World Scientific Annual Review of Artificial Intelligence

Dynamic Meta-path Guided Temporal Heterogeneous Graph Neural Networks

Yugang Ji

Alibaba Group
Hangzhou, China
yugang.jyg@alibaba-inc.com

Chuan Shi

Peng Cheng Laboratory
Shenzhen, China
shichuan@pcl.ac.cn

Yuan Fang

Singapore Management University
Singapore
yfang@smu.edu.sg

Graph Neural Networks (GNNs) have become the *de facto* standard for representation learning on topological graphs, which usually derive effective node representations via message passing from neighborhoods. Although GNNs have achieved great success, previous models are mostly confined to static and homogeneous graphs. However, there are multiple dynamic interactions between different-typed nodes in real-world scenarios like academic networks and e-commerce platforms, forming temporal heterogeneous graphs (THGs). Limited work has been done for representation learning on THGs and the challenges are in two aspects. First, there are abundant dynamic semantics between nodes while traditional techniques like meta-paths can only capture static relevance. Second, different semantics on THGs are often mutually evolved with each other over time, making it more difficult than dynamic homogeneous graph modeling. To address this problem, here we propose the *Dynamic Meta-path guided temporal heterogeneous Graph Neural Networks* (**DyMGNN**). To handle the dynamic semantics, we introduce the concept of dynamic meta-path which is a common base for temporal semantic search engines, and then adopt the temporal importance sampling to extract neighborhoods with temporal bias. Focusing on mutual evolution, we design the heterogeneous mutual evolution attention mechanism, which can model the fine-grained interplay of semantic-level preferences for each node. Extensive experiments on three real-world datasets for node classification and temporal link prediction demonstrate that our method consistently outperforms state-of-the-art alternatives.

Keywords: Temporal heterogeneous graph; graph neural network; dynamic meta-path; temporal importance sampling; heterogeneous mutual evolution attention

1. Introduction

Graphs, such as social networks, e-commerce platforms, and academic graphs, occur naturally in various real-world applications. Graph representation learning, whose goal is to encode high-dimensional non-Euclidean structures into low-dimensional vector space, has shown great popularity in tackling graph analytic problems [1;9;34]. Recently, graph neural networks (GNNs) have profoundly boosted the field of graph representation learning, for their ability to recursively aggregate information from neighborhoods, naturally capturing topological structures and features [10;21;41;52].

However, current GNNs usually assume that graphs are static and homogeneous with same-typed nodes and relations, while in real-world scenarios, nodes are generally associated with different types and dynamically interact with each other in various ways. As shown in Figure 1 (a), there are three types of nodes including authors (A), papers (P), and conferences (C) as well as two kinds of interactions, namely "write" (AP) and "publish" (PC) on an academic graph. Moreover, all timestamps of interactions are recorded as well. These compositions form a typical temporal heterogeneous graph (THG) and contain complex evolution and rich semantics [38;53]. For example, author A_2 contains multiple relations to others like cooperation with A_1 and co-attendance with A_4, indicating different semantics. Besides, A_1 co-operates with A_2 and A_3 at different times, implying his / her evolving research interest.

Obviously, it is promising and necessary to integrate both semantic and dynamic modeling into GNNs to deal with the problem of representation learning on THGs. On the one line, focusing on semantic modeling, the recursive neighborhood aggregation in GNNs has been expanded into heterogeneous message passing in recent works [2;18;42;54;58]. Many of existing heterogeneous GNNs often follow a two-step paradigm: 1) sample neighbors via multiple meta-paths, and 2) hierarchically aggregate and fuse information of the sampled neighbors, so as to preserve the semantics. For instance, by designing several meta-paths [39] like $\mathcal{P}_1$ and $\mathcal{P}_2$ in Figure 1(b) to capture semantics, it is general to respectively gather information from each type of co-attendees (e.g., A_1 and A_3) and co-authors (e.g., A_1, A_3 and A_4) to construct representation of node A_2. However, these works mostly deal with static structures composited with unvarying relations while neglecting the useful temporal information that exists in most applications. On the other line, dynamic GNNs [28;30] have been proposed to exploit the temporal information on graphs. One of the classical paradigms is to divide the global graph into several independent snapshots [19;33] to capture the evolution of snapshot-level node representations. However, such a design of dynamic GNNs is not suitable for modeling heterogeneous graphs with dynamic interactions because of neglecting the rich semantics. In a world, representation learning on THGs has to face the two following essential challenges.

First, how to model the dynamic semantics on THGs? Meta-paths used in heterogeneous GNNs are in the form of a sequence of multiple static relations. When dealing with THGs, semantic modeling via static meta-paths has to suffer from the

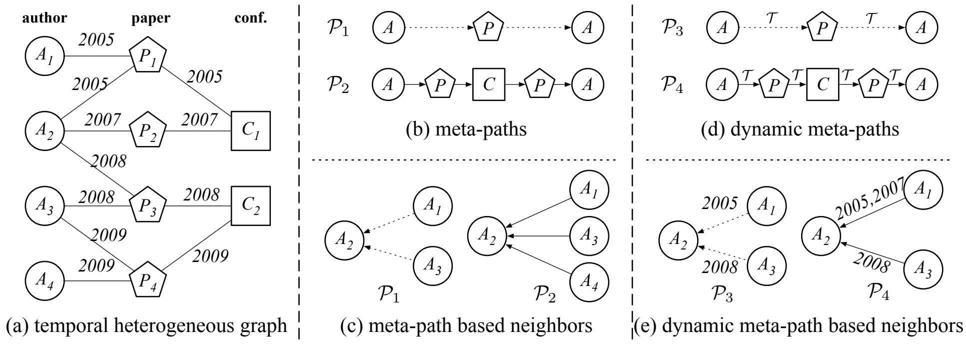


Fig. 1: The toy examples of THGs and the comparison of meta-paths and dynamic meta-paths. (a) is the temporal heterogeneous graph consisting of authors, papers and conferences as well as their dynamic interactions. (b) and (c) respectively showcase the static meta-paths and heterogeneous neighborhoods while (d) and (e) are the corresponding dynamic ones taking temporal bias into consideration.

paradox that current objects can be influenced by future interactions. As shown in Fig 1(b) and (c), based on the meta-path $\mathcal{P}_2$ (i.e., APCPA to describe the "co-attend" of authors), A_4 who was actually unknown to A_2 before 2009, is still assigned into the neighborhoods the temporal dimension is ignored. Moreover, all neighbors with different timestamps are treated the same on static meta-paths, while the latest interacted neighborhoods indeed influence more than historical ones. In other words, traditional meta-path-guided random sampling is likely to extract noisy neighbors rather than temporal representative ones, leading to the limitation in neighborhood aggregation. Some other techniques like meta structure [15] and meta graph [55] are also unsuitable for dynamic semantic modeling due to the same shortcomings.

Second, how to model the mutual evolution of semantics? To derive the evolving representation of nodes, a general idea [26;45;49;51] is to split heterogeneous graphs into several snapshots and then adopt RNNs to capture the evolution of nodes among different snapshots. However, such designs would introduce two major limitations. On the one hand, traditional hard division of snapshots cannot preserve the semantics of nodes between adjacent snapshots, while dynamic interactions on THGs are continuously accumulated. On the other hand, such designs can only model the dynamics of global structures while different semantics of nodes usually mutually evolve with each other. For instance, historical co-authorship (A-P-A) and research interest (A-P-C-P-A) of authors could mutually influence their future co-authorship in different levels, namely the multiple semantic-level evolution. It is challenging to model the complex evolution for a more effective representation of learning of THGs.

