-STRUCTURED data widely exist in the world, such as social networks and biochemical networks. Due to the diversity and complexity of graph data, network embedding has received significant research attention for data mining tasks in recent years [1]–[3]. With the development of theory and practice of network embedding, graph convolutional networks (GCNs) [4], [5] have been demonstrated to achieve better performance than traditional random-walk-based methods [6], [7] and matrix factorization methods [8], [9]. Due to the outperformance of graph convolutional mechanism, GCN-based models have been widely applied in many applications, such as property prediction [10]–[12], graph classification [13]–[15], and recommender systems [16]–[18].

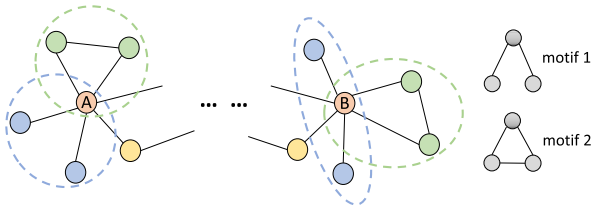


Fig. 1. Node A and node B show significant similarity since their local structure contain common motifs (e.g., local structures in green and blue circles). Furthermore, the two nodes play different roles in different motifs, thus, contain various semantics.

GCNs can be unified into the message-passing neural networks (MPNNs) [19], where nodes propagate their influence along edges and assemble features from neighborhoods to update their representations in each iteration; thus, node features and graph topological structures can be integrated naturally. Although GCNs have achieved significant improvements in various tasks, the message-passing framework limits the expressiveness of GCNs. As the depth of GCNs increases, the performance of the model sharply declines due to the oversmoothing problem [20]–[23], which renders most GCN-based models lacking in the discriminability of node embeddings. As a result, GCN-based models can only have shallow architectures, thus, features from distant but informative nodes cannot be integrated. Recently, some studies [24]–[26] have noticed the weaknesses and proposed various neighborhood expansion strategies to capture high-order information from different perspectives. However, these models limit the expressiveness of node embeddings since they do not fully consider the node homophily or neglect semantics from high-order local structures. For example, as observed in Fig. 1, the local structure of node A is the same as that of node B, where they both contain many common motifs, thus, the two nodes have

Manuscript received December 15, 2020; revised April 20, 2021 and July 4, 2021; accepted September 26, 2021. This work was supported by Guangdong Applied Research and Development Program (2015B010131006), the National Natural Science Foundation of China (62176269, 11801595), Guangdong Basic and Applied Basic Research Foundation (2019A1515011043), and in part by the UX Center, Netease Games. *(Corresponding author: Chuan Chen.)*

Ziyang Zhang, Chuan Chen, Yaomin Chang, Weibo Hu, Yuren Zhou, and Zibin Zheng are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China, and also with the National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510275, China (e-mail: zhangzy233@mail2.sysu.edu.cn; chenchuan@mail.sysu.edu.cn; changym3@mail2.sysu.edu.cn; huwb7@mail2.sysu.edu.cn; zhouyuren@mail.sysu.edu.cn; zhzibin@mail.sysu.edu.cn).

Xingxing Xing is with NetEase, Inc., Guangzhou 510665, China (e-mail: xingxingxing@corp.netease.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3116936.

Digital Object Identifier 10.1109/TNNLS.2021.3116936

significant similarity and are likely to possess akin properties though they are far from each other. Furthermore, both nodes play different roles in various contexts (e.g., motif 1 and motif 2 in Fig. 1). Hence, they contain complicated semantic information.

In order to capture various semantics from local structures and long-range positions in the graph, we utilize both structural similarity and feature similarity to explore node homophily and discover potentially correlated nodes as the nonlocal (NL) neighbors for each node. In terms of structural similarity, the higher order connectivity pattern, motif, can be utilized to explore rich semantic information in complex graph structures. Motifs specify the semantic context of the target node via certain patterns and can accurately distinguish the semantic roles of each node in the receptive field according to different structural connection patterns [27], [28]. Consequently, motifs not only provide a measurement of structural similarity but also reveal rich semantic information in the graph. Therefore, they can be employed to assign certain roles to each node, where nodes with the same role possess common structural semantics. As for feature similarity, the distribution of features reflects the underlying relationship of the data. Two nodes that possess similar features are likely to show akin properties even though they are far apart from each other.

Motivated by the above analysis, we propose a novel Semantics and Homophily preserving Network Embedding (SHNE) model that exploits information beyond local structures. In this framework, we take node homophily into consideration and utilize both feature and structural similarity to discover potentially correlated nodes from the whole graph, thus, informative features from long-range positions can be integrated by the model. In particular, k-neighbor graphs are constructed to connect nodes with their NL neighbors, which are referred to as NL graphs. Concretely, we leverage motifs to assign certain structural roles (SRs) for nodes in the graph and find the NL neighbors for each node base on the node homophily. Then, nodes in the original graph are connected with their NL neighbors in NL graphs. In this way, GCNs can mine rich semantic information from the whole graph thus providing great potential for high-quality node embeddings even if the model is confined to a shallow architecture. In order to fuse various semantics and features from local structures and long-range positions, we propose dual-attention mechanisms to obtain comprehensive node representations that combine embeddings from multiple spaces. Moreover, a simple but effective regularizer is also designed to force embeddings with different semantics to be consistent, reducing noise and improving the quality of combined embeddings. To summarize, this work makes the following contributions.

1) We base our model on the node homophily to discover correlated NL neighbors for each node and propose the notion of SRs to reveal rich semantics. In this way, useful features and semantic information beyond local structures can be effectively explored by the model.
2) We propose dual-attention mechanisms to identify the importance of various semantics and fuse features from local structures and long-range positions automatically, which provides interpretability and enhances the robustness of SHNE.
3) We conduct extensive experiments to evaluate the performance of SHNE, and experimental results demonstrate that SHNE outperforms the state of the arts. Furthermore, we perform a detailed analysis to comprehensively understand the effectiveness of the model.

II. Related Work

A. Graph Convolutional Networks

GCNs inherit key ideas from the design of convolutional neural networks (CNNs) [29] and follow a local feature extraction framework to capture information from structures. GCNs perform propagation guided by graph structures, and nodes collectively aggregate information from their neighborhoods. Intuitively, if nodes can receive more positive messages from others in the graph, the model could learn more informative embeddings. However, recent studies [30] demonstrate that the expressive power of GCN-based models is limited by their designs, and Xu *et al.* [24] point out that the range of neighbors that a node's embedding draws from strongly depends on the graph structure. Consequently, the performance of GCNs is limited by the number of convolutional layers since repeated propagation would make node representations of different classes indistinguishable. DeepGCN [31] applies residual/dense connections and dilated convolutions to GCN architectures for training deep GCNs. DAGNN [32] decouples the transformation and propagation in graph convolution to increase the depth of GCNs. DropEdge [33] randomly removes certain edges from the graph at each training epoch to reduce the convergence speed of oversmoothing and improve performance on deep GCNs. Although these methods alleviate the oversmoothing problem to some extent and enable models to go deep, they are still limited to integrate the features from long-range positions in the graph since the aggregation is essentially local.

