Variational Information Diffusion for Probabilistic Cascades Prediction

Fan Zhou*, Xovee Xu*, Kunpeng Zhang†, Goce Trajcevski‡ and Ting Zhong*§
*School of Information and Software Engineering
University of Electronic Science and Technology of China, Chengdu, China
†Robert H. Smith School of Business, University of Maryland, College Park, USA
‡Department of Electrical and Computer Engineering, Iowa State University, USA
§Corresponding author: zhongting@uestc.edu.cn

***Abstract*—Understanding in-network information diffusion is a fundamental problem in many application domains and one of the primary challenges is to predict the size of the information cascade. Most of the existing models rely either on hypothesized point process (e.g., Poisson and Hawkes process), or simply predict the information propagation via deep neural networks. However, they fail to simultaneously capture the underlying structure of a cascade graph and the propagation of 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: Variational Cascade (VaCas) graph learning networks. VaCas 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 variational inference to learn the node-level and cascade-level latent factors in an unsupervised manner. In addition, VaCas 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 VaCas significantly improves the prediction accuracy, compared to state-of-the-art approaches, while also enabling interpretability.

***Index Terms*—Information cascades, variational autoencoder, graph learning**

I. Introduction

Online social platforms such as Twitter, Weibo, WeChat, Reddit, and Facebook have become the main source of information in people's daily life. Various news, events, and posts are disseminated as cascades spread by users through social networks [1], [2], [3]. As a result, predicting the size of (potentially) affected users after a certain time-period becomes important 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 [1], [2], [4], [5], [6], [7], [8].

In recent years, a series of works have been focusing on this area, 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-based approaches*: Researchers in [9], [10], [11] focus on identifying and incorporating hand-crafted features for cascade prediction. They 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) *Pattern-based approaches*: In [8], [12], [13], [14], 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 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 [2], [13], [15], [16], [17], researchers leverage various deep learning techniques and develop models for capturing the temporal and sequential process of information diffusion, where recurrent neural networks (RNNs) such as LSTM [18], GRU [19], and graph neural networks [20] are usually used for modeling the sequential patterns [15], [17] and graph structures [2], 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 improvement on modeling cascades diffusion, 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) *Incomplete network structure* inherent in cascades impedes the approaches requiring global network information, as obtaining or further embedding a complete graph is often 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.
(4) *Lack of hierarchical cascade modeling* – it makes the existing methods either focus on roughly estimating 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 and cascade-level representation.

(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 [11]) – which is not taken into account in the existing methods.

Our Approach: To address the aforementioned challenges, we present **VaCas** (Variational Cascade) graph learning neural networks – a novel framework integrating the hierarchical diffusion modeling and temporal characteristics of cascades for predicting the popularity of a post. Specifically, VaCas applies graph wavelets to learn the local cascade representation which, in turn, allows varying-size diffusion graph learning and avoids the computationally intensive global network structural embedding. We further employ a novel contextualized diffusion embedding module to learn the 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. To understand both user-level behavior and cascade-level diffusion effect, VaCas introduces a hierarchical variational autoencoder for simultaneously learning fine-grained and structural patterns of information diffusion with probabilistic latent variables. By incorporating amortized variational inference into the generative model with latent variables, VaCas exposes an interpretable representation of the complex distribution and long-term dependencies among nodes in a cascade, thereby incorporating the uncertainty of each retweeting 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 evolving graph embedding and jointly models cascades from both a micro (user) and a macro (overall cascade estimating) level.
- **Diffusion uncertainty model**: VaCas model leverages variational autoencoder for embedding both sub-graphs and the complete cascade as Gaussian 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 GRU module into cascade graph learning, VaCas is able to capture users' different behavior on different information, rather than binary prediction on users' retweeting behavior. 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 real-world datasets, demonstrating that VaCas improves the cascade size prediction performance compared to the state-of-the-art approaches, and also provides explanations on its behavior.

II. 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 study the factors affecting content popularity, including content-related features such as the number of hashtags or mentions [9] and user-related features such as user profile, user attributes and historical activities [21], [22], cascades structural [23], [24], and temporal features [25], [26]. The popularity is predicted via various machine learning models. For example, multiple studies [23], [24], [25], [26] confirmed that user features are informative predictors, especially the features related to early-forwarding users [27]. 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.

Pattern-based approaches exploit the patterns in the sequence of posting/retweeting (a.k.a. event) time and model their arrival process. Generally, the cascade is treated as time-series data, and the model – i.e., learning parameters – is built by maximizing the probability of an event occurring within an observation time window [2]. Different point processes (e.g., Poisson [28], [29], Hawkes [12], [30], [31], [32], etc.), and models (e.g., Cox [33], [34], Weibull [35], and epidemic model [36]) have been used. Despite demonstrating an enhanced prediction accuracy, the methods are unable to fully leverage the implicit information in the cascade dynamics. As shown in a recent review [37]: Poisson process is too simple to capture the propagation patterns and Weibull model tends to overestimate the cascade size, while Hawkes usually overestimates the popularity, probably due to its rudimentary self-excitation mechanism. In contrast, VaCas enables integration of structural and temporal information.

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. One of the pioneers – DeepCas [15] – is a structure-based popularity prediction model learning the representation of cascade graphs in an end-to-end manner. Subsequently, DeepHawkes [13] 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: DFTC [5], CYAN-RNN [16], Topo-LSTM [17], and SNIDSA [38] – 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 VaCas, these approaches usually require a complete graph for cascade representation learning, which is not always available in real world applications.

Motivated by graph neural networks (GNNs) [20], a re-

current cascade convolution model was developed in [2]. It learns the structures 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 [39] 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. 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.

III. Preliminaries

We now introduce the necessary background and formally define the cascade popularity prediction problem. Let C_i denote a content of interest which, starting at some time-instant, is propagated through a network. In the rest of this work, we consider tweet cascades as example-settings, however, our work is directly applicable to other types of information diffusion (e.g., academic publications, news articles, etc.).

Definition 1: Cascade Graph – Given a (portion of a) tweet C_i, its cascade graph $\mathbf{G}_i$ is an evolving sequence of N sub-graphs denoted as $\mathbf{G}_i = \{\mathcal{G}_i(t_0), \mathcal{G}_i(t_1), \ldots, \mathcal{G}_i(t_{N-1})\}$, where $\mathcal{G}_i(t_j) = (\mathcal{V}_i^{t_j}, \mathcal{E}_i^{t_j}, t_j)$ is a snapshot of the cascade graph $\mathbf{G}_i$ at time t_j; and $\mathcal{V}_i^{t_j}$ and $\mathcal{E}_i^{t_j}$ are the sets of nodes and edges of the graph $\mathcal{G}_i(t_j)$ at time $t_j \geq 0$, respectively.

