



PGSL: A probabilistic graph diffusion model for source localization

Xovee Xu^{a,1}, Tangjiang Qian^{a,1}, Zhe Xiao^b, Ni Zhang^c, Jin Wu^a, Fan Zhou^{a,*}

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

^b Science and Technology on Communication Networks Laboratory, Shijiazhuang 050050, Hebei, China

^c The 6th Research Institute of China Electronic Corporation, Beijing, 102209, China

ARTICLE INFO

Keywords:

Source localization
Normalizing flow
Diffusion uncertainty
Information diffusion

ABSTRACT

Source localization, as a reverse problem of the graph diffusion, bears paramount significance for a multitude of applications, such as tracking social rumors, detecting computer viruses, and finding epidemic spreaders. However, the innate uncertainty of the diffusion process complicates this task — different source nodes can result in similar or identical diffusions over time, making the source localization task a complex, ill-posed problem. Most existing solutions utilize deterministic techniques and therefore cannot model the diffusion uncertainty of source nodes. Moreover, current probabilistic approaches are inefficient to conduct smooth transformations with variational inference. To overcome these limitations, we propose a probabilistic framework using normalizing flows with invertible transformations and novel objective optimization methods to explicitly model the uncertainty of the diffusion sources. Moreover, graph neural networks are leveraged to encapsulate propagation patterns between the observed diffusion and sources of high uncertainty. Extensive experiments conducted on six distinct networks demonstrate the effectiveness of our model over strong baselines, up to 11.8% and 8.2% improvements in terms of F1 and AUC, respectively, on Twitter dataset under real-world diffusion.

1. Introduction

Graph diffusion prediction is a critical task in social networks and graph mining, aiming to uncover the propagation patterns of information and predict its future state (e.g., size, speed, and scale) (Yang & Counts, 2010). On the contrary, source localization is the inverse problem of the graph diffusion, seeking to identify the source(s) of the observed diffusion process (Ying & Zhu, 2018). Source localization plays a key crucial role in various practical applications, including detecting misinformation and rumor in social networks, controlling epidemic in infectious diseases, and isolating failures in smart grids. It is also aids in understanding the graph diffusion process. Solving the graph source localization problem usually relies on limited diffusion knowledge, such as a subset of the graph topology and node states (Jiang, Wen, Yu, Xiang, & Zhou, 2016).

Given one or multiple source nodes propagating through a network (e.g., a rumor diffusing through a Twitter social network). After a certain period of time, we have an observed diffusion. The source localization is the inverse process of this diffusion: our objective is to identify the original source(s) by given this partial observed diffusion. Existing approaches to source localization primarily involve

network structure analysis-based methods (Prakash, Vreeken, & Faloutsos, 2012; Shah & Zaman, 2011; Zhu & Ying, 2014), information propagation-based methods (Agaskar & Lu, 2013; Pinto, Thiran, & Vetterli, 2012; Seo, Mohapatra, & Abdelzaher, 2012), and learning-based methods (Dong, Zheng, Hung, Su, & Li, 2019; Guo, Zhang, Zhang, & Fu, 2021; Ling, Liang, Wang, & Zhao, 2022; Wang, Jiang, & Zhao, 2022; Wang, Wang, Pei, & Ye, 2017). Among these, network structure analysis-based methods determine the propagation source by analyzing the graph network structure, including graph centrality analysis, neighbor discovery, and algorithms based on graph models. Information propagation-based methods inject sensors into graph networks for source tracking. Learning-based methods learn from a large number of known information propagation paths or simulation of the propagation process to extract the propagation features and patterns, and then establish a predictive model to predict the propagation sources.

Despite achieving promising results in graph source localization, existing methods still face several notable challenges. First, most learning-based methods focus on deterministic learning and are unable to quantify the diffusion uncertainty of the sources. The diffusion uncertainty arises from the inherent variability and unpredictability in identifying

* Corresponding author.

E-mail addresses: xovee@std.uestc.edu.cn (X. Xu), tj.qian@std.uestc.edu.cn (T. Qian), xiaozhe5401@126.com (Z. Xiao), zhangni@ncse.com.cn (N. Zhang), wj@uestc.edu.cn (J. Wu), fan.zhou@uestc.edu.cn (F. Zhou).

¹ Equal Contribution.

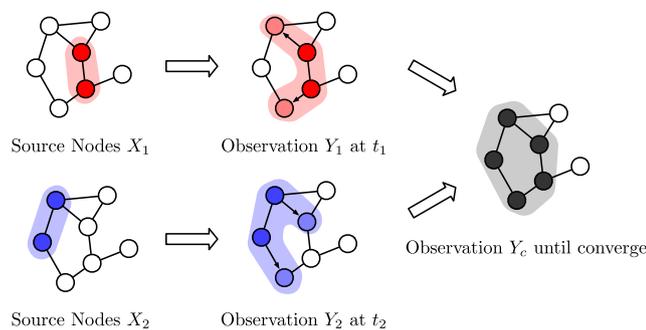


Fig. 1. A simple example of uncertainty problem in graph source localization. Source nodes X_1 and X_2 are totally different from each other, but they finally produce the same diffused results Y_c on the graph. The source localization task from Y_c to X can be highly uncertain.

the source nodes due to different source nodes may resulting in very similar or identical diffusion patterns over time. We illustrate this concept in Fig. 1, where two distinct source nodes X_1 and X_2 , lead to the same diffusion result Y_c on the graph. The source localization task, therefore, becomes highly uncertain and a hard problem for the models to solve. Such uncertainties may stem from the network structures or from the inherent stochasticity of the diffusion process due to random chance or external influences (Wang, Vasilakos, Ma, & Xiong, 2014).

Second, current probabilistic methods are inadequate for conducting smooth transformations between latent space and data distribution with variational inference models (Kingma, Welling, et al., 2019). As the network grows large and complex, transforming the multivariate Gaussian distribution into the data distribution becomes more challenging. Specifically, the uncertainty of the source nodes leads to numerous combinations of the diffusion possibilities and discrete node statuses, requiring us to transform a high-dimensional multivariate Gaussian when building the generative model. Third, network structure analysis-based methods often demand exploring the entire topology space and special node features, resulting in high computational costs for finding sources under variant diffusion patterns (Jiang et al., 2016).

To overcome the limitations mentioned above, we propose a probabilistic graph diffusion model for source localization, named PGSL. First, we adopt a probabilistic framework to handle the diffusion uncertainty challenge. Specifically, we build a probabilistic model that learns graph diffusion patterns given prior knowledge and conditional probability of the source nodes. Then we solve the inverse problem (i.e., source localization) by leveraging variational inference and well-designed optimization methods, allowing us to capture and quantify the inherent uncertainties in the information diffusion process. Second, we use deep generative models – normalizing flows – to conduct smooth transformations between Gaussian and target data to estimate the source node distribution. At last, GNNs are embedded in PGSL to simultaneously handle the learning of complex graph-structured data with uncertainties of the forward process. They learn intrinsic features through neural networks with specifically designed optimization constraints and avoid the expensive searching cost on the entire graph topology. The main contributions are summarized as:

- We design a new probabilistic framework to solve the graph source localization problem, which adopts variational inference and novel objective optimization methods to infer source nodes reversely.
- We propose a generative model that integrates normalizing flows with invertible transformations to explicitly handle the uncertainties of the diffusion processes. It can produce different source nodes depending on the intrinsic characteristics of the diffusion, which in turn helps the training of the forward model.

- Our model is empowered with GNNs to capture information propagation features. GNNs naturally have impressive abilities for modeling graph-structured data, which can rebuild the graph diffusion process under various propagation patterns.
- Extensive experiments on six real-world datasets under different diffusion patterns demonstrate the effectiveness of PGSL over strong baselines in locating the diffusion sources for various kinds of graphs including social, citation, collaboration, and power grid networks.

The rest of the paper is organized as follows. We review the related work in the next Section. Section 3 introduces necessary background knowledge of the source localization problem, propagation models, and normalizing flows. In Section 4, we describe the technical details of our proposed framework PGSL. Experimental settings and results are reported in Section 5. At last, we conclude the paper and discuss future directions in Section 6.

2. Related works

In this section, we review the related literature from three aspects: (1) information diffusion in graphs; (2) source localization (the inverse problem of diffusion); and (3) deep generative models used on inverse problems.

2.1. Information diffusion in graphs

Graph information diffusion estimation (a.k.a. influence spread estimation) is a task of approximating the expected number of influenced nodes given the sources on graphs (Xia, Li, Wu, & Li, 2021). Computing information diffusion in graphs is an NP-hard (Chen, Wang, & Wang, 2010) problem, and thus previous efforts (Goyal, Lu, & Lakshmanan, 2011; Kempe, Kleinberg, & Tardos, 2003; Zhou, Zhang, Guo, Zhu, & Guo, 2013) often adopted Monte Carlo simulations or heuristic methods to estimate the influence spread. Following approaches (Bielski & Trzcinski, 2018; Du et al., 2016; Xie et al., 2020) leveraged deep learning models to predict the diffusion states, but neglected to incorporate graph topologies in their models.

Traditional diffusion models utilize concepts from epidemiology to simulate the propagation of information. Two exemplary models are the Susceptible–Infectious (SI) and Susceptible–Infectious–Recovered (SIR) models, which delineate the process of infection among individuals (Allen, 1994; Anderson & May, 1992). The Bass diffusion models posit that potential adopters are swayed by both personal interactions and exposure to mass media (Bass, 1969). Stochastic process-based models, on the other hand, derive from the behavioral patterns of information sharing in social networks, often making assumptions regarding the intensity of the diffusion process across sequential events. Examples of these models include the Poisson point process (Shen, Wang, Song, & Barabási, 2014) and the self-exciting Hawkes point process (Nickel & Le, 2021; Yu, Xu, Trajcevski, & Zhou, 2022).

Further, Threshold information diffusion models such as Independent Cascade (IC) (Goldenberg, Libai, & Muller, 2001) and Linear Threshold (LT) models (Granovetter, 1978) hypothesize that individual adoption of information is contingent upon the states of their respective neighbors. Machine learning models also come into play in the simulation and prediction of information diffusion. One category of such methods employs hand-crafted features to capture the patterns of diffusion (Cheng, Adamic, Dow, Kleinberg, & Leskovec, 2014; Zhou, Xu, Trajcevski, & Zhang, 2021), whereas another category harnesses the capabilities of end-to-end learning models, such as sequential-based (Li, Ma, Guo, & Mei, 2017; Wang et al., 2017) and embedding-based neural networks (Zhang, Gong, Wu, Huang, & Huang, 2016).