B. Neighborhood Expansion Strategies

Due to the inflexibility of aggregators in GCNs, different graph structures result in very different neighborhood sizes. A few studies exploit higher order information to improve GCNs with various neighborhood expansion strategies. For example, Jumping knowledge networks [24] continuously increase the number of graph convolutional layers to enable the target node to receive information from long-range neighbors by leveraging skip connections. However, the accuracy of the model does not be necessarily improved with the depth increase, which limits the ability to capture informative features from distant positions in the graph. Geom-GCN [25] explores long-range dependencies in disassortative graphs by mapping nodes into embedding spaces through various embedding methods. Nevertheless, it uses manually defined relationships and precomputed node embeddings that are not task-specific, which limits the flexibility and robustness of the model. NLAH [26] aggregates NL features by setting a virtual node for the target node. However, the range of NL neighbors selection is no more than five hops; hence, the expressiveness

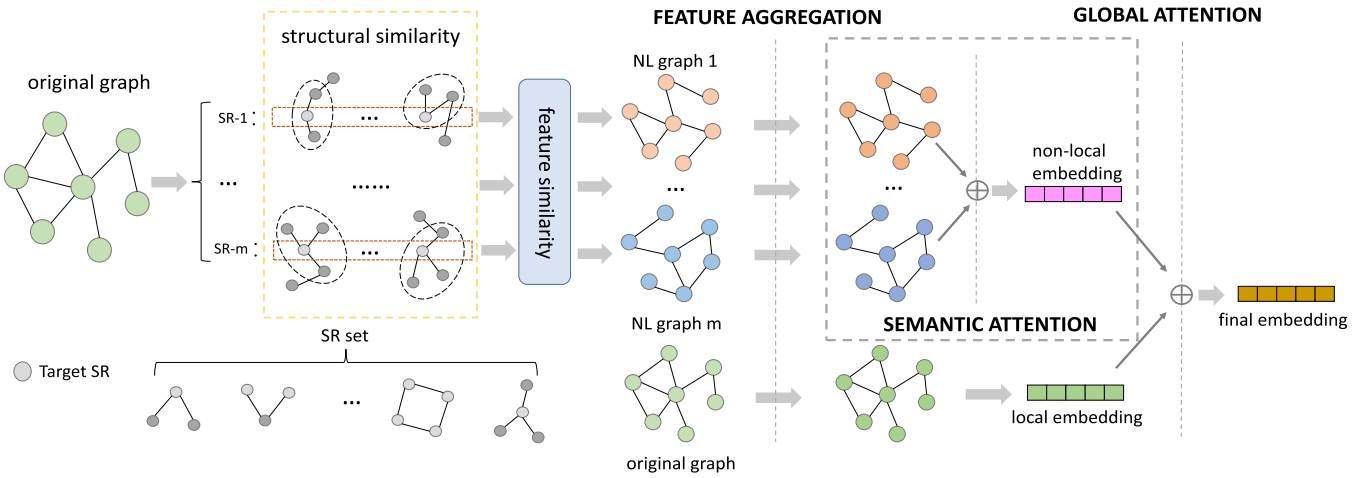


Fig. 2. Framework of the SHNE model. First, feature and structural similarity are utilized to construct NL graphs. Then, SHNE employs GCNs to extract features from NL graphs and the original graph. With semantic attention, embeddings of NL graphs are fused to generate the NL embeddings. Finally, global attention is utilized to combine the NL embedding and local embedding into the final node embedding.

of the model is still restricted to local structures of nodes, thus, lacks the ability to aggregate features from the whole graph. Besides, existing models either do not consider semantics from high-order structures or neglect node homophily, which severely limits the expressive power of the model.

C. Motifs in Graph Learning

Motif, as a high-order structure, plays an important role in many types of networks, such as social networks, biological networks, and neuroscience networks. Motifs are treated as fundamental units of networks and are widely used in multiple data mining tasks. So far, many works have demonstrated that motifs are helpful to explore higher order connection patterns in various graph-learning tasks. Hu *et al.* [34] leverage motifs to discover the cliques in heterogeneous information networks. Gupta *et al.* [35] utilize common network motifs types and identify the topological behaviors of cancer networks and STNs. Wen *et al.* [36] propose motif-based graph convolution to capture hierarchical structures and mine important information of skeleton convey for action recognition. Zhao *et al.* [37] use motifs to capture higher order relations among the nodes for recommendation. Different from the simple paradigms of motif-based feature aggregation of existing methods, SHNE utilizes connection patterns of motifs to measure structural similarity via assigning certain roles to nodes in the graph, thus, semantics under different structural contexts can also be explored. In addition, with the proposed dual-attention mechanisms, SHNE combines multichannel information from both local space and NL spaces, which improves the expressiveness of the model.

III. Methods

Given an unweight graph $G = (V, A, E, X)$ with the vertices set $V = \{1, \ldots, n\}$, the edges set $E \subseteq V \times V$, where $A \in R^{n \times n}$ denotes the adjacency matrix with n nodes, $X \in R^{n \times d}$ denotes the feature matrix, and d is the number of features. In this article, we focus on the task of semi-supervised node classification.

The overall framework of SHNE is shown in Fig. 2. First, to fully exploit complex semantics existing in graph structures, we propose to assign certain SRs for each node in the graph based on the connection patterns in motifs. Hence, nodes with the same SR contain the same type of semantic information, which contributes to the follow-up graph learning. Second, as shown in *structural similarity* in Fig. 2, we group the nodes according to their SRs and then utilize feature similarity to construct the k-neighbor graphs. In this way, nodes in the original graph are connected with their NL neighbors in various semantic-specific NL graphs. After that, GCNs can be employed to extract features from each graph and generate node embeddings. With the proposed dual-attention mechanisms, SHNE combines embeddings from NL graphs and the original graph so that multichannel information is fused to learn comprehensive node embeddings.

A. Nonlocal Graph Construction

The shallow architecture limits GCNs to aggregate features from distant but informative nodes, which usually leads to suboptimal performance of graph learning. In order to discover correlated NL neighbors for each node, both structural similarity and feature similarity are utilized to perform neighborhood expansion.

1) Structural Similarity: To explicitly explore graph structures and mine semantic information in graphs, we employ motifs to capture the structural similarity. As shown in Fig. 3, there are two three-node motifs and four-node motifs, respectively. Motifs represent different high-order structures in the graph, and nodes also play different roles in motifs, which are defined as SRs.

Definition 1 (Structural Role): The SRs of context nodes in motifs are determined by automorphic equivalence, i.e., two nodes a and b are automorphically equivalent if we exchange their positions does not change the relationships among all context nodes in motifs, i.e., node a has the same SR as node b.

In Fig. 4, the nodes are assigned with six different roles; generally speaking, they can be divided into central roles

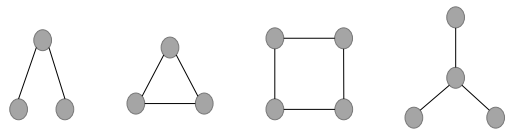


Fig. 3. Sampled three- and four-node motifs.