Let the node $v_i^{t_j}$ denote the user who retweets C_i at time t_j and define $\mathcal{V}_i^{t_j} = \{v_i^{t_0}, v_i^{t_1}, \ldots, v_i^{t_j}\}$, and let the set of edges $\mathcal{E}_i^{t_j}$ represent the retweeting relationships between users in $\mathcal{V}_i^{t_j}$. A toy example of a cascade graph is illustrated in Fig. 1, where the evolving information cascade $\mathbf{G}_i$ can be represented as $\mathbf{G}_i = \{\mathcal{G}_i(t_0), \mathcal{G}_i(t_1) \ldots, \mathcal{G}_i(t_{N-1})\}$, where each $\mathcal{G}_i(t_j)$ reflects the diffusion trend of a tweet C_i at time t_j.

Existing efforts make different cascade predictions in a rather similar way. Some works [23], [40], [41] treat cascade prediction as a classification problem – e.g., predicting whether a cascade can break out a specific threshold [21], [42]; whether a cascade can double its size [11] at the end; or predict the range that a cascade would mostly like to fall into [5], [43]. Similarly to many previous works [2], [13], we also make numerical predictions – i.e., we predict the exact *incremental popularity* P_i (cf. Def. 2 below) for a cascade C_i. However, rather than observing a fixed number of retweets [11], 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. Thus given an observation time window $T = [t_0, t_o]$, we have the partial cascade graph $\mathbf{G}_i = \{\mathcal{G}_i(t_0), \mathcal{G}_i(t_1) \ldots, \mathcal{G}_i(t_o)\}$, from which we predict the popularity.

Definition 2: Popularity Prediction – Given a tweet C_i and its partial cascade graph $\mathbf{G}_i$, the *incremental popularity* P_i is defined as $P_i = |\mathcal{V}_i^{t_r}| - |\mathcal{V}_i^{t_o}|$, where t_o and t_r are the observation time and the end time, respectively; and $|\mathcal{V}_i^*|$ denotes the size of the cascade graph, in terms of the number of nodes that retweeted C_i.

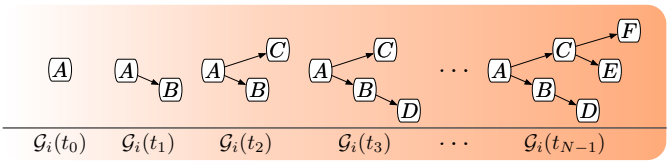


Fig. 1. An example of evolving cascade graph $\mathbf{G}_i$ for a specific tweet C_i

Thus, our main objective is to learn a regression function $f : C_i \rightarrow P_i$ that maps cascade C_i to its incremental popularity P_i. To enable incorporation of the temporal dimension, we need the notion of an *edge weight*. The edge weight is determined by retweeting time: if node v_j forwards v_i's tweet, the weight of edge (v_i, v_j) between nodes v_j and v_i is defined as $W_{v_i,v_j} = (t_j - t_o)/t_o$, $(t_j \geq t_o > 0)$, $W_{v_i,v_j} = 0$ otherwise. Note that all edge weights are therefore normalized into the range $[0, 1]$.

IV. Methodology

In this section, we describe the details of VaCas, which considers the structural (cascade graph) and temporal (forwarding time) information to make cascade popularity prediction. As illustrated in Fig. 2, VaCas consists of four main components:
(A) Contextualized diffusion embeddings: VaCas utilizes the techniques from graph signal processing [44], [45], [46] to generate nodes' structural embeddings from spectral graph wavelets;
(B) Temporal diffusion modeling: VaCas leverages bidirectional GRUs to model the temporal dependencies of information diffusion;
(C) Diffusion uncertainty modeling: VaCas models the uncertainty in information diffusion and cascade size growth through a hierarchical variational autoencoder;
(D) Predictor: combined with recurrent neural networks and hierarchical variational autoencoder, the learned cascade representation are fed into multi-layer perceptrons (MLPs) to make the final popularity prediction.

A. Contextualized User Behavior Learning

To capture the 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. [44]). We note that other graph representation techniques may be used [20], [47], [48], [49].

Given a post/tweet C_i and its observed cascade graph $\mathcal{G}_i(t_o)$, its weighted adjacency matrix $\mathbf{A}$ can be straightforwardly determined. The diagonal degree matrix $\mathbf{D}$ can be computed as each of the diagonal elements is equal to the sum of weights of all edges connected to that node, say $v_i^{t_j}$. We therefore have an unnormalized graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is the eigenvalue decomposition and $\mathbf{\Lambda} = \text{Diag}(\lambda_0, \ldots, \lambda_{N-1})$ is the diagonal matrix of the eigenvalues satisfying $\lambda_0 < \lambda_1 \leq \cdots \leq \lambda_{N-1}$. We can

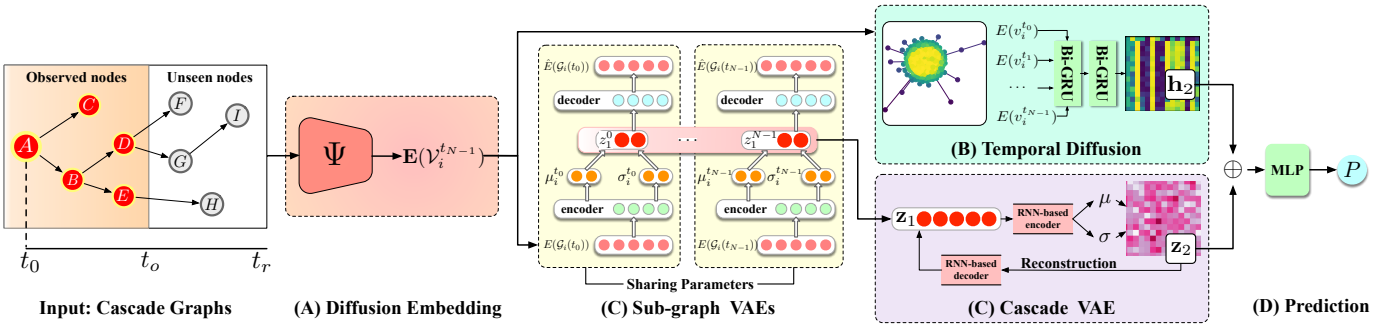


Fig. 2. An overview of our proposed model VaCas. t_0: the first time C_i occurs; t_o: observation time; t_r: the ending time.

now calculate spectral graph wavelets $\Psi_{v,s}$ for each node $v_i^{t_j} \in \mathcal{V}_i^{t_{N-1}}$ as:

$$\Psi_{v,s} = \mathbf{U}\text{Diag}(g_s(\lambda_0), \ldots, g_s(\lambda_{N-1}))\mathbf{U}^T\delta_v, \quad (1)$$

where δ_v is the node v's one-hot encoding vector, and the filter kernel function g_s is continuous, 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 $(\lambda_l)_{l=0,\ldots,N-1}$.

