Social Information Popularity Prediction based on Heterogeneous Diffusion Attention Network

Xueqi Jia
College of Computer Science
Chongqing University
Chongqing, China
jiaxueqi99@126.com

Jiaxing Shang*
College of Computer Science
Chongqing University
Chongqing, China
shangjx@cqu.edu.cn

Linjiang Zheng
College of Computer Science
Chongqing University
Chongqing, China
zlj_cqu@cqu.edu.cn

Dajiang Liu
College of Computer Science
Chongqing University
Chongqing, China
liudj@cqu.edu.cn

Weiwei Cao
Key Laboratory of Flight Techniques and Flight Safety
Civil Aviation Flight University of China
Guanghan, China
ywcao@my.swjtu.edu.cn

Hong Sun
Flight Technology and Flight Safety Research Base
Civil Aviation Flight University of China
Guanghan, China
hanksun@263.net

***Abstract*—Information popularity prediction on social media platforms is a valuable and challenging issue. However, existing studies either neglect the correlation among different cascades, or lack a comprehensive consideration of user behavioral proximity and preference with respect to different messages. In this paper we propose a graph neural network-based framework named HeDAN (heterogeneous diffusion attention network), which comprehensively considers various factors affecting the information diffusion to predict the information popularity more accurately. Specifically, we first construct a heterogeneous diffusion graph with two types of nodes (*user* and *message*) and three types of relations (*Friendship*, *Interaction*, and *Interest*). Among them, *Friendship* reflects the strength of social relationship between users, *Interaction* reflects the behavioral proximity between users, and *Interest* reflects user preference to messages. Next, a graph neural network model with hierarchical attention mechanism is proposed to learn from these relations. Specifically, at the node-level, we utilize the graph attention network to learn the subgraph structure and generate the representations of nodes under each specific relationship. At the semantic-level, we distinguish the importance of different nodes in different relations via multi-head self-attention mechanism. Extensive experimental results on three datasets show the superior performance of our proposed model over the state-of-the-arts.**

***Index Terms*—Information popularity prediction, Graph neural network, Hierarchical attention, Social network analysis**

I. Introduction

Nowadays, social media platforms have greatly promoted the generation and dissemination of information, at the same time intensified the competition among different messages for users' attention. Among many of hot topics [1]–[3] related to social media analysis and mining, the theoretical and practical values of information popularity prediction have been widely recognized by both academia and industry. However, due to the openness of social media platforms and uncertainty of user behaviors, it is challenging to accurately predict the popularity of information on social media platforms.

Considering the importance of graph topology on information diffusion, graph representation-based methods have received more attention in recent years. Previous studies either focused on capturing the topology structure of a single diffusion graph [4]–[6] or mining the social dependencies between active and inactive users [7]–[9], which cannot directly exploit the correlations among different information cascades. However, the simultaneous consideration of all cascades can help to learn the interaction intimacy between users from their historical forwarding behaviors, which is helpful for accurately modeling information diffusion. Furthermore, messages attracting the same group of users are more likely to have similar popularity in the future, which means that establishing the direct links between messages and users can reflect the users' preferences to different messages, thereby benefiting the popularity prediction of messages. Therefore, this paper aims to comprehensively consider the role of social influence, interaction intimacy among users, and user preference to messages on information diffusion, so as to effectively capture predictive factors for more accurate information popularity prediction.

To this end, we propose a graph neural network-based framework named **HeDAN** (**He**terogeneous **D**iffusion **A**ttention **N**etwork), which utilizes a hierarchical attention mechanism to directly learn representations for both users and messages, so as to provide more accurate popularity prediction. Specifically, we first construct a heterogeneous diffusion graph with two types of nodes (*user* and *message*) and three types of relations (*Friendship*, *Interaction*, and *Interest*). Among them, *Friendship* refers to the underlying follower-followee relationships among users on social media platforms, which reflects the influence of users themselves from the perspective of social friendship. *Interaction* refers to the historical forwarding behaviors between users, which reflects the

*Corresponding author: Jiaxing Shang

This work was supported in part by: National Natural Science Foundation of China (Nos. 61966008, U2033213), Open Fund of Key Laboratory of Flight Techniques and Flight Safety, CAAC (Nos. FZ2021KF01, FZ2021KF14)