Hinged on the above insights, in this paper, we put forward the **D**ynamic **M**eta-path **G**uided temporal heterogeneous **G**raph **N**eural **N**etworks (**DyMGNN**), to effectively learn node representations on THGs. To handle the challenge of dynamic semantic modeling, we first propose dynamic meta-paths to extract the correlations between nodes considering temporal bias. With the assumption that recent neighbors are likely to influence more than historical ones to target nodes, we then propose the dynamic meta-path guided *temporal importance sampling* to sample more representative neighbors for each node. To overcome the challenge of semantic-level mutual evolution, we encode semantic-level node representations with temporal heterogeneous neighborhood aggregation of each dynamic meta-path of each node-wise *soft snapshots* (rather than hard global divisions), and further model the evolution of nodes via the *heterogeneous mutual evolution attention* which captures the evolution of both temporal and semantic levels. Finally, we test our DyMGNN on two academic graphs (i.e., DBLP and Aminer) and the Yelp business graph for node classification and link prediction, and do the key factors analysis to discuss the effectiveness and efficiency of our designs.

In summary, the contributions of this work are shown as follows.

- To the best of our knowledge, we are the first to model the dynamics of semantics for temporal heterogeneous graph representation learning. While existing methods mainly treat heterogeneous and temporal information independently, DyMGNN fully integrates both the dynamics and semantics together to ensure effective representation learning on THGs.
- We propose the novel DyMGNN which not only designs the dynamic meta-paths and temporal importance sampling to model multiple dynamic semantics but also integrates with heterogeneous mutual evolution attention to model the evolution among different semantics on THGs.
- We perform extensive experiments on three real-world datasets, including Aminer, DBLP, and a subset of Yelp business dataset. We compare DyMGNN against the various baselines, and the experimental results show that our model consistently outperforms the state of the arts.

2. Related work

In this section, we summarize the related work including heterogeneous graph neural networks, dynamic graph representation learning, and temporal heterogeneous graph representation learning.

2.1. *Heterogeneous graph neural networks*

Graph neural networks (GNNs)[44], aiming to extend the deep neural networks to deal with graph-structured data, have been widely used for representation learning on graphs[10;21;40]. Kipf *et al.*[21] have proposed the Graph Convolutional Networks (GCN) via a localized first-order approximation of spectral graph convolutions. Hamilton *et al.*[10] further extend the convolutional model to an inductive setting via recursive neighborhood aggregation. The powerful technique of recursive neighborhood aggregation has been commonly used in graph neural networks. Besides, an attention mechanism has been introduced in GNNs to help more effective message passing[41]. However, current GNNs are often confined to homogeneous graphs, failing to preserve the abundant semantics when dealing with real-world graphs where nodes and edges are heterogeneous[1;38]. To keep the semantics as much as possible, heterogeneous graphs[16;37;39] are proposed to model such complex data. Recently, heterogeneous graph neural networks have become popular[2;4;7;13;14;36;54] for representation learning on heterogeneous graphs. The earlier Esmi[36] extracts various semantics between nodes by using multiple meta-paths[39], and adopts factorization machines to learn node representations. Metapath2vec[4] and HIN2vec[7] focus on sampling sequences with meta-paths and input them into a heterogeneous skip-gram model or a neural network to generate node representations. However, these methods typically utilize local structures in an unsupervised manner where the available task supervision cannot be utilized. Recently, Wang *et al.*[42] and Hu *et al.*[13] propose the heterogeneous attention mechanisms to learn the weight of different meta-path based information in a semi-supervised manner. Cen *et al.*[2] and Hu *et al.*[14] propose to aggregate information from meta-relations. However, most of the current methods focus on static heterogeneous graphs, failing to deal with temporal heterogeneous graphs where neighbors are changing and nodes are evolving over time.

2.2. *Dynamic graph representation learning*

There has been significant research in representation learning on dynamic graphs (or called temporal networks) in the past decade[19;29;30;32;33;35]. Taking the temporal bias into consideration, CTDNE[31 50] samples sequences via temporal random walks on homogeneous graphs, and then generate node representations with skip-grams. DANE[22] splits a graph into several snapshots and learns node embeddings based on matrix perturbation theory. However, these methods are unsupervised, failing to leverage task-specific supervision. Recent works prefer to expand classical GNNs to the temporal settings via snapshots. DynGEM[8] utilizes deep auto-encoders between different snapshots to keep the structures. Manessi *et al.*[29] attempt to combine LSTMs and GCNs to model the evolution of nodes and graphs. EvolveGCN[33] further models the evolution of GCN-based parameters among different snapshots. While most of these works utilize sequential snapshots to describe evolving structures, the temporal information and dependence of historical correlations are neglected. Inspired by the Transformers[40], Xu *et al.*[46] theoretically design the temporal function which can map continuous timestamps as the temporal vectors, and then generate evolving representations of nodes based on multi-head attentions. Lu *et al.*[25] design the temporal point process-based M^2DNE to model the time decay effects of past events on current events. However, when dealing with THGs, these methods cannot accurately extract the dynamic semantics and mutual evolution of multiple interactions.

2.3. *Temporal heterogeneous graph representation learning*

Most recently, aware of the rich temporal information on real-world heterogeneous graphs, researchers have attempted to introduce typical techniques of dynamic modeling into heterogeneous GNNs. DHNE [50] integrates Metapath2vec [4] with a temporal random walk on meta-paths from now to historical objects. HGT [14] treats dynamics as extra information and aggregates both encoded temporal vectors and neighborhood attributes for target node representation. Focusing on sequential information, some researchers [12;17;27] propose to respectively split temporal interactions into long-term and short-term information to construct preferences of nodes for recommendation on e-commerce platforms. REGNN [26] constructs several heterogeneous sessions rather than traditional objects as events to learn the sequential evolution. Moreover, to capture the fine-grained evolution of structures, a general solution is to split the whole graph into several snapshots and then model the dynamics between different snapshots. DyHNE [43] and DyHATR [48] respectively utilize Matrix Perturbation Theory and attention mechanism to capture the changes of snapshot-based similarities. HTGNN [24] models both spatial structures and temporal evolution patterns together. More recently, DHGAS [56] attempts to introduce neural architecture search to search the efficient message passing. However, all these methods can only model the evolution of structures but fail to model and utilize the high-level dynamic semantics of graphs. The characteristics of related methods are listed in Table 1.

Thus, we consider this work is valuable and meaningful.

3. Preliminaries

In this section, we introduce related concepts including heterogeneous graph, meta-path, temporal heterogeneous graph, and the designed dynamic meta-path. The main notations are summarized in Table 2.

Definition 1. Heterogeneous Graph: A heterogeneous graph is $\mathcal{G}_{Hete} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ denotes the set of nodes, $\mathcal{E}$ denotes the set of edges. It also contains a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ to label the type of each node, and an edge type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$ to label the type of each edge. $\mathcal{A}$ is the set of node types while $\mathcal{R}$ represents the set of edge types. Notice that $|\mathcal{R}| + |\mathcal{A}| > 2$.

Definition 2. Meta-path: A meta-path $\mathcal{P} : \mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_l} \mathcal{A}_{l+1}$ represents the connection from the source node of type $\mathcal{A}_1$ to the target node of type $\mathcal{A}_{l+1}$ based on the composite relation $\mathcal{R} = \mathcal{R}_1 \circ \mathcal{R}_2 \circ \cdots \circ \mathcal{R}_l$.

As shown in Figure 1(a), the academic graph contains three types of nodes (i.e., $\mathcal{A}$ = {A, P, C}) and two types of relations (i.e., $\mathcal{R}$ = {AP, PC}). In Figure 1(b), $\mathcal{P}_1$ and $\mathcal{P}_2$ are designed to respectively capture the "co-author" and "co-publish" semantics of nodes, and the semantic-level neighbors are extracted in Figure 1(c). However, traditional heterogeneous graphs mainly describe static structures while objects on real-world graphs are often dynamically interacted. Besides, the classical

Table 1: The characteristics of related methods.

