# Two-stage Denoising Diffusion Model for Source Localization in Graph Inverse Problems

Bosong Huang¹, Weihao Yu², Ruzhong Xie¹, Jing Xiao¹, and Jin Huang (⊠)¹

<sup>1</sup> South China Normal University, Guangzhou, China {bosonghuang,rzxie,xiaojing,huangjin}@scnu.edu.cn
<sup>2</sup> Research Institute of China Telecom Corporate Ltd., Guangzhou, China yuwh3@chinatelecom.cn

Abstract. Source localization is the inverse problem of graph information dissemination (information diffusion) and has broad practical applications. However, the inherent intricacy and uncertainty in information dissemination pose significant challenges, and the ill-posed nature of the source localization problem further exacerbates these challenges. Recently, deep generative models, particularly diffusion models inspired by classical non-equilibrium thermodynamics, have made significant progress. While diffusion models have proven to be powerful in solving inverse problems and producing high-quality reconstructions, applying them directly to the source localization problem is infeasible for two reasons. Firstly, it is impossible to calculate the posterior disseminated results on a large-scale network for iterative denoising sampling, which would incur enormous computational costs. Secondly, in the existing methods designed for this field, the training data itself are ill-posed (many-to-one); thus simply transferring the diffusion model would only lead to local optima. To address these challenges, we propose a twostage optimization framework, the source localization denoising diffusion model (SL-Diff). In the coarse stage, we devise the source proximity degrees as the supervised signals to generate coarse-grained source predictions. This aims to efficiently initialize the next stage, significantly reducing its convergence time and calibrating the convergence process. Furthermore, the introduction of cascade temporal information in this training method transforms the many-to-one mapping relationship into a one-to-one relationship, perfectly addressing the ill-posed problem. In the fine stage, we design a diffusion model for the graph inverse problem that can quantify the uncertainty in the dissemination process. Thanks to the excellent collaboration of the two stages, the proposed SL-Diff yields excellent prediction results within a reasonable sampling time, as demonstrated in extensive experiments on five datasets.

### 1 Introduction

The exponential growth of network-structured information has aroused widespread interest in studying its dissemination mode [1–4]. This involves modeling the

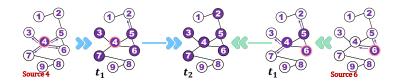


Fig. 1: Many-to-one (ill-posed) relationship when the supervised signal is the source indicator.

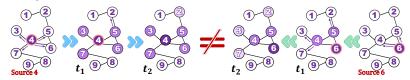


Fig. 2: One-to-one relationship when the supervised signal is the source proximity degree.

information dissemination (information diffusion<sup>3</sup>) based on the corresponding network structure and dissemination source. While there are few research followers, the inverse problem of information dissemination on graphs and source localization is of great practical significance. For instance, locating the source account for spreading rumors in social networks is crucial for rumor detection [5,6]. Similarly, source localization can aid in virus interception [7], malicious email traceability [8], and other areas [9,10].

However, there are currently several challenges hindering this field: **Chall. 1.** Existing models do not fundamentally address the ill-posed problem because their training data are inherently ill-posed (many-to-one). For instance, in the case of blurred image restoration, the ill-posed problem cannot be solved if the training data consist solely of blurred images. In fact, existing methods unfortunately erase the relative temporal correlation of each node from the cascades when setting the training data, which essentially leads to the ill-posed problem of source localization.

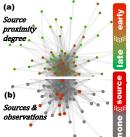


Fig. 3: Two types of supervised signal derived from a cascade.

 $<sup>^3</sup>$  In most studies, this problem is referred to as "graph information diffusion". However, we refer to it as "graph information dissemination" in this paper to disambiguate with the "diffusion model".

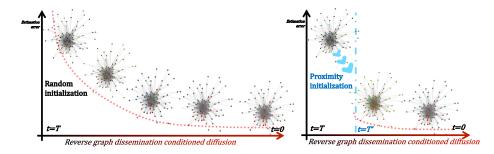


Fig. 4: Random initialization and proximity initialization of sampling.

This abandonment of temporal information also significantly restricts the ability of the model to learn the underlying dissemination pattern. **Chall. 2.** In practical applications, information dissemination in the network will introduce many uncertain factors, that is, the noise irrelevant to the regular dissemination mode. Although some studies [4, 9] have begun to discuss the uncertainty in source localization, how to quantitatively model them and more importantly, eliminate the relevant noise in source localization is unresolved.

The denoising diffusion models [11–13] have flourished recently due to their high-quality reconstructions and powerful inverse problem solving abilities. In addition to leveraging its power and further addressing Chall. 2, we have taken the first step in generalizing this powerful model to the source localization problem. However, the existing family of diffusion models poses the following challenges for direct migration. Chall. 3. As the inverse problem solver, existing diffusion models need to calculate the conditional posterior probability at each diffusion step for Maximum A Posteriori (MAP) approximation. However, in the case of source localization, the calculation of posterior probability involves simulating information dissemination across the entire network structure at each diffusion step, making it extremely computationally expensive. Therefore, it is almost impossible to directly apply existing diffusion models to source localization. Chall. 4. Existing diffusion models [11, 12, 14, 15] that were designed for image or molecule data struggle to model dissemination patterns. For example, the number of nodes in the molecular graph is much smaller than that in the information dissemination network, and does not have the temporal information.

