










ELSEVIER

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Information diffusion prediction via meta-knowledge learners

Zhangtao Cheng^{a, }, Jienan Zhang^{a, }, Xovee Xu^{a, }, Wenxin Tai^{a, },
Fan Zhou^{a, b, c, },*, Goce Trajcevski^{d, }, Ting Zhong^{a, }

^a University of Electronic Science and Technology of China, Chengdu, 610054, Sichuan, China

^b Key Laboratory of Intelligent Digital Media Technology of Sichuan Province, Chengdu, 610054, Sichuan, China

^c Kash Institute of Electronics and Information Industry, Kashgar, 844000, Xinjiang, China

^d Iowa State University, Ames, 50011, IA, USA

ARTICLE INFO

Keywords:

Information diffusion prediction
Meta-knowledge learning
Social network analysis
Graph learning
Spatio-temporal data mining

ABSTRACT

Information diffusion prediction is a fundamental task for a vast range of applications, including viral marketing identification and precise recommendation. Existing works focus on modeling limited contextual information from independent cascades while overlooking the diverse user behaviors during the information diffusion: First, users typically have diverse social relationships and pay more attention to their social neighbors, which significantly influences the process of information diffusion. Second, complex temporal influence among different cascade sequences leads to unique and dynamic diffusion patterns between users. To tackle these challenges, we propose MetaCas, a novel cascade meta-knowledge learning framework for enhancing information diffusion prediction in an adaptive and dynamic parameter generative manner. Specifically, we design two meta-knowledge-aware topological-temporal modules – Meta-GAT and Meta-LSTM – to extract cascade-specific topological and temporal user interdependencies inherent within the information diffusion process. Model parameters of topological-temporal modules are adaptively generated by the constructed meta-knowledge from three important perspectives: user social structure, user preference, and temporal diffusion influence. Extensive experiments conducted on four real-world social datasets demonstrate that MetaCas outperforms state-of-the-art information diffusion models across several settings (up to 16.6% in terms of Hits@100).

1. Introduction

Online social media platforms, such as Twitter, Weibo and Reddit, have enriched real-time communications among individuals and enabled timely access and sharing of information of interest. Upon receiving information contents (e.g., microblogs, news, and videos), users often further disseminate them through the underlying social network, resulting in an information cascade of user activations [1]. Predicting information diffusion has become a fundamental undertaking in various applications on social platforms. For example, recommender systems [2] could benefit from predicting user engagement to formulate effective marketing strategies. Furthermore, this task offers diverse benefits, from helping users navigate information overload to improving real-world applications, including personalized recommendation [3], and micro-video popularity prediction [4].

* Corresponding author.

E-mail addresses: zhangtao.cheng@outlook.com (Z. Cheng), eroiczjn@outlook.com (J. Zhang), xovee@live.com (X. Xu), wxtai@std.uestc.edu.cn (W. Tai), fan.zhou@uestc.edu.cn (F. Zhou), gocet25@iastate.edu (G. Trajcevski), zhongting@uestc.edu.cn (T. Zhong).

<https://doi.org/10.1016/j.ins.2025.122034>

Received 21 January 2023; Received in revised form 25 February 2025; Accepted 25 February 2025

Available online 27 February 2025

0020-0255/© 2025 Elsevier Inc. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

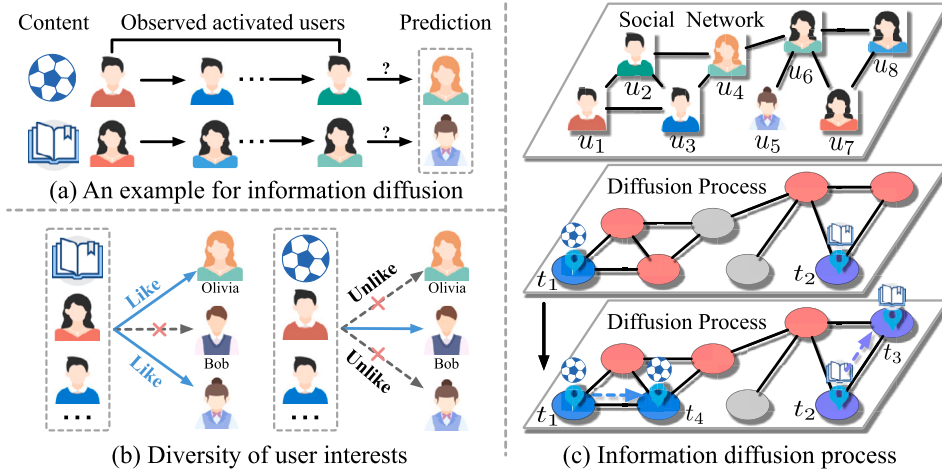


Fig. 1. (a) An example to illustrate the task of information diffusion prediction. (b) An illustration of diverse user interests pertaining to various user-generated content. (c) Visualization of information diffusion processes on the social network depicted in subplot (a).

As shown in Fig. 1(a), information diffusion prediction aims to identify the potential users who are likely to share the information by modeling the previous user sharing process, which can be broadly grouped into three categories: (1) Independent cascade (IC)-based methods rely on the assumption of independent diffusion, and directly calculate diffusion probabilities using pre-designed models, such as the IC model and Linear Threshold (LT) model; (2) Feature engineering-based methods focus on leveraging various cascade-related features to model the information propagation, e.g., social graph [5], and time attribute [6]; (3) Deep learning-based methods primarily apply advanced learning architectures to improve the diffusion propagation modeling. Numerous modern neural architectures, such as recurrent networks [7], and graph learning networks [8], have been employed or adapted to learn social correlations and non-sequential long-term diffusion dependencies for information diffusion prediction.

Challenges. Despite the remarkable performance of existing diffusion models, they typically rely on the assumption that information cascades have independent and invariant diffusion behaviors, and generally adopt a static learning paradigm, i.e., a diffusion model shares the same model parameters among all information cascades to learn social dependencies and temporal influences. However, this assumption presents two significant drawbacks: (1) **Failing to consider diverse user social dependency.** The social homophily theory [9] suggests that social relationships are more likely to happen between users who share common interests, resulting in different social dependencies across distinct users. First, users are more likely to be exposed to online content that matches their interests. For example, in Fig. 1(b), Olivia demonstrates a preference for reading and sharing science-fiction material on online social platforms. Conversely, Bob tends to publish and consume content related to football. When a source user publishes World Cup-related content, Bob is more inclined to share the specific World Cup-related content compared to Olivia. Second, users tend to pay more attention to content posted by their social neighbors. For example in Fig. 1(c), when user u_1 posts a new tweet, the message swiftly spreads to u_1 's neighbors (u_2 and u_3) and may further impact their social behaviors (e.g., comment, like, and re-share). Complementary, u_6 and u_8 are more susceptible to u_7 than u_1 . Considering a large social network with tens of thousands of users, the social dependencies in the network are highly diverse, especially when we take user interest into account. (2) **Overlooking complex temporal influence.** Depending on the dynamic information cascade states and time attributes, the temporal correlations between users in a cascade change over time. Users can browse new information content from their social neighbors at different times depending on many factors (e.g., lifestyle and mood), resulting in distinct cascade evolution of activated users. Furthermore, information content posted by the same user may develop to distinct propagation dynamics given different types of user interests and behaviors. Consequently, simply using the same model parameters across all cascades can hardly capture the specific characteristic of each cascade from different distributions. Meanwhile, the learned diffusion patterns can be highly biased against individual users, especially for those inactive ones. This is because existing works neglect the cascade heterogeneity and the diversity of social dependencies, as well as temporal influence.

To tackle the aforementioned challenges, we propose **MetaCas**, a **Meta**-knowledge-aware framework that generates **Cascade**-specific model parameters to extract the diverse topological-temporal dependencies for enhancing information diffusion prediction. Specifically, we regard each individual information cascade prediction as a single task and transform it into a new-task adaptation problem. As shown in Fig. 2, we first design two meta-knowledge learners, i.e., Structure Meta Knowledge Unit (SMK-Unit) and Dynamic Meta Knowledge Unit (DMK-Unit), in term of users and information cascades. These learners separately extract meta-knowledge reflecting user social dependency and cascade temporal influence from user attributes (i.e., social network and user-item network) and cascade attributes (i.e., time and users involved in the cascades). Second, different from existing diffusion models [1], we propose a function-level sharing scheme based on dynamic adaptive parameter generation. We generate customized and adaptive model weights for topological-temporal modules (Meta-GAT and Meta-LSTM). Moreover, Meta-GAT and Meta-LSTM are designed to capture the diverse topological-temporal correlations of users activated in the information cascade. With such designs, MetaCas equipped with

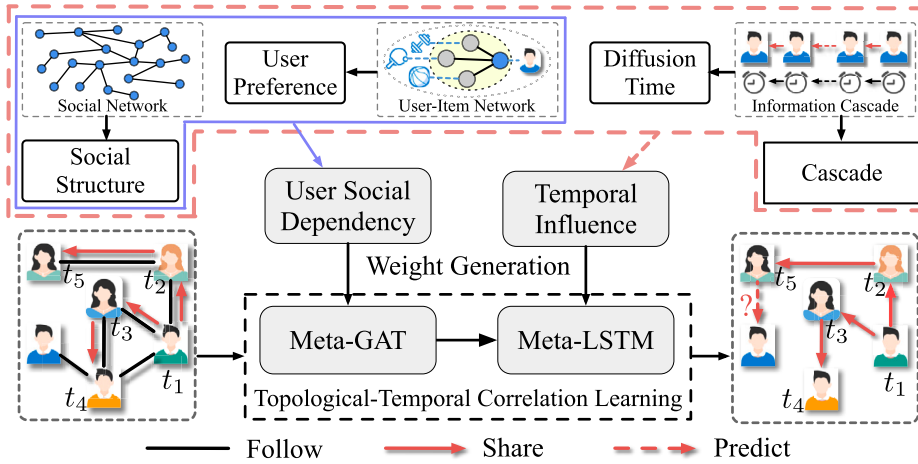


Fig. 2. The insight of MetaCas. First, the structural meta-knowledge learner extracts user-related meta-knowledge reflecting the user social dependency from user attributes, including the social network and user-item network. Second, the dynamic meta-knowledge learner captures cascade-related meta-knowledge reflecting cascade temporal influence by considering cascade attributes like timestamps and users involved in the cascades. Finally, customized and adaptive model weights are generated for each cascade, enabling two topological-temporal modules (Meta-GAT and Meta-LSTM) to effectively capture the diverse topological-temporal correlations during the information diffusion processes.

two meta-knowledge learners not only extracts rich information of user preferences from historical interactions, but also incorporates valuable cascade semantic knowledge, enhancing information diffusion prediction. Our contributions are fourfold:

- We propose a novel meta-knowledge-aware framework named MetaCas for information diffusion prediction. To learn dynamic user social structures and complex temporal influences in information diffusion, MetaCas is composed of two meta-knowledge learning networks, Meta-GAT and Meta-LSTM. It leverages three cascade-related attributes, including social structure, user preference, and temporal influence to generate dynamic model weights that improve the cascade representation learning.
- Meta graph attention network (Meta-GAT) is designed to learn user interdependencies from the underlying social network along with user preferences learned from the user-item interaction network. In addition, the model weights generated from the structure meta-knowledge learner enable the Meta-GAT to more accurately capture diverse topological correlations of users and cascade-related attributes.
- Meta long short-term memory (Meta-LSTM) is designed to learn time-varying diversities of activated users in information cascade. The model parameters are adaptively generated from the dynamic meta knowledge learner that allows the model better learn the temporal user influence on cascade evolution.
- Extensive experiments conducted on four real-world social datasets show that MetaCas significantly improves the information diffusion prediction performance, up to 16.6% in Hits@100, compared to state-of-the-art diffusion baselines. The source code of MetaCas and four datasets are released at <https://github.com/CZ-TAO12/MetaCas>.

The remainder of this work is organized as follows. We review the literature in Section 2 and introduce preliminaries including problem definition in Section 3. The overall framework of the proposed MetaCas as well as model details are described in Section 4. In Section 5, we present performance evaluations, ablation studies, parameter analyses, and case studies between our model and baselines. At last, we conclude our work and point out future directions in Section 6.

2. Related work

2.1. Information diffusion prediction

Existing works primarily focus on inferring the future propagation state of an information item based on the temporally-ordered sequence of affected users and other relevant knowledge (e.g., the underlying social network) [1]. Previous works addressing this problem mainly follow three directions: independent cascade (IC)-based methods, feature engineering-based methods, and deep learning-based methods. (1) IC models [10] make the diffusion prediction with an *independence* assumption: every user-pair has an independent diffusion probability [11]. However, this oversimplifies the complex nature of the information diffusion process, ignoring user heterogeneity and user interests and results in a poor performance for the prediction model on real-world datasets. (2) Feature engineering-based methods aim to identify informative users and cascade features and incorporate extracted features into models such as decision trees and support vector machines for diffusion prediction. Researchers have found that content, user profile, social ties, and temporal features are predictive features for information diffusion prediction [12]. Nonetheless, this kind of models rely on hand-crafted features and a high feature quality requires great effort and expert domain knowledge. Consequently, they are less generalized to new data domains. (3) Deep learning-based models [13] adopt an end-to-end framework that automatically learns user

dynamics and relevant knowledge. Among them, temporal models [13] project the diffusion process on social relationships and use specific mechanisms such as gating and attention [14] for performance improvements. Graph-based models learn diffusion structures by constructing various local and global graphs. For example, dynamic graph [15], hypergraph [16] and sequential hypergraph [17] are utilized to enhance user feature learning.