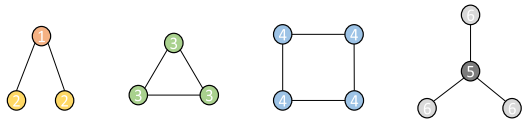


Fig. 4. Toy example of setting SRs for nodes. SR-1: central role; SR-2: marginal role; SR-3: densely connected equivalent role; SR-4: sparsely connected equivalent role; SR-5: densely connected central role; and SR-6: marginal role connected with the densely connected central role.

(i.e., SR-1 and SR-5), marginal roles (i.e., SR-2 and SR-6), and equivalent roles (i.e., SR-3 and SR-4). These six roles represent cardinal positions of nodes in the graph, and more complex high-order SRs can be comprised of these six SRs combined in different ways. For simplicity, here, we do not introduce more other SRs, while the model can be seamlessly extended to handle more complex motifs. Due to the diversity of local structures, a node may have one or more SRs. Given two nodes with the same SR, they satisfy the structural similarity and share the same structural semantics. Besides, to quickly find motifs in the graph, we use an efficient motif matching algorithm from the website.[1]

2) Feature Similarity: Features of nodes play important roles in network embedding because they reflect the underlying relationships of nodes. Based on the homophily setting [38], two nodes are likely to show akin properties as long as they possess similar features. There are many algorithms to measure feature similarity, such as heat kernel similarity [39], second-order proximity [40], and cosine similarity [41]. Here, we uniformly choose the widely used cosine similarity to calculate the similarity matrix. Given the data $X \in R^{n\times d}$, where row X_i represents features of the ith node, n is the number of nodes, and d is the dimension of features. The formula is written as follows:

$$C_{ij} = \frac{X_i \cdot X_j}{\|X_i\|\|X_j\|}. \tag{1}$$

Before calculating the similarity matrix, we should notice that the real networks usually include lots of nodes and features with high dimensions. Hence, it is exhausted to obtain the similarity matrix. What is more, some optimization algorithms (e.g., KD tree [42] and LSH [43]) can be integrated to reduce the computational burden. The above process can be finished in the data preprocessing.

3) Graph Construction: For scalability and flexibility, it is unreasonable to regard all the similar nodes of each node as its NL neighbors. Therefore, we propose to select the top-K representative and similar nodes with the same structural role as corresponding nodes and connected them to construct the semantic-specific NL graphs.

[1]http://www.yfang.site/data-and-tools

Definition 2 (Nonlocal Graph): The NL graph is defined as $G_{r_k} = (V_{r_k}, E_{r_k})$, where $V_{r_k} \subseteq V$ and E_{r_k} denotes the edges among the nodes in V_{r_k}. Specifically, the adjacency matrix of the NL graph under the semantic r_k can be denoted as A_{r_k}

$$A^{ij}_{r_k} = I(r_k \in \varphi(i, j)) \tag{2}$$

where $I(\cdot)$ is the indicator function and $\varphi(i, j)$ is a role mapping function, which returns the projection set of common semantic roles of node i and node j.

Fig. 5 gives a toy example of the NL graph construction.

B. Semantic Information Extraction

SRs correspond to specific semantics in graph structures. Given the SR set $\{r_1, r_2, \ldots, r_p\}$, we can obtain p groups of semantic NL graphs after neighborhood expansion, denoted as $\{A_{r_1}, A_{r_2}, \ldots, A_{r_p}\}$. Due to the success of graph convolutional mechanism in graph learning, we employ GCN [4] to extract features in the graph. Given graph (A_{r_k}, X), the lth layer embeddings are calculated as follows:

$$E^{(l)}_{r_k} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}_{r_k}\widetilde{A}_{r_k}\widetilde{D}^{-\frac{1}{2}}_{r_k}E^{(l-1)}_{r_k}W^{(l)}_{r_k}\right) \tag{3}$$

where σ denotes activation function, e.g., Relu [44], $\widetilde{A}_{r_k} = A_{r_k} + I_{r_k}$, $\widetilde{D}_{r_k}$ is the diagonal degree matrix of $\widetilde{A}_{r_k}$, $W^{(l)}_{r_k}$ is the learnable matrix of the lth layer, $E^{(l)}_{r_k}$ is the lth layer embedding, and $E^{(0)}_{r_k}$ is initialized as X.

The semantic-specific embedding E_{r_k} encodes the NL features captured in semantic space r_k. In addition to extracting features from various NL graphs, the information in the original graph, which reflects the local properties of nodes, is also essential. Similarly, GCN is applied to the original graph to obtain the local embeddings as follows:

$$E^{(l)}_{o} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}_{o}\widetilde{A}_{o}\widetilde{D}^{-\frac{1}{2}}_{o}E^{(l-1)}_{o}W^{(l)}_{o}\right) \tag{4}$$

where $\widetilde{A}_o = A_o + I_o$, A_o denotes the adjacent matrix of the original graph, $W^{(l)}_o$ is the learnable matrix of the lth layer, $E^{(l)}_o$ is the lth layer embedding, and $E^{(0)}_o$ is initialized as X.

C. Dual-Attention Mechanisms

Generally, nodes in the graph contain rich and complex semantic information and the embeddings from one specific space can only reflect nodes from one aspect. To learn comprehensive embeddings, we propose a semantic-level attention mechanism to fuse all the semantics automatically.

Concretely, given node i and its semantic embeddings $\{E^i_{r_1}, \ldots, E^i_{r_p}\}$, where $E^i_{r_k} \in R^{1\times d}$, we transform the embedding through a nonlinear transformation, and the attention value is calculated by multiplying a learnable mapping vector $w \in R^{1\times d'}$

$$E^i_k = \sigma\left(W \cdot \left(E^i_{r_k}\right)^T + b\right) \tag{5}$$

$$a^i_k = w \cdot E^i_k \tag{6}$$

where $W \in R^{d'\times d}$ and $b \in R^{d'\times 1}$ denote the weight matrix and the bias vector, respectively. a^i_k is the nonnormalized attention value under the semantic space k. Given p NL semantic embeddings of node i, we can obtain corresponding attention

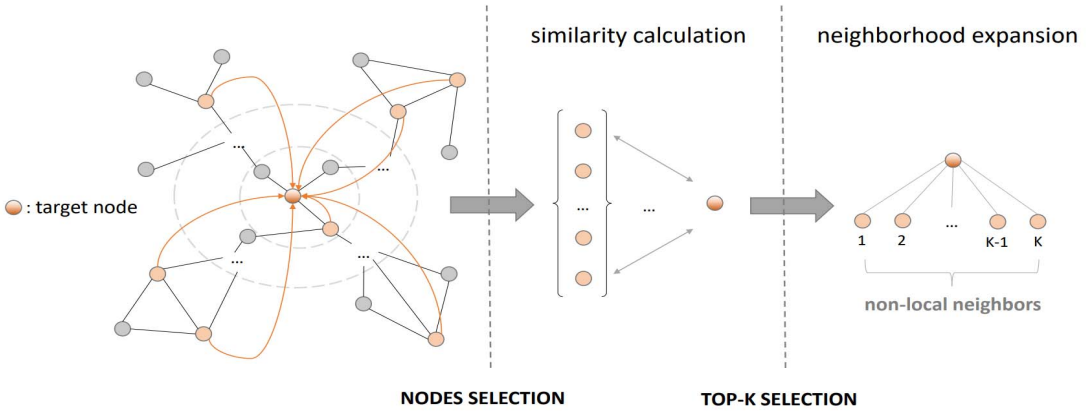


Fig. 5. Toy example on how to construct NL graphs (take SR-5 as an example). First, we choose the nodes with the same SR as the target node in the graph and then use cosine similarity to select top-K similar nodes as the neighbors of the target node in NL graphs.