To address the complex challenges outlined above and effectively harness the benefits of current technological developments, we propose a two-stage optimization framework (Figure 5) named Source Localization Denoising Diffusion model (SL-Diff). In general, SL-Diff leverages an efficient initialization that is supervised with temporal cascade information and an exquisitely redesigned denoising network structure to achieve a series of outstanding performances. Specifically, for Chall. 1, we ingeniously retain the timing of the cascade (defined as the source proximity degree, elaborated in Section 3.2) during the model training phase while remaining consistent with baselines that only take the disseminated

#### 4 F. Author et al.

observation as the input for prediction during testing. This means that the model in the coarse stage aims to learn the source proximity relationship (i.e., a kind of one-to one mapping correlation) through training, which directly addresses the core of Chall. 1. On the other hand, due to the contraction property of stochastic differential equations (SDEs), the error reduction of the diffusion model is exponential. Based on this characteristic, we propose a two-stage optimization framework. In the the coarse stage, we use the one-shot method to generate coarse source proximity degrees quickly and efficiently, reducing the predicted source localization error. In the the fine stage, the diffusion model simulates information dissemination across the entire network, accurately locating the source. Concretely, we optimize the verbose sampling process (Figure 4 (left)) into an efficient two-stage process (Figure 4 (right)) and further balance the twostage diffusion step ratio to achieve an optimum between efficiency and accuracy through parameter experiments. This approach effectively solves Chall. 3. Since the coarse stage provides excellent initialization of the coarse source proximity relationships, it prominently alleviates the local optimum problem caused by class imbalance in the source localization problem. In addition, we propose a new uncertainty graph information dissemination model that quantizes the dissemination noise and a score function approximating network that adapts to the underlying dissemination mode on the graph to jointly address Chall. 2 and Chall. 4. In summary, our contributions are as follows:

- We propose a training framework that employs source proximity degree supervision, which fundamentally addresses the ill-posed problem of source localization.
- Our contribution lies in proposing, for the first time, a two-stage denoising diffusion model for the source localization problem. The targeted design of the two stages effectively addresses the challenge of migrating the diffusion model to this field.
- We design a new model that approximates graph information dissemination and a score function approximating network to enhance the performance of the diffusion model for source localization.
- Experiments on five real-world datasets demonstrate that our proposed SL-Diff model outperforms state-of-the-art models. The code for this study is available at https://anonymous.4open.science/r/SL-Diff-275A.

### 2 Related Work

Graph Information Dissemination and Source Localization. Graph information dissemination modeling aims to predict the nodes to be affected given the source nodes, which is one of the foremost technologies in social network analysis, disease infection prediction, etc. Traditional methods [16–18] manually model the respective dissemination pattern for different application fields while suffering from poor generalizability and high computational complexity. With the blooming of deep learning, [4, 19, 20] incorporate recurrent neural networks

to capture the dynamic relationship of dissemination cascades. Graph neural networks have further been introduced to aggregate the node neighboring information and model the dissemination pattern to facilitate prediction.

Source localization aims to infer source nodes given an observed set of nodes, which has essential applications in rumor tracing and infection source discovery. Generally speaking, source localization is used to study the traceability method based on the dissemination model. Early studies [6,10,21] were based on specific dissemination models such as Susceptible-Infected (SI) and Susceptible-Infected-Recovered (SIR). [5] argues that a fixed propagation model to be preset is not necessary. Furthermore, [8] focuses on detecting the source in the early propagation stage to reduce the loss caused by propagation. [7] develops a framework for the inverse of graph dissemination models to detect the source. From another perspective, [9] introduces VAE [22] to probabilistically model the uncertainty in source localization.

**Diffusion Models.** The diffusion model we discuss in this paper is the score-based generative model [12], which applies the stochastic differential equation (SDE) to learn the gradient of the target distribution. The currently popular DDPM [11] is a particular case of it. As the diffusion model separates the noise from the data step by step at a fine-grained level, its powerful generative capabilities have achieved state-of-the-art results in many fields, e.g., image generation [11–14], graph generation [15, 23], and time series generation [24].

### 3 Preliminaries

#### 3.1 Conditional Score-Based Diffusion Models

Diffusion models aim to approximate the prior distributions by learning the noise of the data reversely. [12] combines SMLD [14] and DDPM [11] into a generalized theoretical framework, known as the Score-Based Diffusion Model. They generally map data to a noise distribution (the prior) with a stochastic differential equation (SDE), and reverse this SDE for generative modeling.

**Forward SDE.** Given the i.i.d. original dataset samples  $\mathbf{x}(0) \sim p_{\mathbf{x}}$ , which are further indexed as  $\mathbf{x}(t)_{t=0}^T$  by the diffusion step  $t \in [0, T]$  to indicate the noise degree, the diffusion process can be modeled as the solution to an Itô SDE:

$$d\mathbf{x} = -\frac{\beta(t)}{2}\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}$$
 (1)

where  $\beta(t) \in \mathbb{R}$  is the noise schedule that we uniformly adopt the one in [11] in this paper, **w** is the Brownian motion.