However, these models typically make homogeneity assumptions and overlook diverse spatial-temporal correlations within the information diffusion process, including diverse user preference and complex temporal influence. In contrast, MetaCas is design to introduce cascade meta-knowledge learned from cascade-related attributes (i.e., social structures, user preference, and temporal influence) via introducing two independent meta-knowledge learners, while modeling diverse structural-temporal correlations within cascades for enhancing information diffusion prediction.

2.2. Graph representation learning

Graph representation learning (GRL) is an emerging topic in information cascade analysis and prediction, which models the social roles and relationships between users in the information diffusion process. GNNs [18] are a family of GRL methods that have been successfully applied to analyze graph-structured data. GNNs typically follow a recursive neighborhood aggregation mechanism to capture the structural information within nodes' neighborhoods. Graph convolutional networks (GCN) [19] is a classical GNN-based framework, which simplifies the graph convolution operation in ChebNet and designs a generic and efficient convolution operation for graph data. Since GCN considers all the neighbors of a node equally important during aggregation, it is hard to accurately model a large noisy graph. Graph attention network (GAT) [20] incorporates attention mechanism into the neighborhood information propagation step to improve the performance of GCN, which specifies each neighbor of a node with a unique attention coefficient.

As for information diffusion prediction task, FOREST [6] constructs a structural context extraction strategy based on the neighborhood aggregation method of GCN to further characterize the influence of the underlying social network. DyHGNC [15] captures the dynamic graph structural representations on the global social graph and then extracts the representation vector of the target node that is inserted into the time-aware attention layer for diffusion prediction. MS-HGAT [17] utilizes hypergraphs to depict time-varying users' interaction preferences in the information diffusion process.

However, existing GRL-based diffusion models primarily aim to model the current time-order diffusion sequence based on the independent and invariant diffusion assumptions but ignore the diffusion dynamics and diverse spatio-temporal correlations within the diffusion process. Different from prior works' limited consideration of influencing factors, we focus on three important cascade-related attributes – social connection, user preference, and temporal influence – that affect the state of information propagation. We construct parameter adaptation mechanism for different cascades to learn various propagation patterns and predict the information diffusion more accurately.

2.3. Meta knowledge learning

Meta knowledge learning (MKL) [21] is a very different kind of learning paradigm from traditional supervised learning. In recent years, MKL has been successfully used for real-world applications. Generally, MKL targets at learning prior knowledge from several tasks such that the model can be quickly adapted to new tasks. The most relevant MKL methods here are those who utilize an auxiliary small network to generate main network parameters. For example, the researchers in [22] proposed the method of predicting main network parameters for modeling dynamic temporal sequences.

Dynamic parameter generation approaches have been integrated with MKL to learn the rich sequential context information. For example, hypernetwork [23] employed the dynamic parameter generation technique to learn location- and context-specific semantic functions through a shared meta-network. NSM proposed in [24] built dynamic memory keys for querying sequential information according to data memory. In addition to the aforementioned methods, there are other types of MKL methods proposed to address specific challenges, e.g., MAML [25] is a model-agnostic strategy that designs a meta-gradient updates mechanism to explicitly train model parameters, which has good generalization performance on new tasks with a small number of gradient steps.

To the best of our knowledge, our work is the first attempt to incorporate meta-knowledge learning into the information diffusion prediction task via extracting meta-knowledge from cascade-related influencing factors. This new design allows us to generate more adaptive parameters and capture unique diffusion patterns across different information cascades.

3. Preliminaries

We now formally define the information diffusion prediction problem and provide the necessary background information. Information cascades in social network can be seen as a sequence of actions that disseminates the information (e.g., a tweet in a microblogging system or a paper in an academic network) to a large body of audience. In this paper, our goal is to predict the future infected users of an information item based on the time-ordered sequence of previous affected users. Table 1 summarizes mathematical notations used throughout the paper.

Definition 1 (Social Network). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents a social network, where $\mathcal{V} = \{u_1, \dots, u_N\}$ is the set of N users, \mathcal{E} denotes the set of relations between all users, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix. The adjacency matrix \mathbf{A} represents the follower-followee relationships among users in the social network. For each user pair (u, v) in \mathbf{A} , $A_{uv} = 1$ if user u has followed user v , $A_{uv} = 0$ otherwise.

Table 1
Mathematical symbols.

Symbol	Description
\mathbf{A}	adjacency matrix.
c	information cascade.
\mathbf{d}	meta knowledge vector.
\mathcal{E}	edge set.
\mathcal{G}	social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
\mathcal{G}_{UI}	user-item interaction network.
I, \mathcal{I}	information item and set of information items.
K	length of the information cascade.
\mathcal{K}	temporal kernel.
M	number of items.
N	number of users of social network.
$\mathcal{N}_S(u)$	neighborhood of user u .
\mathcal{O}	set of user-item interactions.
ω	learnable parameters of temporal encoding Ψ .
Ψ	continuous time encoder.
t, T	retweeting time and scalar timesteps.
u, \mathbf{u}	user and user embeddings.
$\mathbf{U}, \bar{\mathbf{U}}$	user embeddings to and out of Meta-GAT.
\mathcal{U}	set of users and items.
\mathbf{v}	meta knowledge vector.
\mathcal{V}	user set.
\mathbf{x}_u	user preference representation.

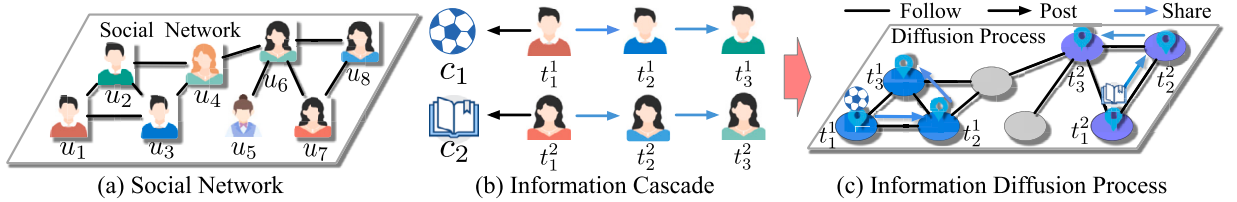


Fig. 3. An example for the information cascade.

Definition 2 (User-Item Interaction Network). Given a set of M information items $I = \{I_1, \dots, I_M\}$, e.g., information that being diffused in a social network, a bipartite graph representing the observed interactions between users \mathcal{V} and information items \mathcal{I} , is given by $\mathcal{G}_{UI} = (\mathcal{U}, \mathcal{O})$, where $\mathcal{U} = \mathcal{V} \cup \mathcal{I}$ denotes the set of all users and items, and \mathcal{O} denotes the set of observed interactions.

Definition 3 (Preference-aware Homophily Ratio). For users u_i and u_j , who have interacted with subsets of information items \mathcal{I}_{u_i} and \mathcal{I}_{u_j} , respectively, we define the edge-wise homophily ratio based on Jaccard similarity [26]:

$$h_{(i,j)} = \frac{|\mathcal{I}_{u_i} \cap \mathcal{I}_{u_j}|}{|\mathcal{I}_{u_i} \cup \mathcal{I}_{u_j}|}. \quad (1)$$

$h_{(i,j)} \in [0, 1]$ represents the degree of similarity between the preferences of two users. Note that the user-user edge with strong homophily has a high homophily ratio $h_{(i,j)} \rightarrow 1$, indicating a greater similarity between the users. Furthermore, $h_{(i,j)} = 0$ is set to 0 when only user u_i or user u_j appears in the training set. By averaging $h_{(i,j)}$ only when $\mathbf{A}_{ij} = 1$, we define the graph-wise homophily ratio:

$$\mathcal{H}_s = \frac{1}{\mathcal{M}} \sum_{(i,j) \in \{\mathbf{A}_{ij}=1\}} h_{(i,j)}, \quad (2)$$

where \mathcal{M} denotes the number of edges in the social network. \mathcal{H}_s reflects the homophily of the holistic social network.

Definition 4 (Information Cascades). Let c denote an information cascade, e.g., a tweet and its retweets, propagating through a social network \mathcal{G} over time. Information cascade is an ordered sequence of variable length user activations. Let $c(t_o)$ be an information cascade observed during time window $[t_1, t_o]$, it is defined as $c(t_o) = \{(u_1, t_1)\} \cup \{(u_j, u_k, t_k) | 2 \leq k \leq K, u_j \in \mathcal{V}, u_k \in \mathcal{V}, j \neq k, t_k \leq t_o\}$, each 3-tuple (u_j, u_k, t_k) represents user u_k is activated by user u_j at time t_k , K is the length of cascade $c(t_o)$. As shown in Fig. 3, the information cascade c_1 is recorded as $c_1 = \{(u_1, t_1^1), (u_1, u_3, t_2^1), (u_3, u_2, t_3^1), \dots\}$ order by timestamp.

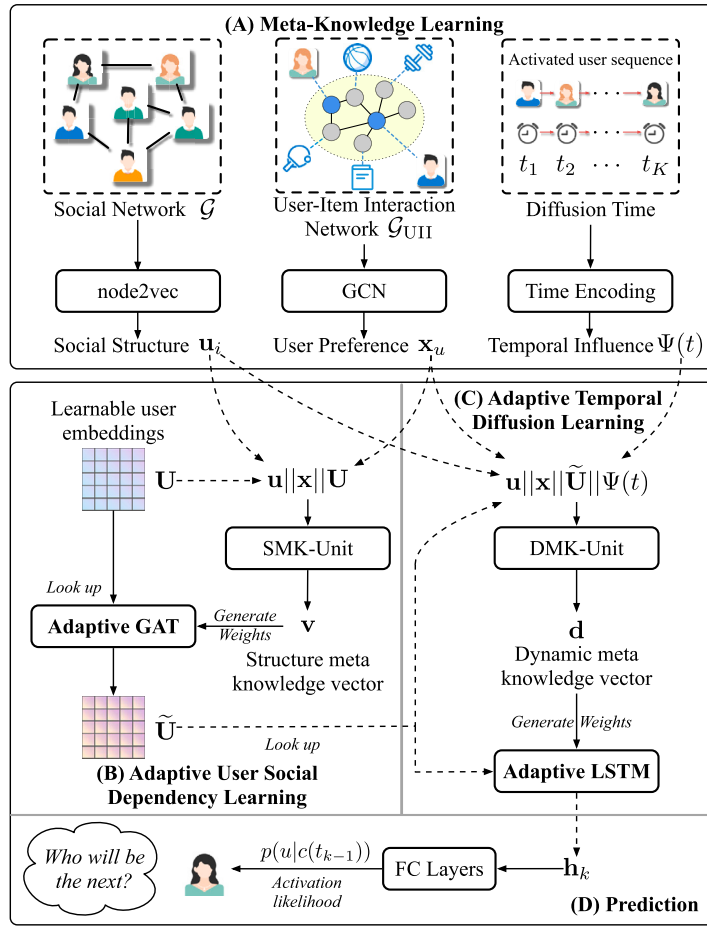


Fig. 4. Overview of our proposed MetaCas. (A) The detailed process of meta-knowledge learning from the social network, user-item interaction network and cascade diffusion timestamps. (B) The detailed illustration of adaptive user social dependency learning. The meta-knowledge learner, i.e., SMK-Unit, leverages the learned user-related knowledge to generate GAT weights, enabling the capture of various social dependencies. (C) The detailed description of adaptive temporal diffusion learning. DMK-Unit encodes cascade-related meta-knowledge to produce desired weights of the adaptive LSTM for extracting the time-varying diversity of correlations. (D) The detailed process of predicting potential users in an information cascade.

Definition 5 (Information Diffusion Prediction). Given a social network \mathcal{G} , a user-item interaction network \mathcal{G}_{UI} , and an information cascade $c(t_o)$, the goal of the information diffusion prediction is to evaluate the activation likelihood $p(u_i|c(t_o))$ for potential users $u_i \in \mathcal{V}$. In short, we want to predict the next activated user given a snapshot of the observed diffusion.

4. Methodology

We now introduce the overall framework of MetaCas and then proceed with detailed descriptions of the model components. MetaCas develops an effective function-level task-adaptive mechanism for dynamic model weights generation via meta-knowledge learning. As shown in Fig. 4, our model is composed by four main components.

(A) Meta-knowledge learning. We construct two meta-knowledge learners: Structure Meta Knowledge Unit (SMK-Unit) and Dynamic Meta Knowledge Unit (DMK-Unit), to respectively learn the meta-knowledge from user attributes (e.g., structures and user preferences) and cascade temporal attributes (e.g., activation time). The learned meta-knowledge is used to produce dynamic weight parameters for downstream topological and sequential modules, enabling further encapsulation of diverse inter-cascade correlations, cascade attributes along with dynamic states, and enhancement of cascade representation capability.

