Inf-VAE: A Variational Autoencoder Framework to Integrate Homophily and Influence in Diffusion Prediction

Aravind Sankar, Xinyang Zhang, Adit Krishnan, Jiawei Han
University of Illinois at Urbana-Champaign, IL, USA
{asankar3, xz43, aditk2, hanj}@illinois.edu

ABSTRACT

Recent years have witnessed tremendous interest in understanding and predicting information spread on social media platforms such as Twitter, Facebook, etc. Existing diffusion prediction methods primarily exploit the sequential order of influenced users by projecting diffusion cascades onto their local social neighborhoods. However, this fails to capture global social structures that do not explicitly manifest in any of the cascades, resulting in poor performance for inactive users with limited historical activities.

In this paper, we present a novel variational autoencoder framework (Inf-VAE) to jointly embed *homophily* and *influence* through proximity-preserving *social* and position-encoded *temporal* latent variables. To model social homophily, Inf-VAE utilizes powerful graph neural network architectures to learn social variables that selectively exploit the social connections of users. Given a sequence of seed user activations, Inf-VAE uses a novel expressive *co-attentive fusion network* that jointly attends over their social and temporal variables to predict the set of all influenced users. Our experimental results on multiple real-world social network datasets, including Digg, Weibo, and Stack-Exchanges demonstrate significant gains (22% MAP@10) for Inf-VAE over state-of-the-art diffusion prediction models; we achieve massive gains for users with sparse activities, and users who lack direct social neighbors in seed sets.

CCS CONCEPTS

• **Information systems → Social networks**; • **Computing methodologies → Neural networks**;

KEYWORDS

Social Network, Diffusion, Deep Learning, Autoencoder, Attention

ACM Reference Format:
Aravind Sankar, Xinyang Zhang, Adit Krishnan, Jiawei Han. 2020. Inf-VAE: A Variational Autoencoder Framework to Integrate Homophily and Influence in Diffusion Prediction. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371811

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WSDM '20, February 3–7, 2020, Houston, TX, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371811

1 INTRODUCTION

In social media, information disseminates or *diffuses* to a large number of users through posting or re-sharing behavior, resulting in a *cascade* of user activations, *e.g.*, a user voting a news story on Digg (a social news sharing website) triggers a series of votes from multiple users, who may be his friends or other users interested in the same story. Given a set of *activated* seed users, diffusion models aim to predict the set of all influenced users. Diffusion modeling has widespread social media applications, including viral marketing [24], recommendations [22, 23], and popularity prediction [51].

The diffusion prediction problem has received significant attention in the research community. Unlike pre-defined propagation hypotheses [18], recent methods learn data-driven diffusion models from collections of user activation sequences (*diffusion cascades*). Existing diffusion models broadly fall into two categories.

Probabilistic generative cascade models use hand-crafted features including roles [47], communities [4], topics [3], and structural patterns [49]. Such methods rely on feature engineering that requires manual effort and extensive domain knowledge, and are limited by the modeling capacity of carefully chosen probability distributions.

Representation learning methods avoid feature extraction by learning user embeddings characterizing their influencing ability and conformity [7, 9]. State-of-the-art methods project cascades onto local social neighborhoods to generate Directed Acyclic Graphs (DAGs), and propose extensions of Recurrent Neural Networks (RNNs). In particular, DAG-structured LSTMs [41] explicitly operate on the induced DAG, while attention-based RNNs [17, 43, 44] implicitly consider cross-dependence for diffusion prediction.

Prior works only consider the sequence or projected social structure (induced DAG) of previously influenced users while ignoring *social structures that do not manifest in cascades*. As a result, they only capture the temporal correlation of diffusion behaviors among users, which is also known as *temporal influence* or *contagion* [37]. Consider a Twitter user with interests in politics, who is likely to follow famous political leaders and join interest groups that induce transitive connections to other users; however, these connections may not appear in cascades unless she re-tweets or posts content. *Social homophily* [27] suggests that ties are more likely between users with shared traits or interests, which can induce correlated diffusion behaviors without direct causal influence. Since a vast majority of social media users seldom post content and thus rarely appear in cascades, it is critical to exploit their social neighborhood structures to characterize social homophily accurately.

However, homophilous diffusion and contagion can result in significantly different dynamics, *e.g.*, contagions are self-reinforcing and viral while homophily hinges on users' preferences or traits. Real-world cascades are often a complex combination of both aspects with user-specific variations. Indeed, it is well known that social homophily and temporal influence are fundamentally confounded in observational studies [37]. Thus, we propose a data-driven framework to contextually model their joint effect when predicting user-level diffusion behaviors. Therefore, our key objective is to *develop a principled neural framework to unify social homophily and temporal influence in diffusion prediction.*

Our architecture Inf-VAE jointly models *homophily* through *social* embeddings preserving social network proximity and *influence* through *temporal* embeddings encoding the relative sequential order of user activations. Motivated by the recent successes of variational autoencoders (VAEs) [19] in characterizing sparse users via Gaussian priors [26], and the expressive power of graph neural networks [14, 21], we adopt VAEs to model social homophily. We learn structure-preserving social embeddings through a VAE framework that supports a wide range of graph neural network architectures as encoders and decoders. Given an initial set of seed user activations, Inf-VAE utilizes an expressive *co-attentive fusion network* that captures complex non-linear correlations between social and temporal embeddings, to model their joint effect on predicting the set of all influenced users. We make the following contributions:

- **Generalizable Variational Autoencoder Framework**: Unlike existing diffusion prediction methods that only consider local induced propagation structures, Inf-VAE is a generalizable VAE framework that models social homophily through graph neural network architectures of arbitrary complexity, to selectively exploit the rich global network of social connections.
- **Efficient Homophily and Influence Integration:** To the best of our knowledge, ours is the first work to comprehensively exploit social homophily and temporal influence in diffusion prediction. Given a sequence of seed user activations, Inf-VAE employs an expressive *co-attentive fusion network* to jointly attend over their social and temporal embeddings to predict the set of all influenced users. Inf-VAE with co-attentions is faster than state-of-the-art recurrent methods by an order of magnitude.
- **Robust Experimental Results:** Our experiments on multiple real-world social networks, including Digg, Weibo, and Stack-Exchanges, demonstrate significant gains for Inf-VAE over state-of-the-art models. Modeling social homophily through VAEs enables massive gains for users with *sparse activities*, and users who *lack direct social neighbors in seed sets.* An ablation analysis of various modeling choices further highlights the synergistic effects of jointly modeling homophily and temporal influence.

2 RELATED WORK

We discuss existing work on diffusion modeling followed by related work on network representation learning and co-attentions.

Information diffusion overview. Historically, information diffusion has been studied through two seminal models: Independent Cascade (IC) [18] and Linear Threshold (LT) [11]. Three distinct applications emerged, namely: *network inference* [10], which infers the underlying social network that best explains the observed cascades; *cascade prediction* [25], which predicts macroscopic properties of cascades, including size, growth, and shape; and *diffusion prediction* [41], which learns a model from social links and cascade sequences, to predict the set of influenced users given a seed set of activated users. In this paper, we focus on diffusion prediction.

Diffusion prediction. The earliest data-driven methods propose several extensions of IC and LT incorporating topics [3], continuous timestamps [31], user profiles [32], and community structure [4]. A few techniques explore probabilistic generative models via latent topics and communities [16, 50]. Most recent studies focus on learning representations to overcome feature engineering or predefined hypotheses in diffusion modeling [6, 7, 9, 30, 41–44]. Emb-IC [7], Inf2vec [9] embed user influencing capability and susceptibility in diffusion. Topo-LSTM [41], CYAN-RNN [43], SNIDSA [44], and DeepDiffuse [17] project the diffusion cascades on local social neighborhoods and model the resulting DAG propagation structures with RNNs. These techniques outperform classical approaches by significant margins in diffusion prediction. Our key observation is that these projected DAGs could ignore social structures that do not appear in any observed cascade. In contrast, our model Inf-VAE can account for unobserved social connections in the user activation process by modeling social homophily through VAEs.