Method	Inductive	Heterogeneous	Dynamic	Snapshot	Soft-snapshot
GCN	✓	×	×	×	×
GraphSAGE	✓	×	×	×	×
CTDNE	×	×	✓	×	×
DANE	×	×	✓	✓	×
EvolveGCN	×	×	✓	✓	×
DGNN	✓	×	✓	×	×
DynGEM	✓	×	✓	✓	×
TGAT	✓	×	✓	✓	×
M^{2}DNE	✓	×	✓	✓	×
Esmi	×	✓	×	×	×
Metapath2vec	×	✓	×	×	×
HIN2Vec	×	✓	×	×	×
HAN	×	✓	×	×	×
HGAT	×	✓	×	×	×
GATNE	✓	✓	×	×	×
HGT	✓	✓	✓	×	×
DyHNE	×	✓	✓	✓	×
DyHATR	×	✓	✓	✓	×
HTGNN	✓	✓	✓	✓	×
DHGAS	✓	✓	✓	✓	×
DyMGNN	✓	✓	✓	✓	✓

meta-paths are unable to capture the actual connections. To fully preserve the temporal information, we introduce the concepts of temporal heterogeneous graph and dynamic meta-path as follows.

Definition 3. Temporal Heterogeneous Graph (THG): A temporal heterogeneous graph is $\mathcal{G} = (\mathcal{V}, \mathcal{E}_T, \mathcal{X})$ where $\mathcal{X}$ denotes the attributes of nodes and $\mathcal{E}_T$ denotes the temporal edges. Besides type mapping functions ϕ and ψ, the time mapping function $\mathcal{T} : \mathcal{E}_T \to \mathbb{R}_+$ is to record the timestamp of edges. Notice that the timestamps of static edges are set as 0, and different-typed nodes contain different attributes.

Definition 4. Dynamic meta-path: A dynamic meta-path $\mathcal{P} : \mathcal{A}_1 \xrightarrow{\mathcal{R}_1,\mathcal{T}} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2,\mathcal{T}} \cdots \xrightarrow{\mathcal{R}_l,\mathcal{T}} \mathcal{A}_{l+1}$ is to describe the dynamic semantic from type-$\mathcal{A}_1$ nodes to type-$\mathcal{A}_{l+1}$ nodes. The time mapping function $\mathcal{T}$ is to label dynamic relations, and the dynamic meta-paths with no $\mathcal{T}$ degenerate into the traditional meta-path. Furthermore, given a dynamic meta-path instance $(v_1, v_2, v_3 \cdots, v_{l+1})$, the corresponding timestamps $(t_1, t_2, t_3 \cdots, t_l)$ should satisfy the temporal bias, namely $\forall t_k > 0, t_k \leq min(\{t_n | k < n \leq l, t_n > 0\})$.

Table 2: Notations

Symbols	Descriptions
$\mathcal{G} = (\mathcal{V}, \mathcal{E}_T, \mathcal{X})$	the THG graph
$\phi(\cdot), \psi(\cdot)$	node type and edge type mapping function
s	the snapshot
T, t_s	the current time, the timestamp of snapshot s
N_S	the number of snapshots
$\mathcal{P}$	the meta-path (or dynamic meta-path)
$d_{\phi(\cdot)}$	the dimension of node attributes
d	the dimension of node representations
$\boldsymbol{g}_i$	the representation of node v_i
$\boldsymbol{h}'_{i,\mathcal{P},s}$	the $\mathcal{P}$ representation of v_i at s
$\boldsymbol{h}_{i,\mathcal{P},s}$	the $\mathcal{P}$ temporal representation of v_i at s
$\boldsymbol{W}^{ATT}_{\phi(\cdot)}$	the node type-aware projection matrix
$\boldsymbol{W}^{ATT}_{\psi(\cdot)}$	the edge type-aware projection matrix
$\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$	the projection matrices of queries, keys and values
$\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$	the embedding of queries, keys and values;
$\mathcal{K}(\cdot)$	temporal encoder

For instance, we can further model the academic graph in Figure 1(a) as a THG, maintaining both the heterogeneity and the dynamics of graph structures. Compared with static meta-paths $\mathcal{P}_1$ and $\mathcal{P}_2$, the dynamic meta-paths $\mathcal{P}_3$ and $\mathcal{P}_4$ designed in Figure 1(d) are able to keep the temporal correlations of authors. The corresponding instances in Figure 1(e) meet the temporal constraints as well. Specifically, the neighbors A_1 and A_3 of A_2 based on path $\mathcal{P}_4$ are labeled with the timestamp, while deleting A_4 who is unknown to A_2 before 2008.

4. The Proposed Model

The overall framework of DyMGNN is shown in Figure 2. Specifically, we first introduce the dynamic meta-path guided temporal importance sampling to sample temporal neighbors for dynamic semantic modeling. As nodes evolve over time and are semantic, we then divide THGs into several soft snapshots, which preserve the historical connections to the given snapshots. Furthermore, DyMGNN effectively aggregates semantic-level temporal embedding of nodes (e.g., $\boldsymbol{h}_{i,\mathcal{P},s}$ from path $\mathcal{P}$ at snapshot s) via attention mechanism and temporal encoding. Furthermore, the heterogeneous mutual evolution attention is designed to capture heterogeneous evolving $\boldsymbol{h}_i$. Finally, the constructed node representation $\boldsymbol{g}_i$ is input into a specific supervised task for optimization.

4.1. *Dynamic semantic modeling*

To capture the dynamic semantics, we first design the dynamic meta-paths in Definition 4 with temporal bias constraints, removing the unknown neighbors from neighborhoods. Following the assumption that interactions that happened recently play a more important role in influencing nodes, here we design the dynamic meta-

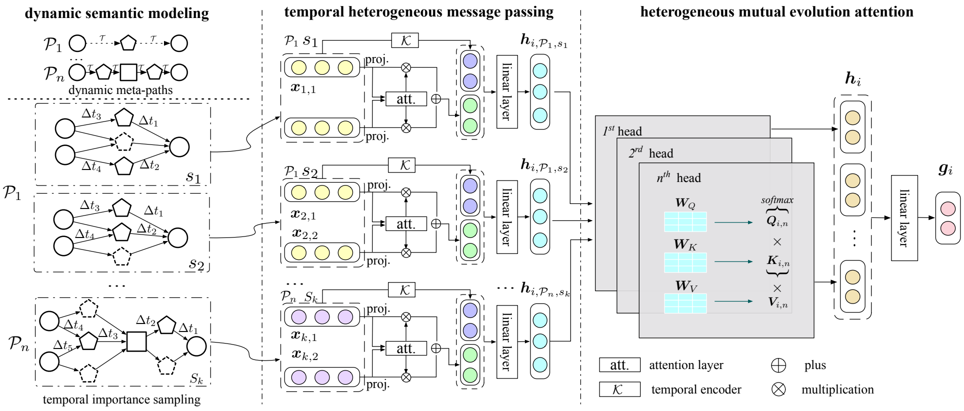


Fig. 2: The framework of our proposed temporal heterogeneous graph neural networks (DyMGNN). (a) is the dynamic semantic modeling on THGs to extract representative neighborhoods of each dynamic meta-path of each snapshot. And then, (b) constructs the corresponding representations with temporal encoding. Finally, (c) models the semantic-level mutual evolution to fully utilize the influence of different semantics to construct evolving node representations for specific tasks.

path guided temporal importance sampling strategies on THGs. Specifically, given a dynamic meta-path $\mathcal{P}: \mathcal{A}_1 \xrightarrow{\mathcal{R}_1,\mathcal{T}} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2,\mathcal{T}} \cdots \xrightarrow{\mathcal{R}_l,\mathcal{T}} \mathcal{A}_{l+1}$, we search the neighbors in reverse and calculate the importance of candidate v_k as follows.

$$p(v_k|v_{k+1},\mathcal{P}) = \begin{cases} 0 & t_k > min_k \\ f(min_k - t_k) & otherwise \end{cases}, \tag{1}$$

where t_k denotes the timestamp of edge $e_{v_k,v_{k+1}}$, $min_k = min(\{t_n|k < n \leq l+1, t_n > 0\}$ denotes the minimum timestamp and $f(\cdot)$ is the activation function (e.g., softmax) to evaluate the importance of v_k with timestamp t_k. We further sample dynamic meta-path guided neighbors according to $p(v_k|v_{k+1},\mathcal{P})$, called temporal importance sampling (TIS). As is demonstrated, the static neighborhoods are randomly sampled, and TIS degenerates into random sampling if all interactions are unvarying. The details are summarized in Algorithm 1.