DOI reference number: 10.18293/SEKE2022-075

proximity among users from the perspective of user behaviors. *Interest* refers to the direct interactions between messages and users, which reflects the attractiveness of messages to users from the perspective of user preference. We creatively combine the above three types of relations to form a heterogeneous diffusion graph. Next, we propose a graph neural network model with hierarchical attention mechanism to learn from this heterogeneous diffusion graph. Specifically, at the node-level, we utilize graph attention network to learn the structure of the subgraphs according to the relationships and characterize the mutual importance of nodes. Then at the semantic-level, we utilize multi-head self-attention mechanism to distinguish the influence of different relationships and users on information diffusion, and finally fuse all kinds of influences to obtain the final representation vector for popularity prediction. Our main contributions and advantages are:

- We consider the correlation among different cascades and creatively construct a heterogeneous diffusion graph which contains three types of relations among users and messages to make information diffusion modeling more comprehensive.
- We propose a graph neural network model with hierarchical attention mechanism to learn from the heterogeneous diffusion graph for more accurate popularity prediction.
- Experimental results on three real-world datasets demonstrate the effectiveness of our proposed model, where the overall prediction errors are significantly reduced.

II. Related Work

In this paper, we will organize related works from the sequential representation-based methods and the graph representation-based methods.

A. Sequential representation-based methods

Sequential representation-based methods usually regard information cascades as dynamic time series and apply recurrent neural networks (RNN) to learn and model the diffusion process. DeepCas [4] utilized random walks to sample the cascade graphs to obtain the sequences of nodes as the input of the bidirectional gated recurrent unit (Bi-GRU). DeepHawkes [10] merged three crucial concepts of Hawkes process, i.e., user influence, self-exciting mechanism, and time decay effect, with RNN to make the modeling process more interpretable. DeepDiffuse [11] employed embedding technique and attention model to learn from the infection timestamp information.

B. Graph representation-based methods

With the development of graph neural networks (GNN) [12], graph representation-based information diffusion studies have received increasing attention in recent years. DeepInf [5] and DiffuseGNN [13] evaluated the social influence of the central user by predicting the user's state (active or inactive) based on the given r-ego network and neighbors' states. CasCN [6] sampled a cascade graph as a series of sequential subcascades and adopted a dynamic multi-directional GCN to learn structural information of cascades. DeepCon+Str [14] proposed two higher-order graphs with cascades as nodes based on content and structural proximity, and learned the higher-order graphs by random walks and semi-supervised language models. CoupledGNN [8] leveraged two specifically designed GNNs, one for node states and the other for influence spread, to model the cascading effect.

III. Methodology

In this section, we present the framework of our HeDAN (Heterogeneous Diffusion Attention Network) model, as illustrated in Fig.1. On the whole, HeDAN consists of the following four major components: (a) Heterogeneous diffusion graph construction module: which extracts the *Interaction* relations among users and *Interest* relations between users and messages from information cascades, and combines them with social relationships to construct a heterogeneous diffusion graph; (b) Node-level attention module: which utilizes graph attention networks to learn the graph structure under each specific relational subgraph, thereby generating node embeddings representing specific relationships; (c) Semantic-level attention module: which utilizes multi-head self-attention mechanism to distinguish the importance of different types nodes in different relationships, and fuse them into the final representation vector; (d) Prediction module: which transforms the final representation vector into the predicted popularity value via a multi-layer perceptron (MLP).

A. Heterogeneous diffusion graph construction module

We first define and construct a heterogeneous diffusion graph which contains two types of nodes (*user* and *message*) and three types of relations (*Friendship*, *Interaction*, and *Interest*). Fig.2 shows how to extract the corresponding relations from the cascade graphs (Fig.2(a)) and the global social graph (Fig.2(b)) to form a heterogeneous diffusion graph (Fig.2(c)).

