SR-HGAT: Symmetric Relations Based Heterogeneous Graph Attention Network

ZHENGHAO ZHANG, JIANBIN HUANG, AND QINGLIN TAN
School of Computer Science and Technology, Xidian University, Xi'an 710071, China
Corresponding author: Jianbin Huang (jbhuang@xidian.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61876138.

ABSTRACT Graph neural network, as a deep learning based graph representation technology, can capture the structural information encapsulated in graphs well and generate more effective feature embedding. We have recently witnessed an emerging research interests on it. However, existing models are primarily focused on handling homogeneous graphs. When designing graph neural networks for heterogeneous graphs, heterogeneity and rich semantic information bring great challenges. In this paper, we extend graph neural network to heterogeneous graph scenes, and propose a novel high-order Symmetric Relation based Heterogeneous Graph Attention Network, denoted as SR-HGAT, which takes into account the features of nodes and high-order relations simultaneously, and exploits the two-layer attention mechanism based aggregator to efficiently capture essential semantics in an end-to-end manner. The proposed SR-HGAT first identifies the latent semantics underneath the observed explicit symmetric relations guided by different meta-paths and meta-graphs in a heterogeneous graph. The nested propagation mechanism for aggregating semantic and structural features that different links contain is then designed to calculate the interaction strength of each symmetric relation. As the core of the proposed model, to comprehensively capture both the structural and semantic feature information, a two-layer attention mechanism is applied to learn the importance of different neighborhood information as well as the weights of different symmetric relations. These latent semantics are then automatically fused to obtain unified embeddings for specific mining tasks. Extensive experimental results offer insights into the efficacy of the proposed model and have demonstrated that it significantly outperforms state-of-the-art baselines across three benchmark datasets on various downstream tasks.

INDEX TERMS Graph neural networks, heterogeneous graphs, graph representation learning, symmetric relation, graph attention network.

I. INTRODUCTION
Graphs are one of the most expressive data structures which have been used to model a variety of problems. The early processing of graph data is mainly done through network embedding ([1]–[3]), and the basic idea behind various embedding approaches is to use dimensionality reduction techniques to distill the high dimensional topology structure information and relationship between nodes into a dense representation space. With the rapid development of deep learning technology, many scholars have begun to leverage the powerful feature extraction ability of deep neural network to process graph data. As a powerful deep learning based representation technology, Graph Neural Network (GNN) employs deep neural network to aggregate feature information of neighboring nodes, which makes the aggregated embedding more powerful and arouses considerable research interest. In addition, the GNN based models (e.g. GCN [4], GraphSAGE [5], and GAT [6]) can be naturally applied to inductive tasks involving nodes that are not present in the training period. However, the advances and applications of GNNs are largely concentrated on homogeneous graphs.

As a matter of fact, the real-world online platforms (e.g Facebook, Amazon, DBLP and IMDB) usually comes together with the heterogeneous graph structure, which contains abundant information with various relations among multi-typed nodes as well as unstructured content associated with each node, also widely known as heterogeneous
information network (HIN). For convenience, we uniformly call it heterogeneous graph in this paper. Current state-of-the-art GNNs lack the aggregation mechanism to encapsulate rich semantics and complex relationships in heterogeneous graphs, and directly applying these to embed heterogeneous graphs will inevitably lead to unsatisfactory embedding results and reduced performance in downstream tasks.

Based on the above analysis, when designing graph neural network architecture to deal with heterogeneous graphs, we need to address the following challenges.

- **Capturing distinctive semantic characteristics of different relations** Existing GNNs usually learn heterogeneous graph embedding by solely focusing on node features without taking into account the features of links, making them insufficient to capture the properties of heterogeneous graphs. Moreover, heterogeneous graphs have been commonly used for abstracting and modeling complex systems, in which various relations with rich latent semantics are involved in. One must effectively discover the corresponding latent semantic causing a specific semantic relationship, and embed these rich semantics into feature vector space.

- **Integrating the attention mechanism into graph neural network for heterogeneous graph** In a heterogeneous graph, any node pair can be connected with various types of relationships. For any symmetric relationship, each node has a large number of neighborhood nodes and links based on this relation. How to distinguish the subtle differences of neighbors and then automatically infer different attention values to them is worth looking forward to. Moreover, according to the form of relationships, meaningful and complex semantic information can be contained in heterogeneous graphs. It is impractical to treat these different semantic relationships equally, which will weaken the semantic information of more useful relationships. How to choose the best semantic relationship information and integrate them according to specific mining tasks is still an open problem.

In light of previous limitations and challenges, we propose a high-order Symmetric Relation based Heterogeneous Graph Attention Network framework (SR-HGAT). The SR-HGAT mainly consists of three modules to correspondingly address the challenges in graph neural network modeling for heterogeneous scenes. First, in the semantics capturing module, the concept of symmetry relations guided by different meta-paths and meta-graphs is proposed, which can facilitate better identification and capture of rich and complex latent semantic information of any node’s neighborhood. Moreover, a specifically designed module is proposed, namely nested propagation of multi-type link features. For each kind of symmetric relation, both semantic and structural feature information of links are nestedly aggregated to capture the distinct characteristics of high-order relations. Hence, the feature of both relations and neighbor nodes can be integrated to capture more accurate neighborhood information.

As the core of our proposed model, in the two-layer attention based aggregator module, the first layer is adopted to learn the attention values between the nodes and their symmetric relation based neighbors, and the second layer is able to automatically determine the weights of these latent semantics represented by symmetric relations for different downstream tasks. Based on the learned attention values in terms of the two-layers mechanism, our proposed model can obtain the optimal combination of neighbors and multiple symmetric relations in a hierarchical manner.

It is worthwhile to highlight our main contributions as follows:

- We formalize the problem of heterogeneous graph representation learning which involves both node features and link features of multiple types simultaneously.
- To explicitly capture the distinctive semantic characteristics of diverse high-order symmetric relations, we design a nested propagation strategy of multi-typed link features on these relations.
- We propose a two-layer attention mechanism, which can automatically capture both the importance of neighbor information of any node as well as the weights of different symmetric relations in a hierarchical manner.

Extensive experiments conducted on three benchmark datasets validate the superiority of our proposed model via comparison with other state-of-the-art baselines for numerous graph mining tasks.

The rest of this paper is organized as follows: A review of related work is provided in Section II. Preliminary and several related definitions are introduced in Section III. The proposed model is elaborated in Section IV. Experimental results and dataset descriptions are reported in Section V, conclusions and future research directions are presented in Section VI.
meta-paths and the weight of each meta-path should be assigned by user in advance. Moreover, Zhang [15] comes up with Metagraph2vec, in an attempt to use meta-graphs as guidelines for random walk. Richer structural details and more complete semantic information are successfully extracted. HIN2Vec [16] carries out multiple prediction training tasks that learn the latent vectors of nodes into the same relation space and conducts heterogeneous link prediction. In addition, there are also several other heterogeneous graph embedding methods designed for some particular tasks, such as identifying authors [17] and recommendation tasks [18].

**B. GRAPH NEURAL NETWORK**

Graph neural network, introduced in [19], [20], aims to develop deep neural network to handle network structural data. It has attracted considerable attention lately, because of its remarkable success in various graph analysis tasks. The early attempts to derive a graph convolutional layer are based on graph spectral theory [21]–[23]. However, spectral theory based graph convolutional networks still have difficulty in capturing local graph information. Recently, non-spectral approaches [24]–[26] are proposed to perform a convolution process to aggregate the neighbor nodes’ information directly so that spatially close neighbors information can be captured directly. One of the most prominent progress is GCN [4], which averages the neighbor of each node in the graph, followed by a linear projection and non-linear activation operations. William et al. [5] proposed GraphSAGE, an inductive framework that leverages node sampling and feature aggregation techniques to efficiently generate node embeddings for unseen data, which breaks the limitation of applying GCN in transductive settings. Graph attention network (GAT) [6] incorporates attention mechanism into GCN. By calculating attention coefficients among nodes, GAT allows each node to focus on the most relevant neighbors to make decisions. The aforementioned models exhibit state-of-the-art performance in various graph representation learning applications. However, above graph neural network cannot be directly applied to heterogeneous graphs with multi-relational edges and multi-typed nodes. Recently, several attempts have been made to adopt GNNs to learn with heterogeneous graphs. Schlichtkrull et al. [27] propose the relational graph convolutional networks (R-GCN) to model knowledge graphs. R-GCN keeps a distinct linear projection weight for each edge type. Zhang et al. [28] present the heterogeneous graph neural networks (HetGNN) that adopts different RNNs for different node types to integrate multi-modal features. Although these methods have been shown to be empirically better than traditional models, they have not fully utilized the heterogeneous graphs’ properties.