4.2. *Temporal heterogeneous message passing*

With the heterogeneous structures of THGs continuously changing, dynamic interactions may result in the different representations of nodes at different times. To learn the representation of nodes at each time, we propose the soft snapshots to extract personal structures of each target node according to their start interactions of the given dynamic meta-paths, rather than the global divisions in [30;43;46]. It is difficult to establish a uniform standard to divide THGs since different dynamic

Algorithm 1: Dynamic meta-path guided TIS strategy.

Input: THG $\mathcal{G} = (\mathcal{V}, \mathcal{E}_T)$, dynamic meta-path $\mathcal{P}$, sample size n, target node v_{l+1}, current time t;
Output: neighbor set $\boldsymbol{Nbr}_{v_{l+1},\mathcal{P}}$;
$\boldsymbol{Nbr}_{v_{l+1},\mathcal{P}} = \{\}$;
while $|\boldsymbol{Nbr}_{\mathcal{P}}| < n$ **do**
  $k = l$, $min_k = t$;
  while $k > 0$ **do**
    calculate $p(v_k|v_{k+1}, \mathcal{P})$ based on (1);
    sample v_k according to $p(v_k|v_{k+1}, \mathcal{P})$;
    $min_k \leftarrow min(t_k, min_k)$, $k \leftarrow k - 1$;
  end
  add v_k into $\boldsymbol{Nbr}_{v_{l+1},\mathcal{P}}$;
end

meta-path-based neighborhoods imply various temporal patterns. Considering that all temporal interactions on THGs are recorded and never going to disappear, we propose to sample from all historical neighbors besides current ones with temporal bias, to keep the dependence of semantics among snapshots.

Taking the dynamic heterogeneity into consideration, we design the temporal heterogeneous message passing for each dynamic meta-path at each snapshot. Given the neighbors $(v_{j1}, v_{j2}, \cdots, v_{jn})$ of node v_i at a snapshot, we adopt the heterogeneous node-level attention mechanism to enhance or weaken neighborhood information, as follows.

$$a'_{i,j} = \sigma[(\boldsymbol{x}_i \boldsymbol{W}^{ATT}_{\phi(v_i)} || \boldsymbol{x}_j \boldsymbol{W}^{ATT}_{\phi(v_j)}) \boldsymbol{W}^{ATT}_{\psi(v_i,v_j)} + b^{ATT}_{\psi(v_i,v_j)}], \tag{2}$$

where $a'_{i,j} \in \mathbb{R}_+$ is the weight of v_j to v_i, $\boldsymbol{x}_i \in \mathbb{R}^{d_{\phi v_i}}$ is the attribute vector of v_i with dimension $d_{\phi v_i}$, $\boldsymbol{W}^{ATT}_{\phi(v_i)} \in \mathbb{R}^{d_{\phi(v_i)} \times d}$ is the type-wise parameters to project attributes into the latent space, $\boldsymbol{W}^{ATT}_{\psi(v_i,v_j)}$ and $b^{ATT}_{\psi(v_i,v_j)}$ are the latent project matrix and bias of type $\psi(v_i, v_j)$ need to learn. And then, we normalize the attention $a_{i,j,s}$ for meta-path $\mathcal{P}$ at snapshot s as

$$a_{i,j,s} = \frac{a'_{i,j}}{\sum_{v'_j \in Nbr(v_i,s,\mathcal{P})} a'_{i,j'}}, \tag{3}$$

where $Nbr(v_i, s, \mathcal{P})$ is the neighborhood of dynamic meta-path $\mathcal{P}$ at snapshot s. The sub-representation $\boldsymbol{h}'_{i,\mathcal{P},s}$ is

$$\boldsymbol{h}'_{i,\mathcal{P},s} = \mathrm{AGG}(\{a_{i,j,s} \cdot \boldsymbol{x}_j \boldsymbol{W}_{\phi(v_j)} | j \in Nbr(v_i, s, \mathcal{P})\}), \tag{4}$$

where $\mathrm{AGG}(\cdot)$ is the pooling function and we select *sum* pooling in this paper. Notice that, as the attributes of different-typed nodes belong to different spaces, we adopt $\boldsymbol{W}_{\phi(x_j)}$ to project all attributes in the same latent space, which is labeled "proj" in Figure 2. Inspired by [46], we construct the embedding of v_i at s^{th} snapshot of meta-path $\mathcal{P}$ with a temporal encoder as

$$\boldsymbol{h}_{i,\mathcal{P},s} = \sigma((\boldsymbol{h}'_{i,\mathcal{P},s} + \mathcal{K}(T - t_s))\boldsymbol{W}_S + b_S), \tag{5}$$

where $\boldsymbol{W}_S \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_S \in \mathbb{R}$ are learn-able parameters, and $\mathcal{K}(\cdot)$ is the temporal encoder,

4.3. *Heterogeneous mutual evolution attention*

As mentioned, given different dynamic meta-paths, there are multiple semantics-level representations of nodes on THGs (e.g., preferences or interests) which mutually evolve over time. To model the evolution of preferences, we propose the heterogeneous mutual evolution attention to detect the potential dependence of mutual evolution of nodes at the semantic level. Given the embedding matrix $\boldsymbol{h}_i$ of node v_i with dimension $N_S \times N_{\mathcal{P}} \times d$ where N_S is the number of snapshots, we respectively calculate $\boldsymbol{Q}_i = \boldsymbol{h}_i\boldsymbol{W}_Q$, $\boldsymbol{K}_i = \boldsymbol{h}_i\boldsymbol{W}_K$ and $\boldsymbol{V}_i = \boldsymbol{h}_i\boldsymbol{W}_V$. And then, All $\boldsymbol{Q}_i, \boldsymbol{K}_i$ and $\boldsymbol{V}_i$ are divided into N_h heads, and the attention $att_{i,n}$ of node v_i of head n is

$$att_{i,n} = softmax(\boldsymbol{Q}_{i,n}^T\boldsymbol{K}_{i,n}/\sqrt{d/n}), \tag{6}$$

and then, we concatenate all sub-embeddings $\boldsymbol{h}_i$ as follows.

$$\boldsymbol{h}_i = ||_{n=0}^{N_h}(att_{i,n} \cdot \boldsymbol{V}_{i,n}). \tag{7}$$

Finally, the node representation is defined as

$$\boldsymbol{g}_i = [\boldsymbol{h_i}||\boldsymbol{x}_i]\boldsymbol{W}_{\phi(v_i),G} + b_{\phi(v_i),G}, \tag{8}$$

where $\boldsymbol{W}_{\phi(v_i),G}$ and $b_{\phi(v_i),G}$ are the learnable type-aware project matrix and bias. Different from self-attention mechanisms[52], the mutual evolution attention takes into account both the dynamics and semantics besides inherent attributes.