Given the global social graph $\mathcal{G}_S$, a set of messages $\mathcal{M} = \{m_1, m_2, \cdots, m_d\}$ and the corresponding set of cascade graphs $\mathbf{G}_C = \{\mathcal{G}_{C_1}, \mathcal{G}_{C_2}, \cdots, \mathcal{G}_{C_d}\}$, the heterogeneous diffusion graph is defined as $\mathcal{G}_H = (\mathcal{V}_H, \mathcal{E}_H)$, where node set $\mathcal{V}_H = \mathcal{M} \cup \mathcal{V}_S \cup \mathcal{V}_{C_1} \cup \mathcal{V}_{C_2} \cup \cdots \cup \mathcal{V}_{C_d}$ is the set of all message nodes and user nodes. Each user node is associated with two states, active or inactive. If the user has participated in one of the messages, then it is active, otherwise it is inactive. The edge set $\mathcal{E}_H = \mathcal{E}_{F(u)} \cup \mathcal{E}_{I(u)} \cup \mathcal{E}_{I(m)}$ contains three subsets $\mathcal{E}_{F(u)}$, $\mathcal{E}_{I(u)}$ and $\mathcal{E}_{I(m)}$, where $\mathcal{E}_{F(u)} = \mathcal{E}_S$ is the *Friendship* edge set, $\mathcal{E}_{I(u)} = \mathcal{E}_{C_1} \cup \mathcal{E}_{C_2} \cup \cdots \cup \mathcal{E}_{C_d}$ is the *Interaction* edge set, and $\mathcal{E}_{I(m)} = \{(u_j, m_i) | u_j \in \mathcal{V}_{C_i}, m_i \in \mathcal{M}\}$ is the *Interest* edge set. As shown in Fig.2, the *Interest* edge set $\mathcal{E}_{I(m)}$ corresponding to m_1 in Fig.2(c) is $\{(u_1, m_1), (u_2, m_1), (u_3, m_1)\}$, where an edge (u, m) indicates that user u is interested in message m.

B. Node-level attention module

The purpose of this module is to model non-linear associations between nodes and generate the node representations under each relation type. As shown in Fig.1(b), this module generates three subgraphs from the original graph according

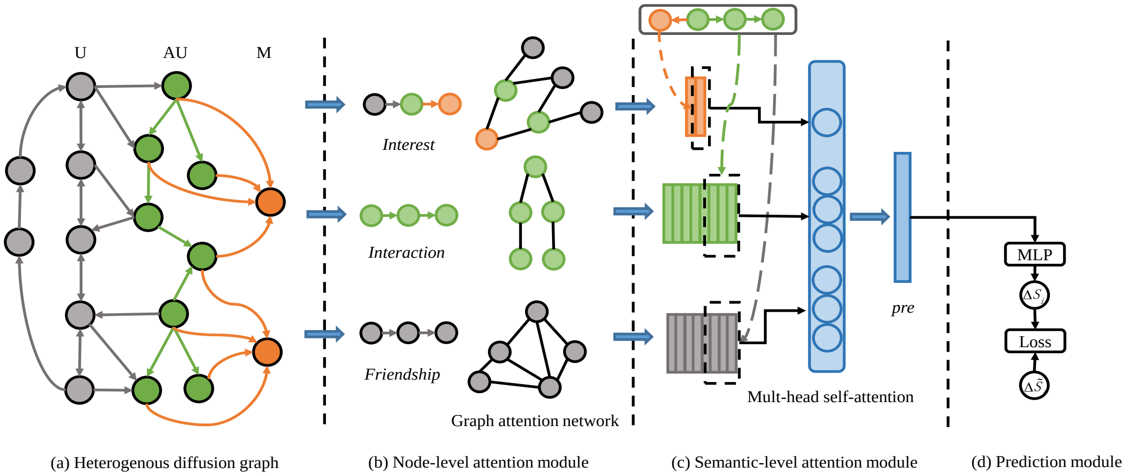

Fig. 1. The framework of HeDAN.

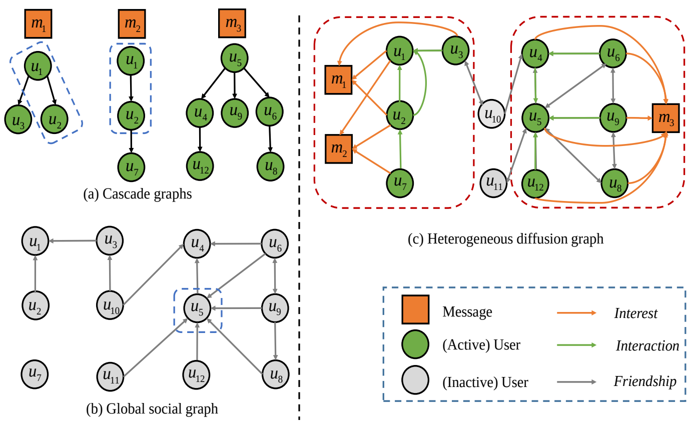

Fig. 2. An example of the heterogeneous diffusion graph. (a) An example of cascade graphs of message m_1, m_2, m_3(marked as orange squares). The edges denote that the user (marked as green circles) reposted a message from another user at a certain timestamp; (b) The global social graph consisting of follower-followee relationships between users; (c) The constructed heterogeneous diffusion graph, which includes two types of nodes (*user* and *message*) and three types of edges (*Friendship*, *Interaction*, and *Interest*). The green circles represent active user nodes, while the gray circles represent inactive user nodes.