(B) Adaptive user social dependency learning. We create a Meta Graph Attention Network (Meta-GAT) to learn user social relations, which consists an SMK-Unit and an adaptive GAT. SMK-Unit takes user social structures and user preference attributes as input and produces GAT weights. Meta-GAT is able to model neighborhood-varying diversity of correlations and capture diverse user dependencies by broadcasting users' states along social connections.

(C) Adaptive temporal diffusion learning. To capture the time-varying diversity of correlations between information cascades, a Meta-LSTM module is designed that contains a DMK-Unit to encode cascade time attributes and dynamic cascade contexts to learn temporal meta-knowledge, and then produces desired weights for adaptive LSTM.

(D) Predictor. The information cascade representations learned from meta-knowledge are fed into fully-connected (FC) layers for calculating the infection probability of candidate users.

4.1. Meta-knowledge learning

Information cascades evolve dynamically and with heterogeneous characteristics, depending on specific cascade attributes – e.g., user neighborhoods, user preferences and temporal cascade states. We now illustrate how to construct cascade attributes and meta knowledge learners that capture the rich semantics of cascades from different perspectives.

4.1.1. Cascade attributes

We consider three types of cascade attributes: (1) user neighbors encompass information dissemination, influencing how and where the information is disseminated (i.e., paths and destinations); (2) user preferences record the information adoption behaviors; and (3) diffusion time stamps show the diffusion speed, trend and users activity periodicity. We employ three learning nets to obtain the cascade heterogeneous attribute representations.

Social structural representation. To capture the user neighborhood structural characteristics, we employ graph embedding techniques as our primary approach to extract meaningful representations from the social network. These techniques encompass traditional methods like DeepWalk, node2vec, and matrix factorization, as well as GNN-based models such as GCN, GAT, and GraphSage. However, GNN-based models often require a large number of labels to effectively supervise parameter learning, leading to complex implementations when extracting user social correlations during the meta-knowledge learning. Hence, we focus on utilizing the unsupervised graph learning method node2vec [27] to learn a low-dimensional structural representations preserving social network proximity for each user. This method offers ease of implementation and efficiency in modeling network structure. We note that other unsupervised graph embedding techniques may be used, e.g., DeepWalk [28] and matrix factorization [29], etc., depending on different learning targets. Specifically, given a social network \mathcal{G} , a target user $u \in \mathcal{V}$, we denote $\mathcal{N}_S(u_i) \subset \mathcal{V}$ as user u_i 's network neighborhood generated by a node sampling strategy S . The mapping process of node2vec can be summarized as:

$$\mathbf{u}_i = f(\theta; \mathcal{G}, u_i, \mathcal{N}_S(u_i)), \quad (3)$$

where θ is model parameters of node2vec, $\mathbf{u}_i \in \mathbb{R}^{d_u}$ is the learned structural representation of user u_i , and d_u is representation dimension. The mapping function f from user to representation is based on skip-gram model [30]. The optimization process on model parameter θ is to maximize the log-probability of observing neighborhood $\mathcal{N}_S(u_i)$ conditional on user u_i 's representation:

$$\max_{\theta} \sum_{u_i \in \mathcal{V}} \left[-\log \sum_{u_j \in \mathcal{V}} \exp(f(u_i) \cdot f(u_j)) + \sum_{u_k \in \mathcal{N}_S(u_i)} f(u_k) \cdot f(u_i) \right], \quad (4)$$

where the first term of the summation is approximated by negative sampling [31] for efficiency.

User preference representation. Inspired by the collaborative filtering method in recommender systems [32], we propose to learn high-quality user preference features as useful meta-knowledge via historical user-item interaction graph \mathcal{G}_{UI} . A graph convolution network (GCN) is adopted to model the higher-order connectivities and learn user preferences. Formally, given a GCN model with L layers, the propagation of the l -th layer fusion is summarized as:

$$\mathbf{x}_u^{(l)} = \sum_{I \in \mathcal{N}_I(u)} \frac{\mathbf{x}_I^{(l-1)}}{\sqrt{|\mathcal{N}_I(u)|} \sqrt{|\mathcal{N}_I(I)|}}, \quad (5)$$

$$\mathbf{x}_I^{(l)} = \sum_{u \in \mathcal{N}_I(I)} \frac{\mathbf{x}_u^{(l-1)}}{\sqrt{|\mathcal{N}_I(I)|} \sqrt{|\mathcal{N}_I(u)|}}, \quad (6)$$

where $\mathbf{x}_u^{(l)}$ and $\mathbf{x}_I^{(l)}$ denote the feature of user u and item I by l -th representation aggregation layers, respectively. Features $\mathbf{x}_u^{(0)}$ and $\mathbf{x}_I^{(0)}$ are initialized by one-hot encoding. Here $\mathcal{N}_I(u)$ represents the set of items i interacted by user u and $\mathcal{N}_I(I)$ indicates the set of users that interact with item I . The denominator $1/(\sqrt{|\mathcal{N}_I(u)|} \sqrt{|\mathcal{N}_I(I)|})$ denotes a symmetric normalization term. The final user and item representation are obtained through the summation readout function: $\mathbf{x}_u = \sum_{l=0}^L \mathbf{x}_u^{(l)}$ and $\mathbf{x}_I = \sum_{l=0}^L \mathbf{x}_I^{(l)}$. Meanwhile, the Bayesian personalized ranking [33] is utilized to optimize the model parameters of GCN based on user-item interaction graph \mathcal{G}_{UI} . It is a pairwise loss that enforces the prediction value of an observed interaction to be higher than its unobserved counterparts.

Diffusion time representation. Existing works [6] assume that user re-sharing behaviors occur discretely with equal time intervals, which solely emphasize the orders/positions of users during the information diffusion, thereby limiting their capacity in expressing the temporal information. While some works [34] also notice the significance of time span, their models either struggle to capture time differences between past interactions or lack the ability to generalize across various time differences. To effectively encode the temporal information of cascades into a continuous vector space, we employ a time encoding module (T-encoder) [35] to build a continuous time encoder, which maps scalar timestamps into d_T -dimensional vector space, i.e., $\Psi : T \rightarrow \mathbb{R}^{d_T}$. Given two timestamps t_1 and t_2 , $t \in T$, the temporal kernel is defined as $\mathcal{K}(t_1, t_2) = \Psi(t_1) \cdot \Psi(t_2) = \phi(t_1 - t_2), \forall t_1, t_2 \in T$ for some $\phi : [-T, T] \rightarrow \mathbb{R}$. The temporal

kernel is translation-invariant based on Bochner's Theorem, since $\mathcal{K}(t_1 + \tau, t_2 + \tau) = \phi(t_1 - t_2) = \mathcal{K}(t_1, t_2)$ for any constant τ . Formally, the temporal kernel \mathcal{K} is defined as:

$$\mathcal{K}(t_1, t_2) = \mathbb{E}_\omega [\cos(\omega(t_1 - t_2))] = \mathbb{E}_\omega [\cos(\omega t_1) \cos(\omega t_2) + \sin(\omega t_1) \sin(\omega t_2)]. \quad (7)$$

By explicitly representing the temporal features, the temporal embedding is:

$$\Psi(t) \rightarrow \sqrt{\frac{1}{d_T}} \left[\cos(\omega_1 t), \dots, \cos(\omega_{d_T} t), \sin(\omega_{d_T} t) \right], \quad (8)$$

where $\omega = [\omega_1, \dots, \omega_{d_T}]^\top$ are learnable parameters.

4.1.2. Meta-knowledge learner

To further characterize the cascade-specific representations from different cascades, we design two kinds of meta-knowledge learning units, i.e., SMK-Unit and DMK-Unit, to capture inter-cascade correlations between cascades and cascade-related attributes, and then adaptively transform the learned meta-knowledge into dynamic network parameters for downstream Meta-GAT and Meta-LSTM models. As shown in Fig. 4, the SMK-Unit learns the user-related meta-knowledge from social structures and user-to-item interactions. The DMK-Unit is composed by a continuous time-aware attention module to jointly capture the sequential and temporal correlations of users. DMK-Unit extracts cascade and time-related meta-knowledge and is able to learn sequential dependencies between activated users for information diffusion prediction. The details of SMK-Unit and DMK-Unit are shown in Section 4.2 and Section 4.3, respectively.

4.2. Adaptive user social dependency learning

Information is propagated through user connections that imply the diffusion routes and reaches, determining the diffusion dependencies among users in social network. Per social homophily theory [9], users with shared interests tend to be connected and mutually affected. As a result, such user social dependencies are diverse for different user communities and are related to user attributes, e.g., neighborhood and preference. We capture such diverse user social correlations via graph attention mechanism. Since a standard graph attention network (GAT) [20] applies the shared attention mechanism for all users and ignores the correlations between social interdependencies and user-related attributes, we propose an enhanced GAT to model heterogeneous social relationships.

Specifically, to capture diverse user social dependencies, we design a meta graph attention network (Meta-GAT), which includes an adaptive GAT and a SMK-Unit. The model parameters of the attention module are dynamically generated from user-related meta-knowledge and user states via SMK-Unit, which contains the correlations between user attributes and dynamic user embeddings. We use a learnable matrix $\mathbf{U} \in \mathbb{R}^{N \times d_U}$ to represent all user embeddings, d_U is the dimension of user embedding. Let the length of the studied cascade c is K and the user set of c is $\mathcal{V}_c = \{u_1, u_2, \dots, u_K\}$. The input of Meta-GAT is a set of user embeddings looked up from \mathbf{U} , denoted as $\mathbf{U}_c = \{\mathbf{U}_{u_1}, \mathbf{U}_{u_2}, \dots, \mathbf{U}_{u_K}\} \in \mathbb{R}^{K \times d_U}$.

Adaptive GAT. The first component of Meta-GAT is an adaptive linear transformation parameterized by a dynamic weight matrix $\mathbf{W}(\mathbf{v})$, where \mathbf{v} is a meta-knowledge vector generated from the SMK-Unit (detailed later). As we discussed that different pairs of users in the social network have distinctive meta knowledge and user hidden states, the attention score of edge e_{ij} is related to implicit features of users u_i and u_j . The adaptive GAT is used to compute attention coefficients:

$$a_{ij} = \text{ATTENTION}(\mathbf{W}(\mathbf{v}_{u_i})\mathbf{U}_{u_i}, \mathbf{W}(\mathbf{v}_{u_j})\mathbf{U}_{u_j}, \mathbf{v}_{u_i u_j}), \quad (9)$$

where $a_{ij} \in \mathbb{R}^{d_U}$ indicates the importance of \mathbf{U}_{u_i} on \mathbf{U}_{u_j} at each channel. The attention network $\text{ATTENTION}(\cdot)$ is a feedforward neural network. Since attention mechanisms differ from user to user, we employ an adaptive layer that uses two edge-specific parameters γ_{ij} and β_{ij} generated from the meta knowledge vector \mathbf{v} . These parameters scale and shift the user hidden states. The adaptive layer is defined as:

$$a_{ij} = \sigma(\gamma_{ij} \odot \mathbf{a}([\mathbf{W}(\mathbf{v}_{u_i})\mathbf{U}_{u_i} \parallel \mathbf{W}(\mathbf{v}_{u_j})\mathbf{U}_{u_j}]) + \beta_{ij}), \quad (10)$$

where $\mathbf{a} \in \mathbb{R}^{2d_U}$ is a learnable weight vector, \odot is element-wise product, and σ is an activation function. Attention coefficient a_{ij} is calculated for user u_i and user $u_j \in \mathcal{N}(u_i)$. To make all coefficients comparable in the same scale, we normalize them using softmax function:

$$\tilde{a}_{ij} = \frac{\exp(a_{ij})}{\sum_{k \in \mathcal{N}(u_i)} \exp(a_{ik})}. \quad (11)$$

Once we have the coefficients for each user, the overall impact of neighborhoods can be calculated by linear combinations of implicit user features:

$$\tilde{\mathbf{U}}_{u_i} = \text{ReLU} \left(\mathbf{W}(\mathbf{v}_{u_i})\mathbf{U}_{u_i} + \sum_{j \in \mathcal{N}(u_i)} \tilde{a}_{ij} \mathbf{W}(\mathbf{v}_{u_j})\mathbf{U}_{u_j} \right). \quad (12)$$

Obtained user embeddings $\tilde{\mathbf{U}}$ are used as the input of Meta-LSTM (cf. Eq. (15)).

SMK-Unit is used to model the correlations between user hidden states and user attributes, its output is the meta knowledge vector \mathbf{v} , which is employed to generate dynamic weight $\mathbf{W}(\mathbf{v})$ of adaptive GAT. Specifically, for user u_i , the meta knowledge \mathbf{v}_{u_i} is learned by an FC layer after the concatenation of three user representations. Given social structure \mathbf{u}_i , user preference \mathbf{x}_{u_i} , and user hidden states \mathbf{U}_{u_i} :

$$\mathbf{v}_{u_i} = \text{FC}(\mathbf{u}_i || \mathbf{x}_{u_i} || \mathbf{U}_{u_i}), \mathbf{W}(\mathbf{v}_{u_i}) = \text{FC}(\mathbf{v}_{u_i}), \quad (13)$$

$$\gamma_{ij} = \sigma(\mathbf{W}_\gamma(\mathbf{v}_{u_i} + \mathbf{v}_{u_j})), \beta_{ij} = \sigma(\mathbf{W}_\beta(\mathbf{v}_{u_i} + \mathbf{v}_{u_j})), \quad (14)$$

where $||$ is the concatenate operation and σ is the activation function. The meta knowledge vector \mathbf{v} is calculated from user attributes and user hidden states and is used to generate the weights of adaptive GAT.