**III. PRELIMINARIES**

We begin this section by introducing the notations used in the rest of the paper, followed by some related definitions about Heterogeneous graphs and a brief background on Graph Attention network (GAT). Finally, we will introduce our proposed symmetric relations guided by meta-path and meta-graph.

**A. HETEROGENEOUS GRAPH**

A heterogeneous graph is a directed graph, denoted as $G = (V, E, A, R)$, which consists of the set of nodes $V$, the set of links $E$, the sets of node types $A$ and the edge types $R$ respectively. In usual cases, a heterogeneous graph is associated with a node type mapping function $\Phi : V \rightarrow A$ which represents that each vertex $v \in V$ can be mapped into a node type $a \in A$, and a link type mapping function is expressed as $\Psi : E \rightarrow R$, meaning that each edge $e \in E$ can be mapped into an edge type $r \in R$. A network can be deemed as a heterogeneous graph when the number of node types $|A|$ and edge types $|R|$ conform to $\mid A \mid \mid R \mid > 2$. Otherwise, it is a homogeneous graph.

**B. SEMANTIC STRUCTURE OF HETEROGENEOUS GRAPH**

Complexity of the heterogeneous graph forces us to propose a structure, which can describe meta-level semantic information of a heterogeneous graph, so that node types and relations between nodes in the network can be better comprehended. Given a heterogeneous graph, a graph schema can be abstracted as $\Theta_G = (A, R)$, where $\Theta_G$ is a directed graph that contains all allowable node and link types, these types are combined together to conduct meta-level semantic description of $G$.

![FIGURE 1. The schema of DBLP dataset.](image1)

![FIGURE 2. The schema of MovieLens dataset.](image2)

To be more specific, in Figure 1, DBLP schema consists of four node types (A for authors, P for papers, V for venue and T for topic), and three link types (an author-paper link signifying papers published by the author. A paper-venue link signifying a venue on which a paper is published. A paper-topic link signifying keywords of a paper). By contrast, in Figure 2, MovieLens schema comprises 6 node types of U (users), M (movies), A (actors), D (directors), T (tags)
and G (genres) and five link types including users watching and reviewing movies, actors in movies, directors directing movies, tags of movies and cinematic genres.

A heterogeneous graph contains abundant semantic information that can be expressed by graph schemas. Moreover, both meta-path and meta-graph are semantic structures generated from heterogeneous graph schema.

**Definition 1 (Meta-Path):** As an abstract sequence of node types connected by link types, the meta-path is formed by transforms of a graph schema and able to capture rich semantic information preserved in heterogeneous graphs. Specifically, given a HIN schema denoted as \( \Theta_G = (A, R) \), a meta-path can be expressed in the form of \( P = a_1 \xrightarrow{r_{1,2}} a_2 \xrightarrow{r_{2,3}} \ldots a_{i-1} \xrightarrow{r_{i-1,i}} a_i \), where \( a_i \in A(i = 1, \ldots, l) \) indicates node types and \( r_{i,j} \in R \) represents link types between \( a_i \) and \( a_j \), \( 1 \leq i, j \leq l \).

![Figure 3. Metapath instances of DBLP dataset.](image)

For example in Figure 3, the meta-path A-P-V-P-A in the academic cooperation network DBLP indicates that papers written by two authors are published on the same venue. In addition, A-P-T-P-A and A-P-A-P-A respectively mean that papers published by two authors share the same subject and papers published by two authors have the same co-author.

![Figure 4. Metapath instances of MovieLens dataset.](image)

Moreover, for example in Figure 4, the meta-path U-M-G-M-U in the movie review network MovieLens indicates that movies rated by two users contain the same genre. In addition, U-M-A-M-U and U-M-D-M-U respectively mean that movies rated by two users are acted by the same actor and have the common director. Clearly, different meta-paths represent different semantics.

**Definition 2 (Meta-Graph):** In terms of topological structures, a meta-graph is more complicated than a meta-path and thus has the potential to represent richer and more complicated semantic contents. Similar to meta-path, given a heterogeneous graph schema \( \Theta_G = (A, R) \), the corresponding meta-graph can be represented as a directed acyclic graph \( M_G = (N, M, n_s, n_t) \), which contains a source node \( n_s \) (in-degree: 0) and a target node \( n_t \) (out-degree: 0). It depicts a complex semantic relation formed between \( n_s \) and \( n_t \) by stitching. Moreover, \( N \) and \( M \) respectively represent subsets of node types and edge types. And they meet \( N \subseteq A \) and \( M \subseteq R \) respectively.

We continue to take DBLP and MovieLens as examples in Figure 5 and 6 respectively. Meta-graph \( (A, P, (V, T), P, A) \) implies that not only are papers written by two authors published on the same venue, but these papers contain identical terms. In a similar way, Meta-graph \( (U, M, (A, D), M, U) \) in MovieLens dataset represents that movies rated by two users are acted by the same actor and have the common director.

![Figure 5. A meta-graph example of DBLP dataset.](image)

![Figure 6. A meta-graph example of MovieLens dataset.](image)

### C. Symmetric Relations

In heterogeneous graphs, there are various relationships between two same-typed nodes guided by meta-paths and meta-graphs. Therefore, we propose to leverage both meta-graph and its embedded meta-paths for describing various symmetric relations between node pairs. Related definitions will be summarized as follows:

**Definition 3 (Symmetric Relation Guided by Meta-Path):**

Given a meta-path denoted as \( P = A_1 \xrightarrow{E_1} A_2 \xrightarrow{E_2} \ldots \xrightarrow{E_{l-1}} A_{l+1} \), where \( A_i \) denoted as an node type, \( E_j \) represents a link type. And we can say that \( P = A_1 \xrightarrow{E_1} A_2 \xrightarrow{E_2} \ldots \xrightarrow{E_{l-1}} A_{l+1} \) is symmetric if \( P \) is equal to \( P^{-1} \) shown in the Figure 7, where \( A_i \) and \( T \) represent the node types connected by link type.\( A_i \) and \( T \) represent the node types connected by link type.
and \( A_{i+2−j} (i = 1, 2, \ldots, l + 1) \) has the same type, \( E_j \) and \( E_{i+1−j} (j = 1, 2, \ldots, l) \) represents the same semantic. Then \( SR_P \) can be defined as a symmetric relation guided by this symmetric meta-path \( P \).

**Definition 4 (Symmetric Relation Guided by Meta-Graph):** In Figure 8, \( (A_{i+1}, A_{i+2}, \ldots, A_{i+n(i)}) \) denoted as the \( i \)-th node type group and \( n_o(i) \) represents the number of node types, similarly, \( (E_j, E_{j+2}, \ldots, E_{j+n(j)}) \) represents the \( j \)-th link type group and \( n_r(j) \) represents the number of link types, and we say \( M \) is symmetric if \( M \) is equal to \( M^{-1} \). \( SR_M \) can be denoted as a symmetric relation guided by meta-graph \( M \).