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 [46]. 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 [44]. For a given node $v_i^{t_j}$, we treat its wavelet coefficients as a probability distribution and then utilize empirical characteristic functions [50] 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 $v_i^{t_j}$ and scale parameter s, the empirical characteristic function is defined as:

$$\phi_{v,s}(p) = \frac{1}{N}\sum_{m=0}^{N-1} e^{ip\Psi_{m,v,s}}, \quad (2)$$

where $\Psi_{m,v,s} = \sum_{l=0}^{N-1} g_s(\lambda_l)U_{ml}U_{vl}$ is the m-th wavelet coefficient of $\Psi_{v,s}$. Then the embedding of node $v_i^{t_j}$ can be obtained by concatenating values of the real part and imaginary part: $E(v_i^{t_j}) = [\text{Re}(\phi_{v,s}(p)), \text{Im}(\phi_{v,s}(p))]_{p_1,p_2,\ldots,p_d}$ on s. The first element of node's embedding $E(v_i^{t_j})$ is set to node's edge weight $W_{v_i^{t_{j-1}}, v_i^{t_j}}$ and the dimensionality of the embedding is $2d$. 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., we can focus on the individual node embedding rather than embedding the entire graphs [48], [49] or emphasizing particular tasks [20].

B. Temporal Diffusion Learning

The embeddings generated from above spectral graph wavelets represent the structural information that nodes carry in the cascade graph. Specifically, structurally equivalent nodes will have similar embeddings (cf. [44]). The nodes playing similar structural roles may make similar contributions to the growth of cascade popularity – e.g., hub nodes are more influential than leaf nodes to propagate information. However, the temporal patterns encoded in the diffusion process are also important and have critical impact on the cascade prediction. To capture such temporal information, we use a bidirectional Gated Recurrent Unit (Bi-GRU) [19] to model the node behavior in the cascades. RNNs are a natural choice and have been widely used in the literature – e.g., [2], [15], [17], [51] have used LSTM for modeling the sequential patterns during the information diffusion.

For a given cascade of C_i, its N node embeddings $\mathbf{E}(\mathcal{V}_i^{t_{N-1}}) = \{E(v_i^{t_0}), E(v_i^{t_1}), \ldots, E(v_i^{t_{N-1}})\}$, would be sequentially fed through the two layers of Bi-GRU to generate context-dependent representation, as shown in the bottom right of Fig. 2. For each input $E(v_i^{t_j})$, GRU computes the updated hidden state with gated units. By concatenating the outputs of the forward $\overrightarrow{\text{GRU}}$ and backward $\overleftarrow{\text{GRU}}$, the final representation $\mathbf{h}_2$ of a cascade is obtained as:

$$\begin{cases} \mathbf{h}_1 = [\overrightarrow{\text{GRU}}(\mathbf{E}(\mathcal{V}_i^{t_{N-1}})), \overleftarrow{\text{GRU}}(\mathbf{E}(\mathcal{V}_i^{t_{N-1}}))]; \\ \mathbf{h}_2 = [\overrightarrow{\text{GRU}}(\mathbf{h}_1), \overleftarrow{\text{GRU}}(\mathbf{h}_1)]. \end{cases} \quad (3)$$

The output of the last hidden $\mathbf{h}_2$ can be readily used for predicting the cascade size, as done in many previous works [2], [15], [17], [51] – except that they use one-layer LSTM or GRU. We call this model as Cas-RNN. However, there is a drawback when using only the last hidden state of a 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 – obviously, the larger ones will have more and longer sub-cascades.

C. 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 hierarchical variational autoencoders [52] to model the diffusion uncertainty in both user-level behavior and cascade-level diffusion, as illustrated in Fig. 2. VAE model [52] 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 [53], node representation [54], prediction [55] and recommendation [56]. As a probabilistic approach, VAE provides a solid mathematical tool for coping with the randomness and uncertainty, which motivates us to model the cascade uncertainty with such a Bayesian framework.

Sub-graph (lower) level uncertainty modeling: A cascade $\mathbf{G}_i$ is composed of an evolving sequence of sub-graphs $\{\mathcal{G}_i(t_0), \ldots, \mathcal{G}_i(t_{N-1})\}$, each representing an observation of information diffusion. However, we now only have node representation leaned from the graph wavelets. To obtain the distribution on these sub-graphs, we need to learn the representation of each sub-graph. Here, we adopt a simple method by averaging the representation of all nodes: $E(\mathcal{G}_i(t_j)) = \frac{1}{N}\sum_{m=1}^{|\mathcal{V}_i^{t_j}|} E(v_i^{t_m})$ where $v_i^{t_m} \in \mathcal{V}_i^{t_j}$ and $E(\mathcal{G}_i(t_j)) \in \mathbb{R}^{2d}$ respectively denote the nodes and embeddings of sub-graphs in a cascade.

For each sub-graph $\mathcal{G}_i(t_j)$, we can obtain its mean μ_i and variance σ_i from the data. The latent factor $\mathbf{z}_1 = \{z_1^0, \ldots, z_1^i, \ldots, z_1^{N-1}\}$ is then calculated through the reparameterization trick [52]: $z_1^i = \mu_i + \sigma_i * \epsilon -$ where ϵ are the samples from the Gaussian $\epsilon \sim \mathcal{N}(0, I)$. For a specific snapshot of the cascade $\mathcal{G}_i(t_j)$, the marginal log-likelihood of this sub-graph embedding $E(\mathcal{G}_i(t_j))$ is $\log p_\theta(E(\mathcal{G}_i(t_j))) = \log \int_{\mathbf{z}_1} p_\theta(E(\mathcal{G}_i(t_j))|\mathbf{z}_1)p(\mathbf{z}_1)\mathrm{d}\mathbf{z}_1$, which is intractable to compute or differentiate directly for flexible generative models, especially for high dimensional latent variables. Instead, one usually resorts to variational inference by defining a simple parametric distribution over the latent variables (e.g., a factorized Gaussian) $q_\phi(\mathbf{z}_1|E(\mathcal{G}_i(t_j)))$, and maximizing the evidence lower bound (ELBO) on the marginal log-likelihood of each observation (for brevity, we denote each sub-graph embedding $E(\mathcal{G}_i(t_j))$ as $\mathcal{G}_i$):