4.3. Adaptive temporal diffusion learning

In addition to user social dependencies, information cascades have various time-related attributes which impact the time-varying diffusion dynamics in social network. Such dynamics are often learned by sequential models such as LSTM in existing learning models. However, for time-varying diversities from user to user, it is insufficient to use a simple sequential model with shared weights to capture all the temporal correlations for different cascades. For example, a temporal model based on LSTM ignores the heterogeneous correlations in cascades, e.g., user and cascade time attributes. To better learn such time-varying diversities, we propose Meta-LSTM, which consists of an adaptive LSTM and a DMK-Unit. The adaptive LSTM is used to model task-specific representations, whose weights are controlled by a meta knowledge vector \mathbf{d} learned from cascade time attributes and dynamic cascade contexts via the DMK-Unit.

Adaptive LSTM. Given an information cascade c of length K , we first look up user embeddings $\{\tilde{\mathbf{U}}_{u_k} | u_k \in \mathcal{V}_c\} \in \mathbb{R}^{K \times d_U}$ from $\tilde{\mathbf{U}}$ and then use T-encoder to obtain the corresponding temporal embeddings $\{\Psi(t_k) | k \in [1, K]\} \in \mathbb{R}^{K \times d_T}$. The adaptive LSTM is used to encode the temporal cascade sequence, whose parameters are controlled by the meta knowledge vector \mathbf{d} learned by the DMK-Unit. The revised equations of the adaptive LSTM are:

$$\begin{bmatrix} i_k \\ f_k \\ g_k \\ o_k \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(\text{LN} \left(\mathbf{W}(\mathbf{d}_k) \begin{bmatrix} \tilde{\mathbf{U}}_k \\ \mathbf{h}_{k-1} \end{bmatrix} + \mathbf{b}(\mathbf{d}_k) \right) \right)$$

$$c_k = g_k \odot i_k + c_{k-1} \odot f_k, \quad \mathbf{h}_k = o_k \odot \tanh(c_k), \quad (15)$$

where $\mathbf{W}(\mathbf{d}_k) \in \mathbb{R}^{4d_h \times (d_h + d_{\tilde{\mathbf{U}}})}$ and $\mathbf{b}(\mathbf{d}_k) \in \mathbb{R}^{4d_h}$ are dynamic parameters of the adaptive LSTM, d_h is the dimension of \mathbf{h} , \odot denotes the element-wise multiplication, and LN represents the layer normalization. Parameters $\mathbf{W}(\mathbf{d}_k)$ and $\mathbf{b}(\mathbf{d}_k)$ are calculated by meta knowledge vector \mathbf{d} :

$$\mathbf{W}(\mathbf{d}_k) = \begin{bmatrix} f_{W_i}(\mathbf{d}_k) \\ f_{W_f}(\mathbf{d}_k) \\ f_{W_g}(\mathbf{d}_k) \\ f_{W_o}(\mathbf{d}_k) \end{bmatrix}, \quad \mathbf{b}(\mathbf{d}_k) = \begin{bmatrix} f_{b_i}(\mathbf{d}_k) \\ f_{b_f}(\mathbf{d}_k) \\ f_{b_g}(\mathbf{d}_k) \\ f_{b_o}(\mathbf{d}_k) \end{bmatrix}, \quad (16)$$

where f_W and f_b are several FC layers. Therefore, different information cascades have different adaptive LSTMs. Our proposed Meta-LSTM can model the temporal correlations of time-related attributes and dynamic cascade contexts.

DMK-Unit. Meta-knowledge vector \mathbf{d} is employed in adaptive LSTM to generate dynamic parameters $\mathbf{W}(\mathbf{d})$. We obtain \mathbf{d}_k via the proposed DMK-Unit, which is a time-aware multi-head attention module extracting the dynamic meta-knowledge \mathbf{d} from cascade attributes and dynamic cascade contexts. For user u_k , DMK-Unit first captures the inherent correlations by meta-knowledge fusion and then takes the fused representation to produce a meta hidden state \mathbf{z}_k at time t_k :

$$\mathbf{m}_k = \mathbf{u}_k || \mathbf{x}_{u_k} || \tilde{\mathbf{U}}_{u_k} || \Psi(t_k), \quad (17)$$

$$\mathbf{z}_{k,q} = \text{ATTENTION}(\mathbf{m}_k \mathbf{W}_q^{\text{query}}, \mathbf{m}_k \mathbf{W}_q^{\text{key}}, \mathbf{m}_k \mathbf{W}_q^{\text{value}}),$$

$$\mathbf{z}_k = (\mathbf{z}_{k,1} || \mathbf{z}_{k,2} || \dots || \mathbf{z}_{k,q} || \dots || \mathbf{z}_{k,Q}) \mathbf{W}^O, \quad (18)$$

where $\mathbf{W}_q^{\text{query}}, \mathbf{W}_q^{\text{key}}, \mathbf{W}_q^{\text{value}} \in \mathbb{R}^{d_z/Q \times d_z}$, $\mathbf{W}^O \in \mathbb{R}^{d_z \times d_z/Q}$ are learnable parameter matrices, d_z is the dimension of \mathbf{z}_k and Q is the number of attention heads. A mask matrix is introduced here to avoid label leakage. At last, the meta knowledge vector \mathbf{d}_k is learned by the concatenation of meta hidden state \mathbf{z}_k and hidden state \mathbf{h}_{k-1} :

$$\mathbf{d}_k = \text{FC}(\mathbf{z}_k || \mathbf{h}_{k-1}). \quad (19)$$

Algorithm 1 Training of the MetaCas model.

Input: Information cascade $c(t_o)$; social network \mathcal{G} ; user-item interaction network \mathcal{G}_{UII} ; user embeddings \mathbf{U} .
Output: The information diffusion prediction probabilities $\hat{\mathbf{y}}_k$ for all users;

- 1: Initialize model parameters with uniform distribution $\theta \sim U(-1, 1)$;
- 2: /* *Meta-Knowledge Learning* */
- 3: Obtain structural representation \mathbf{u}_i from \mathcal{G} (Eq. (3));
- 4: Compute user preference \mathbf{x}_u (Eqs. (5)-(6));
- 5: Compute diffusion time representation $\Psi(t)$ (Eq. (8));
- 6: **for each** epoch **do**
- 7: /* *Adaptive User Social Dependency Learning* */
- 8: Obtain the meta knowledge vector \mathbf{v} via SMK-Unit (Eq. (13));
- 9: Compute the adaptive parameter $\mathbf{W}(v_{ij})$, γ_{ij} and β_{ij} of Meta-GAT via SMK-Unit (Eq. (13)-(14));
- 10: Obtain user embedding $\tilde{\mathbf{U}}$ from \mathcal{G} via the adaptive GAT (Eqs. (9)-(12));
- 11: /* *Adaptive Temporal Diffusion Learning* */
- 12: Obtain the meta-knowledge vector \mathbf{d} via DMK-Unit (Eqs. (17)-(19));
- 13: Compute the adaptive parameter $\mathbf{W}(d_k)$ and $\mathbf{b}(d_k)$ of Meta-LSTM via DMK-Unit (Eqs. (16));
- 14: Compute cascade representation \mathbf{h}_k via Meta-LSTM according to Eq. (15);
- 15: /* *Information Diffusion Prediction* */
- 16: Compute diffusion probabilities $\hat{\mathbf{y}}_k$ in Eq. (20);
- 17: Update model parameters θ via loss in Eq. (21);
- 18: **end for**

4.4. Diffusion prediction network

For the final prediction of the information diffusion, the probabilities $\hat{\mathbf{y}}_k \in \mathbb{R}^{N \times 1}$ for all users are calculated by:

$$\hat{\mathbf{y}}_k = \text{softmax}(\mathbf{W}_p \mathbf{h}_k + \mathbf{M}_{\text{mask}}), \quad (20)$$

where \mathbf{W}_p is a transformation matrix that maps the adaptive LSTM hidden states into user-specific space, and \mathbf{M}_{mask} is used to mask users who have already been activated. The information diffusion prediction can be regarded as a sequential multi-label classification task, we adopt the cross entropy loss as objective:

$$\mathcal{L} = - \sum_{k=1}^K \mathbf{y}_k \log \sigma(\hat{\mathbf{y}}_k) + (1 - \mathbf{y}_k) \log(1 - \sigma(\hat{\mathbf{y}}_k)), \quad (21)$$

where \mathbf{y} is ground truth labels and $\hat{\mathbf{y}}$ is predicted labels.

4.5. Discussion

4.5.1. Relation with graph neural network

In our approach, we introduce the user-related meta-knowledge (i.e., social structure and user preference) into the GAT model, enabling the capture of diverse user social correlations via the adaptive attention mechanism. Inversely, the conventional GCN model [19] utilizes the graph convolution operation to aggregate neighborhood knowledge, treating all node neighbors as equally important. However, this approach is inadequate for learning various user-specific social interactions. Furthermore, GAE and VGAE [36] follow the generate-based paradigm to learn node semantic knowledge. However, the AE-based and VAE-based methods are susceptible to posterior collapse issues, which can lead to suboptimal learning of user social relationships. In generate-based GAE and VGAE, there is a potential for uncertainty and noise to be introduced into user presentations learned from the social network. In contrast, our proposed Meta-GAT offers a natural approach to encode diverse user social dependencies via integrating user-related meta-knowledge, which incorporates adaptive updates of user-specific attention weights through generating the corresponding parameters via the structural meta-knowledge learner (i.e., SMK-Unit).

4.5.2. Complexity analysis

Since online social platforms typically contain millions or even billions of users, it is critical to efficiently model structure and time dependencies in the information diffusion process for information diffusion prediction. We conduct complexity analysis on MetaCas's main components:

- *Complexity of computing user social dependency from the social network.* The time complexity for computing $\tilde{\mathbf{U}}$ features in Meta-GAT's attention head can be expressed as $O(|\mathcal{V}|d_{\mathcal{U}}d_{\tilde{\mathcal{U}}} + |\mathcal{E}|d_{\tilde{\mathcal{U}}})$, where $d_{\mathcal{U}}$ and $d_{\tilde{\mathcal{U}}}$ represent the input and output dimensions of users, respectively, $|\mathcal{V}|$ and $|\mathcal{E}|$ indicate the number of nodes and edges in the social network.
- *Complexity of temporal diffusion modeling.* The model parameters of Meta-LSTM are generated from the DMK-Unit, and the structure of Meta-LSTM is the same as the regular LSTM model. As a result, the time complexity of Meta-LSTM is $O(d_{\mathbf{h}}d_{\tilde{\mathcal{U}}} + d_{\mathbf{h}}^2 + d_{\mathbf{h}})$, which is related to the input feature dimension and the dimension of latent variables from Meta-LSTM.
- *Complexity of meta-knowledge learners.* In SMK-Unit, it aims at generating adaptive model parameters for Meta-GAT, which is related to cascade social attributes – i.e., social structural representation and user preference representation. The time complexity

of SMK-Unit is $O(|\mathcal{V}|(d_u + d_x + d_v))$, which is linear to the number of users in the network. In DMK-Unit, it uses the multi-head self-attention mechanism to learn meta-knowledge for meta-LSTM. The computational complexity of DMK-Unit is $O(L^2 d_z)$, where L is the total number of infected users in an information cascade. The main computational complexity of meta-knowledge learners is related to the matrix product when calculating the similarity of a cascade sequence, which is $\mathcal{O}(L^2 d_z)$ in total.

Overall, the complexity of MetaCas comes from the matrix and vector operations. Generally, model-related operations could be accelerated by GPU/TPU devices. The training process of MetaCas is summarized in Algorithm 1.

5. Experiments

In this section, we report the evaluation results and demonstrate the effectiveness of our proposed MetaCas. Besides performance comparison results, we also conduct ablation study, sensitivity analysis and case study.

5.1. Datasets

To verify the generalizability of MetaCas, we present our experimental results on four publicly available information diffusion datasets from two social platforms *Twitter* and *Douban*, and two online websites *Android* and *Memes*. The statistics of four datasets are summarized in Table 2.

- **Twitter** dataset [37] includes 3,454 tweets and their retweet user sequences in October 2010. Each tweet's spreading path is interpreted as an independent information cascade diffused among users. We use *following* relationships between users on Twitter to build the social graph.
- **Douban** dataset [38] is collected from a Chinese social website where users can share items of interest including books, movies and music. The information cascades in Douban dataset are formed by user book-sharing behaviors. Co-occurrence relationships (e.g., users who read the same book) are assumed to be the interests they have in common.
- **Android** dataset [39] is collected from an online Q&A forum where users can post various questions and answers on a series of issues related to their interests. The answering path of each question is treated as an independent information cascade that spreads among users. User interactions on website channels, e.g., questioning and answering, are considered as their interest/expertise relations.
- **Memes** dataset [40] is a collection of memes from online news websites. Each cascade records the propagation process of a specific key phrase and is described by a sequence of webpage links with the corresponding timestamps. The network structure is established based on the presence of common key phrases among the corresponding webpage links.

