

# Source Localization for Cross Network Information Diffusion

Chen Ling\*  
Emory University  
Atlanta, United States  
chen.ling@emory.edu

Tanmoy Chowdhury\*  
Emory University  
Atlanta, United States  
tchowdh6@gmu.edu

Jie Ji  
Emory University  
Atlanta, United States  
jason.ji@emory.edu

Sirui Li  
Emory University  
Atlanta, United States  
sirui.li@emory.edu

Andreas Züfle  
Emory University  
Atlanta, United States  
azufle@emory.edu

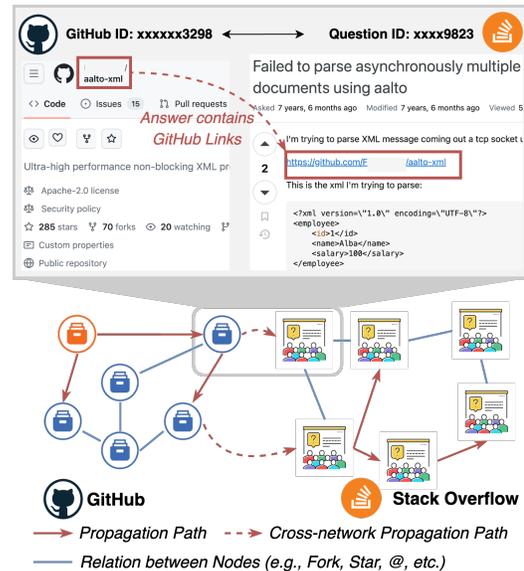
Liang Zhao  
Emory University  
Atlanta, United States  
liang.zhao@emory.edu

## Abstract

Source localization aims to locate information diffusion sources only given the diffusion observation, which has attracted extensive attention in the past few years. Existing methods are mostly tailored for single networks and may not be generalized to handle more complex networks like cross-networks. Cross-network is defined as two interconnected networks, where one network’s functionality depends on the other. Source localization on cross-networks entails locating diffusion sources on the source network by only giving the diffused observation in the target network. The task is challenging due to challenges including: 1) diffusion sources distribution modeling; 2) jointly considering both static and dynamic node features; and 3) heterogeneous diffusion patterns learning. In this work, we propose a novel method, namely *CNSL*, to handle the distribution of diffusion sources through Bayesian inference and leverage disentangled encoders to separately learn static and dynamic node features. The learning objective is coupled with the cross-network information propagation estimation model to make the inference of diffusion sources considering the overall diffusion process. Additionally, we also provide two novel cross-network datasets collected by ourselves. Extensive experiments are conducted on both datasets to demonstrate the effectiveness of *CNSL* in handling the source localization on cross-networks. The code are available at: <https://github.com/tanmoysr/CNSL/>

## 1 Introduction

Source localization aims at locating the origins of information diffusion within networks, which stands as a famous inverse problem to the estimation of information propagation. Source localization not only holds practical significance but also helps us grasp the intricate characteristics of network dynamics. By accurately identifying the sources of information propagation, we can significantly mitigate potential damages by cutting off critical pathways through which information, and potentially misinformation, spreads. Over the past years, existing works have made considerable efforts toward addressing this critical challenge. Earlier works [19, 26, 30, 31] leverage *rule-based* methods to locate diffusion sources under prescribed and known diffusion patterns. Furthermore, *Learning-based* methods [7, 15, 24, 25] were proposed to employ deep neural networks to encode neighborhood and graph topology information, which achieve state-of-the-art performance. These efforts underscore the importance of source localization in maintaining the integrity and reliability of information across the digital landscape.



**Figure 1: Example of misinformation propagation on cross-network between GitHub and Stack Overflow, where each node in the GitHub network denotes a repository, and each node in the Stack Overflow represents a discussion thread.**

Existing techniques for source localization have primarily been designed for single networks. However, much of today’s infrastructure is organized in the form of *cross-networks*. Communications between different communities, cross-country financial transactions, and systems of water and food supply can all be cross-networks, where the functionality or performance of one network depends on other networks. The presence of cross-networks has also made us vulnerable to various network risks that belong exclusively to cross-networks, such as the spreading of misinformation from one social media to another and safety alerts found in downstream stages of the food supply chain. The complexity of cross-network interactions is further illustrated by an incident involving a malicious GitHub repository, as detailed in the upper part of Figure 1 and identified in [10], which was linked to over 40 discussion threads on Stack Overflow. Questioners and less experienced users may be directed into using the alleged solution without maintaining a healthy skepticism. Once using the code from the malicious GitHub repository, the victims’ devices might be compromised (e.g., system operations being disrupted). The challenge of tracing the origins and pathways of such misinformation is exacerbated in cross-network environments, where the initial propagation occurs in a network

different from where the observations are made, as in the transition from GitHub to Stack Overflow. Furthermore, this complexity is compounded by multiple rounds of propagation and the possibility that contributors to Stack Overflow discussions may not intentionally disseminate misinformation, highlighting the need for more powerful source localization methodologies that account for cross-networks.

Cross-network source localization is defined as locating diffusion sources from the source network by only giving the diffused observation of the target network, which is still under-explored. The challenge primarily lies in the separation between the networks where the diffusion originates and where it is observed, making traditional source localization techniques less effective due to the following critical obstacles. 1) *The difficulty of characterizing the distribution of diffusion sources only given the diffused observation of another network.* Understanding the distribution of potential sources is crucial for understanding the nature of diffusion processes and for quantifying the inherent uncertainties associated with identifying these sources [3, 15, 16]. However, accurately learning the distribution of diffusion sources in a cross-network context requires the formulation of a conditional probability model that accounts for the observed diffusion within one network, given the structural and dynamical properties of another. This task requires fully considering different Network topologies, diverse node features, and varied propagation patterns, which makes the learning objective hard to model and optimize. 2) *The difficulty of jointly capturing dynamic and static features of the nodes in the cross-network.* Characterizing the distribution of diffusion sources is often conditioned on the intrinsic nature of the nodes and their connections. Existing works hardly leverage node features (e.g., text description and statistical node features) since entangling both types of features would lead to a high-dimensional and often intractable distribution of diffusion sources. Moreover, the nodes from different networks may have different intrinsic characteristics that help profile their diffusion dynamics and can predominantly help locate the sources. 3) *The difficulty of jointly capturing the heterogeneous diffusion patterns of the cross-network.* Besides the difficulty of learning the distribution of diffusion sources, the diffusion patterns in both networks are also unknown to us. By correctly modeling the overall diffusion process, it is also essential to jointly consider the different propagation patterns of both networks. Additionally, the communication between different networks (i.e., cross-network propagation paths as noted in Figure 1) also cannot be ignored.

In this work, we propose the Cross-Network Source Localization (CNSL) method for locating the diffusion sources from a source network given its diffused observation from another target network under arbitrary diffusion patterns. Specifically, for the first challenge, we design a novel framework to approximate the distribution of diffusion sources by mean-field variational inference. For the second challenge, we propose a disentangled generative prior to encoding both static and dynamic features of nodes. For the last challenge, we model the unique diffusion dynamics of each network separately and integrate the learning process of these information diffusion models with the approximation of diffusion source distribution. This ensures accurate reconstruction of diffusion sources considering the specific propagation mechanisms of each network. We summarize our major contributions of this work as follows:

- **Problem.** We design a novel formulation of the cross-network source localization and propose to leverage deep generative models to characterize the prior and approximate the distribution of diffusion sources via variational inference.
- **Technique.** We propose a unified framework to jointly capture 1) both static and dynamic node features, and 2) the heterogeneous diffusion patterns of both networks. The approximation of diffusion sources is fully aware of various node features and the interplay of cross-network information diffusion patterns.
- **Data.** Cross-network source localization lacks high-quality data, which is highly difficult to craft. We collect and curate a real-world dataset that accounts for the Cross-platform Communication Network, which records the real-world misinformation propagation from Github to Stack Overflow. We also provide a simulated cross-network dataset using agent-based simulation to disseminate misinformation across physical and social networks.
- **Experiments.** We conduct experiments against state-of-the-art methods designed originally for single-network source localization. Results show substantially improved performance of our method for cross-network source localization.

## 2 Related Works

**Information Source Localization.** Diffusion source localization is defined as inferring the initial diffusion sources given the current diffused observation, which has attracted many applications, ranging from identifying rumor sources in social networks [9] to finding blackout origins in smart grids [22]. Early approaches [19, 26, 31, 32] focused on identifying the single/multiple source of an online disease under the Susceptible-Infected (SI) or Susceptible-Infected-Recover (SIR) diffusion patterns with either full or partial observation. Later on, Dong et al. [7] further leverage GNN to enhance the prediction accuracy of LPSI. However, existing diffusion source localization methods cannot well quantify the uncertainty between different diffusion source candidates, and they usually require searching over the high-dimensional graph topology and node attributes to detect the sources, both drawbacks limit their effectiveness and efficiency. Moreover, in the past few years, more methods [15, 20, 24, 25, 27] have been proposed to address the dependency of prescribed diffusion models and characterize the latent distribution of diffusion sources, which have achieved state-of-the-art results. However, their methods still may not generalize to cross-network source localization due to the unique interconnected structure.

