DVCAE: Semi-Supervised Dual Variational Cascade Autoencoders for Information Popularity Prediction

Jiaxing Shang *Member, IEEE,* Xueqi Jia, Xiaoquan Li, Fei Hao, Ruiyuan Li, Geyong Min

***Abstract*—Predicting information popularity in social networks has become a central focus of network analysis. While recent advancements have been made, most existing approaches rely solely on the final cascade size as the primary supervision signal for model optimization. This narrow focus limits the model generalization ability, particularly when faced with highly heterogeneous cascades. Additionally, in real-world scenarios, obtaining detailed social relationships is challenging, complicating effective structural feature learning. To address these issues, this paper proposes a semi-supervised model called Dual Variational Cascade AutoEncoders (DVCAE), which leverages parallel structural and temporal variational autoencoders for enhanced feature learning and popularity prediction. The model first aggregates multiple cascades into a global interaction graph, enabling structural information sharing across cascades. Then, it applies sparse matrix factorization-based graph embedding and graph filtering techniques on global and local cascade graphs respectively, generating initial node embeddings that are insensitive to topological perturbations. After that, two parallel variational autoencoders are designed to generate hidden representations for structural and temporal features respectively, with two self-supervised reconstruction losses integrated into the prediction loss to enrich supervision signals. Extensive experiments conducted on three real-world datasets demonstrate that DVCAE outperforms state-of-the-art models in terms of prediction accuracy.**

***Index Terms*—Popularity prediction, Social network analysis, Graph neural networks, Variational autoencoders.**

I. Introduction

NOWADAYS, social media platforms such as Weibo, Facebook, and Twitter, are playing an unprecedentedly important role in our daily lives [1]. These platforms, which generate huge amounts of data from minute to minute, have greatly facilitated the dissemination of information, resulting in problems such as information overload [2], fake news spreading [3], etc. Information popularity prediction, which refers to predicting the future popularity of a piece of message or news based on its content or early spreading dynamics, is a hot research topic in the research community and has wide applications such as viral marketing [4], content recommendation [5], fake news detection [6], etc. Take viral marketing for example, accurately predicting future information popularity can help decision makers discover potential hot products in advance and make proactive marketing strategies. For social media platforms, knowing which content will be popular in the future is crucial for recommending high-quality material to relevant users, enhancing user loyalty towards the platform. For administrative department, timely monitoring of illegal or harmful information, such as rumors or fake news that would become popular in the future, could help authorities take proactive measures to minimize potential impacts.

Motivation. Recent studies have shown that the effective mining of structural and temporal features from the early spreading dynamics (e.g., cascade graphs, sequential patterns) provides a promising solution to ensure high prediction performance [7]–[9]. However, current studies still face three key challenges yet to be addressed. *Challenge 1): Insufficient structural feature extraction due to unavailability of underlying social networks.* With the increasing privacy concerns and fast growing of social media users, it has become impractical to access the entire underlying social networks, leading to more difficulties in capturing structural features. *Challenge 2): Low generalization capability of prediction models due to cascade heterogeneity and limited supervision signals.* On the one hand, real-world cascades usually exhibit significant heterogeneity and heavy-tailed distribution in terms of popularity [10]. On the other hand, most existing studies rely solely on final popularity (cascade size) as the supervision signal during model training. Consequently, they may shift the model's attention from learning the inherent structural and temporal features, especially when extreme values exist in the supervision signals [11], [12], leading to lower model generalization capability. *Challenge 3: Inefficiency in handling large-scale cascade graphs.* To allow different cascades to collaboratively learn from each other, multiple cascades were usually aggregated, resulting in a large-scale global interaction graph [13], [14]. Therefore, how to efficiently handle such large graphs remains a critical challenge.

Solution. To collectively address the above challenges, this paper proposes a semi-supervised model called **D**ual **V**ariational **C**ascade **A**uto**E**ncoders (**DVCAE**), which leverages parallel variational autoencoders for more effective feature learning and popularity prediction. Specifically, we first construct a large global interaction graph by connecting multiple cascades through the common participants, allow-

Manuscript received XX XX, XXXX. This work was supported in part by the following: National Natural Science Foundation of China (Nos. 62202070, 62477029), Sichuan Science and Technology Program (No. 2025YFHZ0025), Open Fund of Key Laboratory of Dependable Service Computing in Cyber Physical Society, China (No. CPSDSC202207), China Postdoctoral Science Foundation (No. 2022M720567).

J. Shang, X. Jia, X. Li, and R. Li are with the College of Computer Science, Chongqing University, Chongqing, China & Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing, China. E-mail: shangjx@cqu.edu.cn, jiaxueqi99@126.com, li.xiaoquan@foxmail.com, ruiyuan.li@cqu.edu.cn

F. Hao is with the School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an, China. E-mail: feehao@gmail.com

G. Min is with the Department of Computer Science, University of Exeter, Exeter, EX4 4QF, U.K. E-mail: g.min@exeter.ac.uk

Corresponding author: Jiaxing Shang (shangjx@cqu.edu.cn)

ing information sharing across heterogeneous cascades of varying sizes, thus fully capturing the structural information and largely alleviating the issue of cascade heterogeneity. To efficiently handle the large-scale global interaction graph, sparse matrix factorization-based graph embedding technique is applied to obtain the global node embeddings with affordable computational costs. Meanwhile, graph filtering technique is utilized on the local cascade graphs to generate local node embeddings that are less sensitive to small topological perturbations. After that, two parallel variational autoencoders, one for structural learning (with GAT backbone) and the other for temporal learning (with Bi-GRU backbone), are employed to generate hidden representations for structural and temporal node features, followed by respective pooling mechanisms to produce the final cascade-level representations. To further improve the model generalization capability, the self-supervised reconstruction losses from both autoencoders, i.e., the structure learning loss and the temporal learning loss, are creatively integrated into the final popularity prediction loss, thereby enriching the supervision signals for model optimization. Extensive experiments on three real-world datasets demonstrate the superior performance of DVCAE over the state-of-the-art baselines in terms of prediction accuracy.

In summary, this paper makes the following contributions:

- We propose an autoencoder-based semi-supervised deep learning model DVCAE, which employs dual autoencoders for enhanced structural and temporal feature learning without the need of the underlying social network. Additionally, the sparse matrix factorization and graph filtering techniques used in DVCAE allows both efficient handling of large global interaction graph and effective learning of local cascade graphs.
- We creatively integrate the reconstruction losses of the structural and temporal variational autoencoders into the popularity prediction loss. The simultaneous and collective optimization of the three losses allows our model to learn more reliable structural and temporal features, leading to higher generalization ability.
- We conduct extensive experiments on three public datasets and compare DVCAE with 11 representative baselines. The results show that DVCAE significantly outperforms the state-of-the-art baselines in terms of MSLE and MAPE. Ablation experiments further validated the effectiveness of the modules designed in DVCAE. Case studies reveal that DVCAE effectively captures diverse propagation characteristics in complex diffusion scenarios. The source code is publicly available at: https://github.com/jxshang/DVCAE

The rest of this paper is organized as follows: Section II gives a brief review of the related work on information diffusion prediction. In Section III, we introduce the proposed DVCAE model in detail, followed by experimental evaluation in Section IV. In Section V, we conclude our work, discuss its limitations and point out several future directions.

II. Related Work

Currently, deep learning has become the mainstream technique for popularity prediction, so we briefly review the deep learning-based methods from the following three aspects.

A. Methods based on Temporal Learning

These methods mainly focus on learning temporal features for popularity prediction. Cao *et al.* [15] introduced GRU, pooling mechanism and non-parametric time kernel into Hawkes Process, thereby enhancing the interpretability of deep "black-box" models. Wang *et al.* [16] introduced diffusion tree to improve the LSTM network for feature learning. Yang *et al.* [17] used RNN to model microscopic information diffusion processes and transformed it into macroscopic cascade sizes, adopting a reinforcement learning framework to update parameters. Attention mechanism [18] has also been widely used in information diffusion modeling. Wang *et al.* [19] proposed a RNN-based model with improved attention mechanism to capture remote inter-dependencies in the information cascade. Islam *et al.* [20] employed LSTM and attention mechanism to generate content vector from the forwarding time stamp information and node embedding vector for next forwarding user and time prediction. Yang *et al.* [21] established a cascade prediction model based on self-attention mechanism and Convolutional Neural Network (CNN), which uses convolutional operators to alleviate the long-term dependence of propagation sequences. Zhu *et al.* [22] proposed the Cross-Domain Information Fusion Framework (CasCIFF), which exploits multi-hop neighborhood information to generate robust cascade embeddings and incorporates timestamps to capture the evolving patterns of information diffusion. Bao *et al.* [9] comprehensively considered temporal dependencies on dynamic diffusion process by simultaneously modeling the temporal evolution in a separate snapshot and the inherent temporal dependencies among different snapshots.

B. Methods based on Structural Learning

Structural learning-based methods mainly focus on learning structural features for popularity prediction. Qiu *et al.* [23] used the ego network of the target user as input data and modeled the mutual influence of neighbor states with graph convolutional and graph attention networks. Zhang *et al.* [24] further improved the above model by generating egocentric networks with a BFS strategy and learned structural features using a spectral modulation method. Chen *et al.* [25] sampled the cascade graph as a series of sequential subcascades and used dynamic multi-direction GCN to learn the structure information. All of the above models learn local topologies on egocentric networks, while many recent studies learn structures based on global graphs. For example, Cao *et al.* [26] designed two parallel GNNs, one for simulating node state and the other for simulating influence propagation, to deduce the information diffusion process. Yuan *et al.* [27] proposed DyHGCN which models user interactions by global social diffusion graph, and models the change of user preference as graph evolvements. Feng *et al.* [28] considered inter-cascade correlation and proposed the DeepCon&DeepStr model, which constructs high-level graphs between cascades based on their similarity. Wang *et al.* [29] proposed a Multi-scale Context-enhanced Dynamic Attention Network (MCDAN), which takes

full advantage of user friendships and global cascading relationships to capture the global interactive relationships among users.