values $\{a_1^i, \ldots, a_p^i\}$; then, the weight of the kth semantic embedding of node i can be obtained by normalizing the above attention values

$$a'^i_k = \frac{\exp(a_k^i)}{\sum_{j=1}^{p} \exp\left(a_j^i\right)}. \tag{7}$$

The weight a'^i_k indicates the importance of the semantics corresponding to SR-k. With the learned weights, all the semantic embeddings are combined to obtain the overall NL embedding

$$E_{NL}^i = \sum_{j=1}^{p} a'^i_j E_{r_j}^i. \tag{8}$$

Generally, information from NL and local structures contributes differently to the ultimate embedding. Accordingly, we propose a global-level attention mechanism to balance the importance of E_o and E_{NL}. Similarly, the formula is written as follows:

$$E_*^{i'} = \sigma\left(W' \cdot \left(E_*^i\right)^T + b'\right) \tag{9}$$

where $E_*^i \in \{E_o^i, E_{NL}^i\}$, $W' \in R^{d' \times d}$, and $b' \in R^{d' \times 1}$ denotes the weight matrix and the bias vector, respectively, and $E_*^{i'} \in R^{d' \times 1}$ is the transformed embedding. Then, attention values are calculated as

$$a_*^i = q \cdot E_*^{i'} \tag{10}$$

where $q \in R^{1 \times d'}$ is a learnable mapping vector. Then, attention values are normalized, and the local embedding and the NL embedding are combined as follows:

$$E_{\text{final}}^i = \frac{\exp(a_o^i)}{\exp(a_o^i) + \exp(a_{NL}^i)} \cdot E_o^i + \frac{\exp(a_{NL}^i)}{\exp(a_o^i) + \exp(a_{NL}^i)} \cdot E_{NL}^i \tag{11}$$

where E_{final}^i is the final embedding of node i, which fuses semantics from various spaces and integrates useful features from the whole graph.

D. Model Training

For the task of semi-supervised node classification, the cross entropy is minimized over all labeled nodes. The predictions are obtained by a classifier with the final embedding $E_{\text{final}} \in R^{n \times d}$

$$Y = \text{softmax}(E_{\text{final}} * W_f + b_f) \tag{12}$$

where $W_f \in R^{d \times C}$ and $b_f \in R^{1 \times C}$ denote the learnable matrix and the bias vector, respectively, and $Y[y_{ic}^*] \in R^{n \times C}$ is the probability of node i belonging to class c. As mentioned before, the nodes are assigned with certain SRs, and then, the corresponding NL graphs are constructed. However, the construction process would inevitably introduce noise to the NL graph. In order to keep the semantics from NL positions consistent with the local structural information, we design a semantic regularizer

$$L_c = \sum_{j=1}^{p} \theta \cdot \|E_{r_j} - E_o\|_2^2. \tag{13}$$

Finally, the overall objective of the model is composed of the supervised loss and the semantic regularization, which is written as follows:

$$L = -\sum_{l \in N_L} y_l \cdot \log\left(y_l^*\right) + L_c \tag{14}$$

where y_l and y_l^* represent labels and embeddings of labeled nodes, respectively, and N_L is the indices set of labeled nodes. The overall procedure of SHNE is shown in Algorithm 1.

E. Complexity Analysis

In this work, the construction of NL graphs can be finished in the data preprocessing. The proposed SHNE utilizes GCN with two layers. Supposed that $W^{(0)} \in R^{h_1 \times h_2}$ is the weight matrix of the first layer and $W^{(1)} \in R^{h_2 \times h_3}$ is the weight matrix of the second layer, the computational complexity of GCN is $O(|E|h_1h_2h_3)$ [4], which is linear to the number of edges in the graph. With k graphs involved in the SHNE, the complexity of dual-attention mechanisms is $O(k|V|d'd)$. The overall time complexity of the proposed

TABLE I
SUMMARY OF DATASET STATISTICS

Dataset	Nodes	Edges	Avg. Degree	Classes	Features	Training set	Testing set
ACM	3025	13128	8.68	3	1870	180	1000
UAI	3067	28311	18.46	19	4930	1140	1000
Citeseer	3327	4732	2.84	6	3703	360	1000
BlogCatalog	5196	171743	66.11	6	8189	360	1000
Flickr	7575	239738	63.29	9	12047	540	1000
Pubmed	19717	44338	4.49	3	500	60	1000
CoraFull	19793	65311	6.59	70	8710	4200	1000

Algorithm 1 Algorithm of SHNE

Input: Graph $g = (V, E, A, X)$,
motif set $\{M_1, \ldots, M_m\}$,
structural role set $\{r_1, \ldots, r_p\}$,
hyperparameters K and θ;
Output: Final embedding E_{final},
semantic attention weights $\{a'_1, \ldots, a'_p\}$,
global attention weights $\{a_o, a_{NL}\}$;

Use motif matching algorithm to find motifs in the graph;
Set certain structural roles for nodes by motifs;
Select top-K correlated nodes of each node according to structural and feature similarity by (1) and connect them in non-local graphs $\{A_{r_1}, A_{r_2}, \ldots, A_{r_p}\}$;
for $r \in \{r_1, \ldots, r_p\}$ **do**
    Learn semantic-specific NL embedding via Eq. (3);
end for
Learn local embedding via Eq. (4);
Calculate semantic attention weights a'_k of different SRs via Eq. (7);
Calculate final NL embedding with the weights via Eq. (8);
Calculate global attention weights a_o and a_{NL}, then obtain the final embedding E_{final} via Eq. (11);
Update parameters by optimizing L via Eq. (14);
Return E_{final}, $\{a'_1, \ldots, a'_p\}$, $\{a_o, a_{NL}\}$;

model is $O(k(|E|h_1h_2h_3 + |V|d'd))$. Due to the independence among the original graph and NL graphs, the GCN module and dual-attention mechanisms can be easily parallelized; thus, the complexity of SHNE is linearly related to the number of edges and nodes.

IV. EXPERIMENTS

In this section, we evaluate the proposed SHNE in seven datasets and answer four questions.

(Q1) How does SHNE perform compared to start-of-the-art models?
(Q2) How is SHNE impacted by the neighborhood expansion strategy?
(Q3) How does SHNE benefit from the dual-attention mechanisms?
(Q4) How does SHNE benefit from the semantic regularization?

A. Datasets

We conduct experiments on seven datasets, including Citeseer [4], ACM [45], Pubmed [46], CoraFull [3], BlogCatalog [47], UAI [3], and Flickr [47]. We follow existing works [3], [4] that organize these datasets into homogeneous graphs, and their description is shown in Table I.