Recently, researchers have utilized graph neural networks (GNNs) to predict the influence of the diffusion (Leung, Cuzzocrea, Mai, Deng, & Jiang, 2019; Li et al., 2017; Qiu et al., 2018; Xu, Zhong, Li, Trajcevski

and Zhou, 2022; Zhou et al., 2021). GNNs can naturally model the graph topology, thereby enhancing the prediction performance of information diffusion. Qiu et al. (2018) and Leung et al. (2019) leveraged the local subgraphs with neighboring node states to determine whether a node is activated. In Jain, Katarya, and Sachdeva (2023), the authors used GNNs to categorize opinion leaders for information diffusion. On the other hand, Li et al. (2017), Xu, Zhou, Zhang and Liu (2022) and Xu, Zhou, Zhang, Liu, and Trajcewski (2021) sought to predict the popularity of social contents by analyzing the diffusion structures and temporal dependencies between information adoptions.

2.2. Source localization

Diffusion source localization is to identify the source(s) of a diffusion process using observations such as the node states and timing of node infections (Ying & Zhu, 2018). Effective source localization has many practical societal and economic impacts, such as detecting rumor sources in social networks (He, Li, Zhou, & Yang, 2021).

Previous studies (Karamchandani & Franceschetti, 2013; Luo, Tay, & Leng, 2013; Nguyen, Nguyen, & Thai, 2012; Shah & Zaman, 2011) primarily employ centrality measures to identify potential propagation sources. These methods are designed to be used on tree-like networks, and with information propagation following the conventional epidemic model. For example, Shah and Zaman (2011) proposed rumor centrality measures and utilized likelihood function to calculate the rumor centrality score of each node according to the topology of the diffused graph. In addition, epidemic models such as SIR and SIS (Luo & Tay, 2012; Luo, Tay, & Leng, 2014; Zhu & Ying, 2014) are widely adopted to address this problem within more sophisticated propagation mechanisms. Zhu and Ying (2014) proposed a notion called Jordan centrality, which is an extension of the traditional diffusion kernel with sparse observations. However, these studies are limited to solving single source localization with certain underlying propagation model (e.g., SI or SIR) and are constrained by the tree-like topology of the diffusion.

Researchers then extended the single source localization problem to multi-source scenarios and to generic graph topologies. Net-Sleuth (Prakash et al., 2012) employed the Minimum Description Length principle to identify the best set of source nodes. LPSI (Wang et al., 2017) is the first model to identify multiple sources without knowing the underlying propagation model, which is based on the idea of source prominence and inspired by a semi-supervised label propagation. In addition, different to leveraging the snapshot of the diffusion states, researchers have proposed to inject sensors into the diffusion network to capture the steps of propagation, which identify the propagation sources progressively (Agaskar & Lu, 2013; Pinto et al., 2012; Seo et al., 2012).

To address the high computational costs and the complexity of acquiring various underlying propagation models in the real world, current studies have turned to graph neural networks (GNNs) to enhance localization performance. GCNSI (Dong et al., 2019) leverages graph convolutional networks (GCNs) to enhance the performance of LPSI (Wang et al., 2017) for detecting multiple rumor sources with a supervised learning method. IGCN (Guo et al., 2021) combines GCNs with the source localization problem via attention mechanism, which significantly reduces the computational complexity. IVGD (Wang et al., 2022) is an invertible graph diffusion model utilizing the graph residual network with Lipschitz regularization to infer sources on generic graph diffusion models. However, these deterministic learning methods cannot quantify the uncertainty of the diffusion sources.

To tackle the inherent uncertainty issue in diffusion process, deep generative models are employed to learn the complex patterns of graph diffusion sources. One such model is the SL-VAE (Ling et al., 2022), which is the first to utilize a variational inference-based framework to infer the optimal diffusion sources based on its diffused observations. However, when dealing with large-scale graphs and complex diffusion patterns, SL-VAE and similar models are often less accurate and expressive.

Table 1
Mathematical notations.

Notation	Description
G	Graph
V	Nodes set
E	Edges set
Y, \hat{Y}	Infected node observations and predictions
X, \hat{X}	Original and predicted source node vectors
\tilde{X}	Source node vector from training data
F_θ	Normalizing flow bijector
θ, ϕ, Θ	Model parameters

2.3. Deep generative models

Deep generative models use probabilistic transformations to represent a complex distribution as simpler ones, allowing them to model high-dimensional prior distribution. Given that locating graph diffusion sources is an inverse problem of graph diffusion with inherent, intractable uncertainty, prior knowledge of the diffusion data is necessary for accurately locate the diffusion sources (Hegde, 2018).

Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a popular class of deep generative models that use various kinds of learning modules to transform data samples from a fixed base distribution. GANs utilize an auxiliary discriminator to provide training signals and estimate the divergence between data samples. Another two classes of deep generative models are variational autoencoder (VAE) (Kingma & Welling, 2014) and autoregressive models (Germain, Gregor, Murray, & Larochelle, 2015). Both models are widely used classes for learning the priors in many image and game-related problems (Ledig et al., 2017; Yu et al., 2018). These generative models do not have reverse or inverse transformations with prohibitive restrictions. GANs and VAEs do not provide rich posterior and are less expressive for modeling complex data distributions. On the other hand, autoregressive models suffer from slow sampling, as the forward calculation of an autoregressive flow is not parallelizable. Consequently, a reversible deep generative model with low computation cost is desired to transform the base distributions to data distributions while maintaining the capabilities of density estimation and efficient sampling process.

Normalizing flows (NFs) (Rezende & Mohamed, 2015) have gained much attention and have been applied in many research areas. The basic idea behind NFs is to use a series of invertible transformations with easily computable Jacobian determinants to learn a function that maps a simple distribution to a complex one. However, NFs do not have tractable inverse transformations and cannot be trained directly on the sampled data. Furthermore, the cost of computing Jacobian is $\mathcal{O}(D^3)$ in general where D is the dimension of random variables. To address this problem, some studies (Huang, Krueger, Lacoste, & Courville, 2018; Rezende & Mohamed, 2015) constructed a tractable Jacobian matrix called autoregressive flows. However, the ability of arbitrary distributions to be transformed into close-to-truth distributions is limited when dealing with high-dimensional data (Kong & Chaudhuri, 2020). Continuous normalizing flows (CNFs) have been proposed to integrate continuous-time dynamics into the generative model and replace the warping function (Chen, Rubanova, Bettencourt, & Duvenaud, 2018). The transformations of CNFs are time-varying and are solved via an ordinary differential equation (ODE). Motivated by FFJORD (Free-form Jacobian of reversible dynamics) (Grathwohl, Chen, Bettencourt, Sutskever, & Duvenaud, 2019), which enables unrestricted learning architectures in a continuous-time invertible model, we propose a more scalable and robust solution to address the uncertainty problem in graph source localization.

3. Preliminaries

In this section, we formally define the source localization problem and provide background information on propagation and generative models. We list mathematical notations used throughout the paper in Table 1.

3.1. Problem definition

Given an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Let $Y = \{y_i\}_i \in \mathbb{R}^{|V|}$ be the infection state vector, where $y_i = 1$ if the node i is infected and $y_i = 0$ otherwise. Let $X = \{x_i\}_i \in \mathbb{R}^{|V|}$ be the diffusion source vector, where $x_i = 1$ if the node i is the source node and $x_i = 0$ otherwise. A single propagation in the graph can have one or multiple diffusion sources. Due to the highly uncertain process of the diffusion, we aim to build a probabilistic model $p(X|Y)$ to explicitly measure the uncertainty from the partial observation Y to the desired source(s) X . Since the structure and scale of the diffused observation Y depend on the graph topology G , i.e., $p(X|Y, G)$, the source localization problem can be formulated as:

$$\hat{X} = \arg \max_X p(X|Y, G), \quad (1)$$

where \hat{X} denotes the predicted source vector.

3.2. Propagation models

The propagation models can be divided into two categories: infection models and influence models (Easley & Kleinberg, 2010). Infection models are used to characterize the spread of infectious diseases between individuals and can be broadly categorized into two types: Susceptible–Infected (SI) model (Allen, 1994) and Susceptible–Infected–Recovered (SIR) model (Anderson & May, 1992). The SI model assumes that each node is either susceptible or infected, and an infected node has a probability p of infecting its neighbors. Once a node becomes infected, it remains in that state. On the other hand, the SIR model introduces a third state “recovered”, where infected nodes can recover with a probability of q . Influence models are used to describe how users in a social network influence each other. Several influence models have been widely-adopted, such as Independent Cascade (IC) model (Goldenberg et al., 2001) and Linear Threshold (LT) model (Kempe et al., 2003). In both IC and LT models, the active nodes have only one opportunity to influence their susceptible neighbors. For LT model, each node has a threshold t , which represents the probability that the node will be influenced.

3.3. Generative models: Continuous normalizing flows & FFJORD flows

To parameterize a normalized distribution from a complex normalized distribution, a set of invertible functions f_i ($i = 0, \dots, k$) are used to warp the normalized base distribution. By applying these transformations k times, the density $p(x)$ can be obtained from z_0 as follows:

$$x = f_k \circ \dots \circ f_1(z_0), \quad (2)$$

$$\log p(x) = \log p(z) - \sum_{i=1}^k \log \det \left| \frac{\partial f_i}{\partial z_{i-1}} \right|, \quad (3)$$

where z_i is the intermediate variable.

