Temporal Heterogeneous Information Network Embedding

Hong Huang[1,2,3*] , **Ruize Shi**[1,2,3] , **Wei Zhou**[3] , **Xiao Wang**[4*] , **Hai Jin**[1,2,3] and **Xiaoming Fu**[5]
[1]National Engineering Research Center for Big Data Technology and System
[2]Service Computing Technology and Systems Laboratory
[3]Huazhong University of Science and Technology, China
[4]Beijing University of Posts and Telecommunications, China
[5]University of Goettingen, Germany
{honghuang, rzshi, zhouw, hjin}@hust.edu.cn, xiaowang@bupt.edu.cn, fu@cs.uni-goettingen.de

Abstract

Heterogeneous information network (HIN) embedding, learning the low-dimensional representation of multi-type nodes, has been applied widely and achieved excellent performance. However, most of the previous works focus more on static HINs or learning node embeddings within specific snapshots, and seldom attention has been paid to the whole evolution process and capturing all dynamics. In order to fill the gap of obtaining multi-type node embeddings by considering all temporal dynamics during the evolution, we propose a novel *temporal HIN embedding* method *(THINE)*. THINE not only uses attention mechanism and meta-path to preserve structures and semantics in HIN but also combines the Hawkes process to simulate the evolution of the temporal network. Our extensive evaluations with various real-world temporal HINs demonstrate that THINE achieves the SOTA performance in both static and dynamic tasks, including node classification, link prediction, and temporal link recommendation.

1 Introduction

In recent years, network embedding has attracted more and more attention on account of its outstanding performance. It maps nodes to a low-dimensional space and preserves the features and structures of the network simultaneously. A lot of excellent algorithms such as Deepwalk [Perozzi *et al.*, 2014], LINE [Tang *et al.*, 2015b] have been successfully applied in various network-related tasks, such as node classification, node clustering, and link prediction.

However, these methods all focus on homogeneous networks, while most real-world data are *heterogeneous information networks* (HINs) with multiple types of nodes and relations. For example, an academic network generally has three types of nodes: Author (A), Paper (P), Conference (C); and multiple types of relations: co-author relations, reference relations, write/written relations between authors and papers, publish/published relations between papers and conferences. In HIN, different types of nodes and edges could generate various embeddings and contain more complex structures and interrelations. Therefore, more and more researchers have paid attention to HINs such as PTE [Tang *et al.*, 2015a], Metapath2Vec [Dong *et al.*, 2017], and MAGNN [Fu *et al.*, 2020].

Nonetheless, most of the current works are proposed for static HIN, which in contrast to the reality, where the HIN actually evolves over time. Examples include academic networks where authors may publish different papers in different years, and the grade of business varies according to users' reviews with time changing in Yelp. Therefore, simply treating a temporal HIN as a static HIN will inevitably not accurately capture the structure and semantics when HIN changes.

As a result, there is a growing need to understand temporal HIN. However, it faces two serious challenges. First, *how to effectively preserve the structural and semantic dynamics in temporal HINs?* Dynamics describe all the changes of nodes and edges in HINs during its evolution, including the nodes' addition, edges' deletion, etc. Thus, accurately capturing the dynamics is critical to study temporal HIN. However, most previous works like DHNE [Yin *et al.*, 2019], model temporal HINs using snapshots by simply dividing the time into several periods, which would lose dynamics within snapshots.

Another challenge is *how to capture the temporal influence between heterogeneous nodes?* Unlike homogeneous networks, HINs contain multiple types of nodes and edges, thus preserve more complex semantics and structures. For example, in an academic network, we usually consider the temporal influence from the same types of nodes like *author-author* or *paper-paper*. Moreover, in HINs, we should also consider the temporal effects from different types of nodes like *author-paper*. But due to the difficulty of simulating the influence between heterogeneous nodes, the majority of previous works like HDGAN [Li *et al.*, 2020], DyHNE [Wang *et al.*, 2020] only consider the temporal influence from the same types of nodes merely.

To this end, we propose THINE, a novel *temporal HIN embedding* model for capturing the dynamics among all types of nodes. We first define various meta-paths to capture the semantics and structures of a HIN. Then for a specific downstream task, we generate candidate meta-path sets related to the task. By modeling the temporal influence between nodes with the Hawkes process, we obtain each node's embedding. Moreover, two levels of attention mechanisms are

[*]Corresponding Author

applied to distinguish weight in various aspects. One is for different types of meta-path, and the other is for the distance of the neighbor nodes. Experiments on various real-world datasets show that, compared with several SOTA methods, our THINE performs better in both static and dynamic tasks.

The contributions can be summarized as follows:

- We study the problem of temporal HIN embedding by considering the evolutionary dynamics.
- We initiate a novel temporal HIN embedding model, which uses meta-path to capture structural and semantic information of HIN, leverages the Hawkes process to model network evolution, and applies two levels of attention to capture structural and semantic differences respectively.
- Experimental results on three real-world datasets show that THINE outperforms several SOTA methods.