$$\begin{aligned}
&\log p_\theta(\mathcal{G}_i) \\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathcal{G}_i)} \log \left[\frac{p_\theta(\mathcal{G}_i, \mathbf{z}_1)}{q_\phi(\mathbf{z}_1|\mathcal{G}_i)}\right] + \mathbb{KL}\left[q_\phi(\mathbf{z}_1|\mathcal{G}_i)||p_\theta(\mathbf{z}_1|\mathcal{G}_i)\right] \\
\geq&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathcal{G}_i)}\left[\log p_\theta(\mathcal{G}_i, \mathbf{z}_1) - \log q_\phi(\mathbf{z}_1|\mathcal{G}_i)\right] \\
\triangleq&\text{ELBO}(\mathcal{G}_i; \theta, \phi),
\end{aligned} \tag{4}$$

where $q_\phi(\mathbf{z}_1|\mathcal{G}_i)$ (a.k.a. *encoder* parameterized by ϕ) is an approximation to the true posterior $p_\theta(\mathbf{z}_1|\mathcal{G}_i)$ used to generate the latent variables $\mathbf{z}_1$; and $\mathbb{KL}[\cdot]$ is the Kullback-Leibler divergence. Since the objective is to minimize the KL divergence between the proposed $q_\phi(\mathbf{z}_1|\mathcal{G}_i)$ and $p_\theta(\mathbf{z}_1|\mathcal{G}_i)$, we can alternatively maximize ELBO of $\log p_\theta(\mathcal{G}_i, \mathbf{z}_1)$ w.r.t. both parameters θ and ϕ, which are jointly trained with separate nonlinear functions such as neural networks.

By minimizing the reconstruction error between the input $E(\mathcal{G}_i(t_j))$ and output $\hat{E}(\mathcal{G}_i(t_j))$, the learned latent representation $\mathbf{z}_1$ for all sub-graphs captures the data distribution and can be readily used to generate synthetic data or improves particular tasks [53], [56]. Now, we can immediately combine $\mathbf{z}_1$ with the last hidden state $\mathbf{h}_2$ from the Bi-GRU for predicting the final cascade size. We call this variant as *VaCas**, which can be considered as a one-layer VaCas with pre-trained sub-graphs. However, this model only captures the individual sub-graph 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 sub-graphs.

Cascade (higher) level variational inference: To overcome the "shallow" generation problem in VaCas* as mentioned above, we add another RNN layer with a sequential variational autoencoder, as shown in Fig. 2. This higher level (cascade) VAE takes N sequential latent variables $\mathbf{z}_1 = \{z_1^0, z_1^1, \ldots, z_1^{N-1}\}$ generated from the lower level VAE as input, with each z_1^j corresponding to a sub-graph $\mathcal{G}_i(t_j)$, 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 the cascade's evolving trajectory. More specifically, the joint probability for a cascade $\mathbf{G}_i$ is formulated as:

$$p_\theta(\mathbf{G}_i, \mathbf{z}_1, \mathbf{z}_2) = p_\theta(\mathbf{z}_1)p_\theta(\mathbf{z}_2|\mathbf{z}_1)p_\theta(\mathbf{G}_i|\mathbf{z}_1, \mathbf{z}_2) \tag{5}$$

where sub-graph latent factors $\mathbf{z}_1$ are centered isotropic multivariate Gaussian distributions. The conditional distribution $p(\mathbf{z}_2|\mathbf{z}_1)$ is parameterized by a RNN encoder and $p(\mathbf{G}_i|\mathbf{z}_1, \mathbf{z}_2)$ can be considered as cascade reconstruction from the latent factors, formulated as:

$$p_\theta(\mathbf{z}_2|\mathbf{z}_1) = \sum_{k=1}^{K} \mathcal{N}(\mathbf{z}_2|f_\vartheta^\mu(\mathbf{z}_1), \text{diag}(f_\vartheta^{\sigma^2}(\mathbf{z}_1))), \tag{6}$$

$$p_\theta(\mathbf{G}_i|\mathbf{z}_1, \mathbf{z}_2) = \mathcal{N}(\mathbf{G}_i|f_\kappa^\mu(\mathbf{z}_1, \mathbf{z}_2), \text{diag}(f_\kappa^{\sigma^2}(\mathbf{z}_1, \mathbf{z}_2))), \tag{7}$$

where the conditional distribution of the observed cascade $\mathbf{G}_i$ 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 κ. The ELBO on the marginal likelihood of the cascades is derived as:

$$\begin{aligned}
&\log p_\theta(\mathbf{G}_i) \geq \text{ELBO}(\mathbf{G}_i;\theta,\phi).\\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{G}_i)} \log\left[\frac{p_\theta(\mathbf{z}_1)p_\theta(\mathbf{z}_2|\mathbf{z}_1)p_\theta(\mathbf{G}_i|\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_2|\mathbf{G}_i,\mathbf{z}_1)q_\phi(\mathbf{z}_1|\mathbf{G}_i)}\right]\\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{G}_i)}[\log p_\theta(\mathbf{G}_i|\mathbf{z}_1,\mathbf{z}_2)+\log p_\theta(\mathbf{z}_2|\mathbf{z}_1)\\
&+\log p_\theta(\mathbf{z}_1)-\log q_\phi(\mathbf{z}_2|\mathbf{G}_i,\mathbf{z}_1)-\log q_\phi(\mathbf{z}_1|\mathbf{G}_i)]\\
=&\mathbb{E}_{\mathbf{z}_1\sim q_\phi(\mathbf{z}_1|\mathbf{G}_i),\mathbf{z}_2\sim q_\phi(\mathbf{z}_2|\mathbf{z}_1)}[\log p_\theta(\mathbf{G}_i|\mathbf{z}_1,\mathbf{z}_2)]\\
&-\mathbb{KL}[q_\phi(\mathbf{z}_2|\mathbf{G}_i,\mathbf{z}_1)||p_\theta(\mathbf{z}_2|\mathbf{z}_1)]\\
&-\mathbb{KL}[q_\phi(\mathbf{z}_1|\mathbf{G}_i)||p_\theta(\mathbf{z}_1)]. \qquad (8)
\end{aligned}$$

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 subgraphs from a set of latent variables $\mathbf{z}_1$ and $\mathbf{z}_2$. The two $\mathbb{KL}$ 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.