Computing the determinant of the Jacobian matrix can be time-consuming, as it has a complexity of $\mathcal{O}(D^3)$. To overcome this inefficiency, some studies (Huang et al., 2018; Rezende & Mohamed, 2015) have proposed the use of autoregressive flows to construct a tractable Jacobian matrix. However, the capabilities of autoregressive flows to transform arbitrary distributions to close-to-truth ones are still limited when dealing high-dimensional data (Kong & Chaudhuri, 2020). Recently, a new type of generative models called continuous normalizing flows (CNF) has been proposed, which replaces the warping function with an integral of continuous-time dynamics (Chen et al., 2018). It applies a transformation similar to Eq. (2) in a time-varying way and can be solved by an ordinary differential equation (ODE) solver. Specifically, given a base distribution $z_0 \sim p(z_0)$ and an ODE solver defined by the function $f(z(t), t; \theta)$, we can obtain $z(t_1)$ to constitute x

by solving the initial value problem $z(t_0) = z_0, \partial z(t)/\partial t = f(z(t), t; \theta)$. The change in the log density can be computed as:

$$\frac{\partial \log p(z(t))}{\partial t} = -\text{Tr} \left(\frac{\partial f}{\partial z(t)} \right), \quad (4)$$

$$\log p(z(t_1)) = \log p(z(t_0)) - \int_{t_0}^{t_1} \text{Tr} \left(\frac{\partial f}{\partial z(t)} \right) dt. \quad (5)$$

Similar to CNF, FFJORD Flow is a continuous-time reversible generative model (Grathwohl et al., 2019). It leverages an efficient log-likelihood estimator and effectively reduces the high computational costs of CNF from $\mathcal{O}(D^2)$ to $\mathcal{O}(D)$. The computational complexity of the model comes from computing the trace of $\partial f / \partial z(t)$. Specifically, this network uses two tricks to reduce the computational costs, one trick is to use reverse-mode automatic differentiation to make compute vector-Jacobian products with approximately the same cost as evaluating f , and the other trick is to take a double product of the matrix with a noise vector to obtain the unbiased estimate of the trace of the matrix. These lead to FFJORD becoming the first scalable and reversible generative model with an unconstrained Jacobian.

4. Methodology

In this section, we describe the technical details of our proposed model PGSL.

4.1. Overall framework

To quantify uncertainty in graph diffusion source localization, it is necessary to employ a probabilistic model to characterize the conditional probability $p(X|Y)$. For an efficient and accurate modeling of the prior distribution $p(X)$, we leverage normalizing flows as the chosen deep generative models. The invertible model is subsequently trained via variational inference to reconstruct the source node vector X .

Diffusion processes in real-world scenarios are influenced by myriad factors, such as propagation protocol, individual immunity, external influence, and diffusion rate. These contribute to significant variations in the underlying processes. To address this challenge, we propose a generalized approach that incorporates graph neural networks (GNNs) to enable our model to adapt to a wide range of diffusion patterns, thereby transcending the constraints of a limited set of predefined patterns. GNNs provide a robust tool for applying deep learning techniques to non-Euclidean data, and they are particularly well-suited for capturing and expressing complex topological and diffusion patterns inherent in graph source localization, especially when the graph becomes large and complex.

The overall framework of PGSL is depicted in Fig. 2. PGSL consists of three main stages: (1) Modeling the uncertainty of the diffusion sources (Section 4.2); (2) Learning the information propagation patterns (Section 4.3); and (3) Inferring the source nodes (Section 4.4).

Throughout the training phase, we employ a hybrid approach of NFs and GNNs to train an inverse model. We apply the NF bijector to the original source node vector X in the forward direction, yielding the distribution of the latent space. We then sample a latent vector Z from the obtained distribution, and pass it through the NF bijector in the reverse direction to generate \hat{x} . This is reconstructed from the NF bijector and accommodates the uncertainty of the diffusion source(s). Subsequently, we feed \hat{X} into the GNN-based forward model to simulate the real graph diffusion process.

During the inference phase, our objective is to infer the optimal diffusion source \hat{X} given the observed graph diffusion Y . We randomly sample a source node vector X_0 from the binomial distribution and optimize it through the trained inverse model and the training data \hat{X} . The aim is to minimize the difference between the diffused prediction \hat{Y} and the observed diffusion Y . At last, the source node vector X_0 is transformed into the predicted target vector \hat{X} within a few epochs.

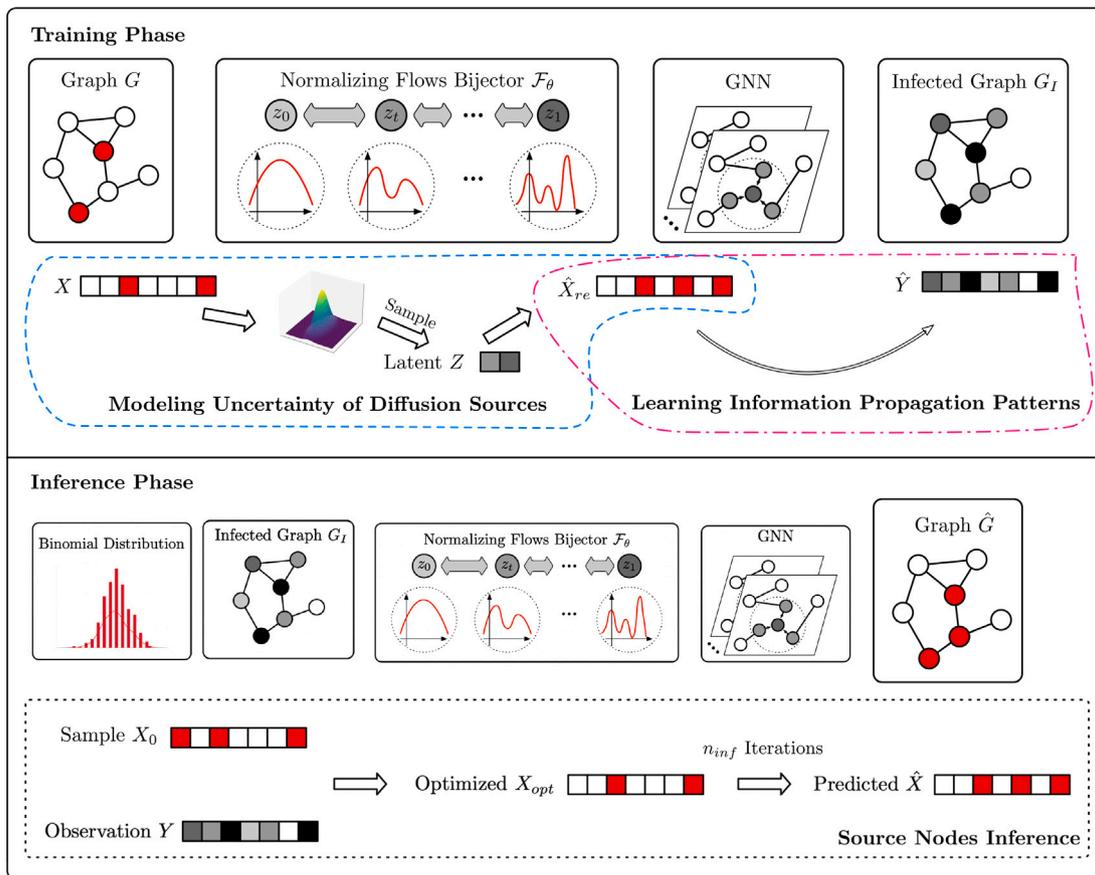


Fig. 2. The proposed model PGSL consists of three stages. The uncertainty modeling stage and propagation patterns learning stage construct the probabilistic graph diffusion model, and the source nodes inference stage infers sources from diffused observations. The solid boxes represent data source and model components, while the dashed box represents model data flow.

4.2. Modeling uncertainty of diffusion sources

The localization problem of the diffusion sources can be framed as a Maximum A Posterior (MAP) estimation problem. To this end, we decompose the conditional probability $p(X|Y, G)$ into two components, given by:

$$\arg \max_X p(X|Y, G) = \arg \max_X p(Y|X, G) \cdot p(X). \quad (6)$$

Estimating the distribution of the diffusion source nodes $p(X)$ is difficult according to Eq. (6). To circumvent this problem, we propose to reduce the dimensionality of X to a latent variable Z in a low-dimensional space, which can be done with low computational cost. However, using plain auto-encoders to directly map the diffusion source node vector X in a deterministic learning way does not fulfill the requirement of measuring the uncertainty in source localization. In most of the graph diffusion cases, the latent variable Z is independent of the diffused observation Y (Ling et al., 2022), thus we can separate the modeling of the graph diffusion into two stages. In the first stage, we train a model to measure the uncertainty of X ; At the second stage, we model the graph diffusion process refer to $p(Y|X, G)$.

4.2.1. The deep generative model

The variational inference offers a reliable mathematical framework for addressing the uncertainty problem involving randomness, which has inspired us to learn the uncertainty of source localization using a probabilistic approach. To achieve this, We first leverage a deep generative model (Kingma et al., 2019) to map the high-dimensional $p(X)$ into low-dimensional $p(Z)$, where $Z \in \mathbb{R}^d$ is the latent random variable vector and $d \ll |V|$. Moreover, due to the rapid growth of

graph size in many real-world networks (e.g., social networks often have tens of thousands of nodes, if not millions), the uncertainty problem becomes complex and intricate under such circumstances, and more exact inference and sampling techniques are needed for tackling this problem. Thus, we capture the intrinsic uncertainty by estimating the density of source nodes, aiming to learn a data-driven statistical model to capture its dynamics. We exploit an invertible neural network f_θ following FFJORD flows (Grathwohl et al., 2019) in the generative model to parameterize the distribution $p(X)$, which conducts a series of smooth and invertible transformations between the latent variable Z and target X . The network f_θ considers a continuous transformation from the latent state $z(t_0)$ to $z(t_1)$ as follows:

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f_\theta(z(t), t) dt, \quad (7)$$

$$\log p(z(t_1)) = \log p(z(t_0)) - \int_{t_0}^{t_1} \text{Tr} \left(\frac{\partial f_\theta}{\partial z(t)} \right) dt, \quad (8)$$

this integration can be solved by ODE solvers (Chen et al., 2018).