C. Methods based on Structural and Temporal Learning

Structural and temporal learning-based methods address the popularity prediction problem by comprehensively considering structural and temporal features. Li *et al.* [30] proposed DeepCas, which first uses the Random Walk algorithm to sample multiple node sequences from the cascade graph, and then employs Bidirectional Gated Recurrent Unit (Bi-GRU) to learn temporal features. Chen *et al.* [31] used graph attention network (GAT) and RNN to generate structural and temporal representations respectively, and then integrated them with attention mechanism for popularity prediction. Xu *et al.* [32] used Graph Wavelet to learn the structural features of cascade graphs and global graphs respectively as initial vectors, followed by variational autoencoders to generate corresponding hidden representations for diffusion prediction. Yu *et al.* [33] used the self-attention mechanism in Transformer to extend the traditional Hawkes Process, enabling the continuous sequence and topological structure learning in the cascade graph. Considering the dynamic evolution of the cascades, Wang *et al.* [34] divided the cascade into multiple snapshots and used GCN to learn the representations of each snapshot. Then they learned the weight of nodes based on the dynamic routing algorithm, and finally used LSTM for temporal learning. Chen *et al.* [35] developed a multi-scale graph capsule network (MUG-Caps) with influence attention mechanism to fully learn cascade graphs by considering the influence at multiple scales. Sun *et al.* [36] improved the self-attention mechanism of Transformer by using global spatio-temporal position encoding and relative relation bias matrices to capture different cascade relationships. Tai *et al.* [37] used GNN and DeepWalk [38] to learn within-path and cross-path influence transmission respectively. They then employed Bi-LSTM and GRU with attention mechanism to learn the weights for different structural representations. Ji *et al.* [39] proposed a dynamic graph learning framework which updates the representations based on newly observed user-message interactions, and designed a community detection module to capture evolving community structures for popularity prediction. Zhu *et al.* [8] improved the dynamic GNN with semantic information and proposed specific attention mechanism based on the embedded semantic information to mine the correlations between users and content. Jin *et al.* [40] proposed a multi-layer temporal GNN framework to learn the temporal representations of target entities in each snapshot and predict their future popularity.

For models that emphasize temporal representations, a major limitation is the insufficient attention given to structural features, often leading to suboptimal prediction performance. Conversely, models based on structural representations excel at capturing structural features but typically overlook the temporal dynamics in cascade graphs, and some struggle with the efficiency of large-scale graph processing. To overcome these limitations, recent approaches have integrated both structural and temporal features. However, most of these models still rely exclusively on final cascade size as the sole supervision signal, which limits their generalization capability.

III. Methodology

A. Problem Definition

Given a message m_i and the corresponding cascade graph $\mathcal{G}_{C_i}$ constructed from the observation window $(0, t]$, the information popularity prediction task aims to train a model $f(\cdot;\theta)$ that maps $(m_i, \mathcal{G}_{C_i})$ to the future increase in cascade size during the prediction period $(t, T]$, i.e. $f : (m_i, \mathcal{G}_{C_i}) \longmapsto \Delta P_i$, where $\Delta P_i = |\mathcal{V}_i^T| - |\mathcal{V}_i^t|$ represents the cascade size increase and θ denotes the model parameter to be optimized.

B. Model Overview

The overall architecture of DVCAE is shown in Fig. 1, which consists of the following four modules from bottom to top: *(a) Feature Preprocessing Module*, which takes both the global interaction graph and the local cascade graphs as input, and then generates initial feature representations of users; *(b) Structural Learning Module*, which takes a variational graph autoencoder (VGAE) followed by SAGPool mechanism to learn structural representations of cascades; *(c) Temporal Learning Module*, which takes a variational temporal autoencoder (VTAE) followed by TimeDecay mechanism to learn temporal representations of cascades; *(d) Loss Calculation Module*, which calculates the mean squared loss of popularity and combines it with the reconstruction loss of two variational autoencoders for optimization. The following parts will give detailed illustrations of the modules.

C. Feature Preprocessing

The feature preprocessing module generates initial feature representations of users. It takes both the global interaction graph and the local cascade graph as input, generating global user embeddings by sparsification and matrix factorization, and local embeddings by graph filtering. These embeddings are concatenated to form the initial user features $\boldsymbol{X}$.

Global features. Different cascades are usually with varying sizes, exhibiting strong heterogeneity, and users may participate in multiple cascades simultaneously. Therefore, to learn more structural features, we first aggregate multiple cascades into a global interaction graph, allowing different cascades to share information with each other. In the constructed undirected graph $\mathcal{G}$, each node represents a user and each edge indicates a forwarding behavior. Note that the edge direction is ignored to allow bidirectional information transmission. This is beneficial for structural feature learning, especially when the underlying social network is unavailable. Since the global graph $\mathcal{G}$ may contain millions of nodes and tens of millions of edges, we adopt the NetSMF [41] algorithm to process the graph and generate the global user embeddings $\boldsymbol{X}_G$. The algorithm is based on graph sparsification and matrix decomposition technique, exhibiting higher computational efficiency. Its key idea is to use the following matrix factorization to approximate the DeepWalk [38] algorithm:

$$\log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{M}\right) \tag{1}$$

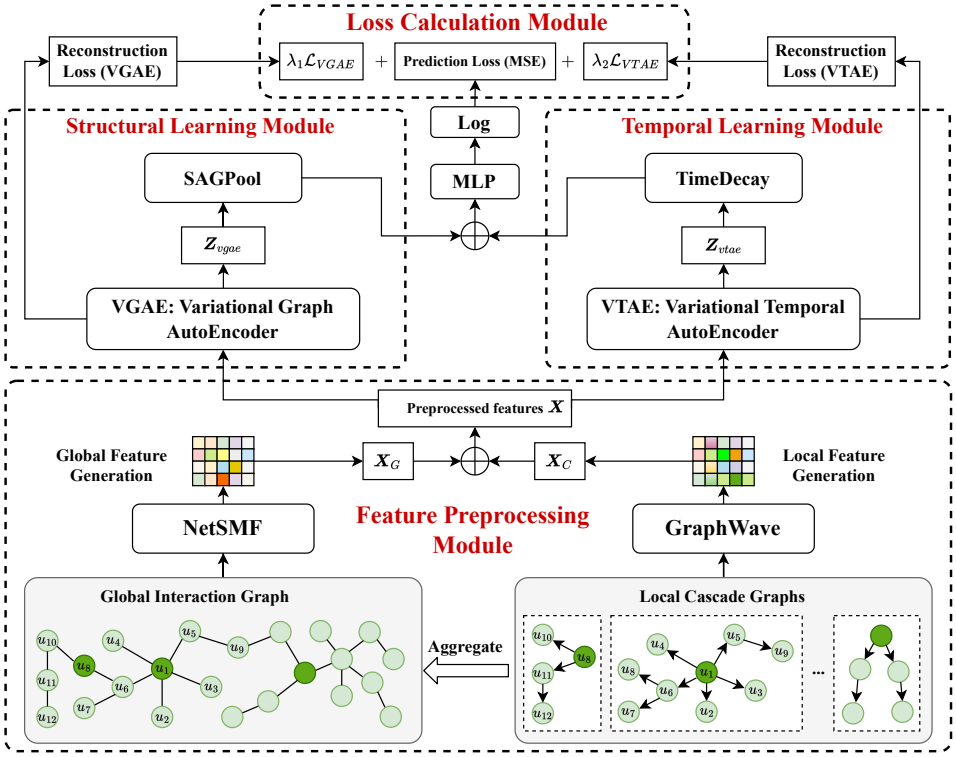


Fig. 1. Overall structure of the DVCAE model.

where $\boldsymbol{M} = 1/C \sum_{r=1}^{C} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r \boldsymbol{D}^{-1}$, $\mathrm{vol}(G) = \sum_i \sum_j A_{ij}$, C is the DeepWalk context window size, b is a hyperparameter used to control regularization terms during matrix decomposition, and $\log^{\circ}(\cdot)$ stands for the element-wise matrix logarithm. The specific process of NetSMF consists of two steps. The first step is to perform the PathSampling [41] algorithm multiple times to generate a sparse matrix similar to the original adjacency matrix. The second step is to perform Randomized SVD [42] on the sparse matrix. Specifically, in the first step, during each iteration, the PathSampling algorithm picks an edge $e \in E$ and an integer r uniformly at random. Then it uniformly draws an integer $k \in [r]$ and performs $(k-1)$-step and $(r-k)$-step random works starting from the two endpoints of edge e respectively, leading to a length-r path. In the second step, the Randomized SVD [42] is performed on the sparse matrix by projecting the original matrix into a low-dimensional space through a Gaussian random matrix.

Local features. Generally, a user may play different roles when participating in different cascades. Therefore, given a cascade graph G_C, it is necessary to generate the local user features within the cascade graph. Considering the high heterogeneity of cascade graphs, we use GraphWave [43], a graph filtering method to generate relatively stable local user embeddings $\boldsymbol{X}_C$ that are less sensitive to local topological noises. Compared to techniques based on random walk, such as DeepWalk [38] and Node2Vec [44], GraphWave is mainly based on vertex-centered spectral graph wavelet diffusion technique, and it uses heat kernel to filter out topologically sensitive signals. Therefore, it is less likely to be affected by small topological perturbations, which will benefit the model generalization ability.

