Time-aware Meta-path Aggregation on Heterogeneous Temporal Graphs

Wei Qin, Yili Wang, Xinqiu Zhang, Tairan Huang, Shuqing Wu, and Jianliang Gao$^{(\boxtimes)}$

School of Computer Science and Engineering, Central South University, Changsha, China {bd19qinwei,wangyili,zhangxinqiu,gaojianliang}@csu.edu.cn

Abstract. Heterogeneous temporal graphs (HTGs) are time-evolving graph structures with various nodes and relations. Meta-paths are high-level semantic relationships composed with multiple types of nodes and relations, which can capture the complex semantic information in HTGs. Their semantics can be aggregated from corresponding instances. In HTGs, the aggregation process of historical meta-path instances have significant reference value for the current aggregation. However, previous meta-path aggregation methods could only aggregate information from a single graph, failing to utilize the historical aggregation information. The number and features of meta-path instances vary over time in HTGs, which makes it difficult to find precise references from the aggregation of historical meta-path instances. To solve this problem, we propose the Time-aware Aggregated Meta-path Search (TiAMS) method, which combines the Time-aware Meta-path Aggregation with a meta-path search framework. Specifically, we categorize meta-path instances into several semantic prototypes and establish a semantic coordinate system with them. We map meta-path instances across different timestamps to a unified semantic coordinate system. After that, we capture the potential temporal dependencies in the aggregation of historical instances to guide the current meta-path aggregation. Extensive experiments demonstrate that TiAMS significantly outperforms state-of-the-art baselines for tasks including link prediction, node classification and node regression.

Keywords: Heterogeneous temporal graph · Meta-path · Graph neural network.

1 Introduction

Heterogeneous temporal graphs (HTGs) are ubiquitous in real-world applications, including social networks, e-commerce networks, academic citation networks, etc [29]. Compared with static heterogeneous graphs, graph structures with multiple types of nodes and relations in HTGs evolve over time and contain rich dynamic information [5]. Multiple types of nodes and relations can be combined to build higher-level semantic relationships called meta-paths [14]. They can be utilized to capture the complex structure and semantic information in HTGs. Meta-path instances are sequences of nodes following the schema defined by a meta-path [20]. For instance, as shown in Fig. 1 (a), the academic citation network, Aminer, contains multiple node types, i.e., Author (A), Paper (P) and Venue (V), as well as multiple relation types, such as Author $\xrightarrow{\text{writes}}$ Paper, Paper $\xrightarrow{\text{publish}}$ Venue. A-P-V is a 2-hop meta-path representing the relationship between authors and venues where their papers are published. Instances of the meta-path A-P-V vary as the graph structure evolves over time. The semantics of meta-paths can be obtained by aggregating the corresponding meta-path instances through weighted summation, which is a commonly used method for meta-path aggregation [7].

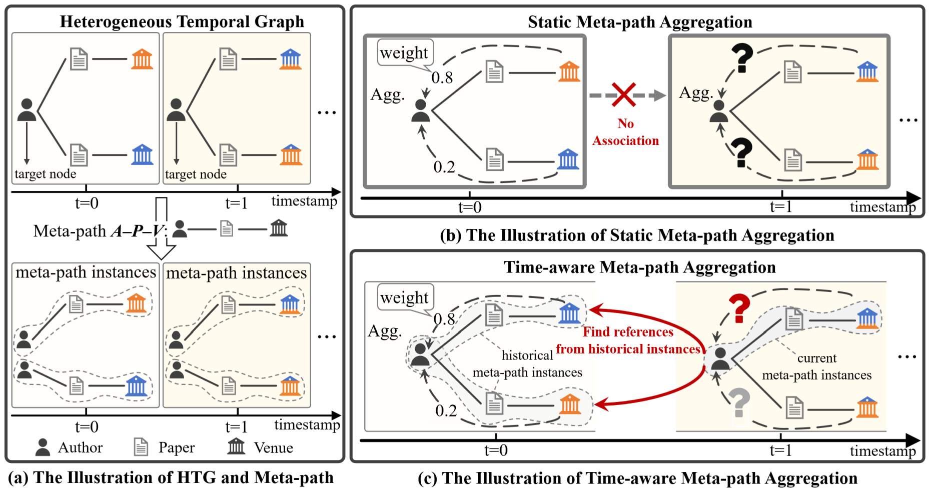
Heterogeneous Temporal Graph
target node
t=0
t=1
timestamp
Meta-path A–P–V:
meta-path instances
Author
Paper
Venue
(a) The Illustration of HTG and Meta-path
Static Meta-path Aggregation
weight
0.8
0.2
Agg.
No Association
(b) The Illustration of Static Meta-path Aggregation
Time-aware Meta-path Aggregation
Find references from historical instances
historical meta-path instances
current meta-path instances
(c) The Illustration of Time-aware Meta-path Aggregation

Fig. 1. The illustration of heterogeneous temporal graphs, meta-paths, and meta-path aggregation. (a) shows a heterogeneous temporal graph including multiple types of nodes and relations, the meta-path Author-Paper-Venue (A-P-V), and the corresponding instances of the meta-path. (b) demonstrates the static meta-path aggregation methods, where the meta-path aggregations across different snapshots are not associated when aggregating on HTGs. (c) describes the time-aware meta-path aggregation. The challenge is how to find references from historical instances for current meta-path instances.

Graph representation learning is the process of converting the raw graph data into low-dimensional vectors while preserving intrinsic graph properties [8, 24]. Recently, many heterogeneous graph neural networks (HGNNs) based on meta-paths have been proposed and have achieved excellent results in heterogeneous graph representation learning [7, 25, 28]. These models capture semantic information with various meta-path aggregation methods. For instance, HAN [25] uses attention mechanism to aggregate neighborhood information for capturing semantic information of meta-paths. MAGNN [7] uses attention mechanisms to aggregate meta-path instances, capturing the semantic information of the meta-paths. SeHGNN [28] uses a light-weight mean aggregator to compute the semantics of meta-paths simply and efficiently.