**Reverse SDE.** [25] clarifies that the reverse of a diffusion process is also a diffusion process, which can be modeled as the reverse SDE:

$$d\mathbf{x} = \left[ -\frac{\beta(t)}{2} \mathbf{x} - \beta(t) \nabla_{\mathbf{x}_t} \log p_t \left( \mathbf{x}_t \right) \right] dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}, \tag{2}$$

where dt is the negative diffusion step from T to 0 and  $\bar{\mathbf{w}}$  is the corresponding Brownian motion of the reverse process. Since the direct estimation of  $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$  is too computationally intensive and badly generalizatial, we train a score-based model to approximate it:

$$\theta^* = \underset{\theta}{\operatorname{arg\,min}} \mathbb{E}_{t \sim U(\varepsilon, 1), \mathbf{x}(t) \sim p(\mathbf{x}(t) | \mathbf{x}(0)), \mathbf{x}(0) \sim p_{\text{data}}} [\xi]$$

$$\xi = \|\mathbf{s}_{\theta}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}_{t}} \log p(\mathbf{x}(t) | \mathbf{x}(0))\|_{2}^{2}$$
(3)

Here  $\varepsilon \simeq 0$  is the small positive constant. After acquiring the trained score function approximation network, the initial noise can be denoised to enable sampling based on specific posterior conditions, leading to the generation of expected data.

#### 3.2 Graph Information dissemination

The graph information dissemination problem involves a graph G = (V, E) with edge set E and node set V. An information dissemination cascade  $D_i$  at length K+1 on the graph is defined as  $D_i = \{(v_{i_k}, k) \mid v_{i_k} \in V, k = 0, 1, \ldots, K-1, K\}$ , where k is non-decreasing and the first  $K_s$  nodes are defined as source nodes. The source indicator  $\mathbf{i} \in \{0, 1\}^{|V|}$  is defined as 0 for being the source and 1 for not, while the source proximity degree  $\mathbf{x} \in [0, 1]^{|V|}$  is defined as  $x_k = \frac{k}{K}$ . The set of affected nodes' observations is denoted as  $\mathbf{y} \in \{0, 1\}^{|V|}$ , where 0 indicates being affected and 1 indicates not being affected. The goal of the graph information dissemination problem is to predict the affected nodes given the source indicator and graph structure.

Thanks to the outstanding performance of GNN on graph data, the state-of-the-art graph information dissemination methods [1,3,4,26] construct various realistic-meaning attribute variables through GNN at the first stage, and then perform variable dissemination to derive the final affected node sets. Specifically: (1) In the variable construction, we define a neural network  $\mathbf{v} = g_{\mathbf{w_1}}(\mathbf{x})$  to construct miscellaneous variables of nodes (such as sender variable or receiver variable). (2) In variable dissemination, we define a dissemination neural network  $f_{\mathbf{w_2}}(\mathbf{v})$  to propagate information to neighbour nodes according to the topology of the graph, where  $w_1$  and  $w_2$  are learnable parameters. Thus, the general paradigm of graph information dissemination can be defined as:

$$\mathbf{y} = f_{\mathbf{w}_2} \left( g_{\mathbf{w}_1}(\mathbf{x}) \right) \tag{4}$$

#### 3.3 Problem Formulation for Source Localization

In order to eliminate ambiguity, the graph mentioned in Subsection 3.2 is referred to as the whole graph  $\mathcal{G}^w = (V^w, E^w)$ . Source localization is the corresponding inverse problem of graph information dissemination, which is formally defined as: given the whole graph information  $\mathcal{G}^w = (V^w, E^w)$  and affected nodes  $\mathbf{y} \in \{0,1\}^{|V|}$ , reconstruct the source nodes  $\hat{\mathbf{i}} \in \{0,1\}^{|V|}$ . For the convenience of mentioning later, we define the cascade graph as  $\mathcal{G}^c_i = (V^c_i \in D_i, E^c)$  where  $E^c = \{(u,v) \in E^w \mid u \in D_i \text{ or } v \in D_i\}$ .

### 4 SL-Diff Method

### 4.1 Two-stage Optimization Framework

To mitigate the computational burden of simulating information dissemination on the whole graph at each diffusion step (chall. 3), and to leverage the cascade temporal information for addressing the ill-posed problem (chall. 3), we propose a two-stage optimization framework, as depicted in Figure 5. Specifically, in the coarse stage, the supervising signals for the model are the source proximity degrees (Figure 3(a)) that fully retain the cascade temporal information. In other words, the final outputs of this stage are the relatively coarse-grained node infection sequence predictions, which are used to initial-

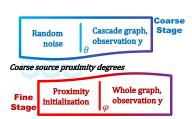


Fig. 5: Two-stage optimization framework

ize the precise source localization for the next stage efficiently. In the second stage, i.e., the fine stage, the supervising signals of model training are source indicators (Figure 3(b)), which aim to accurately output the predicted source under the given disseminated observation conditions. In this stage, the model simulates the information dissemination on the graph at each diffusion step to calibrate predictions.