1) Citeseer is a paper citation network where edges are citation links and nodes are publications. The features of each node correspond to a bag-of-words representation of a publication. Labels of nodes specify the research field of this article.
2) ACM is extracted from the ACM dataset where nodes are papers and edges represent that the connected nodes, i.e., papers, share the same author. The features of the nodes are the bag-of-words representations of paper keywords, and all labels are divided into three areas.
3) Pubmed is a citation network composed of the biomedical literature, which has 19 717 nodes and 44 338 edges. The dataset contains bag-of-words feature vectors for each document. The label of the node is the type of diabetes discussed in this article.
4) UAI is a dataset composed of 3067 nodes and 28 311 edges, which has been tested for node classification in [3].
5) BlogCatalog is a social network, which is organized by bloggers and their social relationships. Node attributes are generated by the keywords of user profiles, and the node labels specify the topic categories of users.
6) Flickr is a social network where nodes represent users and edges represent their relationships. The attributes of users are constructed by lists of tags of interest, and labels indicate the interest groups of users.
7) CoraFull is a citation network constructed from the Cora dataset. The nodes and edges represent papers and citation relation between papers, respectively. Labels of nodes are set by the topics of this article.

B. Baselines

We compare SHNE with the following state-of-the-art methods. The baselines are divided into four categories: random-walk-based models, including DeepWalk [6]; first/second-order proximity-based model LINE [40]; vanilla GNN-based models GCN [4] and GAT [46]; and high-order

TABLE II
QUANTITATIVE RESULTS (%) ON THE NODE CLASSIFICATION TASK. SHNE OUTPERFORMS MOST BASELINE MODELS

Dataset	Metrics	DeepWalk	LINE	GCN	GAT	JK-net	MixHop	FAGCN	PTA	SHNE
Citeseer	Micro-F1	50.2	32.0	74.5	74.7	74.9	70.8	75.6	76.3	**77.2**
	Macro-F1	48.9	31.6	70.7	71.7	70.9	68.2	73.4	72.8	**73.7**
Pubmed	Micro-F1	42.3	66.1	79.0	79.0	74.0	75.9	79.5	79.3	**79.7**
	Macro-F1	37.1	62.9	78.6	78.5	73.8	75.4	**79.1**	78.9	78.8
CoraFull	Micro-F1	55.7	45.1	62.2	64.3	61.8	61.4	65.7	65.2	**66.4**
	Macro-F1	54.9	43.0	58.3	58.8	58.4	57.7	61.2	60.6	**61.7**
ACM	Micro-F1	66.6	48.7	90.5	89.1	90.6	85.7	91.1	90.9	**91.6**
	Macro-F1	66.5	48.7	90.5	89.0	90.6	85.8	91.0	90.8	**91.5**
UAI	Micro-F1	59.1	61.7	67.8	69.2	73.7	72.4	72.9	73.2	**76.2**
	Macro-F1	42.4	46.5	57.3	49.8	61.7	58.5	54.9	60.6	**63.7**
Flickr	Micro-F1	40.8	36.3	50.8	42.8	73.9	71.7	75.0	71.7	**78.8**
	Macro-F1	39.7	35.7	49.9	41.3	74.4	72.2	75.2	71.8	**78.7**
BlogCatalog	Micro-F1	60.4	51.0	73.4	70.8	89.4	87.4	90.8	86.6	**91.5**
	Macro-F1	59.9	50.5	72.3	70.0	89.2	87.2	90.3	86.1	**91.1**

information-based models: JK-net [24], MixHop [48], FAGCN [49], PTA [50], and Geom-GCN [25].

1) DeepWalk is a random-walk-based model that employs the SkipGram algorithm to calculate node embeddings.
2) LINE utilizes the first- and second-order proximities to preserve structural properties.
3) GCN obtains node representations by continuously aggregating features from neighboring nodes.
4) GAT is a spatial graph neural network that uses an attention mechanism to guide neighborhood aggregation.
5) JK-net uses jump connections to increase the number of graph convolutional layers, thus, expand the receptive field of the nodes in the graph.
6) MixHop mixes the features of high-order neighbors in graph convolutional layers to generate node embeddings.
7) FAGCN improves the expressive power of GCNs by adaptively aggregating low- and high-frequency signals in the graph.
8) PAT utilizes improved decoupled graph convolution networks to enable GNNs to be more effective and robust.
9) Geom-GCN combines relation- and neighbor-based aggregation based on the node proximity defined in the embedding space.

C. Parameters Setting

We implement our method in PyTorch [51]. For all compared methods, we report the results by re-running the released code with suggested hyperparameters. For the proposed SHNE, we use Adam [52] to optimize the model and randomly initialize parameters. Besides, we set the learning rate to 0.0005, dropout rate to 0.5, depths of GCNs to 2, and the size of the transformed embedding d' to 16. We conduct heuristic search by exploring weight decay $\in$ {5e-3, 1e-4, 5e-4, 1e-5}, the hidden size in the first layer of GCNs $hidden1 \in \{512, 768, 1024\}$, and the hidden size in the second layer $hidden2 \in \{128, 256\}$. The coefficients θ of semantic regularization are searched ranging from 0.001 to 10. For DeepWalk, the window size is set to 5, the number of walks per node is set to 80, and the walk length is set to 10. For all models, we run five times with the same data partition; then, Macro-F1 and Micro-F1 (Accuracy) are used to evaluate the performance of models. All the experiments are conducted on a machine with an NVIDIA GeForce RTX 2080 (11-GB memory), ten-core Intel Core i9-9900X CPU (3.50 GHz), and 128 GB of RAM.

D. Performance on Classification (Q1)

Table II shows the result of node classification. From the table, we can observe that GCN-based methods achieve better performance than DeepWalk and LINE. The reason behind the improvement is that DeepWalk and LINE mainly utilize graph structures to generate node embeddings, while GCN-based methods consider both structural and feature information of nodes. Moreover, it is clear that SHNE significantly outperforms all the compared methods. As for the rest baselines, they enhance the expressive power of models by mining high-order graph information. However, they are restricted in local structures, thus, cannot capture useful features from

TABLE III
NODE CLASSIFICATION ACCURACIES WITH SHNE AND GEOM-GCN (%). THE PROPOSED SHNE ACHIEVES SIGNIFICANT IMPROVEMENT ESPECIALLY IN DISASSORTATIVE GRAPH DATASETS: TEXAS AND WISCONSIN

Model	Citeseer	Cora	Texas	Wisconsin
Geom-GCN	77.9	85.2	67.6	64.1
SHNE	**78.7**	**88.7**	**86.5**	**88.2**

distant but informative nodes. By contrast, SHNE can aggregate features from the whole graph, which enables the model to achieve great improvement. The results demonstrate that it is quite important to capture features beyond local structures.

For further study, we compare the proposed model with Geom-GCN, which also aims to capture long-range dependencies as SHNE. For a fair comparison, SHNE is employed in the same datasets as Geom-GCN, which includes both assortative graph datasets (Citeseer, Cora) and disassortative graph datasets (Texas and Wisconsin) and follows the same experimental settings (e.g., training/testing set partition and set of node features). What is more, Geom-GCN has three variants, and the best results of the variants are recorded in Table III. From the table, we see that the proposed model performs better than Geom-GCN. This is because SHNE not only considers node homophily but also fuses semantics in the graph, thus, leads to more informative node embeddings. Besides, it is worthy of mentioning that SHNE achieves significant improvement especially in disassortative graph datasets (Texas and Wisconsin) where the homophily hypothesis [53] is not well satisfied, which fully verifies SHNE's ability of generalization.

