Temporal Heterogeneous Information Network Embedding via Semantic Evolution

Wei Zhou, Hong Huang, *Member, IEEE*, Ruize Shi, Xiran Song, Xue Lin, Xiao Wang, *Member, IEEE*, and Hai Jin, *Fellow, IEEE*

Abstract—Real-world networks are often heterogeneous and constantly changing over time. Evolution reveals the trend of network development, which is vital for predicting its future state, and network embedding can effectively learn the information from it. Nevertheless, previous works only consider the impact of meta-path instances or node neighbors on the network dynamics but ignore the relationship between them, and hence the hidden semantic information is missed, which will result in performance deterioration. Therefore, we propose a novel temporal heterogeneous information network embedding method (SemE), which abstracts the instance of the meta-path as semantic units and then considers the interaction between them to discover deeper semantic information. Specifically, based on a pre-designed meta-path-guided random walk, SemE first samples semantic units and constructs semantic networks through the star topology found in Ethernet and the interaction between semantic units. Next, to further model semantic evolution, SemE describes the semantic dynamics of nodes with the attention-Hawkes process. Finally, the final embedding is generated by aggregating the structure, semantic and temporal information with the attention mechanism. Experiments on three real-world temporal heterogeneous information networks show that SemE performs better than competitive counterparts.

Index Terms—Temporal heterogeneous information network, network embedding, semantic evolution.

1 Introduction

As we all know, the world is networked, and network embedding has helped us understand the real-world data because of its powerful representation capability, and benefits to various applications, like social recommendation [1], academic search engines [2], and molecular property prediction [3]. However, previous works mainly focus on homogeneous or static networks, but most networks in the real world are heterogeneous and change over time. For example, an academic network contains three types of nodes: author, paper, conference, and various types of relationships, such as an author 'writes' a paper and 'citations' between papers. With the development of time, an author will write a new paper, a paper may be cited by some new papers, and so on. Therefore, to further understand real-world temporal *heterogeneous information networks* (HINs), more and more scholars have focused on temporal HIN embedding.

In general, previous methods for temporal HIN embedding fall into two categories: 1) Snapshot-based methods, such as DyHAN [4], DHNE [5]. First, they divide the dataset into snapshots of different times and leverage the embedding methods for static HINs on each snapshot, then aggregate the embeddings of different timestamps. However, these methods ignore the temporal information between the snapshot. To solve this problem, another category of methods is proposed, 2) Evolutionary dynamics-based methods, such as THINE [6] and HPGE [7]. They sample meta-path instances or node neighbors from the HIN and then simulated the network dynamics by their temporal information and achieved satisfactory results. For the sake of a concise statement, we refer to meta-path instances, node neighbors, and other units that consider semantic information collectively as *semantic units*.

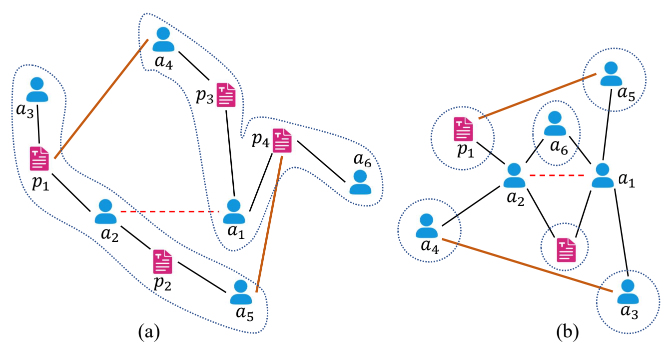
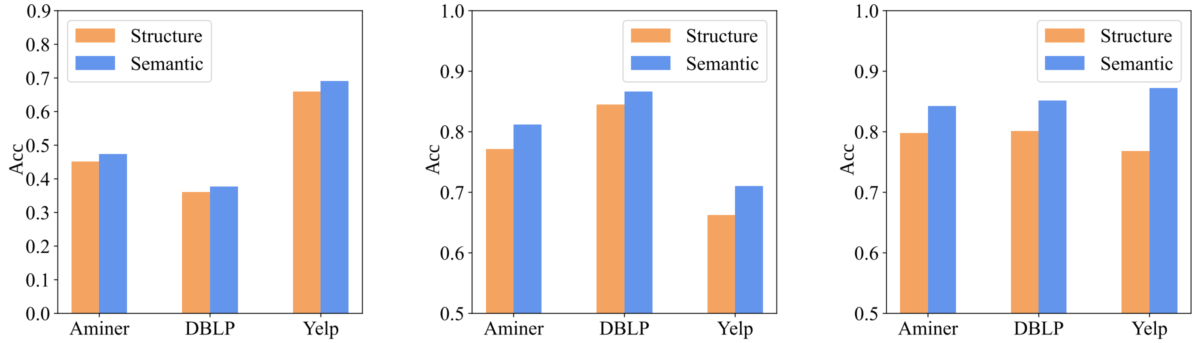


Fig. 1. (a) A toy example of THINE. (b) A toy example of HPGE. The blue dashed box shows the corresponding semantic unit, the red dashed line indicates the target to be predicted, and the solid orange line indicates the ignored information.

However, previous works only consider the influence of semantic units on the network dynamics without further consideration of the relationships between semantic units. For example, THINE and HPGE are recently representative meta-path instance based methods and neighborhood based methods, respectively. As shown in Figure 1(a), to predict whether a_1 and a_2 will form the edge, THINE randomly samples the meta-paths instance related to a_1 or a_2, such as $a_3p_1a_2p_2a_5$ and $a_4p_3a_1p_4a_6$, and uses them to calculate

- *Wei Zhou, Hong Huang, Ruize Shi, Xiran Song, Xue Lin, Hai Jin are with the National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, China. E-mail: {weizhou2021, honghuang, rzshi, xiransong, xuelin, hjin}@hust.edu.cn*
- *Xiao Wang is with the School of Software, Beihang University, China. E-mail: xiao_wang@buaa.edu.cn*
- *Hong Huang is the corresponding author.*



(a) Node Classification (b) Link Prediction (c) Temporal link prediction

Fig. 2. We compare the importance of semantic and structure information on three datasets, Aminer, DBLP, and Yelp. To capture the structure information, we abstract the whole network into a homogeneous network by ignoring relationships and types of nodes and then use ProNE [8] to learn the embedding of all nodes. While to obtain semantic information, we generate semantic units by meta-path guided random walk and construct semantic networks. Then we use ProNE to generate embedding for semantic units and use the average of embedding corresponding semantic units as the embedding of nodes. Please see Section 3 for more details. Finally, we apply structure and semantic embedding to the node classification, link prediction, and temporal link prediction and report their accuracy.

the generation probability of edge a_1a_2. However, THINE ignores the relationship between meta-path instances, i.e., edges a_4p_1 and a_5p_4, which causes THINE to lose much information to help predict the network structure. Similarly, in Figure 1(b), the HPGE samples the neighbors of a_1 and a_2 but does not consider the relationships between neighbors. If HPGE considers the interaction between semantic units, it can capture more information to describe the structure of the network better.

Actually, the relationship between semantic units is essential for learning the embedding of the temporal HIN. On the one hand, *semantic information is more important than structure information*. In practical experiments, we compare the importance of semantic and structure information by three tasks on three HINs, and the experimental results are shown in Figure 2. We can see that semantic information is more valuable than structure information in various tasks. Moreover, in definition, semantic information is obtained by particular network structures that express clear and specific semantics. In HINs, structural information contains a lot of noise that is irrelevant to the downstream task, while carefully selected semantic information can eliminate it and thus outperform structural information. Therefore, the richer semantic information obtained by capturing the interaction between semantic units can help the model learn the representation of nodes in HINs better. On the other hand, *semantic information is helpful to describe the structure of the network*. According to [9], [10], the paths between two nodes help predict the link between them. Similarly, by considering the interactions between semantic units, we can find many hidden paths between two nodes, which is helpful in describbng the structure of the temporal HIN. Thus, hidden semantic information is beneficial for embedding learning.

There are many semantic units in the HIN, and considering the relationship between them will result in a large memory and time overhead. But if we only consider the interaction between some of the semantic units such as those related to edge a_1a_2 in Figure 1, there will be a loss of information. So it is challenging to consider the relationship between semantic units efficiently with as little loss of information as possible.