D. Prediction

Now we have obtained $\mathbf{h}_2$ from the two-layer Bi-GRUs and $\mathbf{z}_2$ from the hierarchical VAEs, which can be fed into multi-layer perceptrons (MLPs) to output the final cascade prediction:

$$\log_2 \hat{P}_i = \text{MLP}([\mathbf{h}_2, \mathbf{z}_2]) \qquad (9)$$

by minimizing the mean square error loss function:

$$\mathcal{L}(P_i,\hat{P}_i) = \frac{1}{N}\sum_{i=0}^{N-1}(\log_2 P_i - \log_2 \hat{P}_i)^2 - \text{ELBO}(\mathbf{G}_i;\theta,\phi), \qquad (10)$$

where N denotes the total number of cascades, P_i is the ground truth and $\hat{P}_i$ is the predicted incremental popularity for the cascade graph $\mathbf{G}_i$, and $\text{ELBO}(\mathbf{G}_i;\theta,\phi)$ is the ELBO that needs to be maximized as given by Eq. (8). Note that the loss function of the variant VaCas* is similar to Eq. (10) – except that VaCas* uses Eq. (4) as the ELBO. The overall training process of VaCas is sketched in Algorithm 1.

Algorithm 1 Learning with VaCas.

Input: Cascade graph $\mathbf{G}_i$ and its evolving sequence $\mathcal{G}_i(t)$, scale parameter s, d evenly spaced points $\{p_1, p_2, \ldots, p_d\}$.
Output: Predicted cascade size $\hat{P}_i$.

```
Compute graph wavelets Ψ_{v,s} for each node v_i^{t_j} (Eq. (1));
for ∀Ψ_{v,s} do
    Compute embedding E(V_i^{t_{N-1}}) (Eq. (2));
end for
for j ∈ [0, N) do
    Compute z_1^j by optimizing Eq. (4) for G_i(t_j);
end for
Obtain z_1 = {z_1^0, ··· , z_1^{N-1}} ;
while not convergence do
    Train the Bi-GRU to obtain h_2 (Eq. (3));
    Train cascade VAE to obtain z_2 by optimizing Eq. (8);
    Combine h_2 and z_2 for cascade prediction via Eq. (10).
end while
```

V. Experiments

We now introduce the details of two evaluation datasets and the competitor algorithms, followed by the evaluation of our models against the state-of-the-art baselines on cascade size prediction. Evaluations of model ablation and model interpretability are also presented.

A. Experimental Settings

Dataset: Cascades can be formed by different types of information, e.g., social tweets, emails, news articles, research papers, and so on. We selected two publicly available datasets – Weibo [13] and APS – that have been commonly used in previous related works [2], [13], [29] for evaluating popularity prediction approaches.

TABLE I
Descriptive Statistics of Two Datasets.

Dataset	Weibo	APS
Number of Cascades	119,313	207,685
Number of Nodes	6,738,040	616,316
Number of Edges	455,412,321	247,319,593
Avg. Popularity	240	51
Avg. Observed Popularity	54	19
Avg. Sequence Length	2.237	3.999
Avg. Structural Virality	2.025	3.114

• *Sina Weibo* is the largest microblogging platform in China (https://www.weibo.com), where every tweet and its retweet can form a retweeting cascade. We randomly select 70% (21,294) for training, 15% (4,563) for testing, and the remaining 15% (4,563) for validation.

• *American Physical Society (APS)* contains scientific papers published by APS journals (https://journals.aps.org/datasets). Every paper in the APS dataset and its citations form a citing cascade. Similarly, 70% (32,102) of the data are used for training and the rest for testing (15%) and validation (15%).

Evaluating models on two distinct information cascades provides a systematic view on the generalization capability, without domain knowledge and feature engineering. Descriptive statistics of two datasets are shown in Table I and Fig. 3. We can observe that: – APS has smaller average item cascade than Weibo; – its structural virality (calculated by Wiener index as a measure of structural virality of cascades [11], [57]) is higher than Weibo dataset, indicating that the popularity of scientific papers is mainly driven by the propagation of other papers rather than directly citing the original papers.

In addition, the observation time windows for Weibo and APS are 0.5 hour and 5 years, respectively. Thus, we select 24 hours as the end time t_r for Weibo tweets and 20 years for APS papers, for each cascade, its observed size is no more than 1,000 in observation window, following previous works [2], [13], [39]. Due to the effect of diurnal rhythm

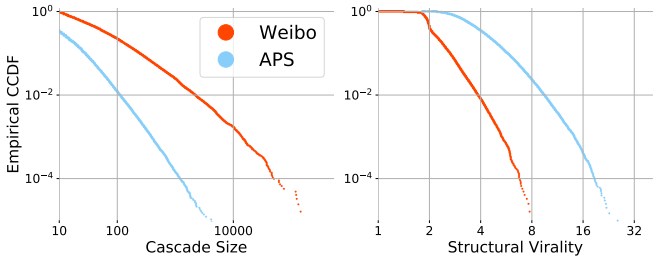


Fig. 3. The empirical complementary cumulative distribution (CCDF) of cascade size and structural virality for two datasets. The popularity of Weibo tweets are typically larger than APS papers, but its structural virality is smaller, while both follow a heavy-tailed distribution.

in the Weibo dataset, we focus on tweets posted between 8:00AM and 6:00PM, leaving each tweet at least 6 hours to reap retweets. As for the APS dataset, we consider papers published between 1893 and 1997 – so that each paper has at least 20 years (1997 - 2017) to gain its citations.

Baselines: For completeness, we compare our method with several state-of-the-art approaches for cascade size prediction covering *feature-based*, *pattern-based* and *deep learning-based approaches*. The baselines include:

• *Feature-based*: Cheng et al. [11] group five classes of factors that drive cascade growth, including content features, original poster/re-sharer features, structural features, and temporal features. Here we use the following ones: observed cascade size, cumulative cascade size, time between original and its first forwarding, mean time between the first half and the last half of forwarding, number of leaf nodes, average node degree, average and max length of sequences. We feed these features into a linear regression model to make predictions.

• *DeepCas*: is the first deep learning architecture for information cascade prediction. It utilizes random walks to sample paths to represent cascade graph (similar to DeepWalk [47]), and uses GRU and attention mechanism for modeling and predicting cascade size in an end-to-end manner [15].