However, all of these meta-path aggregation methods are proposed specifically for static heterogeneous graphs. They can be collectively designated as static meta-path aggregation methods. In HTGs, these static meta-path aggregation methods can only compute neighbor aggregations independently on each snapshot. As shown in Fig. 1 (b), there is no association among the static meta-path aggregation in each snapshot within HTGs. Nevertheless, the snapshots in HTGs are correlated rather than independent. The aggregation process of historical meta-path instances have significant reference value for the aggregation of current instances. For example, if some meta-path instances have been of high importance in historical meta-path aggregations, similar meta-path instances tend to maintain the high-level of importance in subsequent aggregations. Considering the aggregation process of historical instances when meta-path instances participate in aggregation is beneficial for graph representation learning.

The challenge in utilizing the historical aggregation information is finding precise references from the aggregation of historical meta-path instances. As illustrated in Fig. 1 (c), referring to the aggregation process of historical meta-path instances is crucial when current instances participate in the aggregation process. In HTGs, the graph structure changes over time, which leads to variations in both number and features of meta-path instances. Therefore, there are significant differences in the aggregation process across different snapshots. It implies that information such as weights from the aggregation of meta-path instances in one snapshot cannot be directly utilized in others. Moreover, the aggregation of different instances holds different levels of reference value. For example, those with high similarity are more valuable for reference. Therefore, it is essential to make varying degrees of reference to the aggregation processes of each historical instance. These factors make it difficult to identify references from the aggregation of historical instances for the current meta-path aggregation.

To tackle the above-mentioned challenges, we propose the Time-aware Aggregated Meta-path Search (TiAMS) method which combines the Time-aware Meta-path Aggregation with a meta-path search framework. Specifically, we categorize the meta-path instances into several semantic prototypes. Using them as a basis, we construct a semantic coordinate system. Through mapping both historical and current meta-path instances to a unified semantic coordinate system, we can analyze the similarity between current and historical meta-path instances from multiple semantic dimensions. This enables us to capture the potential temporal dependencies in the aggregation process of meta-path instances across various semantic dimensions, finally guiding the participation of current meta-path instances in aggregation. Combining with a progressive sampling algorithm, we implemented the meta-path search in HTGs, avoiding the need for manual design of meta-paths. Extensive experiments on four real-world heterogeneous temporal graph datasets have demonstrated that our proposed method outperforms state-of-the-art baselines in tasks including link prediction, node regression and node classification. Our main contributions are summarized as follows:

- We propose the Time-aware Meta-path Aggregation that utilizes the historical meta-path aggregation information. Through analyzing historical and current meta-path instances across multiple semantic dimensions, we capture the potential temporal dependencies and utilize them to guide the meta-path aggregation.
- We propose the Time-aware Aggregated Meta-path Search (TiAMS) method. To the best of our knowledge, we are the first to propose a meta-path search model on heterogeneous temporal graphs.
- Extensive experiments on real-world datasets demonstrate that our proposed method outperforms current state-of-the-art baselines.

2 Related Works

Meta-path-based Heterogeneous Graph Neural Networks. Because of the advantages of meta-paths in capturing complex semantic information and interpretability in heterogeneous graph analysis [7, 25], many heterogeneous graph neural networks (HGNNs) [1, 10, 18] have been proposed with successful applications. HAN [25] proposes a novel heterogeneous graph neural network, learning the importance of nodes and meta-paths from both node-level and semantic-level attentions. MAGNN [7] uses the intra-meta-path aggregation to incorporate intermediate semantic nodes, and the inter-meta-path aggregation to combine messages from multiple meta-paths. SeHGNN [28] uses a light-weight mean aggregator to pre-compute neighbor aggregation, avoiding repeated neighbor aggregations at each training epoch. The above-mentioned meta-path aggregation methods are all proposed for static heterogeneous graphs, which do not consider the impact that temporal information has on the meta-path aggregation.

Heterogeneous Temporal Graph Neural Networks. Recently, many excellent heterogeneous temporal graph learning models have been proposed. DyHATR [27] introduces attention mechanism to learn heterogeneous information and capture temporal dependencies. HTGNN [6] uses hierarchical attention mechanism to jointly model heterogeneous spatial dependencies and temporal dimensions. DHGAS [29] is a representative heterogeneous temporal graph neural architecture search (NAS) method to automatically discover the optimal heterogeneous temporal graph neural architecture. CasMLN [23] extracts effectively multi-level evolutionary information, and uses large language models to capture the properties of HTGs. These works are based on traditional graph neural networks (GNNs), without utilizing meta-paths to capture complex structural and semantic information. DyMGNN [12] proposes the dynamic meta-path by adding timestamps to the meta-path to capture information across time. Information from different timestamps shares the same attention mechanism, which overlooks the evolution of attention coefficients over time.

Meta-structure Search on Heterogeneous Graphs. Recently, some works have attempted to search for effective meta-structures within heterogeneous graphs. DiffMG [3] search for meta-graphs utilizing the neural architecture search (NAS) [19]. PMMM [17] adopts an efficient differentiable framework to search for a meaningful meta-multigraph, which can capture more flexible and complex semantic relations. LMSPS [16] investigates the importance of different meta-paths and implements the search for long-range meta-paths through a progressive sampling algorithm. The aforementioned works all focus on searching for effective meta-structures in static heterogeneous graphs, while they have not been applied to heterogeneous temporal graphs.

3 Definitions

In this section, we provide several definitions used in the literature including heterogeneous graph, heterogeneous temporal graph, meta-path and meta-path-based neighbor.

Definition 1. Heterogeneous Graph. A heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and the set of edges, respectively. The graph is associated with a node type mapping function $\phi_n : \mathcal{V} \longrightarrow \mathcal{C}_n$ and an edge type mapping function $\phi_e : \mathcal{E} \longrightarrow \mathcal{C}_e$, where $\mathcal{C}_n$ and $\mathcal{C}_e$ are the sets of node types and edge types, respectively, with $|\mathcal{C}_n| + |\mathcal{C}_e| \geq 2$.