To solve this problem, we propose a novel model — **SemE**. For each pre-designed meta-path, we abstract the corresponding semantic units as nodes. To efficiently consider the global interaction between semantic units, based on the star topology found in Ethernet and the relationship between semantic units, we construct a sparse network—semantic network, which can be regarded as a special homogeneous network because it contains only one kind of semantic information. In addition, since semantic information is more important than structure information in HINs, we propose semantic evolution based on semantic networks, which describes the semantic dynamics of nodes. To enhance the scalability of the model, we only extract semantic information in the semantic network and then use the attention-Hawkes process to model temporal information. In this way, all semantic and temporal information is well preserved. Finally, we aggregate the structure, semantic, and temporal information to obtain the final embeddings by the attention mechanism [4]. Experiments prove that SemE can effectively describe the semantic evolution pattern of nodes and perform better than competitive counterparts.

The contributions of our work are as follows:

- To the best of our knowledge, it is the first attempt to consider the relationship between semantic units and propose the semantic evolution, which helps us effectively model the dynamics of the temporal HIN and preserve its dynamics and heterogeneity.
- We propose a novel model that can effectively capture information from temporal HINs by modeling semantic evolution with the attention-Hawkes process.
- With the semantic network, we reduce the complexity of the network and migrate the embedding problem on the HIN to the homogeneous network, which significantly accelerates the speed of node embedding learning.
- Various experiments on three real-world temporal HINs

demonstrate that SemE outperforms competitive counterparts.

2 Problem Definitions and Preliminaries

2.1 Definitions

Definition 1: Temporal HIN. A Temporal HIN is defined as $G = (V, E, T, \Phi)$, where V and E denote the set of nodes and edges, T represents the set of timestamps, and $\Phi = \{\Phi_V, \Phi_E\}$ indicates the sets of node and edge types. In a temporal HIN, each node v or edge e has a corresponding class, obtained by type mapping function $\phi : V \to \Phi_V$ or $\varphi : E \to \Phi_E$, respectively, where $|\Phi_V| + |\Phi_E| > 2$. Furthermore, between two successive timestamps, t - 1 and t, the temporal HIN G satisfies that $V^{t-1} \neq V^t$ or $E^{t-1} \neq E^t$, where V^{t-1} and V^t represent the set of nodes at timestamp t - 1 and t respectively, while E^{t-1} and E^t denote the set of edges at timestamp t - 1 and t respectively. Note that Φ_V and Φ_E do not change at all timestamps.

Definition 2: Meta-path. A meta-path m is defined as $\mathit{\Phi}_{v_1} \xrightarrow{\mathit{\Phi}_{e_1}} \mathit{\Phi}_{v_2} \xrightarrow{\mathit{\Phi}_{e_2}} \cdots \xrightarrow{\mathit{\Phi}_{e_l}} \mathit{\Phi}_{v_{l+1}}$, where node type $\mathit{\Phi}_{v_i} \in \Phi_V$ and edge type $\mathit{\Phi}_{e_i} \in \mathit{\Phi}_E$. It can be abbreviated as $\mathit{\Phi}_{v_1} \mathit{\Phi}_{v_2} \cdots \mathit{\Phi}_{v_{l+1}}$, which describes a complex composite relation between v_1 and v_{l+1}. A path instance of meta-path m is treated as a node sequence $v_1 v_2 \cdots v_{l+1}$, which follows meta-path m.

Definition 3: Semantic Unit. In fact, node neighbors are included in the meta-path instances or the deep semantic relationship between meta-path instances. For example, in Figure 1(b), the edge a_5p_1 ignored by node neighbors is included in the meta-path instance $a_2p_1a_5$. While the edge a_3a_4 ignored by node neighbors will be considered through the relationship between meta-path instances $a_6a_2a_4$ and $a_6a_1a_3$. Therefore, in this paper, given a temporal HIN, we abstract the instance of meta-path as a semantic unit. In order to ensure the integrity and unambiguity of semantic information, the length of semantic units must meet $\ell = (|m| - 1) * q + 1$, where $|m|$ is the length of the meta-path m, and q is a positive integer which indicates the number of semantic information contained in the semantic unit. For example, when $q = 2$, with the meta-path APA, we will get a semantic unit such as $a_1p_1a_2p_2a_3$, which contains two semantic information, $a_1p_1a_2$ and $a_2p_2a_3$. However, too long a semantic unit may bring noise because the semantics far away may not be related to the first node. Finally, we use the latest time of the edge as the time of the semantic unit. For example, a semantic unit $p_1 \xrightarrow{t_1} c_1 \xrightarrow{t_2} p_2$, $t_2 > t_1$, this semantic unit occurs only when the time reaches t_2.

Definition 4: Semantic Network. Given a network G, all nodes in the network are the semantic units related to meta-path m. Then the network G can be called the semantic network of meta-path m. We build this network by the correlation between semantic units. We will discuss the detailed construction process in Section 3.2.

Problem. Temporal HIN Embedding. Given a temporal HIN $G = (V, E, T, \Phi)$, where V and E denote the set of nodes and edges, T represents the set of time, and $\Phi = \{\Phi_V, \Phi_E\}$ indicates the sets of node and edge types. The goal is to obtain a mapping function $f : V \to \mathcal{R}^d$, where d is the embedding dimension and $d << |V|$. f retains the structure features of the network and captures semantic information and temporal information by modeling the semantic evolution pattern of nodes.

2.2 Preliminaries: Star and Mesh Topology

In the real world, there are multiple different topologies in Ethernet. Inspired by the mesh and star topologies found in Ethernet, we build the semantic network to efficiently capture the global interactions between semantic units.

Mesh topology is a common topology in Ethernet, whose all nodes are connected to each other. The mesh topology in small communities can increase reliability and information delivery efficiency. Still, when the community is enormous, the network will be too dense and have high costs. Similar to the mesh topology, star topology is also a common topology in Ethernet. All nodes are individually connected to a central connection point, like a hub or a switch. Star topology has a more straightforward structure and higher scalability than mesh topology in the large community.

3 Our Model

The framework of SemE is shown in Figure 3. Firstly, we capture the semantic units in the network by random walk based on meta-paths. Secondly, with the help of the star topology found in Ethernet, we construct a semantic network consisting of only semantic units and a series of central nodes to extract the global interaction between semantic units. Then, we generate embedding for all nodes in the semantic network by the unsupervised network embedding method. With the attention-Hawkes process, we converge these semantic units in a temporal order to simulate the semantic evolution process in the network. Finally, by fusing the different semantic evolution processes and the structure information of the network with the attention mechanism, we obtain the final temporal HIN embedding and apply it to different downstream tasks.

3.1 Semantic Unit Generation

Semantic units can effectively describe the semantics of nodes. For example, when the edges links to node i increase, node i will express richer semantics. With the increase of edges, there will be many semantic units, which can accurately describe the semantic changes of node i, and SemE obtains them by the meta-path-based random walk. Formally, in a temporal HIN, S_i^m which represents the semantic embedding of node i related to meta-path m can be formulated as

$$S_i^m = g(\{e|v \in V_i^m\}), \tag{1}$$

where V_i^m indicates the set of semantic units related to meta-path m of node i, v is a semantic unit, and e is its embedding. g is a function that leverages the semantic units set related to node i to obtain S_i^m, and we will detail it in Section 3.3.