to the three types of relationships, and then utilizes the graph attention network which incorporates the importance of neighbors to learn node representations on the subgraphs. The detailed process of this module is as follows:

1) Node feature transformation: Following the works [15] and [16] on heterogeneous graph representation learning, and considering that the feature spaces of message nodes and user nodes are different, we use transformation matrices to project both kinds of nodes into the same feature space. The projection process can be expressed as follows,

$$\mathbf{h'}_i^{(u)} = \mathbf{M}^{(u)} \cdot \mathbf{h}_i^{(u)}, \tag{1}$$

$$\mathbf{h'}_j^{(m)} = \mathbf{M}^{(m)} \cdot \mathbf{h}_j^{(m)}, \tag{2}$$

where $\mathbf{h}_i^{(u)}$ and $\mathbf{h'}_i^{(u)}$ are the original and projected features of the user node i, $\mathbf{h}_j^{(m)}$ and $\mathbf{h'}_j^{(m)}$ are the original and projected features of the message node j, $\mathbf{M}^{(u)} \in \mathbb{R}^{d_u \times d'}$ and $\mathbf{M}^{(m)} \in \mathbb{R}^{d_m \times d'}$ are the transformation matrices of user and message nodes respectively.

2) Friendship subgraph attention layer: We utilize the friendship subgraph attention layer to capture the friendship importance among users and obtain user representations based on friendship relations. The friendship subgraph $\mathcal{G}_{F(u)}$ is a bidirectional homogeneous subgraph generated by the edge-set $\mathcal{E}_{F(u)}$. $\mathcal{G}_{F(u)}$ is bidirectional because each social user plays two roles of sender and receiver in information diffusion. For example, if there is a following relationship between node B and node A, the edge (A, B) indicates that A is the sender and B is the receiver, while the edge (B, A) indicates that A is the receiver and B is the sender. Further, we adopt the graph attention layer to learn the importance $e_{ij}^{F(u)}$ on the subgraph $\mathcal{G}_{F(u)}$, which measures how sender j would contribute to receiver i on friendship. It can be formulated as follows,

$$e_{ij}^{F(u)} = \text{LeakyReLU}(\mathbf{w}_{F(u)}^{T} \cdot \left[\mathbf{h'}_i^{(u)} \parallel \mathbf{h'}_j^{(u)}\right]), \tag{3}$$

where $\mathbf{w}_{F(u)} \in \mathbb{R}^{2d'}$ are the parameterized attention vector for subgraph $\mathcal{G}_{F(u)}$ and $\parallel$ denotes the concatenate operation. Therefore, edge (A, B) and edge (B, A) can still learn different weight values, ie $e_{ij}^{F(u)} \neq e_{ji}^{F(u)}$.

Then we apply softmax function to obtain the normalized weight coefficient $\alpha_{ij}^{F(u)}$, which can be formulated as follows,

$$\alpha_{ij}^{F(u)} = softmax_j(e_{ij}^{F(u)}) = \frac{\exp(e_{ij}^{F(u)})}{\sum_{k \in \mathcal{G}_i^{F(u)}} \exp(e_{ik}^{F(u)})}, \tag{4}$$

where $\mathcal{G}_i^{F(u)}$ is the first-order in-degree neighborhood of user i. For users with a large number of followers, due to its large in-degree value, the influence of each follower is lower on average. For users with few followers but who are active in their own communities, the influence of their neighbors' connections is higher on average.

Finally, the embedding of node i in subgraph $\mathcal{G}_{F(u)}$ can be aggregated by the neighbors' projected features with the corresponding coefficients as follows,

$$\mathbf{z}_i^{F(u)} = \sigma(\sum_{j \in \mathcal{G}_i^{F(u)}} \alpha_{ij}^{F(u)} \cdot \mathbf{h'}_j^{(u)}), \tag{5}$$

where $\mathbf{z}_i^{F(u)}$ is the output of node i for subgraph $\mathcal{G}_{F(u)}$, and $\sigma(\cdot)$ is the activation function.