### 4.2 Coarse Proximity Generation

In this section, we propose a coarse proximity generation model at the first stage, which serves to effectively initialize the subsequent stage. The "coarse" is used in two senses. Firstly, the source proximity degree derived from the cascade data does not strictly reflect the infection relationships between nodes, as the nodes in a cascade may be infected by their common ancestors rather than by their directly adjacent forward nodes. Secondly, as no dissemination model is incorporated at this stage, it is not feasible to accurately predict the dissemination source in reverse. Nonetheless, the generated coarse source proximity degrees are well-suited for the initialization of the downstream diffusion

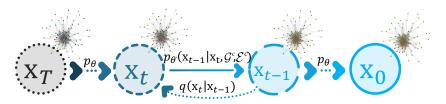


Fig. 6: The forward diffusion and reverse process of the coarse stage.

model. Although the coarse source proximity generation occurs only within the cascade graph, it is incomplete without considering the structure of the entire graph. This is due to the significant impact that the relative position of each node in the graph has on information dissemination. To address this, we utilize Position-aware Graph Neural Networks (P-GNN) [27] to perform node position representation learning on the whole graph of each dataset. Through this process, we obtain the positional embeddings  $\mathcal{E}^w = \{\mathbf{e}_i \mid \mathbf{e}_i = \text{P-GNN}(v_i), v_i \in V^w\}$  that reflect the relative position of each node with respect to other nodes on the whole graph. We use this positional representation in the specific cascade  $\mathcal{E}^c = \{\mathbf{e}_i \mid \mathbf{e}_i = \text{P-GNN}(v_i), v_i \in V^c\}$  as a conditional input to the score function approximation network, enhancing its ability to fit the underlying features of the cascades.

Let the original source proximity degrees derived from a cascade be denoted as  $\mathbf{x}_0$ . The reverse SDE here follows the Equation 2, while the score function conditioned on the the cascade graph structure and the positional embeddings is changed to  $\mathbf{s}_{\theta}(\mathbf{x}_t, t \mid \mathcal{G}^c, \mathcal{E}^c)$ . We adopt the same variance schedule as DDPM [11] to discretize the formula, then the estimated reverse Markov chain is defined as follows:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1 - \beta_t}} \left( \mathbf{x}_t + \beta_t \mathbf{s}_{\boldsymbol{\theta}^*} \left( \mathbf{x}_t, t \mid \mathcal{G}^c, \mathcal{E}^c \right) \right) + \sqrt{\beta_t} \mathbf{z}_t, \quad t = T, T - 1, \cdots, 1 \quad (5)$$

in which we use the following re-weighted variant of the evidence lower bound (ELBO) to train the model to obtain the estimated score function:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{arg\,min}} \sum_{t=1}^{N} (1 - \alpha_t) \, \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{p_{\alpha_t}(\mathbf{x}_t \mid \mathbf{x})} \left[ \| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t \mid \mathcal{G}^c, \mathcal{E}^c) - \nabla_{\mathbf{x}_t} \log p_{\alpha_t}(\mathbf{x}_t \mid \mathbf{x}) \|_2^2 \right]$$
(6)

where  $\alpha := \prod_{j=1}^{i} (1 - \beta_j)$ .

To efficiently approximate the score functions, we propose a new network structure that properly introduces conditional position embeddings while preserving nodes' adjacent characteristics at different orders. Our approach utilizes a multi-head graph attention (GMT) [28] as the basic operation of graph convolution, allowing us to aggregate information across L-layer graphs. Furthermore, we leverage skip connections between each layer to improve the flow of information. Finally, we concatenate the output of each layer and perform multi-layer

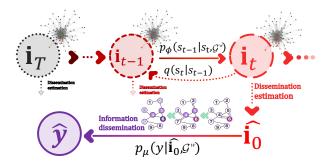


Fig. 7: The forward diffusion and reverse process of the fine stage.

nonlinear transformations to obtain an estimated score function. The proposed conditional score function approximating network is defined as follows:

$$\mathbf{H}^{l+1} = \mathrm{GMT}(\mathbf{H}^{l}, \mathcal{G}^{c}) + \mathbf{H}^{l}$$

$$\mathbf{s}_{\theta}(\mathbf{x}_{t}, t \mid \mathcal{G}^{c}, \mathcal{E}^{c}) = \mathrm{MLP}\left(\mathbf{catenate}[\mathcal{E}^{c}, \mathbf{H}^{0}, \dots, \mathbf{H}^{L-1}, \mathbf{H}^{L}, \mathcal{T}(t)]\right)$$
(7)

where  $\mathbf{H}^0 = \mathbf{x}_t + \mathcal{E}^c$ . We use the positional encodings [29] to encode the diffusion step, and the formula is defined as:  $\mathcal{T}(t) = \left[\dots, \cos\left(t/r^{\frac{-2d}{D}}\right), \sin\left(t/r^{\frac{-2d}{D}}\right), \dots\right]^{\mathrm{T}}$ , where  $d = 1, \dots, D/2$  is the dimension of the embedding, and r is a large constant (set to  $10^5$ ). After obtaining the trained network  $\mathbf{s}_{\theta}$ , we sample according to the Algorithm 1 and get the predicted coarse source proximity degrees.