A related problem is social influence prediction, which aims to classify social media users based on the activation status of their ego-network [30, 48]. Direct extensions to predict the set of all influenced users (diffusion prediction) entails reapplying their models on each candidate inactive user in the social network, resulting in prohibitive inference costs, hence preventing a comparison.

Network representation learning: This line of work captures varied notions of structural node proximity [12, 33] in networks via low-dimensional vectors. Notably, graph neural networks have achieved great success in node classification and link prediction [13, 21, 28, 34–36, 39]. Graph Autoencoders [20, 40] employ various encoding and decoding architectures to embed network structure and learn unsupervised node embeddings. Hamilton et al. [14] unify a large family of network embedding methods in an autoencoder framework. However, general-purpose embeddings modeling structural proximity are not directly suited to diffusion modeling.

Co-attentional models: Our work also leverages recent advances in neural attention mechanisms, especially in Natural Language Processing [2]. Specifically, co-attention has achieved great success in modeling relationships between pairs of sequences, *e.g.*, question-answer [45], etc. Co-attentional methods compute interaction weights between data modalities, learning fine-grained non-linear correlations. In our work, we develop a co-attentive fusion network to capture the contextual interplay of users' social and temporal representations for diffusion prediction.

3 PROBLEM DEFINITION

We study diffusion prediction where the goal is to predict the set of all influenced users, given temporally ordered seed user activations.

Definition 3.1. **Social Network:** The social network is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_i\}_{i=1}^{N}$ is the set of N users and $\mathcal{E} = \{e_{ij}\}_{i,j=1}^{N}$ is the set of links. We denote the adjacency matrix of $\mathcal{G}$ by $A \in \mathbb{R}^{N \times N}$ where $A_{i,j} = 1$ if $e_{i,j} \in \mathcal{E}$ otherwise 0.

Definition 3.2. **Diffusion cascade:** A diffusion cascade D_i is an ordered sequence of user activations in ascending order of time denoted by: $D_i = \{(v_{i_k}, t_k) \mid v_{i_k} \in \mathcal{V}, t_k \in [0, \infty), \ k = 1 \dots K\}$, each v_{i_k} is a distinct user in $\mathcal{V}$ (no repeats) and t_k is non-decreasing, *i.e.*, $t_k \leq t_{k+1}$. The k^{th} user activation is recorded as tuple (v_{i_k}, t_k), referring the activated user and activation time.

We represent cascades by delay-agnostic relative activation orders similar to [7, 9, 41], *i.e.*, a cascade is equivalently written as $D = \{(v_{i_k}, k) \mid v_{i_k} \in \mathcal{V}\}_{k=1}^{K}$. We do not assume the availability of explicit re-share links between users in cascades; this corresponds to the simplest yet most general setting of diffusion [9, 41]. Though timestamps may be easily used as input features, we leave generation of continuous timestamps as future work.

Definition 3.3. **Diffusion prediction:** Given a social network G and a collection of cascade sequences $\mathbb{D} = \{D_i, 1 \le i \le |\mathbb{D}|\}$, learn diffusion model M to predict the future set of influenced users in a cascade with the seed activation sequence $I = \{(v_{i_1}, 1), \dots, (v_{i_k}, k)\}$ of k seed users. Diffusion prediction estimates the probability of influencing each inactive user: $P_\Theta(v \mid I)\ \forall v \in \mathcal{V} - I$, inducing a ranking of activation likelihoods over the set of inactive users.

We create a training set $\mathbb{T}$ of diffusion *episodes* containing (seed activations, activated users) tuples from the cascade collection $\mathbb{D}$, by randomly splitting each cascade $D \in \mathbb{D}$ of length K at each time step $2 \le k \le K - 1$. Specifically, a split at time step $k \ge 2$, creates a training episode (I_k, C_k) where $I_k = \{(v_{i_j}, j); 1 \le j \le k\}$ is the seed set consisting of the cascade sliced at k and $C_k = \{v_{i_{k+1}}, \dots, v_{i_K}\}$ is the set of influenced users after time step k. Thus, we denote the training set by $\mathbb{T} = \{(I_i, C_i)\ 1 \le i \le |\mathbb{T}|\}$.

4 INFLUENCE VARIATIONAL AUTOENCODER

In this section, we describe our proposed Influence Variational Autoencoder (Inf-VAE) for predicting information diffusion.

4.1 Model Description

We describe the latent variables modeling social homophily and temporal influence, followed by our generative network Inf-VAE.

4.1.1 ***Social Homophily.*** Our objective is to define latent *social variables* for users that capture social homophily. The homophily principle stipulates that users with similar interests are more likely to be connected. In the absence of explicit user attributes, we posit that highly interconnected users in social communities share homophilous relationships. We model social homophily through latent *social* variables designed to encourage users with shared social neighborhoods to have similar latent representations.

Specifically, we assign a latent *social* variable $\mathbf{z}_i$ for user v_i, where the prior for $\mathbf{z}_i$ is chosen to be a unit normal distribution, in line with standard assumptions in VAEs. Normal distributions are chosen in VAE frameworks due to their flexibility to support arbitrary functional parameterizations by isolating sampling stochasticity to facilitate back-propagation [19]. We assume the latent social variables Z to collectively generate the social network $\mathcal{G}$, through a graph generation neural network $f_{\text{DEC}}(Z)$ parameterized by θ. The corresponding generative process is given by:

$$\mathbf{z}_i \sim \mathcal{N}(0, I_D) \quad \mathcal{G} \sim p_\theta(\mathcal{G} \mid Z) = p_\theta(\mathcal{G} \mid f_{\text{DEC}}(Z)) \tag{1}$$

where $I_D \in \mathbb{R}^{D \times D}$ is an identity matrix of D dimensions. Here, the graph generation neural network $f_{\text{DEC}}(Z)$ can be instantiated to preserve an arbitrary notion of structural proximity in the social network $\mathcal{G}$ (Sec 4.3). In the above equation, we abuse the notation of $\mathcal{G}$ to denote an appropriate representational form of the social network structure, which can take multiple forms, including the adjacency matrix, random walks sampled from $\mathcal{G}$, etc.

Symbol	Description
$\boldsymbol{Z}$	Social variables modeling network proximity, for all users $\mathcal{V}$
$\boldsymbol{V_S}$	Sender variables for all users $\mathcal{V}$
$\boldsymbol{V_R}$	Receiver variables for all users $\mathcal{V}$
$\boldsymbol{V_T}$	Temporal influence variables for all users $\mathcal{V}$
$\boldsymbol{V_P}$	User-specific popularity variables for all users $\mathcal{V}$
$\boldsymbol{P_K}$	Position-encoded temporal embeddings for all time steps K

Table 1: Notations

While homophily characterizes peer-to-peer interest similarity, its impact on user behaviors tends to asymmetric since users who share interests may drastically differ in their posting rates, *e.g.*, certain users are naturally predisposed to be socially active and hence more *influential* in comparison to others. Thus, it is necessary to differentiate user *roles* when modeling the effect of social homophily on diffusion behaviors. Similar concepts have been examined in social influence literature to characterize users by their influencing capability and conformity [7–9, 41, 42].