3) Interaction subgraph attention layer: We utilize the interaction subgraph attention layer to capture the interaction intimacy among activated users and obtain activated user representations based on interaction relations. Similar to $\mathcal{G}_{F(u)}$, we generate the interaction subgraph $\mathcal{G}_{I(u)}$ by the edge-set $\mathcal{E}_{I(u)}$. $\mathcal{E}_{I(u)}$ includes the forwarding relationship among activated users. We process the generated subgraph as a directed homogeneous graph $\mathcal{G}_{I(u)}$ and employ the graph attention layer to learn interaction attention and interaction-based user representations on $\mathcal{G}_{I(u)}$. Similar to that in friendship subgraph, the calculation formulas involved are as follows,

$$e_{ij}^{I(u)} = \mathrm{LeakyReLU}(\mathbf{w}_{I(u)}^T \cdot \left[\mathbf{h'}_i^{(u)} \parallel \mathbf{h'}_j^{(u)}\right]), \tag{6}$$

$$\alpha_{ij}^{I(u)} = softmax_j(e_{ij}^{I(u)}) = \frac{\exp(e_{ij}^{I(u)})}{\sum_{k \in \mathcal{G}_i^{I(u)}} \exp(e_{ik}^{I(u)})}, \tag{7}$$

$$\mathbf{z}_i^{I(u)} = \sigma(\sum_{j \in \mathcal{G}_i^{I(u)}} \alpha_{ij}^{I(u)} \cdot \mathbf{h'}_j^{(u)}). \tag{8}$$

4) Interest subgraph attention layer: We utilize the interest subgraph attention layer to capture the user preferences for messages and to obtain message representations based on interest relations. When generating the interest subgraph, we consider two type of edges, one is the connections between the active users and the message, and the other is the virtual edges from other users and the message. A virtual edge means that if there is a reachable path of length 2 between an inactive user and a message, then a virtual edge is constructed for the inactive user as the source node and the message as the target node. Therefore, the interest subgraph $\mathcal{G}^{I(m)'}$ contains two directed bipartite subgraph, one is $\mathcal{G}_{IA}$ whose edges directly connect active users to messages, and the other is $\mathcal{G}_{IB}$ whose edges connect inactive users who are 2-hop away from the corresponding messages. Further, we train the graph attention network layer on $\mathcal{G}_{IA}$ and $\mathcal{G}_{IB}$ respectively, and finally get $\mathbf{z}_i^{I(m)}$. The formulas involved are as follows,

$$\alpha_{ij}^{IA} = \frac{\exp(\mathrm{LeakyReLU}(\mathbf{w}_{IA}^T \left[\mathbf{h'}_i^{(m)} \parallel \mathbf{h'}_j^{(au)}\right]))}{\sum_{k \in \mathcal{G}_i^{IA}} \exp(\mathrm{LeakyReLU}(\mathbf{w}_{IA}^T \left[\mathbf{h'}_i^{(m)} \parallel \mathbf{h'}_k^{(au)}\right]))}, \tag{9}$$

$$\alpha_{ij}^{IB} = \frac{\exp(\mathrm{LeakyReLU}(\mathbf{w}_{IB}^T \left[\mathbf{h'}_i^{(m)} \parallel \mathbf{h'}_j^{(u)}\right]))}{\sum_{k \in \mathcal{G}_i^{IB}} \exp(\mathrm{LeakyReLU}(\mathbf{w}_{IB}^T \left[\mathbf{h'}_i^{(m)} \parallel \mathbf{h'}_k^{(u)}\right]))}, \tag{10}$$

$$\mathbf{z}_i^{I(m)} = \sigma(\sum_{j \in \mathcal{G}_i^{IA}} \alpha_{ij}^{IA} \cdot \mathbf{h'}_j^{(au)} + \sum_{k \in \mathcal{G}_i^{IB}} \alpha_{ik}^{IB} \cdot \mathbf{h'}_k^{(u)} + \mathbf{h'}_i^{(m)}). \tag{11}$$