Finally, the above global interaction graph embeddings $\boldsymbol{X}_G$ and local cascade graph embeddings $\boldsymbol{X}_C$ are concatenated as the preprocessed node features $\boldsymbol{X} \in \mathbb{R}^{N \times f}$, which will serve as input of the subsequent variational autoencoders.

D. Structural Learning

Given a specific cascade and its relevant data from the observation window, to comprehensively capture the structural information, we take the adjacency matrix $\boldsymbol{A}$ of the cascade graph and the preprocessed node features $\boldsymbol{X}$ as input, and employ the variational graph autoencoder to generate low-dimensional hidden representations $\boldsymbol{Z}_{vgae}$ of the nodes in the cascade graph. The overall structure of the autoencoder is shown in Fig. 2. Here, we improve the original VGAE framework [45] by replacing the encoder with the graph attention network (GAT), while for the decoder we use the inner product to reconstruct the input adjacency matrix.

The encoder uses a double-layer GAT [46] network as follows:

$$\boldsymbol{Z}_{vgae} = GAT\_Encoder(\boldsymbol{X}, \boldsymbol{A}) \tag{2}$$

where $\boldsymbol{X} \in \mathbb{R}^{N \times f}$ represents the node embedding matrix containing the preprocessed node features, $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ represents the adjacency matrix of the cascade graph, and $\boldsymbol{Z}_{vgae} \in \mathbb{R}^{N \times d}$ represents the low-dimensional hidden embedding matrix of nodes output by $GAT\_Encoder$.

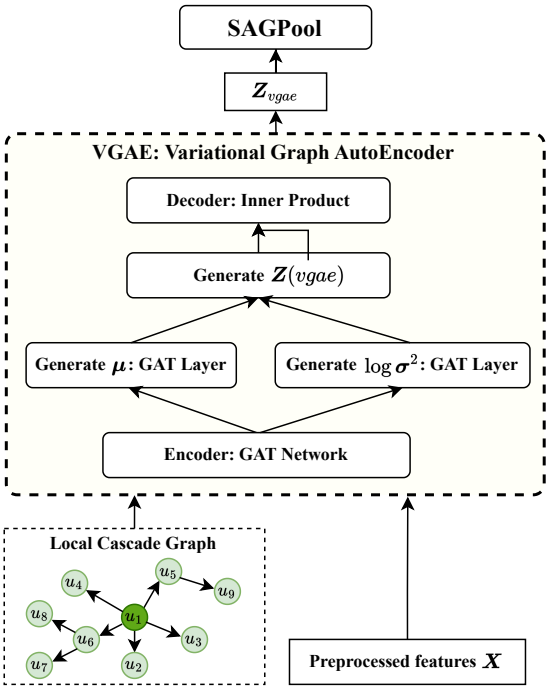


Fig. 2. The generation module of structural representation.

In the encoder, each node applies the following operations:

$$\begin{aligned} \boldsymbol{h}_i &= GAT_1(\boldsymbol{x}_i), \\ \boldsymbol{\mu}_i &= GAT_2(\boldsymbol{h}_i), \\ \log \boldsymbol{\sigma}_i^2 &= GAT_3(\boldsymbol{h}_i), \\ \boldsymbol{z}_i &\sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) \end{aligned} \tag{3}$$

First, GAT_1 in the first layer is used to generate the hidden representation $\boldsymbol{h}_i \in \mathbb{R}^d$ of node i. Then, parallel GAT_2 and GAT_3 convert the hidden representation of the first layer into the mean value $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and the logarithmic variance $\log \boldsymbol{\sigma}_i^2 \in \mathbb{R}^d$, which are used to generate a high-dimensional Gaussian distribution $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$. Finally, we sample on this Gaussian distribution to generate a low-dimensional representation vector of the corresponding node.

For node i, the GAT layer uses the following multi-head attention mechanism to generate the node representations:

$$\boldsymbol{h}_i = \|_{k=1}^{K} ReLU\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(k)} \boldsymbol{W}^{(k)} \boldsymbol{x}_j\right) \tag{4}$$

where $\|$ indicates concatenation operation, K represents the number of attention heads, and $ReLU(\cdot)$ indicates the nonlinear activation function, which makes the learning process more stable.

In the decoder, the inner product of the hidden representations between nodes is used to reconstruct the adjacency matrix $\hat{\boldsymbol{A}}$, which is:

$$\hat{\boldsymbol{A}} = Sigmoid(\boldsymbol{Z}_{vgae} {\boldsymbol{Z}_{vgae}}^T) \tag{5}$$

where $Sigmoid(\cdot)$ represents the nonlinear activation function.

The following cross-entropy loss function is further applied to calculate the reconstruction loss:

$$\mathcal{L}_{stru} = -\frac{1}{N} \sum_{y \in \boldsymbol{A}} y \log \hat{y} + (1-y) \log(1-\hat{y}) \tag{6}$$

where $y \in \{0, 1\}$ represents the value of a specific element in the adjacency matrix $\boldsymbol{A}$, and $\hat{y} \in [0, 1]$ indicates the reconstructed value of the corresponding element. Meanwhile, to maintain the variational characteristics, the KL divergence, which measures the similarity between two distributions, is used in the variational autoencoder for loss calculation. The final loss of VGAE is calculated as follows

$$\mathcal{L}_{VGAE} = \mathcal{L}_{stru} - \mathrm{KL}[q(\boldsymbol{Z}_{vgae} | \boldsymbol{X}, \boldsymbol{A}) || p(\boldsymbol{H})] \tag{7}$$

where $\mathrm{KL}[\cdot]$ is the KL divergence, $q(\boldsymbol{Z}_{vgae} | \boldsymbol{X}, \boldsymbol{A})$ represents the distribution calculated by GAT, and $p(\boldsymbol{H})$ is the prior distribution, for which the standard normal distribution is used.

After the hidden representations of nodes in the cascade graph are generated based on VGAE, the structural representations of the whole cascade graph are further generated via pooling operation. Considering that different users may play different roles in the cascade graph, we employ SAGPool [47] method to perform weighted pooling, generating the hidden representations $\boldsymbol{\zeta}_{vgae} \in \mathbb{R}^{2d}$ for the cascade graph.

Specifically, the SAGPool method consists of three steps. The first step is to construct a self-attention graph pooling layer to obtain the self-attention scores for each node, which is calculated as follows:

$$\boldsymbol{Z}_{vgae}{}^{(l+1)} = Tanh\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{Z}_{\mathrm{vgae}}{}^{(l)} \boldsymbol{\Theta}_{att}\right) \tag{8}$$

where $\boldsymbol{\Theta}_{att} \in \mathbb{R}^{N \times 1}$ is the projection transformation matrix, $\boldsymbol{Z}^{(l+1)} \in \mathbb{R}^{N \times 1}$ indicates the learned self-attention scores, whose values fall within $[-1, 1]$, N is the number of nodes in the global interaction graph, and $Tanh(\cdot)$ represents the nonlinear activation function.

The second step is to select the Top-K nodes according to the above self-attention scores and the pooling rate $k \in [0, 1]$, resulting in a Mask matrix, which is used to obtain a subgraph with the following node ids:

$$\mathbf{idx} = \text{top-rank}(\boldsymbol{Z}_{vgae}, \lceil kN \rceil) \tag{9}$$

Finally, the Readout layer concatenates the output of average pooling and max pooling to generate the final representations for the cascade graph, which is:

$$\boldsymbol{z}_{pool} = \frac{1}{\lceil kN \rceil} \sum_{i=1}^{\lceil kN \rceil} \boldsymbol{z}_{vgae(i)} \Bigg\| \max_{i=1}^{\lceil kN \rceil} (\boldsymbol{z}_{vgae(i)}) \tag{10}$$

After pooling, the structural hidden representations $\boldsymbol{\zeta}_{vgae} = \boldsymbol{z}_{pool}$ is generated for the cascade graph.

E. Temporal Learning

The temporal representation learning module focuses on capturing the temporal information from the cascade. To this end, an improved variational time series autoencoder is used to learn from the cascade sequence and generate the

temporal hidden representations. As shown in Fig. 3, the nodes of a cascade are firstly ordered according to their participating time, resulting in a node sequence. Then, we input the node sequence and the preprocessed node features $\boldsymbol{X}$ into the temporal variational autoencoder. In the autoencoder, the bidirectional GRU (Bi-GRU [48]) network is used as the encoder, and DNN is used to generate the parameters for the Gaussian distribution. The decoder also uses GRU network to reconstruct the node representations. Finally, the loss is calculated by measuring the similarity between the input vector and the output vector of the nodes at each moment in the node sequence.

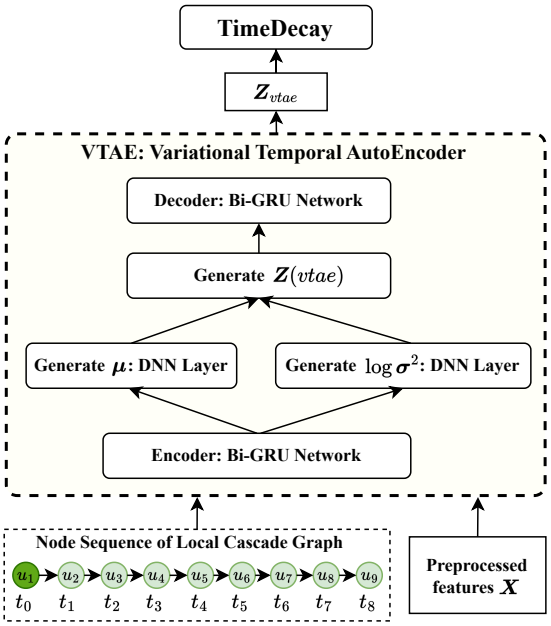


Fig. 3. The generation module of time-series representation.