Here we show how to obtain semantic units by random walk based on meta-path. With a meta-path $m =$

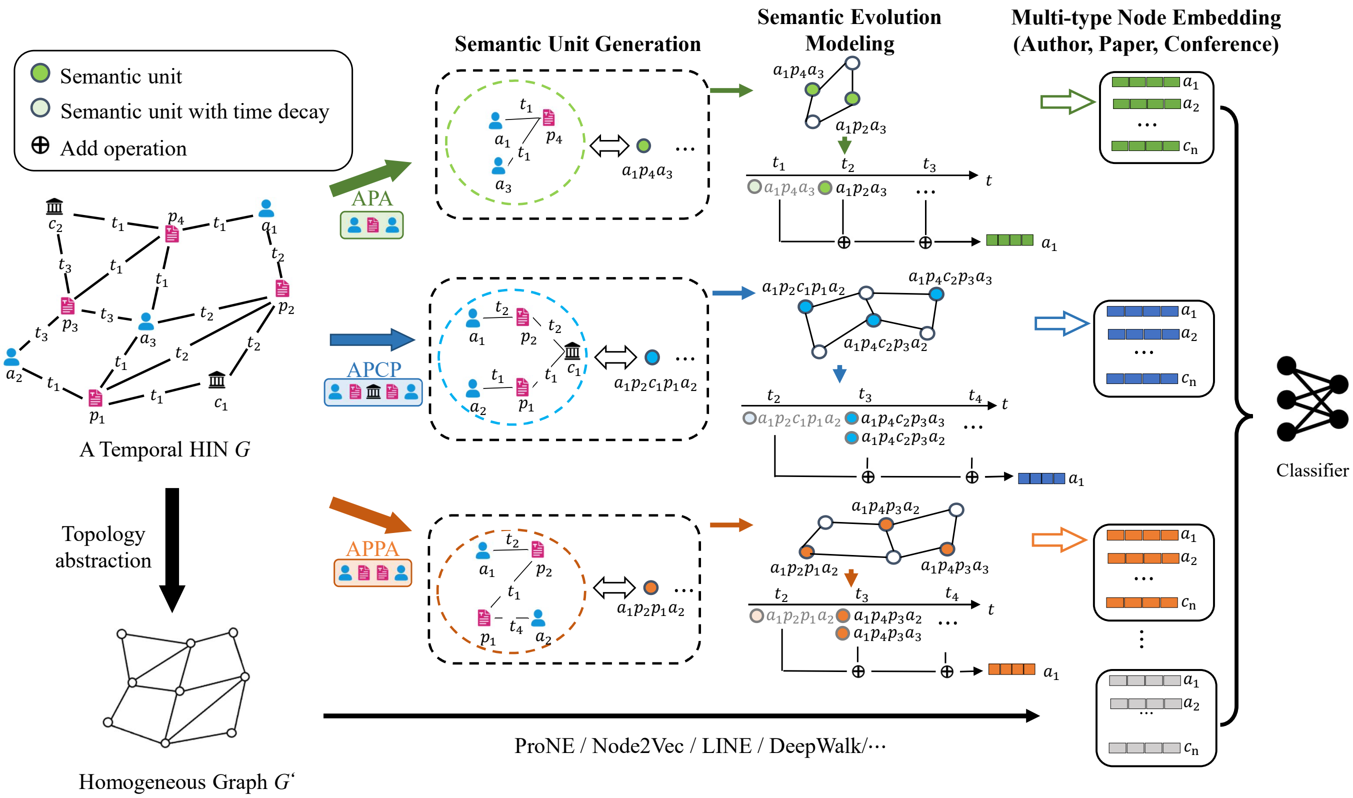


Fig. 3. The overall architecture of SemE.

$\Phi_{v_1} \xrightarrow{\Phi_{e_1}} \Phi_{v_2} \xrightarrow{\Phi_{e_2}} \cdots \xrightarrow{\Phi_{e_l}} \Phi_{v_{l+1}}$, the transition probability at step i is defined as follows

$$p(v_{i+1}|v_i, m) = \begin{cases} \frac{1}{|N_{t_{v_{i+1}}}(v_i)|}, & (v_i, v_{i+1}) \in E, \phi(v_{i+1}) = \Phi_{v_{j+1}} \\ 0, & (v_i, v_{i+1}) \in E, \phi(v_{i+1}) \neq \Phi_{v_{j+1}} \\ 0, & (v_i, v_{i+1}) \notin E \end{cases} \tag{2}$$

where $\phi(v_i) = \Phi_{v_j}$, $N_{t_{v_{i+1}}}(v_i)$ indicates the neighbors of node v_i, and their type is $\Phi_{v_{j+1}}$.

Nonetheless, generating all semantic units of a temporal HIN will take a lot of time and space. To solve this problem, SemE adopts a sampling strategy. Simply, SemE samples a temporal HIN K times, and each time, n semantic units with length l are generated for a node. Note that there should not be the same node in a semantic unit. For example, the information of semantic unit $a_1p_1a_1$ is already contained in the $a_1p_1a_2$. By deleting these duplicate semantics, we can obtain more semantic information with less sampling.

3.2 Semantic Network Construction

After sampling, we construct the semantic network of specific meta-path by semantic units. If any two semantic units include the same nodes, they will establish a connection. For example, $a_1p_1a_2$ and $a_1p_2a_3$ contain the same author node a_1, and we build a correlation between them. Therefore, the relationship between any two semantic units can be expressed as

$$\mathcal{W}(v_i^m, u_j^m) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3}$$

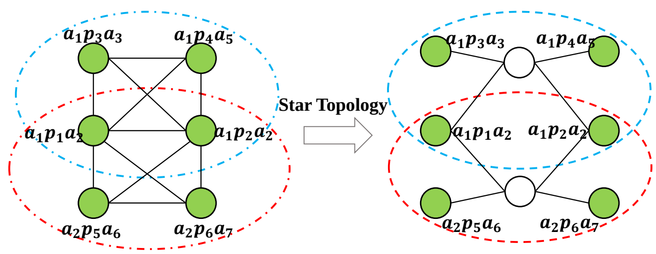
where $v_i^m \in V_i^m$ and $u_j^m \in V_j^m$. In this way, all semantic units in V_i^m connect with each other, forming a fully connected network called the community of node i.

Obviously, the community of node i will result in a large amount of memory and time overhead. Fortunately, the star topology of Ethernet inspires us. In real-world communities, residential neighborhoods use switches to transmit information to each user. Therefore, we add a central node to the community of node i as a switch. As a result, all nodes in the community are no longer connected to each other but to the central node. Therefore, we optimize the semantic mesh network to the semantic star network, which is demonstrated in Figure 4, and Equation (3) is rewritten as

$$\mathcal{W}(v_i^m, c_j^m) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{4}$$

where c_j^m is the central node to the community of node j. This method significantly simplifies the structure of the network and reduces the cost. Note that a temporal HIN has many meta-paths, so there will be many semantic networks.

Furthermore, all semantic units are related to meta-path m and express the same kind of semantics for a semantic network. In addition, the central node is only the hub connecting semantic units and does not contain any semantics. Therefore, we regard the semantic network as a particular homogeneous network and embed semantic units with an unsupervised or self-supervised embedding method , such as ProNE [8]. Relying on well-established embedding methods on homogeneous networks, we are able to learn the embedding of nodes in the semantic network quickly and



(a)Semantic mesh network (b) Semantic star network

Fig. 4. Semantic network construction. Inspired by Ethernet, we transform a semantic mesh network into a semantic star network.

efficiently. In addition, in Section 4.8, we test the influence of different embedding methods for homogeneous networks on the performance of SemE.

3.3 Semantic Evolution Modeling

Semantic information is composed of multiple semantic unit types with different feature spaces. Therefore, before modeling the semantic evolution patterns, we first map them to the same feature space using the projection matrix $\mathcal{M}$ for a specific semantic type to facilitate the merging of different types of semantics later. The details are as follows:

$$H^m = \mathcal{M}^m E^m, \tag{5}$$

where E^m represents the embedding of all semantic units with meta-path m.

With the development of time, the number of edges of the connecting node i will change, and many semantic units will appear, which will affect the semantics of node i. Therefore, we add these semantic units to the set V_i^m. In addition, the earlier the semantic units occur, the less impact it has on the present. Therefore, we model the semantic evolution of nodes by the Hawkes process [11] and attention mechanism [12]. Specifically, Equation (1) is rewritten as

$$S_i^m = \sigma\left(\sum_{v_j \in V_i^m \cup c_i^m} \alpha_{i,j} h_j \mu(t - t_i)\right), \tag{6}$$

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(a^T[Wh_i || Wh_j]))}{\sum_{v_k \in V_i^m \cup c_i^m} \exp(\text{LeakyReLU}(a^T[Wh_i || Wh_k]))}, \tag{7}$$

where h_i is the embedding of central node c_i^m, and $\mu(t - t_j)$ is an excitation function that decays with time. Thus, it represents the influence of semantic units before time t. In SemE, we use the embedding of the central node as the basic semantics of node i, because the central node is the neighbor of all semantic units of node i. It contains the shared information of these semantic units, and no matter how the semantics of node i change, this information should be included.