4.4. *Optimization objective*

In this paper, we focus on the fundamental tasks, including node classification and link prediction, and the overall cross-entropy loss is defined as

$$\mathcal{L} = \sum_{i,z} -y_{i,z}log(\hat{y}_{i,z}) - (1 - y_{i,z})log(1 - \hat{y}_{i,z})) + \alpha\Omega(\boldsymbol{\Theta}), \tag{9}$$

where y is the ground truth, $\hat{y}$ is the prediction, $\Omega(\boldsymbol{\Theta})$ denotes the regularization of total parameters $\boldsymbol{\Theta}$ to avoid over-fitting and α is the rate. For node classification, z is the label and $\hat{y}_{i,z}$ is the class predicted by $MLP(\boldsymbol{g}_i)$ where $MLP(\cdot)$ denotes the Multi-Layer Perception (MLP). For link prediction, i and z are two nodes and we generate the probability MLP($\boldsymbol{g}_i||\boldsymbol{g}_z$) of the connections between two nodes. Notice that we adopt Adam[20] to minimize Equation (9). The unified framework of DyMGNN is shown in Algorithm 2

Algorithm 2: The proposed DyMGNN.

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}_T)$, meta-paths $\{\mathcal{P}\}$, number of snapshots N_S, max iteration K;
Output: node embedding matrix $\boldsymbol{G}$, parameters $\boldsymbol{\Theta}$;
generate N_S snapshots for each node v_i in $\mathcal{V}$;
initialize parameters $\boldsymbol{\Theta}$;
while $k < K$ *and not in convergence* **do**
    for *each node* v_i *in the mini-batch* **do**
        for *each meta-path* $\mathcal{P}$, *each snapshot* s **do**
            sample $\boldsymbol{Nbr}_{v_i,\mathcal{P},s}$ based on Algorithm 1;
            construct $\boldsymbol{h}'_{i,\mathcal{P},s}$ by Eq. (2)-(4);
            generate $\boldsymbol{h}_{i,\mathcal{P},s}$ by Eq. (5);
        end
        calculate mutual evolution $\boldsymbol{h}_i$ of v_i by Eq. (6)-(7);
        generate node embedding $\boldsymbol{g}_i$ by Eq. (8);
    end
    update $\boldsymbol{\Theta}$ by minimizing $\mathcal{L}(\mathcal{V}, \boldsymbol{\Theta})$;
    $k \leftarrow k + 1$;
end

4.5. *Complexity analysis*

DyMGNN contains two main parts: dynamic semantic and mutual evolution modeling. The complexity of temporal sampling is $\mathcal{O}(\sum_{\mathcal{P}} N_S \times n \times |m_{\mathcal{P}}| \times |\mathcal{V}|)$ where N_S is the number of snapshots, n is the sample size and $|m_{\mathcal{P}}|$ is the average number of path $\mathcal{P}$ instances. It is worth noting that we can adopt parallel computing to significantly reduce computational complexity. The complexity of heterogeneous aggregators is $\mathcal{O}(\sum_{i=0}^{N_{\mathcal{P}}} |\mathcal{V}| \times d_{\mathcal{A}_i} \times d^2)$ and $d_{\mathcal{A}_i}$ is the attribute dimension, while the complexity of heterogeneous mutual evolution attention is $\mathcal{O}(N_S \times N_{\mathcal{P}} \times d^2)$. The total complexity of DyMGNN is linear with the number of nodes $|\mathcal{V}|$, the number of snapshots N_S and the number of dynamic meta-paths $N_{\mathcal{P}}$, which makes the proposed method efficient in large datasets.

5. Experiments

In this section, we evaluate the empirical performance of our DyMGNN on three public real-world temporal heterogeneous graphs, including two academic graphs (Aminer and DBLP) and a user-business dataset (Yelp). More Specifically, we study the effectiveness of our DyMGNN on node classification and link prediction tasks, and then analyze the key factors in DyMGNN to showcase the characteristics of our designs.

Table 3: Statistics of the three public datasets.

Datasets	Node Types	#Nodes	Meta-paths	Times
Aminer	Author (A)	22,942	APA APVPA	16 years
	Paper (P)	18,181		
	Venue (V)	22		
DBLP	Author (A)	4,027	APA APCPA	10 years
	Paper (P)	11,128		
	Conf. (C)	20		
Yelp	User (U)	98,165	BRURB	15 years
	Business (B)	6,615	BTUTB	

5.1. *Datasets and metrics*

In this paper, we test our model on three real-world THGs. The statistical information are summarized in Table 3 and we introduce the details as follows.

1) **Aminer academic graph**[a]. This is also a public benchmark dataset made up of three types of nodes, namely, authors (A), papers (P) and venues (V), as well as two types of temporal interactions, namely “write” (A-P) and “publish” (P-V). We generate node features based on metapath2vec[4].

2) **DBLP academic graph**[b]. This is a public bibliographic dataset consisting of three types of nodes including authors (A), papers (P) and conferences (C), as well as two types of dynamic edges, namely “write” (A-P) and “publish” (P-C). Node features are generated by metapath2vec[4] as well. On DBLP, authors are assigned to four research domains.

3) **Yelp Business graph**[c]. This is a public user review dataset, recording users’ reviews and tips with timestamps. It consists of two types of nodes, namely, users (U) and businesses (B), as well as two types of relations, namely, “review” and “tip” (i.e., BRU and B(T)U relations) between users and businesses. Moreover, users and businesses contain several attributes like the average rating, the number of fans and locations. We extract three categories of businesses, including “Fast Food”, “Sushi” and “American (New) Food” with the corresponding reviews and tips to construct a THG. In addition, the depressed or closed businesses are removed. On this graph, we adopt the meta-paths “BRURB” and “B(T)U(T)B” and their corresponding dynamic meta-paths to capture neighborhoods.

We adopt Micro-F1 and Macro-F1 as the evaluation metrics to quantify the performance for node classification while adopting F1, PR-AUC, and ROC-AUC as the evaluation metrics to analyze the performances for temporal link prediction. Notice that all the above metrics are positively related to the performance of methods.

[a]Available at http://resource.aminer.org/lab-datasets/crossdomain/.
[b]Available at https://dblp.uni-trier.de/db/.
[c]Available at https://www.yelp.com/dataset/.

5.2. *Baselines and experimental settings*

There are three types of eight representative baselines, including dynamic GNNs (DGNN[28], EvolvGCN[33], M^{2}DNE[25] and TGAT[46]), heterogeneous GNNs (HAN[42] and HGT[14]), and dynamic heterogeneous graph approaches (DyHNE[43] and DyHATR[48]). The details are shown as follows.

- **DGNN**[28] is a dynamic GNN which inputs sequential interactions of nodes into a modified LSTM to model the dynamic evolution.
- **EvolvGCN**(E.GCN)[33] is a snapshot-based GNN model which captures evolution of structures by sequentially generating and updating parameters of different snapshots.
- **TGAT**[46] is the first temporal GNNs to analyze the encoding of temporal information. It integrates both multi-head attention mechanisms and temporal encoders to model evolving of nodes.
- **M^{2}DNE**[25] is a temporal point process based model which takes both temporal importance and mutual interactions into consideration.
- **HAN**[42] is a representative heterogeneous GNN model that aggregates information from different-typed neighborhoods by utilizing both semantic-level and node-level attention mechanisms.
- **HGT**[14] is a novel heterogeneous GNN model which introduces a heterogeneous mutual attention which considers both edge types and node types when aggregating information from neighborhoods.
- **DyHNE**[43] is a dynamic heterogeneous graph embedding model which splits heterogeneous graphs into several snapshots and captures the dynamics between snapshots based on matrix perturbation theory.
- **DyHATR**[48] is a dynamic heterogeneous GNN which designs a temporal attention mechanism to capture evolution between HAN-based node representation among different snapshots.