• *Topo-LSTM:* uses LSTM to model relationships among nodes in the graph. The hidden state and cell of each node at a given time depend on those from each of its predecessors that have been infected before that time instant [17].

• *DeepHawkes*: combines both deep learning and self-exciting point 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 [13].

• *CasCN*: is a graph convolution network (GCN)-based framework exploiting both temporal and structural information for cascade prediction. It samples a cascade graph as a sequence of sub-cascade graphs and learns the local structure of each sub-cascade by graph convolutions, and then uses LSTM to capture the evolving process of cascade structure [2].

• *DMT-LIC*: is a multi-task learning-based cascade model, which jointly learns both user-level behavior and cascade-level prediction. The shared-representation layer based on the attention mechanism and gated mechanism is used to capture both the underlying structure of a cascade graph and node sequence in the diffusion process [39].

Note that we omit the comparison with some baselines such as CYAN-RNN [16], DeepInf [7], DeepDiffuse [3], SNIDSA [38], FOREST [58], due to the following reasons: (1) these models mainly focus on microscopic predictions (e.g., user-level activation or influence); and (2) these models require information of the whole graph (e.g., friend/follower graph) and do not exhibit comparable performance on cascade size prediction in our setting – i.e., only retweet/citation structure has been used.

Parameter setting: All models, including ours, are tuned to the best performance with early stopping when validation loss has not declined for 10 consecutive epochs. For baselines, the learning rate and $L2$ coefficient are selected from $\{1, 10^{-1}, \ldots, 10^{-8}\}$; node embedding size for DeepCas, Topo-LSTM, DeepHawkes, CasCN and DMT-LIC is set to 50; the batch size is 64; and all the other hyper-parameters of each model are set to their default values.

For the scale parameter s used for node embedding in VaCas, we use a theoretically justified method proposed in [44] 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 $E(v) = [E_{s_{\min}}(v), E_{s_{\max}}(v)]$ by a concatenation operation, with $d = 10$ evenly spaced points, the final embedding size of VaCas is 40 – smaller than baselines for a fair comparison. The dimensionality of the latent factor $\mathbf{z}_1$ and $\mathbf{z}_2$ in VaCas are 32 and 128, respectively.

Evaluation protocols: Following existing works [2], [12], [13], [15], we use mean square log-transformed error (MSLE) and mean absolute percentage error (MAPE), defined as:

$$\text{MSLE} = \frac{1}{N}\sum_{i=0}^{N-1}(\log_2 \hat{P}_i - \log_2 P_i)^2,$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{|\log_2 \hat{P}_i - \log_2 P_i|}{\log_2 P_i},$$

where N is the total number of cascades and P_i is the incremental cascade size.

B. Performance Comparison

The overall performance of our VaCas, as well as the baselines, are shown in Table II. A paired t-test is performed and $*$ indicates a statistical significance $p < 0.001$ compared to the best baseline method. We have the following observations:

(O1): VaCas consistently outperforms the baselines by a significant margin. For the Weibo dataset, the MSLE and MAPE results of VaCas surpass the second best baseline (DMT-LIC) by 16% and 15.5%, respectively, which demonstrates the benefit of our hierarchical information cascade component.

(O2): The gaps between feature-based and other baselines are quite small. On Weibo dataset, feature-based approach even beats some earlier 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

TABLE II
PERFORMANCE COMPARISON. THE OBSERVATION WINDOWS ARE 0.5 HOUR FOR WEIBO AND 5 YEARS FOR APS.

Dataset	Weibo		APS	
Method \ Metric	MSLE	MAPE	MSLE	MAPE
Feature-based	2.834	0.356	1.721	0.318
DeepCas	3.097	0.377	1.713	0.310
Topo-LSTM	2.903	0.363	1.604	0.305
DeepHawkes	2.556	0.320	1.576	0.295
CasCN	2.513	0.306	1.421	0.288
DMT-LIC	2.420	0.291	1.377	0.284
VaCas	**2.032***	**0.246***	**1.337***	**0.279***

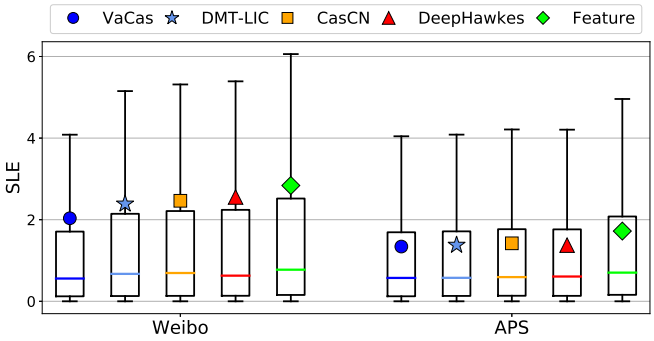


Fig. 4. The SLE distribution of five best performed models on two datasets. The markers and colored lines are the median and the mean value of SLE, respectively. Lower values are better.

to be generalized to other scenarios. This is verified by the performance of feature-based on the APS dataset, where it performs worse on both metrics.

(O3): As pioneers of deep learning-based cascade models, DeepCas and Topo-LSTM do not perform better than feature-based method, because they simply learn the representation of each user and compose the sub-graphs of active users with RNN models. However, they do not consider the time factor and the uncertain cascading effect that may significantly affect the growth of cascades.

(O4): DeepHawkes, on the contrary, does not consider the topology information of cascades. Therefore, its performance relies on the time-series modeling capability, which may prefer to overrate the cascade size due to its rudimentary self-excitation mechanism [37]. CasCN, as our VaCas, only leverages the structural and temporal factors for cascade size prediction. However, it focuses on the cascade-level structure learning, ignoring the micro-level user behavior.

(O5): 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 VaCAS performance gain over DMT-LIC illustrates the superiority of modeling the uncertainty of information diffusion at both the sub-graph level and the cascade level.

TABLE III
THE ABLATION ANALYSIS ON TWO DATASETS.

Dataset	Weibo		APS	
VaCas-Structural	2.501	0.302	1.645	0.301
VaCas-Temporal	2.348	0.279	1.537	0.294
Cas-RNN	2.345	0.277	1.370	0.287
VaCas*	2.162	0.258	1.352	0.282
VaCas	**2.032**	**0.246**	**1.337**	**0.279**

In addition to overall performance depicted in Table II, we also report the distribution of square log-transformed error (SLE) of the results to demonstrate the error distributions for different models, illustrated as box-plots in Fig. 4. We can clearly observe that VaCas outperforms all other methods in terms of both median and mean errors on both datasets.

