CasFlow: Exploring Hierarchical Structures and Propagation Uncertainty for Cascade Prediction

Xovee Xu, *Graduate Student Member, IEEE*, Fan Zhou, Kunpeng Zhang,
Siyuan Liu, and Goce Trajcevski

Abstract—Understanding in-network information diffusion is a fundamental problem in many applications and one of the primary challenges is to predict the information cascade size. Most of the existing models rely either on hypothesized point process (e.g., Poisson and Hawkes processes), or simply predict the information propagation via deep neural networks. However, they fail to simultaneously capture the underlying global and local structures of a cascade and the propagation uncertainty in the diffusion, which may result in unsatisfactory prediction performance. To address these, in this work we propose a novel probabilistic cascade prediction framework *CasFlow*: Hierarchical Cascade Normalizing Flows. CasFlow allows a non-linear information diffusion inference and models the information diffusion process by learning the latent representation of both the structural and temporal information. It is a pattern-agnostic model leveraging normalizing flows to learn the node-level and cascade-level latent factors in an unsupervised manner. In addition, CasFlow is capable of capturing both the cascade representation uncertainty and node infection uncertainty, while enabling hierarchical pattern learning of information diffusion. Extensive experiments conducted on real-world datasets demonstrate that CasFlow reduces the prediction error to 21.0% by only observing half an hour of cascades, compared to state-of-the-art approaches, while also enabling model interpretability.

Index Terms—Information diffusion, information cascade, popularity prediction, social networks, uncertainty, graph learning

1 Introduction

Online social platforms such as Twitter, Weibo, Facebook, YouTube, and Reddit have become the main source of information to guide individuals' everyday decisions. Various news, events, posts, and videos are disseminated as cascades spread by users through social networks [1], [2]. Such Internet technology and social media facilitate free information (both true and false) creation and sharing. Understanding information cascades becomes important and can lead to significant economical and societal impacts, among which predicting the size of (potentially) affected users after a certain time-period is one of typical tasks and has attracted great attention in both academia and industry. It plays a critical role and is involved in many down-stream applications – from rumor detection, through epidemic spread identification and improved recommendation, to accelerating or suppressing information propagation [3], [4]. For example, in the global effort to contain the COVID-19 pandemic, misinformation abounds and flourishes on the Internet, and people have been led to believe that COVID-19 can be cured by ingesting fish tank cleaning products or that 5G networks generate radiation that triggers the virus. Such misinformation not only causes panic among citizens but could potentially undercut collective efforts to control the pandemic. Precisely predicting the cascade as earlier as possible can help social platforms prevent spreading fake news [5], relieve anxiety [6], as well as benefiting individuals.

In recent years, a series of works have been focusing on this area [7], including pattern recognition of information diffusion and popularity prediction of items over social networks and, in a broad sense, they can be summarized into the following categories:

(1) *Feature engineering-based* approaches: Researchers in [8], [9] focus on identifying and incorporating handcrafted features for cascade prediction. These models require extensive domain knowledge and thus are hard to be generalized to new domains. In addition, many features such as user profile and personalized social information are usually inaccessible in practical scenarios due to some privacy concerns.

(2) *Statistical* approaches: In [10], [11], researchers model the intensity function of the arrival for incoming messages to study the propagation process. These methods are mathematically solid and have demonstrated enhanced interpretability, but they require long observation dependency and

- *Xovee Xu and Fan Zhou are with the University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China. E-mail: xovee@ieee.org, fan.zhou@uestc.edu.cn.*
- *Kunpeng Zhang is with the Department of Decision, Operations and Information Technologies, University of Maryland, College Park, MD 20742 USA. E-mail: kpzhang@umd.edu.*
- *Siyuan Liu is with the Department of Supply Chain & Information Systems, Pennsylvania State University, State College, PA 16802 USA. E-mail: siyuan@psu.edu.*
- *Goce Trajcevski is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA. E-mail: gocet25@iastate.edu.*

Manuscript received 25 May 2020; revised 3 October 2020; accepted 29 October 2021. Date of publication 10 November 2021; date of current version 7 March 2023.
This work was supported in part by the National Natural Science Foundation of China under Grants 62072077 and 62176043, and in part by National Science Foundation SWIFT under Grant 2030249.
(Corresponding author: Fan Zhou.)
Recommended for acceptance by Y. Zhang.
Digital Object Identifier no. 10.1109/TKDE.2021.3126475

are still unable to fully leverage the information encoded in the cascade for a satisfactory prediction.

(3) *Deep learning-based* approaches: Recent advance in deep learning has achieved great successes for many applications. In [12], [13], researchers leverage various deep learning techniques and develop models for capturing the temporal and sequential processes of information diffusion, where recurrent neural networks (RNN) such as LSTM [14], GRU [15], and graph neural networks (GNN) [16] are usually used for modeling the sequential patterns [17] and graph structures [18], [19], respectively. However, most existing approaches suffer from the inefficiency of node and graph representation and fail to consider the uncertainty in both node embedding and information diffusion.

Notwithstanding the improvements on cascades modeling, existing methods confront several key challenges:

(1) *Efficient cascade representation* is difficult due to the varying size (from very few to millions), which makes many graph embedding-based models biased and inapplicable (especially the random walk related ones).

(2) *Modeling both local and global structures* in context of popularity prediction is often absent or incomplete, and embedding a complete social graph with millions of nodes is computationally expensive or even impossible.

(3) *Modeling structural and temporal characteristics of information diffusion* – initial spreading is crucial for accurately predicting the size of diffusion, however, it usually lacks sufficient structural information in practice. Capturing underlying structural patterns from the limited information becomes a key to make prediction effective. In addition, the temporal information, e.g., the order of spreading among participants, the spread speed, etc., are also vital in cascade prediction.

(4) *Lack of hierarchical cascade modeling at different levels* – it makes the existing methods either focus on roughly estimating the diffusion size according to few observations, or study user-level modeling (i.e., activation of individual users) without consistently investigating the correlation between node-level (lower) and cascade-level (higher) representations.

(5) *Absence of cascade uncertainty handling* – understanding uncertainty involved in a cascade is important for the formulation of the cascade's information diffusion process (e.g., the observed sharing/retweeting innately introduces noises and uncertainties for the future cascade [9]) – which is not taken into account in the existing methods.

Our Approach. To address the aforementioned challenges, we present *CasFlow* (Hierarchical Cascade Normalizing Flows) graph learning neural networks – a novel framework integrating the hierarchical diffusion modeling both on the global and local information propagation, as well as temporal characteristics of cascades for predicting the popularity of an information item (e.g., a post or a paper). Specifically, CasFlow addresses the existing challenges by: (1) implementing graph wavelets to learn the local cascade representation which, in turn, allows varying-size diffusion graph learning; (2) then it employs sparse matrix factorization to learn global user representation which can efficiently model user behavior and interactions in a social network; (3) it further develops a novel contextualized diffusion embedding module to learn complicated users' sharing behavior which captures different behavior of a particular user in different cascades, in addition to the structural and temporal characteristics of information diffusion; (4) to understand both user-level behavior and cascade-level diffusion effect, CasFlow introduces a hierarchical variational autoencoder for simultaneously learning fine-grained and structural patterns of information diffusion with probabilistic latent variables; (5) by incorporating amortized variational inference and normalizing flows into the generative model with latent variables, CasFlow exposes an interpretable and flexible representation of the complex distribution and long-term cascading dependencies among nodes in a cascade, thereby incorporating the uncertainty of each node behavior and the possibility of cascade size growth.

Our main contributions can be summarized as follows:

- *Hierarchical cascade representation*: We propose a novel hierarchical information cascade learning framework which allows dynamic global and local graph embedding and jointly models cascades from both a micro (user) and a macro (overall cascade estimating) level.
- *Diffusion uncertainty modeling*: *CasFlow* leverages variational autoencoders and normalizing flows for embedding both node- and cascade-level representations as flexible posterior distributions, which models the probabilities of sharing behavior among nodes and preserves the uncertainty of information diffusion and cascade growth.
- *Contextualized user behavior learning*: By introducing a Bi-directional RNN-based module into cascade graph learning, *CasFlow* is able to capture users' different behavior on different information, rather than binary prediction on users' retweeting/citing behaviors. This enables integration of the structural and temporal information associated with the information diffusion, while considering contextualized user behavior.
- *Extensive experimental evaluation*: We conduct experiments on several large-scale real-world datasets, demonstrating that *CasFlow* improves the prediction performance compared to the cutting-edge approaches, and we also provide explanations on its behavior. Source code of CasFlow is publicly available at https://github.com/Xovee/casflow.

We note that this paper is an extension of VaCas [19] presented at IEEE INFOCOM 2020. The reminder of the paper is organized as follows. We give a detailed literature review in the next section. We then introduce the preliminary background. We present details of our cascade popularity prediction approach *CasFlow*. In the experiment section, we evaluate our proposed method using three publicly available datasets. Lastly, we conclude our work and point out potential future directions.

2 Related Work

We now review the related literature grouped in three main categories and position our work in that context by indicating the respective issues (that are addressed by our contributions).

Feature-based approaches [20], [21] study the factors affecting content popularity, including content-related features such as the number of hashtags or mentions [22] and user-related features such as user profiles, user attributes and historical activities [23], [24], cascade texts and images [25], [26], temporal [27] and cascade's structural [28], [29]. The popularity is predicted via various machine learning models. Among various studies [30], [31], they confirmed that user features are informative predictors, especially the features related to early-forwarding users [32]. Feature-based approaches are not easy to generalize, since feature extraction heavily depends on domain knowledge and is usually specific to data types, not to mention the non-existence of systematic way to guide such a process.

Statistical approaches such as [33], [34] exploit the patterns in the sequence of retweeting/citing (a.k.a. event) time and model their arrival process in a generative way. Generally, the cascade is treated as time-series data, and the model – i.e., parameter estimation – is built by maximizing the probability of an event occurring within an observation time window. Different point processes (e.g., Poisson [35], [36], Hawkes process [37], [38]), and models (e.g., Cox [39], [40], Weibull [41], survival analysis [41], and epidemic model [34]) have been developed. Despite demonstrating an enhanced forecasting accuracy and explainable prediction, the methods are unable to fully leverage the implicit information in the cascade dynamics. As shown in a recent review [42]: Poisson process is too simple to capture the propagation patterns, Weibull and Hawkes models tend to overestimate the cascade popularity, probably due to its rudimentary self-excitation mechanism. In contrast, CasFlow enables integration of structural and temporal information in a diffusion process.

Deep learning-based approaches are inspired by the recent advances of deep neural networks in many fields, and have achieved significant performance improvements in many applications, including the popularity prediction of information cascades [7]. One of the pioneers – DeepCas [17] – is a structure-based popularity prediction model learning the representation of cascade graphs in an end-to-end manner. Subsequently, DeepHawkes [12] transformed the cascade graph into a set of diffusion paths according to the diffusion time, each of which depicts the process of information propagation between users within the observation time. There are several similar works, proposed to improve the deep learning-based cascade prediction: DTCN [43], UHAN [44], Topo-LSTM [45], FOREST [46], and DFTC [47] – all of which intend to extract full paths of diffusion from sequential observations of information infections. They leverage RNNs and attention mechanism to model the information growth and predict the diffusion size. However, unlike CasFlow, these approaches usually focus on a simple graph for cascade representation learning, which cannot fully capture the dynamics of graphs.

Motivated by graph neural networks (GNNs) [16], a recurrent cascade convolution model CasCN was developed in [18]. It learns the structure of each cascade by a dynamic graph convolutional network (GCN) and takes into account the directionality of cascades, and time decay effects for cascade prediction. The subsequent work [48] models the information cascade using multi-task learning by simultaneously predicting the information popularity at the macro-level and the user participation in re-posting at the micro-level. A recent work [49] learns the cascading effect in information diffusion by exploiting coupled GNNs to capture the interpersonal influence and individual user adoption, respectively. However, these approaches rely on deterministic inference process, which limits their ability to produce relevant states by sampling from the posterior of cascades. Therefore, how to incorporate the uncertainty of information diffusion remains as one of the unaddressed issues in existing methods.