2 Preliminaries

The evolution of a HIN can be regarded as the addition and deletion of nodes or edges at different times. Traditionally, we define temporal HIN in the following:

Definition 1: ***Temporal HIN.*** A temporal HIN is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \phi, \varphi)$, where $\mathcal{V}$ indicates the set of nodes, $\mathcal{E}$ indicates the set of edges and $\mathcal{T}$ denotes the set of timestamps. Besides, there are two mapping functions $\phi : \mathcal{V} \rightarrow \mathcal{A}$, $\varphi : \mathcal{E} \rightarrow \mathcal{R}$, while $\mathcal{A}$, $\mathcal{R}$ indicate the set of types for nodes and edges. For each HIN, it satisfies that $|\mathcal{A}| + |\mathcal{R}| > 2$.

Especially, an edge $e_{ij}^t \in \mathcal{E}$ represents a relation established between nodes v_i and v_j at time t. It is worth noting that two nodes may establish multiple relations at different times. For instance, conference (C) may publish multiple papers by the same author (A), so it is probable that there exist numerous relations between A and C in academic networks.

Definition 2: ***Influence of Node Pairs.*** The influence of node pairs indicates the contribution of two nodes to establish a connection between them. Given a node pair (v_x, v_y), where $v_x, v_y \in \mathcal{V}$ and $\mathcal{V}$ is the set of nodes, the influence of node pair is $\eta_{x,y}$. Formally, we have:

$$\eta_{x,y} = -\|u_x - u_y\|^2, \tag{1}$$

where u_x, u_y represent the embeddings of nodes v_x and v_y.

Definition 3: ***Meta-path.*** Given a temporal HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \phi, \varphi)$, meta-path $\mathcal{M}$ is a path which is defined as $a_1 \xrightarrow{r_1} a_2 \xrightarrow{r_2} a_3 \xrightarrow{r_3} ... \xrightarrow{r_{l-1}} a_l$, where $r_i \in \mathcal{R}$, $a_i \in \mathcal{A}$, and $\mathcal{A}$, $\mathcal{R}$ indicate the set of types for nodes and edges. Obviously, a meta-path describes a complex relation between types a_1 and a_l. Especially, a sequence of nodes $(v_1, v_2, \ldots, v_l)$ that satisfies the law of meta-path $\mathcal{M}$ is a path instance m of meta-path $\mathcal{M}$.

Definition 4: ***Candidate Meta-path Set.*** The candidate meta-path set $S(t)$ of a temporal edge $e_{ij}^t \in \mathcal{E}$, is a set that includes all the path instances which involve source node v_i and are generated before time t. Note that a path instance formed before time t means that all of the temporal edges it included generated before time t.

In this paper, our goal is to obtain multi-type node embeddings with consideration of temporal dynamics. Formally, we define our problem as follows:

Problem. ***Temporal HIN Embedding.*** Given a temporal HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \phi, \varphi)$, the goal is to learn a mapping function $f : \mathcal{V} \rightarrow \mathbb{R}^d$, where $d \ll |\mathcal{V}|$ and d represents the number of embedding dimensions. The function f needs to not only capture the temporal dynamics of the network but also consider the influences between all types of nodes.

3 The Proposed Model

3.1 Model Overview

In this section, we will explain the details of our proposed model THINE, which can capture the structures and semantics of HIN and combine the impact of temporal dynamics simultaneously. As Figure 1 shown, THINE gets the structural interactions between different types of nodes by meta-path based random walk. Then, we obtain the candidate meta-path set for each edge to model the dynamic structures and semantics of temporal HIN with the Hawkes process [Hawkes, 1971]. Furthermore, optimizing with structural-level and semantic-level attention mechanisms that distinguish the impacts of different relations, we aggregate the effects for every node to obtain multi-type node embeddings.