5.2. Baselines

To verify whether our proposed method is effective for information diffusion prediction task, we compare MetaCas with following 11 strong baselines, which can be categorized into three types: independent cascade-based models, diffusion path-based models and social network-based models. The detailed descriptions of all baselines can be summarized as follows:

(1) Independent cascade-based models:

- **TUIC** [11] proposes a time-aware utility-driven independent cascade framework that combines time-aware multi-item propagation, utility-driven item adoption, and mixed item relationships.
- **Emb-IC** [41] constructs an embedded cascade framework that extends the conventional independent cascade model to extract user representations from partial orders of user activations.
- **Inf2vec** [42] designs an influence embedding method that incorporates the local propagation structure, user co-occurrence and temporal influence to capture the dynamics of information diffusion.

(2) Diffusion path-based methods:

- **Topo-LSTM** [40] extends the vanilla LSTM model to exploit the local diffusion structure of a dynamic directed acyclic graph (DAG) via a dynamic DAG-LSTM.
- **DeepDiffuse** [13] combines the temporal point process and LSTM model to learn temporal information and user sequences for diffusion prediction. It also employs a Transformer module, which enhances the LSTM model to sequentially learn the characteristics of activated users.
- **NDM** [34] models the microscopic cascade diffusion process through self-attention mechanism and convolution neural networks, making relaxed independence assumptions to alleviate the long-term dependencies.

(3) Social network-based methods:

- **SNIDSA** [14] computes structure attention over the diffusion structure of a local cascade graph and incorporates sequential information of cascades through a Gating-RNN.

Table 2
Dataset statistics.

Dataset	Twitter	Douban	Android	Memes
# Cascade	3,454	3,485	679	4,991
# Users	12,627	12,232	9,958	4,155
# Edges in \mathcal{G}	309,631	396,580	48,573	2,716,864
Avg. length	38.22	23.09	41.74	27.43
# Train	2,763	2,788	543	3992
# Val	345	348	68	499
# Test	346	349	68	500

Table 3
Information diffusion performance comparison on Twitter.

Model	Twitter					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
TUIC	5.68 _{±0.20}	16.83 _{±0.25}	24.98 _{±0.19}	6.10 _{±0.13}	6.38 _{±0.15}	7.06 _{±0.17}
Emb-IC	8.78 _{±0.15}	26.69 _{±0.10}	41.93 _{±0.19}	16.33 _{±0.10}	17.42 _{±0.14}	17.59 _{±0.11}
Inf2vec	9.68 _{±0.11}	28.97 _{±0.13}	43.71 _{±0.21}	18.11 _{±0.10}	16.40 _{±0.13}	18.88 _{±0.12}
DeepDiffuse	5.72 _{±0.21}	15.41 _{±0.24}	21.61 _{±0.23}	5.93 _{±0.14}	6.89 _{±0.14}	6.99 _{±0.13}
Topo-LSTM	10.45 _{±0.22}	18.89 _{±0.46}	25.42 _{±0.33}	9.51 _{±0.20}	13.68 _{±0.34}	14.68 _{±0.30}
NDM	22.45 _{±0.24}	30.71 _{±0.47}	35.12 _{±0.42}	15.59 _{±0.14}	15.97 _{±0.15}	16.03 _{±0.15}
SNIDSA	25.67 _{±0.28}	37.24 _{±0.47}	43.59 _{±0.41}	16.34 _{±0.35}	17.64 _{±0.25}	18.89 _{±0.26}
FOREST	30.28 _{±0.49}	42.65 _{±0.68}	50.12 _{±0.48}	21.45 _{±0.31}	21.96 _{±0.32}	22.36 _{±0.32}
Inf-VAE	14.93 _{±0.52}	33.52 _{±0.58}	46.42 _{±0.51}	19.83 _{±0.41}	20.68 _{±0.42}	21.82 _{±0.42}
DyHGCN	32.18 _{±0.26}	45.85 _{±0.41}	52.79 _{±0.38}	22.87 _{±0.12}	23.48 _{±0.22}	23.98 _{±0.24}
MS-HGAT	34.63 _{±0.16}	47.52 _{±0.31}	54.29 _{±0.11}	24.02 _{±0.09}	24.51 _{±0.17}	24.61 _{±0.17}
MetaCas	37.68 _{±0.16}	53.99 _{±0.23}	63.31 _{±0.14}	25.77 _{±0.08}	26.56 _{±0.08}	26.69 _{±0.08}
(improves)	8.81%	13.61%	16.61%	7.28%	10.65%	8.41%

Table 4
Information diffusion performance comparison on Douban.

Model	Douban					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
TUIC	1.65 _{±0.25}	5.69 _{±0.24}	9.99 _{±0.22}	1.77 _{±0.15}	1.18 _{±0.19}	1.22 _{±0.20}
Emb-IC	7.69 _{±0.22}	18.91 _{±0.15}	25.83 _{±0.16}	6.98 _{±0.18}	7.33 _{±0.14}	7.88 _{±0.18}
Inf2vec	8.77 _{±0.21}	20.01 _{±0.18}	27.31 _{±0.19}	7.71 _{±0.10}	7.73 _{±0.13}	8.13 _{±0.11}
DeepDiffuse	9.52 _{±0.31}	14.83 _{±0.11}	20.15 _{±0.33}	6.30 _{±0.26}	6.75 _{±0.26}	6.80 _{±0.26}
Topo-LSTM	8.97 _{±0.28}	16.33 _{±0.41}	21.57 _{±0.42}	6.67 _{±0.24}	7.63 _{±0.12}	7.88 _{±0.10}
NDM	7.43 _{±0.35}	15.65 _{±0.37}	20.62 _{±0.35}	3.07 _{±0.12}	3.44 _{±0.12}	3.51 _{±0.12}
SNIDSA	10.23 _{±0.18}	20.24 _{±0.37}	28.12 _{±0.39}	7.12 _{±0.24}	7.64 _{±0.11}	7.79 _{±0.13}
FOREST	13.92 _{±0.18}	24.42 _{±0.32}	30.35 _{±0.47}	7.66 _{±0.17}	8.14 _{±0.16}	8.23 _{±0.16}
Inf-VAE	10.94 _{±0.24}	21.02 _{±0.34}	34.72 _{±0.30}	7.32 _{±0.20}	7.98 _{±0.19}	8.03 _{±0.20}
DyHGCN	17.71 _{±0.14}	31.33 _{±0.39}	38.71 _{±0.35}	9.51 _{±0.12}	10.26 _{±0.09}	10.36 _{±0.07}
MS-HGAT	18.91 _{±0.10}	32.51 _{±0.37}	40.40 _{±0.40}	10.04 _{±0.08}	10.67 _{±0.07}	10.78 _{±0.07}
MetaCas	19.55 _{±0.17}	34.34 _{±0.14}	42.39 _{±0.12}	10.65 _{±0.14}	11.35 _{±0.14}	11.46 _{±0.14}
(improves)	3.38%	5.63%	5.86%	6.07%	6.37%	6.30%

- **FOREST** [6] predicts the information diffusion based on GRU and GNNs. It builds a structural context extraction mechanism to better characterize the influence of the social graph.
- **Inf-VAE** [39] designs a variational architecture combining a variation graph auto-encoder and a co-attention network to jointly learn social structure relations and sequential composition information. Social homophily and temporal influence are jointly considered to evaluate the set of all candidate users.
- **DyHGCN** [15] considers heterogeneous graphs that contain the social and diffusion relations of users. It implements GCN to obtain the cascade structure representations and utilizes multi-head attention to model the context-dependencies of cascade sequence.
- **MS-HGAT** [17] is the state-of-the-art diffusion prediction model. It develops hypergraph attention networks to obtain the cascade structural information from sequential hypergraphs. Then, it uses the gated fusion strategy to fuse the hypergraph structural information and static social structural information to facilitate prediction performance.

Table 5
Information diffusion performance comparison on Android.

Model	Android					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
TUIC	1.78 _{±0.18}	2.56 _{±0.16}	3.94 _{±0.15}	1.49 _{±0.12}	1.51 _{±0.10}	1.48 _{±0.13}
Emb-IC	2.93 _{±0.19}	9.82 _{±0.25}	17.23 _{±0.16}	3.45 _{±0.17}	2.48 _{±0.13}	2.67 _{±0.15}
Inf2vec	3.11 _{±0.20}	10.40 _{±0.18}	16.37 _{±0.15}	3.57 _{±0.10}	2.57 _{±0.09}	2.82 _{±0.11}
DeepDiffuse	3.77 _{±0.57}	10.98 _{±0.39}	18.26 _{±0.60}	1.58 _{±0.13}	1.88 _{±0.19}	2.02 _{±0.12}
Topo-LSTM	4.66 _{±0.58}	12.83 _{±0.68}	16.73 _{±0.72}	3.63 _{±0.25}	4.15 _{±0.46}	4.19 _{±0.53}
NDM	4.90 _{±0.19}	13.55 _{±0.46}	18.03 _{±0.61}	2.69 _{±0.04}	2.88 _{±0.04}	2.91 _{±0.04}
SNDSA	5.63 _{±0.57}	14.62 _{±0.49}	20.93 _{±0.53}	2.98 _{±0.42}	3.24 _{±0.39}	3.97 _{±0.43}
FOREST	6.20 _{±0.42}	12.22 _{±1.24}	17.10 _{±1.91}	3.72 _{±0.31}	3.98 _{±0.34}	4.05 _{±0.35}
Inf-VAE	5.05 _{±0.39}	13.07 _{±0.21}	19.54 _{±1.65}	4.52 _{±0.56}	4.97 _{±0.51}	5.24 _{±0.56}
DyHGCN	9.13 _{±0.30}	16.48 _{±0.45}	23.09 _{±0.51}	6.09 _{±0.15}	6.40 _{±0.15}	6.50 _{±0.20}
MS-HGAT	10.41 _{±0.20}	20.23 _{±0.38}	27.89 _{±0.72}	6.46 _{±0.05}	6.88 _{±0.04}	6.99 _{±0.04}
MetaCas	11.18 _{±0.09}	20.92 _{±0.25}	28.96 _{±0.25}	6.82 _{±0.05}	7.19 _{±0.05}	7.24 _{±0.04}
(improves)	7.40%	3.41%	3.83%	5.57%	4.50%	3.57%

Table 6
Information diffusion performance comparison on Memes.

Model	Memes					
	Hits@10	Hits@50	Hits@100	MAP@10	MAP@50	MAP@100
TUIC	9.91 _{±0.14}	20.39 _{±0.17}	31.44 _{±0.18}	4.19 _{±0.11}	5.43 _{±0.09}	5.59 _{±0.12}
Emb-IC	34.12 _{±0.16}	54.16 _{±0.14}	63.05 _{±0.14}	16.27 _{±0.18}	17.24 _{±0.10}	17.37 _{±0.11}
Inf2vec	36.32 _{±0.11}	55.54 _{±0.18}	64.95 _{±0.13}	17.17 _{±0.12}	18.44 _{±0.14}	18.67 _{±0.16}
DeepDiffuse	11.98 _{±0.22}	29.78 _{±0.18}	41.59 _{±0.24}	6.85 _{±0.15}	7.65 _{±0.18}	7.81 _{±0.14}
Topo-LSTM	27.58 _{±0.26}	42.99 _{±0.28}	54.77 _{±0.30}	10.56 _{±0.20}	11.54 _{±0.15}	11.97 _{±0.13}
NDM	31.38 _{±0.15}	51.89 _{±0.20}	60.67 _{±0.33}	14.86 _{±0.07}	15.84 _{±0.07}	15.97 _{±0.07}
SNDSA	32.58 _{±0.22}	52.99 _{±0.20}	61.69 _{±0.36}	15.96 _{±0.11}	16.81 _{±0.13}	16.99 _{±0.07}
FOREST	39.38 _{±0.45}	62.35 _{±0.20}	71.56 _{±0.17}	19.36 _{±0.30}	20.48 _{±0.29}	20.71 _{±0.28}
Inf-VAE	12.23 _{±0.45}	33.39 _{±0.67}	45.96 _{±1.16}	19.15 _{±0.07}	20.03 _{±0.29}	20.18 _{±0.40}
DyHGCN	38.95 _{±0.44}	62.89 _{±0.39}	72.48 _{±0.22}	19.48 _{±0.26}	20.59 _{±0.24}	20.73 _{±0.26}
MS-HGAT	40.91 _{±0.48}	65.01 _{±0.29}	74.48 _{±0.18}	19.51 _{±0.21}	20.69 _{±0.21}	20.83 _{±0.21}
MetaCas	42.54 _{±0.19}	67.25 _{±0.09}	76.80 _{±0.11}	20.95 _{±0.11}	22.15 _{±0.08}	22.29 _{±0.11}
(improves)	3.98%	3.44%	3.11%	7.38%	7.05%	6.97%

5.3. Evaluation metrics

Following [17], we rank candidate users in the social network according to their activation probabilities. Then we adopt two widely used ranking metrics: Hits score on Top-K (Hits@K) and Mean Average Precision on Top-K (MAP@K), where K is set to {10, 50, 100}. Generally, MAP@K reflects the existence and position of ground-truth (infected users) in the rank list, while Hits@K pays more attention to the accuracy among the Top-K predicted users. The higher the score, the better the performance.