For excitation function $\mu(t - t_j)$, its standard form is exponential decay so that we can express it as

$$\mu(t - t_j) = e^{-\beta(t - t_j)}, \tag{8}$$

where β is a trainable parameter and is related to nodes [13].

However, the impact of historical events on the present should be divided into two parts. On the one hand, the impact of events at different times is different. On the other hand, the impact of different events is different. Therefore, we define β as

$$\beta = \theta_1 + \theta_2, \tag{9}$$

where θ_1 is related to nodes and θ_2 is only related to time.

3.4 HIN Embedding

Although we obtain S^m which represents the embedding of all nodes based on meta-path m, there are many meta-paths in the temporal HIN. So we need to repeat the above steps with different meta-paths, and get a set of node embedding $S = \{S^{m_i} | m_i \in M\}$, where M is a set of meta-paths. What is more, the node in the temporal HIN should contain semantic information, temporal information, and structure information. Therefore, the embedding of nodes in the temporal HIN can be expressed as

$$H = \sum_{H_i \in S \cup H_{st}} \gamma_i H_i, \tag{10}$$

where H_{st} represents the structure embedding of nodes with the projection matrix $\mathcal{M}_{st}$, which is obtained on the topology of the temporal HIN by using methods such as ProNE.

The Temporal HIN contains multiple semantic information, and their contribution to different downstream tasks is different. Therefore, to learn the importance of different meta-paths and the structure, the importance of each is calculated as:

$$w_i = \frac{1}{|V|} \sum_{h_i \in H_i} \mathbf{a}^T \tanh(Wh_i + b), \tag{11}$$

where $\mathbf{a}$ is a trainable vector, W is the weight matrix, and b is the bias. Then, γ_i denotes the weight of meta-paths and structure, and we can obtain it by normalizing w_i with the softmax function.

$$\gamma_i = \frac{\exp(w_i)}{\sum_{j=1}^{|M|+1} \exp(w_j)}. \tag{12}$$

Finally, H will adapt to different downstream tasks by an mlp. In addition, SemE is an end-to-end model, so that we use cross-entropy as the loss function and optimize all trainable parameters by *Adaptive Moment Estimation* (Adam).

3.5 Complexity Analysis

Our model can be divided into three parts. Suppose only one meta-path m is considered. In the first part, SemE samples $K \times n$ times to generate semantic units with length $\ell = (|m| - 1) * q + 1$, and its time complexity is $O(Knl|V|)$. For the second part, the time complexity of constructing the semantic network by star topology is also $O(Knl|V|)$. Then, we can regard the whole evolutionary modeling process as a GAT [12] layer, which propagates the information of semantic units in the community of node i and updates the representation of the node. There are $Kn|V| + |V|$ nodes and $Knl|V|$ edges in a star semantic network. Therefore, the time complexity of modeling semantic evolution patterns is $O((Kn|V| + |V|)d_h^2 + Knl|V|d_h)$, where d_h is the dimension of the hidden layer. Finally, the overall time complexity is $O((Kn|V| + |V|)d_h^2 + Knl|V|(d_h + 2))$.

TABLE 1
Statistics of three real-world datasets.

Dataset	Node(#Node)	Edge(#Edge)	TS*	Meta-path
Aminer	A(10206) P(10457) C(2584)	A-P(41687) P-P(41678) P-C(10457)	10	APA APCPA APPA
DBLP	A(22662) P(22670) C(2938)	A-P(122984) P-P(122961) P-C(22670)	15	APA APCPA APPA
Yelp	S(5) U(24586) B(800)	B-S(28000) U-B(28000)	15	UBU UBSBU BUB BSB

* TS is short for Timestamp.

TABLE 2
Statistics of 12 baselines.

	Homogeneous	Heterogeneous
Static	Deepwalk [15] LINE [17] Node2Vec [19] ProNE [8]	Metapath2Vec [16] StHNE [18] MAGNN [20] ** HGT [21] **
Temporal	DySAT [22] HTNE [23] MTNE [13]	DHNE [5] DyHNE [18] HDGNN [24] ** THINE [6] HPGE [7] TGT [25] ** KHGT [26] **

** indicates deep methods.

4 Experiments and Analysis

4.1 Experimental Setup

Datasets and Tasks. We evaluate SemE and other baselines on three popular datasets, Aminer [14], DBLP[1], and Yelp[2], whose details are listed in Table 1.

- **Aminer** is an academic network that consists of three types of nodes: **A**uthors, **P**apers, and **C**onferences. For the node classification task, authors are divided into three classes, while for the link prediction task and temporal link prediction task, it is to predict whether the two authors are collaborators.
- **DBLP** is an academic network related to the field of computer science, and it contains the same types of nodes as Aminer. For the node classification task, authors are divided into four classes, while for the link prediction task and temporal link prediction task, we have the same settings with Aminer.
- **Yelp** is a review network that consists of three types of nodes: **U**sers, **B**usinesses, and **S**tars. For the node classification, the task is to predict the business label, so we present the meta-paths, including BSB and BUB. While for the link prediction and temporal link prediction, the task is to predict whether the two users are friends, so we are interested in meta-paths UBU and $UBSBU$.

Baselines. We compare our proposed SemE with 18 baselines, including 7 homogeneous network methods and 11 HIN models, which are demonstrated in Table 2. On the datasets mentioned above, we used the node classification and link prediction tasks to evaluate the performance of all models except HPGE, TGT and KHGT. For the temporal link prediction task, we evaluated the performance of all models. In addition, for all baseline models, we use the code provided by their authors.

Parameter Settings. We evaluate SemE and other baselines on a server with Intel Xeon CPU E5-2680, Tesla V100 GPUs, and 250GB of Memory. The experimental environment of software is Ubuntu 18.04 with CUDA 10.2. In the experiment, we use ProNE [8] to initialize embedding for semantic units and structural information of nodes because it runs faster and performs better than other unsupervised or self-supervised embedding methods on the homogeneous network, which we will prove in Section 4.8. We train SemE with the learning rate of 0.0001 and the Adam optimizer for all tasks. In addition, we set the epoch as 100, the weight decay of the Adam is set as 10e-5, and the dropout is 0.5.

In the node classification experiments, for Aminer and DBLP, we set sampling times K as 8, sampling number n is 8, and the semantic information number q as 1. In contrast, sampling times K, sampling number n, and the semantic information number q are set to 8, 64, 1 for Yelp. In addition, for all datasets, the batch size is set to 256.

In the link prediction experiments, for all datasets, the batch size is 64, sampling times K is 8, sampling number n is 8, and the semantic information number q is 1. In addition, we set the same parameters as link prediction experiments in temporal link prediction experiments.

For all non-end-to-end models, we set embedding dimension d as 100 and use a logistic regression model to predict the results. In contrast, SemE and other end-to-end baselines use the downstream tasks to optimize and directly output the prediction results. For supervised methods, we use one-hot vectors as the initial features of nodes. The meta-paths are shown in Tabel 1. For all baselines, the remaining parameters use the default settings. In addition, we take the best performance on the test set as a result for all models. In addition, to ensure the reliability of experiments, we repeat each task ten times. After that, the average value is taken as the final result.

In particular, node classification and link prediction are not temporal tasks, so we set $\mu(t - t_i) = 1$ for all semantic units in SemE.

4.2 Node Classification

In the node classification experiment, for Aminer and DBLP, the task is to predict the label of authors. While for Yelp, we predict the label of businesses. For all datasets, we set the dimension of the hidden layer as 256. We report test accuracy (macro-F1 and micro-F1) for all datasets. In addition, we set the ratio of the training set as 60%, 80%, and the others as the test set. The final experiment results are demonstrated in Table 3.