### 4.3 Graph Dissemination Conditioned Model

In the second stage, we apply the estimated source proximity degree to initialize the graph dissemination conditioned diffusion model. Significantly different from the conditional diffusion in the first stage, the diffusion model here is an inverse problem solver, which aims to recover the source indicator i from the disseminated observation y. The connection between them is the forward information dissemination process of  $i \rightarrow y$ .

If the assumption is the same as in [12], that is,  $p_t(\mathbf{y} \mid \mathbf{i}_t)$  is tractable, then the reverse SDE of the inverse problem is defined as:

$$d\mathbf{i} = \left[ -\frac{\beta(t)}{2} \mathbf{i} - \beta(t) \left( \nabla_{\mathbf{i}_t} \log p_t \left( \mathbf{i}_t \right) + \nabla_{x_t} \log p_t \left( \mathbf{y} \mid \mathbf{i}_t, \mathcal{G}^w \right) \right) \right] dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}, \quad (8)$$

However, in our case,  $p_t(\mathbf{y} \mid \mathbf{i}_t)$  is intractable, as the dissemination model cannot support the source with noise as input. To bridge this gap, we introduce the approximation method [13] for  $p(\hat{\mathbf{i}}_0 \mid \mathbf{i}_t)$ :

$$\hat{\mathbf{i}}_0 \simeq \frac{1}{\sqrt{\bar{\alpha}(t)}} \left( \mathbf{i}_t + (1 - \bar{\alpha}(t)) \mathbf{s}_{\phi} \left( \mathbf{i}_t, t \mid \mathcal{G}^w \right) \right) \tag{9}$$

where  $\alpha_i \triangleq 1 - \beta_i$ ,  $\bar{\alpha}_i \triangleq \prod_{i=1}^i \alpha_i$  following [11]. Now the problem lies in determining an appropriate dissemination model to calculate the posterior probability of y from estimated i. In real-world applications, various random factors can impact the information dissemination process, which are not accounted for in the basic dissemination function (Equation 4). Our proposed model addresses this challenge by qualifying the uncertain trough Gaussian noise  $\tilde{\sigma}(t)$  with adaptive variance (parameterized by MLP and taking diffusion step as input). Additionally, our model can eliminate interference from noise to source localization through the powerful iterative denoising process of the diffusion model. Thus, the formula is defined as follows:

$$\mathbf{y} = f(g(\mathbf{i})) + \tilde{\sigma}(t)\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$
 (10)

From the above formula, we can calculate the partial derivative  $\nabla_{\mathbf{i}_0} \log p(\mathbf{y} \mid \mathbf{i}_0)$ . And bring the conclusion in Equation 8 into it to get an approximate for the derivative of  $p(\mathbf{y} \mid \mathbf{i}_t)$  with respect to  $\mathbf{i}_t$ .

$$\nabla_{\hat{\mathbf{i}}_{t}} \log p\left(\mathbf{y} \mid \hat{\mathbf{i}}_{t}\right) \simeq -\frac{1}{\tilde{\sigma}(t)^{2}} \nabla_{\hat{\mathbf{i}}_{t}} \left\| \mathbf{y} - \hat{\mathbf{y}} \left( \hat{\mathbf{i}}_{0} \left( \hat{\mathbf{i}}_{t} \right) \right) \right\|_{2}^{2}$$
(11)

Hence, we can discretize Equation 8 similarly to formula 5 and deduce the expression for  $p(\mathbf{i}_{t-1}) \mid \mathbf{i}_t$ ). This allows us to perform iterative denoising of  $\mathbf{i}$ given the posterior y, ultimately leading to an accurate estimation of i. For a more thorough explanation of this procedure, please refer to Algorithm 2.

## Algorithm 1: Sampling of the coarse stage

Require: ground truth source proximity degree  $\mathbf{x}$ , number of diffusion steps  $T_1$ , variance schedule  $\beta_t$ , the cascade graph  $\mathcal{G}_c$ 

1:  $\mathbf{x}_{T_1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 

2: **for**  $t = T_1 - 1$  **to** t = 0 **do** 

 $\hat{\mathbf{s}} \leftarrow \mathbf{s}_{\phi}(\mathbf{x}_t, t \mid \mathcal{G}^w, \mathcal{E}^w)$ 

 $\mathbf{x}'_{t-1} \leftarrow (2 - \sqrt{1 - \beta_t} \mathbf{x}_t) + \beta_t (\hat{\mathbf{s}})$ 

 $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 

 $\mathbf{x}_t \leftarrow \mathbf{x}'_{t-1} + \sqrt{\beta_t}\mathbf{z}$ 

7: end for

8: **return** the estimated  $\mathbf{x}_0$ 

### Algorithm 2: Sampling of the fine stage

Require: disseminated observation y, coarse source proximity degree  $\hat{\mathbf{x}}$ , number of diffusion steps  $T_2$ , variance schedule  $\beta_t$ , the whole graph  $\mathcal{G}_W$ 