TABLE 1
Mathematical Notations Used in This Paper

Symbol	Description
$\mathbf{A}, \mathbf{D}$	Adjacency matrix and diagonal degree matrix.
$C_k(t)$	Information cascade observed at time t.
d	Dimensionality of embedding.
$E_c(u_i)$	Embedding of node u_i in cascade graph.
$E_g(u_i)$	Embedding of node u_i in global graph.
$\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$	Cascade graph with sets of nodes and edges.
$\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$	Global graph with sets of nodes and edges.
$\mathbf{h}_1, \mathbf{h}_2$	Vectors from two-layers of Bi-GRU.
I	Information item, e.g., a tweet or a paper.
K	Number of transformations of NFs.
M	Number of users in cascade.
N	Number of cascades.
$P_k(t)$	Popularity of cascade at time t.
$\mathcal{R} = \{\mathcal{R}_i\}_{i\in\lvert\mathcal{V}_c\rvert}$	Input sequences of VAEs.
t_o, t_p	Observation/prediction time.
u_i	User in cascade.
$\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3$	Latent representations from node level VAEs, cascade level VAE, and from NFs, respectively.

3 Preliminaries

We now give necessary background and formally define the popularity prediction problem of information cascades. We list notations used throughout the paper in Table 1.

Let C_k denote an event of interest which, starting at some time-instant, is propagated through a network. In the rest of this study, we consider tweet cascades as example-settings, however, our work can be directly applicable to other types of information diffusion (e.g., academic publications, news articles, online forums, video and streaming media, etc.). Consider a Twitter user u post a tweet I at time t_0, later, other users can interact with this tweet, e.g., "commenting", "liking", and "retweeting". In this paper, we consider "retweeting" as a major source of information dissemination in the Twitter social network. Given an observation time t_o, we assume there are totally M involved users who retweet this tweet I, in consequence a retweet cascade $C_k(t_o) = \{(v_i, u_i, t_i)\}_{i\in M}$, where each 3-tuple represents user u_i retweet user v_i's tweet at time $t_i \leq t_o$.

Existing efforts make different cascade predictions in a rather similar way. Some works [50], [51] treat cascade prediction as a classification problem – e.g., predicting whether a cascade can break out a certain threshold [23], [52]; whether a cascade can double its size [9] at the end; or predict the range that a cascade would mostly like to fall into [47], [53]. Similar to many previous works [12], [18], we have our information cascade popularity prediction problem defined as:

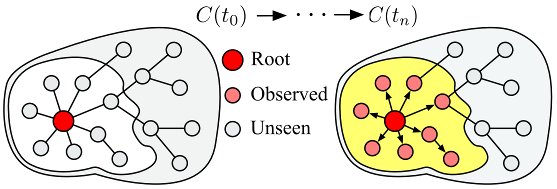


Fig. 1. Illustration of evolving cascade graph $\mathcal{G}_c$ for a specific cascade.

Definition 1. Cascade Popularity Prediction – *Given observed snapshot of cascade $C_k(t_o)$ at time t_o, we aim to predict the future size $P_k(t_p) = |C_k(t_p)|$ (a.k.a. popularity) of this cascade at a prediction time $t_p \gg t_o$, i.e., the number of users who perform the retweeting action to the original tweet after t_o.*

In this formulation setting, rather than observing a fixed number of retweets [9], we peek into cascade's early stage behavior for a fixed time frame, which is a more flexible and realistic task in real-world applications. Overall, recall that for N observed cascades (e.g., N tweets) $\{C_k(t_o)\}_{1\leq k\leq N}$, the popularity prediction can be formalized as a regression problem solved by minimizing the following loss function:

$$\begin{aligned}\mathcal{L}(\Theta) &= \frac{1}{N}\sum_{k=1}^{N}(\hat{P}_k(t_p) - P_k(t_p))^2,\\ \hat{P}_k(t_p) &= \underset{\Theta}{\mathrm{Model}}(C_k(t_o)),\end{aligned} \tag{1}$$

where $P_k(t_p) = |C_k(t_p)|$ is ground truth popularity for cascade C_k at prediction time t_p, Θ are model parameters.

This definition is a vanilla version of cascade popularity prediction. In practical situations, various additional factors may influence the final popularity of cascades, e.g., previous work found that textual and semantic features (length, topics, and sentiments) may have an impact on the future popularity [54], [55]; image latent features extracted from neural network learning [25]; social network features that quantify the influence of individuals, e.g., the follower/followee network of Twitter users [17], [23], scientific collaboration network [1], and cascade spreading networks [9], [12].

In this paper, we mainly model and capture two important graph-based influencing factors – cascade graph and underlying user social networks (we call it global graph). Here we first introduce their formal definitions:

Definition 2. Cascade Graph – *Given a tweet I and its corresponding retweet cascade C, a cascade graph is defined as $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$, with nodes $\mathcal{V}_c = \{u_i | 1 < i \leq M\}$ being the set of all re/tweeting users, and $\mathcal{E}_c \subseteq \mathcal{V}_c \times \mathcal{V}_c$ is a set of $M = |C|$ edges representing all relationships between users in this cascade (e.g., retweeting in this case). An example of cascade graph evolved with time (from t_0 to t_n) is illustrated in Fig. 1.*

Definition 3. Global Graph – *The global graph $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$ is a collection of nodes and edges. The edge represents a different node relationship from cascading. For example, Twitter follower/followee social network is a typical example of a global graph.*

Here the cascade graph implies the characteristics of information diffusion from a local perspective, while the global graph provides us a special angle to analyze the diffusion among users rather than only consider independent cascade graphs. For example, in Twitter, whom a user decide to follow, or which tweet s/he chooses to retweet, all the historical behavior were reflected in the structure of global graph. Earlier work such as [56], [57] simply use number of followers (i.e., node degrees) as the structural feature of users, which can not sufficiently capture the user influences, presences, and preferences. Other structural features that have been used in feature-based models [58], [59] also make strong assumptions with respect to the underlying diffusion mechanism or suffer from overfitting on particular situation thus can not be generalized to other scenarios where the diffusion processes are different or unknown. There's an urgent need to better represent the graph data of cascade. In next section, we show how to model both the cascade graphs and the global graph to facilitate the prediction of cascade popularity.

4 Methodology

In this section, we describe the details of our proposed model *CasFlow*, which considers both structural (cascade graphs and global graph) and temporal (forwarding time and cascading effect) information to make cascade popularity prediction. As illustrated in Fig. 2, CasFlow consists of four main components:

(A) Structure Learning. It mainly models and captures contextualized structural patterns in cascading graphs

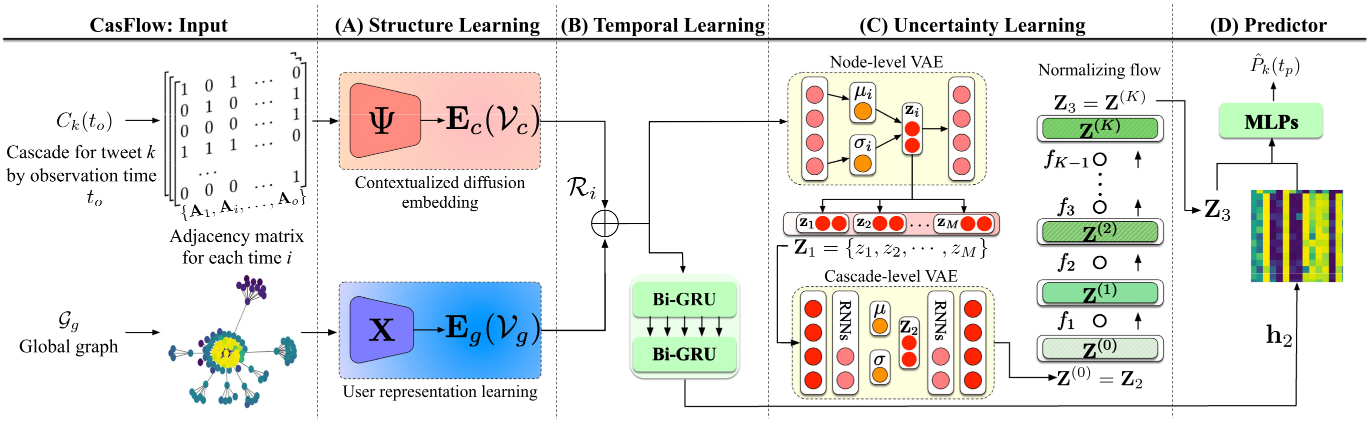


Fig. 2. An overview of our proposed model *CasFlow*. t_0: the first time C_k occurs; t_o: observation time; t_p: the prediction time.

during the diffusion and user latent relationships in their social networks. CasFlow utilizes the techniques from graph signal processing [60], [61] to generate structural embeddings from spectral graph wavelets, and graph representation techniques to learn representation of every individual who participated in cascades in the global graph.

(B) Temporal Diffusion Modeling. CasFlow leverages bidirectional recurrent neural networks to model the temporal dependencies of information diffusion;

(C) Diffusion Uncertainty Modeling. CasFlow models the uncertainty in information diffusion and cascade growth through variational autoencoder and a series of transformations of latent vectors to a more complex and flexible approximated posterior distribution via normalizing flows;

(D) Predictor. Combined with recurrent neural networks and variational inference, the learned cascade representation are fed into multi-layer perceptrons (MLPs) to make the final popularity prediction.

4.1 Structure Learning

4.1.1 Contextualized Cascade Graph Learning

To capture the local structural information and obtain a node-level representation, we employ a graph embedding technique that learns the diffusion of a spectral graph wavelet for each node (cf. [62]). We note that other graph representation techniques may be used, e.g, DeepWalk [63], node2vec [64], etc., depending on different learning targets.

Given a tweet $C \in \{C_1, C_2, \dots C_N\}$ and its observed cascade graph $\mathcal{G}_c(t_o)$ at observation time t_o, its weighted adjacency matrix $\mathbf{A}_c$ can be straightforwardly determined. The diagonal degree matrix $\mathbf{D}_c$ can be computed as each of the diagonal elements is equal to the sum of weights of all edges connected to that node, say u_i. We therefore have an unnormalized graph Laplacian $\mathbf{L}_c = \mathbf{D}_c - \mathbf{A}_c = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is the eigenvalue decomposition and $\mathbf{\Lambda} = \mathrm{Diag}(\lambda_0, \dots, \lambda_{M-1})$ is the diagonal matrix of the eigenvalues satisfying $\lambda_0 < \lambda_1 \leq \dots \leq \lambda_{M-1}$. We can now calculate spectral graph wavelets $\Psi_{u,s}$ for each node $u_i \in \mathcal{V}_c(t_o)$ as

$$\Psi_{u,s} = \mathbf{U}\ \mathrm{Diag}(g_s(\lambda_0), \dots, g_s(\lambda_{M-1}))\mathbf{U}^T\delta_u, \tag{2}$$

where δ_u is the node u's one-hot encoding vector, and the filter kernel function g_s is continuously defined on $\mathbb{R}^+$. Here we use the heat kernel function $g_s(\lambda) = e^{-\lambda s}$ with a scale parameter s on the spectrum $(\lambda_l)_{l=0,\dots,M-1}$.

Graph Laplacian eigenvalues and eigenvectors possess a similar notion to a frequency of a signal, i.e., eigenvectors associated with larger eigenvalues vary fast across the graph and, therefore, these eigenvectors tend to have different values at those locations [61]. In contrast, the eigenvectors associated with smaller eigenvalues carry slowly varying signal across edges, causing the neighboring nodes with high weights to be more likely to have similar values. The heat kernel g_s we employed is directly defined in the graph spectral domain and has a low-pass modulation effect to force a smooth change from high values to low ones.

The basic idea of the node embedding is that the coefficients of the wavelet are directly related to graph topological properties, thereby containing the necessary information to recover structurally similar nodes [62]. For a given node u_i, we treat its wavelet coefficients as a probability distribution and then utilize empirical characteristic functions [65] to represent this distribution. For a scalar random variable X, its characteristic function is defined as $\phi_X(p) = \mathbb{E}[e^{ipX}], p \in \mathbb{R}$. Specifically, for a given node u_i and a scale parameter s, the empirical characteristic function is defined as

$$\phi_{u,s}(p) = \frac{1}{M}\sum_{m=1}^{M} e^{ip\Psi_{m,u,s}}, \tag{3}$$

where $\Psi_{m,u,s} = \sum_{l=0}^{M-1} g_s(\lambda_l)U_{ml}U_{ul}$ is the mth wavelet coefficient of $\Psi_{u,s}$. Then the embedding of node u_i in a cascade graph can be obtained by concatenating values of the real part and imaginary part: $E_c(u_i) = [\mathrm{Re}\ \phi_{u,s}(p), \mathrm{Im}\quad \phi_{u,s}(p)]_{p_1,p_2,\dots,p_d}$ on s. The first element of node's embedding $E_c(u_i)$ is set to node's edge weight (which is normalized by the time user joined in cascade, i.e., $W_u = (t_j - t_o)/t_o \in [0,1], 0 < t_o \leq t_j$) and the dimensionality of the embedding is $d_c = 2d$.