C. Semantic-level attention module

The goal of this module is to model the importance of different relationships on information diffusion, so as to obtain a vector representation that integrates the effects of various impacting factors. Through the learning of each relational subgraph by the node attention module, we obtain the friendship-based user representations $\mathbf{Z}^{F(u)}$, the interaction-based user representations $\mathbf{Z}^{I(u)}$, and the interest-based message representations $\mathbf{Z}^{I(m)}$. Now we apply the semantic attention to distinguish the importance of each relationship and generate the final representation by fusing the above representations. The specific process is as follows:

Suppose the popularity of message m is to be predicted, and the active user list within observation window is $[u_A, u_B, u_C]$. First, we query the message representation vector $\mathbf{v}^{I(m)}$ from matrix $\mathbf{Z}^{I(m)}$ according to the id of message m. Next, we query the friendship-based user representation vector list $[\mathbf{v}_A^{F(u)}, \mathbf{v}_B^{F(u)}, \mathbf{v}_C^{F(u)}]$ from matrix $\mathbf{Z}^{F(u)}$ and the interaction-based user representation vector list $[\mathbf{v}_A^{I(u)}, \mathbf{v}_B^{I(u)}, \mathbf{v}_C^{I(u)}]$ from matrix $\mathbf{Z}^{I(u)}$ according to the user id of the list $[u_A, u_B, u_C]$. Finally, we fuse the above vectors as $\tilde{\mathbf{V}} \in \mathbb{R}^{N \times d}$ for semantic attention learning, where N represents the number of vectors in the list. The specific implementation of semantic attention adopts the following multi-head self-attention mechanism,

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{12}$$

$$h_i = Attention(\tilde{\mathbf{V}}\mathbf{W}_i^{\mathbf{Q}}, \tilde{\mathbf{V}}\mathbf{W}_i^{\mathbf{K}}, \tilde{\mathbf{V}}\mathbf{W}_i^{\mathbf{V}}), \tag{13}$$

$$\mathbf{Y} = [h_1; h_2; ...; h_H]\mathbf{W}^{\mathbf{O}}, \tag{14}$$

$$\mathbf{pre} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n, \tag{15}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}^O \in \mathbb{R}^{Hd_k \times d_Q}$; $d_k = d/H$; H is the number of heads of attention module. $\mathbf{Y} \in \mathbb{R}^{N \times d_Q}$ represents the vector list after semantic fusion.

D. Prediction module

The last component of HeDAN is a multi-layer perceptron (MLP) with one final output unit. Given the representation vector pre_i, we calculate the popularity ΔS_i as:

$$\Delta S_i = MLP(\mathbf{pre}_i) \tag{16}$$

Our ultimate task is to predict the final cascade size of message m_i, which can be done by minimizing the following loss function:

$$loss(\Delta S_i, \Delta \tilde{S}_i) = \frac{1}{M}\sum_{i=1}^{M}\left(\log \Delta S_i - \log \Delta \tilde{S}_i\right)^2 \qquad (17)$$

where M is the number of messages, ΔS_i is the predicted popularity for message m_i, and $\Delta \tilde{S}$ is the ground truth.

IV. Experimental Evaluation

A. Dataset

We select three datasets containing both user social graphs and diffusion cascades [17] for experiments. The detailed statistics are presented in Tab.I.

TABLE I
STATISTICS OF THE TWITTER, DOUBAN, AND WEIBO DATASETS.

Datasets	Twitter	Douban	Weibo
#Users	12,627	23,123	2,000,000
#Links	309,631	348,280	12,822,901
#Cascades	3,442	10.662	22,767
#Train Casacdes	2,768	8,529	18,231
#Valid Cascades	345	1,067	2,265
#Test Cascades	344	1,066	2,271

B. Baseline & Evaluation Metric

1) Baseline: To evaluate the effectiveness of HeDAN, we select four methods from the existing deep learning-based methods for comparison. For the sequential representation-based methods, we select DeepCas [4] and DeepHawkes [10]. For the graph representation-based methods, we select DeepCon+Str [14] and CoupledGNN [8].

2) Evaluation metric: Following the existing works [4], [6], [10], we choose MSLE and mSLE as the evaluation metrics of the experiments.

C. Settings

For the baseline methods, the node embedding size of DeepCas, DeepHawkes and DeepCon+Str is set to 64, and all other hyperparameter settings of each model are set to their default values. For our model, the dimension of the hidden units is set to 64. For GAT, the number of heads in the multi-head attention is 8 and the dimension of each head is 8. For multi-head self-attention mechanism, we set the number of heads in the multi-head attention to 4. Our model is implemented by PyTorch. We employ the Adam optimizer with the learning rate set to 0.005 and the weight decay (L2 penalty) set to 0.001. We set the dropout rate to 0.6.