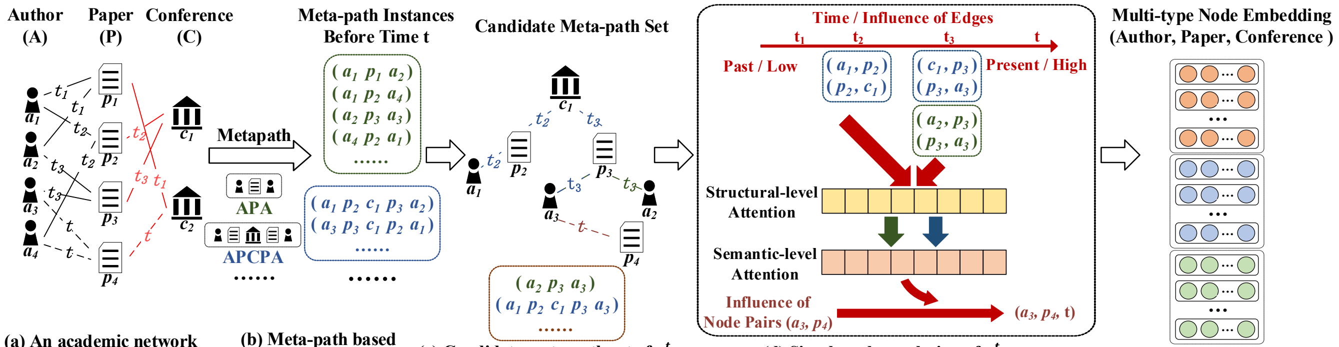
3.2 THINE Model

Capturing semantics with meta-path. THINE first uses meta-path based random walk to extract information of HIN. The construction of meta-path determines what semantics and structures we can capture. Therefore, the selection of meta-path is critical to study HINs. The key to defining meta-path is to contain as many semantics as possible. For example, for academic networks, besides the mate-path of *author-paper* relation that considered by previous models, we also consider the meta-path of *paper-paper* relation, written as *APPA*. In total, the meta-path we defined are listed in Table 1. With those meta-paths, we can well preserve the semantics in HINs. Moreover, the nodes and edges in the network are influenced by the nodes themselves and the related candidate meta-path set. Hence, based on the influence of node pairs, we model the influence of candidate set to understand temporal HIN.

Modeling dynamics of candidate meta-path set. Furthermore, we model the influence of candidate meta-path set to capture semantics and structures of temporal HIN with the Hawkes process. Generally, the Hawkes process is used to simulate the impact of past events on the present. Obviously, the older events are, the fewer impacts they have on today. Specifically, for THINE, we apply an attention with the Hawkes process for every influence. Therefore, the influence of candidate meta-path set, to wit the influence of all related meta-path instances, is formally defined as

$$\eta_s(t) = \sum_{t_m < t} \eta_m(t), \tag{2}$$

where m is one meta-path instance, and $t_m < t$ indicates the meta-path instance m generated before time t. For convenience, we use $\eta_i, i \in (s, m, e)$ to represent the influence of candidate meta-path set, one meta-path, and one temporal edge. Therefore, $\eta_s(t)$, $\eta_m(t)$, and $\eta_e(t)$ indicate corresponding influence before time t. So that we need to primarily


Author (A)
Paper (P)
Conference (C)
a_1 a_2 a_3 a_4
p_1 p_2 p_3 p_4
c_1 c_2
t_1 t_2 t_3 t
Metapath
APA
APCPA
......
Meta-path Instances Before Time t
$(a_1\ p_1\ a_2)$
$(a_1\ p_2\ a_4)$
$(a_2\ p_3\ a_3)$
$(a_4\ p_2\ a_1)$
$(a_1\ p_2\ c_1\ p_3\ a_2)$
$(a_3\ p_3\ c_1\ p_2\ a_1)$
Candidate Meta-path Set
$(a_2\ p_3\ a_3)$
$(a_1\ p_2\ c_1\ p_3\ a_3)$
Time / Influence of Edges
Past / Low
Present / High
(a_1, p_2) (p_2, c_1)
(c_1, p_3) (p_3, a_3)
(a_2, p_3) (p_3, a_3)
Structural-level Attention
Semantic-level Attention
Influence of Node Pairs (a_3, p_4)
(a_3, p_4, t)
Multi-type Node Embedding (Author, Paper, Conference)

(a) An academic network ($t_i < t$, i=1,2,3)

(b) Meta-path based random walk

(c) Candidate meta-path set of $e^t_{a_3p_4}$

(d) Simulate the evolution of $e^t_{a_3p_4}$

Figure 1: The overall architecture of the proposed THINE

model the influence of one meta-path instance $\eta_m(t)$, which can be considered as the influence of all the edges it included. Thus, it is defined as

$$\eta_m(t) = \sum_{e_{ij}\in m} \eta_{e_{ij}}(t), \tag{3}$$

where e_{ij} represents an edge between nodes v_i and v_j. One edge can be represented by two nodes it connects. So we model the influence of one temporal edge by using the influence of node pairs and the Hawkes process, which means

$$\eta_{e_{ij}}(t) = \eta_{i,j} \times \sum_{t_{i,j}<t} \kappa(t - t_{i,j}), \tag{4}$$

where $t_{i,j}$ represents the timestamp of edge e^t_{ij}, $\kappa(t - t_{i,j})$ describes the time decay effect and we define $\kappa(t - t_{i,j}) = e^{(-\delta_i(t - t_{i,j}))}$. Moreover, δ is a trainable parameter related to nodes and adjusts the ratio of time decay effect.

Note that we select a subset of the candidate meta-path set for training due to the computational complexity. Specifically, we choose n meta-path instances generated most recently from the time t while n is a hyperparameter. Likely, for one meta-path instance, we select z candidate edges closest to the source node to train and z is also a hyperparameter.