E. Model Analysis

1) Impact of Neighborhood Expand Strategy (Q2): In this section, we discuss the impact of neighborhood expansion strategy on the model from three aspects: the selection of K, the effectiveness of SRs, and the selection of motifs.

a) Selection of K: The number of NL neighbors influences the quality of the constructed NL graph. The experiments are conducted on six datasets to explore how this hyperparameter affects the performance of the model.

Fig. 6 reports the accuracy of the model with various settings of K. From the figure, we can see that the performance of the model increases first and begins to drop. It is probably because, when the graph becomes sparse, the captured NL information is limited. Also, a large K may bring noise, thus, hinder the performance of the model. Although $K = 7$ or 9 also shows the superiority of SHNE over baselines, it spends more time to construct NL graphs. In terms of accuracy and efficiency, $K = 5$ is the most suitable choice.

b) Effectiveness of structural roles: The SRs defined by motifs furnish a way to dig out rich semantics and measure the structural similarity of nodes. To investigate whether SHNE benefits from SRs, we only use feature similarity to find

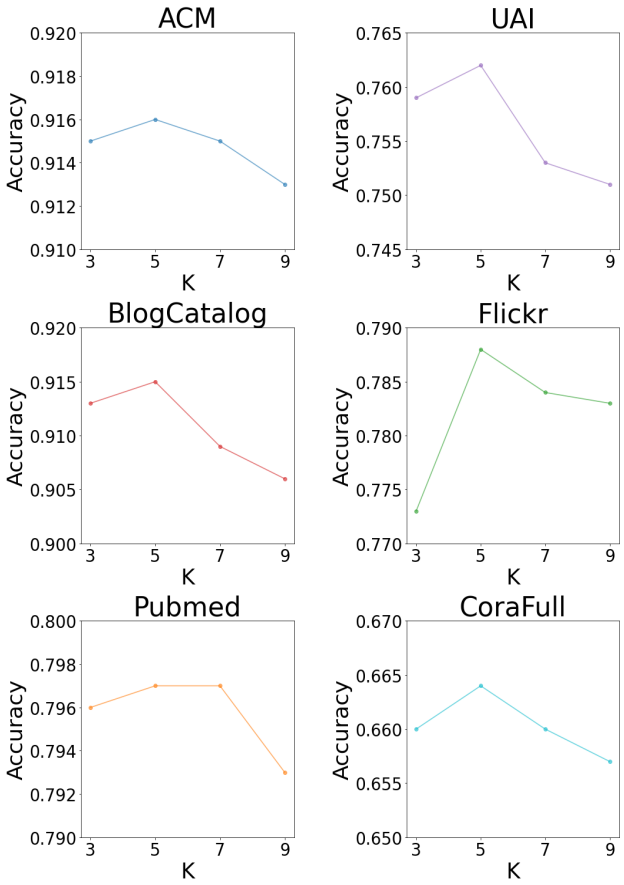


Fig. 6. Parameter sensitivity of SHNE w.r.t. K.

NL neighbors for each node and construct k-nearest neighbor graph calculated from the feature similarity matrix. As can be seen in Table IV, the proposed model outperforms in all the datasets except Flickr. The results indicate that Flickr strongly depends on the feature graph rather than semantic graphs. Nevertheless, the results still demonstrate that the multifaceted semantic information captured by SHNE does benefit the model to possess great expressive power.

c) Selection of motifs: To investigate how motifs affect the performance of SHNE, we conduct experiments with two variants, i.e., SHNE with three-node motifs and SHNE with four-node motifs are referred to as SHNE-m&3 and SHNE-m&4, respectively. From Table V, we can find that the experimental results of SHNE-m&3 and SHNE-m&4 have little difference between each other, while they are both worse than SHNE that contains all six SRs. The reason behind the results is that SHNE explores more structural positions in the graph, thus, fuses more semantic information into the embedding. As mentioned before, the six SRs represent cardinal positions of nodes in the graph, and more complex high-order SRs can be comprised of these six SRs combined in different ways; thus, the proposed model can be seamlessly extended to handle more complex motifs, and we do not conduct extra experiments with more other SRs. Although these variants may not be as accurate as SHNE, it is an optional way to use only three- or four-node motifs for efficiency.

TABLE IV
ACCURACY COMPARISONS BETWEEN THE SHNE AND SHNE WITHOUT SRS (%)

Model	Citeseer	Pubmed	UAI	BlogCatalog	Flickr	ACM	CoraFull
SHNE w/o SRs	75.7	71.8	74.8	87.5	**81.7**	91.2	65.6
SHNE	**77.2**	**79.7**	**76.2**	**91.5**	78.8	**91.6**	**66.4**

TABLE V
ACCURACY COMPARISONS BETWEEN THE SHNE AND SHNE WITH ONLY THREE- OR FOUR-NODE MOTIFS (%)

Model	Citeseer	Pubmed	UAI	BlogCatalog	Flickr	ACM	CoraFull
SHNE-m&3	77	79.5	74.3	91.0	78	91.3	65.6
SHNE-m&4	76.9	79.4	75.6	90.0	78.4	91.2	65.5
SHNE	**77.2**	**79.7**	**76.2**	**91.5**	**78.8**	**91.6**	**66.4**

TABLE VI
ACCURACY COMPARISONS BETWEEN THE SHNE AND SHNE WITHOUT SEMANTIC OR GLOBAL ATTENTION (%)

Model	Citeseer	Pubmed	UAI	BlogCatalog	Flickr	ACM	CoraFull
SHNE w/o semantic attention	74.4	79.4	75.9	91.2	76.4	91.2	64.5
SHNE w/o global attention	73.9	79.2	74.9	90.4	78.6	91.4	63.5
SHNE	**77.2**	**79.7**	**76.2**	**91.5**	**78.8**	**91.6**	**66.4**

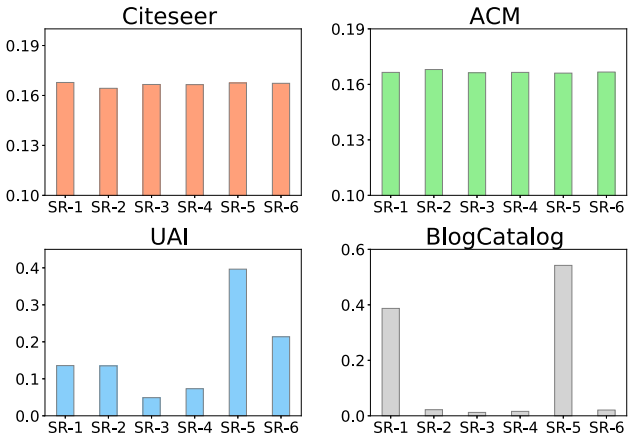


Fig. 7. Attention weight distribution of SRs in four datasets.