We associate each user $v_i \in \mathcal{V}$ with a *sender* $\mathbf{v}_i^s \in \mathbb{R}^D$ and *receiver* $\mathbf{v}_i^r \in \mathbb{R}^D$ latent variable. Our key innovation lies in conditioning the information sending and receiving capabilities of users on their homophilous traits. We use normal distributions centered at $\mathbf{z}_i$ to define the *sender* and *receiver* variables for user v_i as:

$$\mathbf{v}_i^s \sim \mathcal{N}(\mathbf{z}_i, \lambda_s^{-1} I_D) \quad \mathbf{v}_i^r \sim \mathcal{N}(\mathbf{z}_i, \lambda_r^{-1} I_D) \tag{2}$$

where λ_s, λ_r are hyper-parameters controlling the degree of variation or uncertainty for $\mathbf{v}_i^s$ and $\mathbf{v}_i^r$ *w.r.t.* $\mathbf{z}_i$. Let V_S and V_R denote the set of all sender and receiver variables respectively for all users.

4.1.2 ***Temporal Influence.*** Now, we define latent *temporal influence* variables to describe the varying influence effects of seed users depending on the relative sequential order of activations. There are two interesting factors at play: activation orders and popularity effects. A majority of social media users adopt more recent information while often ignoring old and obsolete content [46]. On the other hand, social status impacts the influencing power of seed users independent of their activation order and social neighbors, *e.g.*, famous media figures naturally exert significant influence. Thus, we consider both the relative sequential order of user activations and popularity effects of seed users to model temporal influence.

To quantify the temporal influence exerted by a seed user activation (v_{i_k}, k) of user v_{i_k} at time step k $(1 \le k \le K)$, we first encode the relative position k through positional-encodings [38] to obtain temporal embeddings $\mathbf{p}_k$. Since we expect the variation in popularity effects to be quite small, we draw user-specific popularity variables from a zero-mean normal distribution to serve as offsets to the temporal embeddings. Specifically, the *temporal influence* variable for activation (v_{i_k}, k) denoted by $\mathbf{v}_{i_k}^t$, is given by:

$$\mathbf{v}_{i_k}^p \sim \mathcal{N}(0, \lambda_p^{-1} I_D) \qquad \mathbf{p}_k = PE(k) \qquad \mathbf{v}_{i_k}^t = \mathbf{v}_{i_k}^p + \mathbf{p}_k \tag{3}$$

$$PE(k)_{2d} = sin(k/10000^{2d/D}) \;\; PE(k)_{2d+1} = cos(k/10000^{2d/D})$$

where λ_p is a hyper-parameter to control the popularity effects, and $1 \le d \le D/2$ denotes the dimension in the temporal embedding $\mathbf{p}_k$. Note that the popularity variable $\mathbf{v}_{i_k}^p$ is user-specific, while temporal embedding $\mathbf{p}_k$ only depends on the activation step k. The

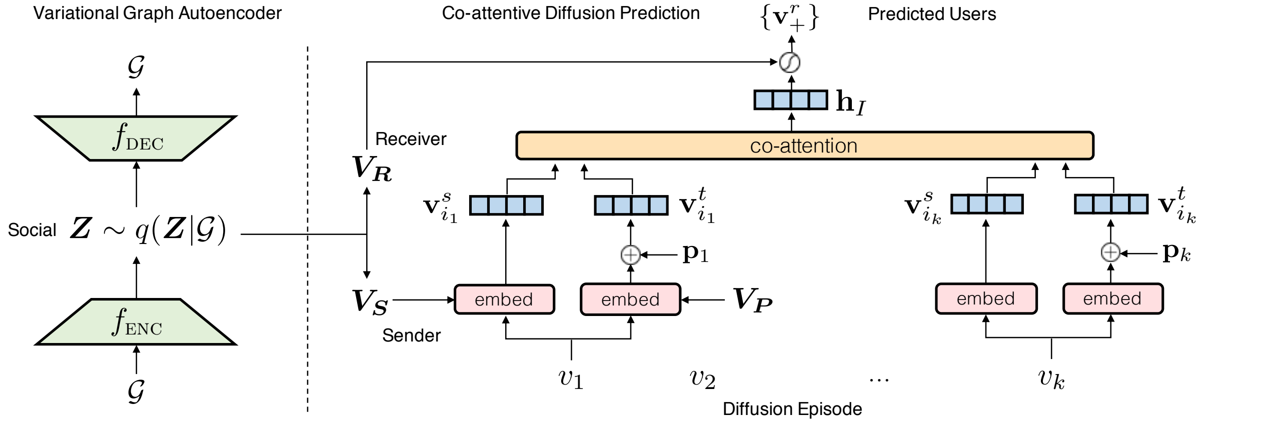


Figure 1: Neural Architecture of Inf-VAE depicting latent variable interactions. The left side indicates the VAE framework to model social homophily; right side denotes the co-attentive fusion network to integrate the social and temporal variables.

set of all latent user popularity variables are denoted by $\boldsymbol{V}_P$, while $\boldsymbol{P}_K$ represents the set of position-encoded *temporal* embeddings.

4.1.3 ***Co-attentive Diffusion Episode Generation.*** Let us consider a single diffusion episode $(I, C) \in \mathbb{T}$, with seed activations $I = \{(v_{i_1}, 1), \ldots, (v_{i_k}, k)\}$ and influenced users $C = \{v_{i_{k+1}}, \ldots, v_{i_K}\}$. A diffusion model aims to predict the set of influenced users C given seed activations I. Since diffusion is always conditioned on I, we propose a conditional generative process to sample C given I.

Let us denote the set of seed users by $I_U = \{v_{i_1}, \ldots, v_{i_k}\}$. Our objective is to jointly model the effects of social homophily and temporal influence exerted by seed users I_U, which can be summarized by: *sender sequence* $(\mathrm{v}^s_{i_1}, \mathrm{v}^s_{i_2}, \ldots, \mathrm{v}^s_{i_K})$; and *temporal influence sequence* $(\mathrm{v}^t_{i_1}, \mathrm{v}^t_{i_2}, \ldots, \mathrm{v}^t_{i_K})$. To model complex correlations between the sender and temporal influence sequences, we propose an expressive *co-attentive* fusion strategy to learn attention scores for each seed user by modeling interactions between the two sequences. We describe the conditional generative process in two steps:

- The social homophily and temporal influence aspects of seed users, are integrated into an aggregate seed set representation $\mathbf{h}_I$. The co-attentive fusion network $G_{\text{DIFF}}(\cdot)$ performs homophily-guided temporal attention, *i.e.*, attends over the temporal influence variables by computing co-attentional weights that jointly depend on both homophily and temporal influence characteristics. As illustrated in Figure 1, the sender and temporal influence variables of seed users feed into a fusion network $G_{\text{DIFF}}(\mathbf{v}^s_{i_k}, \mathbf{v}^t_{i_k})$. The aggregate seed representation $\mathbf{h}_I$ is computed as:

$$\alpha_k = \frac{\exp(G_{\text{DIFF}}(\mathbf{v}^s_{i_k}, \mathbf{v}^t_{i_k}(k)))}{\sum\limits_{j=1}^{K} \exp(G_{\text{DIFF}}(\mathbf{v}^s_{i_j}, \mathbf{v}^t_{i_j}(j)))} \qquad \mathbf{h}_I = \sum_{j=1}^{K} \alpha_j \mathbf{v}^t_{i_j}(j) \tag{4}$$