Definition 2. Heterogeneous Temporal Graph. A heterogeneous temporal graph is defined as $\mathcal{G} = (\{\mathcal{G}^t\}_{t=1}^{T}, \phi_n, \phi_e)$. $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ denotes the heterogeneous graph snapshot at timestamp t. T denotes the total number of timestamps. Also, $\phi_e : \mathcal{V} \longrightarrow \mathcal{C}_n$ is the node type mapping function and $\phi_e : \mathcal{E} \longrightarrow \mathcal{C}_e$ is the edge type mapping function, where $\mathcal{V} = \bigcup_{t=1}^{T} \mathcal{V}^t$ and $\mathcal{E} = \bigcup_{t=1}^{T} \mathcal{E}^t$. $\mathcal{C}_n$ and $\mathcal{C}_e$ are sets of node types and edge types, respectively, with $|\mathcal{C}_n| + |\mathcal{C}_e| \geq 2$. Each node $v_i \in \mathcal{V}$ is attached with a node type $c_i = \phi_n(v_i) \in \mathcal{C}_n$. Each edge $e_{t \leftarrow s} \in \mathcal{E}$ (e_{ts} for short) is attached with a relation $r_{c_t \leftarrow c_s} = \phi_e(e_{ts}) \in \mathcal{C}_e$ ($r_{c_t c_s}$ for short).

The graph structure of $\mathcal{G}$ can be represented by a series of adjacency matrices $\{A_r^t : r \in \mathcal{C}_e\}_{t=1}^{T}$. For each relation $r_{c_t c_s} \in \mathcal{C}_e$, $A_{c_t c_s}^t \in \mathcal{R}^{|V_{c_t}^t| \times |V_{c_s}^t|}$ is the corresponding adjacency matrix at timestamp t.

Definition 3. Meta-path. A meta-path defines a composite relation of several edge types, represented as $\mathcal{P} \triangleq c_1 \leftarrow c_2 \leftarrow \ldots c_l$ ($\mathcal{P} = c_1 c_2 \ldots c_l$ for short).

Given the meta-path $\mathcal{P}$ of a heterogeneous temporal graph, a meta-path instance p^t of $\mathcal{P}$ is defined as a node sequence in the graph slice at timestamp t following the schema defined by $\mathcal{P}$.

Definition 4. Meta-path-based Neighbor. Given a meta-path $\mathcal{P}$ of a heterogeneous temporal graph, the meta-path-based neighbors $\mathcal{N}_{\mathcal{P},v}^t$ is defined as the set of nodes that connect with node v via meta-path instances of $\mathcal{P}$ at timestamp t. A neighbor connected by two different meta-path instances is regarded as the same nodes in $\mathcal{N}_{\mathcal{P},v}^t$.

4 Methodology

In this section, we propose Time-aware Aggregated Meta-path Search (TiAMS) method, which combines the Time-aware Meta-path Aggregation with a meta-path search framework. The overview of the proposed TiAMS is shown in Fig. 2.

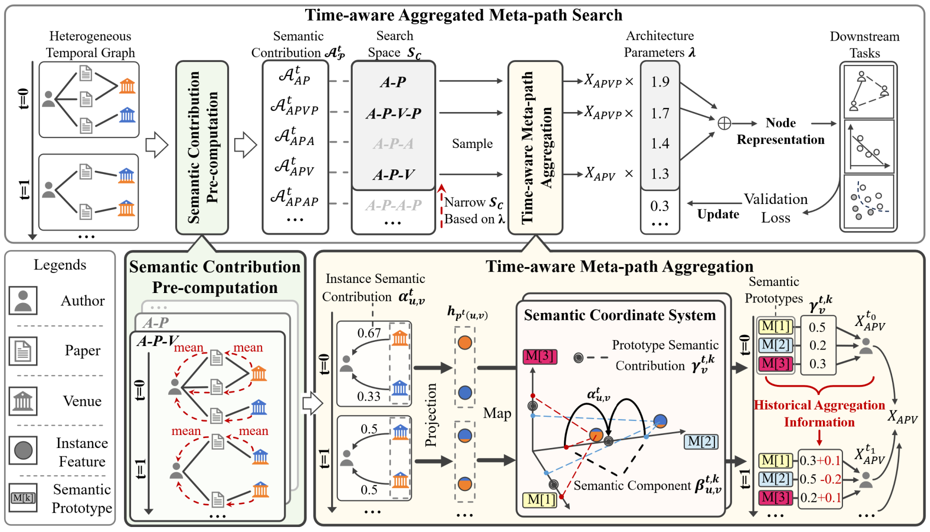
Time-aware Aggregated Meta-path Search
Heterogeneous Temporal Graph
t=0
t=1
Semantic Contribution Pre-computation
Semantic Contribution $\mathcal{A}^t_{\mathcal{P}}$
Search Space S_C
A-P
A-P-V-P
A-P-A
A-P-V
A-P-A-P
Sample
Narrow S_C Based on λ
Time-aware Meta-path Aggregation
Architecture Parameters λ
1.9
1.7
1.4
1.3
0.3
Node Representation
Update
Validation Loss
Downstream Tasks
Legends
Author
Paper
Venue
Instance Feature
Semantic Prototype
mean
Instance Semantic Contribution $\alpha^t_{u,v}$
Projection
Map
Semantic Coordinate System
Prototype Semantic Contribution $\gamma^{t,k}_v$
Semantic Component $\beta^{t,k}_{u,v}$
Semantic Prototypes
Historical Aggregation Information

Fig. 2. The framework of our proposed time-aware aggregated meta-path search (TiAMS). For each meta-path in the search space of the given HTG, we pre-compute the semantic contributions of its instances at each timestamp using a light-weight mean neighbor aggregator prior to training. During the training process, we perform Time-aware Meta-path Aggregation based on these pre-computed semantic contributions to capture the complex semantics of meta-paths. By integrating the semantics from multiple meta-paths, we obtain a representation of the HTG, which is subsequently applied to downstream tasks. We employ a progressive sampling algorithm to automatically search for suitable meta-paths from the search space.