D. Results

1) Overall performance: Tab.II, Tab.III and Tab.IV show the performance of all methods on the three datasets, where the best results are highlighted.

TABLE II
EXPERIMENTAL RESULTS ON TWITTER DATASET.

Dataset	Twitter					
Observation time	1 hour		2 hours		3 hours	
Evaluation metric	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE
DeepCas	1.3770	0.2788	1.3227	0.3092	1.3180	0.2992
DeepHawkes	0.9322	0.1710	0.8953	0.1615	0.8222	0.1552
DeepCon+Str	0.8847	0.1366	0.8521	0.1288	0.7297	0.1243
CoupledGNN	0.7867	0.1301	0.7660	0.1247	0.7045	0.1208
HeDAN	**0.7766**	**0.1263**	**0.7349**	**0.1211**	**0.6606**	**0.1127**

TABLE III
EXPERIMENTAL RESULTS ON DOUBAN DATASET.

Dataset	Douban					
Observation time	1 year		2 years		3 years	
Evaluation metric	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE
DeepCas	1.1293	0.2564	1.0997	0.2369	0.9452	0.2408
DeepHawkes	0.8135	0.1820	0.7335	0.1788	0.7020	0.1665
DeepCon+Str	0.7026	0.1738	0.6854	0.1692	0.6663	0.1673
CoupledGNN	0.6330	0.1696	0.6262	0.1633	0.6035	0.1631
HeDAN	**0.6260**	**0.1676**	**0.6158**	**0.1618**	**0.5927**	**0.1618**

TABLE IV
EXPERIMENTAL RESULTS ON WEIBO DATASET.

Dataset	Weibo					
Observation time	2 hours		4 hours		6 hours	
Evaluation metric	MSLE	mSLE	MSLE	mSLE	MSLE	mSLE
DeepCas	2.2237	1.2260	2.2343	1.2394	2.2053	1.2102
DeepHawkes	1.5527	0.9886	1.5332	0.9727	2.5096	0.9662
DeepCon+Str	1.3724	0.9224	1.3522	0.9103	1.3042	0.8962
CoupledGNN	1.2228	0.7228	1.1055	0.6888	1.0134	0.6835
HeDAN	**1.1139**	**0.6905**	**1.0264**	**0.5707**	**0.9734**	**0.5664**

From Tab.II, Tab.III and Tab.IV, we can see that HeDAN outperforms the state-of-the-art methods by a significant margin. Specifically, we have the following observations: (1) The graph representation-based methods significantly outperform the sequence representation-based methods (over 10% improvement in MSLE on three datasets). This indicates that graph structural information learned by graph representation-based methods is useful for information modeling. (2) HeDAN outperforms DeepCon+Str, with MSLE and mSLE improved by nearly 20% on the Weibo dataset. Unlike DeepCon+Str which ignores fine-grained user-message interactions, HeDAN directly constructs the *Interaction* and *Interest* relationships, which reserves detailed information for users and cascades. Moreover, HeDAN uses the graph representation model such as GAT to learn the node representation, which better captures the internal structure of the cascades compared with the semi-supervised language model. (3) HeDAN outperforms CoupledGNN, with both MSLE and mSLE improved by nearly 10% on the Weibo dataset. This indicates that HeDAN considers co-processing of the cascades to capture the interactions between users and the relationship between cascades, which has a boosting effect in predicting the popularity of cascades.

2) Ablation experiments: To show the relative importance of each module in HeDAN, we perform a series of ablation studies over the key modules of the model. Fig.3 gives the overall performance on several variant methods of HeDAN. We can observe that: The performance of variants (1) (2) (3) shows that all three types of relations have a catalytic effect on information popularity prediction. Variant (4) demonstrates the effectiveness of GAT in node-level modules. The effectiveness of the multi-head self-attention mechanism in the semantic-level module is demonstrated through the variant (5).