**Information Diffusion on Cross Network.** The interconnection between cross-networks allows information to flow seamlessly from one platform to another through overlapping nodes. However, it is important to note that the patterns of influence and information propagation differ between various networks and can even vary within the same network. Recent studies in information diffusion across interconnected networks have made notable advancements. Earlier works [4, 11, 17, 28] have developed different frameworks for correct modeling of the information flow within different network formats, such as wireless networks, social networks, and supply chains. Later on, many works have been proposed to study different features and applications of cross-networks, e.g., mitigating cascading failures [8, 23]. However, until today, there are few works

[6, 14] trying to correctly model the information diffusion pattern in the interconnected network system.

### 3 Cross-network Diffusion Source Localization

In this section, the problem formulation is first provided before deriving the overall objective from the perspective of the divergence-based variational inference. A novel optimization algorithm is then proposed to infer the seed nodes given the observed cross-network diffused pattern.

#### 3.1 Problem Formulation

Cross-network  $\mathcal{G} = (G_s, G_t)$  consists of a *Source Network*  $G_s = (V_s, E_s)$  and a *Target Network*  $G_t = (V_t, E_t)$ . Both  $G_s$  and  $G_t$  are composed of a set of vertices  $V_s$  and  $V_t$  corresponding to individual users of the network as well as a set of edges  $E_s \subseteq V_s \times V_s$  and  $E_t \subseteq V_t \times V_t$  denote connecting pairs of users in both networks, respectively. In addition,  $f_s \in \mathbb{R}^{N_s \times F}$  and  $f_t \in \mathbb{R}^{N_t \times F}$  and denote the static features of both networks (e.g., associated text embedding, user age, social relations, etc.), where  $F$  denotes the dimension of the node feature, and  $N_t, N_s$  denote the number of nodes in each network, respectively. To connect the cross-network  $\mathcal{G}$ , there exists a set of bridge links between  $G_s$  and  $G_t$  denoted by  $L = \{(v_s, v_t) | v_s \in V_s, v_t \in V_t\}$ , which represent the propagation paths from  $G_s$  to  $G_t$ .

The information propagation in the cross-network is a *one-directional* message passing from  $G_s$  to  $G_t$ . More specifically, the propagation initiates from a group of nodes denoted as  $x_s \in \{0, 1\}^{N_s}$  in the source network  $G_s$ , where each entry has a binary value representing whether the node is seed or not. After a certain period, the information propagates from  $G_s$  to  $G_t$  and infects a portion of nodes in  $G_t$  through the bridge links  $L$ . We use  $y_t \in [0, 1]^{N_t}$  to denote the infection probability of each node in  $G_t$ .

**Problem 1 (Cross-network diffusion source localization).** Given  $\mathcal{G}$  and the diffused observation of the target network  $y_t$ , the problem of diffusion source localization in cross-networks (i.e., the inverse problem of diffusion estimation) requires finding  $\tilde{x}_s \in \{0, 1\}^{N_s}$  from the source network  $G_s$ , such that the empirical loss  $\|\tilde{x}_s - x_s\|_2^2$  is minimized, under the constraint that the diffused observation in the target graph  $y_t$  could be generated from  $\tilde{x}_s$  through  $L$ .

However, reconstructing  $\tilde{x}_s$  from  $y_t$  is difficult due to the following challenges. 1) *The difficulty of characterizing the distribution of seed nodes in the cross-network scenario.* To consider all possibilities of the seed nodes in cross-network source localization, it's desired to model the distribution of seed nodes  $p(x_s)$  by characterizing the conditional probability  $p(x_s|y_t)$ . However, learning  $p(x_s|y_t)$  requires jointly considering the topology structure of the cross-network  $\mathcal{G}$  as well as the stochastic diffusion pattern through bridge links  $L$ . Existing works cannot be directly adapted due to the incapability of considering the complex cross-network scenario. 2) *The difficulty of jointly capturing dynamic and static features of the nodes in the cross-network.* The intrinsic patterns of the seed nodes consist of both dynamic patterns (i.e., the choice of seed nodes  $x_s$ ) and static patterns (e.g., node features  $f_s$ ). The correlated factors lead to the high-dimensional and often intractable distribution  $p(x_s)$ , which makes maximizing the joint likelihood  $p(x_s, y_t)$  to be hard and computationally inefficient. 3) *The difficulty of jointly capturing the heterogeneous diffusion patterns of the cross-network.*

The underlying diffusion process from  $x_s$  to  $y_t$  is not only affected by numerous factors (e.g., the infectiousness of the misinformation and the immunity power of the user), but the propagation patterns in the cross-network are inherently different in different networks.

#### 3.2 Latent Distribution Learning of Seed Nodes

To cope with the first challenge of characterizing the distribution of diffusion sources in the cross-network, we propose to utilize graph topology as well as the diffused observation to define the conditional probability  $p(x_s|y_t)$ . Since the diffused observation  $y_t$  is conditioned on both networks  $\mathcal{G}$  as well as the diffusion source  $x_s$ , we derive a conditional probability  $p(y_t|x_s, \mathcal{G}) \cdot p(x_s)$ , where  $p(x_s)$  is the distribution of infection sources within  $G_s$ . To estimate the optimal diffusion source  $\tilde{x}_s$ , we employ the Maximum A Posteriori (MAP) approximation by maximizing the following probability:

$$\tilde{x}_s = \arg \max_{x_s} p(y_t|x_s, \mathcal{G}) \cdot p(x_s) = \arg \max_{x_s} p(x_s, y_t|\mathcal{G}).$$

However, since  $p(x_s)$  is often intractable and entangles both static and dynamic features, we instead leverage deep generative models to characterize the implicit distribution of  $p(x_s)$ .

To tackle the second challenge of jointly considering all static and dynamic node features, we propose a disentangled generative model to map the intractable and potentially high-dimensional  $p(x_s)$  to latent embeddings in low-dimensional latent space. Specifically, we aim to learn the conditional distribution  $p(x_s, y_t, \mathcal{G}|z_s, z_{f_s})$  of  $x_s$  given two latent variables  $z_s$  and  $z_{f_s}$ . Specifically,  $z_s \in \mathbb{R}^{k_1}$  ( $k_1 \ll N_s$ ) and  $z_{f_s} \in \mathbb{R}^{k_2}$  ( $k_2 \ll N_s$ ) are obtained by an approximate posterior  $p(z_s, z_{f_s}|x_s, y_t, \mathcal{G})$ , where  $p(z_s, z_{f_s})$  is the prior distribution of node's dynamic and static features. Note that  $k_1$  and  $k_2$  are the numbers of variables in each group, in order to capture the different types of semantic factors.

The goal here is to learn the conditional distribution of  $p(x_s)$  given  $Z = (z_s, z_{f_s})$ , namely, to maximize the marginal likelihood of the observed cross-network diffusion in expectation over the distribution of the latent variable set  $Z$  as  $\mathbb{E}_{p_\theta(Z)} [p_\theta(x_s, y_t, \mathcal{G}|Z)]$ . For a given observation of the information diffusion in the cross-network, the prior distribution of the latent representations  $p(Z)$  is still intractable to infer. We propose solving it based on variational inference, where the posterior needs to be approximated by the distribution  $q_\phi(Z|x_s, y_t, f_s, \mathcal{G})$ . In this way, the goal becomes to minimize the Kullback–Leibler (KL) divergence between the true prior and the approximate posterior. Moreover, we assume  $z_s$  and  $z_{f_s}$  capture different semantic factors. Specifically,  $z_s$  is required to capture just the independent dynamic semantic factors of which nodes are infection sources, and  $z_{f_s}$  is required to capture the correlated semantic factors considering both dynamic features and static node features. To encourage this disentangling property of both posteriors, we introduce a constraint by trying to match the inferred posterior configurations of the latent factors to the prior  $p(z_s, z_{f_s})$  by setting each prior to being an isotropic unit Gaussian  $N(0, 1)$ , leading to the constrained optimization problem as:

$$\begin{aligned} \max_{\theta, \phi} \quad & \mathbb{E}_{q_\phi(z_s, z_{f_s}|x_s, y_t, \mathcal{G})} \left[ p_\theta(x_s, y_t, \mathcal{G}|z_s, z_{f_s}) \right], \\ \text{s.t.} \quad & \text{KL} [q_\phi(z_s, z_{f_s}|x_s, f_s, y_t, \mathcal{G}) || p(z_s, z_{f_s})] < I. \end{aligned}$$