4.1 Semantic Contribution Pre-computation

Weighted summation of meta-path instances is a common meta-path aggregation method. The weight of a meta-path instance reflects its semantic contribution to the target node. Generally, for the target node v, the process of calculating its representation through the meta-path $\mathcal{P}$ at timestamp t can be expressed as:

$$x^t_{\mathcal{P},v} = \sum_{p \in S^t_{\mathcal{P},v}} \alpha_p \cdot h_p, \tag{1}$$

where $S^t_{\mathcal{P},v}$ denotes the set of meta-path instances of $\mathcal{P}$ with the target node v at timestamp t. h_p represents the feature of a meta-path instance p, and α_p is the corresponding weight in the aggregation process, which reflects its semantic contribution to the representation of the target node v.

However, the number of meta-path instances grows exponentially with the length of the meta-path. To avoid the high overhead caused by this increase, we use a light-weight mean aggregator for neighbor aggregation, and calculate the semantic contributions of meta-path instances with the weights of meta-path-based neighbors during the aggregation. Initially, we utilize the projection of the feature of the source node as the feature for each instance. Consequently, at each timestamp, all instances starting from the same source node share the same feature. We denote the set of meta-path instances of $\mathcal{P}$ from the source node u to the target node v at timestamp t by $p^t(u,v)$, which represents a class of meta-path instances with identical features. Given this, the representation of the target node v can also be calculated using $p^t(u,v)$:

$$\begin{aligned} x^t_{\mathcal{P},v} &= \sum_{u \in \mathcal{N}^t_{\mathcal{P},v}} \alpha^t_{u,v} \cdot h_{p^t(u,v)}, \\ \alpha^t_{u,v} &= \sum_{p \in p^t(u,v)} \alpha_p, \\ h_{p^t(u,v)} &= h_p, \forall p \in p^t(u,v). \end{aligned} \tag{2}$$

Here, $N^t_{\mathcal{P},v}$ represents the set of meta-path-based neighbors of target node v based on $\mathcal{P}$ at timestamp t. $h^t_{p(i,j)}$ denotes the feature of the meta-path instance set from source node u to target node v. $\alpha^t_{u,v}$ indicates the total semantic contribution of all instances from u to v at timestamp t.

Considering the projection relationship between the features of meta-path instances and those of meta-path-based neighbors, the semantic contribution of $p^t(u,v)$ is equal to the weight of the source node u during the neighbor aggregation. To calculate $\alpha^t_{u,v}$ efficiently, we use a light-weight mean aggregator to perform neighbor aggregations under the guidance of meta-path $\mathcal{P} = c_1 c_2 \dots c_l$:

$$\mathcal{A}^t_{c_i,c_l} = \begin{cases} \mathcal{F}_{norm}(A^t_{c_i,c_l}), i = l-1 \\ \mathcal{F}_{norm}(A^t_{c_i,c_{i+1}} \cdot \mathcal{A}^t_{c_{i+1},c_l}), i < l-1. \end{cases} \tag{3}$$

Here, $\mathcal{F}_{norm}$ is a row-normalization function applied to ensure that the sum of contributions from all source nodes to a given target node sums up to one. $\mathcal{A}^t_{c_i,c_l}$ represents the weight of the source nodes with type c_l when passed to the nodes with type c_i through the meta-path $\mathcal{P}$.

Specifically, for the meta-path $\mathcal{P}$, $\mathcal{A}^t_{\mathcal{P}}$ is the matrix of semantic contributions, representing the weights of the source nodes when passed to the target nodes through $\mathcal{P}$. $\alpha^t_{u,v}$ is the semantic contribution of the meta-path instance set with the source node u to the target node v at timestamp t in Equation 2. The process of calculating $\alpha^t_{u,v}$ using $\mathcal{A}^t_{\mathcal{P}}$ can be expressed as:

$$\begin{aligned} \mathcal{A}^t_{\mathcal{P}} &= \mathcal{A}^t_{c_1,c_l}, \\ \alpha^t_{u,v} &= \mathcal{A}^t_{\mathcal{P}}[u][v]. \end{aligned} \tag{4}$$

Since the process of compute the semantic contribution is parameter-free, we can pre-compute it before training. This process avoids the repeated aggregation in each training epoch, reducing the complexity.

4.2 Time-aware Meta-Path Aggregation

In this section, we implemented the meta-path aggregation process guided by historical aggregation information, based on the pre-computed semantic contributions. In HTGs, the varying number and features of meta-path instances over time lead to differences in the aggregation processes across snapshots, making it challenging to utilize aggregation information from other snaphots directly. Inspired by MegaCRN [13], we utilize a set of learnable vectors to capture the typical semantics within meta-path instances, which we term as semantic prototypes. The set of these semantic prototypes is represented by $M_{cl} \in R^{m \times d}$, where cl is the type of source nodes, m is the number of prototypes, and d is the dimension of semantic prototypes. We constructed a semantic coordinate system based on these semantic prototypes in M_{cl}, which is used to locate the position of meta-path instance features within the semantic space. The position of meta-path instances in the semantic space reflects the semantic components of their features. We use the following equations to map the features of instances to the semantic coordinate system and analyze the instances:

$$h_{p^t(u,v)} = F^t_{cl,u} \cdot W_{cl} + b_{cl}, \tag{5}$$

$$\beta^{t,k}_{u,v} = \frac{exp(h_{p^t(u,v)} \cdot M^T_{cl}[k])}{\sum_{j=1}^{m} exp(h_{p^t(u,v)} \cdot M^T_{cl}[j])}. \tag{6}$$

Here, $h^t_{p(u,v)}$ denotes the feature of the set of meta-path instances with source node u and target node v at timestamp t. It is obtained by projecting the feature $F^t_{cl,u}$ of the source node u through weight W_{cl} and bias b_{cl}. $M_{cl}[k]$ represents the k-th semantic prototype in the set of semantic prototypes M_{cl}. $\beta^{t,k}_{u,v}$ is the semantic component on the k-th semantic prototype, representing the similarity between the feature $h_{p^t(u,v)}$ of the meta-path instance set $p^t(u,v)$ and the k-th semantic prototype $M_{cl}[k]$. Through mapping meta-path instances into the semantic coordinate system, we achieve a unified representation of the features of meta-path instances from various timestamps using the semantic prototypes.