  Each α_j is the normalized co-attentional coefficient for seed user v_{i_k} denoting its contribution in computing the aggregate representation $\mathbf{h}_I$. To model the co-dependence between $\mathbf{v}^s_i, \mathbf{v}^t_i$, we define the co-attentive function $G_{\text{DIFF}}(\mathbf{v}^s_{i_k}, \mathbf{v}^t_{i_k}) = tanh(\mathbf{v}^{s\ T}_{i_k} \boldsymbol{W} \mathbf{v}^t_{i_k})$ as a bi-linear product parameterized by $\boldsymbol{W} \in \mathbb{R}^{D \times D}$.
- The probability of influencing an inactive user v_j depends on the sending capacity of seed users (embedded in $\mathbf{h}_I$) and her receiving capability (encoded by *receiver* variable $\mathbf{v}^r_j$). We quantify the likelihood of influencing v_j by $\mathbf{h}_I^T \mathbf{v}^r_j$. For each inactive user $v_j \in \mathcal{V} - I_U$, we draw a binary variable $C_j \in \{0, 1\}$ indicating whether user v_j is influenced by set users I_U or not, given by:

$$C_j \sim Ber(\sigma(\mathbf{h}_I^T \mathbf{v}^r_j)) \; \forall v_j \in \mathcal{V} - \{v_{i_1}, \ldots, v_{i_K}\} \tag{5}$$

  where $\sigma(\cdot)$ is the sigmoid function and $Ber(\cdot)$ is the Bernoulli distribution. The corresponding logistic log-likelihood of generating a single diffusion episode (I, C) is given by:

$$\begin{aligned} \mathcal{L}^{\text{DIFF}}_{I,C} &= \log p_\theta(C \mid I, \boldsymbol{V}_S, \boldsymbol{V}_R, \boldsymbol{V}_P) \qquad (6) \\ &= \sum_{v \in C} \eta \log(\sigma(\mathbf{h}_I^T \mathbf{v}^r_i)) + \sum_{v_n \in \mathcal{V} - C - I_U} \log(1 - \sigma(\mathbf{h}_I^T \mathbf{v}^r_n)) \end{aligned}$$

  Here, η re-weights positive examples since the actual number of influenced users is much smaller than the total number of users.

4.2 Model Likelihood

Due to the intractability of analytically computing the latent posterior distribution $p(\boldsymbol{V}_S, \boldsymbol{V}_R, \boldsymbol{V}_P, \boldsymbol{Z}|\mathcal{G}, \mathbb{T})$, we use variational inference to factorize the posterior with a mean-field approximation:

$$q(\boldsymbol{V}_S, \boldsymbol{V}_R, \boldsymbol{V}_P, \boldsymbol{Z}|\mathcal{G}) = q(\boldsymbol{V}_S)q(\boldsymbol{V}_R)q(\boldsymbol{V}_P)q(\boldsymbol{Z}|\mathcal{G}) \tag{7}$$

The variational distributions of variables $\boldsymbol{V}_S, \boldsymbol{V}_R$, and $\boldsymbol{V}_P$ follow normal distributions while the social variables $\boldsymbol{Z}$ are conditioned on $\mathcal{G}$ through a structure-encoding inference network [19]. Specifically, the variational distribution of $\boldsymbol{Z}$ denoted by $q_\phi(\boldsymbol{Z}|\mathcal{G})$, is a diagonal normal distribution parameterized by $f_{\text{ENC}}(\mathcal{G})$ defined as:

$$f_{\text{ENC}}(\mathcal{G}) \equiv [\mu_\phi(\mathcal{G}), \log \sigma^2_\phi(\mathcal{G})] \quad q_\phi(\boldsymbol{Z}|\mathcal{G}) = \mathcal{N}\left(\mu_\phi(\mathcal{G}), diag(\sigma^2_\phi(\mathcal{G}))\right)$$

The inference network outputs the parameters, $\mu_\phi(\mathcal{G}), \sigma_\phi(\mathcal{G})$ of the variational distribution $q_\phi(\boldsymbol{Z}|\mathcal{G})$, which is designed to approximate the corresponding posterior $p(\boldsymbol{Z}|\mathcal{G})$. The inference network $f_{\text{ENC}}(\mathcal{G})$ endows the model with added flexibility to incorporate arbitrary neighborhood aggregation functions such as graph convolutions [21], attentions [39], etc. The variational structure distribution $q_\phi(\boldsymbol{Z}|\mathcal{G})$ and the structure generative model $p_\theta(\mathcal{G}|\boldsymbol{Z})$ (Eqn. 1) together constitutes a *variational graph autoencoder* [20].

4.3 Neural Graph Autoencoder Details

In this section, we describe functions $f_{\text{ENC}}(\mathcal{G})$ and $f_{\text{DEC}}(\boldsymbol{Z})$ which describe the graph structure inference and generative networks of Inf-VAE. The *encoder* summarizes local social neighborhoods into latent vectors, which are subsequently transformed by the *decoder* into high-dimensional structural information (*e.g.*, adjacency matrix). Hamilton et al. [14] present an encoder-decoder framework to conceptually unify a large family of graph embedding methods. Encoder architectures fall into three major categories: embedding lookups [12, 29], neighborhood vector encoding [40], and neighborhood aggregation [13], while decoders comprise unary and pairwise variants. In Inf-VAE, we explore two representative choices:

- **MLP + MLP**: We use a Multi-Layer Perceptron (MLP) to both encode and decode the laplacian matrix of $\mathcal{G}$, given by $\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$. The neighborhood vector for user v_i, denoted by $\mathbf{a}_i$, is the i^{th} row of $\boldsymbol{L} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]^T$. The encoder is an MLP network $f_{\text{ENC}}(\mathbf{a}_i)$ which encodes $\mathbf{a}_i$ into $\mathbf{z}_i$, while the decoder $f_{\text{DEC}}(\mathbf{z}_i)$ strives to reconstruct $\mathbf{a}_i$ from $\mathbf{z}_i$. We introduce a re-weighting vector $\mathbf{b}_i = \{b_{ij}\}_{j=1}^{N}$ where $b_{ij} = 1$ if $L_{ij} = 0$ and $b_{ij} = \beta > 1$ when $L_{ij} > 0$. β is a confidence parameter that re-weights the positive terms ($L_{ij} > 0$) to balance the unobserved $0's$ which far outnumber the observed links in real-world networks. The generative process to obtain $\mathbf{a}_i$ from $\mathbf{z}_i$ is given by:
  $$\mathbf{a}_i \sim p_\theta(\mathbf{a}_i|\mathbf{z}_i) = \mathcal{N}(f_{\text{DEC}}(\mathbf{z}_i), diag(\mathbf{b}_i))$$
  where $diag(\mathbf{b}_i)$ is a diagonal matrix with non-zero entries from vector b_i. The corresponding Gaussian log-likelihood is given by:
  $$\log p_\theta(A|Z) = \sum_{i=1}^{N} \log p_\theta(\mathbf{a}_i|\mathbf{z}_i) = \sum_{i=1}^{N} \left\|\mathbf{b}_i \odot (\mathbf{a}_i - f_{\text{DEC}}(\mathbf{z}_i))\right\|^2$$
- **GCN + Inner Product**: We use a Graph Convolutional Network (GCN) as the encoder and an inner product decoder that maps pairs of user embeddings to a binary indicator of link existence in $\mathcal{G}$. The GCN network comprises multiple stacked graph convolutional layers to extract features from higher-order structural neighborhoods. The input to a layer is a user feature (or embedding) matrix $X \in \mathbb{R}^{N \times F}$ and a normalized adjacency matrix $\hat{\boldsymbol{A}}$, where each GCN layer computes the function:
  $$f_{\text{ENC}}(\boldsymbol{A}) = \sigma(\hat{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}) \quad \hat{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} + \boldsymbol{I}_N$$
  where $\boldsymbol{X}$ is an identity matrix encoding user identities. Each entry A_{ij} of adjacency matrix $\boldsymbol{A}$ is generated according to:
  $$A_{ij} \sim p_\theta(A_{ij}|\mathbf{z}_i, \mathbf{z}_j) = Ber(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$$
  Similar to above, we re-weight the positive entries of $\boldsymbol{A}$ with a confidence parameter β. The logistic log-likelihood is given by:
  $$\log p_\theta(A|Z) = \sum_{(i,j)\in\mathcal{E}} \beta \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) + \sum_{(i,j)\notin\mathcal{E}} \log(1 - \sigma(\mathbf{z}_i^T \mathbf{z}_j))$$
  As an alternative to re-weighting positive entries, negative sampling [29] can scale this objective to large-scale networks.