In addition to the node representation, learning the structural information with the wavelets is analogous to the diffusion spreads over the network, and allows us to model the contextualized user behavior – i.e., where we focus is on the individual node embedding rather than embedding the entire graph [64], [66] or emphasizing particular tasks [16]. On the other side, with global graph in hand, we now turn to introduce learning of user representations in the global graph which expresses user connectivity and implies user historical behavior.

4.1.2 Scalable Representation Learning in Large-Scale Global Graph

Different from cascade graph $\mathcal{G}_c$, global graph $\mathcal{G}_g$ usually contains millions of nodes which is hard to model and compute efficiently. Existing graph learning models such as [49], [64] are hard to be utilized directly in practical cascade prediction problems. Here we use sparse matrix factorization [67] to handle large-sized graph representation learning in the advantages of efficiency and scalability.

Given global graph $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$, which is defined as a social networking (e.g., the follower/followee graph), or interaction graph between users (e.g., like/mention/retweet graph), or both, $\mathbf{A}_g$ is the weighted adjacency matrix and $\mathbf{D}_g$ is the diagonal degree matrix of graph $\mathcal{G}_g$. In particular, in order to avoid infeasible computation of factorization for a large-sized matrix, a sparse randomized truncated singular value decomposition (TSVD) was used to learn a distributional similarity-based node embeddings, which guarantees both efficiency and effectiveness [68], [69]. Specifically, according to [67], an entry of a proximity matrix $\mathbf{X}$ can be defined as

$$\mathbf{X}_{i,j} = \begin{cases} \ln p_{i,j} - \ln(\tau Q_{\mathcal{E}_g,j}), & (u_i, u_j) \in \mathcal{E}_g \\ 0, & (u_i, u_j) \notin \mathcal{E}_g \end{cases}, \tag{4}$$

where τ is the negative sample ratio, $p_{i,j}$ is the weight of user-pair (u_i, u_j) in $\mathcal{E}_g$, and $Q_{\mathcal{E}_g,j}$ are the negative samples with node u_j. Then the objective becomes to approximate matrix factorization of $\mathbf{X}$

$$\mathbf{X} \approx \tilde{\mathbf{X}} = \mathbf{U}_{d_g}\mathbf{\Sigma}_{d_g}\mathbf{V}_{d_g}^T, \tag{5}$$

where $\mathbf{U}_{d_g}, \mathbf{V}_{d_g} \in \mathbb{R}^{|\mathcal{V}_g| \times d_g}$ are orthonormal matrices, $\mathbf{\Sigma}_{d_g}$ is a rectangular diagonal matrix with d_g selected non-negative singular values. Since $d_g \ll |\mathcal{V}_g|$ the computation of TSVD is much reduced.

To further speed up the expensive computation for a large-sized graph, we utilize randomized TSVD to approximate matrix $\mathbf{X}$ in two steps: (1) first we try to find $\mathbf{R}$ which has d_g orthonormal columns and let $\mathbf{X} \approx \mathbf{R}\mathbf{R}^T\mathbf{X}$ (for brevity we omit the subscripts); (2) assume we have found that $\mathbf{R}$, then let $\mathbf{B} = \mathbf{R}^T\mathbf{X} \in \mathbb{R}^{d_g \times |\mathcal{V}_g|}$ which is a relatively small matrix that can be computed efficiently by standard SVDs, thus we have $\mathbf{B} = \mathbf{S}\mathbf{\Sigma}\mathbf{V}^T$ where $\mathbf{S}, \mathbf{V}$ are orthogonal and $\mathbf{\Sigma}$ diagonal; finally the approximated matrix is $\mathbf{X} \approx \mathbf{R}\mathbf{R}^T\mathbf{X} = \mathbf{R}(\mathbf{S}\mathbf{\Sigma}\mathbf{V}^T)$ by setting $\mathbf{U} = \mathbf{R}\mathbf{S}$. In order to efficiently find matrix $\mathbf{R}$, we first take Gaussian random matrix $\mathbf{\Omega} \in \mathbb{R}^{|\mathcal{V}_g| \times d_g}$, compute $\mathbf{Y} = \mathbf{X}\mathbf{\Omega}$, then to take the QR decomposition of $\mathbf{Y}$. After that the embeddings of nodes in $\mathcal{V}_g$ can be readily obtained as $\mathbf{E}_g(\mathcal{V}_g) = \{E_g(u_i)\}_{u_i \in \mathcal{V}_g} = \mathbf{R}_{d_g}\mathbf{S}_{d_g}\mathbf{\Sigma}_{d_g}^{1/2}$.

Compare to the embeddings of nodes $\mathbf{E}_c(\mathcal{V}_c)$ in cascade graphs which we described in Section 4.1.1, embeddings of nodes $\mathbf{E}_g(\mathcal{V}_g)$ in the global graph express distinct notions of information diffusion in graphs. As for cascade graphs, nodes with similar structural positions will have close embeddings even if they reside in very different areas of the graph – whether those rarely appeared influential nodes, bridging nodes which connect communities, or leaf nodes which dominant in numbers – their position functionalities are all captured by heat wavelet diffusion patterns. As for the global graph, the low dimensional continuous embeddings the model learned by a mapping function $f : \mathcal{V}_g \rightarrow \mathbf{E}_g(\mathcal{V}_g)$ preserves node proximity associated with the graph, in which, nodes with similar preferences and behavior will possess similar geometric embeddings.

4.2 Temporal Diffusion Learning

The embeddings generated from above spectral graph wavelets and sparse matrix factorization represent the structural information that nodes carry in the cascade graphs $\mathcal{G}_c$ and the global graph $\mathcal{G}_g$. Specifically, (1) structurally equivalent nodes in cascade graphs will have similar embeddings (cf. [62]), – e.g., hub nodes are more influential than leaf nodes to propagate information to other nodes; and (2) proximal nodes in the global graph will have close embeddings, i.e., nodes in proximity to each other carry similar interests to facilitate the diffusion of information. However, the temporal pattern encoded in the diffusion process is also important and has a critical impact on the popularity prediction of cascades. To capture such temporal characteristics, we leverage bi-directional GRU (Bi-GRU) [15] to model the cascading behavior in cascades. RNNs are a natural choice and have been widely used in the literature – e.g., [45], [70] have used RNNs for modeling the sequential patterns during the information diffusion.

For a given cascade of C, we have $|\mathcal{V}_c|$ node embeddings $\mathbf{E}_c(\mathcal{V}_c) = \{E_c(u_i)\}_{i \in |\mathcal{V}_c|}$ pre-trained from cascade graphs via spectral graph wavelets, and for each node u_i in $\mathcal{V}_c$, if u_i is also in the global graph, i.e., $u_i \in \mathcal{V}_g$, then we have its corresponding embedding $E_g(u_i)$, otherwise we set $E_g(u_i) = \mathbf{0} \in \mathbb{R}^{d_g}$ as a cold starting. Afterward, embeddings of nodes would be sequentially fed through a two layers of Bi-GRU to generate context-dependent representation. For each input $E_c(u_i)$ and $E_g(u_i)$, GRU computes the updated hidden state with gated units. By concatenating the outputs of the forward $\overrightarrow{\mathrm{GRU}}$ and backward $\overleftarrow{\mathrm{GRU}}$, the final representation $\mathbf{h}_2$ of a cascade is obtained as

$$\begin{gathered}\mathbf{E} = \mathrm{Concat}(E_c(u_i), E_g(u_i)), \overrightarrow{\mathbf{h}}_1 = \overrightarrow{\mathrm{GRU}}(\mathbf{E}),\\ \overleftarrow{\mathbf{h}}_1 = \overleftarrow{\mathrm{GRU}}(\mathbf{E}), \mathbf{h}_1 = \mathrm{Concat}(\overrightarrow{\mathbf{h}}_1, \overleftarrow{\mathbf{h}}_1), \overrightarrow{\mathbf{h}}_2 = \overrightarrow{\mathrm{GRU}}(\mathbf{h}_1),\\ \overleftarrow{\mathbf{h}}_2 = \overleftarrow{\mathrm{GRU}}(\mathbf{h}_1), \mathbf{h}_2 = \mathrm{Concat}(\overrightarrow{\mathbf{h}}_2, \overleftarrow{\mathbf{h}}_2),\end{gathered} \tag{6}$$

where $\overrightarrow{\mathbf{h}}_1$ and $\overleftarrow{\mathbf{h}}_1$ are full sequence output vectors from GRU, $\overrightarrow{\mathbf{h}}_2$ and $\overleftarrow{\mathbf{h}}_2$ are last hidden output vectors. At the moment $\mathbf{h}_2$ can be readily used for predicting the cascade's popularity, as done in many previous works [17], [18]. We call this model as CasFlow-RNN.

However, there is a drawback when using only the last hidden state of an RNN for cascade prediction. This is caused by the flat sequential generation process followed by RNNs, where each embedding of a node is only conditioned on the previous ones. The problem stems from the fact that the model is forced to generate all high-level structures locally on a step-by-step basis, and in a deterministic way. This, in turn, is a heavy constraint for exploring the uncertain dependencies among cascades. In addition, limited by the capability of real implementation (i.e., LSTM and GRU), these models cannot handle long-term dependencies and their performance may significantly drop for predicting the larger size of cascades.

4.3 Modeling Information Diffusion Uncertainty

In this work, we present a deep generative model to capture the uncertainty in the information diffusion. Towards this goal, we employ to model the diffusion uncertainty via variational autoencoders (VAE) [71]. VAE is a generative network consisting of an encoder and a decoder and provides a general framework for learning latent representations, where a joint probability distribution over the data and the posterior on latent random variables are learned. The learned representations can be used for both data generation as well as other tasks, such as classification [72], node representation [73], predictions [74] and recommendation [75]. As a probabilistic approach, VAE provides a solid mathematical foundation to cope with randomness and uncertainty, which motivates us to model the cascade uncertainty with such a Bayesian framework.

Node (Lower) Level Uncertainty Modeling. A cascade C is composed of an evolving sequence of participants, each one is associated with a learned representation on behalf of a certain stage of information diffusion. In Sections 4.1.1 and 4.1.2, for each node in the cascade graph or and the global graph, we have learned $E_c(u_i)$ and $E_g(u_i)$ through graph representation learning, respectively. However, in a more general sense, any other types of representations can be used here to enhance the model learning ability, e.g., text/image embeddings. Without the loss of generality, we use $\mathcal{R}_i, (i \in |\mathcal{V}_c|)$ to represent each participant in cascade C, i.e., in our case, $\mathcal{R}_i = \mathrm{Concat}(E_c(u_i), E_g(u_i))$.

Let $\mathbf{Enc}(\cdot)$ be the encoder of inputs, and $\mathbf{Dec}(\cdot)$ be the decoder to reconstruct the inputs, a deep variational autoencoder based on neural networks (NN) can be formalized as

$$\begin{aligned}&\mathbf{z}_i = \mathbf{Enc}(\mathcal{R}_i), \overline{\mathcal{R}}_i = \mathbf{Dec}(\mathbf{z}_i), \text{ for } i = 1, 2, \ldots, M\\ &\mu_i = \mathrm{NN}(\mathcal{R}_i), \log\sigma_i^2 = \mathrm{NN}(\mathcal{R}_i), \mathbf{z}_i \sim \mathcal{N}(\mu_i, \sigma_i^2),\end{aligned} \tag{7}$$

where $\overline{\mathcal{R}}_i$ is reconstructed input, $\mathbf{z}_i \in \mathbb{R}^{d_z}$ is the latent vector. VAE takes a high dimensional data as input to generate a compressed hidden representation sampled from a conditional prior distribution with μ and $\log\sigma^2$, then to reconstruct the original input from hidden representation.