For Aminer, DBLP, and Yelp datasets, We set the time span of each snapshot as 1 year, and the number of dynamic meta-path-based neighbors as 5 for all three datasets. We respectively set the number of snapshots N_S as 10, 6, and 10. For all the baselines and DyMGNN, we set the max iteration as 200, the dimension of nodes $d = 128$, the learning rate as 0.001, and the weight of regularization $\alpha = 0.001$. The size of each mini-batch is set as 128. The remainder parameters of baselines are set following the original papers. For all homogeneous graph models (DGNN, EvolvGCN, M^{2}DNE, and TGAT), we remove the types of edges. As the original DyHATR is to deal with temporal link prediction, we modify it to handle node classification. For the same reason, we modify HAN to ensure temporal link prediction as well. We set the number of snapshots according to the units of timestamps on Aminer, DBLP, and Yelp. We set the dimension of nodes and the size of the mini-batch the same as those in classical GCN and GraphSAGE. We set the max iteration, the weight of regularization, and the learning rate according to the

Table 4: The performance of methods for node classification. The best performance is in bold and the second best is underlined. Relative improvements of DyMGNN w.r.t. the second best are reported as well.

Dataset	Metric	Training	Methods									Improv.
			DGNN	E.GCN	M^2DNE	TGAT	HAN	HGT	DyHNE	DyHATR	DyMGNN	
Aminer	Micro-F1	40%	0.779	0.819	0.826	0.835	0.868	0.872	0.884	0.877	**0.925**	4.6%
		60%	0.795	0.835	0.830	0.841	0.880	0.892	0.895	0.899	**0.925**	2.9%
		80%	0.812	0.861	0.834	0.850	0.901	0.906	0.918	0.907	**0.947**	3.2%
	Macro-F1	40%	0.794	0.814	0.811	0.829	0.855	0.865	0.876	0.872	**0.923**	5.4%
		60%	0.817	0.821	0.828	0.840	0.871	0.889	0.897	0.902	**0.922**	2.2%
		80%	0.834	0.845	0.829	0.846	0.892	0.918	0.913	0.932	**0.944**	1.3%
DBLP	Micro-F1	40%	0.589	0.659	0.686	0.677	0.698	0.693	0.690	0.700	**0.717**	2.4%
		60%	0.623	0.672	0.701	0.680	0.712	0.717	0.702	0.728	**0.739**	1.5%
		80%	0.644	0.679	0.710	0.691	0.724	0.720	0.733	0.726	**0.745**	1.6%
	Macro-F1	40%	0.581	0.632	0.657	0.649	0.666	0.658	0.642	0.671	**0.689**	2.7%
		60%	0.633	0.658	0.670	0.651	0.684	0.686	0.654	0.689	**0.705**	2.3%
		80%	0.652	0.666	0.688	0.670	0.691	0.694	0.692	0.697	**0.711**	2.0%
Yelp	Micro-F1	40%	0.566	0.592	0.601	0.585	0.620	0.628	0.616	0.633	**0.651**	2.8%
		60%	0.572	0.607	0.602	0.590	0.631	0.658	0.625	0.638	**0.672**	2.1%
		80%	0.587	0.619	0.610	0.608	0.644	0.648	0.652	0.641	**0.662**	1.5%
	Macro-F1	40%	0.540	0.577	0.569	0.545	0.600	0.609	0.607	0.609	**0.621**	2.0%
		60%	0.555	0.582	0.570	0.570	0.610	0.616	0.615	0.612	**0.628**	1.9%
		80%	0.563	0.590	0.579	0.572	0.618	0.632	0.629	0.633	**0.643**	1.6%

performance of the validation set which is sampled from the original training set. Notice that we run all the methods ten times and report the average value.

5.3. *Node classification*

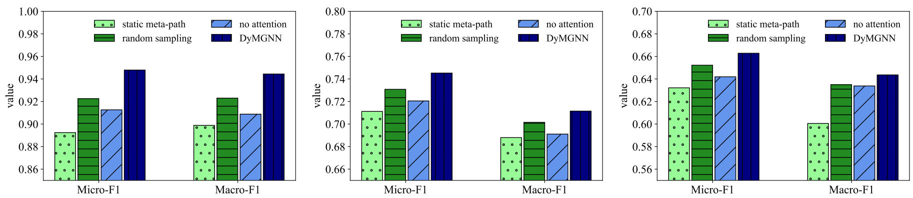
We respectively consider the research domains of authors on Aminer and DBLP datasets and the categories of businesses as labels. The node classification task on Aminer and DBLP is to predict the research area of authors, and the goal on Yelp is to predict the category of businesses. In this task, we train the model with training instances of different scales (i.e., 40%, 60%, and 80%). We report the results in terms of Micro-F1 and Macro-F1 in Table 4 and make the following observations.

First, DyMGNN consistently achieves the best performance on all three datasets with different-scale training instances. The improvement rate to the second best is from 1.5% to 4.9% in the Micro-F1 metric and from 1.3% to 5.4% in the Macro-F1 metric. The advantages of DyMGNN are in two aspects. Compared with dynamic homogeneous approaches (i.e., DGNN, E.GCN, TGAT, and M^2DNE), the advantage is in naturally utilizing semantics rather than single-typed edges. Compared with heterogeneous models including static HAN and HGT as well as dynamic DyHNE and DyHATR, DyMGNN can not only capture the dynamics within semantics but also model the fine-grained mutual evolution between different semantics, resulting in improvements.

Second, dynamic modeling influences and promotes performance significantly. By comparing with static heterogeneous GNNs (i.e., HAN and HAT), DyHNE, DyHATR, and our DyMGNN effectively handle the evolution of nodes and obtain the obvious improvement. Moreover, while DyHATR and DyHNE require hard cutting of snapshots, DyMGNN has the ability to keep all influence of historical semantics.

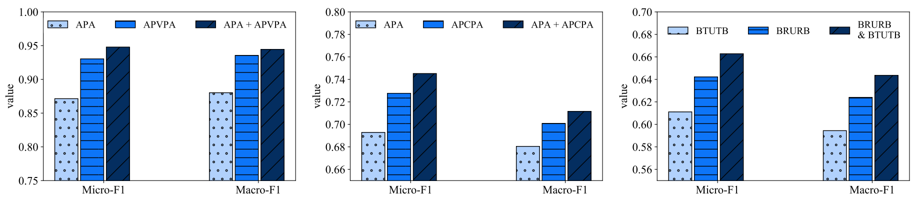
Table 5: The performance of methods for temporal link prediction. The best performance is in bold and the second best is underlined. Relative improvements of DyMGNN w.r.t. the second best are reported as well.

Dataset	Metric	Methods									Improv.
		DGNN	E.GCN	M^2DNE	TGAT	HAN	HGT	DyHNE	DyHATR	DyMGNN	
Aminer	F1	0.744	0.747	0.750	0.772	0.764	0.772	0.789	0.785	**0.799**	1.3%
	PR-AUC	0.769	0.766	0.778	0.800	0.795	0.803	0.815	0.809	**0.820**	0.6%
	ROC-AUC	0.838	0.782	0.848	0.880	0.877	0.882	0.893	0.882	**0.900**	0.8%
DBLP	F1	0.610	0.625	0.606	0.616	0.634	0.639	0.642	0.655	**0.662**	1.1%
	PR-AUC	0.629	0.638	0.636	0.648	0.656	0.652	0.654	0.663	**0.673**	1.5%
	ROC-AUC	0.664	0.669	0.679	0.684	0.683	0.681	0.685	0.690	**0.706**	2.3%
Yelp	F1	0.579	0.618	0.594	0.599	0.605	0.610	0.616	0.626	**0.643**	2.7%
	PR-AUC	0.616	0.629	0.628	0.613	0.647	0.652	0.648	0.657	**0.676**	2.9%
	ROC-AUC	0.635	0.658	0.654	0.647	0.669	0.672	0.664	0.670	**0.685**	1.9%



(a) Aminer (b) DBLP (c) Yelp

Fig. 3: Performance comparison of DyMGNN and its variants on node classification.