C. Ablation Study

To better investigate the contribution of each component in VaCas, we implemented the following four variants:

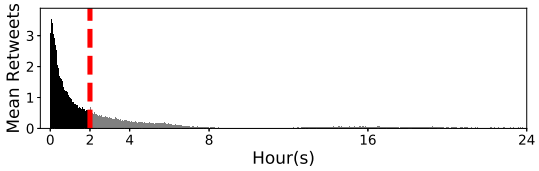
- *VaCas-Temporal* – in which we remove the structural information in cascades, i.e., every retweet/citation is directly connected to the original tweet/paper.
- *VaCas-Structural* – which, in contrast to VaCas-Temporal, only uses the structural information for model training.
- *Cas-RNN* – uses two-layer Bi-GRU output $\mathbf{h}_2$ for modeling cascades and predicting the cascade size, i.e., without hierarchical variational representation learning.
- *VaCas** – is the shallow version of VaCas and only leverages lower-level uncertainty representation, i.e., $\mathbf{z}_1$, combined with $\mathbf{h}_2$, for prediction.

Table III outlines the performance comparison among VaCas and its variants, which illustrates that: **(i)** VaCas-Structural does not show comparable performance, demonstrating the importance of time factor that has been observed in many previous works [2], [11], [13]; **(ii)** VaCas-Temporal outperforms all other baselines. This verifies the benefit of modeling information diffusion uncertainty even without exploiting the topological structure of cascades; **(iii)** Surprisingly, without modeling hierarchical cascading effect, Cas-RNN obtains quite well result, which can be attributed to the node embedding of our method that involves the graph wavelets – arguably, it explores the diffusion effect of nodes during embedding, although such an exploration is random and incomplete; **(iv)** The fact that VaCas* is the best variant demonstrates our motivation of modeling information diffusion uncertainty. In addition, VaCas reaps notable improvement over VaCas*, which directly reflects the benefit of modeling the hierarchical cascade uncertainty.

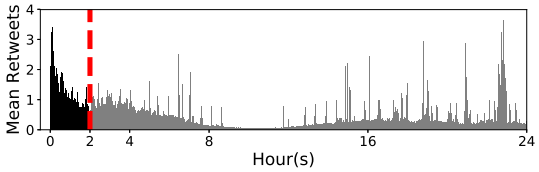
D. Model Interpretability

We now turn to interpret the performance of VaCas.

Counterfactual analysis: To obtain deeper insight of the VaCas performance, we select 1,000 best and 1,000 worst predictions made by VaCas for Weibo dataset and plot their

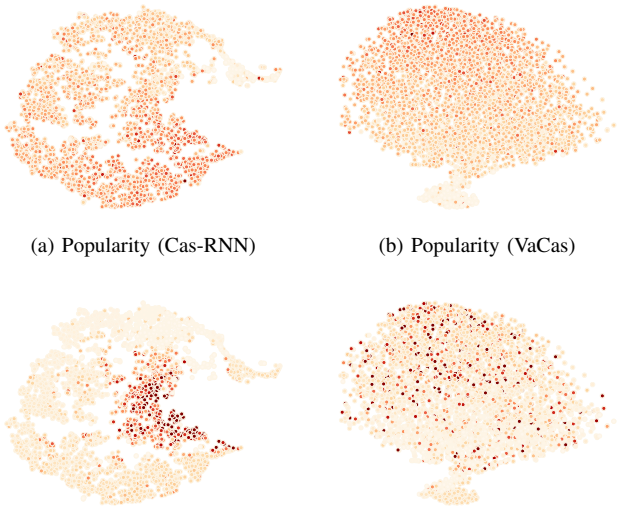


(a) Worst predictions (Weibo).



(b) Best predictions (Weibo).

Fig. 5. VaCas performance on 1,000 best and 1,000 worst predictions in terms of SLE for Weibo dataset. Red dashed line denotes the observation time (2 hours). Y-axis is the mean number of retweets.



(a) Popularity (Cas-RNN) (b) Popularity (VaCas)

(c) Structural Virality (Cas-RNN) (d) Structural Virality (VaCas)

Fig. 6. Visualization of learned latent representation on Weibo dataset. The darker the node, the larger the value of its cascade size or structural virality.

average retweet sizes growing within time, as illustrated in Fig. 5. First, we can observe that VaCas would be more likely to make wrong predictions if the tweets have very fewer forwards after the observation window, as shown in Fig. 5a – on average, their retweeting drops, and even stops after several hours (e.g., 5 or 6 hours). Arguably, this situation is common to see when some unexpected situations occur during the process of information diffusion, e.g., the removal of the fake news, rumors, and information violating the regulations. In contrast, VaCas achieves significantly better performance when the information cascades are normal as described in Fig. 5b. Due to space constraint, we do not show plots of APS here in which the results are also hold.

Latent representation: To have an intuitive explanation regarding the superiority of VaCas (especially the VAE component), we plot the learned latent representation of cascades for Cas-RNN and VaCas in Fig. 6 using t-SNE. Generally, the network properties (e.g., popularity and structural virality) need to be well learned for better predictions. For example, cascades with spatially closed popularity or structural virality would have similar predicted results. In Fig. 6a and Fig. 6c, we see clear clustering phenomena by Cas-RNN on two features: popularity and structures. In contrast, VaCas "smoothes" this clustering effect by modeling the information diffusion uncertainty (as the multi-modal Gaussian in Fig. 6b and Fig. 6d), which should help in exploring more possibilities and is therefore more suitable for cascade prediction – because cascade size prediction is essentially a regression task rather than a classification problem.

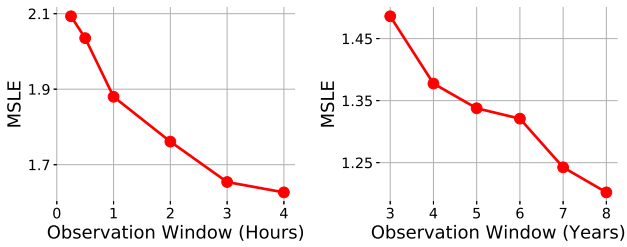


Fig. 7. Impact of the observation window on the performance for two datasets.

Parameter sensitivity: The most important parameter in VaCas (also the case for baselines) is the observation window. As illustrated in Fig. 7, the more sub-graphs we observe, the more accurate prediction VaCas can make, which is a natural outcome of increasing the training data.