In order to learn a probabilistic representation for cascade data to capture its dynamics and uncertainty, VAE draw μ and $\log\sigma^2$ from the output of encoder then use reparameterization to sample latent vector $\mathbf{z}$ from Gaussian distribution [71], i.e., $\mathbf{z}_i = \mu_i + \sigma_i\epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$. Moreover, given each representation of participants in cascade, its marginal log-likelihood of $\mathcal{R}_i$ is $\log p_\theta(\mathcal{R}_i) = \log\int p_\theta(\mathcal{R}_i|\mathbf{z}_i)p(\mathbf{z}_i)\mathrm{d}\mathbf{z}_i$, however, this log-likelihood can not be computed efficiently in cases of the latent representation having a high dimension. In addition to the intractable computation of $\log p_\theta(\mathcal{R}_i)$, maximizing the evidence lower bound (ELBO) by observing a parametric prior $q_\phi(\mathbf{z}_i|\mathcal{R}_i)$ is equivalent or approximate to the true posterior $p_\theta(\mathbf{z}_i|\mathcal{R}_i)$

$$\begin{aligned}&\log p_\theta(\mathcal{R}_i) = \log\int p_\theta(\mathcal{R}_i|\mathbf{z}_i)p(\mathbf{z}_i)\mathrm{d}\mathbf{z}_i\\ &= \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathcal{R}_i})\log\left[\frac{p_\theta(\mathcal{R}_i,\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathcal{R}_i)}\right] + \mathbb{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}_i|\mathcal{R}_i)||p_\theta(\mathbf{z}_i|\mathcal{R}_i))\\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathcal{R}_i)}\left[\log p_\theta(\mathcal{R}_i,\mathbf{z}_i) - \log q_\phi(\mathbf{z}_i|\mathcal{R}_i)\right] \triangleq \mathrm{ELBO}(\mathcal{R}_i),\end{aligned} \tag{8}$$

where $q_\phi(\mathbf{z}_i|\mathcal{R}_i)$ (a.k.a. *encoder* parameterized by ϕ) is an approximation to the true posterior $p_\theta(\mathbf{z}_i|\mathcal{R}_i)$ used to generate the latent vector $\mathbf{z}_i$; and $\mathbb{D}_{\mathrm{KL}}(\cdot)$ is the Kullback-Leibler divergence. Since the objective is to minimize the KL divergence between the proposed $q_\phi(\mathbf{z}_i|\mathcal{R}_i)$ and $p_\theta(\mathbf{z}_i|\mathcal{R}_i)$, we can alternatively maximize ELBO of $\log p_\theta(\mathcal{R}_i, \mathbf{z}_i)$ w.r.t. both parameters θ and ϕ, which are jointly trained with separate nonlinear functions such as neural networks (NN).

By minimizing the reconstruction error between the input $\mathcal{R}_i$ and output $\overline{\mathcal{R}}_i$, the learned latent representation $\mathbf{Z}_1 = \{\mathbf{z}_i\}_{i\in|\mathcal{V}_c|}$ for all participants in cascade C captures the data distribution and can be readily used to generate synthetic data or improve particular tasks [72], [75]. Now, we can immediately combine $\mathbf{Z}_1$ with two layers of Bi-GRUs which we introduced in Section 4.2 for predicting the final popularity of cascades. We call this variant of CasFlow*, which can be considered as a node (lower) level variational inference with temporal modeling. However, this model only captures the individual node uncertainty, ignoring the evolving uncertainty of the cascade – though it indeed improves the prediction performance, as we will see in the experiments. In addition, the lower level variational inference discards the sequential dependencies among the participants.

Cascade (Higher) Level Variational Inference. To overcome the "shallow" generation problem in CasFlow* as mentioned above, we combine RNNs with a sequential VAEs, as shown in Fig. 2. This higher level (cascade) VAE takes M sequential latent variables $\mathbf{Z}_1 = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M\}$ generated from the lower level VAE as inputs, with each $\mathbf{z}_i$ corresponding to a participant in C, to minimize the reconstruction error. Therefore, we can obtain the cascade level latent representation $\mathbf{Z}_2$, which is expected to capture the temporal and sequential relationships, and hence express the causalities and dependencies among information propagation in cascade's evolving trajectory.

Again, let $\mathbf{Enc}(\cdot)$ be an RNN-based encoder of input, and $\mathbf{Dec}(\cdot)$ be an RNN-based decoder to reconstruct the input, an RNN-based VAE can be formalized as follows:

$$\begin{aligned}&\mathbf{Z}_2 = \mathbf{Enc}(\mathbf{Z}_1), \overline{\mathbf{Z}}_1 = \mathbf{Dec}(\mathbf{Z}_2),\\ &\mathbf{Z}_1 = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M\}, \overline{\mathbf{Z}}_1 = \{\overline{\mathbf{z}}_1, \overline{\mathbf{z}}_2, \ldots, \overline{\mathbf{z}}_M\},\\ &\mu = \mathrm{NN}(\mathrm{RNN}(\mathbf{Z}_1)), \log\sigma^2 = \mathrm{NN}(\mathrm{RNN}(\mathbf{Z}_1)),\\ &\mathbf{Z}_2 \sim \mathcal{N}(\mu, \sigma^2), \overline{\mathbf{z}}_i = \mathrm{NN}(\mathrm{RNN}(\mathbf{Z}_2)), \text{ for } i = 1, 2, \ldots, M,\end{aligned} \tag{9}$$

where $\overline{\mathbf{Z}}_1$ is the reconstructed input, M is the length of sequence, $\mathbf{Z}_2 \in \mathbb{R}^{d_{\mathbf{Z}_2}}$ is the learned compressed hidden vector. More specifically, let $\mathcal{R} = \{\mathcal{R}_i\}_{i\in|\mathcal{V}_c|}$ be the input sequences, the joint probability for a cascade C is formulated as

$$p_\theta(\mathcal{R}, \mathbf{Z}_1, \mathbf{Z}_2) = p_\theta(\mathbf{Z}_2|\mathbf{Z}_1)p_\theta(\mathcal{R}|\mathbf{Z}_1, \mathbf{Z}_2), \tag{10}$$

where latent vector $\mathbf{Z}_1$ are centered isotropic multivariate Gaussian distributions. The conditional distribution $p(\mathbf{Z}_2|\mathbf{Z}_1)$ is parameterized by an RNN-based encoder and $p(\mathcal{R}|\mathbf{Z}_1, \mathbf{Z}_2)$ can be considered as cascade reconstruction from the latent factors, formulated as

$$p_\theta(\mathbf{Z}_2|\mathbf{Z}_1) = \sum_{m=1}^{M}\mathcal{N}(\mathbf{Z}_2|f_\vartheta^\mu(\mathbf{Z}_1)), \mathrm{Diag}(f_\vartheta^{\sigma^2}(\mathbf{Z}_1))), \tag{11}$$

$$p_\theta(\mathcal{R}|\mathbf{Z}_1, \mathbf{Z}_2) = \mathcal{N}(\mathcal{R}|f_\varphi^\mu(\mathbf{Z}_1, \mathbf{Z}_2), \mathrm{Diag}(f_\varphi^{\sigma^2}(\mathbf{Z}_1, \mathbf{Z}_2))), \tag{12}$$

where the conditional distribution of the observed cascade $\mathcal{R}$ is the multivariate Gaussian with a diagonal covariance matrix; the mean and diagonal variance are parameterized by neural networks f_*^μ and $f_*^{\sigma^2}$ with parameters ϑ and φ. The ELBO on the marginal likelihood is derived as

$$\begin{aligned}&\log p_\theta(\mathcal{R}) \geq \mathrm{ELBO}(\mathcal{R})\\ &= \mathbb{E}_{q_\phi(\mathbf{Z}_1,\mathbf{Z}_2|\mathcal{R})}\log\left[\frac{p_\theta(\mathbf{Z}_1)p_\theta(\mathbf{Z}_2|\mathbf{Z}_1)p_\theta(\mathcal{R}|\mathbf{Z}_1,\mathbf{Z}_2)}{q_\phi(\mathbf{Z}_2|\mathcal{R},\mathbf{Z}_1)q_\phi(\mathbf{Z}_1|\mathcal{R})}\right]\\ &= \mathbb{E}_{q_\phi(\mathbf{Z}_1,\mathbf{Z}_2|\mathcal{R})}[\log p_\theta(\mathcal{R}|\mathbf{Z}_1,\mathbf{Z}_2) + \log p_\theta(\mathbf{Z}_2|\mathbf{Z}_1)\\ &\quad + \log p_\theta(\mathbf{Z}_1) - \log q_\phi(\mathbf{Z}_2|\mathcal{R},\mathbf{Z}_1) - \log q_\phi(\mathbf{Z}_1|\mathcal{R})]\\ &= \mathbb{E}_{\mathbf{Z}_1\sim q_\phi(\mathbf{Z}_1|\mathcal{R}),\mathbf{Z}_2\sim q_\phi(\mathbf{Z}_2|\mathbf{Z}_1)}[\log p_\theta(\mathcal{R}|\mathbf{Z}_1,\mathbf{Z}_2)]\\ &\quad - \mathbb{D}_{\mathrm{KL}}(q_\phi(\mathbf{Z}_2|\mathcal{R},\mathbf{Z}_1)||p_\theta(\mathbf{Z}_2|\mathbf{Z}_1)) - \mathbb{D}_{\mathrm{KL}}(q_\phi(\mathbf{Z}_1|\mathcal{R})||p_\theta(\mathbf{Z}_1)).\end{aligned} \tag{13}$$

The first term denotes the reconstruction cost – which is the expected negative log-likelihood of the observed diffusion, encouraging the model to efficiently decode the sequential participants from a set of latent variables $\mathbf{Z}_1$ and $\mathbf{Z}_2$. The two $\mathbb{D}_{\mathrm{KL}}(\cdot)$ terms are regularizers that encourage the inferred latent factors to match the two priors – isotropic multivariate Gaussian and conditional mixture of Gaussian, respectively, reflecting the information loss when optimizing the ELBO.

Variational Inference via Normalizing Flows. Above we have our model CasFlow incorporated with lower node level

VAEs and a higher level cascade level VAE to model the uncertainty during the diffusion of information. Specifically, the learned latent representation of input data are sampled from simple families of Gaussian posterior distribution on both level VAEs. However, in practical situations, the Gaussian assumption of the conditional distribution is not as flexible as many other complex distributions exist in real-world applications, especially for information cascade data in which their popularity distribution is highly skewed [9], [76] – thereby significantly influence the quality of variational inference. In order to infer more complex, flexible, and scalable posterior distribution families, we turn to utilize a powerful probabilistic technique – *normalizing flows* (NFs) [77], [78], [79] – to construct rich posterior approximations.

Algorithm 1. CasFlow Learning

```
Input: Observed information cascade $C_k(t_o)$ and its corresponding global graph $\mathcal{G}_g$.
Output: Predicted cascade size $\hat{P}_k(t_p)$.
Obtain the cascade graph $\mathcal{G}_c$ from $C_k(t_o)$;
Compute graph wavelets $\Psi_{u,s}$ for each node $u_i$ (Eq. (2));
Compute embeddings $\mathbf{E}_c(\mathcal{V}_c)$ in cascade graph (Eq. (3));
Compute embeddings $\mathbf{E}_g(\mathcal{V}_g)$ in global graph (Eq. (5));
while not convergent do
    Train the Bi-GRUs to obtain $\mathbf{h}_2$ (Eq. (6));
    for each user $i \in |\mathcal{V}_c|$ do
        Compute $\mathbf{z}_i$ by optimizing Eq. (8) using Eq. (7);
    end for
    Obtain $\vec{\mathbf{Z}}_1 = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{|\mathcal{V}_c|}\}$;
    Train cascade VAE to obtain $\mathbf{Z}_2$ by optimizing Eq. (13);
    Obtain $\mathbf{Z}_3$ using $K$ transformations of $\mathbf{Z}_2$ (Eq. (15));
    Combine $\mathbf{h}_2$ and $\mathbf{Z}_3$ for prediction via Eq. (22);
end while
```

Given a latent random variable $\mathbf{Z} \in \mathbb{R}^{d_\mathbf{Z}}$ (in our case the $\mathbf{Z}_2$ learned from the higher level VAE), normalizing flows are generative models aiming to transform observed vector $\mathbf{Z}$ to the desired target latent vector $\mathbf{Z}^{(K)}$ through a length of K invertible mappings with which the Jacobians are tractable and the functions are differentiable. To be specific, NFs use a mapping function $f : \mathbf{Z} \rightarrow \mathbf{Z}'$ as follows:

$$q(\mathbf{Z}') = q(\mathbf{Z})\left|\det\frac{\partial f^{-1}}{\partial \mathbf{Z}'}\right| = q(\mathbf{Z})\left|\det\frac{\partial f}{\partial \mathbf{Z}}\right|^{-1}, \tag{14}$$

where $q(\mathbf{Z})$ is the distribution of random vector $\mathbf{Z}$, and the transformation f is invertible. To obtain a valid probability density $q_K(\mathbf{Z}^{(K)})$ from the initial density $q_0(\mathbf{Z})$, a K hierarchical transformations of NFs successively applying Eq. (14) to compute the target density

$$\mathbf{Z}^{(K)} = f_K(\mathbf{Z}^{(K-1)}) = f_K(f_{K-1}(\ldots f_2(f_1(\mathbf{Z})))), \tag{15}$$

$$\ln q_K(\mathbf{Z}^{(K)}) = \ln q_0(\mathbf{Z}) - \sum_{k=1}^{K} \ln\det\left|\frac{\partial f_k}{\partial \mathbf{Z}^{(k)}}\right|. \tag{16}$$