From Table 3, we can see that on the non-temporal task, Deepwalk outperforms the vast majority of baselines. This is because classic methods are highly robust and demonstrate outstanding performance across different datasets and

1. https://dblp.org
2. https://www.yelp.com/dataset

downstream tasks. In contrast, many baseline models have poorer robustness and exhibit significant performance differences depending on the dataset or downstream task. We can also observe similar results in two other tasks. In addition, we can find that the results of SemE on all datasets are better than other methods. On the one hand, it shows that semantic units can effectively capture semantic information in the HIN. Using semantic units, our model SemE can effectively describe the properties of nodes. On the other hand, the star semantic network can effectively describe the relationship between semantic units. However, the performance of SemE is only marginally improved compared to the latest baseline. To prove that the improvement is not accidental, we report p-values on all datasets. The results prove that SemE is significantly better than all baseline models. Since THINE include unsupervised tasks for describing the evolution of the network structure, they still consider temporal information in the network for non-temporal tasks. Therefore, they perform better than MAGNN, StHNE and HGT, which only consider heterogeneous information. SemE does not consider temporal information in non-temporal tasks, so the performance improvement is slight.

4.3 Link Prediction

Link prediction is essentially a binary classification of edges. Therefore, we set the label of the actual edge as 1, while the label of the non-existent edge is 0. The more similar nodes are, the more likely they are to link in the network. The similarity between nodes is used to predict the existence of edges, so we define the embedding of edges as $e_{i,j} = |e_i - e_j|$, where e_i and e_j are the embeddings of nodes i and j, respectively. For Aminer and DBLP, the task here is to predict whether the two authors are collaborators. This information is hidden in meta-path APA, so we randomly hide 25% A-P edges and use the remaining edges to predict the co-author relationship. While for Yelp, we hide 25% U-B edges randomly and predict whether the two users are friends. In addition, the number of positive and negative samples for all datasets is 25000, and we randomly take 20% of them as the training set.

From the final experiment results listed in Table 4, we can first see that on the link prediction task, temporal models generally perform well, which indicates that simulating the structure evolution helps predict the structural dynamics of the network. Secondly, based on the experimental results and p-value, we can find that the performance of SemE is significantly better than the baseline model. Especially on the Yelp dataset, SemE can effectively predict the link relationship between users. It shows that semantic units can accurately describe the attributes of nodes and effectively capture the relationship between nodes. In addition, considering the relationship between semantic units can effectively capture the deeper semantic information, which is helpful for the final performance of the model. Finally, the performance improvement of SemE on the link prediction task is also modest for the same reason as analyzed in the node classification experiments.

4.4 Temporal Link Prediction

In the temporal link prediction experiment, we also randomly hide 25% edges, just like link prediction. For Aminer, the task is to predict whether the two authors will cooperate in 1988 by the information before 1988. For DBLP, we use the information before 2004 to predict whether the two authors will cooperate in 2004. While for Yelp, the task here is to predict whether the two users are friends in 2020. In addition, the number of positive samples for all datasets is 25000, while the number of negative samples is 12500, and we randomly take 20% of them as the training set.

Firstly, from the final experiment results listed in Table 5, the temporal methods generally perform better, which indicates that temporal information helps predict link dynamics in the network. Secondly, the performance of THINE and HPGE is better than MTNE, which indicates that the measures (meta-path and relational aggregation) taken by both for HINs are effective in capturing semantic information. Thirdly, we can find that on the three datasets, SemE performs better than other baselines. On the one hand, this illustrates that SemE can more effectively capture the semantic information in the network by taking into account the semantic units and their relationships. Additionally, SemE is capable of accurately capturing temporal information by describing semantic evolution. On the other hand, this suggests that in HINs, semantic evolution holds more significance than structural evolution, and modeling semantic evolution can better capture semantic information and temporal dynamics. Finally, we can see that HGT, TGT and KHGT perform worse than SemE because the characteristics of the dataset prevent them from performing optimally. On the one hand, HGT only uses temporal information as a feature of nodes, but the author nodes do not have exact temporal information. On the other hand, for recommendation methods, the dataset they deal with is a bipartite graph containing multiple relationships. However, in heterogeneous graphs, many relations are not directly related to the target node, which causes these methods to lose information by not covering all the relations.

4.5 Parameter Analysis

We study how parameters influence the performance of the proposed SemE in this part. We evaluate sample times K, sample number n, and semantic information number q. Because we obtain similar conclusions on all tasks of which the temporal link prediction is the most important, we only show the impact of the parameters on it. We report the AUC, F1, and accuracy for temporal link prediction on all datasets with the training set ratio as 20%.

Sample times K. When K changes, n and q are fixed to 8 and 1. The final experiment results are listed in Figure 5(a). From it, we can see that with the increase of sampling times K, the performance of SemE is improved since SemE can sample more semantic units from the temporal HIN. Moreover, it shows that the more semantic units, the more accurate the description of node attributes. In addition, the increasing trend of all metrics is gradually decreasing because the number of semantic units in the network is limited. Therefore, we cannot get better results by increasing the parameters indefinitely.

Sample number n. When n changes, K and q are fixed to 8 and 1, and the final experiment results are demonstrated in the Figure 5(b). From it, we can see that with the increase

TABLE 3
Performance on node classification task. Bolded font indicates the best result, underlined indicates the suboptimal result.

Methods	Aminer		DBLP		Yelp	
	micro-F1 60% / 80%	macro-F1 60% / 80%	micro-F1 60% / 80%	macro-F1 60% / 80%	micro-F1 60% / 80%	macro-F1 60% / 80%
Deepwalk	0.4450 / 0.4467	0.4448 / 0.4473	0.3425 / 0.3600	0.3421 / 0.3549	0.6438 / 0.6593	0.6437 / 0.6591
LINE	0.3633 / 0.3833	0.3496 / 0.3621	0.2975 / 0.3038	0.2973 / 0.3036	0.5643 / 0.6175	0.5530 / 0.5973
Node2Vec	0.4484 / 0.4473	0.4501 / 0.4492	0.3462 / 0.3595	0.3483 / 0.3567	0.6396 / 0.6514	0.6372 / 0.6499
ProNE	0.4613 / 0.4637	0.4591 / 0.4665	0.3596 / 0.3723	0.3611 / 0.3713	0.6718 / 0.6787	0.6696 / 0.6801
DySAT	0.4300 / 0.4266	0.4500 / 0.4474	0.3150 / 0.3225	0.3126 / 0.3230	0.6844 / 0.6999	0.6837 / 0.6970
HTNE	0.3750 / 0.4083	0.3581 / 0.3908	0.3075 / 0.3263	0.3073 / 0.3268	0.6150 / 0.6500	0.6122 / 0.6475
MTNE	0.4133 / 0.4508	0.4051 / 0.4567	0.3125 / 0.3388	0.3104 / 0.3389	0.6500 / 0.6875	0.6499 / 0.6867
Metapath2vec	0.3817 / 0.4033	0.3812 / 0.4013	0.3050 / 0.3450	0.3047 / 0.3428	0.5087 / 0.6088	0.4951 / 0.5980
StHNE	0.3500 / 0.4000	0.2550 / 0.3422	0.2200 / 0.1925	0.1903 / 0.1844	0.6781 / 0.6625	0.6704 / 0.6599
MAGNN	0.3667 / 0.3967	0.3618 / 0.3961	0.2450 / 0.2525	0.2385 / 0.2543	0.4656 / 0.4875	0.4654 / 0.4871
HGT	0.4777 / 0.4802	0.4753 / 0.4811	0.3861 / 0.3899	0.3876 / 0.3921	0.6895 / 0.7033	0.6931 / 0.7030
DHNE	0.4362 / 0.3964	0.4119 / 0.3597	0.3750 / 0.3975	0.3690 / 0.3835	0.6094 / 0.7000	0.6080 / 0.6998
DyHNE	0.3633 / 0.4200	0.3530 / 0.4131	0.2213 / 0.1950	0.906 / 0.899	0.6844 / 0.6813	0.6774 / 0.6776
HDGNN	0.4167 / 0.4483	0.4157 / 0.4461	0.3725 / 0.3750	0.3689 / 0.3685	0.6218 / 0.6187	0.6210 / 0.6180
THINE	0.4833 / 0.4867	0.4803 / 0.4858	0.3850 / 0.3989	0.3743 / 0.3964	0.7037 / 0.7125	0.7023 / 0.7124
SemE	**0.4883 / 0.5020**	**0.4864 / 0.4973**	**0.3955 / 0.4085**	**0.3935 / 0.4029**	**0.7121 / 0.7213**	**0.7105 / 0.7198**
p	1.9e−3 / 7.6e−4	3.0e−3 / 1.7e−5	1.4e−3 / 2.1e−2	1.3e−2 / 4.7e−2	1.1e−2 / 5.1e−3	3.4e−2 / 3.6e−4

TABLE 4
Performance on link prediction task.