Subsequently, we replace the features of meta-path instances in the meta-path aggregation process with their unified representation in the form of semantic components on the semantic prototypes. The process of meta-path aggregation is transformed into a weighted summation of semantic prototypes. The semantic contribution of semantic prototypes to the target node v can be expressed as:

$$\gamma^{t,k}_v = \sum_{u \in \mathcal{N}^t_{\mathcal{P},v}} \alpha^t_{u,v} \cdot \beta^{t,k}_{u,v}. \tag{7}$$

Here, $\gamma^{t,k}_v$ denotes the semantic contribution of the k-th prototype to the target node v at timestamp t. $\alpha^t_{u,v}$ and $\beta^{t,k}_{u,v}$ are aforementioned in Equation 2 and Equation 6.

Through converting the semantic contribution of meta-path instances into the contribution of semantic prototypes, the aggregation process of meta-path instances across different snapshots can be consistently described using a uniform set of semantic prototypes that have consist features and numbers.This addresses the issue where meta-path aggregation information cannot be directly utilized by aggregations in other snapshots. And we can assess the reference value of historical meta-path instances from multiple semantic perspectives.

We employ a long short-term memory (LSTM) to capture the potential temporal dependencies in the aggregation process of meta-path instances across various semantic dimensions, and guide the current meta-path aggregation.

$$\varGamma^t_{\mathcal{P}}[v][k] = \gamma^{t,k}_v, \tag{8}$$

$$\varGamma'^t_{\mathcal{P}}, C^t_{\mathcal{P}} = LSTM_{\mathcal{P}}(\varGamma'^{t-1}_{\mathcal{P}}, C^{t-1}_{\mathcal{P}}, \varGamma^t_{\mathcal{P}}). \tag{9}$$

Here, $\varGamma^t_{\mathcal{P}}$ represents the semantic contribution of semantic prototypes to the target nodes at timestamp t. It is composed of $\gamma^{t,k}_v$ mentioned in Equation 7. C^t and C^{t-1} represent the unit states of $LSTM_{\mathcal{P}}$ at time t and $t-1$, respectively. $\varGamma'^t$ and $\varGamma'^{t-1}$ respectively represent the semantic contribution of the semantic prototypes obtained by absorbing historical aggregation information at timestamp t and $t-1$. We use a separate LSTM for each meta-path to capture the temporal dependence unique to each meta-path.

We utilize semantic prototypes and their contributions to target nodes to calculate the semantic vectors at each timestamp. And we use a simple Multilayer Perceptron (MLP) to aggregate semantic vectors at all timestamps.

$$X^t_{\mathcal{P}} = \varGamma'^t_{\mathcal{P}} \cdot M_{cl}, \tag{10}$$

$$X_{\mathcal{P}} = MLP(X^{t0}_{\mathcal{P}}, X^{t1}_{\mathcal{P}}, \ldots, X^T_{\mathcal{P}}), \tag{11}$$

where $\varGamma'^t_{\mathcal{P}}$ reflects both the historical and current semantic contributions of semantic prototypes to the target nodes. M_{cl} is the semantic prototype set. $X^t_{\mathcal{P}}$ is the semantic vectors of all target nodes aggregated at timestamp t through $\mathcal{P}$, under the guidance of historical meta-path aggregation information. T is the number of timestamps, while $X_{\mathcal{P}}$ is the semantic vectors of meta-path $\mathcal{P}$ at all T timestamps.

4.3 Progressive Sampling Meta-path Search

The research [16] shows that not all meta-paths play a positive role in graph representation learning. Inspired by LMSPS, we utilize a progressive sampling algorithm to search meta-paths in HTGs. Specifically, we create a search space for meta-paths denoted as $\mathcal{S}_{\mathcal{P}} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_K\}$, where K is the number of meta-paths. The initial meta-path search space includes all candidate meta-paths. We assign a structure parameters λ_i to each meta-path, indicating its fitness for meta-path representation learning and downstream tasks. The progressive sampling algorithm uses the structure parameters λ_i to narrow the search space from K to M, where M is the number of meta-paths in the final search space. During the training process, we gradually narrow the search space by retaining meta-paths with high structure parameters:

$$S_C = \{k|\lambda_k \geq \tilde{\lambda}_C, \forall 1 \leq k \leq K\}. \tag{12}$$

Here, S_C is a gradually narrowing search space. After several epochs of non-narrowing search space training, C linearly decreases from K to M as the number of training epochs increases. $\tilde{\lambda}_C$ represents the C-th largest structure parameter.

During each training epoch, we randomly sample N meta-paths from the narrowing search space S_c of meta-paths. For each sampled meta-path, we learn its semantic vectors $X_{\mathcal{P}}$ using the Time-aware Meta-path Aggregation. Then we compute the relative strength of each meta-path among all N sampled meta-paths with a Gumbel-softmax [4].

$$\begin{aligned} q_k &= \frac{exp[(\lambda_k + u_k)/\tau]}{\sum_{j\in S} exp[(\lambda_j + u_j)/\tau]}, \\ S &= UniformSample(S_C, N). \end{aligned} \tag{13}$$

Here S is the meta-path set obtained by randomly sampling N meta-paths from S_c. q_k is the relative strength of the k-th meta-path in S. $u_k = -log(-log(U))$ where $U \sim Uniform(0,1)$, and τ is the temperature controlling the continuous relaxation's extent.