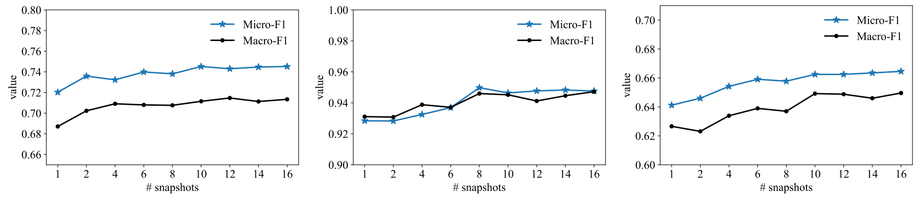
(a) Aminer (b) DBLP (c) Yelp

Fig. 4: The effectiveness of fusing dynamic meta-paths on node classification.

Third, the semantic attention mechanism illustrates the advantages. While DyHNE treats all semantics as equally important, the semantic attention-based DyHATR and our DyMGNN usually achieve better performance due to their ability to evaluate the importance of different semantics.

5.4. *Link prediction*

On the Aminer and DBLP datasets, we focus on predicting the future "co-authorship" of authors. We treat the "APA" links in history and in the latest year



(a) Aminer (b) DBLP (c) Yelp

Fig. 5: The effect of the number of snapshots.

as training samples and test instances respectively. Besides, on the Yelp dataset, we predict the connection of "BRURB" in the latest quarter between businesses and use other "BRURB" links in past quarters for training. We adopt F1, PR-AUC, and ROC-AUC metrics to quantify the performance in Table 5.

As can be seen in Table 5, we can find that DyMGNN performs consistently better than all baselines, and the improvement rate to the second one is from 0.8% to 2.3% in the ROC-AUC metric. Moreover, the dynamic heterogeneous alternatives DyHNE and DyHATR both perform better than other baselines on the three datasets, which verifies the superiority of modeling temporal and heterogeneous information again. In addition, DyHNE outperforms DyHATR on the Aminer dataset while performing worse on the DBLP and Yelp datasets. However, DyMGNN keeps the advantages on all three datasets, indicating the stability of our model.

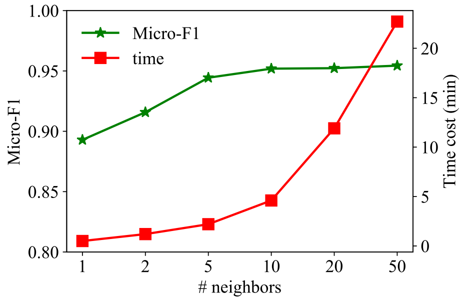
5.5. *Comparison of model variants*

To evaluate the effectiveness of our design choices, we analyze two categories of DyMGNN variants as follows. 1) dynamic semantic modeling: "static meta-path" and "random sampling" where the former samples neighbors overlook temporal bias while the latter randomly samples neighbors on dynamic meta-paths. 2) heterogeneous mutual evolution modeling: "no attention" that models the evolution via GRU rather than heterogeneous mutual evolution attention.

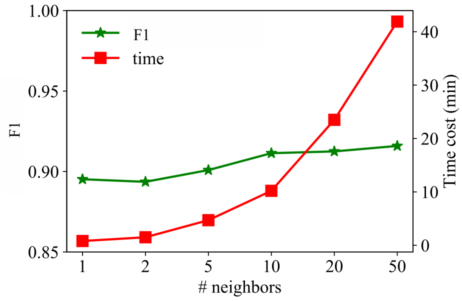
Figure 3 illustrates two main conclusions. (1) DyMGNN performs the best, and the random sampling model achieves better results than the static meta-path-based model, which not only verifies the advantages of dynamic meta-path-guided temporal importance sampling but also indicates the effectiveness of dynamic meta-paths. (2) Compared with the GRU-based no-attention model, DyMGNN models the mutual evolution with temporal encoding rather than the single evolution of sequences, leading to better performance.

5.6. *Analysis of key factors in DyMGNN*

There are three key factors in DyMGNN that may significantly affect the model performance: the types of dynamic meta-paths for semantic modeling, the num-



(a) node classification



(b) link prediction

Fig. 6: The effect of the sample size in DyMGNN on Aminer dataset.

ber of soft snapshots for modeling evolution, and the sample size of neighbors for neighborhood aggregation.

In Figure 4, we demonstrate the contribution from different types of dynamic meta-paths by gradually adding meta-paths for DyMGNN on three datasets. Taking Figure 4(a) as an example, there are two kinds of dynamic meta-paths on Aminer, namely, “APA” and “APVPA”, and we test the performance of DyMGNN for each dynamic meta-path and they both. Obviously, the performance by utilizing both “APA” and “APVPA” is the best, implying the effectiveness of integrating various dynamic meta-paths. Furthermore, we can find that the performance of DyMGNN with the “APVPA”, “APCPA” and “BRURB” paths in three datasets respectively is the second best. This phenomenon is reasonable since these meta-paths contain more information, or even own all neighbors on the other paths as the subsets.

Next, in Figure 5, we focus on detecting the influence of the number of snapshots. Here we vary the snapshots from 1 to 16 for all datasets. Generally, the performance of DyMGNN continuously but slowly improves, since more in-depth patterns can be exploited in more snapshots.

Furthermore, we also showcase the efficiency of DyMGNN by adjusting the sample size of neighbors in Figure 6. We can find that, with the sample size increasing, the performance gradually improves to be stable, meanwhile, the time cost linearly increases. It indicates that the sampling strategies in DyMGNN are able to effectively and stably sample useful neighbors. Besides, we need to set the proper sample size to balance effectiveness and efficiency.

6. Conclusion

In this paper, we address the problem of representation learning on temporal heterogeneous graphs by making full use of both heterogeneous and temporal structure information. We design a novel graph neural network model called DyMGNN. In this model, to overcome the challenges in dynamic semantic and semantic-level mutual evolution modeling, we respectively introduce dynamic meta-path and temporal importance sampling to capture dynamic heterogeneous neighborhoods and design the heterogeneous mutual evolution mechanism based on temporal heterogeneous message passing to fully detect the latent influence among multiple semantic-level snapshots. Experimental results on three real-world public datasets demonstrate that DyMGNN consistently outperforms state-of-the-art baselines for both node classification and temporal link prediction.

Temporal heterogeneous graph representation learning still remains an open problem in interaction networks. It is worth considering the changes in interactive attributes such as ratings and reviews and modeling the evolution of emotions by integrating with recommender systems. More future work can be done along this line.

References

1. Cai, H., Zheng, V. W., and Chang, K. C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.
2. Cen, Y., Zou, X., Zhang, J., Yang, H., Zhou, J., and Tang, J. (2019). Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1358–1368. ACM.
3. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.
4. Dong, Y., Chawla, N. V., and Swami, A. (2017). metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 135–144. ACM.
5. Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., and Li, Y. (2019a). Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2478–2486. ACM.
6. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, Y. E., Tang, J., and Yin, D. (2019b). Graph neural networks for social recommendation. pages 417–426. ACM.
7. Fu, T., Lee, W., and Lei, Z. (2017). Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the*

2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017, pages 1797–1806. ACM.