![Symmetric relation guided by a meta-graph.](image)

**Definition 5 (Symmetric Relation Based Neighbors):** We notice that, each node connects with other nodes with the same type according to symmetric relations guided by meta-paths or meta-graphs. Given a node-relation triple \((v_s, r, v_t)\) in a heterogeneous graph, where \( r \) denotes as a symmetric relation between node \( v_s \) and \( v_t \). Then the relation based neighbors \( N_{sr} \) of node \( v_s \) are defined as the set of nodes which connect with current node \( v_s \) via symmetric relation \( sr \). Due to different strength of semantic relation, different symmetric relations based neighbors of each node show different importance. Notice that we can actually analyze all possible symmetric relations based on meta-paths and meta-graphs. However, not all relations have meaningful semantics and positive effects on embeddings. Hence, we only choose important and meaningful symmetric relations in this paper.

IV. SR-HGAT: SYMMETRIC RELATIONS BASED HETEROGENEOUS GRAPH ATTENTION NETWORK

In this paper, we extend graph attention network to heterogeneous graph scenes, and propose a semi-supervised Symmetric Relation Based Heterogeneous Graph Attention Network, denoted as SR-HGAT, to take into account the features of nodes and high-order symmetric relations simultaneously, and adopt a two-layer hierarchical attention mechanism to embed heterogeneous graph in an end-to-end fashion. Figure 9 shows the overall framework of our proposed SR-HGAT, and we will provide a detailed description of it in the following subsections.

A. MODEL INPUTS

In our model, the feature vectors of nodes and high-order relations are taken simultaneously as inputs. For any node \( v_i \), we define the triple \((v_i, sr_k, v_j)\) at first, which means neighbor node \( v_j \) connects with node \( v_i \) via symmetric relation \( sr_k \), where the neighbor node \( v_j \) \( v_j \in N_{v_i}^{sr_k} \) \((N_{v_i}^{sr_k} \) represents the neighbor nodes set of node \( v_i \) via symmetric relation \( sr_k \)). Moreover, \( h_i, \tilde{h}_i \in H \) represent feature vectors of \( v_i, v_j \) respectively and \( r_k \) represents feature vector of \( sr_k \). Specifically, features of nodes can be expressed as matrices \( H \in \mathbb{R}^{N_v \times T_o} \), where the \( i \)-th line \( h_i \in H \) represents the vector of node \( v_i \). \( N_{v_i} \) represents the total number of nodes and \( T_o \) is the feature dimension of each node representation.

B. NESTED PROPAGATION OF MULTI-TYPED LINK FEATURES

The GAT designed for homogeneous graphs ignores the semantic and structural features of different types of links, as well as the complex and rich semantic relationship between nodes, which is the extremely important part of heterogeneous graphs. To fit the general framework to heterogeneous graph, in addition to the information from neighbor nodes, the features of symmetric relations connected to them are also collected in the proposed model. Each symmetric relation guided by a meta-path or a meta-graph is composed of multiple types of links, and the hidden state is updated according to the nested aggregation of features of the composed links. Subsequently, the detailed process will be described as follows:

**Definition 6 (Structural and Semantic Feature of Link):** Considering that multi-typed links in a heterogeneous graph have significantly different characteristics, for any link \( e_i \in sr_k \) which belongs to \( sr_k \), we first explore how to calculate the structural and semantic features of it respectively.

**In structural feature view:** Considering that the degree of nodes can well reflect the topological characteristics of networks, a degree based measure is defined to explore the structural features of various links. We assume that \( d_i \) represents the degree of source node in a directed link \( e_i \), and then the structural strength of link \( e_i \), denoted as \( \tilde{d}_i \), can be defined as:

\[
\tilde{d}_i = \frac{1}{d_i},
\]

(1)

Moreover, link with smaller value of \( \tilde{d}_i \) imply stronger structural strength.

**In semantic feature view:** It is assumed that the semantic feature \( a_i \) represents the semantic features of link \( e_i \), then the absolute semantic strength of link \( e_i \), denoted as \( \tilde{a}_i \), can be defined as below:

\[
\tilde{a}_i = \frac{|M_i - a_i| + 1}{M_i},
\]

(2)

where \( M_i \) is the maximum strength of semantic which link \( e_i \) represents (belongs to).
For instance, in MovieLens dataset, the semantic of link User → Movie represents that user has watched a movie, and the semantic strength of this link represents user’s attitudes to films. The larger the value is, the more importance of the film contributes to the user. Therefore, User → Movie should be attached more importance than User → Movie. Similarly, in DBLP dataset, the semantic of link Author → Paper represents that author has written a paper, and the semantic strength of this link is the author’s serial number. Author → Paper should be attached more importance than Author → Paper. Therefore, according to different heterogeneous graphs, we should setup the semantic strength of each link in advance.

**Definition 7 (Aggregation of Symmetric Link Pairs on Meta-Path):** Considering that the symmetry relationship is generated by nested aggregating of symmetric link pairs, we thus continue to introduce how to aggregate features of symmetric link pairs that belong to any symmetric relation guided by a meta-path or meta-graph as follows.

For a meta-path instance \( p = a_1 \xrightarrow{e_1} a_2 \xrightarrow{e_2} \ldots \xrightarrow{e_l} a_{l+1} \), it is assumed that link pair \( e_i \) and \( e_{l+1-i} \) are a symmetric link pair and they represent the same semantic. A measure \( rs(i, l+1-i) \) is then defined to explore the distinction of this link pair from both structural and semantic view in a unified manner.

Specifically, the difference of semantic strength between a symmetric link pair, denoted as \( ra(i, l+1-i) \), can be calculated as:

\[
ra(i, l+1-i) = \frac{D_i - |a_i - a_{l+1-i}|}{D_i} \times \tilde{a}_i \times \tilde{a}_{l+1-i},
\]  

where \( D_i \) is the maximum semantic strength of specific link type that \( e_i \) or \( e_{l+1-i} \) belongs to.

Then the difference of structural strength between any symmetric link pair, denoted as \( rd(i, l+1-i) \), can be calculated as:

\[
rd(i, l+1-i) = \frac{\min(d_i, \tilde{d}_{l+1-i})}{\max(d_i, \tilde{d}_{l+1-i})},
\]

in Equ.4, the outer degrees of two source nodes connected with the links \( e_i \) and \( e_{l+1-i} \) are compared by dividing the larger one by the smaller one. A large value indicates the quite inequivalent structural strength of the symmetric link pair. Let’s see a toy example, symmetric link pair (A-P, P-A) represent the same semantic that “authors write papers”. A famous author \( a_1 \) has published 20 papers, and author \( a_2 \) may be a student and has published only one paper, then symmetric link pair connected with two authors as respective source nodes are structural inequivalent, and the value of \( rd \) is only 0.05, which is consistent with common sense.

Combining \( ra(i, l+1-i) \) and \( rd(i, l+1-i) \), the distinction of symmetric link pair can then be calculated as follows:

\[
rs(i, l+1-i) = ra(i, l+1-i) \times rd(i, l+1-i),
\]

a large value of \( rs(i, l+1-i) \) indicates quite equivalent semantic and structural strength of symmetric link pair on the relations sr. In other words, any symmetric link pairs with a large value of \( rs(i, l+1-i) \) show much stronger affiliation relationships and much more similar properties.

**Definition 8 (Aggregation of Symmetric Link Pairs on Meta-Graph):** In a similar manner, for a meta-graph instance, we may face the following two situations:

**For each symmetric link pair:** the distinction measure is the same as it in meta-path above.

**For each symmetric link group pair:** \( (e_{i,1}, e_{i,2}, \ldots, e_{i,n}) \) and \( (e_{i+1-i,1}, e_{i+1-i,2}, \ldots, e_{i+1-i,n}) \), the difference of

---

**FIGURE 9.** The overall framework of SR-HGAT, which consists of four steps: (a) semantics capturing module, which models multiple symmetric relations guided by meta-paths and meta-graphs to capture rich latent semantics of any node’s neighborhood. (b) nested propagation of multi-typed links, which can explicitly capture the distinctive semantic features of different symmetric relations. (c) two-layer attention based aggregator, which jointly learns the importance of both neighborhood information and semantic relations in a hierarchical manner. (d) calculate the loss and optimize the overall model via backpropagation in an end-to-end manner.
structural and semantic strength between any symmetric link pair $e_{i,j} \in (e_{i,1}, e_{i,2}, \ldots, e_{i,n})$ and $e_{i+1-i,j} \in (e_{i+1-i,1}, e_{i+1-i,2}, \ldots, e_{i+1-i,n})$, denoted as $rd(i, l+1-i, j)$ and $ra(i, l+1-i, j)$, can be calculated in a similar manner respectively:

$$rd(i, l+1-i, j) = \frac{\min(\tilde{a}_{i,j}, \tilde{a}_{l+1-i,j})}{\max(\tilde{a}_{i,j}, \tilde{a}_{l+1-i,j})},$$

$$ra(i, l+1-i, j) = \frac{D_{i,j} - |a_{i,j} - a_{l+1-i,j}|}{D_{i,j}} \times \tilde{a}_{i,j} \times \tilde{a}_{l+1-i,j},$$

where $D_{i,j}$ is the difference between the maximum and minimum semantic strength of the $j$-th link that belongs to $i$-th link group.