Furthermore, given the assumption that  $p(z_s)$  represents the distribution of dynamic node features and  $p(z_{f_s})$  denotes the distribution of joint node features (entangles with both static and dynamic features), the constraint term can be decomposed as:

$$q_\phi(z_s, z_{f_s}|x_s, f_s, y_t, \mathcal{G}) = q_{\phi_1}(z_s|x_s, y_t, \mathcal{G}) \cdot q_{\phi_2}(z_{f_s}|y_t, f_s, x_s, \mathcal{G})$$

Then the objective function can be written as:

$$\begin{aligned} \max_{\theta, \phi} \quad & \mathbb{E}_{q_\phi(z_s, z_{f_s}|x_s, y_t, \mathcal{G})} \left[ p_\theta(x_s, y_t, \mathcal{G}|z_s, z_{f_s}) \right], \quad (1) \\ \text{s.t.} \quad & KL[q_{\phi_1}(z_s|x_s, y_t, \mathcal{G})||p(z_s)] < I_s, \\ & KL[q_{\phi_2}(z_{f_s}|y_t, x_s, \mathcal{G})||p(z_{f_s})] < I_{f_s}, \end{aligned}$$

where we decompose  $I$  into two separate parts (i.e.,  $I_s$  and  $I_{f_s}$ ) of the information capacity to control each group of latent variables so that the variables inside each group of latent variables are disentangled. In practice,  $q_{\phi_1}(\cdot)$  and  $q_{\phi_2}(\cdot)$  are implemented as two encoders with multi-layer perceptron structure. More details can be found in Figure 2.

### 3.3 Cross Network Diffusion Model Learning

To address the third challenge, i.e., making the source localization be aware of the heterogeneous diffusion patterns between networks, locating diffusion origins  $x_s$  may not only involve estimating the distribution of seed nodes but the process should also be determined by correctly modeling the information diffusion across diverse and interlinked network structures  $\mathcal{G}$ . In the context of cross-network information diffusion, the diffused observation  $y_t$  is determined by the diffusion source  $x_s$  under the cross-network  $\mathcal{G}$  through bridge links  $L$ . Therefore, the conditional distribution  $p_\theta(x_s, y_t, x_f, \mathcal{G}|z_s, z_{f_s})$  can further be decoupled as:

$$\log p_\theta(x_s, y_t, x_f, \mathcal{G}|z_s, z_{f_s}) = \log[p_\psi(y_t|x_s, \mathcal{G})] + \log[p_\theta(x_s|z_s, z_{f_s})],$$

where  $p_\psi(\cdot)$  models the probability of the infection status  $y_t$  of nodes in  $G_t$  given seed nodes  $x_s$  in  $G_s$ . Moreover, the second term  $p_\theta(x_s|z_s, z_{f_s})$  reveals that the latent variables  $Z$  only encodes information from  $x$  (i.e.,  $y_t \perp Z|x_s, \mathcal{G}$ ). According to the assumption, we could also simplify both encoders as  $q_{\phi_1}(z_s|x_s, \mathcal{G})$  and  $q_{\phi_2}(z_{f_s}|x_s, f_s, \mathcal{G})$  in Eq. (1) by removing  $y_t$  from the input.

**Cross-network Information Propagation.** Modeling the diffusion from  $x_s$  to  $y_t$  is complex due to multiple influencing factors, such as misinformation’s infectiousness and the distinct propagation patterns across networks like GitHub and Stack Overflow, which cater to different user communities. The unknown nature of these diffusion patterns prevents the use of standard models like Linear Threshold or Independent Cascade. This complexity underlines the need to decompose and simplify  $p_\psi(y_t|x_s, \mathcal{G})$  to analyze the diverse diffusion behaviors in  $G_s$  and  $G_t$  through a learning approach:

$$\log p_\psi(y_t|x_s, \mathcal{G}) = \log p_{\psi_1}(y_s|x_s, G_s) + \log p_{\psi_2}(y_t|y_s, x_t, G_t). \quad (2)$$

In this simplified decomposition,  $p_{\psi_1}(\cdot)$  characterizes the diffusion pattern of  $G_s$  given the seed nodes  $x_s$ , which is independent of the information propagation in  $G_t$ .  $y_s \in [0, 1]^{N_s}$  records the infection status of all nodes in the source network  $G_s$ . When the diffusion is complete in  $G_s$ , the infection probability is directly transferred to the target network  $G_t$  through bridge links  $L$  so that some nodes

in  $G_t$  have initial infection status (denoted as  $x_t$ ) to initiate the infection process in  $G_t$ . The propagation in  $G_t$  is then modeled by  $p_{\psi_2}(y_t|y_s, x_t, G_t)$  by taking the graph structure  $G_t$  and initial seed infection probability  $x_t$  as inputs. More details of the derivation is provided in the Appendix.

**Monotonic Constraint on Information Diffusion.** The information diffusion on the regular network is often regularized by the monotone increasing property [5, 15]. In this work, we also assume the same monotonic property holds in the cross-network information diffusion, namely  $y_t^{(i)} \geq y_t^{(j)}$ ,  $\forall x_s^{(i)} \supseteq x_s^{(j)}$ . Specifically, selecting more seed nodes in  $G_s$  would result in a generally higher (or at least equal) infection probability of nodes in  $G_s$  according to the property of diminishing returns. Subsequently, the bridge links would transfer the infection probability from  $y_s$  to  $x_t$ , and similarly, the probability of each node being infected in  $y_t^{(i)}$  (estimated from  $x_t^{(i)}$ ) should be greater or equal to  $y_t^{(j)}$  (estimated from  $x_t^{(j)}$ ), such that  $y_t^{(i)} \geq y_t^{(j)}$ . Therefore, owing to the monotonic increasing property of the information diffusion, we add the constraint  $\lambda \|\max(0, y_t^{(j)} - y_t^{(i)})\|_2^2, \forall x_s^{(i)} \supseteq x_s^{(j)}$ , to Eq. (1), where we transform the inequality constraint into its augmented Lagrangian form to minimize  $\|\max(0, y_t^{(j)} - y_t^{(i)})\|_2^2$  and  $\lambda > 0$  denotes regularization hyperparameter.

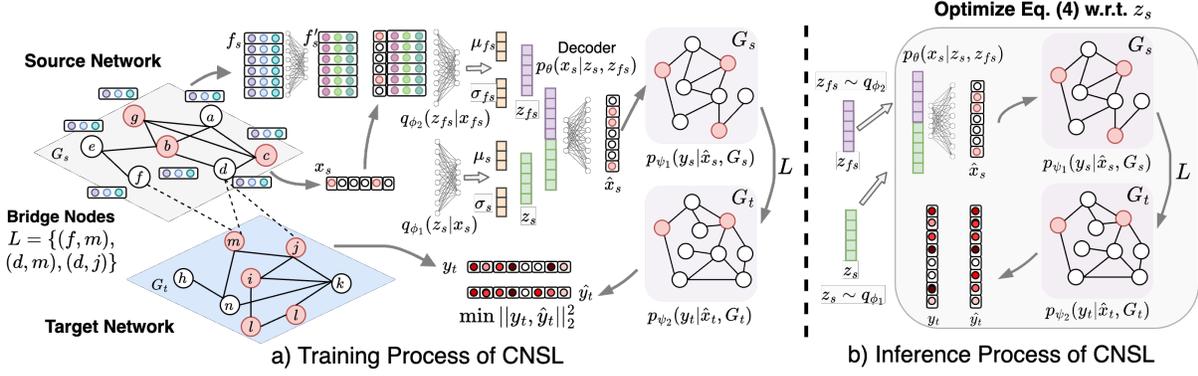
**Overall Objective for Training.** The training procedure of the proposed CNSL model is coupled with Eq. (1), Eq. (2), and the monotonic increasing constraint:

$$\begin{aligned} \mathcal{L}_{\text{train}} = \max_{\theta, \phi_1, \phi_2} \quad & \mathbb{E}_{q_\phi} [p_\theta(x_s, y_t, x_f, \mathcal{G}|z_s, z_{f_s})], \quad (3) \\ \text{s.t.} \quad & KL[q_{\phi_1}(z_s|x_s, \mathcal{G})||p(z_s)] < I_s, \\ & KL[q_{\phi_2}(z_{f_s}|x_s, f_s, \mathcal{G})||p(z_{f_s})] < I_{f_s}, \\ & y_t^{(i)} \geq y_t^{(j)}, \forall x_s^{(i)} \supseteq x_s^{(j)}, \\ = \min_{\theta, \phi_1, \phi_2, \psi_1, \psi_2} \quad & -\mathbb{E}_{q_\phi} [\log p_\theta(x_s|z_s, z_{f_s}) \\ & + \log p_{\psi_1}(y_s|x_s, G_s) + \log p_{\psi_2}(y_t|y_s, x_t, G_t)], \\ \text{s.t.} \quad & KL[q_{\phi_1}(z_s|x_s, \mathcal{G})||p(z_s)] < I_s, \\ & KL[q_{\phi_2}(z_{f_s}|x_s, f_s, \mathcal{G})||p(z_{f_s})] < I_{f_s}, \\ & \|\max(0, y_t^{(j)} - y_t^{(i)})\|_2^2, \end{aligned}$$

where we only need to sample one  $x_s^{(i)}$  and many  $x_s^{(j)}$ ’s (such that  $x_s^{(i)} \supseteq x_s^{(j)}$ ) as training samples for each mini-batch. The  $y_t^{(i)}$  and  $y_t^{(j)}$ ’s are estimated by arbitrary diffusion patterns. For simplicity, we omit the subscript of  $\mathbb{E}_{q_\phi(z_s, z_{f_s}|x_s, y_t, \mathcal{G})}$  as  $\mathbb{E}_{q_\phi}$  when the context is clear. The overall framework is summarized in Figure 2.