Once the mapping functions are expressive and appropriate, the learned mixing distribution of latent random vector is more fit to the true distribution than to simply modeled as independent Gaussians. To enable efficient inference with NFs, consider transformation $f(\mathbf{Z}) = \mathbf{Z} + \mathbf{u}h(\mathbf{w}^T\mathbf{Z} + b)$, where $\mathbf{w} \in \mathbb{R}^{d_\mathbf{Z}}$, $\mathbf{u} \in \mathbb{R}^{d_\mathbf{Z}}$ and $b \in \mathbb{R}$ are parameters, $h(\cdot)$ is a smooth element-wise non-linearity. Then logdet-Jacobian term (Eq. (14)), approximate posterior distribution (Eq. (16)), and marginal likelihood (Eq. (13)) can be rewrite as

$$\psi(\mathbf{Z}) = h'(\mathbf{w}^T\mathbf{Z} + b)\mathbf{w}, \tag{17}$$

$$\det\left|\frac{\partial f}{\partial \mathbf{Z}}\right| = \left|\det(\mathbf{I} + \mathbf{u}\psi(\mathbf{Z})^T)\right| = \left|1 + \mathbf{u}^T\psi(\mathbf{Z})\right|, \tag{18}$$

$$\ln q_K(\mathbf{Z}^{(\mathbf{K})}) = \ln q_0(\mathbf{Z}) - \sum_{k=1}^{K} \ln\left|1 + \mathbf{u}_k^T\psi_k(\mathbf{Z}^{(k)})\right|, \tag{19}$$

$$\log\, p_\theta(\mathcal{R}) \geq \mathrm{ELBO}(\mathcal{R}) + \mathbb{E}\left[\sum_{k=1}^{K} \ln\left|1 + \mathbf{u}_k^T\psi_k(\mathbf{Z}^{(k)})\right|\right]. \tag{20}$$

4.4 Prediction

Now we have obtained $\mathbf{h}_2$ from the two layers of Bi-GRUs (cf. Section 4.2), $\mathbf{Z}_3 = \mathbf{Z}^{(K)}$ from the hierarchical VAEs and K transformations of flows (cf. Section 4.3), which can be fed into MLPs to make the final cascade prediction

$$\hat{P}_k(t_p) = \mathrm{MLPs}(\mathrm{Concat}(\mathbf{h}_2, \mathbf{Z}_3)), \tag{21}$$

by minimizing the mean square logarithmic error (MSLE) loss function

$$\begin{aligned}\mathcal{L}(\mathcal{R}_k;\Theta) = &\frac{1}{N}\sum_{k=1}^{N}(\log\hat{P}_k(t_p) - \log P_k(t_p))^2 \\ &- \mathrm{ELBO}(\mathcal{R}_k),\end{aligned} \tag{22}$$

where N is the total number of cascades, $P_k(t_p)$ is the ground truth (e.g., the number of users who retweet the cascade C_k) and $\hat{P}_k(t_p)$ is the predicted popularity for cascade C_k, and $\mathrm{ELBO}(\mathcal{R}_k)$ is the ELBO that needs to be maximized as given by Eq. (20).

4.5 Complexity Analysis

Since the popularity of information cascades often follows a heavy-tailed distribution [9], [42], and typical online social networks have millions of nodes and edges, efficiently modeling of both cascade graphs and global graph is of great importance for cascade learning systems. Compared to conventional graph cascade models, especially those random walk-based [17] and GNN-based models [18], [49], CasFlow can handle large-sized graphs efficiently, the time complexities for cascade graphs and global graph are both linear to the number of edges.

Specifically, recall that $|\mathcal{V}_c|$ and $|\mathcal{E}_c|$ are number of nodes and edges in cascade graph $\mathcal{G}_c$, $|\mathcal{V}_g|$ and $|\mathcal{E}_g|$ are number of nodes and edges in global graph, d_c and d_g are dimensions of nodes in cascade graph and global graph, respectively.

- *Complexity for Computing Embeddings of Nodes in Cascade Graph*: the spectral graph wavelets (Eq. (2)) are computed by Chebyshev polynomials [80], the time complexity is $O(h|\mathcal{E}_c|)$, which is linear to the number of edges and h is the order of Chebyshev polynomial approximation [81].
- *Complexity for Computing Embeddings of Nodes in Global Graph*: as in [67], the computation of TSVD and QR

TABLE 2
Descriptive Statistics of Three Datasets

Dataset	Twitter	Weibo	APS
# Cascades	88,440	119,313	207,685
# Nodes in $\mathcal{G}_g$	490,474	6,738,040	616,316
# Edges in $\mathcal{G}_g$	1,903,230	15,249,636	3,304,400
Avg. popularity	142	240	51
Number of cascades in two observation settings			
Train (1d/0.5h/3y)	9,639	21,463	18,511
Val (1d/0.5h/3y)	2,066	4,599	3,967
Test (1d/0.5h/3y)	2,065	4,599	3,966
Train (2d/1h/5y)	12,739	29,908	32,102
Val (2d/1h/5y)	2,730	6,409	6,879
Test (2d/1h/5y)	2,729	6,408	6,879
Basic statistics of cascade graphs			
Avg. sequence length	2.196	2.237	3.999
Avg. structural virality	1.995	2.025	3.114
Avg. page rank	0.073	0.045	0.189
Avg. graph density	0.183	0.090	0.320

decomposition is $O(d_g^2|\mathcal{V}_g|)$, and because of $d_g \ll |\mathcal{V}_g|$, the overall complexity of sparse matrix factorization is $O(d_g^2|\mathcal{V}_g| + |\mathcal{E}_g|)$.

• *Complexity for Other Parts of CasFlow*: the time and space complexities of GRU and MLP are related to the input dimensions of latent variables.

Under our experimental settings, *CasFlow* has ~2M parameters, and costs ~83 ms for one step training with batch size of 64 and ~6.78 mins for generating embeddings of global graph $\mathcal{G}_g$ which has ~15M edges. We conduct time cost comparison between baselines in Section 5.4. The overall learning process of CasFlow is sketched in Algorithm 1.

5 Experiments

We now introduce the details of three benchmarking datasets, followed by the evaluation of our models against the state-of-the-art baselines on information cascade popularity prediction. Extensive studies on model ablation and interpretability are also discussed.

5.1 Experimental Settings

Datasets. Cascades can be formed by different types of information, e.g., social tweets, online images/videos, emails, news articles, research papers, and so on. We selected three publicly available datasets – Twitter, Weibo, and APS – that have been commonly used in previous related works [12], [18], [36] for evaluating cascade popularity prediction.

• *Twitter* dataset collected by [82] contains public English written tweets published between Mar 24 and Apr 25, 2012. We take hashtags and their adopters as independent information cascades. The global graph of Twitter dataset is constructed using multiple relations, including reciprocal follower/followee, retweeting, and mentioning interactions between users. The cascade graphs are built based on all three relationships above.

• *Sina Weibo* is the largest microblogging platform in China, where every tweet and its retweets can form a retweeting cascade [12]. The global graph in this Weibo dataset is constructed by all user retweeting relationships.

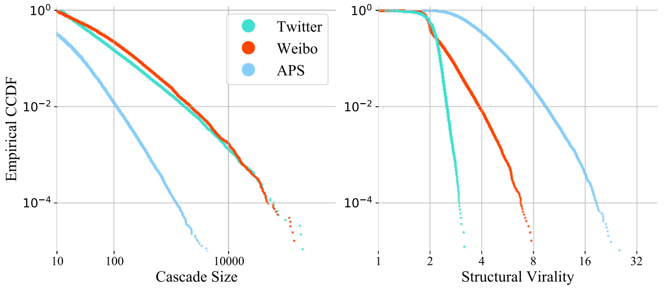


Fig. 3. The empirical complementary cumulative distribution (CCDF) of cascade size and structural virality for three datasets. The popularity of Twitter and Weibo tweets are typically larger than APS papers, but their structural virality is smaller.

• *American Physical Society (APS)* contains scientific papers published by APS journals. Every paper in the APS dataset and its citations form a citation cascade. We define the global graph in APS as an author interaction graph.

Evaluating models on three distinct datasets offers a systematic view on the generalization capability, without domain knowledge and feature engineering. Descriptive statistics of three datasets are shown in Table 2 and Fig. 3. We can observe that APS has smaller average popularity than Weibo and Twitter, and its structural virality (calculated by Wiener index [9], [83]) is higher than the other two, indicating that the popularity of scientific papers is mainly driven by the propagation of other papers rather than directly citing the original. As for Twitter hashtags, their structural virality is generally smaller, in which shows the diffusion mechanism is mainly driven by broadcasting.

In addition, the observation time t_o is set to 1 and 2 days for Twitter, 0.5 and 1 hour for Weibo, 3 and 5 years for APS. We select 32 days as the prediction time t_p for Twitter hashtags, 24 hours for Weibo tweets and 20 years for APS papers, following previous works [12], [18]. We filter out cascades whose $|C(t_o)| < 10$. And for cascades whose $|C(t_o)| > 100$, we only select first 100 participants. We track Twitter hashtags before Apr 10, ensuring at least 15 days for each hashtag to grow adopters. Due to the effect of diurnal rhythm in Weibo, we focus on tweets posted between 8 a.m. and 6 p.m., leaving each tweet at least 6 hours to reap retweets. As for APS, we consider papers published between 1893 and 1997 – so that each paper has at least 20 years (1997 - 2017) to gain citations.

Baselines. To evaluate whether our design is effective in cascade prediction, we compare our model with following three groups of baselines that are either unable to capture deep structural and temporal information or uncertainty.

• *Feature Engineering-Based*: is the most widely used prediction models for information cascades. These models first extract hand-crafted features from data, then feed features into machine learning models for training and evaluating, e.g., Szabo & Huberman [27] use observed popularity $P_j(t_o)$ to predict $\hat{P}_j(t_p)$ of news articles and online videos; It uses observed popularity and cumulative popularity series as features. We denote this method as *Feature-S&H*. Cheng *et al.* [9] grouped five classes of features that drive cascade growth, including content features, original poster/re-sharer features, structural and temporal features. Specifically, it includes cumulative popularity series, time between original and first participant, mean time between the first half and the second half of participants, number of leaf nodes, mean node degree, mean and max length of sequences. We feed these features into a linear regression model and a MLP model. We denote these methods as *Feature-Linear* and *Feature-Deep*, respectively.

• *Statistical Model-Based*. Researchers build time series models to make cascade popularity prediction, such as Pinto & Almeida *et al.* [31], denoted as *TimeSeries*. Cao *et al.* combine both deep learning and Hawkes process for cascade size prediction. It considers three key aspects of Hawkes process, i.e., influence of users, self-exciting mechanism, and time decay effect [12]. We denote this method as *DeepHawkes*.

• *Deep Learning-Based*: *CasCN* is a graph convolution network (GCN)-based framework exploiting both temporal and structural information for cascade prediction. It samples sub-cascade graphs and uses LSTM to capture the evolving process [18]. *DMT-LIC* is a multi-task model, which jointly learns user-level behavior and cascade-level prediction via a shared-representation layer and attention/gated mechanisms [48]. Note that we omit the comparison with some prediction models such as DeepCas [17], CYAN-RNN [13], DeepInf [84], FOREST [85], etc., since they mainly predicting the microscopic node activation rather than cascade popularity, or only considering the structural modality.

Parameter Setting: For each of three datasets, we randomly split it into training set (70%), validation set (15%), and test set (15%). All models, including ours, are tuned to the best performance with early stopping when validation errors has not declined for 10 consecutive epochs. For baselines, the learning rate and L_2 coefficient are selected from $10^{\{0,-1,-2,\ldots,-8\}}$; node embedding size for DeepHawkes, CasCN and DMT-LIC is set to 50; the batch size is 64; and all the other hyper-parameters are set to the same values as used in the original papers.