Optimization with attention mechanism. Specifically, we show how THINE calculates the influence of the candidate meta-path set in detail. As shown in Figure 1 (c), the candidate meta-path set of $e^t_{a_3p_4}$ includes $m_1 : (a_1, p_2, c_1, p_3, a_3)$, $m_2 : (a_2, p_3, a_3)$, etc. Thus, $\eta_s(t) = \eta_{m_1}(t) + \eta_{m_2}(t)$. Obviously, m_1 and m_2 are generated following different meta-path APCPA and APA, which dissimilarly affect different tasks. To catch this subtle distinction, we apply a semantic-level attention mechanism [Bahdanau *et al.*, 2015]. Formally, we define the weights for different types of meta-path as follows:

$$\omega_b = \frac{e^{\omega_b}}{\sum_c e^{\omega_c}}, \tag{5}$$

where c is the set of all meta-path defined in the task, and ω_b represents the weight of b-th meta-path. Taking account of the semantics of different meta-path, we reformulate the influence of candidate meta-path set as

$$\eta_s(t) = \sum_{t_m<t} \omega_{\mathcal{M}} \times \eta_m(t), \tag{6}$$

where $\omega_{\mathcal{M}}$ is the weight of meta-path $\mathcal{M}$, and meta-path instance m belongs to $\mathcal{M}$. As we defined above, $\eta_{m_1}(t) = \eta_{e_{a_1p_2}}(t) + \eta_{e_{p_2c_1}}(t) + \eta_{e_{c_1p_3}}(t) + \eta_{e_{p_3a_3}}(t)$. Clearly, the influence of every edge should be different on account of the hop counts to training edge $e^t_{a_3p_4}$. Therefore, a structural-level attention mechanism is used to capture this difference, and we denote the weights related to the hop counts as

$$\theta_{h_{oq}} = \frac{e^{\theta_{h_{oq}}}}{\sum_{z'} e^{\theta_h}}, \tag{7}$$

where h_{oq} represents the hop counts of edge e_{oq} to the source node, while the weight of h_{oq}-th hop is indicated as $\theta_{h_{oq}}$. z' is positively related to the number of candidate edges z. Therefore, the influence of one meta-path instance is re-defined as

$$\eta_m(t) = \sum_{e_{ij}\in m} \theta_{h_{ij}} \times \eta_{e_{ij}}(t). \tag{8}$$

With the meta-path, two-level attention mechanisms, and Hawkes process, we model the evolution of every temporal edge. In this way, THINE is able to obtain multi-type node embedding with preserving semantics and structures in both static and dynamic networks. Based on the formulas above, we define the conditional intensity function $\widetilde{\lambda}_{x,y}(t)$ to represent the intensity of generating a temporal edge between nodes v_x and v_y at time t as

$$\begin{aligned}\widetilde{\lambda}_{x,y}(t) &= \eta_{x,y} + \eta_s(t) \\ &= \eta_{x,y} + \sum_{t_m<t} (\omega_{\mathcal{M}} \times \sum_{e_{ij}\in m} \theta_{h_{ij}} \times \eta_{e_{ij}}(t)).\end{aligned} \tag{9}$$

Considering that the conditional intensity function should return a positive real number, an exponential function is used to transfer $\widetilde{\lambda}_{x,y}(t)$, which can be formulated as $\lambda_{x,y}(t) = exp(\widetilde{\lambda}_{x,y}(t))$. Moreover, $\lambda_{x,y}(t)$ is squeezed between 0 and 1 simultaneously to indicate the probability of establishing a relation between nodes v_x and v_y.

Loss function. Therefore, we can represent the probability of establishing a connection between nodes v_x and v_y at time t. Especially, with the consideration of conditional intensity function and candidate meta-path set $S(t)$ before time t, we

Datasets	Node Types	# Nodes	Meta-path	Time Steps

Aminer	Author(A)
	Paper(P)
	Conference(C)
		10206
		10457
		2584
			APA
			APPA
			APCPA
				10

DBLP	Author(A)
	Paper(P)
	Conference(C)
		22662
		22670
		2938
			APA
			APPA
			APCPA
				15

Yelp	Star(S)
	User(U)
	Business(B)
		5
		24586
		800
			USU
			BSB
			BUB
			BSUSB
			UBSBU
				15

Table 1: Data statistics

	Methods	Heterogeneous	Temporal
Shallow	Deepwalk [Perozzi *et al.*, 2014]	×	×
	LINE [Tang *et al.*, 2015b]	×	×
	DySAT [Sankar *et al.*, 2020]	×	√
	HTNE [Zuo *et al.*, 2018]	×	√
	MTNE [Huang *et al.*, 2020]	×	√
	Metapath2vec [Dong *et al.*, 2017]	√	×
	StHNE [Wang *et al.*, 2020]	√	×
	DHNE [Yin *et al.*, 2019]	√	√
	DyHNE [Wang *et al.*, 2020]	√	√
Deep	MAGNN [Fu *et al.*, 2020]	√	×
	HDGNN [Zhou *et al.*, 2020]	√	√