Similar to VGAE, the VTAE encoder first generates temporal hidden representations using a Bi-GRU network, and then generates a Gaussian distribution using a DNN network on this basis. The corresponding formulas are as follows:

$$\boldsymbol{Z}_{vtae} = RNN\_Encoder(\boldsymbol{X}) \tag{11}$$

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_i \parallel \overleftarrow{\boldsymbol{h}}_i &= Bi\_GRU(\boldsymbol{x}_i), \\ \boldsymbol{\mu}_i &= DNN_1(\overrightarrow{\boldsymbol{h}}_i \parallel \overleftarrow{\boldsymbol{h}}_i), \\ \log \boldsymbol{\sigma}_i^2 &= DNN_2(\overrightarrow{\boldsymbol{h}}_i \parallel \overleftarrow{\boldsymbol{h}}_i) \end{aligned} \tag{12}$$

where $\overrightarrow{\boldsymbol{h}}_i \parallel \overleftarrow{\boldsymbol{h}}_i \in \mathbb{R}^{2d}$ represents the bidirectional embeddings learned for user i, and the arrows above the symbols represent the directions. The mean $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and variance $\log \boldsymbol{\sigma}_i^2 \in \mathbb{R}^d$ are generated based on two relatively independent DNNs. Similarly, the low-dimensional representations $\boldsymbol{z}_i \in \mathbb{R}^d$ corresponding to each user in a cascade sequence is sampled from the Gaussian distribution.

The decoder again uses a Bi-GRU network to reconstruct the representation vector of each user in the cascade sequence, calcuated as follows:

$$\hat{\boldsymbol{X}} = RNN\_Decoder(\boldsymbol{Z}_{vtae}) \tag{13}$$

$$\begin{aligned} \overrightarrow{\boldsymbol{h}'}_i \parallel \overleftarrow{\boldsymbol{h}'}_i &= Bi\_GRU(\boldsymbol{z}_i), \\ \hat{\boldsymbol{x}}_i &= DNN(\overrightarrow{\boldsymbol{h}'}_i \parallel \overleftarrow{\boldsymbol{h}'}_i) \end{aligned} \tag{14}$$

where $\overrightarrow{\boldsymbol{h}'}_i \parallel \overleftarrow{\boldsymbol{h}'}_i \in \mathbb{R}^{2d}$ represents the embedding of user i after decoding, and $\hat{\boldsymbol{x}}_i \in \mathbb{R}^f$ represents the reconstructed node vector.

The loss of VATE is calculated by measuring the difference between the input and output node representations. Similarly, KL divergence is added to the loss calculation to prevent overfitting, which is:

$$\mathcal{L}_{VTAE} = \mathcal{L}_{feat} - \mathrm{KL}[q(\boldsymbol{Z}_{vtae}|\mathbf{X})||p(\boldsymbol{H})] \tag{15}$$

$$\mathcal{L}_{feat} = \sum_{i=1}^{N} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2 \tag{16}$$

After node representations are generated through VTAE, we use the Readout operation to obtain the representations for the entire cascade sequence. Considering the time decay effect in information propagation [15], the GRU network is used to produce the final output, and the time decay coefficient is used as the weight to realize weighted pooling, which is calculated as follows:

$$\boldsymbol{\zeta}_{vtae} = \sum_{j=1}^{R^T} \lambda_{f(T-t_j)} \boldsymbol{z}_j \tag{17}$$

where $\boldsymbol{z}_j$ represents the hidden representations of user j, $\lambda_{f(T-t_j)}$ represents the decay coefficient, calculated as $f(T-t_j) = l, T-t_j \in [t_{l-1}, t_l)$. R^T represents the length of the cascade within the observation window. After pooling, the temporal hidden representations $\boldsymbol{\zeta}_{vtae} \in {}^{2d}$ of the entire cascade sequence is generated.

F. Loss Calculation and Prediction

After concatenating $\boldsymbol{\zeta}_{vgae}$ and $\boldsymbol{\zeta}_{vtae}$, we feed them into the multi-layer perceptron to generate the prediction $\Delta\hat{P}$. Then, mean squared error (MSE) loss is used to calculate the difference between the predicted value and the true value. To avoid the overfitting issue caused by only using the MSE loss for model optimization, we enrich the supervision signals by integrating the reconstruction losses of the two autoencoders. The final loss is calculated as follows:

$$\begin{aligned} \mathcal{L}_{DVCAE} = \frac{1}{M} \sum\nolimits_{m=1}^{M} \big(&(\log_2(\Delta P_m) - \log_2(\Delta\hat{P}_m))^2 \\ &+ \lambda_1 \mathcal{L}_{VGAE}(m) + \lambda_2 \mathcal{L}_{VTAE}(m) \big) \end{aligned} \tag{18}$$

where $\lambda_1, \lambda_2 \in [0, 1]$ are the weighting parameters. The introduction of reconstruction losses could balance the learning of cascade popularity as well as the structural and temporal information, leading to better generalization ability.

G. Complexity Analysis

The time complexity of DVCAE model comes from two parts: preprocessing and dual variational autoencoder.

For preprocessing, the NetSMF algorithm optimizes network embedding through multiple iterations. Each iteration needs to update the representations of nodes by traversing their

neighbors, leading to $\mathcal{O}(|\mathcal{V}_\mathcal{G}|^2)$ time, where $|\mathcal{V}_\mathcal{G}|$ is the graph size after sparsification. Suppose that the algorithm iterates T times, then the overall iteration time complexity is $\mathcal{O}(|\mathcal{V}_\mathcal{G}|^2 \cdot T)$. The time complexity of GraphWave involves eigenvalue decomposition of the Laplace matrix of the graph. The time complexity of eigenvalue decomposition is usually $\mathcal{O}(|\mathcal{V}_C|^3)$, where $|\mathcal{V}_C|$ is the local cascade graph size. Therefore, the preprocessing time complexity is $\mathcal{O}(|\mathcal{V}_\mathcal{G}|^2 \cdot T) + \mathcal{O}(|\mathcal{V}_C|^3)$

For the variational autoencoders, the time complexity of the encoder of VGAE equals that of the three-layer GAT, whose complexity mainly lies in the inner product calculation of attention coefficients, which is $O(|\mathcal{V}_C|^2 \cdot d)$. For the decoder, since it relies on inner product to reconstruct the input, so its complexity is also $O(|\mathcal{V}_C|^2)$. Therefore, the computational complexity of VGAE is $O(|\mathcal{V}_C|^2 \cdot d)$. The time complexity of encoder and decoder of VTAE mainly lies in the calculation in Bi-GRU, whose time complexity is $O(|\mathcal{V}_C| \cdot d^2)$. Therefore, the time complexity of the dual variational autoencoders is $\mathcal{O}(|\mathcal{V}_C|^2 \cdot d + |\mathcal{V}_C| \cdot d^2)$.

In sum, although the preprocessing time complexity is slightly higher, the model training is much more efficient, which is suitable for iterative training in large-scale datasets.

IV. Experimental Evaluation

A. Datasets

We select three large-scale public datasets for experimental evaluation. Each dataset includes the forwarding paths and timestamps, allowing us to build the global interaction graph and local cascade graphs. Detailed statistics of the three datasets are summarized in Table I.

APS[1]**:** It includes scientific papers published by the American Physical Society from July 1, 1893 to December 29, 2017. We take papers from 1893 to 1997 as training samples, with a 20-year period (1997-2017) left for cascade grow. Each paper from 1893 to 1997 is regarded as a node, connected by the citation relationship to construct an undirected global interaction graph. The monitoring time windows are set to 3 and 5 years respectively, and the future prediction time window is set to 20 years.

Weibo-II[2]**:** It contains microblogs and their reposting records published on June 1, 2016, collected from the the Weibo platform (including reposting paths and timestamps). Only microblogs posted between 8:00 AM and 6:00 PM are retained, ensuring that each microblog has at least 6 hours of reposting time. The monitoring windows chosen are 0.5 hours and 1 hour, with a future prediction window of 24 hours.

Twitter-II[3]**:** It consists of public tweets posted on the Twitter platform from March 24, 2012, to April 25, 2012. In this dataset, user sequences with the same label within the monitoring time are treated as independent information cascades. The global graph for this dataset is constructed based on multiple relationships, including follower/followee, retweeter/blogger interactions. The cascade graph is constructed based on these relationships. The monitoring time window is set to 1 day and 2 days respectively, and the future prediction window is set to 32 days.

[1]https://journals.aps.org/datasets.

[2]https://bit.ly/weibodataset.

[3]https://carl.cs.indiana.edu/data/#virality2013.

TABLE I
Statistics and divisions of datasets.

Dataset	APS	Weibo-II	Twitter-II
Total # of cascades	207,685	119,313	88,440
Total # of nodes (users)	616,316	6,738,040	490,474
Total # of forwards	3,304,400	15,249,636	1,903,230
Average popularity	51	240	142
Training set (3y/0.5h/1d)	18,511	21,463	9,639
Validation set (3y/0.5h/1d)	3,967	4,599	2,066
Test set (3y/0.5h/1d)	3,966	4,599	2,065
Training set (5y/1h/2d)	32,102	29,908	12,739
Validation set (5y/1h/2d)	6,879	6,409	2,730
Test set (5y/1h/2d)	6,879	6,408	2,729

B. Experimental Setup

1) Parameter Settings: In our experiments, grid search is used to determine the model hyperparameters according to Table II. Adam optimizer is adopted, with training epochs set to 1000. The training will be stopped in advance when the loss value and the MSLE of the validation set do not decline for 10 consecutive epochs. The learning rate of optimizer is 0.005 and the weight decay parameter (L2 penalty) is set to 0.001. The initial global and local embedding dimension is set to 40. The hidden representation dimension d of VGAE and VTAE is 64. The pooling rate k of SAGPool is 0.5, and the time decay dropout is 0.5. The structural and temporal loss coefficients λ_1 and λ_2 are both 0.5. The dimensions of the last two MLP layers are $2 \times d$ and d, i.e. 128 and 64 respectively.

