

# Denoising Propagation Uncertainty for Information Source Localization

Leyuan Liu\*, Zhangtao Cheng\*, Xovee Xu\*, Fan Zhou\*<sup>†‡</sup>

\*University of Electronic Science and Technology of China, Chengdu, China

<sup>†</sup>Kash Institute of Electronics and Information Industry, Kash, China

<sup>‡</sup>Corresponding author: fan.zhou@uestc.edu.cn

**Abstract**—Source localization, as a reverse problem of information dissemination on graphs, is crucial for tracking social rumors, detecting computer viruses, and identifying epidemic spreaders. However, existing methods face challenges due to the inherent uncertainty of graph diffusion, as the same diffused observations may start with diverse sources. Furthermore, general graph diffusion models did not consider important properties of the diffusion process. To address these issues, we propose a denoising diffusion probabilistic model (DDPM)-based source localization framework, DDSL. In this framework, we consider two distinct characteristics of information dissemination, namely source prominence and monotone increasing, and present a source localization-oriented invertible graph neural network (GNN). To capture the propagation uncertainties of sources, we design a DDPM-based source generator to generate effective and diverse sources for enhancing model’s robustness. Our experiments demonstrate the effectiveness of the proposed model in improving source localization performance.

**Index Terms**—Graph diffusion, source localization, information dissemination

## I. INTRODUCTION

Social networking services (e.g. Twitter, Weibo, and Facebook) provide convenient and rapid ways for information dissemination, facilitating people’s lives and work. However, they also lead to misinformation and malware spreading quickly and widely on the Internet. To address these challenges, source localization [1] plays a crucial role by identifying the source(s) of the observed dissemination processes. This approach helps to trace and control the spread of misinformation and malware, mitigating their negative impact across various domains, including detecting misinformation and rumor in social networks [2], controlling epidemic in infectious diseases and isolating failures smart grids. Correspondingly, source localization has attracted significant interest from researchers and engineers and various localization methods have been proposed.

**Related work.** 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 [1]. One group of the existing works [3], [4], [5] in the field of diffusion source localization utilize centrality measures within tree-like networks to identify potential propagation sources. For example, in [3] the authors model rumor spreading in a network with the susceptible-infected (SI) model and then using a maximum likelihood estimator to calculate the rumor

centrality scores according to the topology of diffusion graph. Another group of works leverages network sensor injection for source tracking. These methods offer a unique and dynamic approach to source localization by strategically placing sensors to track and identify sources of diffusion step by step. This approach is exemplified in works such as [6], [7], [8], where sensor-based tracking is used for various propagation models, e.g., SI or susceptible-infected-recovered (SIR).

While these methods are effective, they suffer from high computational costs and complexity. To address these challenges, graph neural networks (GNNs) have been proposed as a promising solution for source localization. GNNs excel in learning graph diffusion patterns and have shown strong performance in tasks like node classification and link prediction [9], [10]. They effectively capture complex relationships within diffusion networks, making them suitable for source localization problems [11]. Huang et al. [12] introduced a two-stage approach using Position-aware GNNs for positional embeddings and a denoising diffusion model to handle the graph inverse problem. Other GNN-based models, such as GCNSI [13] and IGCN [14], utilize graph convolutional networks and attention mechanisms to enhance performance and reduce computational complexity in detecting multiple rumor sources.

In addition, source localization can be formulated as an inverse problem of graph diffusion. This has led to the development of invertible graph diffusion models that reverse inputs and outputs for source localization tasks [15], [16]. However, these models often overlook key diffusion characteristics, such as *source prominence*—the idea that nodes surrounded by a higher proportion of infected nodes are more likely to be sources [17]—and *monotone increasing*, where each node should contribute non-negatively to the diffusion process [18]. IVGD [16] introduces a graph residual network with Lipschitz regularization to infer sources in generic diffusion models, but it does not account for the uncertainty associated with diffusion sources.