We use a simple weighted sum to compute the representation of the target node using the representations of the N sampled meta-paths:

$$Z = \sum_{k=1}^{N} q_k \cdot X_{\mathcal{P}_k}. \tag{14}$$

Here, $X_{\mathcal{P}_k}$ is the semantic vectors of target nodes obtained through the k-th sampled meta-path, and q_k is its corresponding relative strength.

The parameter update in the meta-path search involves a bi-level optimization problem [26].

$$\begin{aligned} &\underset{\lambda}{min}\, \mathcal{L}_{val}(\mathcal{S}, w^*(\lambda), \lambda) \\ &s.t.\; w^*(\lambda) = argmin_w \mathcal{L}_{train}(\mathcal{S}, w, \lambda). \end{aligned} \tag{15}$$

Here S denote the sampled meta-path set. λ is the architecture parameters calculating the meta-path strength. w is the parameters of the downstream task predictor and the Time-aware Meta-path Aggregation. $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ denote the training loss and the validation loss. We alternatively freeze architecture parameters λ when training w on the training set and freeze w when training λ on the validation set.

After narrowing down the number of meta-paths in the search space from K to M, we proceed to obtain a set of N meta-paths with the smallest loss on the validation set through random sampling. Finally, we fix these N meta-paths and fine-tune the model.

5 Experiments

In this section, we evaluate the proposed method through tasks including link prediction, node classification, and node regression.

Table 1. The overall results for different methods for three downstream tasks. The evaluation metrics are in parentheses, and ↑(↓) means that higher(lower) value indicate better results. All methods are trained for five times. The best results are in bold and the second-best results are underlined. "OOM" indicates the method is out-of-memory on NVIDIA RTX 3090 GPU.

Task Metric	**Link Prediction (AUC%)↑**		**Node Classification (F1%)↑**	**Node Regression (MAE)↓**
Dataset	**Aminer**	**OGBN-MAG**	**Yelp**	**COVID-19**
GCN	74.80±0.76	78.06±1.44	37.09±0.41	842±97
GAT	82.61±0.29	79.79±1.87	35.58±1.09	814±95
RGCN	83.13±0.28	81.39±1.76	37.85±0.77	832±82
HGT	79.11±1.30	84.39±1.23	34.42±0.89	801±80
DyHATR	74.31±0.66	89.29±0.51	34.77±0.42	667±51
HGT+	85.58±0.36	OOM	38.33±0.61	OOM
DiffMG	85.11±0.29	87.98±1.25	39.12±0.82	622±61
HTGNN	78.01±0.89	<u>91.12±0.79</u>	36.67±1.10	555±34
DHGAS	<u>87.01±0.71</u>	OOM	<u>40.67±1.81</u>	<u>538±42</u>
TiAMS	**90.25±0.26**	**92.74±0.13**	**43.96±1.10**	**510±11**

5.1 Datasets

We conduct experiments for the link prediction task on two academic datasets, Aminer [11] and OGBN-MAG [6]. For the node classification task, we compare different methods adopting a business review dataset Yelp [11]. For the node regression task, we adopt an epidemic disease dataset COVID-19 [6], an network with constantly changing node features.

- **Aminer** is an academic citation dataset for papers that were published during 1990-2006. The dataset has three types of nodes, and two types of relations.
- **OGBN-MAG** is a HTG extracted from the Microsoft Academic Graph (MAG), consisting of 10 graphs slices spanning from 2010 to 2019. Each graph slice is a heterogeneous graph that contains four types of nodes, and four types of relations among them.
- **Yelp** is a business review dataset, containing user reviews and tips on business. The dataset contains two types of nodes and three types of edges.
- **COVID-19** is an epidemic disease dataset, containing daily new COVID-19 cases in the US. Following HTGNN and DHGAS, we extract graph slices from 05/01/2020 to 02/28/2021. The dataset contains two types of nodes and three types of relations.

5.2 Baselines

Recently, many representative graph neural networks (GNNs) have been proposed. We compare our methods with three classes of state-of-the-art baselines.

– **Static GNNs.** We consider several static homogeneous/heterogeneous GNNs. GCN [15] and GAT [22] work on homogeneous graphs. RGCN [21] and HGT [9] are static heterogeneous GNNs.
– **Dynamic GNNs.** We select three heterogeneous temporal graph models as baselines. DyHATR [27], HGT+ [9] and HTGNN [6] are three representative heterogeneous temporal graph models.
– **Graph NAS.** We also compare our methods with two state-of-the-art graph NAS [2] methods. DiffMG [3] and DHGAS [29] are representative heterogeneous graph and heterogeneous temporal graph NAS methods respectively.

5.3 Link Prediction

We conduct experiments for the link prediction task on two datasets: Aminer and OGBN-MAG. The results are shown in Table 1. We have the following findings. (1) TiAMS achieves the best result on both datasets, i.e., improving the AUC by more than 3% and 1.5% over the most competent baseline, respectively. The results demonstrate that TiAMS can effectively handle the link prediction task on HTGs datasets by capturing semantic information with Time-aware Meta-path Aggregation. (2) DHGAS represents heterogeneous temporal graph neural network that does not utilize meta-structures to capture complex semantic information in its design. DHGAS tailors the most suitable architectures for performing downstream tasks on such graphs. However, its performance still falls short compared to our TiAMS, which highlights the efficacy of incorporating meta-paths in handling HTGs. (3) DiffMG searches for a meta-graph, which can capture complex semantic relations. However, DiffGM does not have the ability to capture dynamic features of meta structures. Compared to DiffMG, TiAMS can capture the potential temporal dependencies in the aggregation of meta-path instances and utilize it to guide the meta-path aggregation, leading to better performance in link prediction tasks.