2) Impact of Dual-Attention Mechanisms (Q3): To better understand the impact of dual-attention mechanisms, we conduct experiments on the proposed semantic- and global-level attention mechanisms, respectively, and report experimental results in Table VI. From the table, we can observe that the proposed model has a better performance than the model without attention (i.e., assigns the same weight to each embedding), which verifies the necessity of attention mechanisms.

Furthermore, attention weights reflect the importance of different SRs. Nevertheless, it could be not all the datasets are sensitive to motifs. In order to explore the importance of SRs, we take four datasets as examples and draw the distribution of attention weights of six SRs in Fig. 7. It is worth noting that, because we only focus on the importance of six SRs, but the semantic regularizer would reduce the difference of node embeddings under different SRs, the models are learned without semantic regularization. From the figure, we can see that the attention weights in Citeseer and ACM have less difference, while they are different obviously in BlogCatalog and UAI. In terms of BlogCatalog, the attention weights of SR-1 and SR-5 are significant, which demonstrates that central roles are the most crucially structural patterns. As for UAI, densely connected central role and marginal role (SR-5 and -6) are important, while equivalent roles (SR-3 and -4) have fewer attention weights than others, indicating that cyclic structures do not contribute valuable semantics as other connection patterns to the model. Though it is hard to find influential SRs in Citeseer and ACM, the proposed model still achieves significant improvement than GCN, which verifies the effectiveness of SHNE.

3) Impact of Semantic Regularization L_c (Q4): To further demonstrate the efficacy of the semantic regularizer, we conduct experiments on SHNE and SHNE without regularizer. From Table VII, we can observe that SHNE achieves maximum relative improvements of about 4% with the semantic regularizer, suggesting that keeping the semantic embeddings and the local embeddings consistent with each other is beneficial to improve the accuracy and robustness of the model.

F. Visualization

In order to make a more intuitive comparison and verify the effectiveness of the proposed SHNE, we conduct visualization

TABLE VII
ACCURACY COMPARISONS BETWEEN THE SHNE AND SHNE WITHOUT SEMANTIC REGULARIZATION (%)

Model	Citeseer	Pubmed	UAI	BlogCatalog	Flickr	ACM	CoraFull
SHNE w/o L_c	75.7	79.5	74.7	87.8	77.4	91.4	65.3
SHNE	**77.2**	**79.7**	**76.2**	**91.5**	**78.8**	**91.6**	**66.4**

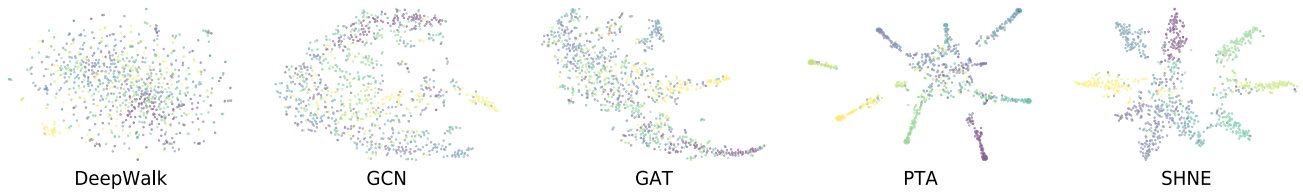


Fig. 8. Visualization embedding on Flickr.

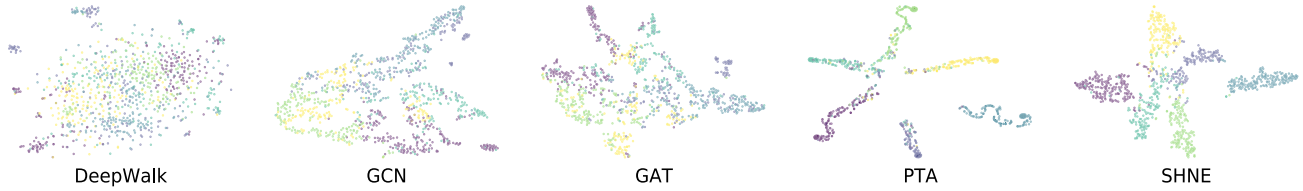


Fig. 9. Visualization embedding on BlogCatalog.

experiments that project the learned node embedding into a 2-D space. Specifically, we utilize t-SNE [54] to perform visualization in BlogCatalog and Flickr and report the results in Figs. 8 and 9.

From the two figures, we can observe that DeepWalk, GCN, and GAT do not perform well because the nodes with different labels are mixed together. Compared to the above methods, it is evident that PTA and the proposed SHNE separate nodes with relatively clear borders. However, there are many misclassified nodes in several clusters from PTA in BlogCatalog, and PTA only displays eight clusters in Flickr, which indicates suboptimal experimental results. Apparently, SHNE performs best because the divided clusters correspond to all the classes of nodes, and the node embeddings have the highest intraclass similarity among different classes.

V. CONCLUSION

In this article, we propose a novel SHNE model for capturing useful information beyond local structures. The proposed model mines rich semantics from local structures and long-range positions. Moreover, with the proposed dual-attention mechanisms, embeddings with various semantics are fused to generate comprehensive node representations. Besides, a simple but effective regularizer is also designed to improve the quality of combined representations. Experimental results demonstrate that SHNE is superior to state-of-the-art models. Based on the experimental analysis, there will be some interesting directions for further studies, such as the design of NL GNNs that encode more complex semantics for heterogeneous graphs. Although SHNE achieves better performance than baselines, there is still room for improvement. In the future, we would like to explore other ways to build more robust NL graphs rather than simply constructing k-neighbor graphs, and we are also interested in exploring the adoption of our model for other graph-based tasks, such as social recommendation or fraudster detection.

REFERENCES

[1] J. Huang, C. Chen, F. Ye, W. Hu, and Z. Zheng, "Nonuniform hyper-network embedding with dual mechanism," *ACM Trans. Inf. Syst.*, vol. 38, no. 3, pp. 1–18, Jun. 2020.

[2] C. Chen, Y. Li, H. Qian, Z. Zheng, and Y. Hu, "Multi-view semi-supervised learning for classification on dynamic networks," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105698.

[3] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "AM-GCN: Adaptive multi-channel graph convolutional networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 1243–1253.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[5] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[6] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[7] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 135–144.

[8] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, and P. S. Yu, "A survey on heterogeneous graph embedding: Methods, techniques, applications and sources," 2020, *arXiv:2011.14867*. [Online]. Available: http://arxiv.org/abs/2011.14867

[9] F. Ye, C. Chen, Z. Wen, Z. Zheng, W. Chen, and Y. Zhou, "Homophily preserving community detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2903–2915, Aug. 2020.

[10] X. Lin, Z. Quan, Z.-J. Wang, T. Ma, and X. Zeng, "KGNN: Knowledge graph neural network for drug-drug interaction prediction," in *Proc. IJCAI*, 2020, pp. 2739–2745.

[11] S. Harada *et al.*, "Dual graph convolutional neural network for predicting chemical networks," *BMC Bioinf.*, vol. 21, no. S3, pp. 1–13, Apr. 2020.

[12] K. Liu *et al.*, "Chemi-Net: A molecular graph convolutional network for accurate drug property prediction," *Int. J. Mol. Sci.*, vol. 20, no. 14, p. 3389, Jul. 2019.