Therefore, the distinction of any symmetric link group pair $(e_{i,1}, e_{i,2}, \ldots, e_{i,n})$ and $(e_{i+1-i,1}, e_{i+1-i,2}, \ldots, e_{i+1-i,n})$ which both belong to the meta-graph and represent the same semantic can be defined as:

$$\sum_{j=1}^{n} ra(i, l+1-i, j) \times rd(i, l+1-i, j)$$

$$rs(i, l+1-i) = \frac{n}{\sum_{j=1}^{n} ra(i, l+1-i, j) \times rd(i, l+1-i, j)},$$

**Definition 9 (Nested Propagation of Symmetric Relations):** The idea of “nested propagation” is employed to recursively propagate the distinction of all link pairs. After the nested aggregation process, the distinction strength of any symmetric relation guided by meta-path or meta-graph can be learned as follows:

$$RS = \prod_{i=1}^{n} rs(i, l+1-i),$$

symmetric relation with smaller value of $RS$ implicates much stronger interaction relationships between nodes.

**Definition 10 (Neighbor Representations):** The features of symmetric relation learned from above nested propagation process, together with representations of the node that this relation connected with, is utilized to obtain the “neighbor representation”, which enables the neighborhood feature information to be depicted in more detail and can be defined as follows:

$$\tilde{H}_{i}^{k,j} = \{\text{concat}(RS_k, \tilde{h}_j)\},$$

where $\text{concat}$ represents a linear transformation operation over the combination of symmetric relation $sr_k$’s features and neighbor node $v_j$’s features $\tilde{h}_j \in \mathbb{R}^d, \forall v \in V$ for the source node $v_i$.

**C. TWO-LAYER ATTENTION MECHANISM**

Before aggregating the information from symmetric high-order relation based neighbors for each node, it should be noted that different relation based neighbors of each node play a different role and show different importance in learning node embedding for the specific task. As discussed above, distinct from traditional neighborhood aggregators (e.g., GCN and GAT) which only takes node representations as inputs, we adopt a two-layer hierarchical attention mechanism as for the specific forms of aggregation function which can holistically encode semantically different high-order relations in the neighborhood of any given node during propagation on a heterogeneous graph.

1) INTRA-RELATION ATTENTION

The intra-relation attention is applied first to infer the weight coefficients of neighborhood that are actually connected with source node due to a specific symmetric relation. The weight parameters $\hat{e}^k_{ij}$, which means how important neighbor node $v_j$ via symmetric relation $sr_k$ will be for node $v_i$, can be formulated as follows:

$$\hat{e}^k_{ij} = att_{node}(\tilde{h}_i, \tilde{H}_{i}^{k,j}),$$

where $att_{node}$ denotes the deep neural network that performs the intra-relation attention, and $\tilde{H}_{i}^{k,j}$ the “neighbor vector”, which means the combination of symmetric relation $sr_k$ features and neighbor node $v_j$’s features for the source node $v_i$. Note also that $\hat{e}^k_{ij}$ is asymmetric which maintains the different contributions of the neighbor vector $\tilde{H}_{i}^{k,j}$ to current node $v_i$.

After obtaining the importance between symmetric relation based node pairs, they are normalized to obtain the weight coefficient $a_{ij}^k$ via a softmax function:

$$a_{ij}^k = \text{soft max}(\hat{e}^k_{ij}) = \frac{\exp(\gamma(\hat{h}_i||\tilde{H}_{i}^{k,j}))}{\sum_{n \in N^k_i} \exp(\gamma(\hat{h}_i||\tilde{H}_{i}^{k,n}))},$$

where $\gamma$ denotes the activation function, $||$ denotes the concatenate operation, $N^k_i$ denotes the neighborhood of entity $v_i$ via symmetric relation $sr_k$.

Then, the embedding of node $i$ can be updated by the symmetric relation based neighbor features with the corresponding coefficients as follows:

$$z_i^k = \sigma(\sum_{j \in N^k_i} a_{ij}^k \cdot \tilde{H}_{i}^{k,j}),$$

**FIGURE 10.** Explanation of aggregation process in both intra-relation and relation-level attention.
where $z_i$ is the updated embedding of node $v_i$ for the symmetric relation. The new embedding of the entity is the sum of each triple representation weighted by their attention values as shown in Eq.14. Since the attention weight $\beta_i$ is generated for one single symmetric relation, it is semantic specific and able to capture one kind of semantic information.

Generally speaking, each node pair in a heterogeneous graph contains multiple types of semantic information and specific semantic based embedding can only reflect node feature from one aspect. To learn a more comprehensive node embedding, we need to further fuse multiple semantics which can be revealed by symmetric relations guided by different meta-paths and meta-graphs.

2) RELATION-LEVEL ATTENTION
To address the challenge of symmetric relation selection and semantic fusion in a heterogeneous graph, a novel combiner with relation-level attention mechanism is proposed to automatically learn the importance of different symmetric relations and fuse them for the specific tasks to obtain the updated node representations.

Given the symmetric relations set $\{sr_1, sr_2, \ldots, sr_K\}$, after feeding neighbor features into intra-relation level attention, we can obtain $k \in K$ groups of semantic specific node embeddings, denoted as $(Z_{sr_1}, Z_{sr_2}, \ldots, Z_{sr_K})$. Taking these $K$ groups of semantic specific node embeddings as inputs, the learned weight of each aggregated symmetric relation $\{\beta_{sr_1}, \beta_{sr_2}, \ldots, \beta_{sr_K}\}$ can be expressed as follows:

$$ w_{sr_i}(Z_{sr_1}, Z_{sr_2}, \ldots, Z_{sr_K}) = \alpha_1\cdot \text{softmax}(\gamma(w_{sr_1}Z_{sr_1} + \ldots + w_{sr_K}Z_{sr_K})),$$

where $\alpha_1$ denotes the deep neural network which performs the symmetric relation level attention. This shows that the semantic-level attention can capture various types of semantic information behind a heterogeneous graph.

After obtaining the importance of each symmetric relation, they are normalized via a softmax function. Specifically, the weight of the $k$-th symmetric relation, denoted as $\beta_{sr_k}$, can be obtained by normalizing the importance of all above relations:

$$ \beta_{sr_k} = \frac{\exp(w_{sr_k})}{\sum_{i=1}^{K}\exp(w_{sr_i})},$$

which can be interpreted as the contribution of the symmetric relations for the specific task, obviously, the higher $\beta_{sr_k}$, the more important relation $sr_k$ is. Note that for different tasks, different symmetric relations may have different weights. With the learned weights as coefficients, these semantic specific representations can be fused to obtain the final representation $Z$ as follows:

$$ Z = \sum_{i=1}^{K} \beta_{sr_i} \cdot Z_{sr_i},$$

With the learned importance of neighbors and symmetric relations, the proposed model can pay more attention to meaningful neighbors or relations for the specific task and give a more comprehensive description of a heterogeneous graph.