8. Goyal, P., Kamra, N., He, X., and Liu, Y. (2018). Dyngem: Deep embedding method for dynamic graphs. *CoRR*, abs/1805.11273.
9. Hamilton, W. L., Ying, R., and Leskovec, J. (2017a). Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3):52–74.
10. Hamilton, W. L., Ying, Z., and Leskovec, J. (2017b). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.
11. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
12. Hu, L., Li, C., Shi, C., Yang, C., and Shao, C. (2020a). Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management*, 57(2):102142.
13. Hu, L., Yang, T., Shi, C., Ji, H., and Li, X. (2019). Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4820–4829. Association for Computational Linguistics.
14. Hu, Z., Dong, Y., Wang, K., and Sun, Y. (2020b). Heterogeneous graph transformer. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2704–2710. ACM / IW3C2.
15. Huang, Z., Zheng, Y., Cheng, R., Sun, Y., Mamoulis, N., and Li, X. (2016). Meta structure: Computing relevance in large heterogeneous information networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1595–1604. ACM.
16. Ji, Y., Shi, C., Zhuang, F., and Yu, P. S. (2019). Integrating topic model and heterogeneous information network for aspect mining with rating bias. In *Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part I*, volume 11439 of *Lecture Notes in Computer Science*, pages 160–171. Springer.
17. Ji, Y., Yin, M., Fang, Y., Yang, H., Wang, X., Jia, T., and Shi, C. (2020). Temporal heterogeneous interaction graph embedding for next-item recommendation. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part III*, volume 12459 of *Lecture Notes in Computer Science*, pages 314–329. Springer.
18. Ji, Y., Yin, M., Yang, H., Zhou, J., Zheng, V. W., Shi, C., and Fang, Y. (2021). Accelerating large-scale heterogeneous interaction graph embedding learning via importance sampling. *ACM Transactions on Knowledge Discovery from Data*,

15(1):10:1–10:23.

19. Kazemi, S. M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P., and Poupart, P. (2020). Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21:70:1–70:73.
20. Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
21. Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
22. Li, J., Dani, H., Hu, X., Tang, J., Chang, Y., and Liu, H. (2017). Attributed network embedding for learning in a dynamic environment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 387–396. ACM.
23. Li, T., Zhang, J., Yu, P. S., Zhang, Y., and Yan, Y. (2018). Deep dynamic network embedding for link prediction. *IEEE Access*, 6:29219–29230.
24. Liang, T., Ma, L., Zhang, W., Xu, H., Xia, C., and Yin, Y. (2022). Content-aware recommendation via dynamic heterogeneous graph convolutional network. *Knowledge-Based Systems*, 251:109185.
25. Lu, Y., Wang, X., Shi, C., Yu, P. S., and Ye, Y. (2019). Temporal network embedding with micro- and macro-dynamics. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 469–478. ACM.
26. Luo, W., Zhang, H., Yang, X., Bo, L., Yang, X., Li, Z., Qie, X., and Ye, J. (2020). Dynamic heterogeneous graph neural network for real-time event prediction. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3213–3223. ACM.
27. Lv, F., Jin, T., Yu, C., Sun, F., Lin, Q., Yang, K., and Ng, W. (2019). SDM: sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2635–2643. ACM.
28. Ma, Y., Guo, Z., Ren, Z., Tang, J., and Yin, D. (2020). Streaming graph neural networks. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 719–728. ACM.
29. Manessi, F., Rozza, A., and Manzo, M. (2020a). Dynamic graph convolutional networks. *Pattern Recognition*, 97.
30. Manessi, F., Rozza, A., and Manzo, M. (2020b). Dynamic graph convolutional networks. *Pattern Recognition*, 97.
31. Nguyen, G. H., Lee, J. B., Rossi, R. A., Ahmed, N. K., Koh, E., and Kim, S.

(2018a). Continuous-time dynamic network embeddings. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 969–976. ACM.

32. Nguyen, G. H., Lee, J. B., Rossi, R. A., Ahmed, N. K., Koh, E., and Kim, S. (2018b). Dynamic network embeddings: From random walks to temporal random walks. In *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, pages 1085–1092. IEEE.
33. Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., and Leiserson, C. E. (2020). Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5363–5370. AAAI Press.
34. Qiao, Y., Luo, X., Li, C., Tian, H., and Ma, J. (2020). Heterogeneous graph-based joint representation learning for users and pois in location-based social network. *Informantion Processing & Management*, 57(2):102151.
35. Rahman, M., Saha, T. K., Hasan, M. A., Xu, K. S., and Reddy, C. K. (2018). Dylink2vec: Effective feature representation for link prediction in dynamic networks. *CoRR*, abs/1804.05755.
36. Shang, J., Qu, M., Liu, J., Kaplan, L. M., Han, J., and Peng, J. (2016). Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. *CoRR*, abs/1610.09769.
37. Shi, C., Hu, B., Zhao, W. X., and Yu, P. S. (2019). Heterogeneous information network embedding for recommendation. *IEEE Translation on Knowledge and Data Engineering*, 31(2):357–370.
38. Shi, C., Li, Y., Zhang, J., Sun, Y., and Yu, P. S. (2017). A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37.
39. Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T. (2011). Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of VLDB Endow.*, 4(11):992–1003.
40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
41. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
42. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., and Yu, P. S. (2019). Heterogeneous graph attention network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2022–2032. ACM.
43. Wang, X., Lu, Y., Shi, C., Wang, R., Cui, P., and Mou, S. (2020). Dynamic

heterogeneous information network embedding with meta-path based proximity. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

44. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks Learning Systems*, 32(1):4–24.
45. Xie, Y., Ou, Z., Chen, L., Liu, Y., Xu, K., Yang, C., and Zheng, Z. (2021). Learning and updating node embedding on dynamic heterogeneous information network. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pages 184–192. ACM.
46. Xu, D., Ruan, C., Körpeoglu, E., Kumar, S., and Achan, K. (2020). Inductive representation learning on temporal graphs. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
47. Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
48. Xue, H., Yang, L., Jiang, W., Wei, Y., Hu, Y., and Lin, Y. (2020). Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal RNN. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I*, volume 12457 of *Lecture Notes in Computer Science*, pages 282–298. Springer.
49. Yang, L., Xiao, Z., Jiang, W., Wei, Y., Hu, Y., and Wang, H. (2020). Dynamic heterogeneous graph embedding using hierarchical attentions. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 425–432. Springer.
50. Yin, Y., Ji, L., Zhang, J., and Pei, Y. (2019). DHNE: network representation learning method for dynamic heterogeneous networks. *IEEE Access*, 7:134782–134792.
51. Yuan, C., Li, J., Zhou, W., Lu, Y., Zhang, X., and Hu, S. (2020). Dyhgcn: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part III*, volume 12459 of *Lecture Notes in Computer Science*, pages 347–363. Springer.
52. Yun, S., Jeong, M., Kim, R., Kang, J., and Kim, H. J. (2019). Graph transformer networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11960–11970.
53. Zarrinkalam, F., Kahani, M., and Bagheri, E. (2018). Mining user interests over active topics on social networks. *Informantion Processing & Management*,

54(2):339–357.

54. Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. (2019). Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 793–803. ACM.
55. Zhang, W., Fang, Y., Liu, Z., Wu, M., and Zhang, X. (2020). mg2vec: Learning relationship-preserving heterogeneous graph representations via metagraph embedding. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.
56. Zhang, Z., Zhang, Z., Wang, X., Qin, Y., Qin, Z., and Zhu, W. (2023). Dynamic heterogeneous graph attention neural architecture search. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 11307–11315. AAAI Press.
57. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H. (2020). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858.
58. Zheng, J., Li, Q., and Liao, J. (2021). Heterogeneous type-specific entity representation learning for recommendations in e-commerce network. *Informantion Processing & Management*, 58(5):102629.