1:  $\mathbf{i}_{T_2} \leftarrow \hat{\mathbf{x}}$ 

2: **for**  $t = T_2 - 1$  **to** t = 0 **do** 

3:  $\hat{\mathbf{s}} \leftarrow \mathbf{s}_{\phi}(\mathbf{i}_{t}, t \mid \mathcal{G}^{w})$ 4:  $\mathbf{i}'_{t-1} \leftarrow (2 - \sqrt{1 - \beta_{t}}\mathbf{i}_{t}) +$  $\beta_t \left( \hat{\mathbf{s}} - \frac{1}{\tilde{\sigma}(t)^2} \nabla_{\mathbf{i}_t} \left\| \mathbf{y} - \hat{\mathbf{y}} \left( \hat{\mathbf{i}}_0 \left( \mathbf{i}_t \right) \right) \right\|_2^2 \right)$ 

 $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 

 $\mathbf{i}_t \leftarrow \mathbf{i}'_{t-1} + \sqrt{\beta_t}\mathbf{z}$ 

7: end for

8: **return** the estimated  $i_0$ 

Datasets	Methods	Netsleuth	OJC	LPSI	GCNSI	IVGD	SL-VAE	SL-Diff
Digg	RE	0.0142	0.0781	0.2352	0.0135	0.2310	0.5420	0.7813
	$_{\mathrm{PR}}$	0.0023	0.0554	0.0072	0.2369	0.1397	0.4216	0.5839
	F1	0.0040	0.0648	0.0140	0.0255	0.1741	0.4743	0.6683
	ACC	0.7714	0.9035	0.9531	0.8064	0.9327	0.9742	0.9824
Memetracker	RE	0.0647	0.0256	0.3047	0.2953	0.5954	0.5010	0.6902
	PR	0.0247	0.0360	0.1145	0.0172	0.1556	0.4592	0.4721
	F1	0.0358	0.0299	0.1665	0.0325	0.2467	0.4792	0.5607
	ACC	0.5688	0.6675	0.9174	0.8428	0.8947	0.9420	0.9562
Android	RE	0.3172	0.1401	0.3407	0.7434	0.7253	0.6261	0.8260
	PR	0.0422	0.0610	0.2323	0.3024	0.4105	0.5284	0.5945
	F1	0.0745	0.0850	0.2762	0.4299	0.5243	0.5731	0.6914
	ACC	0.6215	0.8337	0.9404	0.8211	0.9530	0.9245	0.9937
Christianity	RE	0.2491	0.3478	0.5309	0.7294	0.6433	0.8011	0.8352
	PR	0.1184	0.2823	0.6249	0.2300	0.5202	0.4894	0.5120
	F1	0.1605	0.3116	0.5741	0.3497	0.5752	0.6076	0.6348
	ACC	0.7140	0.9304	0.9122	0.9673	0.9781	0.9529	0.9818
Twitter	RE	0.0184	0.0154	0.2091	0.3770	0.6219	0.3273	0.9037
	PR	0.0021	0.0238	0.1295	0.3719	0.4427	0.4210	0.7839
	F1	0.0038	0.0187	0.1599	0.3744	0.5172	0.3683	0.8395
	ACC	0.6348	0.8358	0.9149	0.9231	0.9381	0.9027	0.9630

Table 1: Model performance across five datasets.

### 5 Experiments

### 5.1 Settings

**Data.** To better demonstrate the practical value of the proposed model, we conducted sufficient experiments on five datasets of various scales: Digg<sup>4</sup>, Memetracker<sup>5</sup>, Android<sup>6</sup>, Christianity<sup>6</sup>, Twitter<sup>7</sup>. Each dataset includes real information cascade data, and the specific dataset details are shown in the Appendix. In order to unify the training standard with the previous method, we define the nodes at the first 5% of the infection time in a cascade as dissemination sources, and all the nodes in the cascade as disseminated observations. We set the ratio of training, validation, and testing to 2:2:6.

Baselines & Metrics. We adopt the following two types of baselines to make a more comprehensive comparison (please refer to the Appendix for details). (1) Methods of presupposing dissemination mode: NetSleuth [30], OJC [10]. (2) Methods compatible with multiple modes of dissemination: LPSI [5], GCNSI [6], IVGD [7], SL-VAE [9].