Methods	Aminer			DBLP			Yelp		
	auc	f1	acc	auc	f1	acc	auc	f1	acc
Deepwalk	77.07%	71.19%	71.09%	85.59%	81.28%	81.28%	50.32%	63.28%	52.78%
LINE	68.49%	64.79%	63.54%	75.11%	71.24%	70.09%	58.30%	57.56%	55.57%
Node2Vec	79.11%	75.38%	74.92%	83.54%	80.96%	80.08%	60.31%	59.17%	59.27%
ProNe	82.43%	80.09%	79.86%	87.44%	82.65%	81.97%	72.79%	67.36%	63.03%
DySAT	80.25%	67.91%	66.14%	82.16%	75.61%	75.03%	53.67%	54.27%	52.88%
HTNE	76.53%	73.18%	72.55%	90.77%	83.16%	82.87%	65.27%	64.77%	60.65%
MTNE	82.78%	75.23%	74.72%	94.06%	86.74%	86.61%	67.75%	65.34%	61.69%
Metapath2vec	70.20%	65.43%	64.83%	79.17%	74.02%	73.17%	51.18%	46.99%	51.03%
StHNE	79.19%	73.66%	69.61%	81.58%	76.02%	72.47%	73.60%	70.30%	63.50%
MAGNN	66.34%	63.90%	62.83%	67.80%	70.02%	63.62%	73.29%	69.19%	60.88%
HGT	87.51%	85.39%	84.96%	92.13%	90.89%	90.27%	80.65%	77.37%	76.86%
DHNE	63.86%	76.89%	64.97%	75.46%	70.82%	69.27%	50.41%	50.52%	50.53%
DyHNE	72.06%	74.25%	74.25%	82.77%	76.54%	72.97%	75.58%	71.33%	66.67%
HDGNN	89.80%	82.33%	82.09%	92.03%	84.37%	84.17%	76.87%	71.86%	71.04%
THINE	91.16%	88.08%	88.25%	94.65%	90.66%	90.71%	79.33%	72.36%	72.51%
SemE	**92.08%**	**89.27%**	**89.43%**	**96.65%**	**91.25%**	**91.39%**	**88.23%**	**78.58%**	**78.33%**
p	6.67e−3	2.45e−3	8.05e−5	3.11e−6	4.71e−2	1.16e−2	4.90e−11	1.05e−3	2.94e−4

of sampling number n, the performance of SemE is also improved for the same reason. In addition, the increasing trend of all metrics is gradually decreasing.

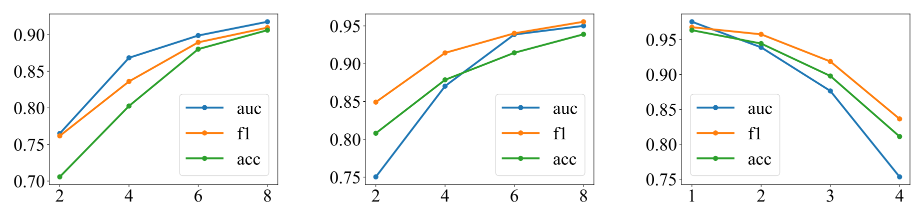
Semantic information number q. The hyper-parameter l is controlled by the hyper-parameter q, and they have the same impact on the model performance. Therefore, we only present the results of q here. When q changes, K and n are both fixed to be 8, and the final experiment results are demonstrated in the Figure 5(c). From it, we can see that with the increase of walk length q, the performance of SemE is decreased. It means that the more extended semantic units, the less accurate the description of node attributes is. That is, too-long semantic units will contain noise. Therefore, the optimal length of semantic units is 1 because such semantic units only describe the slightest semantic events based on meta-paths. Therefore, we can use them to to describe the semantic evolution pattern of nodes accurately.

4.6 Ablation Study

In this experiment, we compare SemE with its three variants to demonstrate the effectiveness of each of our proposed components. The "Structure" indicates using the structure information H_{st} of the network. The "Semantic" represents using semantic units to capture semantic information, and the "Temporal" denotes capturing temporal information by simulating the temporal dynamics of semantic units. Since the temporal information is only available based on the semantic units, we cannot prove the effectiveness of the

TABLE 5
Performance on temporal link prediction task.

Methods	Aminer			DBLP			Yelp		
	auc	f1	acc	auc	f1	acc	auc	f1	acc
Deepwalk	81.60%	80.77%	77.95%	81.50%	80.39%	77.57%	75.08%	66.33%	70.67%
LINE	78.12%	68.34%	71.52%	76.56%	62.90%	69.60%	88.05%	81.08%	79.78%
Node2Vec	80.38%	78.19%	78.64%	83.62%	81.29%	81.21%	80.58%	79.37%	80.46%
ProNe	83.16%	82.54%	80.63%	85.59%	85.11%	84.07%	91.82%	89.97%	90.08%
DySAT	80.94%	78.82%	78.01%	82.05%	79.81%	78.90%	91.09%	90.47%	88.52%
HTNE	80.34%	73.63%	72.81%	85.08%	82.02%	82.20%	89.54%	87.23%	86.19%
MTNE	81.21%	75.34%	74.23%	84.63%	81.80%	80.21%	91.49%	88.41%	87.13%
Metapath2vec	77.64%	71.75%	71.46%	81.07%	78.12%	78.40%	85.06%	84.31%	84.48%
StHNE	76.54%	76.13%	71.53%	75.19%	74.72%	72.06%	90.06%	82.37%	77.39%
MAGNN	72.41%	60.05%	66.23%	74.19%	65.08%	64.14%	80.98%	78.67%	77.09%
HGT	87.26%	85.84%	86.43%	90.91%	89.64%	89.02%	96.81%	92.56%	92.13%
DHNE	81.78%	74.42%	76.27%	82.09%	63.78%	80.12%	93.63%	89.98%	90.32%
DyHNE	77.16%	77.01%	71.70%	80.07%	78.52%	72.80%	92.09%	87.03%	85.58%
HDGNN	81.09%	78.08%	78.60%	86.76%	83.33%	82.17%	93.92%	89.07%	88.51%
THINE	83.61%	80.21%	79.72%	87.09%	84.69%	83.67%	95.20%	90.33%	90.55%
HPGE	84.92%	81.67%	81.28%	88.85%	87.33%	85.96%	96.37%	91.03%	90.19%
TGT	88.63%	86.06%	86.72%	92.83%	92.01%	91.54%	96.45%	93.96%	93.32%
KHGT	90.35%	87.26%	87.98%	93.17%	92.61%	92.09%	96.13%	93.68%	93.43%
SemE	**91.76%**	**90.96%**	**90.61%**	**95.01%**	**95.55%**	**93.89%**	**97.55%**	**96.77%**	**96.34%**
p	2.06e−6	3.99e−9	1.09e−6	2.96e−5	9.60e−7	1.27e−5	6.99e−5	4.97e−6	7.67e−8



(a) Parameter K (Aminer) (b) Parameter n (DBLP) (c) Parameter q (Yelp)

Fig. 5. Parameter analysis.

temporal components individually, so we illustrate their effectiveness by comparing experimental results.

We finally obtain similar results on all datasets, so we take Aminer as an example for analysis, and results are listed in Table 6. By comparing the data of line 2 and line 3 in table 6, we can find that semantic information is crucial for the temporal HIN, which will significantly affect the performance of the model SemE in downstream tasks. Semantic units based on the meta-path can effectively capture the semantic information in the network, and the star semantic network can effectively describe the relationship between semantic units. Besides, by comparing line 3 and line 5, we can see that in the temporal HIN, temporal information has a significant impact on the performance of the model. Our proposed attention-Hawkes process can effectively describe the semantic evolution pattern of nodes. Finally, by comparing line 4 and line 5, we will find that structure information is also indispensable in the temporal HIN. Using the embedding method on static homogeneous networks can effectively capture the structure information of the network. To sum up, our model SemE can efficiently capture the structure information, semantic information, and temporal information in the network. Hence, each component in SemE is necessary, and they ensure that SemE performs well in the task of temporal link prediction.