4.2.2. Objective function

During the generating process, we have the following conditional probability:

$$p(\hat{X}_{re}|X, G) = p(\hat{X}_{re}|Z)p(Z|X, G), \quad (9)$$

where the posterior $p(Z|X, G)$ can be used to infer the latent variable Z , but $p(X)$ is intractable. We instead approximate the posterior $q_\theta(Z|X, G)$ parameterized by θ , then the likelihood $p(X, G|Z)$ can be obtained through calculating the Kullback–Leibler (KL) divergence between $p(Z|X, G)$ and $q_\theta(Z|X, G)$. The approximated posterior

Algorithm 1 Modeling Uncertainty of Source Localization

Input: Source node vector $X \in \mathbb{R}^{|V|}$, invertible transformation f , start time t_0 , stop time t_1 , Hutchinson's estimator \mathbb{H} , predetermined ODE solver Int_{ode} .

function $f_{aug}([z(t), X], t)$
 $\text{Tr} \leftarrow -\mathbb{H}(f^{-1}, z(t));$
return $[f, \text{Tr}];$
end function

while not convergent **do**
 $H_0 \leftarrow \text{FC}(X);$ ▷ Reduce input dimension with FC layers
 $[z(t_0), \text{Tr}] \leftarrow \text{Int}_{ode}(f_{aug}, [z(t_1), 0], t_0, t_1);$
 $\log p(z(t_1)) \leftarrow \log p(z(t_0)) - \text{Tr};$
 $H_1 \leftarrow f^{-1}(Z);$
 $\hat{X}_{re} \leftarrow \text{FC}(H_1);$
Minimize \mathcal{L}_{USL} via gradient descent optimizer;
Update parameters $\theta;$
end while

$q_\theta(Z|X, G)$ can be obtained by minimizing the KL divergence:

$$q_\theta(Z|X, G) = \min_{\theta} \mathbb{D}_{\text{KL}} [q_\theta(Z|X, G) \| p(X, G, Z)]. \quad (10)$$

Because of the intractable joint distribution $p(X, Y, G)$, we approximate the posterior $q_\theta(Z|X, G)$ by maximizing the Evidence Lower Bound (ELBO), which reduces the computational cost rather than directly calculating the KL-divergency:

$$\text{ELBO} = \mathbb{E}_{q_\theta} [\log p(X, G, Z) - \log q_\theta(Z|X, G)], \quad (11)$$

where \mathbb{E}_{q_θ} denotes $\mathbb{E}_{q_\theta(Z|X, G)}$ for simplicity.

Then we can minimize the KL divergence between the $q_\theta(Z|X, G)$ and $p(Z|X, G)$ by optimizing the negative ELBO via Jensen's inequality, the loss function \mathcal{L}_{USL} can be described as:

$$\mathcal{L}_{USL} = \min_{\theta} [-\mathbb{E}_{q_\theta} [\log p_\theta(X, G|Z)]] \quad (12)$$

$$+ \mathbb{D}_{\text{KL}} [q_\theta(Z|X) \| p(Z)] \quad (13)$$

$$- \mathbb{E}_{q_\theta} \left[\int_{t_0}^{t_1} \text{Tr} \left(\frac{\partial f_\theta}{\partial z(t)} \right) dt \right]. \quad (14)$$

In practice, we build a bijector \mathcal{F}_θ with FFJORD flows and fully-connected (FC) layers to reduce the input dimension. The bijector has two directions, and \mathcal{F}_θ^{-1} denotes the inverse direction when we map Z back to X . To overcome the issue of imbalance, we use the variant cross-entropy loss for the calculation of the reconstruction loss $\log p_\theta(X, G|Z)$. The training procedure of uncertainty modeling for graph source localization is depicted in Algorithm 1.

4.3. Learning information propagation patterns

In the second stage, we simulate the graph diffusion process using a GNN-based model. Specifically, according to Eq. (1), the posterior $p(Y|X, G)$ denotes the forward process of the problem which should be maximized together with the uncertainty modeling process. In order to maintain the same form as other items in Eq. (12), we calculate the negative log of $p_\phi(Y|X, G)$ with GNN-based method parameterized by ϕ . The loss function is defined as:

$$\mathcal{L}_{IP} = \min_{\phi} [-\log p_\phi(Y|X, G)]. \quad (15)$$

In addition to optimizing the graph diffusion process, a monotone increasing constraint on the diffusion is required, which means that if one source set X_i is a subset of another source set X_j , the probability of each node being infected $p_{\text{inf}}(X_i)$ from the source set X_i should be smaller than source set X_j . This constraint is described as follows:

$$\forall X_i \subseteq X_j, \quad \text{s.t. } p_{\text{inf}}(X_i) \leq p_{\text{inf}}(X_j). \quad (16)$$

Algorithm 2 Learning Information Propagation Patterns

Input: Reconstructed source node vector $\hat{X}_{re} \in \mathbb{R}^{|V|}$; Graph adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$; GNN-based method; Iterations for constructing input feature n_{feat} .

while not convergent **do**
for $i = 0 \dots n_{feat}$ **do**
 $X_i \leftarrow A \cdot \hat{X}_{re}$
end for
 $\hat{Y} \leftarrow \text{GNN}_\phi(X_n)$
Minimize \mathcal{L}_{IP} via gradient descent optimizer;
Update parameters $\phi;$
end while

However, it is difficult to model the inequality constraints directly. We convert the constraint Eq. (16) into a Lagrangian form as follows:

$$\mathcal{C}_{\text{mono}} = \lambda \|\max\{0, p_{\text{inf}}(X_i) - p_{\text{inf}}(X_j)\}\|_2^2, \forall X_i \subseteq X_j, \quad (17)$$

where λ is the regularization hyperparameter. Then the loss function is defined as:

$$\mathcal{L}_{IP} = \min_{\phi} [-\log p_\phi(Y|X, G)]. \quad (18)$$

In practice, we calculate the forward loss $\log p_\phi(Y|X, G)$ with Mean Square Entropy loss. The training procedure of learning information propagation patterns is depicted in Algorithm 2.

At last, combining Eqs. (12) and (15) together, we minimize the negative ELBO with the monotone constraint in Eq. (17):

$$\mathcal{L}_{\text{train}} = \min_{\theta, \phi} \{ -\mathbb{E}_{q_\theta} [\log p_\phi(Y|X, G) + \log p_\theta(X|Z)] \quad (19)$$

$$+ \mathbb{D}_{\text{KL}} [q_\theta(Z|X) \| p(Z)] + \mathcal{C}_{\text{mono}} \quad (20)$$

$$- \mathbb{E}_{q_\theta} [\log p(z(t_0)) - \log p(z(t_1))] \}. \quad (21)$$

4.4. Source nodes inference

Since the distribution $p(X)$ is modeled by $p(Z)$ after training, we can solve the MAP in Eq. (1) via $p(X) = p(X|Z)p(Z)$. However, due to the computational complexity of sampling Z from $p(Z)$, we propose to sample Z from the posterior $q(Z|X)$. The objective function of the inference phase is defined as:

$$\mathcal{L}_{\text{infer}} = \min_X [-\log p_\phi(Y|X, G) - \log [p_\theta(X|Z)q_\theta(Z|\tilde{X})]], \quad (22)$$

where \tilde{X} is the source node vector from the training data.

We sample an initial diffusion source X_0 from a binomial distribution in which the probability p_B is set to 0.5 (Ling et al., 2022):

$$X_0 \sim \mathcal{B}(|V|, p_B). \quad (23)$$

However, randomly initializing the input X_0 will increase the search space during the optimization. We leverage the prior knowledge of the observed diffusion source \tilde{X} by using the mean of latent variables obtained from diffusion sources in the training set.

$$\tilde{Z} = \frac{1}{N} q_\theta(Z|\tilde{X}). \quad (24)$$

The objective function is then defined as:

$$X_0 = \min_x [-\log p_\phi(Y|X, G) - \log p_\theta(X|\tilde{Z})]. \quad (25)$$

For the purpose of minimizing the objective function, we should maximize both $p_\phi(Y|X, G)$ and $p_\theta(X|\tilde{Z})$. For maximizing $p_\phi(Y|X, G)$, we choose to measure mean squared error $\|Y - \hat{Y}\|_2^2$ between the ground truth Y and the prediction $\hat{Y} = p_\phi(Y|X, G)$. For maximizing $p_\theta(X|\tilde{Z})$, we leverage the maximum likelihood estimation (MLE) method. Since $X \in \{0, 1\}^{|V|}$, we can consider the mapping of Z to X as a multilabel

Table 2
Dataset statistics.

	Jazz	Network Science	Cora-ML	Power Grid	Weibo-60k	Twitter-20k
# Nodes	198	1589	2810	4941	63,433	26,382
# Edges	2742	13,532	7981	6549	70,102	33,242
Avg degree	27.7	17.29	5.68	2.67	2.21	2.52
Graph density	0.1406	0.0022	0.0020	0.0005	0.00003	0.0010
Assortativity Coef.	0.0202	0.4616	-0.0766	0.0035	-0.6256	-0.5641
Transitivity	0.5203	0.6934	0.1143	0.1032	0.00061	0.00006

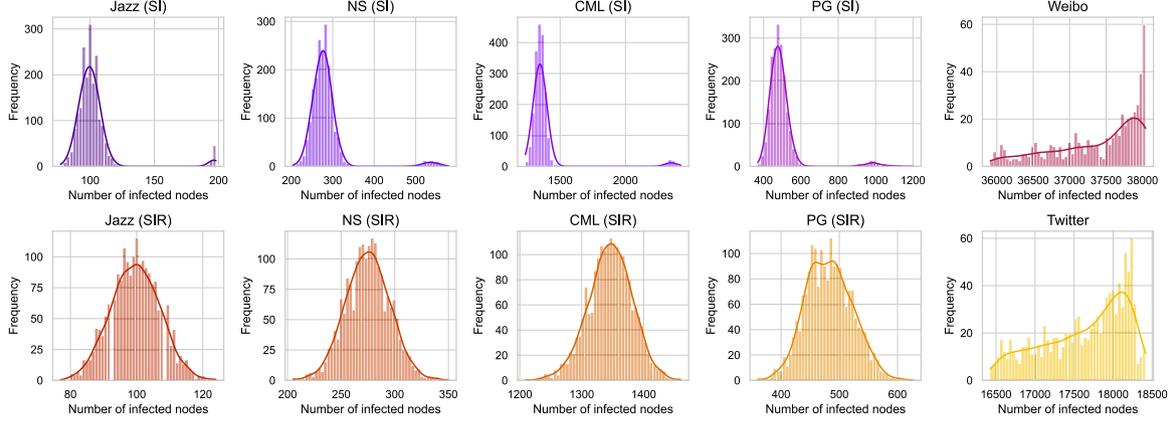


Fig. 3. Distributions of the infected nodes for six datasets.