Four evaluation matrices are used in the experiments to expose model performance more objectively and comprehensively. First, we use the most commonly

<sup>&</sup>lt;sup>4</sup> https://archive.org/details/stackexchange

<sup>&</sup>lt;sup>5</sup> http://snap.stanford.edu/memetracker

<sup>&</sup>lt;sup>6</sup> https://github.com/aravindsankar28/InfVAE/tree/master/data

<sup>&</sup>lt;sup>7</sup> https://github.com/albertyang33/FOREST/tree/master/data

Datasets	Digg		Memetracker		Android		Christianity		Twitter	
	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC
SL-Diff(1)	0.4224	0.8510	0.4001	0.7410	0.5188	0.7309	0.4511	0.7792	0.6530	0.8630
SL-Diff(2)		—				_	0.6024	0.9530		_
SL-Diff(3)	0.6202	0.9627	0.4346	0.9023	0.6427	0.9705	0.6210	0.9722	0.7401	0.9328
SL-Diff(4)	0.5985	0.9329	0.5792	0.9527	0.6132	0.9595	0.6285	0.9717	0.7620	0.9441
SL-Diff	0.6683	0.9824	0.5607	0.9565	0.6914	0.9937	0.6348	0.9818	0.8395	0.9630

Table 2: Ablation study (— represents GPU memory overflow).

used accuracy (ACC), the proportion of correctly classified samples to the total sample. However, since the source localization is essentially an unbalanced classification problem, we added a more appropriate metric, F1-Score (F1) <sup>8</sup> to reconcile the average of precision (PR) and recall (RE). We also attached the results of PR and RE for reference.

Implementation Details. For our SL-Diff model, the details of its implementation are as follows. For the diffusion steps of the coarse stage and the fine stage, we set them to  $T_1 = 800$  and  $T_2 = 80$ , respectively. For the specific dissemination model  $f_{\mathbf{w_2}}(g_{\mathbf{w_1}}(\mathbf{i}))$  we chose DeepIS [2]. For the score function approximating network (Equation 7), four layers of MLP are applied. For other benchmarks, we follow the original structural design. Specifically, for the dissemination model that SL-VAE combines, we adopt DeepIS, while the IC function is used for IVGD.

The models are trained on a single NVIDIA GeForce RTX 3090 GPU. We use a grid search to find the most appropriate combination of parameters for each model. Specifically, the search range for the number of GCN stacks is from 2 to 8. The learning rate is tuned within  $\{5 \times 10^{-2}, 10^{-2}, 5 \times 10^{-3}, 10^{-3}\}$ . The range of Riemannian SGD weight decay is  $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ .

#### 5.2 Overall Performance

We have conducted in-depth comparisons of SL-Diff with state-of-the-art baselines on five real datasets, and the results are presented in Table 1. Generally, models with preset dissemination modes perform relatively poorly, as real information dissemination modes are complex and full of uncertainties. SL-Diff, on the other hand, achieves significantly optimal results in most cases. Among them, the improvement on the Twitter and Android datasets is particularly significant which can be attributed to its ability to denoise the dissemination noise. And this noise is more prominent on datasets with a large cascade length and a large number of nodes. Correspondingly, on the Christianity dataset with a small number of nodes, the performance of SL-Diff is limited. In summary, SL-Diff is more capable of handling large-scale datasets.

 $<sup>8</sup> F1 = \frac{2PR*RE}{PR+RE}$ 

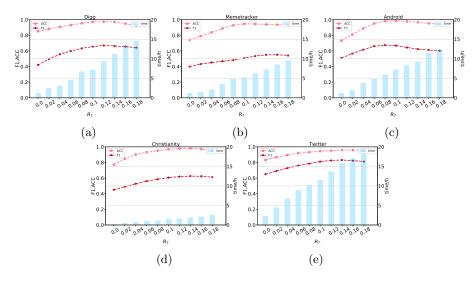


Fig. 8: Analysis on the ratio of the diffusion step of the fine stage to the coarse stage.

#### 5.3 Ablation Study

To verify the effectiveness of each component of our proposed model, we conduct the following ablation experiments. Compared with the complete model, SL-Diff(1) only retains the coarse stage and is initialized randomly. SL-Diff(2) only retains the fine stage and we take the top 5% of nodes with the largest source proximity degree generated as the source of prediction. SL-Diff(3) is the model with the cascade positional representations removed, and SL-Diff(4) removes the simulated dissemination noise term in Equation 10. The diffusion step of the above four models is all set to 800. From the results in Table 2, we can conclude that only the coarse stage model can make rough predictions to a certain extent, and it is difficult to make accurate predictions. The model with only the fine stage is not necessarily more accurate than the two-stage model, and the computational overhead is huge, which is not feasible. From the results of SL-Diff(3), we can see that cascade positional representations generally have a more important influence on larger-scale graphs, which may be due to the fact that the relative position of cascades in large-scale graphs will be more complicated, and their influence on propagation will be larger. From the results of SL-Diff(4), we can find that simulated dissemination noise improves the model with a longer cascade length more significantly, which may be because a cascade with a longer propagation chain will introduce relatively more noise.

### 5.4 Parameter Analysis

In this section, we explore the interdependence of the two stages in greater detail. Specifically, we set the ratio of the diffusion step of the fine stage to the coarse

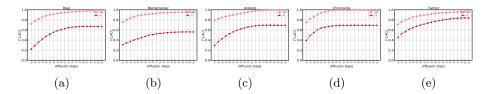


Fig. 9: Convergence analysis.

stage as  $R_T = \frac{T_2}{T_1}$  and keep the total diffusion step at 800. By adjusting the value of  $R_T$ , we gain a deeper understanding of how the two stages mutually reinforce each other, and determine the optimal ratio. As shown in Figure 8, we observe that the model's performance initially improves significantly with an increasing proportion of the fine stage, but then gradually declines, while the sampling time steadily increases. This suggests that we can identify the most suitable ratio of the two stages at a relatively low time cost across different datasets. The performance degradation resulting from too high  $R_T$  may be due to insufficient denoising in the coarse stage. Additionally, we analyze the impact of the diffusion step on the model's performance under the configuration with the optimal  $R_T$ . Figure 9 reveals that the convergence of the sampling process is comparatively faster for datasets with smaller scales.