4.7 Performance and Effciency on the larger dataset

In this section, we demonstrate the efficiency and performance of SemE on a larger and more complex dataset Freebase [27]. Freebase contains node types Book(B), Film(F), Music(M), Sports(S), People(P), Location(L), Organization(O), Business($\hat{B}$). The task is to predict the categories of books. The specific information of Freebase, the 8 metapaths we used and the experimental results are shown in Table 7. Moreover, SemE-x is a method that represents SemE only considers the top x meta-paths.

From the experimental results, we can see that when there are fewer meta-paths, the efficiency of SemE is acceptable and its performance is comparable to the baseline model. While as the number of meta-paths increases, the time overhead of SemE increases rapidly. However, we can see that more meta-paths do not necessarily mean better performance. The reason for this might be that some meta-paths do not provide any gain for downstream tasks and may even introduce noise. Therefore, what we need is just a small number of carefully selected meta-paths, which will not consume a lot of time and memory.

TABLE 6
Comparative results of SemE with its three variants.

Dataset	Structure	Semantic	Temporal	auc	f1	acc
Aminer	✓	✗	✗	81.90%	79.57%	79.15%
	✓	✓	✗	88.16%	87.71%	86.33%
	✗	✓	✓	90.31%	89.94%	89.72%
	✓	✓	✓	**91.76%**	**90.96%**	**90.61%**
p		-		6.9e−5	1.9e−5	4.8e−4

TABLE 7
Node Classification on Freebase.

Freebase	#node	#ntype	#edge	#etype
	180,098	8	2,115,376	64
Meta-path	BB,BFB,BMB,BSB,BPB,BLB,BOB,B$\hat{B}$B			
Method	Memory	Time	macro-F1	micro-F1
Simple-HGN	13.7GB	**423.8s**	46.26%	**65.75%**
SemE-2	**12.8GB**	1503.7	48.46%	65.37%
SemE-4	14.2GB	2728.1s	50.61%	65.37%
SemE-6	14.4GB	4066.0s	**50.89%**	64.15%
SemE-8	14.4GB	5652.4s	50.78%	64.76%

TABLE 8
Experiment results of SemE with different initialization methods.

Methods	macro-F1		micro-F1	
	60%	80%	60%	80%
Deepwalk	46.4%	45.5%	46.5%	47.3%
Node2Vec	46.0%	47.0%	46.8%	47%
LINE	39.5%	43.6%	39.5%	45%
ProNE	**48.6%**	**49.7%**	**48.8%**	**50.2%**
p	4.47e−7	5.86e−8	3.46e−6	7.24e−7

TABLE 9
Results of relation sampling and semantic sampling.

Task	Methods	Relation -sampling	Semantic -sampling	p
Node classification	macro-F1(60%)	45.85%	**48.64%**	6.0e−7
	micro-F1(60%)	46.11%	**48.83%**	8.7e−7
	macro-F1(80%)	48.24%	**49.73%**	2.6e−5
	micro-F1(80%)	48.33%	**50.20%**	8.4e−7
Link prediction	auc	86.45%	**88.23%**	7.5e−6
	f1	77.50%	**78.58%**	1.6e−4
	acc	77.52%	**78.33%**	6.9e−4

4.8 Initialization with Different Embedding Methods

In this section, we try to study the impact of different initialization methods on the performance of the SemE. Therefore, we use different methods to feature semantic units and compare their performance. In addition, for all methods, we set the embedding dimension d as 256, and we report the F1 value of node classification on Aminer. The final results are listed in Table 8.

It can be seen that initialized by ProNE performs significantly better than other methods. Therefore, in other experiments, we use ProNE by default to initialize embedding for semantic units and structure information of nodes. On the one hand, the result shows that the star semantic network can effectively describe the relationship between semantic units, and more advanced embedding methods can extract this information more accurately. On the other hand, it indicates that the performance of SemE is positively correlated with the effect of the initialized method. Therefore, with the innovation of the network embedding method, the performance of our model will be better.

4.9 Relation Sampling or Semantic Sampling?

When generating the semantic network, SemE considers the impact of the semantic unit. While NARS [28] considers the influence of relationship and extract relation information from HINs. Therefore, here we will compare the performance of relation sampling and semantic sampling.

For simplicity, we report the results of node classification on Aminer and link prediction on Yelp. For Aminer, the relationship set we sample is $\{\{A-P\},\{A-P,P-C\},\{A-P,P-P\},\{A-P,P-C,P-P\}\}$. While for Yelp, the relationship set is $\{\{U-B\},\{U-B,B-S\}\}$. Because the temporal information cannot be considered when using relation sampling in SemE, to ensure the fairness of the experiment, we omit the step of semantic evolution modeling in SemE for both approaches. For the relationship sampling model, we use ProNE to generate embedding for each relation subgraph, then merge these embedding and structure embedding H_{st} by the attention mechanism, and train the downstream tasks with an mlp. While for the SemE, we use ProNE to generate embedding for semantic units, then we converge semantic units to learn the embedding of nodes with the excitation function $\mu(t-t_i)=1$ for all semantic units. In addition, for all methods, we set the embedding dimension d of ProNE as 256, and other parameters follow the default settings. The final results are listed in Table 9. From the experimental results and p-value, we can see that the performance of semantic sampling is significantly better than relation sampling, which indicates that semantic sampling can better preserve the semantics of HINs. This is because semantic units can accurately describe the attributes of nodes, while random sampling relationship subgraphs may bring noise.

4.10 Contructing Semantic networks with different topologies

In Table 10, we show the effect of mesh and star topologies on performance, efficiency, and memory overhead of node classification on Aminer. Except for the topology and sample times K, other parameters are set to default values.

On the one hand, from the experimental results, we can see that adding central nodes does not lose information. On the other hand, the star topology can effectively reduce the time and memory overhead due to the sparser network constructed. Furthermore, since constructing a star semantic network incurs lower cost, we can consider more semantic units and achieve better performance by setting a larger sampling times K.

TABLE 10
Comparative results of SemE with different topologies.

Topology	K	macro-F1	micro-F1	Time	Memory
Mesh	1	46.89%	46.83%	10.87h	170GB
Star	1	46.87%	47.00%	**327.12s**	**2.61GB**
Star	8	**48.64%**	**48.83%**	2208.65s	8.36GB
p	-	1.04e−5	2.03e−5	-	-

5 Related Work

Since 2019, more and more attention has been paid to temporal HINs. As a result, some related works have been put forward and some reviews [29], [30] have summarized them. Here we will introduce these HIN embedding methods in two categories.

1) Snapshot-based methods. Change2vec [31] models the opening and closing process of shapes (such as triangles) in the network by meta-path and successfully integrates heterogeneous information and time information in the embedding. After dividing the network into several snapshots, DHNE [5] uses historical-current graphs to learn embedding from the network. DyHNE [18] generates static embeddings by decomposing the matrix generated by meta-paths on the earliest network snapshot. Then, based on matrix perturbation theory, it iteratively updates the embeddings on the remaining network snapshots. DyHAN [4] generates the embedding of nodes on each snapshot by the hierarchical attention mechanism and then aggregates the historical embedding of nodes with Time-Level attention.

2) Evolutionary dynamics-based methods. HDGNN [24] extends heterogeneous GNN by combining time evolution information and then captures the structure, semantic, and temporal information of the temporal HIN. HDGAN [32] captures structural information, semantic information, and temporal information in the network by three attention mechanisms, in which temporal attention is based on the time attenuation effect. DyHINE [33] aggregates neighbor features by a hierarchical attention mechanism, then uses temporal random walk and a dynamic operator to capture dynamic interaction and update node embedding in real-time, respectively. LIME [34] quickly and efficiently adapts to a constantly evolving network with lower memory and time cost by using the recursive neural network (RsNN) with optimization strategies. HINTS [35] transforms embeddings of papers into the parameters of a formal model to predict citation counts immediately after publication. THINE [6] and HPGE [7] simulate the evolution of the temporal HIN by semantic units and the Hawkes process. Based on these temporal methods, SemE further considers the deeper semantic information in HINs, improving the performance of the model. While by simplifying the network structure by the star topology, it reduces time and memory overhead.