VI. Conclusion

In this work, we introduced VaCas - the first Bayesian learning-based approach for information cascade prediction. It leverages a hierarchical variational information diffusion model to exploit the uncertainties at the sub-graph level and the cascade level, and learn the posterior of cascade distribution with variational inference. Our experimental evaluations on two real-world datasets demonstrated that VaCas significantly improves the cascade prediction accuracy, outperforming the state of the art baselines. In addition, VaCas provides interpretation of its behavior. 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 VaCas to incorporate other contexts – e.g., business-related – and apply it to problem domains such as more effective advertisement and interpreting of viral information spreading (e.g., rumors and fake news) in network settings.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant No.61602097, 61472064, U19B2028 and 61772117), NSF grants III 1213038 and CNS 1646107 and ONR grant N00014-14-10215.

REFERENCES

[1] Y. Dong, R. A. Johnson, and N. V. Chawla, "Will this paper increase your h-index?: Scientific impact prediction," in *WSDM*, 2015.

[2] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, and F. Zhang, "Information diffusion prediction via recurrent cascades convolution," in *ICDE*, 2019.

[3] M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrishnan, "Deepdiffuse: Predicting the'who'and'when'in cascades," in *ICDM*, 2018.

[4] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM Sigmod Record*, vol. 42, no. 2, pp. 17–28, 2013.

[5] 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 *AAAI*, 2019.

[6] S. Mishra, M.-A. Rizoiu, and L. Xie, "Modeling popularity in asynchronous social media streams with recurrent neural networks," in *AAAI*, 2018.

[7] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," in *KDD*, 2018.

[8] 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 *WWW*, 2018.

[9] O. Tsur and A. Rappoport, "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities," in *WSDM*, 2012.

[10] S. Mishra, M.-A. Rizoiu, and L. Xie, "Feature driven and point process approaches for popularity prediction," in *CIKM*, 2016.

[11] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, "Can cascades be predicted?" in *WWW*, 2014.

[12] 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 *KDD*, 2015.

[13] Q. Cao, H. Shen, K. Cen, W. Ouyang, and X. Cheng, "Deephawkes: Bridging the gap between prediction and understanding of information cascades," in *CIKM*, 2017.

[14] 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 *WWW*, 2017.

[15] C. Li, J. Ma, X. Guo, and Q. Mei, "Deepcas: An end-to-end predictor of information cascades," in *WWW*, 2017.

[16] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, "Cascade dynamics modeling with attention-based recurrent neural network." in *IJCAI*, 2017.

[17] J. Wang, V. W. Zheng, Z. Liu, and K. C.-C. Chang, "Topological recurrent neural network for diffusion prediction," in *ICDM*, 2017.

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

[19] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arxiv*, 2014.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[21] P. Cui, S. Jin, L. Yu, F. Wang, W. Zhu, and S. Yang, "Cascading outbreak prediction in networks: a data-driven approach," in *KDD*, 2013.

[22] Z. Ma, A. Sun, and G. Cong, "On predicting the popularity of newly emerging hashtags in twitter," *JASIST*, 2013.

[23] L. Weng, F. Menczer, and Y.-Y. Ahn, "Predicting successful memes using network and community structure." in *ICWSM*, 2014.

[24] P. Bao, H. Shen, J. Huang, and X. Cheng, "Popularity prediction in microblogging network: a case study on sina weibo," in *WWW*, 2013.

[25] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of youtube videos," in *WSDM*, 2013.

[26] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Communications of the ACM*, 2010.

[27] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *WSDM*, 2011.

[28] T. Iwata, A. Shah, and Z. Ghahramani, "Discovering latent influence in online social activities via shared cascade poisson processes," in *KDD*, 2013.

[29] H. Shen, D. Wang, C. Song, and A.-L. Barabási, "Modeling and predicting popularity dynamics via reinforced poisson processes," in *AAAI*, 2014.

[30] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2012.

[31] S.-H. Yang and H. Zha, "Mixture of mutually exciting processes for viral diffusion," in *ICML*, 2013.

[32] T. Zaman, E. B. Fox, E. T. Bradlow *et al.*, "A bayesian approach for predicting the popularity of tweets," *The Annals of Applied Statistics*, vol. 8, no. 3, pp. 1583–1611, 2014.

[33] D. Hunter, P. Smyth, D. Q. Vu, and A. U. Asuncion, "Dynamic egocentric models for citation networks," in *ICML*, 2011.

[34] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Modeling information propagation with survival theory," in *ICML*, 2013.

[35] 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 *ICDM*, 2015.

[36] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: model and implications," in *KDD*, 2012.

[37] X. Gao, Z. Cao, S. Li, B. Yao, G. Chen, and S. Tang, "Taxonomy and evaluation for microblog popularity prediction," *TKDD*, vol. 13, no. 2, p. 15, 2019.

[38] Z. Wang, C. Chen, and W. Li, "A sequential neural information diffusion model with structure attention," in *CIKM*, 2018.

[39] X. Chen, K. Zhang, F. Zhou, G. Trajcevski, T. Zhong, and F. Zhang, "Information cascades modeling via deep multi-task learning," in *SIGIR*, 2019.

[40] B. Shulman, A. Sharma, and D. Cosley, "Predictability of popularity: Gaps between prediction and understanding," in *ICWSM*, 2016.

[41] Z. T. Kefato, N. Sheikh, L. Bahri, A. Soliman, A. Montresor, and S. Girdzijauskas, "Cas2vec: Network-agnostic cascade prediction in online social networks," in *SNAMS*, 2018.

[42] C. Gou, H. Shen, P. Du, D. Wu, Y. Liu, and X. Cheng, "Learning sequential features for cascade outbreak prediction," *Knowledge and Information Systems*, pp. 1–19, 2018.

[43] M. Jenders, G. Kasneci, and F. Naumann, "Analyzing and predicting viral tweets," in *WWW*, 2013.

[44] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *KDD*, 2018.

[45] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[46] 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 Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[47] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014.

[48] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016.

[49] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *KDD*, 2017.

[50] E. Lukacs, "Characteristic functions," *Griffin*, 1970.

[51] S. Lamprier, "A recurrent neural cascade-based model for continuous-time diffusion," in *ICML*, 2019.

[52] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[53] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *AAAI*, 2017.

[54] D. Zhu, P. Cui, D. Wang, and W. Zhu, "Deep variational network embedding in wasserstein space," *KDD*, 2018.

[55] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, Z. Ting, and F. Zhang, "Predicting human mobility via variational attention," in *WWW*, 2019.

[56] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *WWW*, 2018.

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

[58] C. Yang, J. Tang, M. Sun, G. Cui, and Z. Liu, "Multi-scale information diffusion prediction with reinforced recurrent networks," in *IJCAI*, 2019.