Table 2: Baselines

define this probability as

$$p(v_x, v_y|S(t)) = \frac{\lambda_{x,y}(t)}{\sum_{y'} \lambda_{x,y'}(t)}, \tag{10}$$

where y' denotes all nodes except v_x in temporal HIN, and the log likelihood of all node pairs can be indicated as

$$logL = \sum_{(v_x,v_y)\in\mathcal{E}} logp(v_x, v_y|S(t)). \tag{11}$$

Obviously, the calculation of $p(v_x, v_y|S(t))$ needs to summarise the information of the whole network nodes. We apply negative sampling techniques [Zuo *et al.*, 2018] to make our model more effective to mitigate huge computing overhead. So the loss function can be reformulated as

$$-log\sigma(\widetilde{\lambda}_{x,y}(t)) - \sum_{k=1}^{K} E_{v^k\sim P_n(v)}[log\sigma(-\widetilde{\lambda}_{x,k}(t))], \tag{12}$$

where K is the number of negative nodes sampled according to $P_n(v)$. $P_n(v)$ is positive correlated with $d_v^{3/4}$ while d_v denotes the degree of node v. Furthermore, σ is a sigmoid function which can be represented as $\sigma(x) = \frac{1}{1+e^{-x}}$. Finally, we optimize our model by *Adaptive Moment Estimation* (Adam).

4 Experiments and Discussions

4.1 Experimental Setup

Datasets. In order to demonstrate the effectiveness of THINE, we evaluate it on three real-world datasets. They are Aminer [Tang *et al.*, 2008], DBLP[1], and Yelp[2], respectively. The statistics of these datasets are shown in Table 1.

Baselines. We compare THINE with several state-of-the-art network embedding methods, including nine shallow and two deep embedding models, which are listed in Table 2.

Experimental settings. We evaluate THINE and other baselines on a server with Intel Xeon CPU E5-2680, Tesla V100 GPUs, and 250GB Memory. The experimental environment of software is Ubuntu 18.04 with CUDA 10.2. To be fair, the embedding dimension d is set as 100 for all methods. For THINE, the learning rate of Adam is set as 0.003 while the batch size is 500. Moreover, we set the number of walks per node as 10, the walk length is 30, the number n of candidate meta-path instances is 20, the number z of candidate edges is 4, and the negative samples is set to be 5. Furthermore, we use the default parameter settings for other baselines. To ensure experiments more reliable, we take the average of ten results as the final result for each experiment.

4.2 Experiment Performance

We first obtain node embeddings of THINE and other baselines. After that, we evaluate it by both static and dynamic tasks. For static tasks, we use two traditional measures, node classification and link prediction. Further, we use temporal link recommendation to test for dynamic effectiveness.

Node classification. We first apply all the methods to get node embeddings respectively, which are viewed as features to train a Logistic Regression classifier. Especially, the size of the training set is set as 60%, 80% and the remaining nodes as the test set. We use micro-f1 and macro-f1 to evaluate results.

As we can see in Table 3, THINE outperforms other methods on three datasets, which demonstrates that capturing all dynamics and the influence between different types of nodes is helpful for understanding structures of HIN. Moreover, our model has similar results when using 60 or 80 percent of the training set, which also shows the robustness of THINE.

Link prediction. We denote the representation of edge $e_{x,y}$ is $|u_x - u_y|$, where u_x and u_y are embeddings of nodes v_x and v_y. We also train a Logistic Regression classifier, in which input is the representation of edges. For Aminer and DBLP, we focus on co-author relations. Therefore, we hide 25% *AP* randomly since co-author information is implicit in meta-path *APA*. For Yelp, we hide 25% *UR* to predict the connection between users. On each dataset, we randomly choose 25,000 edges as positive ones while generating 25,000 negative ones. The results of link prediction are listed in Table 4.

We can observe that THINE outperforms all the baselines, and most dynamic methods have a good effect on link prediction, which further indicates that temporal information helps extract the structures and semantics of temporal HINs.

Temporal link recommendation. We do a temporal link recommendation experiment to test dynamic effectiveness. Especially, we train all the methods on HINs before time t. Then we make recommendations after time t. For each dataset, the first 80% of the period is used for training while