D. MODEL OPTIMIZATION

Compared with the general GNN method or traditional heterogeneous graph representation learning model, our proposed model optimizes the representation learning and specific task together in an end-to-end manner. In the previous models, once the feature vectors of nodes are learned, they will be solid down and the supervised information generated in the downstream task cannot effectively guide the updating of graph representation results. On the contrary, in our model, the representation learning of nodes and the learning of downstream tasks are put together for end-to-end learning. The supervised signals of the entire model guide the relevant parameters updating of specific task layer (such as classification layer) and graph attention layer at the same time.

1) FOR SEMI-SUPERVISED NODE CLASSIFICATION

We can minimize the cross entropy over all labeled node between the ground truth and the prediction:

$$ L_{clf} = - \sum_{i \in \mathcal{L}} \omega_i \log(o_{zi}),$$

where $\omega$ represents parameters of the classifier, $\omega_L$ is the set of node indices that have labels, and $o_i$ stands for labels of $z_i$. With the guide of labeled data, we can optimize the proposed model via back propagation and learn the embeddings of nodes.

2) FOR RELATION PREDICTION TASK

(RECOMMENDATION TASK)

The labeled instances are a collection of practical links between nodes that belong to different two types, such as viewing relationships between user nodes and movie nodes in MovieLens dataset, and purchasing relationships between user nodes and item nodes in Amazon dataset. For any node pair $v_i^{A}$ and $v_i^{B}$ which respectively belongs to type $A$ and $B$, after a series of temporal encoding from different views, we can obtain the aggregated latent representations of them, denoted as $(z_i^{A}, z_i^{B})$. The probability of the interaction between $(z_i^{A}, z_i^{B})$ can then be calculated as follows:

$$ \hat{y}_{ij} = \text{sigmoid}(z_i^{A} \cdot z_j^{B}) = \frac{1}{1 + e^{-z_i^{A} \cdot z_j^{B}}},$$

FIGURE 11. Depiction of our SR-HGAT model for node classification and relation prediction.
TABLE 1. Statistics of the datasets.

| Dataset | Relation         | Number of A | Number of B | Number of A-B | Avg. degree of A | Avg. degree of B |
|---------|------------------|-------------|-------------|---------------|----------------|-----------------|
| DBLP    | Paper-Author     | 25473       | 28241       | 70652         | 2.77           | 2.51            |
|         | Paper-Venue      | 25473       | 18          | 25437         | 1              | 13.54           |
|         | Paper-Term       | 25473       | 11230       | 132105        | 5.97           |                 |
| MovieLens | User-Movie   | 2106        | 10197       | 37555         | 17.8           | 3.68            |
|         | Movie-Actor      | 10197       | 95241       | 231737        | 22.73          | 2.43            |
|         | Movie-Director   | 10197       | 4053        | 10197         | 1              | 2.20            |
|         | Movie-Tag        | 10197       | 13222       | 51795         | 5.07           | 3.92            |
|         | Movie-Genres     | 10197       | 20          | 20809         | 2.04           | 1040.45         |
| Amazon  | User-Item        | 6084        | 2753        | 195791        | 32.18          | 71.12           |
|         | Item-Category    | 2753        | 22          | 5508          | 2              | 250.36          |
|         | Item-Brand       | 2753        | 334         | 2753          | 1              | 8.24            |
|         | Item-View        | 2753        | 3857        | 5694          | 2.07           | 1.48            |

where the $\text{sigmoid}(\cdot)$ is the sigmoid layer, and $\hat{y}_{ij}$ is the probability in the range of $[0, 1]$.

Then, the loss function of the proposed model is a point-wise loss function in the following equation:

$$L_{r,p} = - \sum_{i,j \in Y \cup Y^-} (y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log(1 - \hat{y}_{ij})).$$  \hspace{1cm} (19)

where $y_{ij}$ is the ground truth of the label instance, $Y$ and $Y^-$ are the positive and negative instances sets respectively.

E. MODEL TRAINING

By combining the stochastic gradient descent (SGD) and Adaptive Moment Estimation (Adam), relevant parameters including structure loss function and interaction loss function can be continuously optimized. We first make a forward propagation to calculate the loss and then back propagate based on the minimizing the loss function, then relevant model parameters and weights that correspond to different views can be automatically and continuously updated in each iteration. Here, only a few label data corresponding to a specific mining task is needed to train the attention mechanism and fine-tune the GNN encoder.

V. EXPERIMENTAL EVALUATION

The proposed method was evaluated on three real-world datasets to answer the following research questions:

Q1: How does the SR-HGAT perform compared with other state-of-the-art methods for various graph mining tasks, such as node classification, node clustering, and relation prediction?

Q2: How do different components (e.g., nested propagation of multi-typed links, attention mechanism, and aggregator selection) affect the representation capability of the proposed SR-HGAT?

Q3: How do different hyper-parameter settings affect model performance?

A. EXPERIMENTAL SETTING

1) DATASETS

We study all models using data collected from three well-known online applications. The detailed description statistics for all datasets are shown in Table 1.

a: DBLP

The DBLP dataset is an academic network dataset in the field of computer science. Here we extract a subset, denoted as DBLP-4area, which contains relevant literature information of four research areas: databases, information retrieval, data mining and machine learning. In addition, each author’s research area was labeled according to the venues in which they mainly published. For each paper, we use a pre-trained XLNet [29], [30] to get the representation of each word in its title, then we average them weighted by each word’s attention to get the title representation for each paper. The initial feature of each author is then simply an average of his/her published papers’ representations.

b: MovieLens

The MovieLens dataset comprises movie viewing records of a massive number of users and other details related to movies. Here we extract a movie sub-set which consists of five genres including action, adventure, science and education and crime. Each movie falls into at least one of these genres. Movie features correspond to elements of a bag-of-words represented of plots.

c: AMAZON

The Amazon dataset records user ratings on businesses and contains social relations and attribute data of businesses. In the experiment, we select the items of Electronics categories for evaluation, and each item falls into corresponding category. Item features correspond to elements of a bag-of-words represented of product description. The initial feature of each user is then simply an average of his/her purchased items’ representations.

2) BASELINES FOR COMPARATIVE EVALUATION

We compare with some state-of-the-art baselines, including the network embedding methods and graph neural network based methods, to verify the effectiveness of the proposed SR-HGAT.

The first class of baselines includes several state-of-the-art network embedding approaches:
**a: DEEPWALK [2]**
a homogeneous graph embedding method, which performs a random walk on networks and then learns low-dimensional node representations via the skip-gram model.

**b: Node2vec [1]**
a homogeneous graph embedding method. Node2vec defines two parameters of p and q, so that random walk strikes a balance between BFS and DFS to preserve local and global information of nodes.

**c: Metapath2vec [3]**
a heterogeneous graph embedding method, which leverages meta-path based random walk and utilizes skip gram to perform node embedding. Considering that heterogeneous graph may have diverse meta-paths, we select the most efficient meta-path for experiment here to guide random walk.

**d: HIN2Vec [16]**
a heterogeneous network embedding approach. The core of the HIN2Vec framework is a neural network model, which is designed to capture the rich semantics embedded in HINs by exploiting different types of relationships among nodes.

The second class includes several GNN variants that designed for homogeneous and heterogeneous graphs respectively as baselines:

**e: GCN [4]**
a GNN variant designed for homogeneous graphs, which is a semi-supervised graph convolutional network and simply averages the neighbor’s embedding followed by linear projection. Here we test all symmetric relations for GCN and report the best performance.

**f: GAT [6]**
a GNN variant designed for homogeneous graphs, which adopts multi-head additive attention on neighbors. Similar with GCN, we also test all symmetric relations for GAT and report the best performance.

**g: R-GCN [27]**
a dedicated heterogeneous GNN, which keeps a different weight for each relationship, i.e., a relation triplet.

The third class includes several our SR-HGAT variants as baselines:

**h: SR-HGATnl**
It is a variant of SR-HGAT, which removes the nested propagation of multi-typed link features mechanism and only considers the node features.

**i: SR-HGATnr**
It is a variant of SR-HGAT, which removes the relation-level attention and assigns the same weight to each symmetric relation.

**j: SR-HGATnan**
It is a variant of SR-HGAT, which removes the intra-relation attention and assigns the same weight to each neighbor vector.