Algorithm 3 Overall Framework of PGSL

Input: Graph adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$; The diffused observations $Y \in \mathbb{R}^{|V|}$; NF bijector \mathcal{F}_θ parameterized by θ ; GNN-based forward model parameterized by ϕ ; Iterations for inference phase n_{inf} .
Output: The predicted diffusion source $\hat{X} \in \mathbb{R}^{|V|}$.

```

while not convergent do
     $Z \leftarrow \mathcal{F}_\theta(X)$ ; ▷ Obtain latent variables
    Sample  $Z_i$  from standard normal distribution  $\mathcal{N}(0, 1)$ ;
     $\hat{X}_{\text{re}} \leftarrow \mathcal{F}_\theta^{-1}(Z_i)$ ; ▷ Obtain reconstructed sources
     $\hat{Y} \leftarrow \text{GNN}_\phi(\hat{X}_{\text{re}}, A)$ ; ▷ Obtain diffused predictions
    Minimize  $\mathcal{L}_{\text{train}}$  via gradient descent optimizer;
    Update parameters  $\theta$  and  $\phi$ ;
end while
for  $i = 0 \dots n_{\text{inf}}$  do
    Sample  $X_0$  from binomial distribution  $B(|V|, 0.5)$ ;
    Optimize  $X_0$  through Eq. (25) via gradient descent;
    Minimize  $\mathcal{L}_{\text{infer}}$ ; ▷ Obtain optimized sources  $X_{\text{opt}}$ 
end for
 $\hat{X} \leftarrow X_{\text{opt}}$ ;

```

classification problem, where the label is $\{0, 1\}$ with $|V|$ categories. Then the loss function can be converted to:

$$\mathcal{L}_{\text{MLE}} = \sum_{i=1}^N \prod_{j=1}^{|V|} [\mathcal{F}_\theta^{-1}(z_{i,j})]^{x_i} [1 - \mathcal{F}_\theta^{-1}(z_{i,j})]^{(1-x_i)}, \quad (26)$$

where \mathcal{F}_θ^{-1} is the inverse of FJORD bijector. Thus we can derivate Eq. (22) into:

$$\mathcal{L}_{\text{infer}} = \min_X \{ \|Y - \hat{Y}\|_2^2 - \log \mathcal{L}_{\text{MLE}} \}. \quad (27)$$

To obtain valid source values, we discretize the continuous values using a threshold τ and update the initial diffusion source vector X_0 . The updated vector is fed into the forward model to obtain the diffused prediction Y . In this process, we optimize the value of X_0 using Eq. (22) and the diffused observation Y .

The overall framework of PGSL, which combines uncertainty modeling of source localization, information propagation pattern learning, and source node inference, is illustrated in Algorithm 3.

5. Experiment

We present the results of our experiments on graph source localization, starting with a description of our experimental setup, including data, baselines, metrics, and implementation details. We then compare our proposed model with baselines on both synthetic and real-world scenarios. In addition, we provide ablation studies and model analyses to better understand the performance of our approach.

5.1. Experimental settings

5.1.1. Datasets

Four synthetic datasets and two real-world diffusion datasets are selected to evaluate the performance of our model as well as baselines. The basic statistics and distributions of datasets are presented in Table 2, Figs. 3–7. For four synthetic datasets (*Jazz*, *Network Science*, *Cora-ML*, and *Power Grid*), where we had only graph topology information, we randomly chose 10% of nodes as sources and simulate the graph diffusion based on the SI or SIR propagations with sufficient iterations. For the remaining two real-world datasets (*Weibo-60k* and *Twitter-20k*), we sample 10% and 4% nodes, respectively. Their associated diffusion paths are also included. The source and infected nodes were determined based on observation time. We do not use any special node features for the purpose of generalization. Using certain node features – such as number of followers, user attributes, or post texts – may improve model’s performance, but their effectivenesses are dependent on certain types of networks and in this work we mainly focus on network structures and diffusion uncertainties. The overall ratio of the training set and test set is 9:1. Specifically, we select diffusion cascade with a similar number of source nodes from all valid diffusions from original *Weibo* and *Twitter* datasets to build a sub-graph. The number of source nodes was determined using half an hour and one day as observation times for the *Weibo* and *Twitter* datasets, respectively. The nodes that had propagated during the observation time were considered as source nodes.

- *Jazz* (Glaiser & Danon, 2003): A collaboration network of Jazz bands.

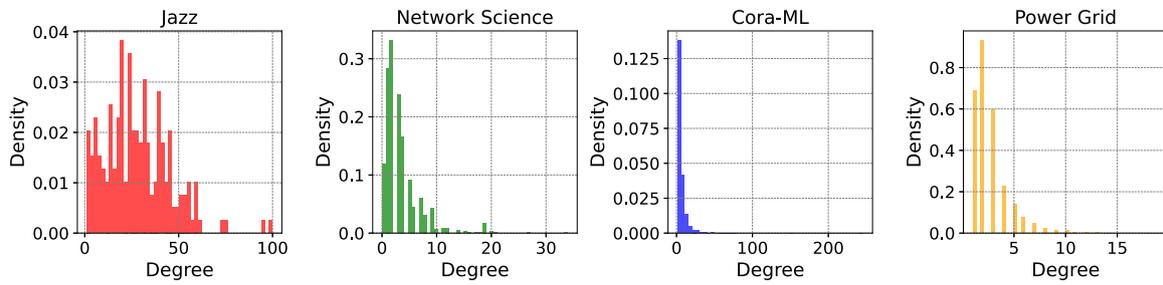


Fig. 4. Node degree distribution.

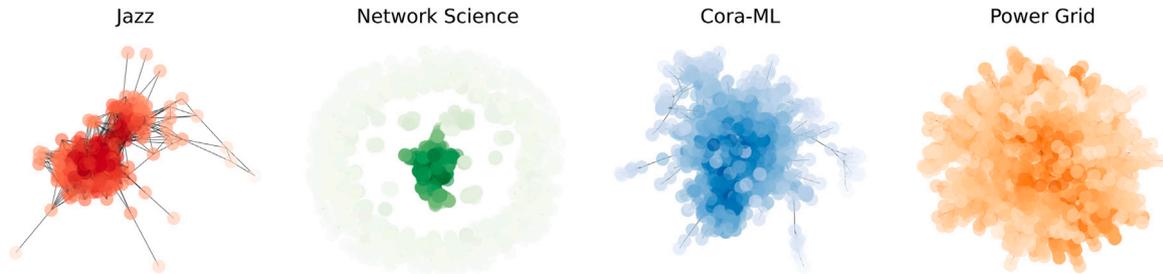


Fig. 5. Node closeness centrality. Darker node color indicates higher closeness centrality.

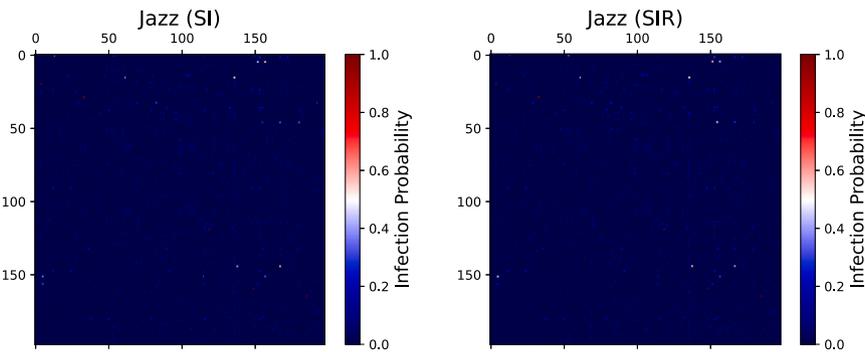


Fig. 6. The infection probability matrix of the nodes in Jazz dataset. Lighter nodes indicate their infection probabilities are around 0.5, with greater uncertainties during information diffusion.

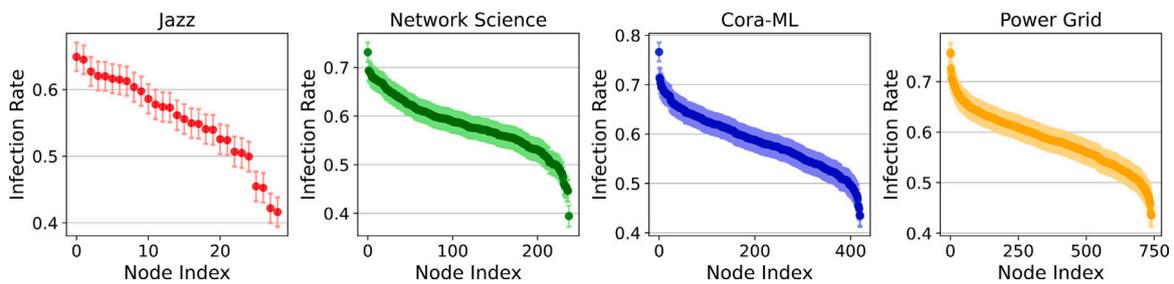


Fig. 7. The distribution of node infection rate. The errorbars indicate confidence intervals for the likelihood of infection. We only show nodes with non-zero infection rate.

- *Network Science (NS)* (Newman, 2006): A coauthor network of scientists working on network theory.
- *Cora-ML (CML)* (McCallum, Nigam, Rennie, & Seymore, 2000): A citation network contains computer science research papers.
- *Power Grid (PG)* (Watts & Strogatz, 1998): A topology network of the Western States Power Grid in the US.
- *Weibo* (Xu et al., 2021): A social network from Weibo, which is formed by user following relationships.
- *Twitter* (Weng, Menczer, & Ahn, 2013): A social network from Twitter containing public English-written tweets published between Mar 24 and Apr 25, 2012.

5.1.2. Baselines

We use five strong source localization models as the baselines, including LPSI (Wang et al., 2017), GCNSI (Dong et al., 2019), OJC (Zhu, Chen, & Ying, 2017), NetSleuth (Prakash et al., 2012), and SL-VAE (Ling et al., 2022), and each is outlined as follows:

- *LPSI* (Wang et al., 2017): Label propagation based source identification model, which predicts the rumor sources without knowing the underlying information propagation model.
- *GCNSI* (Dong et al., 2019): GCN-based source identification model, which is a deep-learning based method with GCN layers

to learn latent node embeddings for identifying multiple rumor sources.