5.4. Parameter settings

For each dataset, we randomly selected 80% cascades for training, 10% for validation and the remaining 10% for testing. All models are tuned to the best performance according to early stopping strategy when validation errors are not declined for five consecutive epochs. Following previous works [17], the maximum cascade length is set to 200, the dimension of user representation is set to 64 for all methods. Other hyperparameters for baselines are set to the recommended values described in their papers. For MetaCas, the batch size is 64 and the attention head number Q of DMK-Unit is 8. Model parameters are updated by Adam optimizer. For the node2vec method, we utilize the rid search to identify the optimal parameters.

5.5. Performance comparison

The performance comparison between MetaCas and baselines on four datasets is presented in Tables 3 to 6. We run all models on each dataset five times and report the mean and standard deviation. The best results are in **bold** font and the second underlined. We have the following notable observations based on our experimental evaluations:

(O1): We can see from the tables that our proposed MetaCas outperforms eight strong baselines consistently on four datasets for information diffusion prediction, which verifies the effectiveness and generalizability of MetaCas. Specifically, on Twitter dataset, the performance of MetaCas improves the best baseline (MS-HGAT) by 8.81%, 13.61%, and 16.61% in terms of Hits@10, Hits@50,

Table 7
Ablation study of MetaCas on four datasets.

Model	Twitter		Douban		Android		Memes	
	Hits	MAP	Hits	MAP	Hits	MAP	Hits	MAP
MetaCas	63.31 _{±0.14}	26.69 _{±0.08}	42.39 _{±0.12}	11.46 _{±0.14}	28.96 _{±0.25}	7.24 _{±0.04}	76.80 _{±0.11}	22.29 _{±0.11}
w/o Meta-GAT	58.36 _{±0.02}	25.94 _{±0.01}	41.43 _{±0.14}	11.07 _{±0.16}	27.73 _{±0.00}	7.13 _{±0.01}	75.72 _{±0.10}	21.48 _{±0.14}
w/o Meta-LSTM	53.82 _{±0.04}	22.36 _{±0.13}	36.97 _{±0.08}	10.61 _{±0.10}	23.23 _{±0.01}	6.41 _{±0.03}	69.14 _{±0.07}	19.66 _{±0.08}
vanilla GAT	60.69 _{±0.41}	26.05 _{±0.10}	41.82 _{±0.32}	11.13 _{±0.20}	28.30 _{±0.64}	7.07 _{±0.13}	75.74 _{±0.22}	21.72 _{±0.15}
vanilla LSTM	61.85 _{±0.16}	25.54 _{±0.63}	40.66 _{±0.22}	10.49 _{±0.32}	27.95 _{±0.39}	6.89 _{±0.21}	75.50 _{±0.16}	21.30 _{±0.17}
DeepWalk	62.98 _{±0.19}	26.13 _{±0.12}	41.54 _{±0.09}	11.08 _{±0.15}	28.05 _{±0.15}	6.98 _{±0.12}	76.19 _{±0.10}	21.98 _{±0.11}
LINE	63.12 _{±0.22}	26.28 _{±0.15}	42.04 _{±0.19}	11.18 _{±0.14}	28.53 _{±0.20}	7.01 _{±0.13}	76.21 _{±0.18}	22.01 _{±0.14}
SMF	63.11 _{±0.10}	26.58 _{±0.08}	41.92 _{±0.07}	11.13 _{±0.08}	28.79 _{±0.10}	7.13 _{±0.05}	76.44 _{±0.15}	22.18 _{±0.12}
LT-Encoder	63.15 _{±0.20}	26.38 _{±0.19}	41.63 _{±0.16}	11.05 _{±0.13}	27.98 _{±0.25}	7.15 _{±0.22}	75.19 _{±0.19}	21.25 _{±0.23}
PT-Encoder	63.25 _{±0.12}	26.43 _{±0.16}	41.89 _{±0.20}	11.26 _{±0.14}	28.38 _{±0.22}	7.22 _{±0.23}	76.10 _{±0.20}	21.43 _{±0.19}
w/o SSR	63.07 _{±0.15}	26.24 _{±0.17}	42.26 _{±0.10}	11.12 _{±0.08}	28.35 _{±0.24}	7.17 _{±0.12}	76.51 _{±0.09}	22.01 _{±0.07}
w/o UPR	63.22 _{±0.17}	26.39 _{±0.12}	42.27 _{±0.14}	11.18 _{±0.10}	28.14 _{±0.10}	7.04 _{±0.04}	76.50 _{±0.10}	22.02 _{±0.13}
w/o DTR	63.13 _{±0.24}	26.33 _{±0.22}	41.50 _{±0.16}	10.92 _{±0.15}	27.91 _{±0.40}	7.19 _{±0.15}	74.99 _{±0.16}	20.65 _{±0.18}

and Hits@100, respectively. Compared to these competitive baselines, our MetaCas model achieves superior prediction accuracy, due to its adaptive cascade-specific meta-knowledge learning. Specifically, our approach employs two meta-knowledge learners that effectively capture cascade-related social dependencies and temporal influences, enabling the generation of cascade-specific meta-knowledge. Additionally, the learned meta-knowledge is utilized to adaptively adjust the model parameters of topological-temporal modules (i.e., Meta-GAT and Meta-LSTM), which not only extract rich information of user preferences from historical interactions, but also incorporate valuable cascade semantic knowledge, enhancing information diffusion prediction. Inversely, other baselines only capture limited contextual information from social networks and user re-sharing behaviors via the topological-temporal modeling. Meanwhile, these results show explicit benefits of learning topological-temporal meta-knowledge for information diffusion prediction.

(O2): Diffusion path-based methods (e.g., Topo-LSTM, DeepDiffuse and NDM) that only models user activation correlations within an independent cascade, perform markedly worse than their counterparts. It is because they simply learn user representations following the sequential assumption and ignore user social relationships.

(O3): In terms of IC-based models, we have observed relatively small gaps between IC-based models (i.e., Emb-IC and Inf2vec) and social network-based methods. In certain cases, IC-based models have even outperformed diffusion path-based methods and certain social network-based models, implying that deep learning models are not always better than IC-based methods. However, it is noteworthy that TUIC performs significantly worse compared to its counterparts. These findings suggest that the incorporation of graph embedding-based methods into independent cascade models (i.e., Emb-IC and Inf2vec) enhances the modeling capability compared to the conventional independent cascade methods (i.e., TUIC). Nonetheless, our MetaCas still outperforms IC-based models by a considerable margin. We attribute the performance deficiency to the inherent incapability of above baselines in effectively modeling comprehensive cascade contextual information (i.e., user-related and cascade-related meta-knowledge).

(O4): For models that jointly learn social topology derived from social network and temporal correlations of the diffusion behaviors among users (e.g., SNISDA, FOREST, and Inf-VAE), they generally perform better than cascade-based models. This result indicates that the rich information implied by social structures can indeed help the task of information diffusion prediction. DyHGCM and MS-HGAT, on the contrary, consider the complex diffusion behaviors and user dynamic preference from a dynamic graph perspective, achieving better performance than other baselines. However, they still fall short of learning spatio-temporal correlations, cascade attributes, dynamic cascade hidden states, and the heterogeneity between cascades. MetaCas closes these gaps by utilizing two meta-knowledge learners (SMK-Unit and DMK-Unit) to learn adaptive user embeddings and expressive cascade representations.

5.6. Ablation study and parameter analysis

To investigate the contributions of different components and cascade attributes in MetaCas, we conduct ablation studies by examining the performance change after removing components in MetaCas. We design seven variants of MetaCas, they are:

- **w/o Meta-GAT:** without user social dependencies, i.e., we remove the Meta-GAT module in MetaCas and only employ the user embeddings looked up from \mathbf{U} instead of $\tilde{\mathbf{U}}$.
- **w/o Meta-LSTM:** contrary to *w/o Meta-GAT*, this variant removes the Meta-LSTM module and directly uses $\tilde{\mathbf{U}}$ to predict the next activated users.
- **vanilla GAT:** we use a standard graph attention network (GAT) to replace the Meta-GAT module for modeling user social dependencies.
- **vanilla LSTM:** we use a vanilla LSTM model to replace the Meta-LSTM for modeling the time-varying diversities of activated users in cascade.

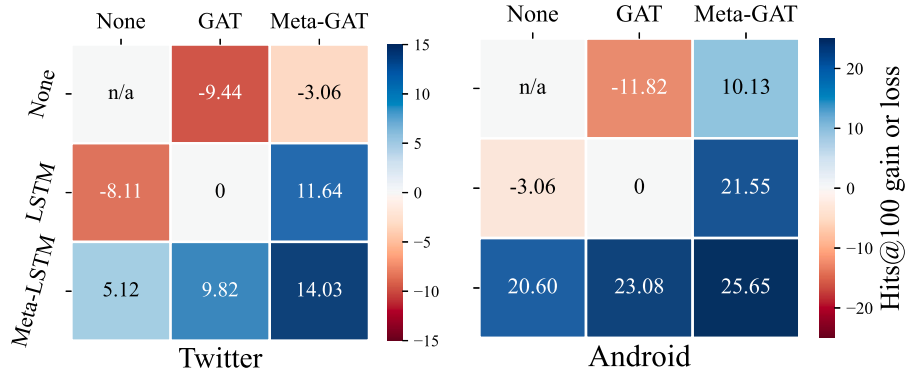


Fig. 5. Meta-knowledge learner evaluation on Twitter and Android datasets measured by Hits@100. The colors denote Hits@100 improvement gain or loss (%) compared to the base model (i.e., GAT-LSTM, the mid cell).

- **DeepWalk:** We introduce a variant model that leverages the graph embedding – DeepWalk [28] – as a substitute for the node2vec module in the process of learning social structural representations.
- **LINE:** In the process of learning social structural representations, we design a variant model that utilizes the graph embedding method, i.e., LINE [43], to replace the node2vec module.
- **SMF:** We construct a variant that utilizes the sparse matrix factorization [29] as a replacement for the node2vec method.
- **LT-Encoder:** Inspired by the baseline NDM, we propose a variant model that utilizes a linear layer to transform the timestamps into a low-dimensional vector representation.
- **PT-Encoder:** This variant utilizes the position embedding to model the temporal information.
- **w/o social structural representation (SSR):** this variant removes the user embeddings \mathbf{u} that previously as part of the input of Meta-GAT and Meta-LSTM.
- **w/o user preference representation (UPR):** this variant removes the user preference \mathbf{x}_u that previously as part of the input of Meta-GAT and Meta-LSTM.
- **w/o diffusion time representation (DTR):** this variant removes the temporal embedding $\Psi(t)$ that previously as part of the input of Meta-LSTM.

5.6.1. Ablation study

The ablation results are shown in Table 7, where we compare the performance of MetaCas with seven variant models on Twitter and Android datasets. The results align with our expectation that all modules in MetaCas contribute to the information diffusion prediction performance in terms of Hits@100 and MAP@100. Specifically, we have the following findings:

(1): Compared to variants that remove Meta-GAT or Meta-LSTM, we can see that removing one of the cascade attribute representations (w/o SSR/UPR/DTR) would not severely decrease the prediction performance. This property of MetaCas is especially useful when we lack specific content in practical situations due to privacy policies.

(2): There are significant performance degradations when we remove Meta-GAT or Meta-LSTM modules. In particular, Meta-LSTM is more effective than Meta-GAT. This is attributed to the fact that Meta-GAT primarily models user social correlations from the social network, while Meta-LSTM focuses on the learning of cascade temporal dependencies through user retweet sequences. The social network typically reflects the global relationships among users but does not incorporate the actual process of information diffusion. User retweet sequences exhibit real local retweet correlations among users, which may not be captured in the social network alone. Furthermore, the inputs of Meta-LSTM both have user structural and preference embeddings \mathbf{u} and \mathbf{x} . Thus, the Meta-LSTM does learn user structures and preferences to some extent, resulting in significant prediction results compared to Meta-GAT. Nevertheless, removing Meta-GAT still causes noticeable performance degradation. This verifies the effectiveness of Meta-GAT which can learn diverse user social interdependencies and enhance the learned user embeddings $\tilde{\mathbf{U}}$.

(3): During the process of learning social structural representations, we design various variant models to analyze the impact of different unsupervised graph embedding techniques, such as DeepWalk, LINE and sparse matrix factorization, on the model performance. In Table 7, we find that the prediction performance experiences a slight decrease when the node2vec method is replaced. This finding suggests that our MetaCas model can be readily substituted with other unsupervised graph embedding techniques. Moreover, it validates the scalability of our MetaCas method.