[1] https://dblp.org

[2] https://www.yelp.com/dataset

Methods	Aminer				DBLP				Yelp			
	micro-f1		macro-f1		micro-f1		macro-f1		micro-f1		macro-f1	
	60%	80%	60%	80%	60%	80%	60%	80%	60%	80%	60%	80%
Deepwalk	44.50%	44.67%	44.48%	44.73%	34.25%	36.00%	34.21%	35.49%	64.38%	65.93%	64.37%	65.91%
LINE	36.33%	38.33%	34.96%	36.21%	29.75%	30.38%	29.73%	30.36%	56.43%	61.75%	55.30%	59.73%
DySAT	43.00%	42.66%	45.00%	44.74%	31.50%	32.25%	31.26%	32.30%	68.44%	69.99%	68.37%	69.70%
HTNE	37.50%	40.83%	35.81%	39.08%	30.75%	32.63%	30.73%	32.68%	61.50%	65.00%	61.22%	64.75%
MTNE	41.33%	45.08%	40.51%	45.67%	31.25%	33.88%	31.04%	33.89%	65.00%	68.75%	64.99%	68.67%
Metapath2vec	38.17%	40.33%	38.12%	40.13%	30.50%	34.50%	30.47%	34.28%	50.87%	60.88%	49.51%	59.80%
StHNE	35.00%	40.00%	25.50%	34.22%	22.00%	19.25%	9.03%	8.44%	67.81%	66.25%	67.04%	65.99%
DHNE	43.62%	39.64%	41.19%	35.97%	37.50%	39.75%	36.90%	38.35%	60.94%	70.00%	60.80%	69.98%
DyHNE	36.33%	42.00%	35.30%	41.31%	22.13%	19.50%	9.06%	8.99%	68.44%	68.13%	67.74%	67.76%
MAGNN	36.67%	39.67%	36.18%	39.61%	24.50%	25.25%	23.85%	25.43%	46.56%	48.75%	46.54%	48.71%
HDGNN	41.67%	44.83%	41.57%	44.61%	37.25%	37.50%	36.89%	36.85%	62.18%	61.87%	62.10%	61.80%
THINE	**48.33%**	**48.67%**	**48.03%**	**48.58%**	**38.50%**	**39.89%**	**37.43%**	**39.64%**	**70.37%**	**71.25%**	**70.23%**	**71.24%**

Table 3: Performance on node classification

Datasets	Metrics	Deepwalk	LINE	DySAT	HTNE	MTNE	Metapath2vec	StHNE	DHNE	DyHNE	MAGNN	HDGNN	MTNE
Aminer	auc	77.07%	68.49%	80.25%	76.53%	82.78%	70.20%	79.19%	63.86%	72.06%	66.34%	89.80%	**91.16%**
	f1	71.19%	64.79%	67.91%	73.18%	75.23%	65.43%	73.66%	76.89%	74.25%	63.90%	82.33%	**88.08%**
	acc	71.09%	63.54%	66.14%	72.55%	74.72%	64.83%	69.61%	64.97%	74.25%	62.83%	82.09%	**88.25%**
DBLP	auc	85.59%	75.11%	82.16%	90.77%	94.06%	79.17%	81.58%	75.46%	82.77%	67.80%	92.03%	**94.65%**
	f1	81.28%	71.24%	75.61%	83.16%	86.74%	74.02%	76.02%	70.82%	76.54%	70.02%	84.37%	**90.66%**
	acc	81.28%	70.09%	75.03%	82.87%	86.61%	73.17%	72.47%	69.27%	72.97%	63.62%	84.17%	**90.71%**
Yelp	auc	50.32%	58.30%	53.67%	65.27%	67.75%	51.18%	73.60%	50.41%	75.58%	73.29%	76.87%	**79.33%**
	f1	63.28%	57.56%	54.27%	64.77%	65.34%	46.99%	70.30%	50.52%	71.33%	69.19%	71.86%	**72.36%**
	acc	52.78%	55.57%	52.88%	60.65%	61.69%	51.03%	63.50%	50.53%	66.67%	60.88%	71.04%	**72.51%**

Table 4: Performance on link prediction

remaining as a test set. Using negative squared Euclidean distance, we predict the top-5, 10 (top-2, 4) connection after time t for test nodes. We then estimate results with precision and recall.

We first make recommendations between nodes of the same type since some heterogeneous baselines only obtain one type of node embedding. From Table 5, THINE has the best performance in this dynamic task. Furthermore, we can see methods for homogeneous networks generally are better than the previous HIN embedding model.

To make our results more convincing, we also make recommendations between multi-type nodes. Note that we remove some baselines or datasets that do not fit for this task. As shown in Table 6, THINE achieves the highest precision and recall among competitors. Furthermore, the methods for HIN perform better than models for homogeneous networks. This may indicate that the methods for HIN should also pay attention to the influence between nodes of the same type, which helps obtain better performance in downstream tasks.

4.3 Parameter Analysis

Besides, we analyze several important parameters of THINE: the number of candidate meta-path instances n, the number of candidate edges z, and the number of negative samples.

The number of candidate meta-path instances n. We show how the number of candidate meta-path instances n influences THINE. As shown in Figure 2 (a), with the increase of candidate meta-path instances, the performance first improves and then decreases, which means that THINE does not need too little or too superfluous information from the past.

The number of candidate edges z. Obviously, the influence of neighbor nodes is related to the number of candidate edges z. From Figure 2 (b), similarly, THINE does not need too much or too little neighbor information. For a balance between complexity and performance, we set the number of candidate edges as 4.

The number of negative samples. Finally, we test the impact of the number of negative samples on model performance. The experiment results are listed in Figure 2 (c). From the results, THINE achieves the best performance when the number of negative samples is around 5.