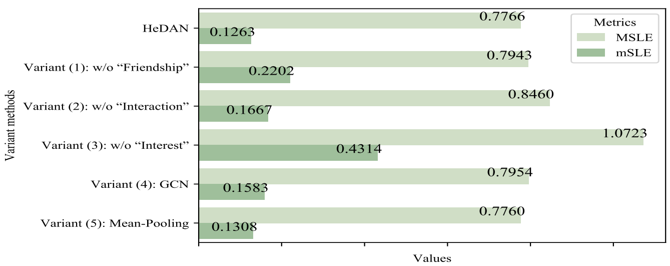

Fig. 3. Results of ablation experiments under a 1-hour observation time window on Twitter dataset.

3) Visualization: In this section, we utilize the t-SNE [18] algorithm to visualize the final prediction representations learned by HeDAN, as shown in Fig.4. We find a clear change in the popularity distribution in Fig.4 (weaker from left to right), which indicates that the latent representations learned by HeDAN are more expressive. Moreover, the distribution of datapoints in Fig.4 is aggregated rather than scattered, which reflects the characteristics of the regression problem.

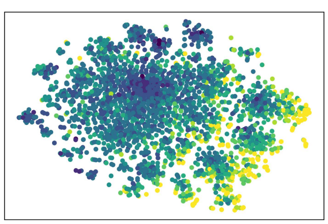

Fig. 4. A visualization of the final prediction representations of cascades on the Twitter dataset. Each datapoint represents a cascade. The darker the datapoint, the higher its popularity value. The locations of datapoints are obtained by performing t-SNE dimensionality reduction on their final prediction representations.

V. Conclusion

In this paper, we studied the information popularity prediction problem on social media platforms. To comprehensively consider various factors that affect information diffusion, we proposed a novel heterogeneous diffusion attention network model to characterize both the user and message representations through hierarchical attention. Specifically, we learned various subgraph structures through node-level attention, and creatively integrated the roles of friendship, user interaction and user preference through semantic-level attention. We conducted experiments on three real-world datasets. The experimental results indicate that our model achieved significant improvements over state-of-the-art models. As for future work, we will extend our model to fine-grained problems such as user-level diffusion behavior prediction. We will also consider model interpretability in our future work.

References

[1] S. Bhattacharya, K. Gaurav, and S. Ghosh, “Viral marketing on social networks: An epidemiological perspective,” *Physica A: Statistical Mechanics and its Applications*, vol. 525, pp. 478–490, 2019.

[2] F. Ducci, M. Kraus, and S. Feuerriegel, “Cascade-lstm: A tree-structured neural classifier for detecting misinformation cascades,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2666–2676.

[3] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xiao, M. Lu, L. Bo, H. Xing, X. Lai, and Y. Ye, “Knowledge-aware coupled graph neural network for social recommendation,” in *35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[4] C. Li, J. Ma, X. Guo, and Q. Mei, “Deepcas: An end-to-end predictor of information cascades,” in *Proceedings of the 26th international conference on World Wide Web*, 2017, pp. 577–586.

[5] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, “Deepinf: Social influence prediction with deep learning,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2110–2119.

[6] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, and F. Zhang, “Information diffusion prediction via recurrent cascades convolution,” in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 770–781.

[7] Z. Wang, C. Chen, and W. Li, “A sequential neural information diffusion model with structure attention,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1795–1798.

[8] Q. Cao, H. Shen, J. Gao, B. Wei, and X. Cheng, “Popularity prediction on social platforms with coupled graph neural networks,” in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 70–78.

[9] S. Molaei, H. Zare, and H. Veisi, “Deep learning approach on information diffusion in heterogeneous networks,” *Knowledge-Based Systems*, vol. 189, p. 105153, 2020.

[10] Q. Cao, H. Shen, K. Cen, W. Ouyang, and X. Cheng, “Deephawkes: Bridging the gap between prediction and understanding of information cascades,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1149–1158.

[11] M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrishnan, “Deepdiffuse: Predicting the’who’and’when’in cascades,” in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 1055–1060.

[12] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020.

[13] F. Zhang, J. Tang, X. Liu, Z. Hou, Y. Dong, J. Zhang, X. Liu, R. Xie, K. Zhuang, X. Zhang *et al.*, “Understanding wechat user preferences and “wow” diffusion,” *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[14] X. Feng, Q. Zhao, and Z. Liu, “Prediction of information cascades via content and structure integrated whole graph embedding.” IJCAI, 2019.

[15] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 793–803.

[16] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous graph attention network,” in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[17] F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, “A survey of information cascade analysis: Models, predictions, and recent advances,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–36, 2021.

[18] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of machine learning research*, vol. 9, no. 11, 2008.