**k: SR-HGAT**
Our proposed semi-supervised graph neural network which simultaneously employs two-layer hierarchical attention and nested propagation of multi-typed link features.

3) IMPLEMENTATION DETAILS
For the proposed SR-HGAT, in the training stage, we randomly initialize parameters and optimize them via Adam optimizer. The batch size and learning rate are searched in {64, 128, 256, 512} and {0.0005, 0.001, 0.002, 0.005}, respectively. Meanwhile, the dropout is applied to our model, and the dropout rate is test in {0.1, 0.4, 0.5, 0.6}. Then these parameters are carefully fine-tuned to achieve optimal setting. Early stopping with a patience of 100 is applied to avoid overfitting, i.e., we stop training if the validation loss does not decrease for 100 consecutive epochs. For all the baseline methods, we follow the instructions described in their corresponding literatures. We perform a grid search on the values of hyper-parameters and choose an optimal combination of them on each dataset. For the sake of fair comparison, we setup the same embedding dimension of all above algorithms. Moreover, for semi-supervised graph neural network, including GCN, GAT, R-GCN and our SR-HGAT, we split exactly the same training set and testing set to ensure fairness. For random walk based methods include Deepwalk and metapath2vec, we set window size to 5, walk length to 100, walks per node to 40, the number of negative samples to 5. Specifically, in the parameter sensitivity analysis subsection, we vary the number of symmetric relations K in range {1, 2, 3, 4, 5} and the embedding dimension d in range {32, 64, 128, 256, 512, 1024}, and their impact on the performance of the proposed model under different datasets are further investigated. All experiments are conducted on a Ubuntu 14.04 LTS system with two CPUs(Intel Xeon E5*2) and two GPUs(NVIDIA GTX-1080ti*2).

B. NODE CLASSIFICATION
We start by evaluating the quantitative results through the node classification task. Node representations, which were learned on DBLP, Amazon and MovieLens from various baselines, were treated as input features in the node classification task. Relevant classification results were assessed to estimate whether network embedding results are good or not. A logistic regression classifier was chosen as the classifier algorithm. In the course of experiments, the ratio of training data is randomly sampled to 90% from 10%. Node representations in the training set are used to train the logistic regression classifier which is then used on the testing set. The experiment was repeated for 10 times and the average experimental results were reported. Micro-F1 and Macro-F1
were chosen as evaluation metrics of this task. The results are presented in Table 2, 3 and 4.

As demonstrated by experimental results, node representations that are learned from the proposed SR-HGAT model perform best on the classification task. From the score point of view, it is clear that performance of SR-HGAT achieve significant improvements in terms of the Micro-F1 and Macro-F1 over other baselines on all three real-world datasets. We also noted that in Amazon and MovieLens datasets, SR-HGAT improves classification results more significantly than that in DBLP. Because single symmetric relation guided by meta-path A-P-V-P-A is much more important than the rest symmetric relations in DBLP datasets. Compare to heterogeneous graph approaches, homogeneous graph embedding methods, such as Deepwalk, fail to perform well. In addition, in terms of traditional heterogeneous graph embedding methods, HIN2Vec, which can leverage multiple meta-paths performs better than metapath2vec. Generally, GNN-based models that combine the structure and feature information, e.g., GCN and GAT, usually perform better. Moreover, compared to simply averaging over node neighbors of GCN, GAT can weigh the information properly and improve the performance of the learned embedding. The proposed SR-GHAT, which is designed for heterogeneous graph, can encode more complex neighborhood information into latent embedding space and shows its superiority. Furthermore, without nested propagation or two-layer attention mechanism, the performance becomes worse than that of the complete SR-HGAT. Specifically, SR-HGATnl performs worse than other two variants, including SR-HGATnn and SR-HGATnr.

C. NODE CLUSTERING

Node clustering was further carried out to evaluate embedding results learned from above embedding algorithms. In DBLP dataset, we cluster author using the set of author labels. Regarding MovieLens and Amazon datasets, movie clustering and item clustering were conducted, and corresponding label sets selected according to genres of movies and categories of items respectively. Here, the K-Means algorithm was utilized to perform node clustering and the number of clusters K is set to the number of classes. Since the

| Table 2. Node classification results on DBLP dataset. |
|-----------------------------------------------|
| Method             | Micro-F1 | Macro-F1 | Micro-F1 | Macro-F1 |
|-------------------|----------|----------|----------|----------|
| Deepwalk          | 0.7969   | 0.8111   | 0.8279   | 0.8169   |
| Metapath2vec      | 0.8395   | 0.8666   | 0.8840   | 0.8717   |
| HIN2Vec           | 0.8713   | 0.8906   | 0.9019   | 0.9068   |
| GCN               | 0.9086   | 0.9105   | 0.9256   | 0.9272   |
| GAT               | 0.9055   | 0.9157   | 0.9200   | 0.9262   |
| R-GCN             | 0.9082   | 0.9185   | 0.9243   | 0.9250   |
| SR-HGATnn         | 0.9034   | 0.9074   | 0.9261   | 0.9306   |
| SR-HGATnr         | 0.8985   | 0.9120   | 0.9147   | 0.9352   |
| SR-HGAT            | 0.9068   | 0.9222   | 0.9326   | 0.9390   |
|                  | 0.9201   | 0.9247   | 0.9376   | 0.9462   |

| Table 3. Node classification results on MovieLens dataset. |
|-----------------------------------------------|
| Method             | Micro-F1 | Macro-F1 | Micro-F1 | Macro-F1 |
|-------------------|----------|----------|----------|----------|
| Deepwalk          | 0.3877   | 0.4013   | 0.4089   | 0.4135   |
| Metapath2vec      | 0.4199   | 0.4250   | 0.4327   | 0.4383   |
| HIN2Vec           | 0.4833   | 0.5358   | 0.5521   | 0.5630   |
| GCN               | 0.4622   | 0.5224   | 0.5397   | 0.5537   |
| GAT               | 0.4979   | 0.5219   | 0.5335   | 0.5509   |
| R-GCN             | 0.5230   | 0.5564   | 0.5758   | 0.5780   |
| SR-HGATnn         | 0.5507   | 0.5680   | 0.5846   | 0.5870   |
| SR-HGATnr         | 0.5321   | 0.6005   | 0.6170   | 0.6267   |
| SR-HGAT            | 0.5457   | 0.5796   | 0.5934   | 0.5932   |
|                  | 0.5397   | 0.6262   | 0.6428   | 0.6587   |

| Table 4. Node classification results on Amazon dataset. |
|-----------------------------------------------|
| Method             | Micro-F1 | Macro-F1 | Micro-F1 | Macro-F1 |
|-------------------|----------|----------|----------|----------|
| Deepwalk          | 0.5233   | 0.5413   | 0.5482   | 0.5582   |
| Metapath2vec      | 0.5596   | 0.5868   | 0.6048   | 0.6199   |
| HIN2Vec           | 0.6050   | 0.6344   | 0.6543   | 0.6661   |
| GCN               | 0.6619   | 0.6714   | 0.6795   | 0.6908   |
| GAT               | 0.6798   | 0.6888   | 0.6999   | 0.7117   |
| R-GCN             | 0.7274   | 0.7370   | 0.7489   | 0.7675   |
| SR-HGATnn         | 0.7219   | 0.7252   | 0.7275   | 0.7441   |
| SR-HGATnr         | 0.7312   | 0.7391   | 0.7393   | 0.8074   |
| SR-HGAT            | 0.7449   | 0.7877   | 0.8030   | 0.8038   |
|                  | 0.7540   | 0.7908   | 0.8094   | 0.8130   |
performance of K-means is affected by initial centroids, all clustering experiments were repeated 10 times and the average results in the end reported. Clustering performances were evaluated by normalized mutual information (NMI) metrics. The higher the NMI value, the better the clustering performance will be.