6 Conclusion

In this paper, we propose SemE to solve the temporal HIN embedding problem. By the star topology found in Ethernet, SemE constructs the sparse semantic to consider the global interaction between semantic units, which significantly reduces the memory and time overhead. Experiments on three real-world temporal HINs demonstrate that SemE performs better than competitive counterparts.

Acknowledgments

The work is supported by the National Natural Science Foundation of China under grant No.62172174.

References

[1] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.
[2] G. Mai, K. Janowicz, and B. Yan, "Combining text embedding and knowledge graph embedding techniques for academic search engines," in *Proceedings of the ISWC*, vol. 2241, 2018, pp. 77–88.
[3] E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar, and V. S. Pande, "Potentialnet for molecular property prediction," *ACS central science*, vol. 4, no. 11, pp. 1520–1530, 2018.
[4] L. Yang, Z. Xiao, W. Jiang, Y. Wei, Y. Hu, and H. Wang, "Dynamic heterogeneous graph embedding using hierarchical attentions," in *Proceedings of the ECIR*, 2020, pp. 425–432.
[5] Y. Yin, L.-X. Ji, J.-P. Zhang, and Y.-L. Pei, "Dhne: Network representation learning method for dynamic heterogeneous networks," *IEEE Access*, pp. 134 782–134 792, 2019.
[6] H. Huang, R. Shi, W. Zhou, X. Wang, H. Jin, and X. Fu, "Temporal heterogeneous information network embedding," in *Proceedings of the IJCAI*, 2021, pp. 1470–1476.
[7] Y. Ji, T. Jia, Y. Fang, and C. Shi, "Dynamic heterogeneous graph embedding via heterogeneous hawkes process," in *Proceedings of the ECML-PKDD*, 2021, pp. 388–403.
[8] J. Zhang, Y. Dong, Y. Wang, J. Tang, and M. Ding, "Prone: Fast and scalable network representation learning," in *Proceedings of the IJCAI*, 2019, pp. 4278–4284.
[9] F. Fouss, L. Yen, A. Pirotte, and M. Saerens, "An experimental investigation of graph kernels on a collaborative recommendation task," in *Proceedings of the ICDM*. IEEE, 2006, pp. 863–868.
[10] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "Fast and accurate link prediction in social networking systems," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2119–2132, 2012.
[11] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.
[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proceedings of the ICLR*, 2018.
[13] H. Huang, Z. Fang, X. Wang, Y. Miao, and H. Jin, "Motif-preserving temporal network embedding," in *Proceedings of the IJCAI*, 2020, pp. 1237–1243.
[14] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *Proceedings of the KDD*, 2008, pp. 990–998.
[15] B. Perozzi, R. Al-Rfou', and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the KDD*, 2014, pp. 701–710.
[16] Y. Dong, V. N. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the KDD*, 2017, pp. 135–144.
[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the WWW*, 2015, pp. 1067–1077.
[18] X. Wang, Y. Lu, C. Shi, R. Wang, P. Cui, and S. Mou, "Dynamic heterogeneous information network embedding with meta-path based proximity," *TKDE*, pp. 1–1, 2020.
[19] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the KDD*, 2016, pp. 855–864.
[20] F. Xinyu, Z. Jiani, M. Ziqiao, and K. Irwin, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of the WWW*, 2020, pp. 2331–2341.
[21] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of the WWW*, 2020, pp. 2704–2710.
[22] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dysat - deep neural representation learning on dynamic graphs via self-attention networks," in *Proceedings of the WSDM*, 2020, pp. 519–527.

[23] Y. Zuo, G. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, "Embedding temporal network via neighborhood formation," in *Proceedings of the KDD*, 2018, pp. 2857–2866.

[24] X. Xu, T. Zhong, C. Li, G. Trajcevski, and F. Zhou, "Heterogeneous dynamical academic network for learning scientific impact propagation," *Knowledge-Based Systems*, p. 107839, 2022.

[25] L. Xia, C. Huang, Y. Xu, and J. Pei, "Multi-behavior sequential recommendation with temporal graph transformer," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[26] L. Xia, C. Huang, Y. Xu, P. Dai, X. Zhang, H. Yang, J. Pei, and L. Bo, "Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation," in *Proceedings of the AAAI*, vol. 35, no. 5, 2021, pp. 4486–4493.

[27] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, "Heterogeneous network representation learning: A unified framework with survey and benchmark," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4854–4873, 2020.

[28] L. Yu, J. Shen, J. Li, and A. Lerer, "Scalable graph neural networks for heterogeneous graphs," *arXiv preprint arXiv:2011.09679*, 2020.

[29] Y. Dong, Z. Hu, K. Wang, Y. Sun, and J. Tang, "Heterogeneous network representation learning," in *Proceedings of the IJCAI*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 4861–4867, survey track. [Online]. Available: https://doi.org/10.24963/ijcai.2020/677

[30] 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," *IEEE Transactions on Big Data*, vol. 9, pp. 415–436, 2020.

[31] R. Bian, S. Y. Koh, G. Dobbie, and A. Divoli, "Network embedding and change modeling in dynamic heterogeneous networks," in *Proceedings of the SIGIR*, 2019, pp. 861–864.

[32] Q. Li, Y. Shang, X. Qiao, and W. Dai, "Heterogeneous dynamic graph attention network," in *Proceedings of the ICKG*, 2020, pp. 404–411.

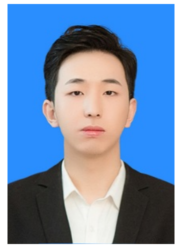
[33] Y. Xie, Z. Ou, L. Chen, Y. Liu, K. Xu, C. Yang, and Z. Zheng, "Learning and updating node embedding on dynamic heterogeneous information network," in *Proceedings of the WSDM*, 2021, pp. 184–192.

[34] H. Peng, R. Yang, Z. Wang, J. Li, L. He, P. Yu, A. Zomaya, and R. Ranjan, "Lime: Low-cost incremental learning for dynamic heterogeneous information networks," *IEEE Transactions on Computers*, 2021.

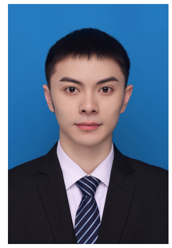
[35] S. Jiang, B. Koch, and Y. Sun, "Hints: Citation time series prediction for new publications via dynamic heterogeneous information network embedding," in *Proceedings of the WWW*, 2021, pp. 3158–3167.

Wei Zhou is a graduate student in the School of Computer Science at Huazhong University of Science and Technology in China. His research interests include data mining and heterogeneous information network analysis.

Hong Huang is an Associate Professor in Huazhong University of Science and Technology, China. She received her PhD in Computer Science from University of Göttingen, Germany in 2016, and her M.E. degree in Electronic Engineering from Tsinghua University, Beijing, China in 2012. Her research interests lie in social network analysis, data mining and knowledge graph.



Ruize Shi received the B.S. degree from Huazhong University of Science and Technology in 2020. He is currently a master student at Huazhong University of Science and Technology. His current research interests are in data mining and social network analysis.



Xiran Song is a graduate student in the School of Computer Science at Huazhong University of Science and Technology in China. His research interests include the recommendation system and graph representation learning.

Xue Lin is a graduate student in the School of Computer Science at Huazhong University of Science and Technology in China. Her research interests include big data analysis.

Xiao Wang is an Associate Professor in the the School of Software, Beihang University. He received his Ph.D. degree from Tianjin University, China. He was a joint-training Ph.D. in Washington University in St. Louis, USA and a postdoctoral researcher in Tsinghua University, China. His current research interests include data mining, social network analysis, and machine learning. Until now, he has published more than 70 papers in refereed journals and conferences.

Hai Jin is a Chair Professor of computer science and engineering at Huazhong University of Science and Technology (HUST) in China. Jin received his PhD in computer engineering from HUST in 1994. In 1996, he was awarded a German Academic Exchange Service fellowship to visit the Technical University of Chemnitz in Germany. Jin worked at The University of Hong Kong between 1998 and 2000, and as a visiting scholar at the University of Southern California between 1999 and 2000. He was awarded Excellent Youth Award from the National Science Foundation of China in 2001. Jin is a Fellow of IEEE, Fellow of CCF, and a life member of the ACM. He has co-authored more than 20 books and published over 900 research papers. His research interests include computer architecture, parallel and distributed computing, big data processing, data storage, and system security.