### 3.4 Cross-network Seed Set Inference

Upon training completion, the joint probability  $p(z_s, z_{f_s})$  is approximated by the posterior  $q_\phi(z_s, z_{f_s}|x_s, f_s, y_t, \mathcal{G})$ . Both  $p_{\psi_1}(\cdot)$  and  $p_{\psi_2}(\cdot)$  effectively classify the diffusion patterns across networks. This study introduces a sampling method for  $\tilde{x}_s \sim p(x_s)$  by marginalizing over  $p(z_s) \cdot p(z_{f_s})$  to conduct MAP estimation, where  $p(x_s) = \sum_{z_s} \sum_{z_{f_s}} p_\theta(x_s|z_s, z_{f_s}) p(z_s, z_{f_s})$ . However, marginalizing the standard Gaussian prior  $p(z_s, z_{f_s})$  necessitates extensive sampling to align the sample distribution with the target distribution, increasing computational load. Additionally, it is also hard to sample



**Figure 2: The training pipeline of CNSL contains three steps: 1)  $q_{\phi_1}$  and  $q_{\phi_2}$  approximate the distribution of  $p(z_s, z_{f_s})$  in a disentangled manner; 2) the inferred latent variables  $z_s$  and  $z_{f_s}$  are concatenated to reconstruct  $\hat{x}_s$ ; 3) the reconstructed  $\hat{x}_s$  is leveraged as initial seed nodes to initiate the cross-network information propagation and predict expected diffusion  $\hat{y}_t$ .**

individual latent variables from the joint distribution of  $p(z_s, z_{f_s})$ . To cope with both challenges, we consider the density over the inferred latent variables induced by the approximate posterior inference mechanism, and we propose the following objective w.r.t.  $z_s$  to infer  $\hat{x}_s$  in an optimized manner. Specifically, the inference objective function  $\mathcal{L}_{\text{pred}}$  is written as:

$$\begin{aligned} \mathcal{L}_{\text{pred}} &= \max_{z_s} \mathbb{E} [p_{\psi}(y_t | x_s, \mathcal{G}) \cdot p_{\theta}(x_s | z_s, z_{f_s})], \\ &\text{s.t. } z_s \sim q_{\phi_1}(z_s | \hat{x}_s, \mathcal{G}), \quad z_{f_s} \sim q_{\phi_2}(z_{f_s} | \hat{x}_s, f_s, \mathcal{G}), \\ &= \min_{z_s} -\mathbb{E} \left[ \log p_{\psi}(y_t | x_s, \mathcal{G}) + \log \left[ \sum_{z_s} \sum_{\hat{x}_s} p_{\theta}(x_s | z_s, z_{f_s}) \right] \right] \\ &\text{s.t. } z_s \sim q_{\phi_1}(z_s | \hat{x}_s, \mathcal{G}), \quad z_{f_s} \sim q_{\phi_2}(z_{f_s} | \hat{x}_s, f_s, \mathcal{G}), \end{aligned} \quad (4)$$

where we sample many  $\hat{x}_s$  from the training set, and obtain equal amount of  $z_s$  from  $q_{\phi_1}(\cdot)$ . Note that we optimize  $z_s$  (dynamic latent variable) only, instead of both  $z_s$  and  $z_{f_s}$  (static-dynamic entangled latent variable), which is rooted in the specific roles these variables play in the model.  $z_s$  is targeted for optimization because it encodes dynamic information crucial for identifying better seed nodes in the context of information diffusion. This dynamic aspect is mutable and can be optimized to improve source localization accuracy. On the other hand,  $z_{f_s}$  entangles both dynamic and static information, where the static part represents unchangeable node features. Optimizing  $z_{f_s}$  would be less efficient because static features, by their nature, cannot be optimized. The optimization process aims to adjust variables to improve model performance, but since static features remain constant, attempting to optimize  $z_{f_s}$  would not enhance the model's ability to localize diffusion sources.

**Implementation of the Seed Set Inference.** We provide implementation details of the overall inference process here. Specifically, the inference framework first samples  $k$  seed node set  $\hat{x}_s$  from the training set, and we can take the average value  $\bar{z}_s$  and  $\bar{z}_{f_s}$  from the learned latent distributions with taking  $k$  different  $\hat{x}_s$  as input:

$$\bar{z}_s = \frac{1}{k} \sum_i^k q_{\phi_1}(z_s | \hat{x}_s^{(i)}, \mathcal{G}), \quad \bar{z}_{f_s} = \frac{1}{k} \sum_i^k q_{\phi_2}(z_{f_s} | \hat{x}_s^{(i)}, f_s, \mathcal{G}). \quad (5)$$

We concatenate  $\bar{z}_s$  and  $\bar{z}_{f_s}$  as input to minimize the inference loss in Eq. 4. The latent variable  $z_s$  is iteratively optimized according to the inference objective function to minimize  $-\log p_{\psi}(y_t | x_s, \mathcal{G})$ . In practice, Eq. (4) cannot be optimized directly, we thus provide a practical version of the inference objective function: since the

diffused observation  $y_t$  fits the Gaussian distribution and the seed set  $x_s$  fits the Bernoulli distribution, we can simplify Eq. (4) as:

$$\begin{aligned} \mathcal{L}_{\text{pred}} &= \min_{z_s} - \left[ \log \left[ \prod_{i=0}^{N_s} f_{\theta}(z_s^{(i)}, z_{f_s}^{(i)})^{x_s^{(i)}} (1 - f_{\theta}(z_s^{(i)}, z_{f_s}^{(i)})^{1-x_s^{(i)}} \right) \right. \right. \\ &\quad \left. \left. + \|\hat{y}_t - y_t\|_2^2 \right] \right] \quad (6) \end{aligned}$$

where the  $\hat{y}$  is given as the optimal influence spread (i.e.,  $\hat{y}_t = N_t$ ). In other words, the inference objective is guided by the discrepancy between the inferred  $y_t$  and the ground truth  $\hat{y}_t$ . We visualize the overall inference procedure in Figure 2 (b). Specifically, we sample  $\bar{z}_{f_s}$  and  $\bar{z}_s$ , according to Eq. (5), and leverage  $p_{\theta}(\cdot)$  to decode  $\hat{x}_s$ . The predicted  $\hat{x}_s$  is leveraged to initiate the cross-network diffusion and predict  $\hat{y}_t$ . The optimization supervision consists of 1) the mean squared loss between  $\hat{y}_t$  and the ground truth  $y_t$  as well as 2) the probability of node  $v_i$  being seed node  $f_{\theta}(z_s^{(i)}, z_{f_s}^{(i)}) \in [0, 1]$ .

## 4 Experimental Evaluation

This section reports both qualitative and quantitative experiments that are carried out to test the performance of CNSL and its extensions on a simulated dataset that simulates the spread of misinformation across a city-level population and a collected real-world cross-network dataset obtained by crawling two online networking platforms and cross-references between them.

### 4.1 Real-world Dataset: Cross-Platform Communication Network

We collected real-world data from GitHub and Stack Overflow to form the cross-platform communication network, where information flows from Github to Stack Overflow since many posts in Stack Overflow have mentioned or discussed Github Repositories when addressing user's questions. We started by downloading the Stack Overflow public data dump provided by the Internet Archive. Then, we extracted all the Stack Overflow posts where their post texts contain a URL to Github (i.e., 439,753 posts mapping to 439,753 repositories). We further built the Stack Overflow network by finding the question posts, answer posts, and related posts of the current 439,753 posts. This yielded a total of 1,410,600 Stack Overflow posts, encompassing data from 2008 up to 2023.