For the scale parameter s used for node embedding $\mathbf{E}_c(\mathcal{V}_c)$ in CasFlow, we use a theoretically justified method proposed in [62] to select s in the appropriate range $[s_{\min}, s_{\max}]$. That is, we directly use two scale parameters $s_{\min}$ and $s_{\max}$ to generate the final node embedding $E_c(u_i) = [E_{c,s_{\min}}(u_i), E_{c,s_{\max}}(u_i)]$ by a concatenation operation, with $d = 10$ evenly spaced points, the final embedding size d_c is 40. For node embedding $\mathbf{E}_g(\mathcal{V}_g)$ in the global graph, embedding size d_g is set to 40, too. The dimensionality of the latent factor $\mathbf{Z}_1$, $\mathbf{Z}_2$, and $\mathbf{Z}_3$ are all set to 64. The number of GRU units is 128. The K of NFs is 8. The hidden units in two-layer MLPs are 64 and 32, respectively.

Evaluation Protocols. Following existing works [10], [12], we use mean square logarithmic error (MSLE) and mean absolute percentage error (MAPE) for prediction performance evaluation, which are defined as

$$\mathrm{MSLE} = \frac{1}{N}\sum_{i=1}^{N}(\log_2 \Delta\hat{P}_i - \log_2 \Delta P_i)^2, \tag{23}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\log_2 \Delta\hat{P}_i - \log_2 \Delta P_i|}{\log_2 \Delta P_i}, \tag{24}$$

where N is the total number of cascades in test set and $\Delta P_i = P_i(t_p) - P_i(t_o)$ is the incremental cascade size.

We also report the results of coefficient of determination (R^2) and the percentage of Top-k coverage (COV-k) – the latter is defined by the ratio of successfully predicted cascades among the largest k, i.e., given a set of k largest cascades and a set of k predicted largest cascades, COV-k is calculated by the size of intersection of two sets divided by k. We set $k = \lfloor N/10 \rfloor$ in this study.

Experimental Environment. The experiments are conducted on Intel E5-2680 v4 2.40 GHz, one NVIDIA GeForce GTX 1080 Ti, and with 64 GB memory. The CasFlow was trained using TensorFlow with Adam [86] optimizer, the total time cost (including preprocessing, training, and evaluation) is less than an hour.

5.2 Performance Comparison

The overall performance of *CasFlow*, as well as the baselines, are shown in Table 3 and Figure 4. We have the following observations:

(O1): *CasFlow* outperforms the baselines by a significant margin. For the Weibo dataset, the results of *CasFlow* surpass the best baseline (DMT-LIC) by 12.7%~15.2%, which demonstrates the benefit of our hierarchical information cascade component.

(O2): The gaps between Feature models and other baselines are quite small, and in some cases feature engineering-based and statistical approaches even beat deep learning models, implying that deep learning models are not always better than feature engineering-based methods. However, its performance heavily relies on hand-crafted features, which are labor intensive and difficult to be generalized to other scenarios. This is verified by the results of Feature-based on the APS dataset, where it performs worse on MSLE.

(O3): DeepHawkes, on the contrary, does not consider the topology information of cascades. Therefore, its performance relies on the time-series modeling capability and diffusion route, which may prefer to overrate the cascade size due to its rudimentary self-excitation mechanism [42]. CasCN only leverages the structural and temporal factors for cascade prediction. However, it focuses on the local structure learning, ignoring the global user behavior.

(O4): Among the baselines, DMT-LIC performs the best because of its multi-task learning mechanism, which not only considers the structural propagation of cascades, but also investigates the individual behavior of nodes. To an extent, it implicitly learns the hierarchical information of cascades. Thus, the CasFlow performance gain over DMT-LIC illustrates the superiority of modeling the uncertainty of information diffusion at both the node- and cascade-level.

5.3 Ablation Study

To better investigate the contribution of each component in CasFlow, we design and implement eight variants:

• *CasFlow-LocalStruct* and *CasFlow-GlobalStruct* – we separately remove the global node embeddings and cascade node embeddings, i.e., $\mathbf{E}_g(\mathcal{V}_g)$ and $\mathbf{E}_c(\mathcal{V}_c)$, respectively.

• *CasFlow-Temporal* and *CasFlow-Structural* – we separately remove the structural and temporal information, respectively. For *CasFlow-Temporal*, nodes in cascade graphs are directly connected to the root and we don't use global graph.

TABLE 3
Performance Comparison Between Baselines and CasFlow on Three Datasets With Different Observation Times, Measured by MSLE and MAPE (Lower is Better)

Model	Twitter				Weibo				APS			
	1 Day		2 Days		0.5 Hour		1 Hour		3 Years		5 Years	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
Feature-S&H	14.792	0.960	13.515	0.983	4.455	0.390	4.001	0.398	2.382	0.316	2.348	0.350
TimeSeries	8.214	0.547	6.023	0.445	3.119	0.277	2.693	0.268	1.867	0.271	1.735	0.291
Feature-Linear	9.326	0.520	6.758	0.459	2.959	0.258	2.640	0.271	1.852	0.272	1.728	0.291
Feature-Deep	7.438	0.485	6.357	0.500	2.715	0.228	2.546	0.272	1.844	0.270	1.666	0.282
DeepHawkes	7.216	0.587	5.788	0.536	2.891	0.268	2.796	0.282	1.573	0.271	**1.324**	0.335
CasCN	7.183	0.547	5.561	0.525	2.804	0.254	2.732	0.273	1.562	0.268	1.421	0.265
DMT-LIC	7.152	0.467	5.427	0.481	2.752	0.249	2.689	0.270	1.539	0.264	1.398	0.258
CasFlow-LocalStruct	7.254	0.475	5.366	0.370	2.681	0.228	2.488	0.251	1.814	0.267	1.686	0.285
CasFlow-GlobalStruct	11.244	0.704	10.619	0.709	3.014	0.274	2.780	0.291	1.478	0.241	1.546	0.266
CasFlow-Temporal	7.258	0.450	5.436	0.375	2.691	0.228	2.566	0.272	1.798	0.266	1.682	0.283
CasFlow-Structural	10.860	0.680	9.927	0.620	2.939	0.266	2.797	0.292	1.480	0.237	1.574	0.273
CasFlow-RNN	7.273	0.467	5.392	0.377	2.444	0.217	2.234	**0.232**	1.367	0.227	1.365	**0.244**
CasFlow-VAE	7.138	**0.428**	5.178	**0.337**	2.712	0.260	2.561	0.272	1.463	0.234	1.481	0.271
CasFlow*	7.340	0.435	5.119	0.383	2.429	0.217	**2.206**	0.245	**1.346**	0.223	1.373	0.251
CasFlow-noNF	7.272	0.429	**5.083**	0.345	2.501	0.223	2.291	0.246	1.370	0.227	1.401	0.251
CasFlow	**6.954***	0.455*	5.143*	0.361*	**2.402***	**0.210***	2.279*	0.238*	1.361*	**0.222***	1.354*	0.248*
(improves)	↑2.7%	↑8.3%	↑6.3%	↑24.3%	↑12.7%	↑15.7%	↑18.0%	↑13.4%	↑12.5%	↑15.9%	↓-2.2%	↑5.4%

A paired t-test is performed and $*$ indicates a statistical significance $p < 0.001$ as compared to the best baseline method.

- *CasFlow-RNN* and *CasFlow-VAE* – the former uses two-layer Bi-GRU to output $\mathbf{h}_2$ for modeling cascades and predicting the cascade popularity, i.e., without hierarchical variational representation learning, which is just on the contrary to the latter.
- *CasFlow** – is the shallow version of CasFlow and only leverages lower-level uncertainty representation, i.e., $\mathbf{Z}_1$, combined with $\mathbf{h}_2$, for cascade popularity prediction.
- *CasFlow-noNF* – in which we remove the normalizing flows part, i.e., we use $\mathbf{Z}_2$ and $\mathbf{h}_2$ for prediction.

Table 3 outlines the performance comparison among CasFlow and its variants, which illustrates that: (i) CasFlow-LocalStruct shows considerably better performance than CasFlow-GlobalStruct for Twitter and Weibo, but for APS, the global structure is a more reliable predictor. In addition, the combination of local and global structure indeed improves the performance; (ii) CasFlow-Temporal consistently outperforms CasFlow-Structural for Twitter and Weibo. However, the structural information shows more importance for APS papers. This verifies the benefit to model both temporal and structural information of cascades, and none of the above two would dominant in all three scenarios; (iii) Surprisingly, without modeling the hierarchical cascading effect, CasFlow-RNN obtains quite well results, which can be attributed to the node embeddings of our method that involves both local and global representations. In addition, CasFlow-VAE performs very good on Twitter dataset, which shows another benefit to model the diffusion uncertainty; and (iv) The fact that both CasFlow* and CasFlow-noNF show comparable or better performances demonstrates our motivation of modeling hierarchical diffusion uncertainties.

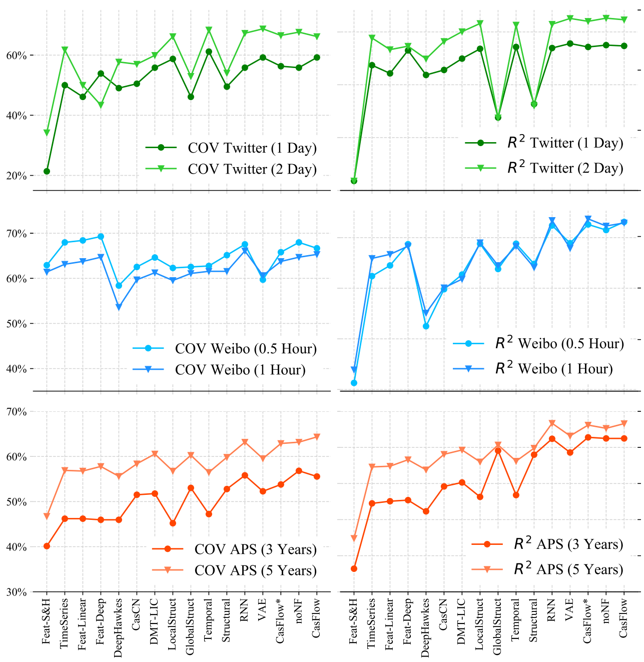
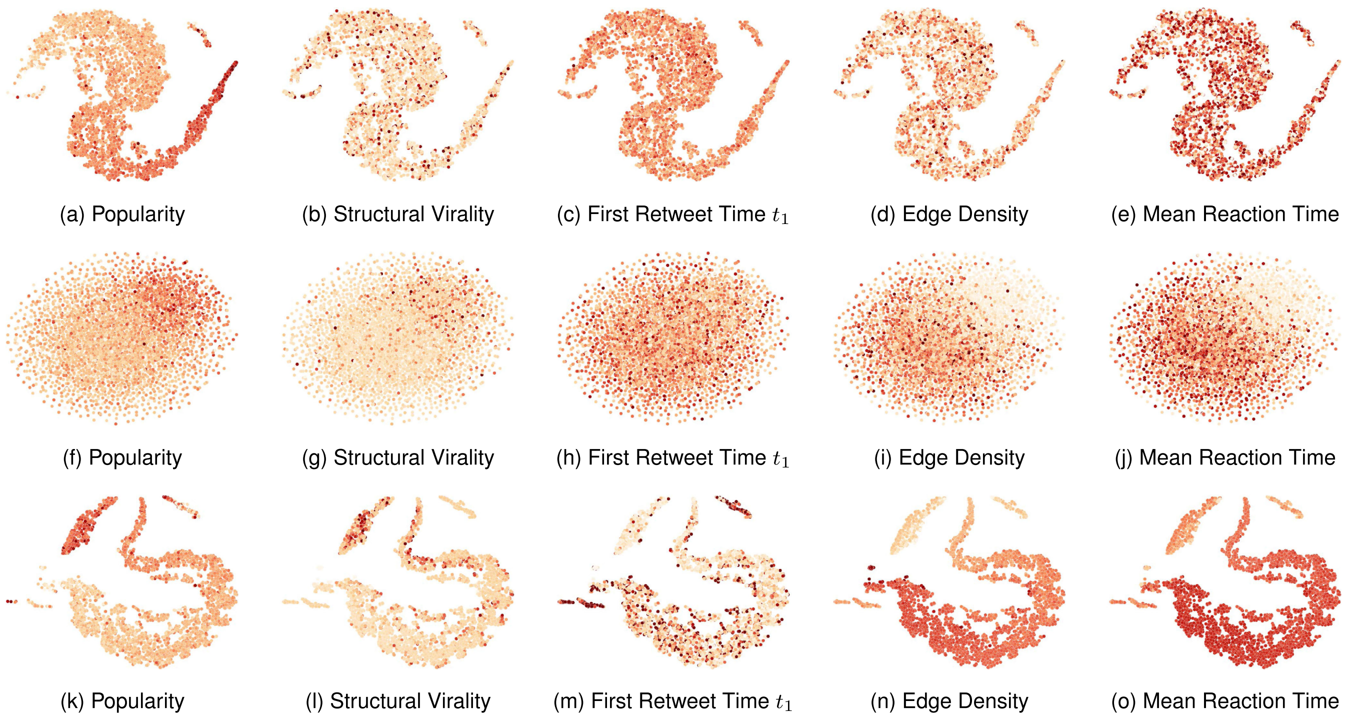


Fig. 4. Performance comparison on three datasets. The evaluation protocols are Top-10% coverage (COV) and coefficient of determination R^2.