Clustering results of three datasets are presented in the table 5. It can be seen that the proposed model performs consistently much better than other baseline methods. Specifically, it can be observed at least 1.5%, 3.9% and 1.8% improvements in NMI values were generated by SR-HGAT compared with other baseline models on clustering author nodes in DBLP, movie nodes in MovieLens and item nodes in Amazon respectively. The SR-HGAT model is able to produce high-quality representations that contain accurate network information specific for features of multi-typed nodes and links in a uniform way. As proved by above experimental results, traditional network embedding method performs as poorly as on node classification task. Moreover, graph neural network based algorithms usually achieve better performance. Besides, without distinguishing the importance of neighbors, GCN cannot perform well. With the guide of multiple symmetric relations, SR-HGAT performs significantly better than GCN and GAT. However, without intra-relation attention or relation-level attention, the performance of SR-HGAT has shown various degrees of degeneration. This demonstrates that by assigning different importance to neighbors and semantics relations, the proposed SR-HGAT can learn a more meaningful and comprehensive representation. Based on the above analysis, it can be found that the proposed SR-HGAT can give a comprehensive description of complex structures and rich semantics behind heterogeneous graph, and achieves a significant improvement.

D. RELATION PREDICTION
A qualified network representation method can not only reconstruct the visible edge in training, but can also predict the edge that should appear but be lost in training data. Considering that the research object in this paper is heterogeneous graph, there is no direct connection between nodes of the same type, so we choose the relation prediction (recommendation) task, which is used to test the ability to predict the interaction links between two types of nodes. In recommendation task, for Amazon dataset, we predict the purchase relationship between user nodes and commodity nodes, and for Movielens dataset, we predict the rating relationship between user nodes and movie nodes. We apply the leave-one-out method for evaluation. For a fair comparison with the baselines, we use the same negative sample set for each (user, item) or (user, movie) pair in the testing set of Amazon and MovieLens respectively for all the methods. After that, we adopt the widely-used evaluation protocols: HR@K and NDCG@K metrics to measure the quality of recommendation. We set k=5, 10, 15, 20, and the average metrics are reported for all users in the test set. The results of recommendation task are reported in Table 6 and 7 with HR and NDCG score respectively.

We have the following observations: (1) Our model consistently outperforms all the baselines, suggesting the effectiveness of SR-HGAT for relation prediction task. In Amazon dataset, it is observed that at least 7.1% and 4.3% improvement in HR@K and NDCG@K values respectively generated by SR-HGAT when compared with other baselines. Moreover, in MovieLens dataset, the average relative improvements over the best performance of baselines are 1.7% and 2.2% respectively. The results also show that our model achieves the best among all baselines regardless whether the datasets are sparse or dense. (2) Meanwhile among these baselines, it is observed that GCN and GAT generally outperform traditional embedding approaches, such as metapath2vec and HIN2Vec, suggesting the power of graph neural network models in representation learning for graph data. It is worth noting that conventional embedding methods do not perform well, since they cannot simultaneously preserve features of links and nodes well. (3) Upon comparison with SR-HGAT, it is found that the performances of SR-HGATnar and SR-HGATnan have various degrees of degeneration. These results are consistent with our assumptions that not all neighbors contribute equally and not all relations have the same importance in learning the final embeddings. This phenomenon also demonstrates that hierarchical two-layer attention is effective for generating meaningful node representations. Moreover, SR-HGATnl exhibits worse performance than other variants in most cases, indicating that our proposed nested propagation of link features has superior capability in capturing in “deep” content of link features.

E. HYPER-PARAMETER SENSITIVITY ANALYSIS
The hyper-parameters play important roles in SR-HGAT, as they determine how the node embeddings will be generated. We conduct experiments to analyze the impacts of several key parameters, involving (1) the dimension of node embeddings, (2) the selection of different symmetric relations and (3) the number of symmetric relations. In these experiments, we investigate a specific parameter by changing its value and fixing the others.

TABLE 5. Quantitative results on the node clustering task.

| Dataset   | DBLP (author) | MovieLens (movie) | Amazon (item) |
|-----------|---------------|-------------------|---------------|
| Deepwalk  | 0.4429        | 0.2095            | 0.6831        |
| Metapath2vec | 0.4524     | 0.2210            | 0.7466        |
| HIN2Vec   | 0.4997        | 0.2353            | 0.7623        |
| GCN       | 0.5458        | 0.2787            | 0.7960        |
| GAT       | 0.5311        | 0.2825            | 0.7996        |
| R-GCN     | 0.5587        | 0.3079            | 0.8716        |
| SR-HGATnl | 0.3437        | 0.3044            | 0.8544        |
| SR-HGATnan| 0.5590        | 0.2999            | 0.8736        |
| SR-HGATnar| 0.5595        | 0.3324            | 0.8598        |
| SR-HGAT   | 0.3681        | 0.3455            | 0.8893        |
TABLE 6. Relation prediction results on MovieLens datasets.

| Method   | HR@k | NDCG@k |
|----------|------|--------|
| k=5      |      |        |
| k=10     |      |        |
| k=15     |      |        |
| k=20     |      |        |
| DeepWalk | 0.0257 | 0.0842 |
| Metapath2vec | 0.1500 | 0.2463 |
| HIN2Vec  | 0.2461 | 0.1490 | 0.5654 | 0.5943 |
| GCN      | 0.3345 | 0.3866 | 0.4597 | 0.5477 |
| GAT      | 0.3886 | 0.5818 | 0.6495 | 0.5319 |
| R-GCN    | 0.4171 | 0.8223 | 0.6949 | 0.6075 |
| SR-HGAT  | 0.4670 | 0.6698 | 0.7449 | 0.7485 |
| SR-HGATD | 0.4431 | 0.8154 | 0.8364 | 0.8385 |
| SR-HGATT | 0.4385 | 0.8209 | 0.8458 | 0.8474 |
| SR-HGAT  | 0.4564 | 0.8399 | 0.8615 | 0.8616 |

TABLE 7. Relation prediction results on Amazon datasets.

| Method   | HR@k | NDCG@k |
|----------|------|--------|
| k=5      |      |        |
| k=10     |      |        |
| k=15     |      |        |
| k=20     |      |        |
| DeepWalk | 0.0521 | 0.1079 |
| Metapath2vec | 0.0863 | 0.1689 |
| HIN2Vec  | 0.0930 | 0.1803 | 0.2590 | 0.3367 |
| GCN      | 0.1057 | 0.2316 | 0.3555 | 0.4686 |
| GAT      | 0.1215 | 0.2662 | 0.4086 | 0.5386 |
| R-GCN    | 0.1300 | 0.2848 | 0.4372 | 0.5763 |
| SR-HGAT  | 0.1640 | 0.5479 | 0.6321 | 0.6434 |
| SR-HGATD | 0.1389 | 0.2069 | 0.3371 | 0.3783 |
| SR-HGATT | 0.1727 | 0.2488 | 0.4474 | 0.5504 |
| SR-HGAT  | 0.1829 | 0.2797 | 0.4978 | 0.6175 |

FIGURE 12. Impact of embedding dimension.

(a) Micro- and Macro-F1.
(b) NMI.
(c) HR@k.
(d) NDCG@k.

FIGURE 13. Impact of different relations on Amazon.

(a) HR@K.
(b) NDCG@K.

FIGURE 14. Impact of different relations on MovieLens.

(a) HR@K.
(b) NDCG@K.

1) DIMENSION OF NODE REPRESENTATIONS

How to determine the optimal number of embedding dimensions is still an open research problem. The dimension of representations (denoted by d) is also a key parameter that controls the complexity and capacity of SR-HGAT. Node clustering and relation prediction performance are analyzed when node representation dimension learned by ST-HGAT varies. We explore the experimental results with various dimensions.

The result is shown in Figure 12. Generally speaking, as embedding dimension is gradually increased, the performance is improved, since a larger dimension could enhance the representation capability. Nevertheless, when dimension is larger than the optimal values, performance starts to drop slowly. The reason is that SR-HGAT needs a suitable dimension to encode the semantics information and a larger dimension may introduce additional redundancies. It can be seen that SR-HGAT can achieve the best performance when the dimension d is set to 128 in DBLP and to 256 in MovieLens and Amazon. After that, the performance
of SR-HGAT starts to degenerate which may be due to overfitting. Therefore, a proper length of embedding dimension was employed to balance the trade-off between performance and complexity.