**Challenges.** While existing GNN methods achieved promising results on graph source localization, they still face several notable challenges. First, traditional graph algorithms to locate diffusion sources require exhaustive searches through the topology space, such as traversing all possible paths for a specific source node. This is computationally expensive for large-scale networks and overlooks key aspects like temporal

dynamics, node relationships, and variability in diffusion patterns [19]. Second, most learning-based methods use deterministic learning, which fails to capture the diffusion uncertainty associated with the sources. This is due to the inherent variability in diffusion processes, where different sources may lead to similar patterns over time. Such uncertainties arise from network structures and the stochastic nature of diffusion, influenced by random or external factors [20].

To address these issues, we propose a denoising diffusion probabilistic model (DDPM)-based source localization framework, DDSL. In this framework, to enhance the efficiency of source inference, we design a source localization-oriented invertible GNN to invert the graph diffusion process. Compared with typical invertible networks [15], [16], we incorporate two important properties – the *source prominence* [13] and *monotone increasing* [18] – into the invertible GNN for enhancing model’s learning capability on source localization. Here, source prominence refers to the sources tend to be surrounded by infected nodes [17], while monotone increasing indicates that infected nodes will monotonously increase during diffusion [18]. These properties allow the GNN to effectively capture key characteristics of the diffusion process and improve the robustness of source localization.

To further address the challenge of diffusion uncertainty, we integrate a DDPM-based source generator that creates diverse source vectors maintaining the same observations as the input. By considering input source vectors, centrality [19], and observations as conditions in the generative process [21], our generator offers a more robust approach than traditional probabilistic models like VAEs [11], [22]. The generated vectors help the invertible GNN better handle diffusion uncertainty. The invertible GNN and DDPM-based source generator are optimized interactively, where each component refines the quality of the generated vectors and extracted features, leading to improved localization performance.

The main contributions of our paper are summarized as follows:

- We propose a DDPM-based source localization framework, DDSL, which effectively generates diverse source vectors to help GNNs capture diffusion uncertainty.
- We present a source localization-oriented invertible GNN, incorporating distinct diffusion characteristics—source prominence and monotone increasing—into the network to improve source inference.
- We validate our model on four real-world datasets with varying propagation patterns, demonstrating its effectiveness and analyzing its strengths and limitations.

## II. METHODOLOGY

In this section, we first formally define the task of graph source localization and then elaborate on the proposed DDSL.

### A. problem statement

**Graph Source Localization:** Consider an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  is the node set and  $\mathcal{E}$  is the edge set. Let  $Y = \{y_0, \dots, y_i, \dots\} \in \mathbb{R}^{|\mathcal{V}|}$  be the infection state vector,

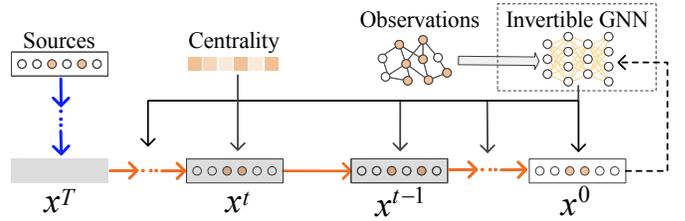


Fig. 1. Overview of the framework.

$y_i = 1$  if node  $i$  is infected; otherwise  $y_i = 0$ . Let  $X = \{x_0, \dots, x_i\} \in \mathbb{R}^{|\mathcal{V}|}$  be the source vector,  $x_i = 1$  if node  $i$  is source node; otherwise  $x_i = 0$ . The source localization is to identify source nodes  $X$  given  $Y$  in  $\mathcal{G}$ .

### B. Architecture Overview

An overview of our model, DDSL, is depicted in Fig. 1. It consists of two main components: a source localization-oriented invertible GNN and a DDPM-based source generator.

The model leverages two key properties of graph diffusion: the prominence of the source and the monotonous increase of dissemination. This is used to construct an invertible GNN designed for source localization, efficiently inferring the source vector from an input observation.

The DDPM-based source generator takes a real source vector as input and generates diverse source vectors with the same observations as the input source vector. To ensure observations of generated source vectors invariant, we impose the real source vector, centrality, and observation information on the generative process.