5 Related Work

Network embedding, namely graph embedding, aims to represent nodes in a low-dimensional space while preserving the properties and structures of the network. Incipiently, inspired by natural language processing [Mikolov *et al.*, 2013], a lot of researchers focus on applying embedding on the network. Mainly, models at that time aimed to represent nodes by using the information of neighbor nodes such as Deepwalk [Perozzi *et al.*, 2014] and LINE [Tang *et al.*, 2015b]. However, these methods are mainly proposed for homogeneous networks and do not consider the temporal information.

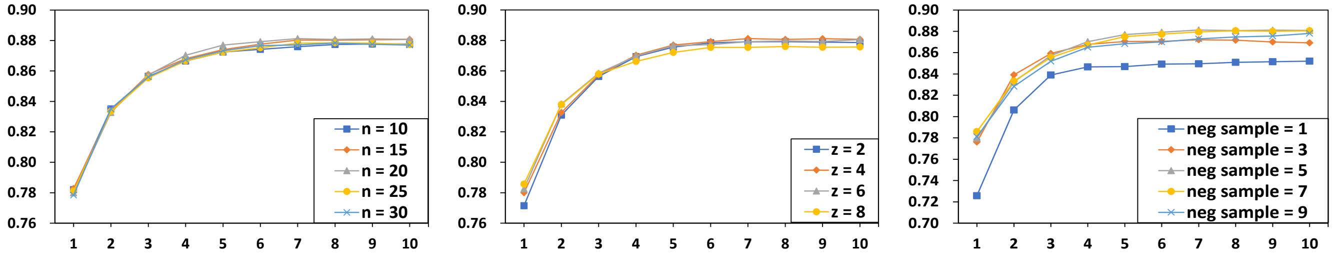
Since then, network embedding goes in different directions: HIN and temporal network. For HIN, plenty of methods capture semantics and structures based on meta-path [Sun and Han, 2012] and have got powerful performance (i.e., Meta-path2vec [Dong *et al.*, 2017], HAN [Wang *et al.*, 2019]). Towards temporal network, some of works use matrix decomposition (DHPE [Zhu *et al.*, 2018]), and others

Methods	Aminer				DBLP				Yelp			
	precision		recall		precision		recall		precision		recall	
	top@5	top@10	top@5	top@10	top@5	top@10	top@5	top@10	top@5	top@10	top@5	top@10
Deepwalk	9.81%	8.03%	2.29%	4.50%	9.80%	7.98%	2.19%	4.25%	0.21%	0.34%	0.065%	0.120%
LINE	7.51%	6.11%	2.24%	3.66%	5.23%	4.12%	1.47%	2.31%	0.17%	0.14%	0.049%	0.058%
DySAT	3.35%	2.74%	0.96%	1.59%	1.80%	1.13%	0.48%	0.62%	0.39%	0.20%	0.061%	0.064%
HTNE	9.98%	8.20%	3.01%	4.94%	7.75%	6.46%	2.25%	3.68%	0.33%	0.27%	0.078%	0.156%
MTNE	10.45%	8.38%	3.15%	5.06%	7.96%	6.39%	2.30%	3.65%	0.05%	0.21%	0.014%	0.122%
Metapath2vec	2.21%	2.46%	0.67%	1.49%	2.33%	2.07%	0.69%	1.23%	0.11%	0.21%	0.027%	0.122%
StHNE	4.60%	3.40%	1.40%	2.03%	3.67%	2.91%	1.01%	1.60%	0.39%	0.20%	0.064%	0.069%
DHNE	3.32%	2.24%	1.02%	1.34%	6.34%	4.87%	1.96%	2.91%	0.23%	0.11%	0.033%	0.033%
DyHNE	6.04%	4.04%	1.74%	2.40%	3.89%	3.09%	1.12%	1.75%	0.24%	0.30%	0.035%	0.122%
MAGNN	2.65%	2.18%	0.81%	1.33%	1.62%	1.26%	0.45%	0.69%	0.32%	0.34%	0.055%	0.167%
HDGNN	12.04%	10.85%	3.82%	6.45%	9.13%	8.79%	2.03%	4.78%	0.39%	0.40%	0.090%	0.175%
THINE	**14.05%**	**12.07%**	**4.31%**	**7.25%**	**11.67%**	**9.47%**	**3.48%**	**5.51%**	**0.40%**	**0.41%**	**0.105%**	**0.217%**

Table 5: Performance on temporal link recommendation between nodes of the same type

Datasets	Metrics	top	Deepwalk	LINE	DySAT	HTNE	MTNE	Metapath2vec	DHNE	MAGNN	HDGNN	MTNE
Aminer	precision	@2	10.33%	6.28%	10.46%	8.60%	10.46%	17.44%	9.43%	8.47%	18.30%	**22.79%**
		@4	10.21%	3.60%	7.56%	6.51%	6.98%	13.83%	7.23%	6.35%	14.37%	**18.62%**
	recall	@2	2.28%	1.97%	3.84%	2.98%	3.24%	5.96%	4.12%	3.14%	6.68%	**8.31%**
		@4	6.75%	2.21%	5.64%	4.61%	4.73%	9.76%	5.11%	4.23%	10.94%	**12.91%**
DBLP	precision	@2	1.76%	1.65%	3.90%	1.95%	2.20%	2.72%	3.40%	3.18%	4.23%	**4.55%**
		@4	2.55%	1.02%	3.39%	1.29%	1.76%	2.57%	3.23%	2.97%	3.87%	**4.08%**
	recall	@2	0.54%	0.46%	0.96%	0.55%	0.64%	0.83%	0.84%	0.79%	1.26%	**1.53%**
		@4	1.65%	0.63%	2.09%	0.72%	1.03%	1.68%	1.84%	1.51%	2.41%	**2.70%**