5.4 Node Classification

We compare different methods for the node classification task adopting the Yelp dataset. From the results also shown in Table 1, we have the following observations. (1) Our proposed method TiAMS reports the best result on Yelp by improving the Macro-F1 score by more than 3%. (2) TiAMS achieves better results than DiffMG on Yelp. It demonstrates the effectiveness of using historical aggregation information to guide meta-structure aggregation in node classification tasks on HTGs. The heterogeneous temporal graph NAS method DHGAS reports the second-best results on Yelp, showing that temporal information plays a crucial role. However, its performance gap compared to TiAMS is still considerable, which demonstrates the effectiveness of meta-paths in node classification tasks.

5.5 Node Regression

For the node regression task, we adopt an epidemic disease dataset COVID-19. We report the results in Table 1 and observe the following findings. (1) TiAMS again achieves the best performance. The results demonstrate that TiAMS can adaptively handle diverse applications of HTGs. (2) For this task, dynamic baselines (i.e., DyHATR, HTGNN and DHGAS) greatly outperforms static methods, showing that modeling temporal information is critical to predicting the COVID-19 cases. (3) Though not considering dynamic information, DiffMG again shows competitive performance, illustrating the great potential of meta-structures. TiAMS can fully utilize such potentials by capturing the semantic information with the Time-aware Meta-path Aggregation.

5.6 Ablation Study

In this section, we verify the role of each module in the model by gradually eliminating the modules in TiAMS. TiAMS$_{w/o\ T}$ denotes TiAMS without the Time-aware Meta-path Aggregation module. TiAMS$_{w/o\ T\&S}$ denotes TiAMS without Time-aware Meta-path Aggregation module and Progressive Sampling Meta-path Search module. The results of the ablation experiment are shown in Table 2. We have the follow two findings. (1) Utilizing the historical meta-path aggregation to guide current meta-path instances participation in aggregation is beneficial for the representation learning in HTGs. It demonstrates that the aggregation process of historical meta-path instances holds significant reference value for the aggregation of current meta-path instances. (2) After eliminating the meta-path search framework, the model performance slightly decreased, indicating that the progressive sampling meta-path module can search for suitable meta-paths.

Table 2. Results for ablation study.

Dataset	Aminer	OGBN-MAG	Yelp	COVID-19
Metric	(AUC%↑)	(AUC%↑)	(F1%↑)	(MAE↓)
TiAMS$_{w/o\ T\&S}$	88.73	88.61	36.72	542
TiAMS$_{w/o\ T}$	89.33	91.23	37.34	538
TiAMS	**90.25**	**92.74**	**43.96**	**510**

To further investigate the reference value of the historical aggregation process, and the impact of utilizing historical aggregation information on semantic expression, on the Aminer, we use attention aggregation from HAN, mean meta-path aggregation from SeHGNN, and Time-aware Meta-path Aggregation to capture the semantics of meta-paths. The comparison of the graph representation ability of several meta-paths under three aggregation methods is shown in Fig. 3. The semantics obtained for each meta-path through Time-aware Meta-path Aggregation significantly outperform those obtained through static meta-path aggregation methods in terms of representational capability. The result further indicates that the aggregation process of historical meta-path instances has significant reference value. Utilizing historical aggregated information to guide the aggregation of meta-paths is beneficial for the representation learning of heterogeneous temporal graphs.

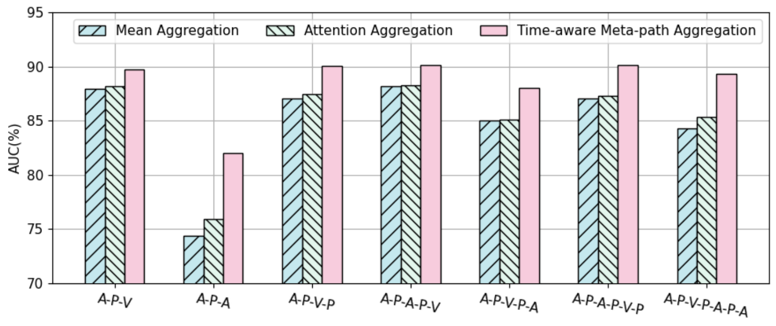

Fig. 3. Results on Aminer of several meta-paths under different aggregation methods.

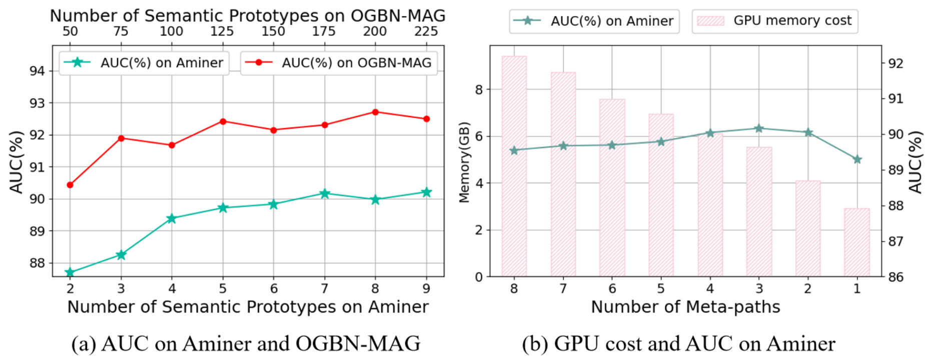

Fig. 4. Results for parameter study.

5.7 Parameter Study

In this section, we investigate the impact of the number of semantic prototypes and meta-paths in the model. As shown in Fig 4 (a), both in Aminer and OGBN-MAG datasets, accuracy gradually increases with the number of semantic prototypes. This indicates that the increase in the number of semantic prototypes is beneficial for improving the ability to analyze the semantics of meta-path instances. As shown in Fig. 4 (b), as the number of meta-paths decreases, the GPU memory cost gradually reduces, and the AUC on Aminer dataset peaks when the number is three. This suggests that not all meta-paths are effective, and the meta-path search framework can reduce the GPU memory cost and improve the model performance. This also indicates that our model can automatically search for meta-paths suitable to downstream tasks.