The generated source vectors are then used to train the invertible GNN. Since they all have the same observations with the original input source(s), the trained invertible GNN is expected to have the ability to capture diffusion uncertainties and further improve the model performance.

The DDPM-based generator and invertible GNN are optimized interactively. In each loop, the generator first adopts the initialized invertible GNN to extract observation information for generating source vectors, which are then used to train the invertible GNN. As the loop continues, the weights of both extracted observation information and generated source vectors improve in quality.

The inverse GNN and DDPM-based source generator are optimized together in a two-step process. First, the invertible GNN is trained on real source vectors and observations. The generator then uses this trained GNN to extract observation features, which serve as conditions for its generative process. Second, the generated source vectors, together with the real ones, train the GNN. This process is repeated in multiple loops, with increasing weights assigned to both extracted features and generated sources, improving their quality. Consequently, the GNN can capture uncertainties more effectively, enhancing overall performance.

### C. Invertible GNN

To address the limitations of existing methods that overlook key diffusion characteristics like source prominence and monotone increasing, we propose an enhanced invertible GNN model. This model integrates these features to better capture the inverse diffusion processes for source localization. Next, we detail the invertible network architecture and its design components.

1) *Invertible network architecture*: In particular, to consider these distinct characteristics, we divide the invertible GNN  $H$  into two parts, feature construction  $F_{\psi_1}(\cdot)$  and propagation  $G_{\psi_2}(\cdot)$ . We define the residual blocks for the GNN as  $(f(x) + x)/2$  and integrate the residual block into these structures. Now we provide the mathematical guarantees on the invertibility of the invertible GNN:

**Theorem 1.** *Let  $H = F_{\psi_1} \circ G_{\psi_2} : \mathbb{R}^{|\mathcal{V}|} \rightarrow \mathbb{R}^{|\mathcal{V}|}$  denotes a GNN with residual blocks  $(f(x) + x)/2$  and  $(g(x) + x)/2$  in  $F_{\psi_1}$  and  $G_{\psi_2}$ , respectively. Then, the GNN  $H$  is invertible if  $Lip(f) < 1$  and  $Lip(g) < 1$ , where  $Lip(\cdot)$  is the Lipschitz-constants.*

*Proof.* All the residual blocks we used are implemented by a composition of non-linear activation  $\sigma$  and linear mappings  $Wx + b$ , where  $\sigma$  denotes the non-linear activate function and  $W$  denotes the weight matrix of linear mappings. Hence, we have  $Lip(f) < 1$  and  $Lip(g) < 1$  if  $\|W_i\|_2 < 1$ , where  $\|W_i\|_2$  is the spectral norm of the linear matrix  $W_i$  in  $f$  or  $g$ . We need to conduct spectral norm to each linear layer  $W_i$ , and then multiply a hyper-parameter  $c \in (0, 1)$  to them [23]. Therefore, GNN  $H$  with the normalized linear layers  $\tilde{W}_i$  is invertible.  $\square$

The forward process (i.e. information dissemination) of the invertible GNN can be described as:

$$X_{\text{attr}} = F_{\psi_1}(X) = (f(X) + X)/2, \quad (1)$$

$$\hat{Y} = G_{\psi_2}(X_{\text{attr}}) = (g(X_{\text{attr}}) + X_{\text{attr}})/2, \quad (2)$$

where  $X$  denotes input the source vector,  $X_{\text{attr}}$  is the constructed feature, and  $\psi_1$  and  $\psi_2$  are the parameters of feature construction network  $F$  and propagation network  $G$ , respectively. Based on this process, we calculate the forward loss of information dissemination with Mean Square Error:

$$\mathcal{L}_D = \|Y - \hat{Y}\|_2^2, \quad (3)$$

where  $Y$  is the real observation of input source vector  $X$ .