Efficient Analysis. We compared the training time of various models (sampling time is also included since SL-Diff and SL-VAE are generative models). As shown in Figure 10, the generative model exhibits a shorter overall running time for the source localization task. Among all the models evaluated, SL-Diff effectively controls the computational cost at a lower level, emphasizing the significance of the coarse stage for initialization.

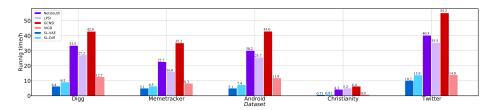


Fig. 10: Model efficiency comparison.

### 6 Conclusion

In this paper, we propose a new two-stage training paradigm to solve the ill-posed problem in graph source localization from the root. SL-Diff also overcomes

the difficulties of introducing the powerful diffusion model into this problem, and achieves optimal results on various real-world datasets. The modeling of dissemination noise further improves the approximation performance of SL-Diff. Overall, by leveraging the diffusion model, we have gained deeper insights into the mechanism of the source localization problem, thereby elevating research in this field to a new level.

### References

- Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. Inf-vae: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *Proceedings of the 13th international conference on web search and data mining*, pages 510–518, 2020.
- Wenwen Xia, Yuchen Li, Jun Wu, and Shenghong Li. Deepis: Susceptibility estimation on social networks. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pages 761–769, 2021.
- 3. Shanshan Feng, Kaiqi Zhao, Lanting Fang, Kaiyu Feng, Wei Wei, Xutao Li, and Ling Shao. H-diffu: Hyperbolic representations for information diffusion prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- 4. Ruijie Wang, Zijie Huang, Shengzhong Liu, Huajie Shao, Dongxin Liu, Jinyang Li, Tianshi Wang, Dachun Sun, Shuochao Yao, and Tarek Abdelzaher. Dydiff-vae: A dynamic variational framework for information diffusion prediction. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 163–172, 2021.
- 5. Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. Multiple source detection without knowing the underlying propagation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 569–578, 2019.
- Junxiang Wang, Junji Jiang, and Liang Zhao. An invertible graph diffusion neural network for source localization. In *Proceedings of the ACM Web Conference 2022*, pages 1058–1069, 2022.
- 8. Zhen Wang, Dongpeng Hou, Chao Gao, Jiajin Huang, and Qi Xuan. A rapid source localization method in the early stage of large-scale network propagation. In *Proceedings of the ACM Web Conference 2022*, pages 1372–1380, 2022.
- Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. 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, pages 1010–1020, 2022.
- 10. Kai Zhu, Zhen Chen, and Lei Ying. Catch'em all: Locating multiple diffusion sources in networks with partial observations. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- 11. Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

- Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. arXiv preprint arXiv:2209.14687, 2022.
- 14. Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020.
- 15. Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2021.
- 16. David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.
- 17. Matt J Keeling and Ken TD Eames. Networks and epidemic models. *Journal of the royal society interface*, 2(4):295–307, 2005.
- 18. Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. A peek into the future: Predicting the evolution of popularity in user generated content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 607–616, 2013.
- Yongqing Wang, Huawei Shen, Shenghua Liu, Jinhua Gao, and Xueqi Cheng. Cascade dynamics modeling with attention-based recurrent neural network. In *IJCAI*, volume 17, pages 2985–2991, 2017.
- Jiayi Xie, Yaochen Zhu, Zhibin Zhang, Jian Peng, Jing Yi, Yaosi Hu, Hongyi Liu, and Zhenzhong Chen. A multimodal variational encoder-decoder framework for micro-video popularity prediction. In *Proceedings of The Web Conference 2020*, pages 2542–2548, 2020.
- Kai Zhu and Lei Ying. Information source detection in the sir model: A sample-path-based approach. IEEE/ACM Transactions on Networking, 24(1):408–421, 2014.
- 22. Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Int. Conf. on Learning Representations*, 2013.
- Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. arXiv preprint arXiv:2202.02514, 2022.
- Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.
- 25. Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes* and their Applications, 12(3):313–326, 1982.
- Ling Sun, Yuan Rao, Xiangbo Zhang, Yuqian Lan, and Shuanghe Yu. Ms-hgat: Memory-enhanced sequential hypergraph attention network for information diffusion prediction. 2022.
- Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks.
   In International conference on machine learning, pages 7134–7143. PMLR, 2019.
- 28. Jinheon Baek, Minki Kang, and Sung Ju Hwang. Accurate learning of graph representations with graph multiset pooling. arXiv preprint arXiv:2102.11533, 2021.
- 29. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

30. B Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. Spotting culprits in epidemics: How many and which ones? In 2012 IEEE 12th International Conference on Data Mining, pages 11–20. IEEE, 2012.