[13] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," 2019, *arXiv:1904.08082*. [Online]. Available: http://arxiv.org/abs/1904.08082

[14] L. Zhang *et al.*, "Structure-feature based graph self-adaptive pooling," in *Proc. Web Conf.*, Apr. 2020, pp. 3098–3104.

[15] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph convolutional networks with EigenPooling," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 723–731.

[16] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*. [Online]. Available: http://arxiv.org/abs/1706.02263

[17] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[18] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," 2020, *arXiv:2002.02126*. [Online]. Available: http://arxiv.org/abs/2002.02126

[19] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," 2017, *arXiv:1704.01212*. [Online]. Available: http://arxiv.org/abs/1704.01212

[20] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," 2018, *arXiv:1801.07606*. [Online]. Available: http://arxiv.org/abs/1801.07606

[21] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 27–34.

[22] I. Spinelli, S. Scardapane, and A. Uncini, "Adaptive propagation graph convolutional network," 2020, *arXiv:2002.10306*. [Online]. Available: http://arxiv.org/abs/2002.10306

[23] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional ARMA filters," 2019, *arXiv:1901.01343*. [Online]. Available: http://arxiv.org/abs/1901.01343

[24] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," 2018, *arXiv:1806.03536*. [Online]. Available: http://arxiv.org/abs/1806.03536

[25] H. Pei, B. Wei, K. Chen-Chuan Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," 2020, *arXiv:2002.05287*. [Online]. Available: http://arxiv.org/abs/2002.05287

[26] Y. Xiao, Z. Zhang, C. Yang, and C. Zhai, "Non-local attention learning on large heterogeneous information networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 978–987.

[27] A. Sankar, X. Zhang, and K. C.-C. Chang, "Meta-GNN: Metagraph neural network for semi-supervised learning in attributed heterogeneous information networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2019, pp. 137–144.

[28] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[30] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*. [Online]. Available: http://arxiv.org/abs/1810.00826

[31] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNs?" in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9267–9276.

[32] M. Liu, H. Gao, and S. Ji, "Towards deeper graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 338–348.

[33] Y. Rong, W. Huang, T. Xu, and J. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[34] J. Hu, R. Cheng, K. C.-C. Chang, A. Sankar, Y. Fang, and B. Y. H. Lam, "Discovering maximal motif cliques in large heterogeneous information networks," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 746–757.

[35] R. Gupta, S. M. Fayaz, and S. Singh, "Identification of gene network motifs for cancer disease diagnosis," in *Proc. IEEE Distrib. Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*, Aug. 2016, pp. 179–184.

[36] Y.-H. Wen, L. Gao, H. Fu, F.-L. Zhang, and S. Xia, "Graph CNNS with motif and variable temporal block for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8989–8996.

[37] H. Zhao, Y. Zhou, Y. Song, and D. L. Lee, "Motif enhanced recommendation over heterogeneous information network," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2189–2192.

[38] B. Xu, J. Huang, L. Hou, H. Shen, J. Gao, and X. Cheng, "Label-consistency based graph neural networks for semi-supervised node classification," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1897–1900.

[39] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proc. Web Conf.*, Apr. 2020, pp. 1400–1410.

[40] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

[41] R. H. Singh *et al.*, "Movie recommendation system using cosine similarity and KNN," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 5, pp. 556–559, Jun. 2020.

[42] S. Popov, J. Gunther, H.-P. Seidel, and P. Slusallek, "Experiences with streaming construction of SAH KD-trees," in *Proc. IEEE Symp. Interact. Ray Tracing*, Sep. 2006, pp. 89–94.

[43] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe LSH: Efficient indexing for high-dimensional similarity search," in *Proc. 33rd Int. Conf. Very Large Data Bases, (VLDB)*, 2007, pp. 950–961.

[44] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 1–8.

[45] X. Wang *et al.*, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, May 2019, pp. 2022–2032.

[46] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903

[47] Z. Meng, S. Liang, H. Bao, and X. Zhang, "Co-embedding attributed networks," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 393–401.

[48] S. Abu-El-Haija *et al.*, "MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," 2019, *arXiv:1905.00067*. [Online]. Available: http://arxiv.org/abs/1905.00067

[49] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," 2021, *arXiv:2101.00797*. [Online]. Available: http://arxiv.org/abs/2101.00797

[50] H. Dong *et al.*, "On the equivalence of decoupled graph convolution network and label propagation," 2020, *arXiv:2010.12408*. [Online]. Available: http://arxiv.org/abs/2010.12408

[51] A. Paszke *et al.*, "Automatic differentiation in Pytorch," Tech. Rep., 2017.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[53] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[54] L. Van der Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.

Ziyang Zhang received the B.S. degree from the Hebei University of Technology, Tianjin, China, in 2020. He is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China.

His research interests include network embedding, data mining, and machine learning.

Chuan Chen (Member, IEEE) received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2016.

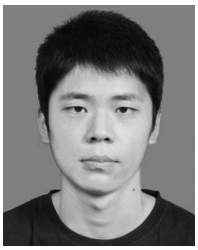
He is currently an Associate Professor with the School of Computer Science and Engineering, Sun Yat-Sen University. His current research interests include machine learning, numerical linear algebra, social network analysis, and network learning.

Yaomin Chang received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2019, where he is currently pursuing the M.S. degree.

His current research focuses mainly involve graph representation learning and adversarial learning.

Weibo Hu received the B.S. degree in computer science from Chongqing University, Chongqing, China, in 2018. He is currently pursuing the master's degree in computer science with Sun Yat-sen University, Guangzhou, China.

His research interests include data clustering, representation learning, network embedding, and graph neural networks.



Xingxing Xing received the B.S. degree from Beijing Normal University, Beijing, China, in 2012, and the Ph.D. degree from Peking University, Beijing, in 2017.

He is currently a Research Scientist with NetEase, Inc., Hangzhou, China. His current research interests include machine learning and recommend systems.

Yuren Zhou received the B.Sc. degree in mathematics from Peking University, Beijing, China, in 1988, and the M.Sc. degree in mathematics and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 1991 and 2003, respectively.

He is currently a Professor with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include the design and analysis of algorithms, evolutionary computation, and social networks.

Zibin Zheng (Senior Member, IEEE) received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2011.

He is currently a Professor with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China, where he is also the Chairman of the Software Engineering Department. He has authored or coauthored more than 120 international journal articles and conference papers, including three ESI highly cited papers. According to Google Scholar, his papers have more than 7000 citations, with an H-index of 42. His current research interests include blockchain, services computing, software engineering, and financial big data.

Dr. Zheng was a recipient of several awards, including the Top 50 Influential Papers in Blockchain of 2018, the ACM SIGSOFT Distinguished Paper Award at the International Conference on Software Engineering (ICSE) 2010, and the Best Student Paper Award at the International Conference on Web Services (ICWS) 2010. He has served as the General Co-Chair of BlockSys'19 and CollaborateCom'16 and the PC Co-Chair of the 19th International Symposium on Cloud and Service Computing (SC2'19), the 18th International Congress on Internet of Things (ICIOT'18), and the 14th International Conference on Internet of Vehicles (IoV'14).