2) *Source prominence*: Directly mapping sources to diffused observations ignores the source prominence rules of the graph topology. This can result in an insufficient expression of the structural information of the dissemination graph. Thus, we incorporate the source prominence rule into the training process. The feature of source prominence can be captured by LPSI method [13], and the iteration equation of label propagation in LPSI can be written as:

$$\mathcal{G}^{m+1} = \alpha S \mathcal{G}^m + (1 - \alpha)Y, \quad (4)$$

where  $\mathcal{G}^m$  denotes the infection state at iteration  $m$ ,  $S$  is the regularized Laplace matrix of the graph  $\mathcal{G}$ , and  $\alpha$  is the parameter to control the influence factors between nodes and their neighbors. The feature of source prominence is constructed by the convergent state of the iteration above.

We add this information into  $X_{\text{attr}}$  in the invertible GNN. The reverse process with the prominence feature on  $X_{\text{attr}}$  according to Eq. (4) is:

$$\hat{Y}^* = (1 - \alpha)X_{\text{attr}}(I - \alpha S), \quad (5)$$

where  $\hat{Y}^*$  is derived from  $X_{\text{attr}}$  with the feature of source prominence, which should be close to  $Y^*$  in order to make  $F_{\psi_1}(\cdot)$  in the invertible GNN learn the features of graph topology. Here  $Y^*$  is derived from  $Y$  where the uninfected nodes are set as  $-1$ . Meanwhile, the constraint of source prominence is also considered according to Eq. (5). The loss function is defined as:

$$\mathcal{L}_P = \|Y^* - \hat{Y}^*\|_2^2. \quad (6)$$

3) *Dissemination monotone increment*: In addition, we impose the monotone increasing constraint [18] on the dissemination. That is to say if one source set  $X^i$  is the subset of another source set  $X^j$ , the probability of each node being infected  $p_{\text{inf}}(X^i)$  from  $X^i$  should be smaller than source set  $X^j$ . This constraint is described as:

$$\forall X^i \subseteq X^j, \quad \text{s.t. } p_{\text{inf}}(X^i) \leq p_{\text{inf}}(X^j). \quad (7)$$

As it is difficult to model the inequality constraints directly, we convert the constraint into a Lagrangian form as follows:

$$\mathcal{L}_M = \lambda \|\max\{0, p_{\text{inf}}(X^i) - p_{\text{inf}}(X^j)\}\|_2, \forall X^i \subseteq X^j, \quad (8)$$

where  $\lambda$  is the regularization hyper-parameter,  $p_{\text{inf}}(X^i)$  and  $p_{\text{inf}}(X^j)$  are estimated from an  $X^i$  and many  $X^j$  sampled in each batch by the invertible GNN, respectively.

4) *Source inference*: At last, we combine the objective functions in the training phase together for training the invertible GNN:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_D + \mathcal{L}_P + \mathcal{L}_M. \quad (9)$$

We directly feed the diffused observation  $Y$  into the trained invertible GNN to infer the sources  $\hat{X}$  through fixed point iterations:

$$z^0 = Y, \dots, z^n = 2Y - g(z^{n-1}) \quad (10)$$

$$X^0 = z^n, \dots, \hat{X}^n = 2z^n - f(\hat{X}^{n-1}), \quad (11)$$

where  $n$  is the number of iterations,  $z^0$  and  $X^0$  denote the initial states,  $f$  and  $g$  denote the linear mappings in  $F_{\psi_1}$  and  $G_{\psi_2}$ , respectively, and  $\hat{X}^n$  is the output of the inverse process.

### D. DDPM-based Source Generator

To further capture the uncertainty in graph diffusion processes, we employ DDPM as the basis for our source generator. This generator produces diverse source vectors that share the same observed characteristics as the input source,

enhancing the invertible GNN’s ability to manage diffusion uncertainty.

The generative process of DDPM must satisfy several conditions to ensure consistency between the generated sources and the original observations: (1) Generated sources should closely resemble the original real sources. (2) They should adhere to network centrality regulations, where potential propagation sources are more likely to be centrally located in diffusion graphs [19]. (3) The generated sources should replicate the given observations.