(4): In the process of capturing diffusion time representation, we construct multiple variant models to investigate the influence of different time encoding methods, such as LT-Encoder and PT-Encoder, on the performance of the models. We observe that using other temporal encoding methods leads to decrease the prediction performance. This result indicates the effectiveness of our T-encoder design for the continuous time encoding.

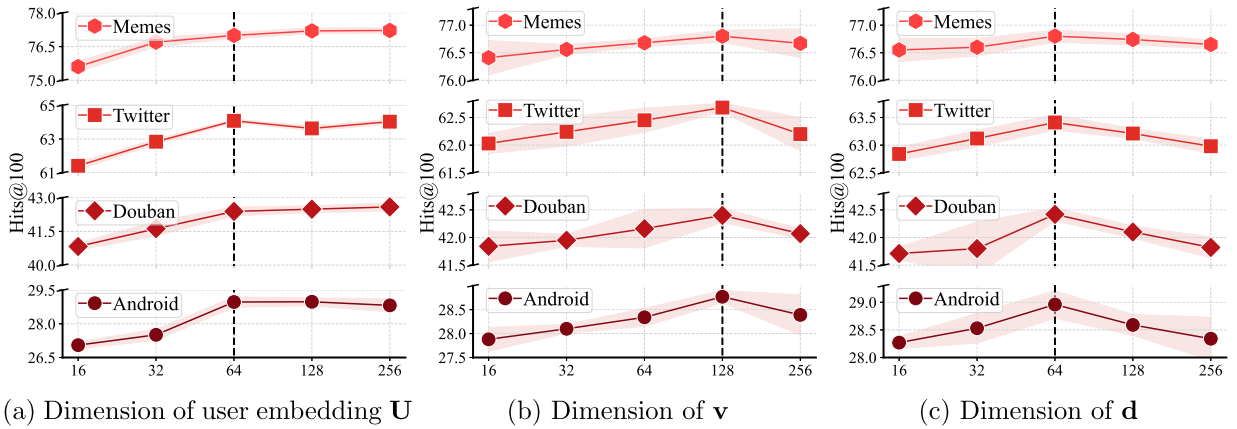


Fig. 6. Hyperparameter sensitivity of MetaCas on four datasets. We run each model five times and report the mean and standard deviation of Hits@100.

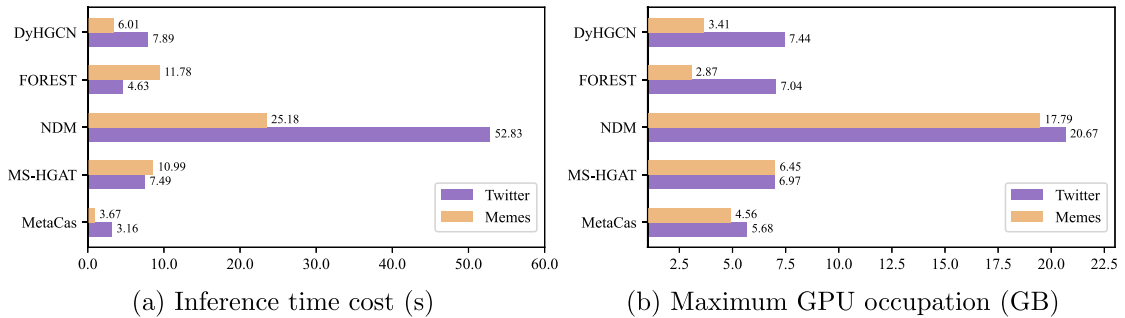


Fig. 7. Efficiency analysis on the Twitter and Meme datasets. The batch size is configured as 64.

(5): When we simply use the vanilla versions of GAT or LSTM, the performances of MetaCas are significantly decreased. This performance discrepancy indicates that our designed two meta-knowledge learners (SMK-Unit and DMK-Unit) can generate efficacious model weights for the two topological/temporal learning nets (adaptive GAT and LSTM).

5.6.2. Meta-knowledge learners

To further verify the effectiveness of Meta-GAT and Meta-LSTM, we show performance gain or loss in terms of Hits@100 when we use different combinations of learners on Twitter and Android datasets in Fig. 5. The reference model consists of a vanilla GAT and a vanilla LSTM (i.e., the mid cell). We can observe that our designed meta-knowledge learners are superior to the vanilla modules in all combinations. Specifically, combining Meta-GAT and Meta-LSTM achieves the highest Hits@100 on both datasets, demonstrating that our motivations are practical for learning the diverse user social dependencies and complex temporal influences in the process of information diffusion.

5.6.3. Hyperparameter sensitivity

We now investigate the impacts of three important hyperparameters – dimensions of user embedding U , meta knowledge vectors v and d from $\{16, 32, 64, 128, 256\}$ – in Fig. 6. Regarding the dimension of user embedding U , we made the following observations: Initially, on the all datasets, the model performance of MetaCas increases as the dimension increases up to 64. However, on the Memes and Douban datasets, the rate of improvement gradually decreases thereafter. In contrast, on the Twitter and Android datasets, the model performance starts to slightly decrease as the dimension increases beyond 64. To ensure a balance between model efficiency and performance across all datasets, we identify that a dimension of 64 for U is appropriate. As can be seen from the results, the appropriate dimensions for v and d are 128 and 64, respectively. The performance of MetaCas first increases as the dimension of meta-knowledge learner increases and then starts decreasing when the dimension is too large. Specifically, MetaCas employs cascade-specific meta-knowledge to effectively adjust model parameters of topological-temporal modules. The structural meta-knowledge vector v is used to adapt the process of user social dependency in the Meta-GAT. Similarly, the meta-knowledge vectors d focuses on incorporating temporal influence information in the Meta-LSTM. Meta-LSTM effectively captures the temporal influence, which is crucial for reflecting the dynamic nature of the actual information diffusion process. Conversely, the modeling of social dependencies serves the purpose of providing auxiliary signals to enhance prediction. Notable, the dimension of user representation is set to 64 in MetaCas. Therefore, the dimension of the meta-knowledge vector d should not be excessively large to avoid interfering with the learning of primary tasks.

Table 8

Model performance comparison on three datasets from different research areas. Higher values of Hits and MAP indicate better performance. We run each model on each dataset five times and report the mean and standard deviation. The best results are in **bold font**.

Model	Twitter15		Weibo22		Christianity	
	Hits	MAP	Hits	MAP	Hits	MAP
NDM	9.38 _{±0.01}	0.61 _{±0.01}	5.96 _{±0.10}	3.11 _{±0.03}	45.86 _{±0.25}	7.86 _{±0.13}
FOREST	18.76 _{±0.22}	3.41 _{±0.10}	16.62 _{±1.35}	8.63 _{±0.30}	60.56 _{±0.15}	14.49 _{±0.12}
DyHGAT	20.64 _{±0.08}	4.93 _{±0.04}	31.83 _{±0.57}	8.27 _{±0.07}	60.47 _{±0.11}	14.32 _{±0.13}
MS-HGAT	23.54 _{±0.05}	5.83 _{±0.04}	39.52 _{±0.15}	7.26 _{±0.16}	57.03 _{±0.14}	18.36 _{±0.13}
MetaCas	27.49 _{±0.01}	6.11 _{±0.01}	56.06 _{±0.04}	8.71 _{±0.04}	61.24 _{±0.01}	19.38 _{±0.01}

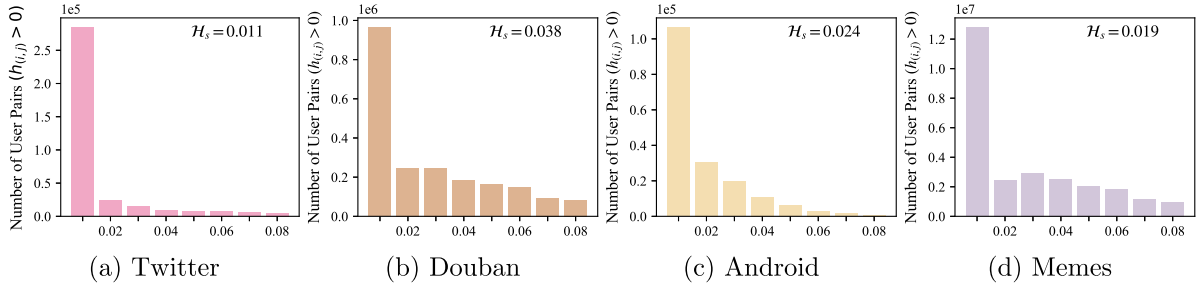


Fig. 8. Preference-aware homophily ratio distributions of social network on four real-world datasets, where $h(i, j)$ is the edge-wise homophily ratio of user-user edge (u_i, u_j) , H_s is the graph-wise homophily ratio of social network.

5.7. Model efficiency analysis

To evaluate the model computational complexity, we conduct an analysis comparing our proposed MetaCas with four competitive baselines (NDM, FOREST, DyHGAT, and MS-HGAT) on Twitter and Memes datasets. We included two important measures, i.e., the inference time cost and maximum GPU occupation. The complexity analyzing results are summarized in Fig. 7. The fixed batch size is set to 64 for all models. Specifically, we can see that MetaCas exhibits the shortest inference time in comparison to all baselines across two distinct datasets. This advantage can be explained by the fact that MetaCas employs a simple and efficient combination of GAT and LSTM networks through local meta-knowledge injection. In contrast, baselines design complex social graph learning structure (e.g., dynamic graph and sequential hypergraph) and temporal dependency modeling architectures (e.g., transformer-based mechanisms), resulting in significant inference time overhead. Furthermore, all baselines typically exhibit higher space requirements compared to MetaCas, except for DyHGAT and FOREST on the Meme dataset. This indicates that the high efficiency exhibited by FOREST and DyHGAT is limited to scenarios where cascade lengths are short (e.g., the Meme dataset). Hence, it is evident that MetaCas achieves a balance between model performance and efficiency – relatively smaller GPU memory occupation and inference time, but higher prediction performance. This achievement can be attributed to the effectiveness of our cascade-related meta-knowledge learner design.

5.8. Case study

Model Generalizability. To better analyze our model’s generalizability under situations of diverse information diffusion patterns, we conduct additional experiments on three new datasets: Twitter15 [44], Weibo22¹ and Christianity [39]. The Twitter15 dataset is obtained from the Twitter/X platform and is often used in the field of misinformation diffusion. Weibo22 dataset is obtained from the Weibo platform and is used for predicting social content popularity. The Christianity dataset is collected from the Christian-themed section of the Stack Exchange website, which is commonly utilized for the task of information diffusion prediction. We compare MetaCas with four strong baselines (NDM, FOREST, DyHGAT, and MS-HGAT) on these new datasets. The experimental results are shown in Table 8. Notably, MetaCas outperforms all baselines on three datasets in terms of both Hits and MAP. This observation signifies the generalizability of MetaCas across various social platforms and diffusion patterns. The improved performance can be attributed to the design of extracting cascade-specific meta-knowledge, which enables the adaptive adjustment of key model parameters in MetaCas for effectively capturing diverse user temporal dependencies during the information diffusion process.

Homophily Analysis. To analyze the affect of social homophily on the model performance, we conduct an experiment to evaluate the user preferences inherent in the social graph. Specifically, we calculate the preference-aware homophily ratios (see definitions in Section 2) across four real-world cascade datasets. As shown in Fig. 8, our observations reveal that the distributions of edge-wise

¹ <http://data.skccc.com/2022/>.