To obtain the GitHub network, we expanded our initial GitHub network by finding all GitHub repositories that the existing repositories depended upon. We utilized an open-source tool<sup>1</sup>, which uses the GitHub GraphQL API to obtain the dependency information. The resulting GitHub network contains 533,240 repositories. For our experiment, we sampled GitHub repositories from the year 2021 and their dependent repositories from the year before 2021 (i.e., 1204 nodes and 1043 edges). We then found their corresponding Stack Overflow posts (i.e., 3862 nodes and 3149 edges). We obtained the ground truth in a pseudo-setting: we randomly sampled 10% of the GitHub nodes as seed nodes, and simulated their diffusion process within the GitHub network and the Stack Overflow network (i.e., 120 GitHub seed nodes, 354 GitHub infected nodes, 195 Stack Overflow seed nodes, and 482 Stack Overflow infected nodes).

## 4.2 Simulated Dataset: Agent-Based Geo-Social Information Spread

We leverage an agent-based simulation framework based on realistic Patterns of Life [12, 13] to simulate the spread of misinformation across social and physical networks. In this simulation, an agent represents a simulated individual who commutes to their workplace, eats at restaurants, and meets friends and recreational sites. Inspired by the Theory of Planned Behavior [1] and Maslow’s Hierarchy of Needs [18] as theories of human behavior, agents are driven by physiological needs to eat and have a shelter, safety needs such as financial stability requiring them to go to work, and needs for love requiring them to meet friends and build and maintain a social network. Details of the theories of social science informing this simulation are found in [33] and details to use this simulation for data-generation are described in [2].

We augmented this simulation framework to simulate the spread of misinformation using a simple Susceptible-Infectious disease model. The simulation is initialized with 15,000 agents. A small number of  $n$  (by default,  $n = 5$ ) agents are selected randomly as the sources of misinformation and flagged as “Infectious” and all other agents are initially flagged as “Susceptible”. Agents can spread misinformation in two ways: 1) through collocation, allowing an agent to spread the misinformation in-person to other agents located at the same workplace, restaurant, or recreational site, and 2) through the social network, allowing an agent to spread misinformation to their friends regardless of their location. To allow the generation of large datasets for source localization, each spreading of misinformation is stopped after five simulation days. At this time, the following datasets are recorded:

- *Ground Truth*. The set of  $n$  agents that were initially seeded with the misinformation.
- *Misinformation Spread*. The list of agents to whom the misinformation has spread after five days.
- *The Complete Co-location Network*. This network captures the agents who meet each other and thus, may spread misinformation through co-location.
- *The Observed Co-location Network*. This network is a randomly sampled subset of agents from the complete co-location network. It represents the agents in the complete co-location network that are parts of the simulated location tracking. This network is used

to simulate the realistic case of not having access to the location data of every individual.

- *The Complete Social Network*. This network records the friend and family connections of all agents which may infect each other through social contagion.
- *The Observed Social Network*. This network includes a randomly sampled subset of agents from the complete social network and simulates the social media environment. This network simulates the realistic case where an observed social media network may not capture the entire population.
- *Cross-Network Links through Identity*. Links between the two observed networks are defined through identity. Any individual agent in the co-location network is (trivially) connected to itself in the social network.

Once this data is collected, the misinformation spread status of all agents is set to “Susceptible” and  $n$  new agents are selected as the seed nodes of a new case of misinformation. This process of creating new cases of misinformation is iterated every five simulation days to create an unlimited number of realistic datasets of information spread across the physical and social spaces.

For the dataset used for the following experiments, there are 5,281 agents and 8,276 edges in the observed co-location network, and 5,669 agents and 17,948 edges in the observed social network. Each case of misinformation spread yields between 50-200 agents to which the misinformation spreads after five days. This synthetic dataset allows us to capture realistic misinformation spread across both networks. Due to some agents not being captured in the two networks, this dataset allows us to simulate the realistic case where misinformation may spread outside of the observed networks. We provide the code for our agent-based misinformation simulation framework in a GitHub repository found at <https://github.com/Siruiruirui/misinformation>. This repository also contains the generated dataset used in the following experiments.

## 4.3 Experiment Setup

**Implementation Details.** We employ a two-layer MLP for learning node features, which are concatenated with the seed vector in the subsequent stage before being input to the encoder  $q_{\phi_2}(\cdot)$ . Both encoders ( $q_{\phi_1}(\cdot)$ ,  $q_{\phi_2}(\cdot)$ ) and the decoder  $p_{\theta}(\cdot)$  utilize three-layer MLPs with non-linear transformations. We use GNN model architecture coupled with a two-layer MLP network as the aggregation network with 64 hidden units for the two propagation models ( $p_{\psi_1}(\cdot)$  and  $p_{\psi_2}(\cdot)$ ). The learning rates for encoder-decoder,  $p_{\psi_1}(\cdot)$ , and  $p_{\psi_2}(\cdot)$  are set to 0.0001, 0.005, and 0.01 respectively in a multi-optimization manner. Additionally, the number of epochs is 15 for all datasets, with a batch size of 2. The iteration numbers for inference are set to 2 for all datasets.

**Comparison Methods.** We illustrate the performance of CNSL in various experiments against two sets of methods: 1) **Rule-based methods**: *LPSI* [26] predicts the rumor sources based on the convergent node labels without the requirement of knowing the underlying information propagation model; *OJC* [31] aims at locating sources in networks with partial observations, which has strength in detecting network sources under the SIR diffusion pattern. 2) **Learning-based methods**: *GCNSI* [7] learns latent node embedding with GCN to identify multiple rumor sources close to the

<sup>1</sup><https://github.com/edsu/xkcd2347>

Category	Method	LT2LT				LT2IC				LT2SIS			
		PR	RE	F1	AUC	PR	RE	F1	AUC	PR	RE	F1	AUC
Rule-based	LPSI	0.156	0.841	0.263	0.583	0.141	0.849	0.242	0.533	0.079	0.942	0.127	0.497
	OJC	0.104	0.035	0.052	0.500	0.116	0.036	0.054	0.502	0.113	0.036	0.053	0.501
Learning based	GCNSI	0.103	0.858	0.184	0.636	0.103	0.866	0.184	0.622	0.114	0.801	0.199	0.635
	IVGD	0.228	0.948	0.368	0.139	0.227	0.874	0.359	0.138	0.123	0.985	0.215	0.240
	SL-VAE	0.249	0.947	0.395	0.703	0.192	0.847	0.313	0.689	0.242	0.931	0.385	0.612
	DDMSL	0.251	0.923	0.394	0.815	0.309	0.845	0.454	0.732	0.320	0.842	0.464	0.772
Our Method	CNSL	<b>0.332</b>	<b>0.996</b>	<b>0.498</b>	<b>0.888</b>	<b>0.332</b>	<b>0.997</b>	<b>0.498</b>	<b>0.889</b>	<b>0.332</b>	<b>0.997</b>	<b>0.498</b>	<b>0.890</b>
	CNSL-W/O	0.103	0.922	0.185	0.520	0.103	0.930	0.186	0.511	0.103	0.917	0.186	0.517

**Table 1: Performance comparison for cross-platform communication network under LT diffusion pattern for the first network with LT, IC, and SIS diffusion pattern for the second network.**

Category	Method	IC2LT				IC2IC				IC2SIS			
		PR	RE	F1	AUC	PR	RE	F1	AUC	PR	RE	F1	AUC
Rule-based	LPSI	0.124	0.868	0.217	0.489	0.215	0.657	0.324	0.562	0.129	0.906	0.226	0.522
	OJC	0.117	0.032	0.050	0.503	0.097	0.027	0.042	0.499	0.115	0.032	0.050	0.502
Learning based	GCNSI	0.142	0.638	0.233	0.623	0.170	0.476	0.251	0.627	0.152	0.602	0.243	0.630
	IVGD	0.120	0.979	0.210	0.733	<b>0.548</b>	0.391	0.083	0.439	0.115	0.825	0.195	0.733
	SL-VAE	0.254	0.881	0.394	0.719	0.195	0.909	0.321	0.703	0.185	0.829	0.302	0.592
	DDMSL	0.286	0.827	0.425	0.818	0.318	0.886	0.468	0.753	0.270	0.833	0.408	0.689
Our Method	CNSL	<b>0.333</b>	<b>0.990</b>	<b>0.498</b>	<b>0.887</b>	0.333	<b>0.998</b>	<b>0.499</b>	<b>0.891</b>	<b>0.332</b>	<b>0.997</b>	<b>0.498</b>	<b>0.888</b>
	CNSL-W/O	0.103	0.922	0.186	0.514	0.103	0.935	0.185	0.515	0.103	0.928	0.185	0.516

**Table 2: Performance comparison for cross-platform communication network under IC diffusion pattern for first network with LT, IC, and SIS diffusion pattern for the second network.**