To achieve this, these conditions are integrated into the reverse diffusion process of DDPM to generate source vectors that meet these criteria.

As shown in Fig. 1, the generator adopts the forward process to add noises into the real source vector  $x^{\text{real}}$  to form a prior  $x^T$ , which is fed into the reverse diffusion process to generate a new source vector via gradual denoising, i.e.,  $x^T \rightarrow x^t \rightarrow x^{t-1} \rightarrow x^0$ . The denoising process iteratively imposes centrality information and observation features on the latent variables. This forces the generated source vectors to have the same diffusion observations as the original source vectors.

1) *Source vector condition:* We first impose the real source vector  $x^{\text{real}}$  on the reverse generative process. Noises are added into  $x^{\text{read}}$  to form prior  $x^T$  using the forward process:

$$x^T = \sqrt{\bar{\alpha}_T} x^{\text{real}} + \sqrt{1 - \bar{\alpha}_T} \epsilon, \quad (12)$$

where  $\bar{\alpha}_T = \prod_{t=1}^T (1 - \beta_t)$ , and  $\beta_t$  is a fixed variance schedule [21]. According to the Markov chain, the conditional reverse diffusion process aims to predict  $x^{t-1}$  based on  $x^t$  and  $x^{\text{real}}$ . After adding the condition  $x^{\text{real}}$ , the reverse process of the generator becomes:

$$p_\theta(\hat{x}^{t-1} | x^t, x^{\text{real}}) = \mathcal{N}(\hat{x}^{t-1}; \mu_\theta(x^t, x^{\text{real}}, t), \tilde{\beta}_t \mathbf{I}), \quad (13)$$

where  $\mu_\theta(\hat{x}^t, x^{\text{real}}, t)$  is the estimated mean of the conditional reverse process, and  $\tilde{\beta}_t$  is a fixed constant. The reverse process starts with Gaussian noise  $x^T$ , and generates a clean sample  $x^0$  by sampling reverse steps  $p_\theta(\hat{x}^{t-1} | x^t, x^{\text{real}})$ .

To parameterize  $\mu_\theta(x^t, x^{\text{real}}, t)$ , we train a neural denoising model  $f_\theta(x^t, x^{\text{real}}, t)$  to predict the noise vector  $\epsilon$ . The objective  $\mathcal{L}_D$  is:

$$\mathbb{E}_{x^{\text{ob}}} \mathbb{E}_{(\epsilon, t)} \left[ \|f_\theta(x^t, x^{\text{real}}, t) - \epsilon\|_2^2 \right], \quad (14)$$

where  $\mu_\theta(x^t, x^{\text{real}}, t)$  can be derived from  $f_\theta(x^t, x^{\text{real}}, t)$ :

$$\mu_\theta(x^t, x^{\text{real}}, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( x^t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} f_\theta(x^t, x^{\text{real}}, t) \right). \quad (15)$$

Consequently, the generative process is defined as:

$$\hat{x}^{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( x^t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} f_\theta(x^t, x^{\text{real}}, t) \right) + \sqrt{\tilde{\beta}_t} \mathbf{z}, \quad (16)$$

where  $\mathbf{z} \sim N(0, \mathbf{I})$ , implying that each generation step is stochastic.

2) *Centrality and observation conditions:* Besides the constraint of approaching real source vector, we further consider the centrality and observation information as two new conditions and impose them into the generative process. Towards this purpose, we formulate the centrality scores and observation features.

Centrality scores describe the node influence on propagation, widely adopted to identify potential propagation sources [19], [24], [25]. We combine eccentricity and closeness [24], [19] to calculate centrality scores  $C$ . Since source nodes tend to be located in the center regions, we impose this condition on the generative process, i.e., the node with a higher centrality score has a greater probability of being the source node. The observation features  $O$  extracted from the invertible GNN also describe the probability of being the source node. To combine  $C$  and  $O$  together, the generative process of producing  $x^{t-1}$  is defined as:

$$\max \left\{ (1 + C) \hat{x}^{t-1} - C^{\text{avg}}, 0 \right\} + \max \left\{ (1 + h_1(l)O) \hat{x}^{t-1} - O^{\text{avg}}, 0 \right\}, \quad (17)$$

where  $C^{\text{avg}}$  and  $O^{\text{avg}}$  denote the averages of  $C$  and  $O$ , respectively,  $h_1(l)$  is a monotone increasing weight function, and  $l$  is the loop times. The reverse generative process with these conditions are used to manufacture a new source vector that has the same observation as the input source but also expresses diffusion uncertainties.