Time:	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
Current user:	5634	10714	2840	11893	1674	1832	11444	9703	10456	9494	7247	8147	1546	8206	137
Next user:	10714	2840	11893	1674	1832	11444	9703	10456	9494	7247	8174	1546	8206	137	11186
MetaCas Top-10 User Predictions	10714	11893	11893	5953	1546	8431	9703	10259	8174	12037	1546	1546	137	137	11186
	11893	12063	1546	8431	1	9703	10259	1546	4353	10259	10259	10283	7099	10283	8697
	9949	1	12051	9703	8431	145	11520	10026	10026	4352	8406	10259	11520	11186	11204
	1569	2840	393	1569	6596	12189	1	2744	10283	9498	2437	10026	10283	5734	5734
	1	12051	1	6595	1569	5326	10283	12037	10259	1546	1	2744	10386	11520	4712
	2945	9949	9703	7462	2353	1	1546	1898	94	7557	12037	12037	5604	10386	2610
	5953	393	6332	3373	9949	1546	1898	10456	1546	10026	1898	8406	8347	12063	10622
	10386	1898	9043	2945	10259	11482	8406	4352	12037	10283	9498	5734	8375	2429	2683
	12063	6332	8431	6452	7462	11017	9484	8406	2744	7510	7557	4352	4870	1898	2477
	2840	12354	1569	1	4164	7023	11482	10283	7557	3032	9484	4870	12063	10622	1898
FOREST Top-10 User Predictions	12505	12505	12505	12505	12505	12505	2744	2744	8174	1	4712	5210	12505	12505	11186
	9316	9316	11169	10023	11261	10023	5210	2496	9332	4712	10953	12505	137	137	11169
	7692	11261	7692	11169	5320	7966	12505	12505	10317	6500	1	9846	3875	11261	12505
	3389	8895	9316	7966	11169	867	5133	5210	5210	8406	4433	3398	5727	11169	8607
	851	2986	5320	2945	867	11169	3398	3398	3373	5936	6500	10023	8347	8607	11261
	8895	5320	11261	6053	6053	3398	9846	6500	9269	7557	2610	2496	11261	11186	5320
	6805	7692	851	11261	2986	6053	10023	4352	2744	10437	5936	2744	9021	5320	7966
	5936	12451	7966	867	3875	2945	10259	8406	12156	4164	4164	3875	4164	9021	2986
	11169	2477	10023	2804	12066	11261	11169	10259	10023	7378	7557	11522	11169	2945	2945
	2477	12010	867	7692	4178	10308	7966	1	10308	12505	7358	4164	8607	5727	867
MS-HGAT Top-10 User Predictions	11173	4342	7509	3012	9150	1880	5031	3012	9494	4599	3761	8507	2566	3364	11186
	10932	11963	10932	12177	137	8645	3134	11816	7535	10646	4566	4112	6777	137	3517
	9032	3700	10127	4842	2786	7069	674	11451	1880	4351	26	11292	312	7759	3617
	3700	1513	3364	3385	6132	3775	9658	7980	2378	3700	12217	7993	7179	5031	341
	2726	1	9658	174	7019	180	1546	10456	7798	1	674	6816	1695	177	1939
	4759	3364	3517	4119	12078	3235	10127	7019	7890	11362	1546	4347	4318	4097	11292
	174	4119	12078	8282	3364	12352	4119	1269	1546	3012	7972	1013	1	6777	4112
	3775	3775	213	1	12352	674	1646	12352	2497	10225	1025	739	341	10220	1488
	7248	7713	1860	11173	4701	137	1860	3848	4199	10127	6840	5012	11186	962	161

Fig. 9. A case study of information diffusion prediction, showing the first 15 activated users in the underlying information cascade from Twitter dataset (numbers in the boxes are user IDs). For each current user, we use MetaCas, FOREST [6] and MS-HGAT [17] to predict the top ten users with highest activation likelihoods. Correct predictions are marked with red background. If the predicted user is one of the 15 activated users, we mark it with light red background.

homophily ratios in these datasets approximately follow the power-law distribution. Moreover, the majority of user-user edges exhibit edge-wise homophily ratios that are in close proximity to 0. In addition, we calculate the graph-wise homophily ratio \mathcal{H}_s for each dataset. We find that users connected in the social graph exhibit diverse preferences and the social network is low homophilic (ratios around $\mathcal{H}_s = 0.1$ or smaller). These observations verify our motivation of modeling diverse user social dependency. Moreover, our MetaCas model achieves average improvements of 10.89%, 5.60%, 4.71%, and 5.32% over the most competitive baseline on the Twitter, Douban, Android, and Memes datasets, respectively. This indicates that our MetaCas model exhibits strong scalability when dealing with social graphs characterized by a low homophily rate. We attribute this to the fact that MetaCas, equipped with a structure meta-knowledge learner, is capable of capturing user preferences to compensate for the limitations of social networks with low homophily.

Prediction Visualization. To provide further insight into the behavior of MetaCas, we provide a visualization of a case study showing a specific Twitter tweet with 15 activated users in Fig. 9. For each current user, we use MetaCas, FOREST [6] and MS-HGAT [17] to predict the top ten users with the highest activation likelihoods. Similar to MetaCas, FOREST learns social graph information via a structural context extraction module and utilizes gated recurrent unit for user sequential modeling. MS-HGAT is a competitive baseline for information diffusion prediction by utilizing sequential hypergraphs. As Fig. 9 shows, MetaCas successfully predicted 8 out of 15 activated users, while FOREST and MS-HGAT only has 2 and 4 correct predictions, respectively. In addition, if the predicted user belongs to one of the remaining activated users, we mark it with light red background. We can see that MetaCas predicts 17 times the users that finally got activated in the cascade, while FOREST and MS-HGAT only predicts 3 and 6 times, respectively. This experiment suggests that MetaCas can predict considerably more potential users in advance, which is beneficial for online social recommendation systems. Overall, these results further demonstrate the superiority of MetaCas for learning topological-temporal correlations (e.g., user social dependencies and preferences) over its counterparts.

6. Conclusion

We took a first step toward exploiting topological-temporal meta-knowledge from information cascades and presented MetaCas, a novel diffusion prediction framework to model the correlations in diverse social dependencies and user preferences along the temporal evolution. We constructed two types of meta-knowledge learners – SMK-Unit and DMK-Unit – which build an adaptive correlation between cascades and cascade attributes and transfer the learned knowledge to downstream Meta-GAT and Meta-LSTM modules via dynamic parameter generation. We evaluated MetaCas over four real-world information diffusion datasets and compared it with state-of-the-art baselines, and our experiments demonstrated a very competitive performance and showed that meta-knowledge learning is beneficial for predicting the information diffusion between users. As for future work, several aspects of our model warrant further

investigations, e.g., incorporating meta-gradient methods to develop a generalized meta-knowledge learning model for different information cascade prediction tasks.

CRedit authorship contribution statement

Zhangtao Cheng: Writing – review & editing, Writing – original draft, Visualization, Validation. **Jienan Zhang:** Software, Resources, Methodology. **Xovee Xu:** Writing – review & editing, Methodology. **Wenxin Tai:** Formal analysis, Data curation. **Fan Zhou:** Funding acquisition, Formal analysis, Data curation. **Goce Trajcevski:** Validation, Supervision, Methodology, Investigation. **Ting Zhong:** Project administration, Investigation, Funding acquisition.

Funding

This work was supported by National Natural Science Foundation of China (Grant No. 62176043, No. 62072077, and No. U22A2097), Kashgar Science and Technology Bureau (Grant No. KS2023025), and National Science Foundation SWIFT (Grant No. 2030249).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data and source code are provided (see link in the paper).

References

- [1] F. Zhou, X. Xu, G. Trajcevski, K. Zhang, A survey of information cascade analysis: models, predictions, and recent advances, *ACM Comput. Surv.* (2021) 1–36, <https://doi.org/10.1145/3433000>.
- [2] F. Zhou, T. Qian, Y. Mo, Z. Cheng, C. Xiao, J. Wu, G. Trajcevski, Uncertainty-aware heterogeneous representation learning in poi recommender systems, *IEEE Trans. Syst. Man Cybern. Syst.* (2023) 4522–4535, <https://doi.org/10.1109/TSMC.2023.3252079>.
- [3] Z. Hu, S. Nakagawa, Y. Zhuang, J. Deng, S. Cai, T. Zhou, F. Ren, Hierarchical denoising for robust social recommendation, *IEEE Trans. Knowl. Data Eng.* (2024), <https://doi.org/10.1109/TKDE.2024.3508778>.
- [4] T. Zhong, J. Lang, Y. Zhang, Z. Cheng, K. Zhang, F. Zhou, Predicting micro-video popularity via multi-modal retrieval augmentation, in: *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 2579–2583.
- [5] Y. Wang, H. Shen, S. Liu, J. Gao, X. Cheng, Cascade dynamics modeling with attention-based recurrent neural network, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2985–2991.
- [6] C. Yang, J. Tang, M. Sun, G. Cui, Z. Liu, Multi-scale information diffusion prediction with reinforced recurrent networks, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 4033–4039.
- [7] S. Lamprier, A recurrent neural cascade-based model for continuous-time diffusion, in: *Proceedings of the International Conference on Machine Learning (ICML)*, 2019, pp. 3632–3641.
- [8] Z. Wang, W. Li, Hierarchical diffusion attention network, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 3828–3834.
- [9] M. McPherson, L. Smith-Lovin, J.M. Cook, Birds of a feather: homophily in social networks, *Annu. Rev. Sociol.* (2001) 415–444, <https://doi.org/10.1145/2675133.2675179>.
- [10] V. Indu, S.M. Thampi, A nature-inspired approach based on forest fire model for modeling rumor propagation in social networks, *J. Netw. Comput. Appl.* (2019) 28–41, <https://doi.org/10.1016/j.jnca.2018.10.003>.
- [11] X. Miao, H. Peng, K. Chen, Y. Peng, Y. Gao, J. Yin, Maximizing time-aware welfare for mixed items, in: *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2022, pp. 1044–1057.
- [12] J. Cheng, L. Adamic, P.A. Dow, J.M. Kleinberg, J. Leskovec, Can cascades be predicted?, in: *Proceedings of the International World Wide Web Conference (WWW)*, 2014, pp. 925–936.
- [13] M.R. Islam, S. Muthiah, B. Adhikari, B.A. Prakash, N. Ramakrishnan, Deepdiffuse: predicting the ‘who’ and ‘when’ in cascades, in: *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 1055–1060.
- [14] Z. Wang, C. Chen, W. Li, A sequential neural information diffusion model with structure attention, in: *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, 2018, pp. 1795–1798.
- [15] C. Yuan, J. Li, W. Zhou, Y. Lu, X. Zhang, S. Hu, Dyhgen: a dynamic heterogeneous graph convolutional network to learn users’ dynamic preferences for information diffusion prediction, in: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (PKDD)*, 2020, pp. 347–363.
- [16] Z. Cheng, W. Ye, L. Liu, W. Tai, F. Zhou, Enhancing information diffusion prediction with self-supervised disentangled user and cascade representations, in: *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2023, pp. 3808–3812.
- [17] L. Sun, Y. Rao, X. Zhang, Y. Lan, S. Yu, Ms-hgat: memory-enhanced sequential hypergraph attention network for information diffusion prediction, in: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022, pp. 4156–4164.
- [18] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Netw. Learn. Syst.* (2020) 4–24, <https://doi.org/10.1109/TNNLS.2020.2978386>.
- [19] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.
- [20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: *Proceedings of the International Conference on Learning Representations*, 2018.
- [21] T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey, Meta-learning in neural networks: a survey, *IEEE Trans. Pattern Anal. Mach. Intell.* (2021) 5149–5169, <https://doi.org/10.1109/TPAMI.2021.3079209>.

- [22] J. Schmidhuber, Learning to control fast-weight memories: an alternative to dynamic recurrent networks, *Neural Comput.* (1992) 131–139, <https://doi.org/10.1162/neco.1992.4.1.131>.
- [23] D. Ha, A. Dai, Q.V. Le, Hypernetworks, in: *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.
- [24] H. Le, T. Tran, S. Venkatesh, Neural stored-program memory, in: *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.
- [25] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: *Proceedings of the International Conference on Machine Learning (ICML)*, 2017, pp. 1126–1135.
- [26] P. Jaccard, The distribution of the flora in the Alpine zone. 1, *New Phytol.* (1912) 37–50.
- [27] A. Grover, J. Leskovec, node2vec: scalable feature learning for networks, in: *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2016, pp. 855–864.
- [28] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2014, pp. 701–710.
- [29] J. Zhang, Y. Dong, Y. Wang, J. Tang, M. Ding, Prone: fast and scalable network representation learning, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 4278–4284.
- [30] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: *Proceedings of the International Conference on Learning Representations (Workshop Track)*, 2013.
- [31] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, *Adv. Neural Inf. Process. Syst.* (2013) 3111–3119.
- [32] S. Wang, L. Hu, Y. Wang, X. He, Q.Z. Sheng, M.A. Orgun, L. Cao, F. Ricci, P.S. Yu, Graph learning based recommender systems: a review, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2021, pp. 4644–4652.
- [33] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.
- [34] C. Yang, M. Sun, H. Liu, S. Han, Z. Liu, H. Luan, Neural diffusion model for microscopic cascade study, *IEEE Trans. Knowl. Data Eng.* (2019) 1128–1139, <https://doi.org/10.1109/TKDE.2019.2939796>.
- [35] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, K. Achan, Self-attention with functional time representation learning, in: *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [36] T.N. Kipf, M. Welling, Variational graph auto-encoders, *CoRR*, arXiv:1611.07308, 2016.
- [37] N.O. Hodas, K. Lerman, The simple rules of social contagion, *Sci. Rep.* (2014) 1–7, <https://doi.org/10.1038/srep04343>.
- [38] E. Zhong, W. Fan, J. Wang, L. Xiao, Y. Li, Comsoc: adaptive transfer of user behaviors over composite social network, in: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2012, pp. 696–704.
- [39] A. Sankar, X. Zhang, A. Krishnan, J. Han, Inf-vae: a variational autoencoder framework to integrate homophily and influence in diffusion prediction, in: *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, 2020, pp. 510–518.
- [40] J. Wang, V.W. Zheng, Z. Liu, K.C.-C. Chang, Topological recurrent neural network for diffusion prediction, in: *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2017.
- [41] S. Bourigault, S. Lamprier, P. Gallinari, Representation learning for information diffusion through social networks: an embedded cascade model, in: *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2016, pp. 573–582.
- [42] S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, Y.M. Chee, Inf2vec: latent representation model for social influence embedding, in: *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2018, pp. 941–952.
- [43] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: large-scale information network embedding, in: *Proceedings of the International Conference on World Wide Web (WWW)*, 2015, pp. 1067–1077.
- [44] J. Ma, W. Gao, K.-F. Wong, Detect rumors in microblog posts using propagation structure via kernel learning, in: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 708–717.