2) IMPACT OF DIFFERENT SYMMETRIC RELATIONS
As mentioned before, the proposed SR-HGAT can learn the importance of relations for the specific task. To verify the ability of relation-level attention, taking DBLP and MovieLens as examples, we rerun SR-HGAT with individual relation and report the clustering results (NMI) of each symmetric relation guided by meta-path or meta-graph and corresponding attention values in Figure 13 and 14. Obviously, a positive correlation exists between the performance of each symmetric relation and its attention value. For MovieLens dataset, SR-HGAT gives symmetric relation guided by meta-graph U-M-(A,D)-M-U and M-(A,D)-M the largest weight, which means that SR-HGAT considers them as the most critical relation in inferring the relation between user nodes and movie nodes. This makes sense because the users’ preferences and stars (including famous actors and directors) they pursue are highly correlated. Moreover, from the results shown in figure 13, one can observe that the symmetric relation about purchasing (UIU-IUI) can achieve superior performance over other individual relations because it contains more important information which indicate the purchase history and plays a leading role in identifying the user’s preference. It can be seen that relation-level attention can reveal the difference between these symmetric semantic relations and weights them properly.

3) NUMBER OF SYMMETRIC RELATIONS
To investigate whether SR-HGAT can benefit from multiple symmetric relations, we gradually add number of relations $K$ in the range of $[1,2,3,4,5]$ into our model and check the performance changes, while keeping the other parameters the same. Figure 15 shows the experimental results on different datasets. Note that the attention is removed when the number of symmetric relation considered is set to 1.

Figure 15 shows the experimental results in real datasets. It can be observed that generally the performance improves with the incorporation of more symmetric relations. At this point, the power of multiple relations is more prominent. However, it does not always yield the improvement with more relations, and the performance slightly fluctuates. The reason is that some relations guided by meta-paths or meta-graphs may contain noisy or conflict information with existing ones. Moreover, the corresponding performance will remain stable or become slightly worse when relation number further increases. In addition, analysis is also made about how many relations we should consider simultaneously so as to achieve a balance between better performance and lower computational cost. In our experiment, we can see that the optimal number of relations varies with the specific dataset. Specifically, three relations are taken into account simultaneously on Amazon datasets, and four for MovieLens, which are sufficient to meet the demands of most downstream applications. The experiment results also show that our proposed collaborative framework can indeed improve performance by facilitating alignment of different symmetric relations.

VI. CONCLUSION AND FUTURE WORK
In this paper, we propose an end-to-end GAT framework for heterogeneous graph, named SR-HGAT, which applies the attention mechanism to the aggregation of heterogeneous graphs by jointly considering characteristics for both nodes and edges. Our proposed model can capture rich semantics beneath the observed explicit symmetric relations guided by different meta-paths and meta-graphs in heterogeneous graphs. Meanwhile, our proposed SR-HGAT utilizes the nested aggregation mechanism for these link features to calculate the interaction strength of each symmetric relation. As the core of proposed model, a two-layer attention mechanism is adopted to learn the importance of different neighborhood information as well as the weights of different symmetric relations respectively, then these latent semantics are fused automatically to obtain unified embedding specific to graph mining tasks. Extensive experiments on various tasks demonstrate the effectiveness of our proposed model. In the future, we will explore whether SR-HGAT is able to handle heterogeneous graph dynamics, and whether we can conduct efficient and scalable training of our model on Web-scale graph data.

ACKNOWLEDGMENT
Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.
REFERENCES

[1] 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.

[2] 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, 2014, pp. 701–710.

[3] 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.

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

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

[6] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” 2017, arXiv:1710.10903. [Online]. Available: https://arxiv.org/abs/1710.10903

[7] D. Wang, P. Cui, and W. Zhu, “Structural deep network embedding,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 1225–1234.

[8] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Asymmetric transitivity preserving graph embedding,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 1105–1114.

[9] X. Tang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, “Community preserving network embedding,” in Proc. AAAI, 2017, pp. 203–209.

[10] 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, 2015, pp. 1067–1077.

[11] P. Cui, X. Wang, M. Zhang, J. Yan, and Q. Mei, “A survey on network embedding,” in Proc. Trans. Knowl. Data Eng., vol. 31, no. 5, pp. 833–852, May 2019.

[12] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” Knowl.-Based Syst., vol. 151, pp. 78–94, Jul. 2018.

[13] Z. Zhang and J. Han, “Mining heterogeneous information networks: Principles and methodologies,” Synth. Lectures Data Mining Knowl. Discovery, vol. 3, no. 2, pp. 1–159, Jul. 2012, doi: 10.2200/S00433ED1V01Y201207DMK005.

[14] J. Tang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, “Meta-path guided embedding for similarity search in large-scale heterogeneous information networks,” 2016, arXiv:1610.09769. [Online]. Available: http://arxiv.org/abs/1610.09769

[15] Y. Fan, S. Hou, Y. Zhang, Y. Ye, and M. Abdullhayoglu, “Gotcha-sly malware! scorpion a metagraph 2vec based malware detection system,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2018, pp. 253–262.

[16] T.-Y. Fu, W.-C. Lee, and Z. Lei, “HIN2 Vec: Explore meta-paths in heterogeneous information networks for representation learning,” in Proc. ACM Conf. Inf. Knowl. Manage., Nov. 2017, pp. 1797–1806.

[17] T. Chen and Y. Sun, “Task-guided and path-augmented heterogeneous network embedding for author identification,” in Proc. 10th ACM Int. Conf. Web Search Data Mining, 2017, pp. 295–304.

[18] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, “Heterogeneous information network embedding for recommendation,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 2, pp. 357–370, Feb. 2019.

[19] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in Proc. IEEE Int. Joint Conf. Neural Netw., vol. 2, Jul. 2005, pp. 729–734.

[20] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” IEEE Trans. Neural Netw., vol. 20, no. 1, pp. 61–80, Jan. 2009.

[21] M. Henaff, J. Bruna, and Y. Lecun, “Deep convolutional networks on graph-structured data,” arXiv:1506.05163, 2015. [Online]. Available: https://arxiv.org/abs/1506.05163

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3844–3852.

[23] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, “Spectral networks and locally connected networks on graphs,” 2013, arXiv:1312.6203. [Online]. Available: https://arxiv.org/abs/1312.6203

[24] J. Chen, T. Ma, and C. Xiao, “Fastgcn: Fast learning with graph convolutional networks via importance sampling,” 2018, arXiv:1801.10247. [Online]. Available: https://arxiv.org/abs/1801.10247

[25] J. Chen, J. Zhu, and L. Song, “Stochastic training of graph convolutional networks with variance reduction,” 2017, arXiv:1710.10568. [Online]. Available: https://arxiv.org/abs/1710.10568

[26] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, “Disentangled graph convolutional networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 4212–4221.

[27] M. S. Schlichtkrull, T. Kipf, P. Bloem, R. V. Den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in Proc. Eur. Semantic Web Conf., 2018, pp. 593–607.

[28] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous neural graph network,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Jul. 2019, pp. 793–803.

[29] T. Wolf, “HuggingFace’s transformers: State-of-the-art natural language processing,” 2019, arXiv:1910.03771. [Online]. Available: http://arxiv.org/abs/1910.03771

[30] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 5753–5763.

ZHENGHAO ZHANG is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Xidian University, Xi’an. His research interests include graph data mining, machine learning, and heterogeneous information networks.

JIANBIN HUANG received the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi’an. In 2009, he held a postdoctoral research position with the University of Illinois at Urbana–Champaign. He is currently a Professor with the School of Computer Science and Technology, Xidian University. His research interests include data mining and knowledge discovery.

QINGLIN TAN is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Xidian University, Xi’an, China. His research interests include data mining, reinforcement learning, and traffic pattern analysis.

IEEE Access