- *NetSleuth* (Prakash et al., 2012): NetSleuth identifies the best set of sources via the minimum description length principle. It is designed for the SI diffusion.
- *OJC* (Zhu et al., 2017): OJC locates sources with partial observations by using a candidate selection algorithm.
- *SL-VAE* (Ling et al., 2022): The first probabilistic approach use VAEs to tackle the ill-posed source localization problem.

5.1.3. Metrics

We use Precision, Recall, F1-Score, and AUC to evaluate the performance of PGSL. We first define the following terms. True Positive (TP): number of source nodes that correctly predicted. True Negative (FP): number of non-source nodes that correctly predicted. False Positive (FP): number of non-source nodes that incorrectly predicted as source node. False Negative (FN): number of source nodes that incorrectly predicted as non-source node. Then Precision, Recall, and F1-Score can be defined as:

- $Precision = TP / (TP + FP)$
- $Recall = TP / (TP + FN)$
- $F1 = 2TP / (2TP + FP + FN)$

The Area Under the Receiver Operating Characteristics Curve (AUC) is another important metric to evaluate a classifier and is less sensitive to data imbalance.

5.1.4. Implementation details

We utilized a 2-layer fully-connected (FC) layer and 3-layer neural ODE functions to implement the NF bijector with “dopri5” ODE solver. As for the GNN model, we chose the base models from several state-of-the-art information propagation methods (Ko, Lee, Shin, & Park, 2020; Xia et al., 2021): Graph Attention Network (GAT), GCN, and simple GNN. For all three GNN models, we set the hidden unit size to 128. The GAT model has two attention heads. We use dropout in FC layers to avoid overfitting issue and the dropout rate is set to 0.5. The overall training learning rate was set to $1e^{-3}$, and the number of epochs is 200 for all datasets. All the experiments were performed on Intel Xeon Platinum 8124M machine equipped with one NVIDIA GeForce RTX 3090 and 128 GB of memory. We implemented PGSL using PyTorch and trained it with Adam optimizer.

5.2. Evaluations

The evaluations are performed under two settings, i.e., propagation under synthetic diffusion (SI and SIR) and under real-world diffusion. The advantages and limitations of our proposed PGSL are also discussed.

5.2.1. Performance under synthetic diffusion

Tables 3 and 4 show the comparison results between our model and baselines under SI and SIR diffusion algorithms, respectively. We can see that our model significantly outperformed most of the source localization baselines (except SL-VAE) in terms of all metrics on all datasets. Moreover, all five baselines exhibited inferior performance in SIR diffusion compared to SI, owing to the more complex diffusion processes of the SIR. In contrast, the performance of our model is robust in both situations. However, we found that our model performed relatively poorer on the *Jazz* and *NS* datasets in comparison to SL-VAE model. There are several potential reasons that degenerate our model’s learning ability on certain datasets.

First, the sizes of *Jazz* and *NS* are significantly smaller than other datasets. Since the performance of our model is largely driven by the proposed NF and GNN modules that rely on more training instances and larger network sizes, such small networks may not be able to provide sufficient diffusion patterns for our model to capture data distributions

Table 3
Performance comparison under SI diffusion.

Model	Metric	Data			
		Jazz	NS	CML	PG
LPSI	PR	0.105	0.423	0.155	0.454
	RE	0.478	0.604	0.595	0.495
	F1	0.171	0.497	0.246	0.473
	AUC	0.484	0.837	0.667	0.933
GCNSI	PR	0.158	0.137	0.118	0.141
	RE	0.436	0.224	0.361	0.347
	F1	0.232	0.171	0.178	0.209
	AUC	0.642	0.475	0.538	0.504
OJC	PR	0.101	0.204	0.104	0.371
	RE	0.180	0.224	0.287	0.123
	F1	0.129	0.213	0.153	0.185
	AUC	0.505	0.563	0.501	0.533
NetSleuth	PR	0.109	0.265	0.498	0.328
	RE	0.132	0.265	0.597	0.395
	F1	0.119	0.265	0.543	0.359
	AUC	0.543	0.469	0.765	0.653
SL-VAE	PR	0.719	0.599	0.571	0.589
	RE	0.947	0.935	0.899	0.932
	F1	0.818	0.729	0.697	0.721
	AUC	0.978	0.949	0.941	0.944
PGSL	PR	0.490	0.591	0.611	0.591
	RE	0.778	0.950	0.956	0.963
	F1	0.600	0.727	0.744	0.731
	AUC	0.886	0.944	0.948	0.953

Table 4
Performance comparison under SIR diffusion.

Model	Metric	Data			
		Jazz	NS	CML	PG
LPSI	PR	0.115	0.136	0.107	0.486
	RE	0.363	0.432	0.477	0.472
	F1	0.169	0.207	0.175	0.478
	AUC	0.501	0.561	0.498	0.582
GCNSI	PR	0.141	0.104	0.115	0.113
	RE	0.373	0.351	0.338	0.237
	F1	0.205	0.161	0.172	0.153
	AUC	0.641	0.543	0.532	0.503
OJC	PR	0.154	0.141	0.141	0.398
	RE	0.220	0.168	0.168	0.123
	F1	0.181	0.154	0.154	0.180
	AUC	0.501	0.511	0.501	0.510
SL-VAE	PR	0.503	0.571	0.582	0.580
	RE	0.787	0.942	0.919	0.933
	F1	0.613	0.709	0.711	0.714
	AUC	0.789	0.951	0.930	0.947
PGSL	PR	0.490	0.573	0.610	0.603
	RE	0.778	0.937	0.946	0.944
	F1	0.600	0.710	0.741	0.734
	AUC	0.873	0.943	0.943	0.952

and diffusion structures. This is also reflected in the better performance results seen on larger networks such as CML, PG, Twitter and Weibo.

Second, our model is more suitable for handling complex propagation protocols given the powerful learning abilities of the proposed probabilistic framework using normalizing flows with invertible transformations and graph neural networks. This is demonstrated by the better performance of our model under the SIR diffusion, where all baselines’ performances are decreased. Given the above discussions, we suggest using PGSL for larger networks with complex diffusion patterns.

5.2.2. Performance under real-world diffusion

We now evaluate the performance of our model and baselines under the real-world information diffusion setting. The comparison results on *Weibo* and *Twitter* datasets are shown in Table 5. The results indicate

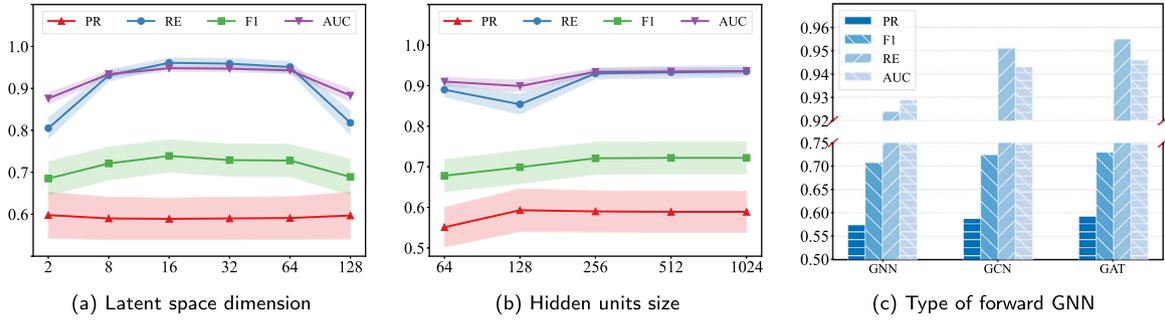


Fig. 8. Parameter analysis of PGSL on CML dataset (SI).

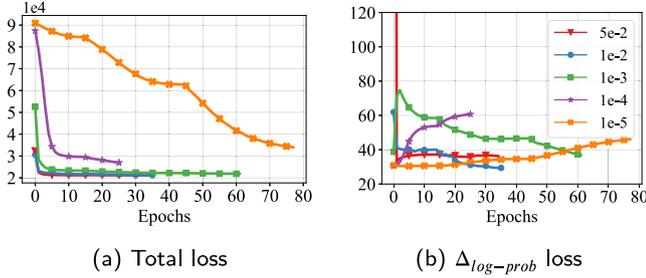


Fig. 9. Convergence analysis of PGSL on CML dataset (SI).

Table 5

Performance comparison under real-world diffusion.

Data	Metric	Model			
		LPSI	GCNSI	SL-VAE	PGSL
Weibo	PR	0.101	0.009	0.112	0.386
	RE	0.008	0.007	0.011	0.073
	F1	0.013	0.008	0.010	0.061
	AUC	0.499	0.501	0.501	0.533
Twitter	PR	0.103	0.112	0.134	0.417
	RE	0.009	0.011	0.013	0.088
	F1	0.015	0.024	0.024	0.142
	AUC	0.496	0.509	0.504	0.586

that all models experienced a drop in performance, mainly due to the complexity of underlying diffusion patterns in the real world and the classification imbalance that arises from the increasing graph size. Despite this, PGSL significantly outperformed SL-VAE, suggesting that VAEs may not sufficiently model the uncertainty of diffusion sources in real-world datasets. The improvement verifies our motivation to integrate normalizing flows with invertible transformations to explicitly handle the uncertainty of the diffusion process. Additionally, the varying number of sources in each instance of the real-world datasets, caused by different lengths of the diffusion cascades, is distinct from the synthetic datasets (which has a fixed rate of the sources), resulting in a significant performance drop compared to the results of models on synthetic datasets.

5.3. Parameter analysis

To investigate the significance of hyperparameters in PGSL, we experimented with different parameter values to analyze their impacts on model performance. The results are shown in Fig. 8.

5.3.1. Latent space dimension

The dimension of latent space (z-dim) in NF bijector determines the space size when generating the source nodes. We set the z-dim in NF bijector to [2, 8, 16, 32, 64, 128]. From Fig. 8(a) we can see that the best performance is obtained when the z-dim is around 16–32.

The Precision changes smoothly while the Recall changes significantly when z-dim varies from 2 to 128. Additionally, the error bands widen when z-dim is either too small or too large. A z-dim of 16 is sufficient to represent the latent features of sources distributions. Increasing the z-dim results in a significant increase in parameters, leading to an overfitting problem during model training.