Category	Method	G2S-A-D0				G2S-B-D0				G2S-A-D1				G2S-B-D1			
		PR	RE	F1	AUC												
Rule-based	LPSI	0.147	0.982	0.256	0.512	0.165	0.954	0.281	0.609	0.152	0.903	0.260	0.475	0.224	0.973	0.364	0.578
	OJC	0.053	0.018	0.022	0.496	0.125	0.039	0.051	0.507	0.063	0.040	0.043	0.497	0.115	0.058	0.071	0.505
Learning based	GCNSI	0.123	<b>1.000</b>	0.216	0.744	0.117	<b>1.000</b>	0.207	0.351	0.183	<b>1.000</b>	0.300	0.250	0.221	<b>1.000</b>	0.341	0.193
	IVGD	0.139	<b>1.000</b>	0.244	0.502	0.138	<b>1.000</b>	0.242	0.500	0.218	<b>1.000</b>	0.352	0.490	0.266	<b>1.000</b>	0.409	0.500
	SL-VAE	0.364	0.863	0.512	0.707	0.289	0.788	0.423	0.611	0.289	0.754	0.418	0.664	0.425	0.893	0.576	0.725
Our Method	CNSL	<b>0.481</b>	0.816	<b>0.605</b>	<b>0.931</b>	<b>0.452</b>	0.885	<b>0.598</b>	<b>0.933</b>	<b>0.499</b>	0.779	<b>0.609</b>	<b>0.894</b>	<b>0.539</b>	0.987	<b>0.698</b>	<b>0.901</b>
	CNSL-W/O S	0.122	1.000	0.219	0.503	0.117	1.000	0.2101	0.488	0.183	0.998	0.309	0.499	0.221	0.999	0.362	0.501

**Table 3: Performance comparison for Geo-Social Information Spread Data (G2S) for two types (A, B) of simulation. Here D0 considers the initial sources of misinformation as seed nodes and D1 considers the initial sources of misinformation and the infected agents at the first day as seed nodes.**

actual source; *IVGD* [24] propose a graph residual model to make existing graph diffusion models invertible; *SL-VAE* [15] proposed to learn the graph diffusion model with a generative model to characterize the distribution of diffusion sources. *DDMSL* [29] proposed a diffusion model-based source localization method to recover each diffusion step iteratively. Note that existing comparison methods are not designed for cross-network source localization, in order to conduct a fair comparison, we repeated each model separately for two networks and learned the two networks. We used bridge links  $L$  to connect these two models.

**Evaluation Metrics.** Source localization is a classification task so that we use two main metrics to evaluate the performance of our proposed model: 1) *F1-Score (F1)* and 2) *ROC-AUC Curve (AUC)*, as they are classical metrics for classification tasks. since most real-world scenarios tend to have an imbalance between the number of diffusion sources and non-source nodes (fewer diffusion sources),

we additionally leverage  $PR@100$  to evaluate the precision of the top-100 prediction returned by models.

#### 4.4 Quantitative Analysis

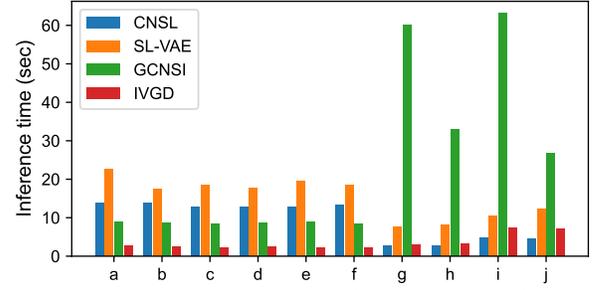
We evaluated the models in different diffusion configurations. For the cross-platform communication data, the underlying diffusions are LT (Table 1) and IC (Table 2) for the first network which was followed by other three diffusion patterns (LT, IC, and SIS) for the second network in each case. For the Geo-Social information spread data (Table 3), the underlying diffusion pattern has been explained in Section 4.2. For that dataset, we used two different simulations (A and B) and also used two different types of seed selections. Here  $D0$  considers the initial sources of misinformation as seed nodes and  $D1$  considers the initial sources of misinformation and the infected agents on the first day as seed nodes.

### Performance in the cross-platform communication network.

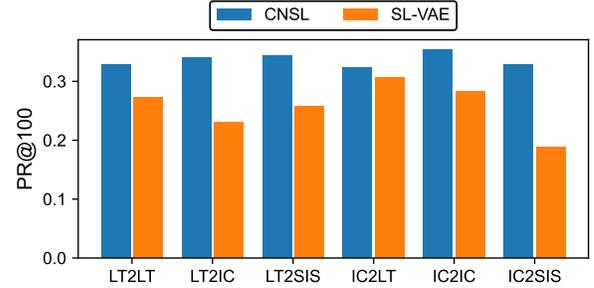
Table 1 shows that CNSL excels others across all metrics and diffusion patterns. In the first network with LT diffusion pattern (LT2LT, LT2IC, LT2SIS), CNSL achieves the highest recall (RE) in all scenarios, with scores of 0.996, 0.997, and 0.997, respectively, indicating its superior ability to identify all relevant instances in the dataset. Additionally, CNSL also exhibits the best precision (PR) in LT2LT and LT2IC scenarios, and competitive precision in the LT2SIS scenario. The F1 scores, which balance precision and recall, are also highest for CNSL, peaking at 0.498 in both LT2LT and LT2IC patterns, demonstrating the method’s overall efficiency and accuracy. The AUC scores for CNSL are robust, ranking highest in LT2LT and LT2SIS scenarios, signifying excellent model performance across various threshold settings. In the Table 2 first network with IC diffusion pattern (IC2LT, IC2IC, IC2SIS), CNSL’s performance remains impressive, maintaining the highest recall scores of 0.990, 0.998, and 0.997, respectively. CNSL also boasts the highest F1 scores in all scenarios, with a notable 0.499 in IC2IC, suggesting a balanced performance between precision and recall. The AUC scores for CNSL are again the highest, with 0.887 in IC2LT and 0.891 in IC2IC, indicating its strong discriminative ability. Overall, CNSL demonstrates considerable strength in reliably identifying relevant instances across different diffusion patterns and networks, while maintaining high precision and excellent area under the ROC curve.

**Performance in geo-social information spread data.** In Table 3, the performance of various methods on Geo-Social Information Spread Data (G2S) is evaluated for two simulation types, A and B, with two different seeding strategies, D0 and D1. Our method, CNSL, exhibits strong performance across all scenarios. In the G2S-A-D0 simulation, CNSL achieves a high precision (PR) of 0.481, showing its effectiveness in correctly identifying misinformation spread. It also has the highest F1 score of 0.605 and an AUC of 0.931, indicating a balanced precision-recall trade-off and excellent model discrimination ability, respectively. For the G2S-B-D0 simulation, CNSL’s precision (0.452) and F1 score (0.598) are notable, and the AUC of 0.933 is the highest compared to other methods, suggesting CNSL’s consistency and reliability. In the G2S-A-D1 scenario, CNSL maintains a high recall (RE) of 0.779 and an impressive AUC of 0.894, which signifies its capacity to identify true misinformation cases effectively when the seeding includes infected agents from the first day. Remarkably, in the G2S-B-D1 scenario, CNSL stands out with the highest precision (0.539) and F1 score (0.698), and it achieves an outstanding AUC of 0.901. This demonstrates CNSL’s superior ability to differentiate between misinformation and non-misinformation spread, especially when the initial condition includes both sources of misinformation and infected agents. The recall of 0.987 in this scenario also indicates that CNSL can identify nearly all instances of misinformation spread. Overall, the CNSL method outperforms other rule-based and learning-based methods in most metrics across different simulations and seeding strategies in geo-social networks.

**Runtime Analysis.** Figure 3 presents a runtime comparison among four learning-based methods: CNSL, SL-VAE, GCNSI, and IVGD across ten different diffusion configurations (a to j). CNSL, which is our method, shows a competitive inference time in all datasets when compared to the SL-VAE. In cross-platform communication network datasets (a) LT2LT, b) LT2IC, c) LT2SIS, d) IC2LT, e) IC2IC, and f)



**Figure 3: Runtime Comparison with learning based methods for dataset a) LT2LT, b) LT2IC, c) LT2SIS, d) IC2LT, e) IC2IC, f) IC2SIS, g) G2S-A-D0, h) G2S-A-D1, i) G2S-B-D0, j) G2S-B-D1**



**Figure 4: Precision@100: the precision rate of the top 100 nodes being predicted as seed nodes. The comparison is conducted between our method: CNSL and the current state-of-the-art: SL-VAE.**

IC2SIS)), CNSL demonstrates an inference time that is neither the fastest nor the slowest, indicating a balanced computational demand for these more complex scenarios. However, in datasets geo-social information spread data (g) G2S-A-D0, h)G2S-A-D1, i)G2S-B-D0, and j)G2S-B-D1), CNSL’s runtime is noticeably lower, suggesting that while CNSL is highly effective in identifying misinformation spread. Overall, CNSL shows a strength in providing a good balance between accuracy and computational efficiency. While there are scenarios where CNSL’s runtime is higher, these may correlate with more complex network conditions where deeper analysis is necessary, which CNSL seems to handle without compromising the quality. This makes CNSL a robust method for practical applications where runtime is a critical factor alongside precision and accuracy.