Table 6: Performance on temporal link recommendation between the multi-type nodes



(a)

(b)

(c)

Figure 2: Experimental results of the influence with different parameters: the number of candidate meta-path instances n (a), the number of candidate edges z (b), and negative samples (c). The X-axis is epochs, while the Y-axis represents the micro-f1 of link prediction on Aminer.

divide networks into different subgraphs of snapshots, and then aggregate the information of each subgraph, like DySAT [Sankar *et al.*, 2020]. Fortunately, a few methods like HTNE [Zuo *et al.*, 2018], M^{2}DNE [Lu *et al.*, 2019], and MTNE [Huang *et al.*, 2020] simulate the evolution of networks.

Currently, there is an increasing trend that focuses on temporal HIN embedding. Most of such works use meta-path to capture semantics in HIN and divide time into snapshots to obtain node embeddings, like DHNE [Yin *et al.*, 2019], Change2vec [Bian *et al.*, 2019], and DyHNE [Wang *et al.*, 2020]. Besides, HDGAN [Li *et al.*, 2020] takes advantage of time-level attention mechanism to simulate network evolution. Furthermore, a few semi-supervised methods are designed for HIN, such as static method MAGNN [Fu *et al.*, 2020] and dynamic model HDGNN [Zhou *et al.*, 2020], but there is still a lack of deep methods for temporal HINs. Moreover, none of them considers the dynamics of network evolution and multi-types node embeddings simultaneously.

6 Conclusion

In this paper, we propose THINE for investigating the problem of temporal HINs embedding. THINE takes advantage of the candidate meta-path set to capture structures and semantics, and uses the Hawkes process to simulate the evolution of networks simultaneously. Extensive experiments on three real-world temporal HINs show that THINE achieves the best performance in both static and dynamic tasks.

Acknowledgements

The research was supported by National Natural Science Foundation of China (No. 61802140).

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[Bian *et al.*, 2019] Ranran Bian, Yun Sing Koh, Gillian Dobbie, and Anna Divoli. Network embedding and change modeling in dynamic heterogeneous networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 861–864, 2019.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 135–144, 2017.

[Fu *et al.*, 2020] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, pages 2331–2341, 2020.

[Hawkes, 1971] Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[Huang *et al.*, 2020] Hong Huang, Zixuan Fang, Xiao Wang, Youshan Miao, and Hai Jin. Motif-preserving temporal network embedding. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1237–1243, 2020.

[Li *et al.*, 2020] Qiuyan Li, Yanlei Shang, Xiuquan Qiao, and Wei Dai. Heterogeneous dynamic graph attention network. In *Proceedings of the 2020 IEEE International Conference on Knowledge Graph (ICKG)*, pages 404–411. IEEE, 2020.

[Lu *et al.*, 2019] Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S. Yu, and Yanfang Ye. Temporal network embedding with micro-and macro-dynamics. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 469–478, 2019.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of ICLR Workshops Track*, 2013.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710. ACM, 2014.

[Sankar *et al.*, 2020] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 519–527, 2020.

[Sun and Han, 2012] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 990–998, 2008.

[Tang *et al.*, 2015a] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174, 2015.

[Tang *et al.*, 2015b] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *Proceedings of The Web Conferenc 2019*, pages 2022–2032, 2019.

[Wang *et al.*, 2020] Xiao Wang, Yuanfu Lu, Chuan Shi, Ruijia Wang, Peng Cui, and Shuai Mou. Dynamic heterogeneous information network embedding with meta-path based proximity. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[Yin *et al.*, 2019] Ying Yin, Li-Xin Ji, Jian-Peng Zhang, and Yu-Long Pei. Dhne: Network representation learning method for dynamic heterogeneous networks. *IEEE Access*, 7:134782–134792, 2019.

[Zhou *et al.*, 2020] Fan Zhou, Xovee Xu, Ce Li, Goce Trajcevski, Ting Zhong, and Kunpeng Zhang. A heterogeneous dynamical graph neural networks approach to quantify scientific impact. *arXiv preprint arXiv:2003.12042*, 2020.

[Zhu *et al.*, 2018] Dingyuan Zhu, Peng Cui, Ziwei Zhang, Jian Pei, and Wenwu Zhu. High-order proximity preserved embedding for dynamic networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(11):2134–2144, 2018.

[Zuo *et al.*, 2018] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2857–2866, 2018.