5.3.2. Hidden unit size

The size of hidden units in the NF bijector is another important hyperparameter in PGSL. We vary the size of hidden units from 64 to 1024, and the results are shown in Fig. 8(b). It can be observed that PGSL achieved the best and worst performances when the hidden units are set to 256 and 64, respectively. This result can be explained by the fact that smaller hidden units cannot fully represent the features of nodes in the graph.

5.3.3. GNN layer type

The type of GNN layers in the forward model affects the graph learning capability of PGSL and may significantly influence the source localization performance. We compare three types of GNN layers (simple GNN, GCN, and GAT), and the result is shown in Fig. 8(c). It can be observed that all the metrics reach the best when GAT is used as base of forward model and the worst for simple GNN. This result may be due to the usage of attention mechanisms in GAT, which can learn diffusion patterns more effectively. Additionally, this result suggests that PGSL is flexible with various forward base models and can adapt to specific diffusion patterns that can be captured by the graph neural networks.

5.3.4. Dropout rate and attention heads

We further investigate the effects of two important hyperparameters: dropout rate and the number of attention heads in GAT. The results on the NS dataset under SI diffusion are shown in Figs. 10 and 11. We can observe that for dropout rate a value around 0.5 achieves the best performance. For number of heads in GAT, we find that the value of this hyperparameter has minimal influence on the model's performance.

5.4. Convergence analysis

We now conduct convergence analysis of our model. We adopted cosine annealing method to control the learning rate of PGSL on different learning rate settings from [$5e^{-2}$, $1e^{-2}$, $1e^{-3}$, $1e^{-4}$, $1e^{-5}$]. We compared the changes of overall loss and $\Delta_{\log-prob}$ loss during training and the results are shown in Figs. 9(a) and 9(b), respectively. In addition, we adopted an early-stop strategy with 10 epochs of patience on each model training to prevent overfitting.

As demonstrated in Fig. 9(a), we observe that for learning rate larger than $1e^{-4}$, the total loss decreases rapidly within the first five epochs, and the training terminates between 30 to 50 epochs. However, when the learning rate is set to $1e^{-5}$, the loss curve declines much slower, suggesting that the model may not converge when learning rate is too small.

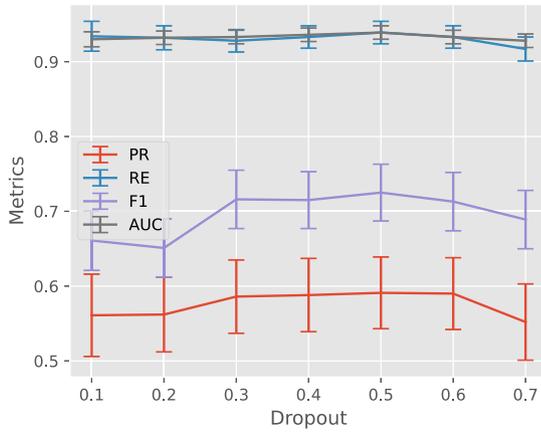


Fig. 10. The effect of dropout rate for mitigating the overfitting issue on NS dataset under SI diffusion.

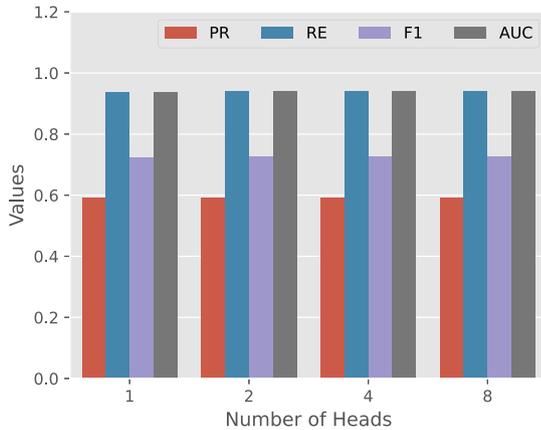


Fig. 11. The effect of the number of attention heads in GAT on NS dataset under SI diffusion.

As shown in Fig. 9(b), we can find that when the learning rate is larger than $1e^{-3}$, the $\Delta_{log-prob}$ loss keeps increasing after the training reaches a certain point. When the learning rate is set to $5e^{-2}$, this phenomenon occurs again. These results indicate that the convergence of neural ODEs is significantly affected by very high or very low learning rates due to the stiffness of ODEs.

5.5. Ablation study

The NF and GNN are two critical modules in our model. To investigate their effects on the model performance, we conduct ablation studies and designed two variants of PGSL. Specifically, we remove the NF layer in the NF bijector, which simplifies the model into a plain autoencoder. We denote this variant as *PGSL w/o NF*. We also replace the GNN module with FC layers in the forward model, and denote this variant as *PGSL w/o GNN*. The performance changes on six datasets under SI propagation or real-world diffusion (Weibo and Twitter) are shown in Table 6. We have the following findings: (1) The removal of any module within *PGSL* leads to a decline in performance, demonstrating that both modules significantly contribute to the overall efficacy of the model and both modules collaborate with each other in inferring the diffusion sources; (2) The performance drops of *PGSL w/o NF* can be attributed to the failure in effectively capturing the uncertainties associated with the sources, which becomes more apparent when the network size is larger, as in the *Weibo* and *Twitter* datasets. (3) The aim of the GNN module is to learn node interactions and network structures, its usefulness is more significant for improving

Table 6
Ablation study on six datasets under SI propagation.

Data	Metric	Model		
		w/o NF	w/o GNN	PGSL
Jazz	PR	0.496	0.527	0.547
	RE	0.741	0.789	0.791
	F1	0.593	0.631	0.645
	AUC	0.837	0.866	0.883
NS	PR	0.575	0.559	0.591
	RE	0.912	0.900	0.917
	F1	0.718	0.688	0.718
	AUC	0.927	0.917	0.947
CML	PR	0.588	0.591	0.596
	RE	0.949	0.927	0.949
	F1	0.725	0.721	0.731
	AUC	0.943	0.932	0.944
PG	PR	0.581	0.591	0.593
	RE	0.892	0.950	0.963
	F1	0.703	0.729	0.731
	AUC	0.915	0.943	0.953
Weibo	PR	0.326	0.374	0.386
	RE	0.070	0.071	0.073
	F1	0.055	0.059	0.061
	AUC	0.516	0.528	0.533
Twitter	PR	0.397	0.414	0.417
	RE	0.082	0.086	0.088
	F1	0.136	0.141	0.142
	AUC	0.571	0.573	0.586

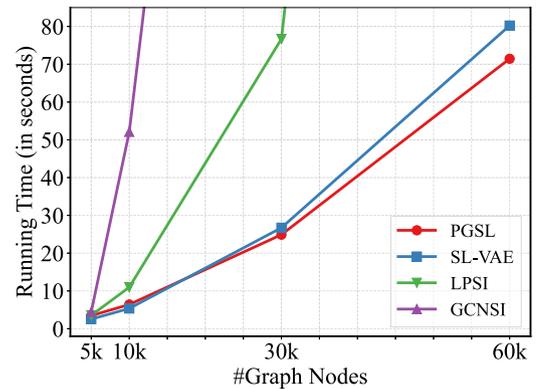


Fig. 12. Running time comparison with different graph sizes.

the Recall metric. In other words, it aids in identifying as many sources as possible. (4) The significance of NF and GNN modules varies across different source localization datasets. In practical scenarios, their usage should be tailored to the unique characteristics of individual datasets. For instance, the PG and Twitter datasets exhibit less dependence on network structures and are more susceptible to the uncertainties associated with diffusion.

5.6. Model scalability

The overall complexity of GCNSI, as proven in Dong et al. (2019) and Wang et al. (2017), is $\mathcal{O}(k \times D^3)$, while the overall complexity of LPSI is $\mathcal{O}(D^3)$, where D is the dimension of the input sources (i.e., graph size), and k is the number of layers in GCNSI. In contrast, SL-VAE has linear run-time with the growth of graph size (Ling et al., 2022) and its complexity only depends on the complexity of the forward model. In PGSL, we leverage continuous normalizing flows, which reduces the overall complexity to $\mathcal{O}((D \times H + D)L)$ (linear to the graph size D) compared to other methods. We use the GAT model as the forward model, whose complexity is $\mathcal{O}(|V|) + \mathcal{O}(|E|)$ with fixed feature dimension and attention heads. Therefore, the complexity of

PGSL is comparable to SL-VAE. We show the run-time of our model and baselines with different numbers of nodes (5k, 10k, 30k, 60k) on Weibo dataset in Fig. 12. It is observed that the run-times of PGSL and SL-VAE increase linearly with the size of graph nodes, while LPSI and GCNSI are not scalable to large graphs.

5.7. Limitation on real-world diffusion datasets

The real-world diffusions are much more complex than synthetic diffusions, posing both internal and external correlations and interferences that influence the diffusion process. In our work, we utilized two real-world diffusion datasets – Weibo and Twitter – constructed from information cascades. We note that for these two real-world diffusion datasets, we used merging operations on multi-sourced cascades to ensure a balanced dataset.

For future improvements of multi source localization problem on real-world diffusion cascades, we suggest the following directions. First, randomly merging cascades did not take the relations between cascades into consideration. Future efforts can be devoted to designing more advanced ways of selecting cascades for constructing the diffusion instances, e.g., clustering techniques. Second, researchers can propose new methods for addressing the data imbalance issue (diffused vs. entire graph) during the training and inference phases. Third, we can decouple the entire underlying graph from identifying sources. On the one hand, we can efficiently identify sources from the local subgraph or node neighborhood. On the other hand, we can concentrate on individual diffusions that accurately simulate the whole diffusion process.

6. Conclusion

Localizing the sources of diffusions in graph is a challenging task with a wide range of real-world applications. In this paper, we propose a novel probabilistic model PGSL, which addresses the inherent uncertainty problem in inverse graph diffusion. PGSL uses continuous normalizing flows to construct a reverse deep generative model, which can generate expressive posteriors with smooth and invertible transformations to learn the intrinsic features of diffusion sources. Extensive experiments conducted on synthetic datasets and real-world diffusion datasets demonstrate the effectiveness of PGSL over strong baselines.

In the future, we plan to extend our model to incorporate other features, such as texts or images, to enhance the performance of specific applications. Moreover, the current model can only operate on static graphs and with only the nodes' states. Therefore, other types of graph learning models should be considered to extend the model's applicability, such as heterogeneous information networks or dynamical graph neural networks.