4.4 Model Inference

The overall objective maximizes a lower bound on the marginal log likelihood, also named evidence lower bound (ELBO) [5], given by:

$$L_q = \mathbb{E}_q[\log p(\mathcal{G}, \mathbb{T}, V_S, V_R, V_P, Z) - \log q(V_S, V_R, V_P, Z|\mathcal{G})] \quad (8)$$

Note that L_q is a function of both generative (θ) and variational (ϕ) parameters. However, an analytical computation of the expectation with respect to $q_\phi(Z|\mathcal{G})$ is intractable, while Monte Carlo sampling prevents gradient back-propagation to the neural parameters of $f_{\text{ENC}}(\mathcal{G})$. With the reparametrization trick [19], we instead sample $\epsilon \sim \mathcal{N}(0, I_{N\times D})$ and form samples of $Z = \mu_\phi(\mathcal{G}) + \epsilon \odot \sigma_\phi(\mathcal{G})$. This isolates the stochasticity during sampling and the gradient with respect to ϕ can be back-propagated through the sampled Z.

Algorithm 1 Inf-VAE training with block coordinate ascent.

```
Input: Social Network ($\mathcal{G}$), Training episodes ($\mathbb{T}$)
Output: MAP estimates of $V_S, V_R, V_P$ and parameters $\theta, \phi$.
Initialize latent variables from a standard normal distribution.
Pre-training: Train $f_{\text{DEC}}(\mathcal{G}|Z)$ and $f_{\text{ENC}}(Z|\mathcal{G})$ on $\mathcal{G}$ using a
VAE with log-likelihood:
    $L^{\text{VAE}} = \mathbb{E}_{q_\phi(Z|\mathcal{G})} \log p_\theta(\mathcal{G}|Z) - D_{\text{KL}}(q_\phi(Z|\mathcal{G}), p(Z))$
while not converged do
  ▷ Optimize over social network $\mathcal{G}$
  for each batch of users $\mathcal{U} \subseteq \mathcal{V}$ do
      Fix $V_S, V_R, V_P, G_{\text{DIFF}}(\cdot)$ and update weights of $f_{\text{ENC}}(\mathcal{G})$
      and $f_{\text{DEC}}(Z)$ using mini-batch gradient ascent (Eqn. 9)
  ▷ Optimize over diffusion episodes $\mathbb{T}$
  for each batch of diffusion episodes $B \subseteq \mathbb{T}$ do
      Fix $Z$, $f_{\text{ENC}}(\mathcal{G})$, $f_{\text{DEC}}(Z)$ and update $V_S, V_R, V_P$, and
      $G_{\text{DIFF}}(.)$ using mini-batch gradient ascent. (Eqn. 9)
```

4.4.1 ***Optimization***. Since bayesian methods to infer latent posteriors incur high computational costs, and considering our goal of making good predictions rather than explanations, we resort to MAP (Maximum A Posteriori) estimation. Thus, we sample Z from $q_\phi(Z|\mathcal{G})$ using point estimates for V_S, V_R and V_P. We maximize the joint log-likelihood with MAP estimates of latent variables V_S, V_R, V_P, inference and generative network parameters θ, ϕ, and observations $\mathbb{T}$ and $\mathcal{G}$, given hyper-parameters $\lambda_s, \lambda_r, \lambda_p$:

$$\mathcal{L}^{\text{MAP}} = \mathbb{E}_{q_\phi}[\log p_\theta(\mathcal{G}|Z)] - D_{\text{KL}}(q_\phi, p(Z)) + \sum_{(I,C)\in\mathbb{T}} \mathcal{L}^{\text{DIFF}}_{I,C} \quad (9)$$
$$- \sum_{i=1}^{N} \left(\frac{\lambda_s}{2}\mathbb{E}_{q_\phi}||\mathbf{v}_i^s - \mathbf{z}_i||^2 + \frac{\lambda_r}{2}\mathbb{E}_{q_\phi}||\mathbf{v}_i^r - \mathbf{z}_i||^2 + \frac{\lambda_p}{2}||\mathbf{v}_i^p||^2 \right)$$

where q_ϕ is a shorthand for $q_\phi(Z|\mathcal{G})$, and $\mathbb{E}_{q_\phi(Z|\mathcal{G})}[Z]$ is equal to $\mu_\phi(\mathcal{G})$ output by the inference network. To optimize this objective, we employ block coordinate ascent with two sets of variables, $\{f_{\text{ENC}}(G), f_{\text{DEC}}(Z)\}$ and $\{V_S, V_R, V_P, G_{\text{DIFF}}\}$. As illustrated in Alg 1, each iteration of the algorithm proceeds in two steps, by alternating optimization over the social network and diffusion cascades.

4.4.2 ***Diffusion Prediction***. After learning the (locally) optimal model parameters and MAP estimates of latent variables, the likelihood of influencing user v_j given seed activations I is given by:

$$p(v_j|I) = \sigma(h_I^T \mathbf{v}_j^r) \quad (10)$$

4.4.3 ***Complexity***. The cost per iteration comprises two parts: (a) optimizing over social network $\mathcal{G}$ gives $O(|\mathcal{E}| \cdot F^2 + |\mathcal{E}| \cdot D)$ assuming GCN + Inner Product (b) optimizing over diffusion episodes is $O(|\mathbb{T}| \cdot D \cdot N)$ where F is the maximum layer dimension in f_{ENC}. The overall complexity per iteration is $O(|\mathcal{E}| \cdot F^2 + |\mathcal{E}| \cdot D + |\mathbb{T}| \cdot D \cdot N)$.

	Social Networks		Stack-Exchange Networks		
Dataset	**Digg**	**Weibo**	**Android**	**Christianity**	**Travel**
# Users	8,602	5,000	9,958	2,897	8,726
# Links	173,489	123,691	48,573	35,624	76,555
# Cascades	968	23,475	679	589	711
Avg. cascade len	100.0	23.6	33.3	22.9	26.8

Table 2: Statistics of datasets used in our experiments

5 EXPERIMENTS

In this section, we present our experimental results on multiple datasets from real-world social networks and public Stack-Exchanges[1]. We examine two popular social networks Digg and Weibo.

- **Digg** [15]: A social platform where users vote on news stories. The sequence of votes on each story constitutes a diffusion cascade, while the social network comprises friendship links among voters. We retain only users who have voted on at least 40 stories.
- **Weibo** [48]: A Chinese micro-blogging platform, where the social network consists of follower links, and cascades reflect retweeting behavior. We choose the 5000 most popular users.

Stack-Exchanges: Community Q&A websites where users post questions and answers on a wide range of topics. The inter-user knowledge-exchanges on various interaction channels (*e.g.*, question, answer, comment, upvote, etc.), constitute the social network. Cascades correspond to chronologically ordered series of posts associated with the same tag, *e.g.*, "google-pixel-2" on Android. We choose three Stack-Exchanges, Android, Christianity and Travel, spanning diverse themes. Dataset statistics are provided in Table 2.

5.1 Baselines

We compare Inf-VAE against state-of-the-art representation learning methods for diffusion prediction since they have been shown to significantly outperform classical models (*e.g.*, IC and LT) [9, 41].

- **CDK** [6]: an embedding method that models information spread as a heat diffusion process in the representation space of users.
- **Emb-IC** [7]: an embedded cascade model that generalizes IC to learn user representations from partial orders of user activations.
- **Inf2vec** [9]: an influence embedding method that combines local propagation structure and user co-occurrence in cascades.
- **DeepDiffuse** [17]: an attention-based RNN that operates on just the sequence of previously influenced users, to predict diffusion.
- **CYAN-RNN** [43]: a sequence-based RNN that uses an attention mechanism to capture cross-dependence among seed users.
- **SNIDSA** [44]: an RNN-based model to compute structure attention over the local propagation structure of a cascade.
- **Topo-LSTM** [41]: a recurrent model that exploits the local propagation structure of a cascade through a dynamic DAG-LSTM.