5.4 Model Interpretability

We now turn to interpret the performance of CasFlow.

Latent Representation. To have an intuitive explanation regarding the superiority of CasFlow (especially the VAE and NF components), following previous works [17], [18], we plot the learned latent representation of cascades for CasFlow-RNN, CasFlow-noNF and CasFlow in three respective lines of Fig. 5 using *t*-SNE [87]. Each point in the plot represents a cascade in test set (cascades with similar latent vectors are close in the plot), and the color of point indicating one of the five feature groups: *popularity*, *structural virality*, *first retweet time*, *edge density*, and *mean reaction time*. The darker the point, the larger value of that feature to the cascade. As shown in Figs. 5a, 5b, 5c, 5d, and 5e, except cascade *popularity*, other features - whether structural or temporal - didn't show explicit patterns w.r.t. cascades' 2-D



(a) Popularity (b) Structural Virality (c) First Retweet Time t_1 (d) Edge Density (e) Mean Reaction Time

(f) Popularity (g) Structural Virality (h) First Retweet Time t_1 (i) Edge Density (j) Mean Reaction Time

(k) Popularity (l) Structural Virality (m) First Retweet Time t_1 (n) Edge Density (o) Mean Reaction Time

Fig. 5. Visualization of the learned latent representation on Weibo dataset (observation time is set to 0.5 hour) using t-SNE. Each point is a sample from 4,599 test cascades. The darker the point, the larger the value of popularity, structural virality, first retweet time t_1, edge density, or mean reaction time, for each one of five plots in one line, respectively. First line (a-e): latent representation from the last MLPs layer of CasFlow-RNN; Second line (f-j): latent representation $\mathbf{Z}_2$ from CasFlow-noNF; Third line (k-o): latent representation from the last MLPs layer of CasFlow.

projections. This phenomenon indicates that by only utilizing the RNNs, our model cannot explain the relationship between cascade features and the learned latent representations. Instead, points in Figs. 5f, 5g, 5h, 5i, and 5j are latent vectors $\mathbf{Z}_2$ retrieved form cascade VAE of CasFlow-noNF, and the shapes of the plots are consistent with the Gaussian assumption of VAE in our model. We can see that except the *first retweet time*, other features show clear clustering effects compared to the ground truth *popularity*, i.e., with larger *structural virality* and smaller *edge density* and *mean reaction time*, the popularity of cascades tends to be larger than others. What is also worth noticing is that in CasFlow we did not use these features for training/testing but the model itself learns meaningful and explainable semantics of which feature correlates to the future popularity. Finally, as shown in Table 3 and Figs. 5k, 5l, 5m, 5n, and 5o, our model learns better representations in terms of both accuracy and interpretability compared to CasFlow-RNN and CasFlow-noNF – e.g., Fig. 5m indicates large cascades are often associated with smaller *first retweet time* t_1 – which points out the benefits by incorporating the variational inference and normalizing flows into the learning of popularity prediction.

Hyper-Parameter Sensitivity. CasFlow involves many hyper-parameters, some of which are important and might affect the model performance. We use the Weibo dataset to conduct an ablation study on four important hyper-parameters. The value of vertical line in Fig. 6 indicates default parameter setting used in experiments. Detailed results are explained below.

- *Impact of weight for VAE and NF losses*: we give a weight on the losses of VAEs and NF (cf. Eqs. (13) and (20)), which

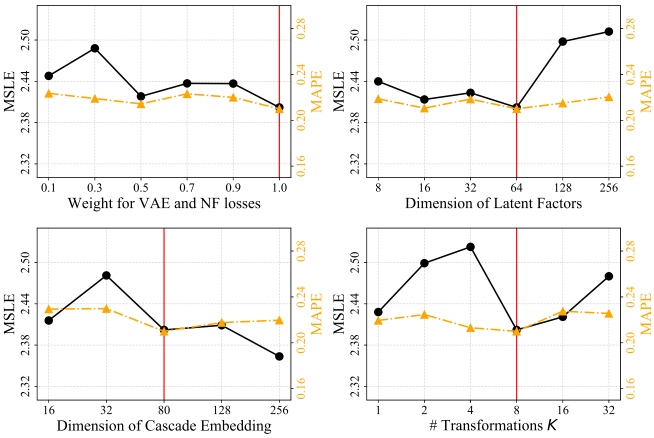


Fig. 6. Impact of four important hyper-parameters of CasFlow on Weibo dataset (observation time is 0.5 hour), measured by MSLE and MAPE. The value of vertical line indicates default parameter settings in experiments.

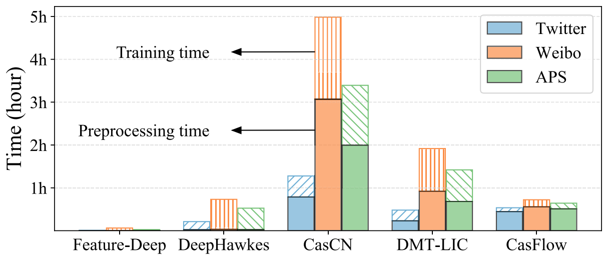


Fig. 7. Time cost of CasFlow on preprocessing & training compared to baselines on Twitter (1 day), Weibo (0.5 hour), and APS (3 years).

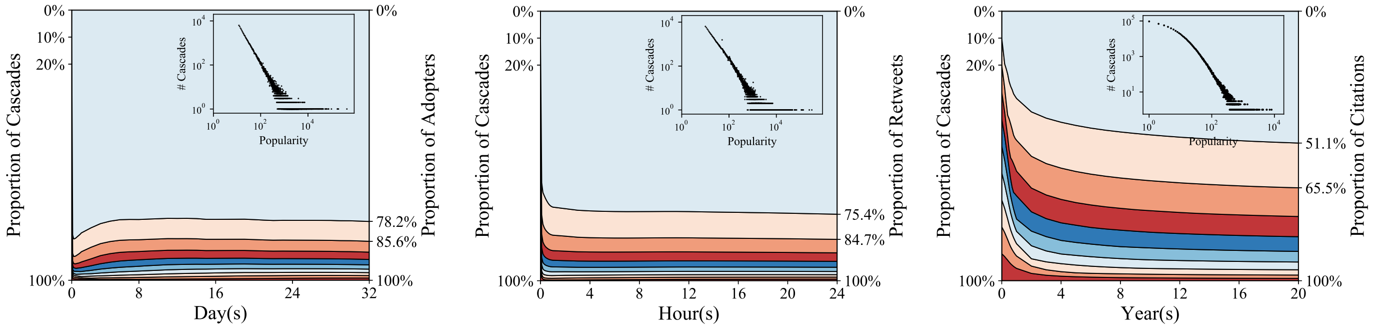


Fig. 8. Stack plot for the proportion of adopters/retweets/citations changed over time on three datasets. Inset: the distribution of cascade popularity is depicted on the upper right corner of each figure, both of which follow power-law distributions.

trades off the supervised learning w.t.r. cascade popularity and uncertainties during information diffusion.

- *Impact of dimensions of latent factors and cascade embedding*: we change the dimensions of latent factors $d_{\mathbf{z}}$ and cascade embedding $d_c + d_g$ from 8 to 256. We can see that large embedding size sometimes may degrades the performance.
- *Impact of NF transformations*: we tried different values of NF transformations from 1 to 32 and observed that the best performance was obtained when $K = 8$.

Other two datasets have similar observations which are ignored due to the space limitation.

Time Cost. We compute the time cost of preprocessing & training for *CasFlow* and baselines, where CasFlow is more efficient compared to CasCN and DMT-LIC and comparable to DeepHawkes. The results are shown in Fig. 7.

Revisiting Cascade Distribution. Finally, we investigate the cascade distribution of different types of information, which could help understand interpretable factors governing the cascade. Fig. 8 plots the cascade distribution of the three datasets. For Twitter and Weibo, the top 20% popular hashtags/tweets yield more than 80% adopters/retweets soon after posting, which shows a Pareto distribution (i.e., the 80-20 rule). For APS, the situation is similar but with relatively moderate monopoly, e.g., around 50% of the citations come from the 10% most influential papers after 20 years of publishing. Moreover, the decaying trend of different information considerably varies from each other, e.g., the most popular hashtags/tweets dominate the hot topics at the very beginning, while most cited papers are gradually and continuously increasing their influence. Therefore, how to identify the influential information, as well as discrimination time decaying factor for different kinds of information, are interesting topics left as our future work.

6 Conclusion

In this work, we introduced *CasFlow* – the first Bayesian learning-based approach for cascade popularity prediction. It leverages a hierarchical variational information diffusion model to exploit the uncertainties at the node level and the cascade level, and learns the posterior of cascade distribution with variational inference and normalizing flows. Our experimental evaluations on three large-scale real-world datasets demonstrated that CasFlow significantly improves the cascade popularity prediction accuracy, outperforming the state-of-the-art baselines. In addition, CasFlow provides interpretation of its behavior to some extent. Overall, our findings indicate that training and optimizing diffusion-related tasks using deep generative models is a promising direction for future investigation.

As part of our future work, we plan to extend CasFlow to incorporate other features – e.g., fusing multiple content features such as texts and figures of microblogs, titles and abstracts of papers, as well as other individual features including number of followers of bloggers, h-index and historical publications of scholars, etc. More complicated graph learning settings can also be considered, e.g., heterogeneous information networks (multiple node/edge types associated) and dynamical graph neural networks (learning representation of evolving graphs). Other forms of variational inference and normalizing flows [88] can be incorporated into CasFlow to learn high-level latent variables and rich families of posterior distribution. CasFlow can be generalized to many business-related contexts, in particular in problem domains such as effective advertising and interpretation of viral information spreading (e.g., rumors, fake news, and epidemic) in network settings.

References

[1] Y. Dong, R. A. Johnson, and N. V. Chawla, “Will this paper increase your H-index?: Scientific impact prediction,” in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 149–158.

[2] M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrishnan, “DeepDiffuse: Predicting the ’Who’ and ’When’ in cascades,” in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 1055–1060.

[3] S. Mishra, M.-A. Rizoiu, and L. Xie, “Modeling popularity in asynchronous social media streams with recurrent neural networks,” in *Proc. 12th Int. AAAI Conf. Web Soc. Media*, 2018, pp. 201–210.

[4] G. Li, S. Chen, J. Feng, K.-L. Tan, and W.-S. Li, “Efficient location-aware influence maximization,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 87–98.

[5] D. M. Lazer *et al.*, “The science of fake news,” *Sci.*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[6] N. Ta, K. Li, Y. Yang, F. Jiao, Z. Tang, and G. Li, “Evaluating public anxiety for topic-based communities in social networks,” *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 27, 2020, doi: 10.1109/TKDE.2020.2989759.

[7] F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, “A survey of information cascade analysis: Models, predictions, and recent advances,” *ACM Comput. Surv.*, vol. 54, no. 2, 2021, Art. no. 27.

[8] S. Mishra, M.-A. Rizoiu, and L. Xie, “Feature driven and point process approaches for popularity prediction,” in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1069–1078.

[9] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, “Can cascades be predicted?,” in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 925–936.

[10] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, "SEISMIC: A self-exciting point process model for predicting tweet popularity," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1513–1522.