**Precision analysis at top 100 nodes predicted by models.** Figure 4 illustrates the precision at top 100 (PR@100) comparison between CNSL and the state-of-the-art SL-VAE across various diffusion patterns. PR@100 measures the precision rate of the top 100 nodes predicted as seed nodes, indicating how accurately each method can identify the most influential nodes in the spread of information or misinformation. CNSL shows a strong performance in this metric, outperforming SL-VAE in all diffusion patterns. CNSL exhibits higher PR@100 rates, indicating that it is more precise in identifying the key seed nodes. This precision is crucial in scenarios where it is important to quickly and accurately pinpoint the main drivers of information spread within a network. Notably, CNSL’s precision suggests that its algorithm is particularly adept at handling complex diffusion patterns where the identification of influential nodes is more challenging. The strength of CNSL, as highlighted by Figure 4, lies in its ability to consistently rank

the most relevant nodes higher than SL-VAE. The precision at the top 100 nodes is essential for practical applications where interventions need to be targeted and efficient, such as in the case of misinformation containment or viral marketing.

## 5 Conclusion

In conclusion, information diffusion source localization on cross-networks requires locating the origins of information diffusion within and across networks. We propose a Cross-Network Source Localization (CNSL) framework in this work, which stands as a pivotal advancement in addressing the complexities introduced by cross-network environments, where traditional source localization methods fall short. By ingeniously approximating the distribution of diffusion sources through mean-field variational inference, encoding both static and dynamic features of nodes via a disentangled generative prior, and uniquely modeling the diffusion dynamics of interconnected networks, CNSL offers a comprehensive solution to the problem. Extensive experiments, including quantitative analysis, case studies, and runtime analysis, have been conducted to verify the effectiveness of the framework across different real-world and synthetic cross-networks. The significance of this work lies not only in its methodological innovation but also in its practical implications for safeguarding the integrity and reliability of information in an increasingly interconnected digital world.

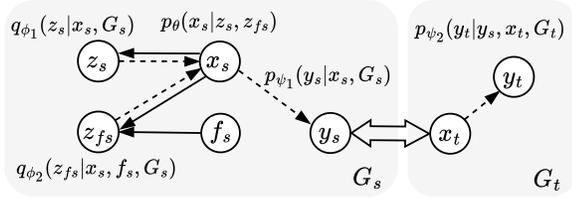
## References

- [1] Icek Ajzen. 1991. The Theory of Planned Behavior. *Organizational Behavior and Human Decision Processes* 50 (12 1991), 179–211. [https://doi.org/10.1016/0749-5978\(91\)90020-T](https://doi.org/10.1016/0749-5978(91)90020-T)
- [2] Hossein Amiri, Shiyang Ruan, Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2023. Massive Trajectory Data Based on Patterns of Life. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*. 1–4.
- [3] Tanmoy Chowdhury, Chen Ling, Junji Jiang, Junxiang Wang, My T Thai, and Liang Zhao. 2023. Deep Graph Representation Learning Influence Maximization with Accelerated Inference. *Available at SSRN 4663083* (2023).
- [4] Chunhui Deng, Huifang Deng, and Zhipeng Fu. 2018. Modeling and Study of Information Transfer in Complex Network. In *2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*. 668–675.
- [5] Swapnil Dhamal, KJ Prabuchandran, and Y Narahari. 2016. Information diffusion in social networks in two phases. *IEEE TNSE* 3, 4 (2016), 197–210.
- [6] Nguyen Do, Tanmoy Chowdhury, Chen Ling, Liang Zhao, and My T Thai. 2024. MIM-Reasoner: Learning with Theoretical Guarantees for Multiplex Influence Maximization. In *International conference on artificial intelligence and statistics*.
- [7] Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. 2019. Multiple rumor source detection with graph convolutional networks. In *CIKM*. 569–578.
- [8] Abdorasoul Ghasemi and Hermann de Meer. 2023. Robustness of interdependent power grid and communication networks to cascading failures. *IEEE Transactions on Network Science and Engineering* (2023).
- [9] Jiaojiao Jiang, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou. 2016. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Commun. Surv. Tutor* 19, 1 (2016), 465–481.
- [10] Yanjie Jiang, Hui Liu, Nan Niu, Lu Zhang, and Yamin Hu. 2021. Extracting Concise Bug-Fixing Patches from Human-Written Patches in Version Control Systems. In *IEEE/ACM 43rd International Conference on Software Engineering (ICSE 2021)*. 686–698.
- [11] Hana Khamfroush, Novella Bartolini, Thomas F La Porta, Ananthram Swami, and Justin Dillman. 2016. On propagation of phenomena in interdependent networks. *IEEE Transactions on network science and engineering* 3, 4 (2016), 225–239.
- [12] Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Ovi Chris Rouly, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2020. Location-Based Social Network Data Generation Based on Patterns of Life. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*. 158–167. <https://doi.org/10.1109/MDM48529.2020.00038>
- [13] Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2019. Simulating urban patterns of life: A geo-social data generation framework. In *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems*. 576–579.
- [14] Chen Ling, Di Cui, Guangmo Tong, and Jianming Zhu. 2021. On Forecasting Dynamics in Online Discussion Forums. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6.
- [15] Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. 2022. Source localization of graph diffusion via variational autoencoders for graph inverse problems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 1010–1020.
- [16] Chen Ling, Junji Jiang, Junxiang Wang, My T Thai, Renhao Xue, James Song, Meikang Qiu, and Liang Zhao. 2023. Deep graph representation learning and optimization for influence maximization. In *International Conference on Machine Learning*. 21350–21361.
- [17] Chen Ling, Guangmo Tong, and Mozi Chen. 2020. Nestpp: Modeling thread dynamics in online discussion forums. In *HT*. 251–260.
- [18] Abraham H Maslow. 1943. A theory of human motivation. *Psychological review* 50, 4 (1943), 370.
- [19] B Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. 2012. Spotting culprits in epidemics: How many and which ones?. In *ICDM*. 11–20.
- [20] Ruizhong Qiu, Dingsu Wang, Lei Ying, H Vincent Poor, Yifang Zhang, and Hanghang Tong. 2023. Reconstructing Graph Diffusion History from a Single Snapshot. *arXiv preprint arXiv:2306.00488* (2023).
- [21] Ali Razavi, Aaron van den Oord, Ben Poole, and Oriol Vinyals. 2019. Preventing Posterior Collapse with delta-VAEs. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=Bje0Gn0cY7>
- [22] Sushila Shelke and Vahida Attar. 2019. Source detection of rumor in social network—a review. *Online Social Networks and Media* 9 (2019), 30–42.
- [23] Diman Zad Tootaghaj, Novella Bartolini, Hana Khamfroush, Ting He, Nilanjan Ray Chaudhuri, and Thomas La Porta. 2018. Mitigation and recovery from cascading failures in interdependent networks under uncertainty. *IEEE Transactions on Control of Network Systems* 6, 2 (2018), 501–514.
- [24] Junxiang Wang, Junji Jiang, and Liang Zhao. 2022. An invertible graph diffusion neural network for source localization. In *Proceedings of the ACM Web Conference 2022*. 1058–1069.
- [25] Zhen Wang, Dongpeng Hou, Chao Gao, Xiaoyu Li, and Xuelong Li. 2023. Lightweight source localization for large-scale social networks. In *Proceedings of the ACM Web Conference 2023*. 286–294.
- [26] Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. 2017. Multiple source detection without knowing the underlying propagation model. In *AAAI*.
- [27] Xovee Xu, Tangjiang Qian, Zhe Xiao, Ni Zhang, Jin Wu, and Fan Zhou. 2024. PGS�: A probabilistic graph diffusion model for source localization. *Expert Systems with Applications* 238 (2024), 122028.
- [28] Qi Xuan, Xincheng Shu, Zhongyuan Ruan, Jinbao Wang, Chenbo Fu, and Guanrong Chen. 2019. A self-learning information diffusion model for smart social networks. *IEEE Transactions on Network Science and Engineering* 7, 3 (2019), 1466–1480.
- [29] Xin Yan, Hui Fang, and Qiang He. 2024. Diffusion model for graph inverse problems: Towards effective source localization on complex networks. *Advances in Neural Information Processing Systems* 36 (2024).
- [30] Wenyu Zang, Peng Zhang, Chuan Zhou, and Li Guo. 2015. Locating multiple sources in social networks under the SIR model: A divide-and-conquer approach. *Journal of Computational Science* 10 (2015), 278–287.
- [31] Kai Zhu, Zhen Chen, and Lei Ying. 2017. Catch'em all: Locating multiple diffusion sources in networks with partial observations. In *AAAI*.
- [32] Kai Zhu and Lei Ying. 2016. Information Source Detection in the SIR Model: A Sample-Path-Based Approach. *IEEE/ACM TON* 24, 1 (2016), 408–421.
- [33] Andreas Züfle, Carola Wenk, Dieter Pfoser, Andrew Crooks, Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, and Hyunjee Jin. 2023. Urban life: a model of people and places. *Computational and Mathematical Organization Theory* 29, 1 (2023), 20–51.