5.2 Experimental Setup

We denote our two model variants with GCN and MLP architectures, by Inf-VAE+GCN and Inf-VAE+MLP respectively. We randomly sample 70% of the cascades for training, 10% for validation and remaining 20% for testing. We consider the task of predicting the set of all influenced users as a retrieval problem [7, 9, 41, 43]. The fraction of users sampled from each test cascade to serve as the seed set is defined as ***seed set percentage***, which is varied from 10% to 50% to create a large evaluation test-bed spanning diverse cascade lengths. The likelihood of influencing an inactive user determines its rank (Eqn. 10). We use MAP@K (Mean Average Precision) and Recall@K as evaluation metrics. Note that MAP@K considers both the *existence* and *position* of ground-truth target users in the rank list, while Recall@K only reports occurrence within top-K ranks.

Hyper-parameters are tuned by evaluating MAP@10 on the validation set. Since Emb-IC generalizes IC, we use 1000 Monte Carlo simulations to estimate influence probabilities. Since the recurrent neural models (*e.g.,* Topo-LSTM) are trained for next user prediction, we use the ranking induced by user activation probabilities for diffusion prediction, which we found to significantly outperform a similar simulation approach. For Inf2vec, we examine several seed influence aggregation functions (Ave, Sum, Max, and Latest) to report the best results. Our reported results are averaged over 10 independent runs with different random weight initializations. Our implementation of Inf-VAE is publicly available[2].

5.3 Experimental Results

We note the following key observations from our experimental results comparing Inf-VAE against competing baselines (Table 3).

Methods that do not explicitly model sequential activation orders (*e.g.*, CDK and Emb-IC), perform markedly worse than their counterparts. Modeling local projected cascade structures with neural recurrent models results in improvements (*e.g.*, Topo-LSTM and others). Jointly modeling social homophily derived from global network structure and temporal influence by our model Inf-VAE yields significant relative gains of 22% (*MAP*@10) on average across all datasets. Inf-VAE+GCN consistently beats the MLP variant, validating the power of graph convolutional networks in effectively propagating higher-order local neighborhood features.

Figure 2 depicts the variation in recall with size of rank list K. As expected, recall increases with K, however, the relative differences across methods is much smaller. Inf-VAE consistently outperforms baselines across a wide range of K values. For instance, the Christianity dataset has seed sets with 2-10 users, and corresponding target sets with 10-15 users out of a possible 3000. Here, a recall@100 of 0.45 for Inf-VAE is quite impressive, especially considering the absence of explicit re-share links and the noise associated with real-world diffusion processes. We restrict our remaining analyses to Inf-VAE+GCN since it consistently beats the MLP variant.

5.4 Impact of Social and Behavior Sparsity

In this section, we analyze the benefits of explicitly modeling social homophily through VAEs, in comparison to the best baseline (Topo-LSTM) that only considers local propagation structures.

- **Users with sparse diffusion activities.** We divide users into quartiles by their *activity levels*, which is the number of participating cascades per user. We evaluate *target recall*@100 for each user u, defined as the fraction of times u was predicted correctly within top-100 ranks. In Figure 3(a), we depict both recall scores and relative gains (over Topo-LSTM) across activity quartiles.

[1] https://archive.org/details/stackexchange

[2] https://github.com/aravindsankar28/Inf-VAE

Method	**Digg**			**Weibo**			**Android**			**Christianity**			**Travel**		
MAP	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**	**@10**	**@50**	**@100**
CDK	0.0437	0.0222	0.0228	0.0130	0.0106	0.0123	0.0319	0.0121	0.0125	0.0876	0.0531	0.0578	0.0650	0.0333	0.0341
Emb-IC	0.0862	0.0431	0.0431	0.0140	0.0116	0.0131	0.0505	0.0248	0.0267	0.1340	0.0905	0.0962	0.0924	0.0584	0.0609
Inf2vec	0.1189	0.0554	0.0546	0.0156	0.0103	0.0121	0.0412	0.0141	0.0150	0.1824	0.0790	0.0852	0.1245	0.0495	0.0529
DeepDiffuse	0.0919	0.0460	0.0471	0.0291	0.0186	0.0213	0.0437	0.0228	0.0250	0.1632	0.0828	0.0831	0.1220	0.0675	0.0693
CYAN-RNN	0.1188	0.0479	0.0427	0.0296	0.0207	0.0234	0.0520	0.0276	0.0296	0.1971	0.1229	0.1304	0.1551	0.0791	0.0799
SNIDSA	0.0941	0.0363	0.0348	0.0224	0.0146	0.0169	0.0397	0.0207	0.0222	0.1233	0.0699	0.0781	0.0857	0.0562	0.0585
Topo-LSTM	0.1193	0.0577	0.0587	0.0325	0.0226	0.0247	0.0595	0.0283	0.0289	0.1811	0.0989	0.0991	0.1393	0.0773	0.0783
Inf-VAE+MLP	0.1587	0.0774	0.0719	0.0322	0.0211	0.0234	0.0584	0.0272	0.0285	0.2549	0.1355	0.1402	0.1865	0.0897	**0.0913**
Inf-VAE+GCN	**0.1642**	**0.0779**	**0.0724**	**0.0373**	**0.0230**	**0.0257**	**0.0601**	**0.0290**	**0.0304**	**0.2594**	**0.1413**	**0.1461**	**0.1924**	**0.0906**	0.0910

Table 3: Experimental results for diffusion prediction on 5 datasets (*MAP@K* scores for $K = 10, 50$ and 100), the *seed set percentage* varies in the range to 10 to 50% users in each test cascade. 22% relative gains in MAP@10 (on average) over the best baseline.

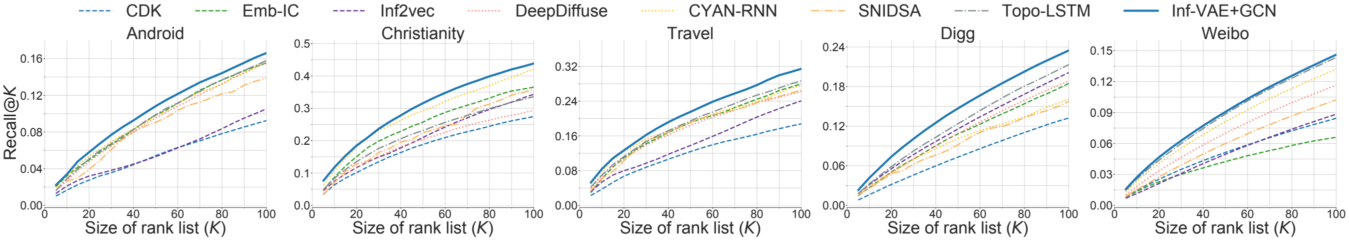


Figure 2: Experimental results for diffusion prediction on 5 datasets, Recall@K scores on varying size of the rank list K

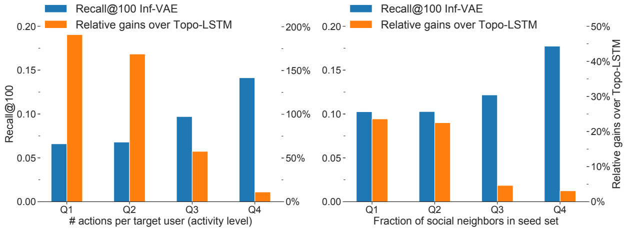


Figure 3: Performance across user quartiles on *diffusion activity level*, and *seed neighbor fraction* (Q1: lowest, Q4: highest). Inf-VAE has higher gains for users with sparse activities and lacking direct neighbors in seed sets (quartiles Q1-Q3).