CRedit authorship contribution statement

Xovee Xu: Conceptualization, Methodology, Investigation, Writing – original draft. **Tangjiang Qian:** Methodology, Writing – original draft, Software, Visualization. **Zhe Xiao:** Conceptualization, Writing – review & editing. **Ni Zhang:** Conceptualization, Writing – review & editing. **Jin Wu:** Conceptualization, Writing – review & editing, Supervision. **Fan Zhou:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition, Project administration.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62072077 and 62176043, in part by the Natural Science Foundation of Sichuan Province, China, under Grant 2022NSFSC0505.

References

- Agaskar, A., & Lu, Y. M. (2013). A fast Monte Carlo algorithm for source localization on graphs. In *Wavelets and sparsity XV*, vol. 8858 (pp. 429–434).
- Allen, L. J. (1994). Some discrete-time SI, SIR, and SIS epidemic models. *Mathematical Biosciences*, 124(1), 83–105.
- Anderson, R. M., & May, R. M. (1992). *Infectious diseases of humans: Dynamics and control*. Oxford University Press.
- Bass, F. M. (1969). A new product growth for model consumer durables. *Management Science*, 15(5), 215–227.
- Bielski, A., & Trzcinski, T. (2018). Understanding multimodal popularity prediction of social media videos with self-attention. *IEEE Access*, 6, 74277–74287.
- Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. In *NeurIPS*, vol. 31.
- Chen, W., Wang, C., & Wang, Y. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *SIGKDD* (pp. 1029–1038).
- Cheng, J., Adamic, L., Dow, P. A., Kleinberg, J. M., & Leskovec, J. (2014). Can cascades be predicted?. In *WWW* (pp. 925–936).
- Dong, M., Zheng, B., Hung, N. Q. V., Su, H., & Li, G. (2019). Multiple rumor source detection with graph convolutional networks. In *GIM* (pp. 569–578).
- Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., & Song, L. (2016). Recurrent marked temporal point processes: Embedding event history to vector. In *SIGKDD* (pp. 1555–1564).
- Easley, D., & Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press.
- Germain, M., Gregor, K., Murray, I., & Larochelle, H. (2015). Made: Masked autoencoder for distribution estimation. In *ICML* (pp. 881–889).
- Gleiser, P. M., & Danon, L. (2003). Community structure in Jazz. *Advances in Complex Systems*, 6(04), 565–573.
- Goldenberg, J., Libai, B., & Muller, E. (2001). Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3), 211–223.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *NIPS*.
- Goyal, A., Lu, W., & Lakshmanan, L. V. (2011). Celf++ optimizing the greedy algorithm for influence maximization in social networks. In *WWW* (pp. 47–48).
- Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology*, 83(6), 1420–1443.
- Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., & Duvenaud, D. (2019). FFJORD: free-form continuous dynamics for scalable reversible generative models. In *ICLR*.
- Guo, Q., Zhang, C., Zhang, H., & Fu, L. (2021). IGCN: Infected graph convolutional network based source identification. In *GLOBECOM* (pp. 1–6). IEEE.
- He, Z., Li, C., Zhou, F., & Yang, Y. (2021). Rumor detection on social media with event augmentations. *SIGIR*.
- Hegde, C. (2018). Algorithmic aspects of inverse problems using generative models. In *Annual allerton conference on communication, control, and computing* (pp. 166–172).
- Huang, C.-W., Krueger, D., Lacoste, A., & Courville, A. (2018). Neural autoregressive flows. In *ICML* (pp. 2078–2087).
- Jain, L., Katarya, R., & Sachdeva, S. (2023). Opinion leaders for information diffusion using graph neural network in online social networks. *ACM Transactions on the Web*.
- Jiang, J., Wen, S., Yu, S., Xiang, Y., & Zhou, W. (2016). Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys & Tutorials*, 19(1), 465–481.
- Karamchandani, N., & Franceschetti, M. (2013). Rumor source detection under probabilistic sampling. In *IEEE international symposium on information theory* (pp. 2184–2188).
- Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *SIGKDD* (pp. 137–146).
- Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*.
- Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, (4), 307–392.
- Ko, J., Lee, K., Shin, K., & Park, N. (2020). Monstor: an inductive approach for estimating and maximizing influence over unseen networks. In *ASONAM* (pp. 204–211).
- Kong, Z., & Chaudhuri, K. (2020). The expressive power of a class of normalizing flow models. In *International conference on artificial intelligence and statistics* (pp. 3599–3609).
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR* (pp. 4681–4690).

- Leung, C. K., Cuzzocrea, A., Mai, J. J., Deng, D., & Jiang, F. (2019). Personalized DeepInf: enhanced social influence prediction with deep learning and transfer learning. In *IEEE international conference on big data* (pp. 2871–2880).
- Li, C., Ma, J., Guo, X., & Mei, Q. (2017). Deepcas: An end-to-end predictor of information cascades. In *WWW* (pp. 577–586).
- Ling, C., Liang, J., Wang, J., & Zhao, L. (2022). Source localization of graph diffusion via variational autoencoders for graph inverse problems. In *SIGKDD* (pp. 1010–1020).
- Luo, W., & Tay, W. P. (2012). Identifying infection sources in large tree networks. In *Annual IEEE communications society conference on sensor, mesh and Ad Hoc communications and networks* (pp. 281–289).
- Luo, W., Tay, W. P., & Leng, M. (2013). Identifying infection sources and regions in large networks. *IEEE Transactions on Signal Processing*, 61(11), 2850–2865.
- Luo, W., Tay, W. P., & Leng, M. (2014). How to identify an infection source with limited observations. *IEEE Journal of Selected Topics in Signal Processing*, 8(4), 586–597.
- McCallum, A. K., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2), 127–163.
- Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), Article 036104.
- Nguyen, D. T., Nguyen, N. P., & Thai, M. T. (2012). Sources of misinformation in online social networks: Who to suspect? In *IEEE military communications conference* (pp. 1–6).
- Nickel, M., & Le, M. (2021). Modeling sparse information diffusion at scale via lazy multivariate Hawkes processes. In *The web conference* (pp. 706–717).
- Pinto, P. C., Thiran, P., & Vetterli, M. (2012). Locating the source of diffusion in large-scale networks. *Physical Review Letters*, 109(6), Article 068702.
- Prakash, B. A., Vreeken, J., & Faloutsos, C. (2012). Spotting culprits in epidemics: How many and which ones? In *ICDM* (pp. 11–20).
- Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., & Tang, J. (2018). Deepinf: Social influence prediction with deep learning. In *SIGKDD* (pp. 2110–2119).
- Rezeman, D., & Mohamed, S. (2015). Variational inference with normalizing flows. In *ICML* (pp. 1530–1538).
- Seo, E., Mohapatra, P., & Abdelzaher, T. (2012). Identifying rumors and their sources in social networks. In *Ground/Air multisensor interoperability, integration, and networking for persistent ISR III*, vol. 8389 (pp. 417–429).
- Shah, D., & Zaman, T. (2011). Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory*, 57(8), 5163–5181.
- Shen, H., Wang, D., Song, C., & Barabási, A.-L. (2014). Modeling and predicting popularity dynamics via reinforced poisson processes. In *AAAI*, vol. 28.
- Wang, J., Jiang, J., & Zhao, L. (2022). An invertible graph diffusion neural network for source localization. In *WWW* (pp. 1058–1069).
- Wang, Y., Vasilakos, A. V., Ma, J., & Xiong, N. (2014). On studying the impact of uncertainty on behavior diffusion in social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2), 185–197.
- Wang, Z., Wang, C., Pei, J., & Ye, X. (2017). Multiple source detection without knowing the underlying propagation model. In *AAAI* (pp. 217–223).
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684), 440–442.
- Weng, L., Menczer, F., & Ahn, Y.-Y. (2013). Virality prediction and community structure in social networks. *Scientific Reports*, 3(1), 1–6.
- Xia, W., Li, Y., Wu, J., & Li, S. (2021). Deepis: Susceptibility estimation on social networks. In *WSDM* (pp. 761–769).
- Xie, J., Zhu, Y., Zhang, Z., Peng, J., Yi, J., Hu, Y., et al. (2020). A multimodal variational encoder-decoder framework for micro-video popularity prediction. In *WWW* (pp. 2542–2548).
- Xu, X., Zhong, T., Li, C., Trajcevski, G., & Zhou, F. (2022). Heterogeneous dynamical academic network for learning scientific impact propagation. *Knowledge-Based Systems*, 238, Article 107839.
- Xu, X., Zhou, F., Zhang, K., & Liu, S. (2022). CCGL: Contrastive cascade graph learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.
- Xu, X., Zhou, F., Zhang, K., Liu, S., & Trajcevski, G. (2021). CasFlow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, <http://dx.doi.org/10.1109/TKDE.2021.3126475>.
- Yang, J., & Counts, S. (2010). Predicting the speed, scale, and range of information diffusion in twitter. In *ICWSM*, vol. 4 (pp. 355–358).
- Ying, L., & Zhu, K. (2018). Diffusion source localization in large networks. *Synthesis Lectures on Communication Networks*, 11(1), 1–95.
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In *CVPR* (pp. 5505–5514).
- Yu, L., Xu, X., Trajcevski, G., & Zhou, F. (2022). Transformer-enhanced Hawkes process with decoupling training for information cascade prediction. *Knowledge-Based Systems*, 255, Article 109740.
- Zhang, Q., Gong, Y., Wu, J., Huang, H., & Huang, X. (2016). Retweet prediction with attention-based deep neural network. In *CIKM* (pp. 75–84).
- Zhou, F., Xu, X., Trajcevski, G., & Zhang, K. (2021). A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Computing Surveys*, 54(2).
- Zhou, C., Zhang, P., Guo, J., Zhu, X., & Guo, L. (2013). Ublf: An upper bound based approach to discover influential nodes in social networks. In *ICDM* (pp. 907–916).
- Zhu, K., Chen, Z., & Ying, L. (2017). Catch'em all: Locating multiple diffusion sources in networks with partial observations. In *AAAI*, vol. 31.
- Zhu, K., & Ying, L. (2014). Information source detection in the SIR model: A sample-path-based approach. *IEEE/ACM Transactions on Networking*, 24(1), 408–421.