TABLE II
The tuning of DVCAE model hyperparameters.

Hyperparameter	Grid search values
Learning rate	{1e-3, 5e-3, 1e-4, 5e-4}
Weight decay parameter	{1e-3, 1e-4, 1e-5}
Hidden representation dimension d	{16, 32, 64, 128, 256}
Pooling rate of SAGPool k	{0.3, 0.4, 0.5, 0.6, 0.7}
TimeDecay dropout	{0.3, 0.4, 0.5, 0.6, 0.7}
Structural loss weight λ_1	{0.3, 0.4, 0.5, 0.6, 0.7}
Temporal loss weight λ_2	{0.3, 0.4, 0.5, 0.6, 0.7}

2) Evaluation Metrics: Following previous studies [15], [25], [36], we choose MSLE and MAPE as the evaluation metrics in this paper. MSLE is calculated as follows:

$$MSLE = \frac{1}{M}\sum_{i=1}^{M}(\log_2(\Delta\hat{P}_i) - \log_2(\Delta P_i))^2 \tag{19}$$

MAPE reduces the effect of extreme valued samples by the normalization of error, and it is calculated as follows:

$$MAPE = \frac{1}{M}\sum_{i=1}^{M}\frac{|\log_2(\Delta P_i) - \log_2(\Delta\hat{P}_i)|}{\log_2(\Delta P_i)} \tag{20}$$

C. Baselines

In order to comprehensively evaluate the effectiveness of DVCAE model, we choose 11 baseline methods from the following categories.

1) Feature Engineering Method: We consider the features designed in [49], including the cumulative popularity sequence, the time between the content generator and the first participant, the average time between the first half and the second half participants, the number of leaf nodes, the average node degree, the average and maximum sequence lengths. These features are input into the MLP model to generate predictions. We refer to this baseline as ***Feature-based***.

2) Methods Based on Temporal Representation: The representative models include ***DeepCas*** [30] and ***DeepHawkes*** [15]. For the Random Walk algorithm used in DeepCas and DeepHawkes, it is set to sample paths with K = 200 and length T = 10. The hidden layer of each GRU is set to 32 units. The hidden dimension of the two fully connected layers is set to 32 and 16 respectively. The embedding dimension is set to 64. In addition, we further include a simple time series model, denoted as ***TimeSeries*** [50], for comparison.

3) Methods Based on Graph Representation: ***CasCN*** [25], ***DMT-LIC*** [31], and ***DeepCon&DeepStr*** [28] are selected as the representative models. For CasCN, we keep the original optimal parameters setting (2 convolutional layers) and set the embedding dimension to 50. For DMT-LIC, the embedding dimension is also set to 50, and the number of RNN units is set to 32. The DeepCon&DeepStr parameter settings are the same as the DeepCas model.

4) Methods Based on Autoencoder: ***AECasN*** [14] utilizes autoencoder to capture structural and temporal information from the whole cascade graphs, mapping them into low-dimensional vectors for popularity prediction. ***CasFlow*** [32] uses variational autoencoders to encode pre-learned structural features at user and cascade levels, and proposes normalized flow (NF) module for structural representation normalization. The VAE and NF loss ratio in CasFlow is set to 1.0, while other hyperparameters retain their original settings.

Additionally, two recent models, ***CasTformer*** [36] and $\boldsymbol{I^3T}$ [37], are included for comparison. CasTformer enhances the self-attention mechanism by using global spatio-temporal coding and a relative relation bias matrix to model cascade relationships, along with self-knowledge distillation for better cascade representation. I^3T combines GNN and DeepWalk for learning intra- and inter-path influence, then applies Bi-LSTM for temporal feature learning, and finally uses an improved GRU-based attention mechanism to determine structure weight factors. The original parameter settings for both models are retained.

D. Experimental Results

1) Overall Results: The overall experimental results are shown in Table III, where the best results are in boldface.

Compared with the simple Feature-based and TimeSeries models, DVCAE performs much better in MSLE and MAPE metrics (MSLE improved by 20.3% on average and MAPE improved by 5.2% on average). It indicates that DVCAE has stable and better prediction performance.

Compared with the temporal learning-based DeepCas and DeepHawkes models, DVCAE shows significant improvement on APS and Twitter-II datasets (MSLE improved by 16.2% on average). The DeepCas model mainly relies on a Random Walk algorithm to generate node sequences, which results in the loss of essential structure information. Similarly, DeepHawkes only takes node sequence as input data, ignoring the real propagation structure. On the contrary, both temporal and structural characteristics are elaborately considered in DVCAE, leading to better performance over DeepCas and DeepHawkes.

Compared with the graph learning-based DeepCon&DeepStr, CasCN and DMT-LIC models, DVCAE achieves an average improvement of 14.1% in the MSLE metric across the three datasets. Unlike DeepCon&DeepStr which uses Random Walk and semi-supervised language model for feature learning, GNN-based models can better capture the internal structural features of the cascade. Although CasCN and DMT-LIC models have considered both structural and temporal features, they only used the popularity prediction loss for model optimization, which may result in the overfitting issue. Besides, they did not consider the global interactions between different cascades. It shows that the utilization of global interaction graphs and local cascade graphs in DVCAE makes feature learning more effective, and the features learned from VGAE and VTAE are not only used for prediction but also be constrained by the reconstruction losses to improve the model generalization ability.

Compared with autoencoder-based AECasN and CasFlow models, it is observed that DVCAE and CasFlow perform better than AECasN on the three datasets. This is mainly because AECasN ignores the features in the global interaction graph, indicating the importance of global interaction information. Although DVCAE performs sparsification on the global interaction graph, it still learns valuable features that are beneficial to the model performance. Compared with CasFlow which also adopts global and local features, DVCAE is superior on APS and Twitter-II dataset in both metrics. This is because CasFlow only uses feature reconstruction loss in VAE, while DVCAE considers both structural and feature reconstruction losses. However, we also obseve that the performance of DVCAE on Weibo-II (0.5h) is less significant. This is mainly because there are few users to construct a stable global interaction graph due to the relatively short observe window. It can be seen that with the increase of the observation window, the performance of DVCAE on Weibo-II and Twitter-II is significantly improved, which also indicates the importance of graph structural information learned by DVCAE.

Compared with the Transformer-based CasTfomer model, DVCAE model has better performance across all datasets. This is because CasTformer indistinguishably learns the structural and temporal representations through Transformer sequence modeling, while DVCAE uses hyperparameters to determine the contribution of different reconstruction losses, which can directly adjust the influence of structural learning and temporal learning. Compared with CasTfomer model, DVCAE model has fewer parameters and better efficiency. Compared with I^3T model, DVCAE exhibits more advantages in APS (5y) and Twitter-II (2d), because DVCAE's preprocessing module adopts global interaction graph, which can capture more useful information than the I^3T model.

TABLE III
COMPARISON WITH THE EXPERIMENTAL RESULTS OF BASELINES.

Model	APS				Weibo-II				Twitter-II			
	3y		5y		0.5h		1h		1d		2d	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
Feature-based	1.844	0.270	1.666	0.282	2.715	0.267	2.546	0.272	7.438	0.590	6.357	0.500
TimeSeries	1.867	0.271	1.735	0.291	2.990	0.277	2.693	0.268	7.814	0.547	6.023	0.493
DeepCas	1.548	0.286	1.532	0.285	2.692	0.259	2.582	0.270	7.963	0.598	6.725	0.534
DeepHawkes	1.573	0.271	1.324	0.335	2.891	0.268	2.796	0.282	7.216	0.587	5.788	0.536
CasCN	1.562	0.268	1.421	0.265	2.804	0.254	2.732	0.273	7.183	0.547	5.561	0.532
DeepCon&DeepStr	1.570	0.272	1.562	0.269	2.595	0.261	2.571	0.271	7.044	0.567	5.734	0.569
DMT-LIC	1.539	0.264	1.398	0.258	2.752	**0.249**	2.689	0.270	7.434	0.545	5.427	0.481
AECasN	1.482	0.269	1.384	0.260	2.540	0.273	2.468	0.266	7.021	0.560	5.872	0.559
CasFlow	1.387	0.249	1.398	0.252	**2.402**	0.281	2.409	0.273	6.997	0.568	5.220	0.472
CasTformer	1.533	0.241	1.492	0.254	2.539	0.250	2.471	0.269	6.291	0.531	4.821	0.465
I^3T	1.372	0.247	1.361	0.253	2.459	0.261	2.409	0.267	6.224	0.593	5.001	0.474
DVCAE	**1.219**	**0.238**	**1.220**	**0.245**	2.468	0.253	**2.397**	**0.265**	**6.114**	**0.525**	**4.601**	**0.463**

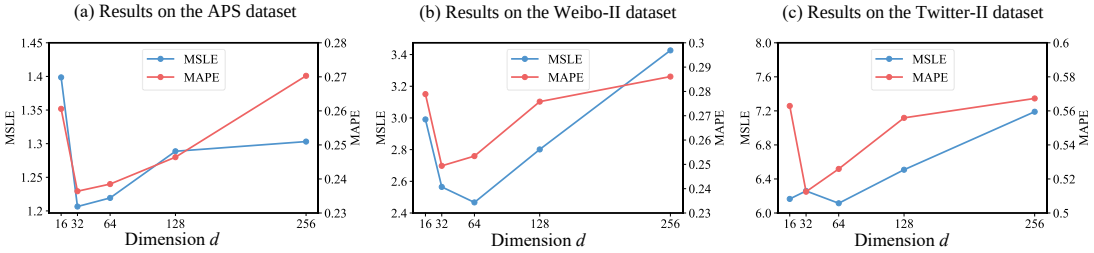


Fig. 4. The impact of representation dimension d on model performance.