While target recall increases with activity levels, Inf-VAE significantly improves performance for inactive users (quartiles Q1-Q3). Thus, modeling social homophily through VAEs contributes to massive gains for users with *sparse diffusion activities.* Interestingly, Topo-LSTM performs comparably on the most active users (quartile Q4), which indicates the potential of purely local sequential modeling techniques for highly active users.

- **Users that lack direct social neighbors in seed sets.** We separate users into quartiles by *seed neighbor fraction*, which is computed as the fraction of seed users that are direct social neighbors, averaged over the training examples. We similarly report target recall@K and relative gains across quartiles (Figure 3(b)).
  As expected, performance increases with seed neighbor fraction. Note higher relative gains over Topo-LSTM for users that lack direct neighbors in the seed set (quartiles Q1-Q3). This demonstrates the ability of Inf-VAE to implicitly regularize seed user representations based on higher-order social neighborhoods captured by GCN-based autoencoders. Again, we find that local sequential models suffice for users with large seed neighbor fractions, as evidenced by the results of Topo-LSTM in quartile Q4.

Metric		**Weibo**			**Android**	
MAP	@10	@50	@100	@10	@50	@100
(0) Default	**0.0373**	**0.0230**	**0.0257**	**0.0601**	**0.0290**	**0.0304**
(1)$V_S = V_R \not\perp\!\!\!\perp Z$	0.0353	0.0220	0.0248	0.0558	0.0275	0.0287
(2)$V_S \perp\!\!\!\perp Z$	0.0351	0.0213	0.0240	0.0595	0.0285	0.0301
(3)$V_R \perp\!\!\!\perp Z$	0.0326	0.0217	0.0241	0.0567	0.0276	0.0291
(4)$V_S \perp\!\!\!\perp Z, V_R \perp\!\!\!\perp Z$	0.0313	0.0205	0.0235	0.0542	0.0274	0.0289
(5) Remove Coattention	0.0307	0.0207	0.0233	0.0553	0.0270	0.0284
(6) Separate Attentions	0.0293	0.0217	0.0192	0.0570	0.0277	0.0291
(7) Static-Pretrain	0.0342	0.0203	0.0226	0.0606	0.0281	0.0292

Table 4: Ablation study on architecture design (*MAP@K* scores for $K = 10, 50, 100$), $\perp\!\!\!\perp$ denotes variable independence

5.5 Ablation Study and Sensitivity Analysis

In this section, we first present an *ablation study* followed by a sensitivity analysis on *seed set percentage* and *hyper-parameters.*

5.5.1 ***Ablation Study.*** We analyze model design choices including homophily via VAEs and co-attention, in Android and Weibo.

Social Homophily: We examine ways to condition V_S, V_R on Z:

(1) V_S and V_R are identical and are conditioned on Z through hyper-parameter $\lambda_s (= \lambda_r)$, *i.e.*, $V_S = V_R \not\perp\!\!\!\perp Z$ (note that this is different from setting $\lambda_s = \lambda_r$ without enforcing $V_S = V_R$).
(2) V_S is a free variable conditionally independent of Z, *i.e.*, $V_S \perp\!\!\!\perp Z$, which is equivalent to setting $\lambda_s = 0$.
(3) V_R is a free variable, *i.e.*, $V_R \perp\!\!\!\perp Z$, which is the inverse of (3).
(4) V_S and V_R are both free variables conditionally independent of Z ($\lambda_s = \lambda_r = 0$), *i.e.*, $V_S \perp\!\!\!\perp Z, V_R \perp\!\!\!\perp Z$.

Independent conditioning of V_S and V_R on Z (default) achieves best results. Enforcing $V_S = V_R$ (row 1) is clearly inferior, which validates the choice of differentiating user roles. Notably, allowing V_S to be a free variable results in minor performance degradation

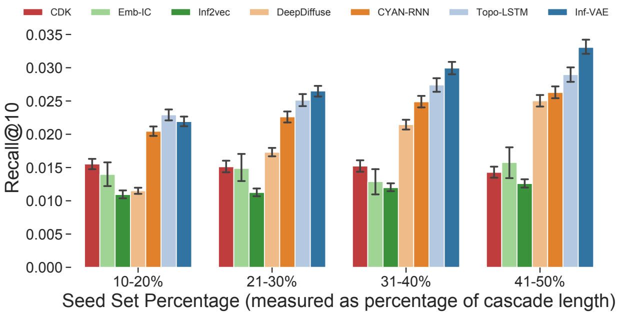


Figure 4: Impact of seed set percentage in Weibo. Inf-VAE achieves higher gains for larger seed set fractions.

(row 2), while the drop is significant when V_R is independent of Z (row 3). As expected, setting both V_S and V_R as free variables (row 4), performs the worst due to lack of social homophily signals.

Co-attention: We conduct two ablation studies defined by:

(5) Replace co-attention with meanpool over concatenated sender and temporal influence vectors, followed by a dense layer.

(6) Replace co-attention with two separate attentions on the sender and temporal influence sequences, followed by concatenation.

Learning co-attentional weights (default) consistently outperforms mean pooling (5), illustrating the benefits of assigning variable contributions to seed users. Using separate attentions (6) significantly deteriorates results, which indicates the existence of complex nonlinear correlations between the social and temporal latent factors.

Joint Training: In (2), we replace joint block-coordinate optimization (Alg 1) with a single step over cascades with pre-trained user embeddings (line 2), *i.e.*, Z is not updated based on cascades. Joint training is beneficial when social interactions are noisy (*e.g.*, Weibo) in comparison to focused stack-exchanges such as Android.

5.5.2 *Impact of Seed Set Percentage.*

We divide the test set into quartiles based on *seed set percentage*, and report performance per quartile. Since we require a sizable number of test examples per quartile to obtain unbiased estimates, we use the Weibo dataset.

Figure 4 depicts Recall@10 scores in different ranges. First, recall scores increase with seed set percentage since larger seed sets enable better model predictions; and target set size reduces with increase in seed set percentage. Second, relative gains of Inf-VAE over baselines increase with seed set percentage. This highlights the capability of co-attention in focusing on relevant users based on both social homophily and temporal influence factors.

5.5.3 *Impact of λ_s and λ_r.*

Hyper-parameters λ_s and λ_r control the degree of dependence of the sender and receiver variables V_S, V_R on the social variables Z. Figure 5 depicts performance (*MAP*@10) on Android and Weibo datasets. The performance is sensitive to variations in λ_r with best values around 0.01 and 0.1, while λ_s results in minimal variations. Furthermore, the best values of λ_s, λ_r are stable in a broad range of values that transfer across datasets, indicating that Inf-VAE requires minimal tuning in practice. Since λ_p has minimal performance impact, we exclude it from our analysis.

5.5.4 *Runtime Analysis.*

In our experiments, all methods converge within 50 epochs with similar convergence rates. For the sake of brevity, we only compare runtime per epoch, which includes one step over the social network and cascades for Inf-VAE.

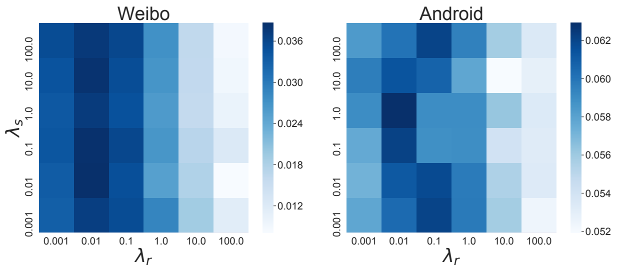
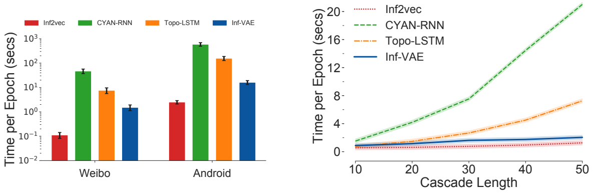


Figure 5: *MAP*@10 on varying λ_s, λ_r over Android and Weibo. Performance is more sensitive to variations in λ_r than λ_s.