6 Conclusion

In this section, we have summarized our work. The previous meta-path aggregation methods are static meta-path aggregation, which are proposed for static heterogeneous graphs. To fill this gap, we propose the Time-aware Aggregated Meta-path Search (TiAMS), which combines the Time-aware Meta-path Aggregation with a meta-path search framework. The Time-aware Meta-path Aggregation captures the potential temporal dependencies in the aggregation of historical instances to guide the current meta-path aggregation. The semantics of meta-paths obtained through Time-aware Meta-path Aggregation outperform those obtained through static meta-path aggregation methods. And after a large number of experiments, it has been proven that our proposed model TiAMS performs well in tasks such as link prediction, node classification, and node regression due to the current optimal baseline.

Acknowledgments. The work is supported by the National Natural Science Foundation of China (No. 62272487, 62433016).

References

1. Chang, Y., Chen, C., Hu, W., Zheng, Z., Zhou, X., Chen, S.: Megnn: Meta-path extracted graph neural network for heterogeneous graph representation learning. Knowl.-Based Syst **235**, 107611 (2022)
2. Chen, J., Gao, J., Chen, Y., Oloulade, B.M., Lyu, T., Li, Z.: Auto-gnas: A parallel graph neural architecture search framework. TPDS pp. 3117–3128 (2022)
3. Ding, Y., Yao, Q., Zhao, H., Zhang, T.: Diffmg: Differentiable meta graph search for heterogeneous graph neural networks. In: SIGKDD. pp. 279–288 (2021)
4. Dong, X., Yang, Y.: Searching for a robust neural architecture in four gpu hours. In: CVPR. pp. 1761–1770 (2019)
5. Fan, W., Liu, M., Liu, Y.: A dynamic heterogeneous graph perception network with time-based mini-batch for information diffusion prediction. In: DASFAA. pp. 604–612 (2022)
6. Fan, Y., Ju, M., Zhang, C., Ye, Y.: Heterogeneous temporal graph neural network. In: SDM. pp. 657–665 (2022)
7. Fu, X., Zhang, J., Meng, Z., King, I.: Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In: WWW. pp. 2331–2341 (2020)
8. Guo, J., Li, S., Zhao, Y., Zhang, Y.: Learning robust representation through graph adversarial contrastive learning. In: DASFAA. pp. 682–697 (2022)

9. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: WWW. pp. 2704–2710 (2020)
10. Ji, H., Wang, X., Shi, C., Wang, B., Philip, S.Y.: Heterogeneous graph propagation network. TKDE **35**(1), 521–532 (2021)
11. Ji, Y., Jia, T., Fang, Y., Shi, C.: Dynamic heterogeneous graph embedding via heterogeneous hawkes process. In: ECML PKDD. pp. 388–403 (2021)
12. Ji, Y., Shi, C., Fang, Y.: Dynamic meta-path guided temporal heterogeneous graph neural networks. WSARAI **1**, 1 (2024)
13. Jiang, R., Wang, Z., Yong, J., Jeph, P., Chen, Q., Kobayashi, Y., Song, X., Fukushima, S., Suzumura, T.: Spatio-temporal meta-graph learning for traffic forecasting. In: AAAI. pp. 8078–8086 (2023)
14. Jiang, X., Shen, Y., Wang, Y., Shen, H., Xu, C., Ma, S.: Meta-path based social relation reasoning in a deep and robust way. In: DASFAA. pp. 3–20 (2023)
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
16. Li, C., Guo, Z., He, Q., He, K.: Long-range meta-path search on large-scale heterogeneous graphs. In: NeurIPS (2024)
17. Li, C., Xu, H., He, K.: Differentiable meta multigraph search with partial message propagation on heterogeneous information networks. In: AAAI. pp. 8518–8526 (2023)
18. Li, Y., Jin, Y., Song, G., Zhu, Z., Shi, C., Wang, Y.: Graphmse: Efficient meta-path selection in semantically aligned feature space for graph neural networks. In: AAAI. pp. 4206–4214 (2021)
19. Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G.G., Tan, K.C.: A survey on evolutionary neural architecture search. TNNLS pp. 550–570 (2021)
20. Park, J., Jeong, S., Lee, B.S., Lim, S.: Migtnet: Metapath instance-based graph transformation network for heterogeneous graph embedding. FGCS pp. 390–401 (2023)
21. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC. pp. 593–607 (2018)
22. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
23. Wang, F., Zhu, G., Yuan, C., Huang, Y.: Llm-enhanced cascaded multi-level learning on temporal heterogeneous graphs. In: SIGIR. pp. 512–521 (2024)
24. Wang, L., Shen, Y., Chen, L.: Te-dyge: Temporal evolution-enhanced dynamic graph embedding network. In: DASFAA. vol. 13945, pp. 183–198 (2023)
25. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: WWW. pp. 2022–2032 (2019)
26. Xue, C., Wang, X., Yan, J., Hu, Y., Yang, X., Sun, K.: Rethinking bi-level optimization in neural architecture search: A gibbs sampling perspective. In: AAAI. pp. 10551–10559 (2021)
27. Xue, H., Yang, L., Jiang, W., Wei, Y., Hu, Y., Lin, Y.: Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal rnn. In: ECML PKDD. pp. 282–298 (2021)
28. Yang, X., Yan, M., Pan, S., Ye, X., Fan, D.: Simple and efficient heterogeneous graph neural network. In: AAAI. pp. 10816–10824 (2023)
29. Zhang, Z., Zhang, Z., Wang, X., Qin, Y., Qin, Z., Zhu, W.: Dynamic heterogeneous graph attention neural architecture search. In: AAAI. pp. 11307–11315 (2023)