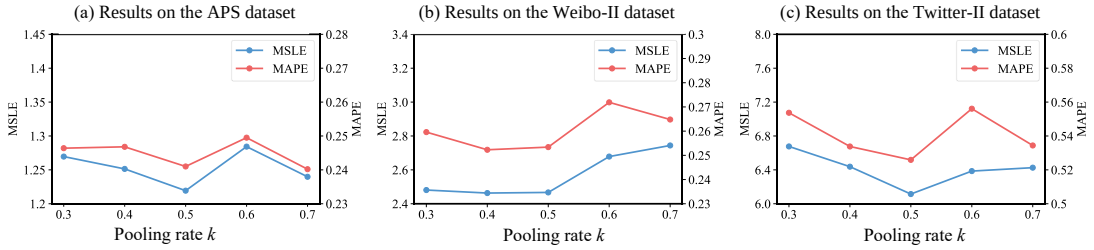


Fig. 5. The impact of pooling rate k on model performance.

2) Parameter Sensitivity Analysis: We investigate how different hyperparameters affect the performance of DVCAE. To this end, we mainly consider four hyperparameters, including hidden representation dimension d, pooling rate k in SAGPool, structural loss weight λ_1 of VGAE and temporal loss weight λ_2 of VTAE. The MSLE and MAPE results of DVCAE model under different parameter settings on three datasets are reported and shown in Fig. 4, Fig. 5, and Fig. 6.

Impact of hidden representation dimension d. Fig. 4 shows the model performance when the value of d varies in $\{16, 32, 64, 128, 256\}$. It can be seen that the MSLE and MAPE metrics show the same trend as the dimension d grows from 16 to 256, and the DVCAE model reaches its best performance when $d \in \{32, 64\}$. Later, the model performance gradually degrades as the parameter d further increases, which may be related to the overfitting phenomenon caused by the excessive large dimension size.

Impact of pooling rate k **in SAGPool**. Pooling rate k in SAGPool determines the proportion of nodes retained from the original cascade graph to produce final cascade representations. A higher k indicates more scattered information gathered from nodes in the cascade graph, which may lead to the over-smoothing issue, while a lower k means focusing on information from a few critical nodes. Fig. 7 shows the model performance when k varies in $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. From the results we see that either a too low or a too high value of k would result in inferior model performance, and it is interesting so see that across all the three datasets, the model reaches its optimal performance when $k = 0.5$.

Impact of λ_1 **and** λ_2. The DVCAE model jointly optimize the prediction loss, the structural reconstruction loss of VGAE and the temporal reconstruction loss of VTAE. Therefore, the hyperparameters λ_1 and λ_2 are used to balance the importance of VGAE and VTAE losses respectively. In Fig. 6, the 3D heatmaps show how the model loss (left), MSLE (middle), and MAPE (right) change along with λ_1 and λ_2 on APS, Weibo-II, and Twitter-II datasets respectively, where the values of λ_1 and λ_2 varies from 0.3 to 0.7. On the APS dataset, it can be seen that the loss increases with the increase of λ_1, while it does not vary significantly under different values of λ_2. It

TABLE IV
ABLATION EXPERIMENT RESULTS.

Dataset	APS		Weibo-II		Twitter-II	
Observation time	3y		0.5h		1d	
Evaluation metrics	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
DVCAE	**1.219**	**0.238**	**2.468**	0.253	**6.114**	0.525
DVCAE-noNetSMF	1.832	0.266	3.002	0.293	11.738	0.809
DVCAE-noGraphWave	1.634	0.259	2.852	**0.228**	7.624	0.594
DVCAE-noVGAE	2.314	0.346	2.916	0.286	12.744	0.985
DVCAE-noVTAE	1.273	0.247	2.657	0.277	6.369	**0.492**

indicates that the structural loss has contributed to the majority of the overall loss. In our experiment, MSLE is used as the criterion for hyperparameter fine-tuning. As can be seen from Fig. 6a, the model reaches its local minimum of MSLE when $\lambda_1 = 0.4$ and $\lambda_2 = 0.5$, which means the model properly balances the three losses and can stably learn the structure and temporal information. On the Weibo-II dataset, similar phenomenon can be observed from Fig. 6b where the loss firstly increases with the increase of λ_1, and then fluctuates with the change of λ_2, which indicates that structural loss also contributes to a larger proportion of the total loss. In terms of the MSLE and MAPE measures, when λ_1 and λ_2 are small ($\lambda_1 = \lambda_2 = 0.3$), the error of DVCAE model is significantly larger ($\text{MSLE} > 2.7, \text{MAPE} > 0.3$). With the increase of λ_1 and λ_2, the error gradually decreases, which indicates that structural loss and temporal loss are important to ensure the model performance. Specifically, the model can maintain good performance when $\lambda_1 = \lambda_2$ and take value from $[0.5, 0.7]$. On the Twitter-II dataset, it can be seen from Fig. 6c that the loss fluctuates greatly under different λ_1 and λ_2 values, which means both structural and temporal reconstruction loss contribute significantly to the total loss. When we look at the MLSE and MAPE measures, we see similar results as observed on the Weibo-II dataset, i.e., the model achieves its local optimum when λ_1 and λ_2 take value from $[0.5, 0.7]$. In general, the results in Fig. 6 clearly validates the importance of structural and temporal reconstruction losses designed in this paper. Properly incorporating the reconstruction losses into the prediction loss could significantly benefit the model performance and improve its generalization ability.

3) Ablation Experiments: In order to demonstrate the importance of each module in DVCAE, a series of ablation studies are further conducted on the DVCAE model. The observation time windows are set to 3 years, 0.5 hour and 1 day for the APS, Weibo-II, and Twitter-II datasets respectively. The specific ablation variants are as follows:

- **DVCAE-noNetSMF:** This variant removes the global user embeddings generated by NetSMF from the input.
- **DVCAE-noGraphWave:** This variant removes the local user embeddings generated by GraphWave from the input.
- **DVCAE-noVGAE:** This variant removes the VGAE and SAGPool parts of the model.
- **DVCAE-noVTAE:** This variant removes the VTAE and TimeDecay parts of the model.

Table IV shows the ablation results of DVCAE and related variants on three datasets.

Effectiveness of the feature preprocessing module: In terms of MSLE metric, DVCAE-noGraphWave and DVCAE-noNetSMF are significantly inferior to DVCAE, and the MSLE of DVCAE-noNetSMF is more than 20% higher on average as compared with DVCAE on the three datasets. It is even doubled on the Twitter-II dataset, which indicates that both global and local preprocessed node features capture valuable structural information from cascade graphs, and the global features learned with NetSMF ensure more stable and better performance.

Effectiveness of the VGAE module: It is observed that when the VGAE module is removed from DVCAE, its performance deteriorates significantly as compared to other variants in terms of the two evaluation metrics. The results indicate that the VGAE and SAGPool mechanisms proposed in this paper can effectively learn the key structural information from cascade graphs to reflect their future diffusion trend.

Effectiveness of the VTAE module: Among the five variants, the results of the DVCAE-noVTAE model are the closest to those of the DVCAE model, and its MAPE metric on the Twitter-II dataset is even better than the DVCAE model, indicating that temporal information plays a relatively weak role in the DVCAE model. It may be due to the fact that GRUs are not very suitable for long-term time series modeling of information cascade propagation, and may ignore the role of the earlier forwarders.

4) Representation Visualization: To visually shown that our DVCAE model can learn meaningful representations, we employ the t-SNE [51] algorithm to project the hidden representations of test sets into a two-dimensional space, as shown in Fig. 7, where the left, middle and right columns correspond to the global representation ($\boldsymbol{\zeta}_{vgae} \| \boldsymbol{\zeta}_{vtae}$), structural representation ($\boldsymbol{\zeta}_{vgae}$), and temporal representation ($\boldsymbol{\zeta}_{vtae}$), respectively. Each point in Fig. 7 indicates a cascade example, and the color reflects the true popularity label (after logarithmic transformation), with darker colors indicating higher popularity.