## A CNSL Technical Supplements

### A.1 Derivation of Eq. (2)

$\log p_{\psi}(y_t | x_s, G_s, G_t) = \log [\sum_{x_t} p_{\psi_1}(x_t | x_s, G_s) \cdot p_{\psi_2}(y_t | x_t, G_t)]$ , where  $x_t$  inherited infection probability from  $y_s$ . In practice, we assume  $p_{\psi_1}(x_t | x_s, G_s)$  follows delta distribution, where only the  $x_t$  is 1 that corresponds to the  $x_s$  and the rest of  $x_t$ 's are 0. This property is also assumed in many works [21] using VAE. Therefore,  $\log p_{\psi}(y_t | x_s, G_s, G_t)$  is simplified as Eq. (2).



**Figure 5: The graphical model for CNSL, where the solid arrows indicate the variational approximation  $q_{\phi_1}(z_s|x_s, G_s)$  and  $q_{\phi_2}(z_{f_s}|x_s, f_s, G_s)$  to the intractable posterior  $p(Z|x_s, f_s, G_s)$ . Dashed arrows denote the generative process that decodes  $x_s$  from  $p_{\theta}(x_s|Z)$  and predicts the information diffusion  $p_{\psi}(y_t|x_s, \mathcal{G})$ .**

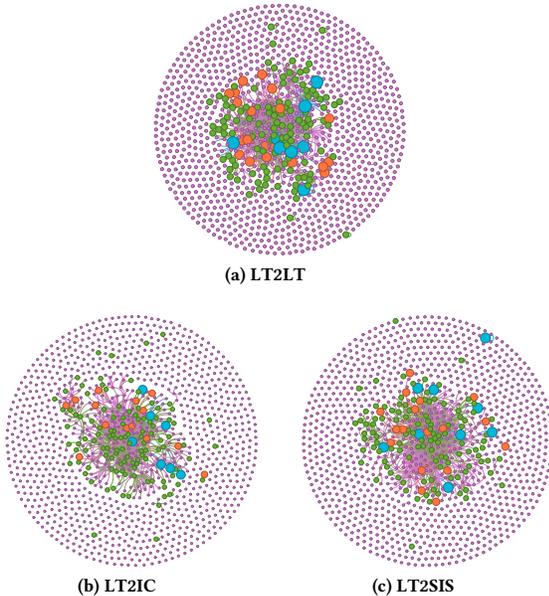
## A.2 Graphical Model of CNSL

We provide the graphical model for the CNSL framework in Figure 5. As shown in the figure, solid arrows indicate the variational approximation  $q_{\phi_1}(z_s|x_s, G_s)$  and  $q_{\phi_2}(z_{f_s}|x_s, f_s, G_s)$  to the intractable posterior  $p(Z|x_s, f_s, G_s)$ . Dashed arrows denote the generative process that decodes  $x_s$  from  $p_{\theta}(x_s|Z)$  and predicts the information diffusion  $p_{\psi}(y_t|x_s, \mathcal{G})$ . The two directional arrow between  $y_s$  and  $x_t$  indicates  $x_t$  inherits the infection probability from the diffusion observation  $y_s$  through bridging nodes  $L$ .

## B Experiment Supplement

### B.1 Case Study

In a case study depicted in Figure 6, we illustrate the distribution of selected seed nodes. Here violet nodes represent the nodes that are not seeds. On the other hand, green nodes are the original seeds that were not selected by CNSL; orange nodes are wrongly identified as seeds by CNSL; and the Blue color nodes are correctly identified as seeds by CNSL.



**Figure 6: Seed inference by CNSL.**

## B.2 Algorithm

### Algorithm 1 CNSL Training Framework

---

**Require:**  $G_s, G_t, f_s, L, x_s, y_t$   
**Ensure:** Trained  $q_{\phi}(\cdot)$ ,  $p_{\theta}(\cdot)$ , and  $p_{\psi}(\cdot)$

- 1: **for** epoch in 1 to  $num\_epochs$  **do**
- 2:     **for** each batch in  $train\_set$  **do**
- 3:          $z_s = q_{\phi_1}(x_s, G_s)$  ▷ Dynamic Encoder
- 4:          $z_{f_s} = q_{\phi_2}(x_s, x_{f_s}, G_s)$  ▷ Static Encoder
- 5:          $\hat{x}_s = p_{\theta}(z_s, z_{f_s})$  ▷ Decoder
- 6:          $\hat{y}_s = p_{\psi_1}(\hat{x}_s, G_s)$  ▷ Source Network Diffusion
- 7:          $\hat{x}_t \leftarrow \hat{y}_s$  ▷  $L = \{(v_s, v_t) | v_s \in V_s, v_t \in V_t\}$
- 8:          $\hat{y}_t = p_{\psi_2}(\hat{x}_t, G_t)$  ▷ Target Network Diffusion
- 9:         Calculate  $\mathcal{L}_{train}$  ▷ Equation (3)
- 10:         Backpropagate loss
- 11:         Update model parameters
- 12:     **end for**
- 13: **end for**

---

For training, we want to use observed  $x_s$  and  $y_t$  to learn the approximate posterior  $q_{\phi}(Z|x_s, \mathcal{G})$ , the decoding function  $p_{\theta}(x_s|Z)$ , and the cross-network diffusion prediction function  $p_{\psi}(y_t|x_s, \mathcal{G})$ . Specifically, we separately obtain two latent variables  $z_s$  and  $z_{f_s}$  in Line 2-3. Both  $z_s$  and  $z_{f_s}$  are fed to reconstruct  $\hat{x}_s$  in Line 5. After the seed set reconstruction, we conduct cross-network diffusion prediction as shown in Line 6-8. The backpropagation is calculated based on Eq. (3) that consists of seed nodes reconstruction error, diffusion estimation error, as well as constraints of KL divergence and influence monotonicity.

### Algorithm 2 CNSL Inference Framework

---

**Require:**  $p_{\theta}(x_s|z_s, z_{f_s})$ ;  $p_{\psi_1}(y_s|x_s, G_s)$ ;  $p_{\psi_2}(y_t|y_s, x_t, G_t)$ ; the number of iteration  $\eta$ ; learning rate  $\alpha$ .

**Ensure:**  $\hat{x}_s$

- 1:  $\bar{z}_s = \frac{1}{k} \sum_i^k q_{\phi_1}(z_s|\hat{x}_s^{(i)}, \mathcal{G})$  ▷  $\hat{x}_s^{(i)}$  sampled from training set.
- 2:  $\bar{z}_{f_s} = \frac{1}{k} \sum_i^k q_{\phi_2}(z_{f_s}|\hat{x}_s^{(i)}, \mathcal{G})$  ▷  $\hat{x}_s^{(i)}$  sampled from training set.
- 3: **for**  $i = 0, \dots, \eta$  **do**
- 4:      $\hat{x}_s = p_{\theta}(\bar{z}_s, \bar{z}_{f_s})$  ▷ Decoder
- 5:      $\hat{y}_s = p_{\psi_1}(\hat{x}_s, G_s)$  ▷ Source Network Diffusion
- 6:      $\hat{x}_t \leftarrow \hat{y}_s$  ▷  $L = \{(v_s, v_t) | v_s \in V_s, v_t \in V_t\}$
- 7:      $\hat{y}_t = p_{\psi_2}(\hat{x}_t, G_t)$  ▷ Target Network Diffusion
- 8:      $\bar{z}_s \leftarrow \bar{z}_s - \alpha \cdot \nabla \mathcal{L}_{pred}(\hat{y}_t, \bar{z}_s, \bar{z}_{f_s})$
- 9: **end for**
- 10:  $\hat{x}_s = p_{\theta}(\bar{z}_s, \bar{z}_{f_s})$

---

For the seed set inference, we first sample  $k$  different  $\hat{x}_s^{(i)}$  from the training set, and we marginalize them to obtain two latent variables  $\bar{z}_s$  and  $\bar{z}_{f_s}$  (Line 1-2). For  $\eta$  iterations, we decode the predicted  $\hat{x}_s$  based on  $(\bar{z}_s, \bar{z}_{f_s})$  (Line 4) and conduct cross-network information diffusion prediction (Line 5-7). The error between predicted  $\hat{y}_t$  and the observed  $y_t$  is leveraged to update  $\bar{z}_s$  based on Eq. (6).