[11] M.-A. Rizoiu, S. Mishra, Q. Kong, M. Carman, and L. Xie, "SIR-Hawkes: Linking epidemic models and hawkes processes to model diffusions in finite populations," in *Proc. World Wide Web Conf.*, 2018, pp. 419–428.

[12] Q. Cao, H. Shen, K. Cen, W. Ouyang, and X. Cheng, "DeepHawkes: Bridging the gap between prediction and understanding of information cascades," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1149–1158.

[13] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, "Cascade dynamics modeling with attention-based recurrent neural network," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2985–2991.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop Deep Learn.*, 2014.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[17] C. Li, J. Ma, X. Guo, and Q. Mei, "DeepCas: An end-to-end predictor of information cascades," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 577–586.

[18] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, and F. Zhang, "Information diffusion prediction via recurrent cascades convolution," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 770–781.

[19] F. Zhou, X. Xu, K. Zhang, G. Trajcevski, and T. Zhong, "Variational information diffusion for probabilistic cascades prediction," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1618–1627.

[20] S. Petrovic, M. Osborne, and V. Lavrenko, "Rt to win! predicting message propagation in Twitter," in *Proc. 5th Int. AAAI Conf. Web Soc. Media*, 2011, pp. 586–589.

[21] R. Bandari, S. Asur, and B. A. Huberman, "The pulse of news in social media: Forecasting popularity," in *Proc. 6th Int. AAAI Conf. Web Soc. Media*, 2012, pp. 26–33.

[22] O. Tsur and A. Rappoport, "What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 643–652.

[23] P. Cui, S. Jin, L. Yu, F. Wang, W. Zhu, and S. Yang, "Cascading outbreak prediction in networks: A data-driven approach," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 901–909.

[24] Z. Ma, A. Sun, and G. Cong, "On predicting the popularity of newly emerging hashtags in Twitter," *J. Assoc. Inf. Sci. Technol.*, vol. 64, no. 7, pp. 1399–1410, 2013.

[25] A. Khosla, A. D. Sarma, and R. Hamid, "What makes an image popular?," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 867–876.

[26] F. Gelli, T. Uricchio, M. Bertini, A. Del Bimbo, and S.-F. Chang, "Image popularity prediction in social media using sentiment and context features," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 907–910.

[27] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Commun. ACM*, vol. 53, no. 8, pp. 80–88, Aug. 2010.

[28] S. Jamali and H. Rangwala, "Digging digg: Comment mining, popularity prediction, and social network analysis," in *Proc. Int. Conf. Web Inf. Syst. Mining*, 2009, pp. 32–38.

[29] P. Bao, H. Shen, J. Huang, and X. Cheng, "Popularity prediction in microblogging network: A case study on Sina Weibo," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 177–178.

[30] L. Weng, F. Menczer, and Y.-Y. Ahn, "Predicting successful memes using network and community structure," in *Proc. 8th Int. AAAI Conf. Web Soc. Media*, 2014, pp. 535–544.

[31] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of YouTube videos," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 365–374.

[32] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on Twitter," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 65–74.

[33] R. Crane and D. Sornette, "Robust dynamic classes revealed by measuring the response function of a social system," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 41, pp. 15 649–15 653, 2008.

[34] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: Model and implications," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 6–14.

[35] D. Wang, C. Song, and A.-L. Barabási, "Quantifying long-term scientific impact," *Sci.*, vol. 342, no. 6154, pp. 127–132, 2013.

[36] H. Shen, D. Wang, C. Song, and A.-L. Barabási, "Modeling and predicting popularity dynamics via reinforced poisson processes," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 291–297.

[37] S.-H. Yang and H. Zha, "Mixture of mutually exciting processes for viral diffusion," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1–9.

[38] M.-A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. Van Hentenryck, "Expecting to be HIP: Hawkes intensity processes for social media popularity," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 735–744.

[39] D. Hunter, P. Smyth, D. Q. Vu, and A. U. Asuncion, "Dynamic egocentric models for citation networks," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 857–864.

[40] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Modeling information propagation with survival theory," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1–9.

[41] L. Yu, P. Cui, F. Wang, C. Song, and S. Yang, "From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 559–568.

[42] X. Gao, Z. Cao, S. Li, B. Yao, G. Chen, and S. Tang, "Taxonomy and evaluation for microblog popularity prediction," *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 2, pp. 1–40, 2019.

[43] B. Wu, W.-H. Cheng, Y. Zhang, Q. Huang, J. Li, and T. Mei, "Sequential prediction of social media popularity with deep temporal context networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3062–3068.

[44] W. Zhang, W. Wang, J. Wang, and H. Zha, "User-guided hierarchical attention network for multi-modal social image popularity prediction," in *Proc. World Wide Web Conf.*, 2018, pp. 1277–1286.

[45] J. Wang, V. W. Zheng, Z. Liu, and K. C.-C. Chang, "Topological recurrent neural network for diffusion prediction," in *Proc. IEEE Int. Conf. Data Mining*, 2017, pp. 475–484.

[46] C. Yang, J. Tang, M. Sun, G. Cui, and Z. Liu, "Multi-scale information diffusion prediction with reinforced recurrent networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4033–4039.

[47] D. Liao, J. Xu, G. Li, W. Huang, W. Liu, and J. Li, "Popularity prediction on online articles with deep fusion of temporal process and content features," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 200–207.

[48] X. Chen, K. Zhang, F. Zhou, G. Trajcevski, T. Zhong, and F. Zhang, "Information cascades modeling via deep multi-task learning," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 885–888.

[49] Q. Cao, H. Shen, J. Gao, B. Wei, and X. Cheng, "Popularity prediction on social platforms with coupled graph neural networks," in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 70–78.

[50] B. Shulman, A. Sharma, and D. Cosley, "Predictability of popularity: Gaps between prediction and understanding," in *Proc. 10th Int. AAAI Conf. Web Soc. Media*, 2016, pp. 348–357.

[51] Z. T. Kefato, N. Sheikh, L. Bahri, A. Soliman, A. Montresor, and S. Girdzijauskas, "CAS2VEC: Network-agnostic cascade prediction in online social networks," in *Proc. Int. Conf. Soc. Netw. Anal. Manage. Secur.*, 2018, pp. 72–79.

[52] C. Gou, H. Shen, P. Du, D. Wu, Y. Liu, and X. Cheng, "Learning sequential features for cascade outbreak prediction," *Knowl. Inf. Syst.*, vol. 57, pp. 721–739, 2018.

[53] M. Jenders, G. Kasneci, and F. Naumann, "Analyzing and predicting viral tweets," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 657–664.

[54] M. Tsagkias, W. Weerkamp, and M. De Rijke, "Predicting the volume of comments on online news stories," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1765–1768.

[55] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi, "Bad news travel fast: A content-based analysis of interestingness on Twitter," in *Proc. 3rd Int. Web Sci. Conf.*, 2011, pp. 1–8.

[56] F. Chen *et al.*, "Marked self-exciting point process modelling of information diffusion on Twitter," *Ann. Appl. Statist.*, vol. 12, no. 4, pp. 2175–2196, 2018.

[57] R. Kobayashi and R. Lambiotte, "TiDeH: Time-dependent Hawkes process for predicting retweet dynamics," in *Proc. 10th Int. AAAI Conf. Web Soc. Media*, 2016, pp. 191–200.

[58] L. Hong, O. Dan, and B. D. Davison, "Predicting popular messages in Twitter," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 57–58.

[59] A. Kupavskii, A. Umnov, G. Gusev, and P. Serdyukov, "Predicting the audience size of a tweet," in *Proc. 7th Int. AAAI Conf. Web Soc. Media*, 2013, pp. 693–696.

[60] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.

[61] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[62] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1320–1329.

[63] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[64] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[65] E. Lukacs, *Characteristic Functions*. London, UK: Griffin, 1970.

[66] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 385–394.

[67] J. Zhang, Y. Dong, Y. Wang, J. Tang, and M. Ding, "ProNE: Fast and scalable network representation learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4278–4284.

[68] T. Tao, *Topics in Random Matrix Theory*, vol. 132. Providence, RI, USA: American Mathematical Society, 2012.

[69] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[70] S. Lamprier, "A recurrent neural cascade-based model for continuous-time diffusion," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 1–10.

[71] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Representations*, 2014.

[72] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3358–3364.

[73] D. Zhu, P. Cui, D. Wang, and W. Zhu, "Deep variational network embedding in Wasserstein space," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2827–2836.

[74] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, Z. Ting, and F. Zhang, "Predicting human mobility via variational attention," in *Proc. World Wide Web Conf.*, 2019, pp. 2750–2756.

[75] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. World Wide Web Conf.*, 2018, pp. 689–698.

[76] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.

[77] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1530–1538.

[78] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," in *Proc. Int. Conf. Learn. Representations*, 2017.

[79] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4743–4751.

[80] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. Workshops*, 2011, pp. 1–8.

[81] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3837–3845.

[82] L. Weng, F. Menczer, and Y.-Y. Ahn, "Virality prediction and community structure in social networks," *Sci. Rep.*, vol. 3, 2013, Art. no. 2522.

[83] S. Goel, A. Anderson, J. Hofman, and D. J. Watts, "The structural virality of online diffusion," *Manage. Sci.*, vol. 62, no. 1, pp. 180–196, 2015.

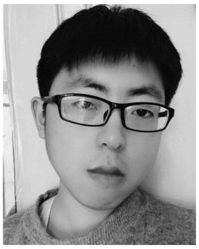
[84] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "DeepInf: Social influence prediction with deep learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2110–2119.

[85] C. Yang, J. Tang, M. Sun, G. Cui, and Z. Liu, "Multi-scale information diffusion prediction with reinforced recurrent networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4033–4039.

[86] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[87] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, 2014.

[88] E. Hajiramezanali, A. Hasanzadeh, K. Narayanan, N. Duffield, M. Zhou, and X. Qian, "Variational graph recurrent neural networks," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2019, pp. 10 700–10 710.



Xovee Xu (Graduate Student Member, IEEE) received the BS and MS degrees in software engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2018 and 2021, respectively. He is currently working toward the PhD degree in computer science with the University of Electronic Science and Technology of China, China. He is the author of several research articles published in INFOCOM, SIGIR, *IEEE Transactions on Knowledge and Data Engineering*, AAAI and *ACM Computing Surveys*. His recent research interests include social network data mining and knowledge discovery, primarily focuses on information diffusion in full-scale graphs, human behavior understanding, spatial-temporal data modeling, representation learning, and their novel applications in various social and scientific scenarios.

Fan Zhou received the BS degree in computer science from Sichuan University, China, in 2003, and the MS and PhD degrees from the University of Electronic Science and Technology of China, China, in 2006 and 2012, respectively, where he is currently an associate professor with the School of Information and Software Engineering. His research interests include machine learning, neural networks, spatio-temporal data management, graph learning, recommender systems, and social network data mining and knowledge discovery.

Kunpeng Zhang received the PhD degree in computer science from Northwestern University, Evanston, Illinois. He is currently an assistant professor with Robert H. Smith School of Business, University of Maryland, College Park, Maryland. He is interested in large-scale data analysis, with particular focuses on social data mining, image understanding via machine learning, social network analysis, and causal inference. He has published papers in the area of social media, artificial intelligence, network analysis, and information systems on various conferences and journals.

Siyuan Liu received the one PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, and another PhD degree from the University of Chinese Academy of Sciences, China. He is assistant professor of information systems with Smeal College of Business, Pennsylvania State University, State College, Pennsylvania. His research interests include spatial and temporal data mining, social networks analytics, and data-driven behavior analytics.

Goce Trajcevski received the BSc degree from the University of Sts. Kiril i Metodij, North Macedonia, and the MS and PhD degrees from the University of Illinois at Chicago, Chicago, Illinois. He is currently an associate professor with the Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa. His main research interests are in the areas of spatio-temporal data management, uncertainty and reactive behavior management in different application settings, and incorporating multiple contexts. In addition to three book chapter and three encyclopedia chapters, he has coauthored more than 170 publications in refereed conferences and journals. His research has been funded by the NSF, ONR, BEA, and Northrop Grumman Corp. He was the general co-chair of the IEEE ICDE 2014, ACM SIGSPATIAL 2019, the PC co-chair of the ADBIS 2018 and ACM SIGSPATIAL 2016 and 2017, and has served in various roles in organizing committees in numerous conferences and workshops. He is an associate editor of the *ACM Transactions on Spatial Algorithms and Systems* and the *Geoinformatica Journals*.