APS: The APS test set contains 3,966 cascade samples, as shown in Fig. 7a, from which we see that the structural representations exhibit a significant trend in the distribution of cascade popularity, where the color gradually becomes darker from right to left. This indicates that the structural information learned from the cascade graph is strongly related to the future growth of the cascade. On the contrary, there is no obvious trend in popularity distribution under the temporal

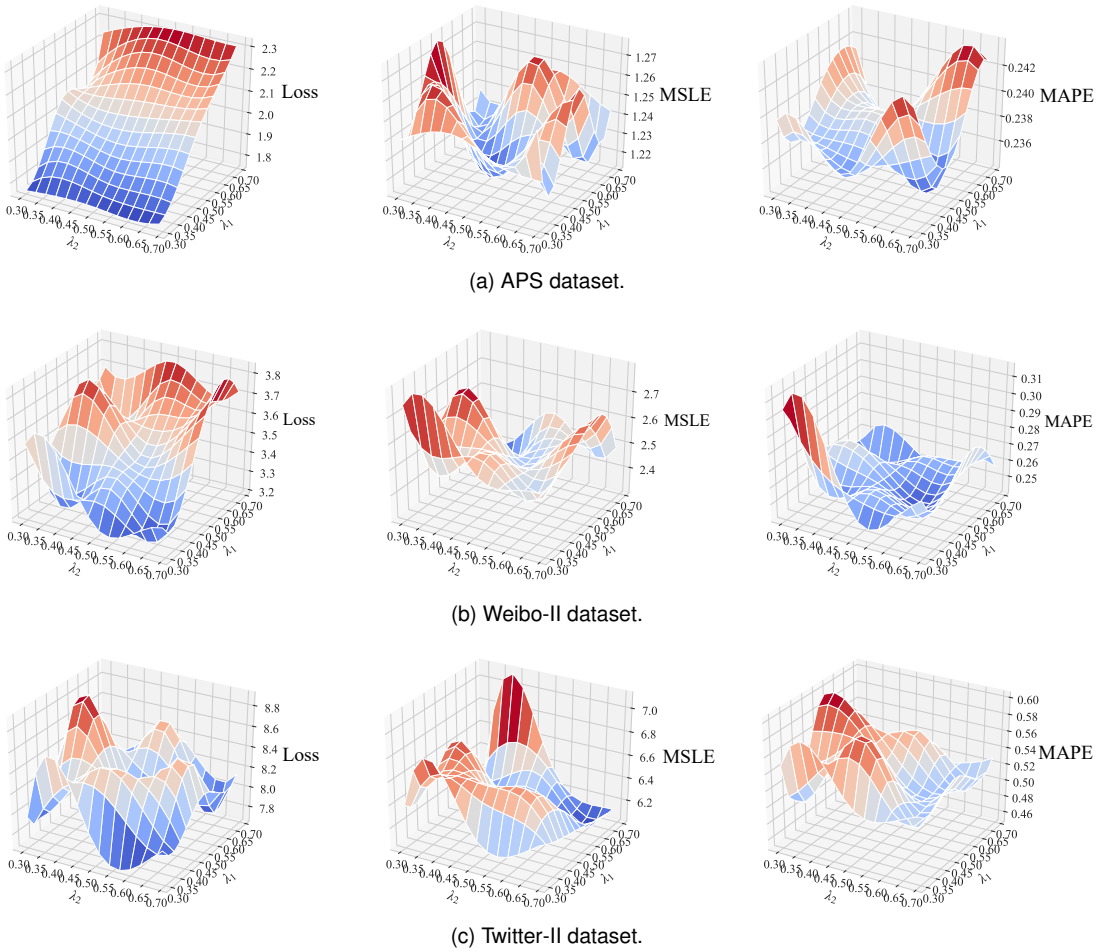


Fig. 6. The impact of λ_1 and λ_2 on model performance.

representations, which means that temporal information has higher uncertainty in information diffusion. The global representations still exhibit a clear trend in popularity distribution (gradually becoming darker from left to right), suggesting that the global representations can properly capture the real popularity distribution.

Weibo-II: The Weibo-II test set contains 4,599 cascade samples. By comparing Fig. 7b and Fig. 7a, it can be observed that the structural representations learned on the Weibo-II dataset are even more distinguishable than the APS dataset, and the popular and unpopular cascade samples are clearly separated. Similarly, the temporal representations learned by the VTAE module are less distinguishable than the structural representations learned by the VGAE module.

Twitter-II: The Twitter-II test set contains 2,065 cascade samples. As shown in Fig. 7c, the structural representations can better separate popular cascades from unpopular ones, exhibiting similar behavior to the Weibo-II and APS datasets. The major difference is that the Twitter-II test set contains more extreme examples (as shown by the dark points located in the light area), and has popularity with a wider range.

In short, the visualization results consistently show that the representations learned by the VGAE module are more distinguishable in terms of popularity. The temporal information in the cascade can be fine-tuned on this basis, and finally leading to more accurate predictions.

5) Case Study: To validate the practicability and reliability of our DVCAE model, we use a case study to show its superb ability in predicting information popularity for cascades with completely different structural characteristics. Specifically, Fig. 8 shows the graph structure of two cascades within the 1-hour observation window, where the left cascade has 113 participants while the right one contains 72 participants during the observation window. It is straightforward to see that the two cascades exhibit significantly different structures, where cascade A is a typical star graph while cascade B shows more complex connections. We use DVCAE to predict the 24-hour (1-day) cascade size increase. Specifically, for cascade A, its actual cascade size increase is 17, and the prediction is 14. For cascade B, its actual cascade size increase is 386, and the prediction is 397. Obviously, the two cascades not only have different structures, but also exhibit completely different growing patterns, where cascade A grew by only 15%, while cascade B grew by more than 500%. Nevertheless, DVCAE can still accurately predict the cascade growth, with a relative error of 17.6% and 2.8% respectively. The results indicate that the DVCAE model can accurately and reliably capture the propagation and growing patterns for different cascade structures. Moreover, our statistical results show that nearly 3% of the cascade samples contain less than 15 nodes in

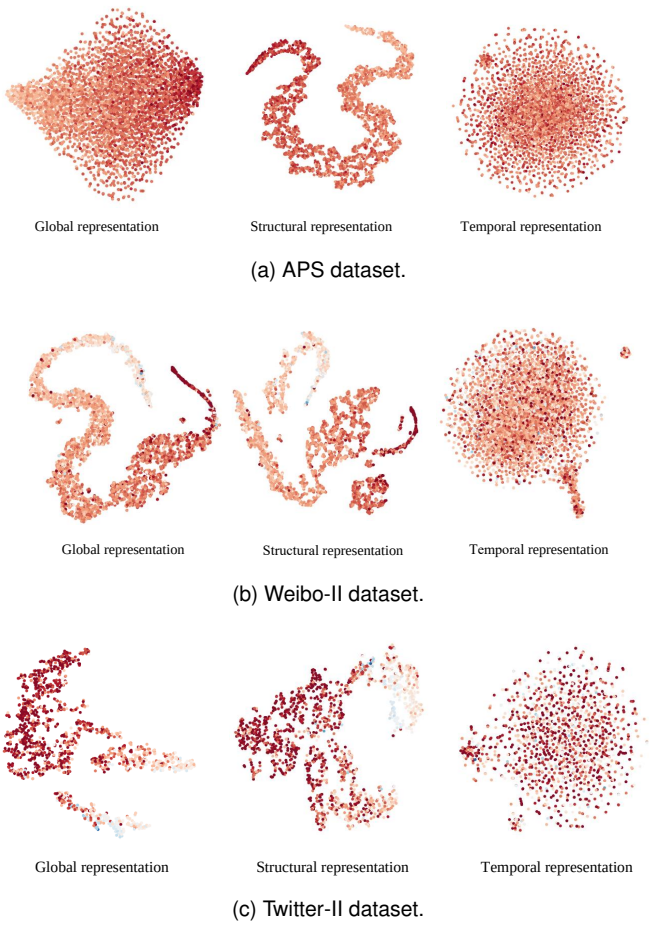


Fig. 7. Dimensionality reduction visualization of hidden variable representation of DVCAE model on three datasets.

the 1-hour observation window, while their final cascade sizes reach more than 100. For these cascades, the DVCAE model exhibits an average prediction error of less than 2.5, validating its superb ability for popularity prediction.

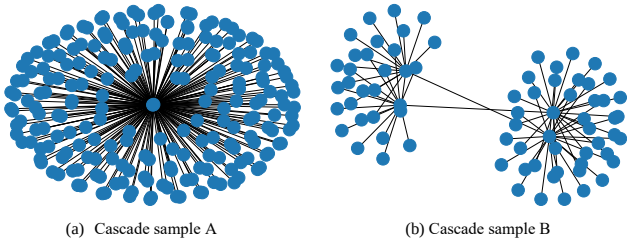


Fig. 8. Case study for two cascades with completely different structures.

V. Conclusion

In this paper, we considered the key challenges of unavailable underlying social relations, cascade heterogeneity and limited supervision signals in the popularity prediction problem, and proposed a semi-supervised Dual Variational Cascade AutoEncoders model for accurate cascade popularity prediction. To this end, we constructed a global interaction graph by aggregating multiple cascades to learn structural information more comprehensively. To fully capture both structural and temporal information from the cascades, we employed two parallel variational autoencoders and incorporated their reconstruction losses into the popularity prediction loss to enrich the supervision signals for optimization. We conducted extensive experiments on three real-world datasets and the results clearly show the superiority of our model over the state-of-the-art methods.

There are several possible directions for the improvement of our work. Firstly, the structural and temporal loss weights are manually set in the DCVAE model. In the future we will consider adaptive weights to reduce the retraining costs. Secondly, our model is mainly based on the structural and temporal information of cascades. In the future we will consider incorporating more information such as the message content and topic information to further increase the prediction performance. Finally, our model is only used for macroscopic popularity prediction, and it can be extended to support microscopic diffusion prediction in the future.

References

[1] D. Jin, Z. Yu, P. Jiao, S. Pan, D. He, J. Wu, P. S. Yu, and W. Zhang, "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1149–1170, 2021.

[2] P. Wan, X. Wang, G. Min, L. Wang, Y. Lin, W. Yu, and X. Wu, "Optimal control for positive and negative information diffusion based on game theory in online social networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 1, pp. 426–440, 2022.

[3] K. Aslett, Z. Sanderson, W. Godel, N. Persily, J. Nagler, and J. A. Tucker, "Online searches to evaluate misinformation can increase its perceived veracity," *Nature*, vol. 625, no. 7995, pp. 548–556, 2024.

[4] X. Yang, J. Shang, Q. Hu, and D. Liu, "Aris: Efficient admitted influence maximizing in large-scale networks based on valid path reverse influence sampling," *IEEE Transactions on Emerging Topics in Computing*, vol. 12, no. 3, pp. 700–714, 2024.