(a) Runtime on Weibo and Android (b) Scalability on synthetic dataset

Figure 6: Running time and scalability comparison of Inf-VAE with several baselines. Inf-VAE is faster than recurrent models (Topo-LSTM, CYANRNN) by an order of magnitude.

From figure 6(a), Inf2vec is the fastest while Inf-VAE comes second. Thus, Inf-VAE achieves a good trade-off between expensive recurrent models (*e.g.*, Topo-LSTM) and simpler embedding methods (*e.g.*, Inf2vec), with consistently superior results.

5.5.5 *Scalability Analysis.*

We analyze scalability on cascade sequences of varying lengths. Since real-world datasets possess heavily biased length distributions, we instead synthetically generate a Barabasi-Albert [1] network of 2000 users and simulate diffusion cascades using an IC model. We compare training times per epoch for each cascade length (l) in the range of 10 to 50.

Figure 6(b) depicts linear scaling for Inf-VAE and Inf2vec *wrt* cascade length. Recurrent methods scale poorly due to the sequential nature of back-propagation through time (BPTT), resulting in prohibitive costs for long cascade sequences. On the other hand, Inf-VAE avoids BPTT through efficient parallelizable co-attentions.

6 CONCLUSION

In this paper, we present a novel variational autoencoder framework (Inf-VAE) to jointly embed homophily and influence in diffusion prediction. Given a sequence of seed user activations, Inf-VAE employs an expressive co-attentive fusion mechanism to jointly attend over their social and temporal variables, capturing complex correlations. Our experimental results on two social networks and three stack-exchanges indicate significant gains over state-of-the-art methods.

In future, Inf-VAE can be extended to include multi-faceted user attributes owing to the generalizable nature of our VAE framework. While the current implementation employs GCN networks, we foresee direct extensions with neighborhood sampling [13] to enable scalability to social networks with millions of users. We also plan to explore neural point processes to predict user activation times. Finally, similar frameworks may be examined for joint temporal co-evolution of social network and diffusion cascades.

7 ACKNOWLEDGEMENTS

Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreements No. W911NF-17-C-0099 and FA8750-19-2-1004, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, and DTRA HDTRA11810026.

REFERENCES

[1] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

[3] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2012. Topic-aware social influence propagation models. In *ICDM*. IEEE, 81–90.

[4] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2013. Influence-based network-oblivious community detection. In *ICDM*. IEEE, 955–960.

[5] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[6] Simon Bourigault, Cedric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. 2014. Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 393–402.

[7] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. 2016. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 573–582.

[8] Robert B Cialdini and Noah J Goldstein. 2004. Social influence: Compliance and conformity. *Annu. Rev. Psychol.* 55 (2004), 591–621.

[9] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. 2018. Inf2vec: Latent Representation Model for Social Influence Embedding. In *ICDE*. IEEE, 941–952.

[10] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. 2012. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5, 4 (2012), 21.

[11] Mark Granovetter. 1978. Threshold models of collective behavior. *American journal of sociology* 83, 6 (1978), 1420–1443.

[12] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[14] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[15] Tad Hogg and Kristina Lerman. 2012. Social dynamics of digg. *EPJ Data Science* 1, 1 (2012), 5.

[16] Zhiting Hu, Junjie Yao, Bin Cui, and Eric Xing. 2015. Community level diffusion extraction. In *SIGMOD 2015*. ACM, 1555–1569.

[17] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B Aditya Prakash, and Naren Ramakrishnan. 2018. DeepDiffuse: Predicting the'Who'and'When'in Cascades. In *ICDM*. IEEE, 1055–1060.

[18] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 137–146.

[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[20] Thomas N Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *NIPS Workshop on Bayesian Deep Learning* (2016).

[21] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[22] Adit Krishnan, Hari Cheruvu, Cheng Tao, and Hari Sundaram. 2019. A Modular Adversarial Approach to Social Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 1753–1762.

[23] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1491–1494.

[24] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)* 1, 1 (2007), 5.

[25] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. DeepCas: An end-to-end predictor of information cascades. In *Proceedings of the 26th International Conference on World Wide Web*. 577–586.

[26] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. 689–698.

[27] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[28] Kanika Narang, Chaoqi Yang, Adit Krishnan, Junting Wang, Hari Sundaram, and Carolyn Sutter. 2019. An Induced Multi-Relational Framework for Answer Selection in Community Question Answer Platforms. *arXiv preprint arXiv:1911.06957* (2019).

[29] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[30] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018. DeepInf: Social Influence Prediction with Deep Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18)*.

[31] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. 2009. Learning continuous-time information diffusion model for social behavioral data analysis. *Advances in Machine Learning* (2009), 322–337.

[32] Kazumi Saito, Kouzou Ohara, Yuki Yamagishi, Masahiro Kimura, and Hiroshi Motoda. 2011. Learning diffusion probability based on node attributes in social networks. In *International Symposium on Methodologies for Intelligent Systems*. Springer, 153–162.

[33] Aravind Sankar, Adit Krishnan, Zongjian He, and Carl Yang. 2019. Rase: Relationship aware social embedding. In *IJCNN*. IEEE, 1–8.

[34] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2018. Dynamic Graph Representation Learning via Self-Attention Networks. *arXiv preprint arXiv:1812.09430* (2018).

[35] Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. 2017. Motif-based Convolutional Neural Network on Graphs. *arXiv preprint arXiv:1711.05697* (2017).

[36] Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. 2019. Meta-GNN: Metagraph Neural Network for Semi-supervised learning in Attributed Heterogeneous Information Networks. In *ASONAM*. IEEE, 137–144.

[37] Cosma Rohilla Shalizi and Andrew C Thomas. 2011. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research* 40, 2 (2011), 211–239.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[39] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

[40] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1225–1234.

[41] Jia Wang, Vincent W. Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. 2017. Topological Recurrent Neural Network for Diffusion Prediction. In *ICDM*. 475–484.

[42] Yongqing Wang, Huawei Shen, Shenghua Liu, and Xueqi Cheng. 2015. Learning User-Specific Latent Influence and Susceptibility from Information Cascades. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[43] Yongqing Wang, Huawei Shen, Shenghua Liu, Jinhua Gao, and Xueqi Cheng. 2017. Cascade Dynamics Modeling with Attention-based Recurrent Neural Network.. In *IJCAI*. 2985–2991.

[44] Zhitao Wang, Chengyao Chen, and Wenjie Li. 2018. A Sequential Neural Information Diffusion Model with Structure Attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1795–1798.

[45] Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *ICLR*.

[46] Jaewon Yang and Jure Leskovec. 2010. Modeling information diffusion in implicit networks. In *ICDM*. IEEE, 599–608.

[47] Yang Yang, Jie Tang, Cane Wing-ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. 2015. RAIN: Social Role-Aware Information Diffusion.. In *AAAI*. 367–373.

[48] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. 2013. Social influence locality for modeling retweeting behaviors. In *IJCAI*.

[49] Jing Zhang, Jie Tang, Yuanyi Zhong, Yuchen Mo, Juanzi Li, Guojie Song, Wendy Hall, and Jimeng Sun. 2017. StructInf: Mining Structural Influence from Social Streams.. In *AAAI*. 73–80.

[50] Yuan Zhang, Tianshu Lyu, and Yan Zhang. 2017. Hierarchical community-level information diffusion modeling in social networks. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 753–762.

[51] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1513–1522.