[5] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, "Causal intervention for leveraging popularity bias in recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, p. 11–20.

[6] A. Silva, L. Luo, S. Karunasekera, and C. Leckie, "Unsupervised domain-agnostic fake news detection using multi-modal weak signals," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[7] Z. Cheng, J. Zhang, X. Xu, G. Trajcevski, T. Zhong, and F. Zhou, "Retrieval-augmented hypergraph for multimodal social media popularity prediction," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 445–455.

[8] J. Zhu, R. Li, X. Chen, S. Mao, J. Wu, and Z. Zhao, "Semantics-enhanced temporal graph networks for content popularity prediction," *IEEE Transactions on Mobile Computing*, 2024.

[9] P. Bao, R. Yan, and C. Yang, "Popularity prediction via modeling temporal dependencies on dynamic evolution process," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[10] H. Li, C. Xia, T. Wang, S. Wen, C. Chen, and Y. Xiang, "Capturing dynamics of information diffusion in sns: A survey of methodology and techniques," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–51, 2021.

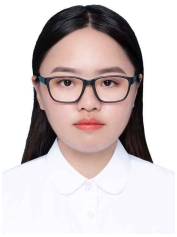
[11] X. Gao, Z. Cao, S. Li, B. Yao, G. Chen, and S. Tang, "Taxonomy and evaluation for microblog popularity prediction," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 2, pp. 1–40, 2019.

[12] 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.

[13] X. Zhang, J. Shang, X. Jia, D. Liu, F. Hao, and Z. Zhang, "Collaborate-cas: popularity prediction of information cascades based on collaborative graph attention networks," in *International Conference on Database Systems for Advanced Applications*. Springer, 2022, pp. 714–721.

[14] X. Feng, Q. Zhao, and Y. Li, "Aecasn: An information cascade predictor by learning the structural representation of the whole cascade network with autoencoder," *Expert Systems with Applications*, vol. 191, p. 116260, 2022.
[15] 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.
[16] J. Wang, V. W. Zheng, Z. Liu, and K. C.-C. Chang, "Topological recurrent neural network for diffusion prediction," in *2017 IEEE international conference on data mining (ICDM)*. IEEE, 2017, pp. 475–484.
[17] C. Yang, H. Wang, J. Tang, C. Shi, M. Sun, G. Cui, and Z. Liu, "Full-scale information diffusion prediction with reinforced recurrent networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 5, pp. 2271–2283, 2021.
[18] A. Vaswani, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.
[19] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, "Cascade dynamics modeling with attention-based recurrent neural network." in *IJCAI*, vol. 17, 2017, pp. 2985–2991.
[20] 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.
[21] C. Yang, M. Sun, H. Liu, S. Han, Z. Liu, and H. Luan, "Neural diffusion model for microscopic cascade study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 3, pp. 1128–1139, 2019.
[22] H. Zhu, S. Yuan, X. Liu, K. Chen, C. Jia, and Y. Qian, "Casciff: A cross-domain information fusion framework tailored for cascade prediction in social networks," *Knowledge-Based Systems*, p. 112391, 2024.
[23] 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.
[24] 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*, vol. 34, no. 12, pp. 6033–6046, 2021.
[25] 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.
[26] 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.
[27] C. Yuan, J. Li, W. Zhou, Y. Lu, X. Zhang, and S. Hu, "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*. Springer, 2021, pp. 347–363.
[28] X. Feng, Q. Zhao, and Z. Liu, "Prediction of information cascades via content and structure proximity preserved graph level embedding," *Information Sciences*, vol. 560, pp. 424–440, 2021.
[29] X. Wang, L. Wang, Y. Su, Y. Zhang, and A.-A. Liu, "Mcdan: a multi-scale context-enhanced dynamic attention network for diffusion prediction," *IEEE Transactions on Multimedia*, 2024.
[30] 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.
[31] X. Chen, K. Zhang, F. Zhou, G. Trajcevski, T. Zhong, and F. Zhang, "Information cascades modeling via deep multi-task learning," in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 885–888.
[32] X. Xu, F. Zhou, K. Zhang, S. Liu, and G. Trajcevski, "Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3484–3499, 2021.
[33] L. Yu, X. Xu, G. Trajcevski, and F. Zhou, "Transformer-enhanced hawkes process with decoupling training for information cascade prediction," *Knowledge-Based Systems*, vol. 255, p. 109740, 2022.
[34] Y. Wang, X. Wang, Y. Ran, R. Michalski, and T. Jia, "Casseqgcn: Combining network structure and temporal sequence to predict information cascades," *Expert Systems with Applications*, vol. 206, p. 117693, 2022.
[35] X. Chen, F. Zhang, F. Zhou, and M. Bonsangue, "Multi-scale graph capsule with influence attention for information cascades prediction," *International Journal of Intelligent Systems*, vol. 37, no. 3, pp. 2584–2611, 2022.
[36] X. Sun, J. Zhou, L. Liu, and Z. Wu, "Castformer: A novel cascade transformer towards predicting information diffusion," *Information Sciences*, vol. 648, p. 119531, 2023.
[37] Y. Tai, H. He, W. Zhang, H. Yang, X. Wu, and Y. Wang, "Predicting information diffusion using the inter-and intra-path of influence transitivity," *Information Sciences*, vol. 651, p. 119705, 2023.
[38] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
[39] S. Ji, X. Lu, M. Liu, L. Sun, C. Liu, B. Du, and H. Xiong, "Community-based dynamic graph learning for popularity prediction," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 930–940.
[40] R. Jin, X. Liu, and T. Murata, "Predicting popularity trend in social media networks with multi-layer temporal graph neural networks," *Complex & Intelligent Systems*, pp. 1–17, 2024.
[41] J. Qiu, Y. Dong, H. Ma, J. Li, C. Wang, K. Wang, and J. Tang, "Netsmf: Large-scale network embedding as sparse matrix factorization," in *The World Wide Web Conference*, 2019, pp. 1509–1520.
[42] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.
[43] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1320–1329.
[44] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
[45] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.
[46] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.
[47] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *International conference on machine learning*. pmlr, 2019, pp. 3734–3743.
[48] K. Hu, Y. Cheng, J. Wu, H. Zhu, and X. Shao, "Deep bidirectional recurrent neural networks ensemble for remaining useful life prediction of aircraft engine," *IEEE Transactions on Cybernetics*, vol. 53, no. 4, pp. 2531–2543, 2021.
[49] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, "Can cascades be predicted?" in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 925–936.
[50] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of youtube videos," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 365–374.
[51] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *The journal of machine learning research*, vol. 15, no. 1, pp. 3221–3245, 2014.

Jiaxing Shang received the B.S. and Ph.D. degrees in Control Science and Engineering from Tsinghua University, Beijing, China, in 2010 and 2016 respectively. He is a professor at the College of Computer Science in Chongqing University, Chongqing, China. Currently, he is doing the Marie Sklodowska-Curie Fellow with the University of Exeter, Exeter, UK. His research interests include social networks analysis, explainable AI, industrial big data analytics, etc. He has published more than 80 high quality journal and conference articles, including TKDE, TITS, TNNLS, PR, TCAD, HPCA, etc.



Xueqi Jia was born in Xingtai, Hebei, China in 1999. She received the B.S. degree in Computer Science and Technology from Hebei University of Technology, Tianjin, China, in 2016 and the M.S. degree in Computer Science and Technology from Chongqing University, Chongqing, China, in 2023. Her research interests include data mining, graph neural network, and social network analysis.

Xiaoquan Li was born in Yantai, Shandong, China in 2000. She received the B.S. degree in in Computer Science and Technology from Chongqing University in 2022. She is currently doing the M.S. degree in Computer Science and Technology at Chongqing University. Her research interests include data mining, machine learning and industrial big data analytics.

Fei Hao received the Ph.D. degree in Computer Science and Engineering from Soonchunhyang University, South Korea, in 2016. From 2020 to 2022, he was a Marie Curie Fellow with the University of Exeter, Exeter, United Kingdom. He is currently a Professor with the School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an, China. He has published more than 150 papers in the leading international journals and conference proceedings, such as IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Services Computing, IEEE Transactions on Network Science and Engineering, IEEE Communications Magazine, IEEE Internet Computing, ACM Transactions on Multimedia Computing, Communications and Applications as well as ACM SIGIR, IEEE GLOBECOM. His research interests include social computing, soft computing, big data analytics, pervasive computing, and data mining. He is also a member of ACM, CCF, and KIPS. In addition, he was the recipient of the Best Paper Award from IEEE GreenCom 2013. He was also the recipient of the Outstanding Service Award at IEEE DSS 2018, and IEEE SmartData 2017, the IEEE Outstanding Leadership Award at IEEE CPSCom 2013, and the 2015 Chinese Government Award for Outstanding Self-Financed Students Abroad.

Ruiyuan Li is an associate professor with Chongqing University, China. He is the director of Start Lab (Spatio-Temporal Art Lab). He received the B.E. and M.S. degrees from Wuhan University, China in 2013 and 2016, respectively, and the Ph.D. degree from Xidian University, China in 2020. He was the Head of Spatio-Temporal Data Group in JD Intelligent Cities Research, leading the research and development of JUST (JD Urban Spatio-Temporal data engine). Before joining JD, he had interned in Microsoft Research Asia from 2014 to 2017. His research focuses on Spatio-temporal Data Management and Urban Computing.

Geyong Min is a Professor of High Performance Computing and Networking in the Department of Computer Science at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Computer Networks, Wireless Communications, Parallel and Distributed Computing, Ubiquitous Computing, Multimedia Systems, Modeling and Performance Engineering.