### E. Joint Optimization

After obtaining the generated source vectors  $X_G$ , we fed them into the forward process in Eq. (2) to compute observation  $\hat{Y}_G$ . The forward loss of generated sources is defined as:

$$\mathcal{L}_G = \|Y - \hat{Y}_G\|_2^2. \quad (18)$$

Coupled with the combined losses in Eq. (9), the final objective is:

$$\mathcal{L}_{\text{train}}^G = \mathcal{L}_D + \mathcal{L}_P + \mathcal{L}_M + h_2(l) \mathcal{L}_G, \quad (19)$$

where  $h_2(l)$  is a monotone increasing function similar to  $h_1(l)$ . The invertible GNN is first trained based on the real source vectors and then in each loop of the generation it extracts observation features as the condition exerted on the generative process for generating new source vectors. We note in the initial loops, both  $h_1(l)$  and  $h_2(l)$  have smaller values since features and vectors are assigned lower weights. Along with the loops progresses, larger weights are assigned to them and finally, we can obtain high-quality source vectors for improving the inference performance.

## III. EXPERIMENTAL RESULTS

### A. Experimental Settings

**Dataset.** To ensure consistency and comparability with baseline methods, we selected four commonly used real-world diffusion datasets. These datasets provide diverse network

TABLE I  
DATASET STATISTICS

Name	# Nodes	# Edges	Avg Degree
Jazz	198	2,742	27.7
Network Science	1,589	13,532	17.29
Cora-ML	2,810	7,981	5.68
Power Grid	4,941	6,549	2.67

structures and diffusion scenarios for evaluating the performance of our model. The statistics of the datasets are shown in TABLE I.

- **Jazz** [26]: A collaboration network of Jazz bands.
- **Network Science (NS)** [27]: A co-authorship network of scientists working on network theory.
- **Cora-ML (CML)** [28]: A citation network contains computer science research papers.
- **Power Grid (PG)** [29]: A topology network of the Western States Power Grid in the US.

Since we only have graph topology information, following previous works [11], [16], we randomly select 10% of the nodes as sources and simulate the graph diffusion based on the SI and SIR propagation protocols with enough iterations. The choice of 10% is commonly adopted in related works as it offers a balance between sparse and dense diffusion scenarios, providing sufficient challenge for source localization while keeping the complexity manageable. This setting also allows for consistent and comparable experimental conditions across different datasets, ensuring that the results are not biased by varying dataset characteristics. The ratio of training and test sets is 9:1.

**Metrics.** We use four metrics frequently used in classification tasks to evaluate the model performance: Precision (PR), Recall (RE), F1-score (F1), and ROC-AUC (AUC). These metrics serve as the evaluation protocol.

**Baselines.** We use three strong source localization baselines:

- **LPSI** [17]: a label propagation-based source identification model, which predicts the rumor sources without knowing the underlying information propagation;
- **GCNSI** [13]: a GCN-based source identification method, which adopts GCN layers to learn latent node embeddings to identify multiple rumor sources;
- **SL-VAE** [11]: a probabilistic approach which utilizes VAEs to tackle source localization.

### B. Performance Evaluation

The experimental results are shown in TABLE II and TABLE III, where we compare our model with the baselines under SI and SIR propagations, respectively. (1) For SI, our model significantly outperforms *LPSI* and *GCNSI*, and achieves comparable performance compared to *SL-VAE*. As for *CML* and *PG* which have a large number of nodes, our model outperforms *SL-VAE* by 4% on average. We speculate this is because as the network size grows, the uncertainty of the diffusion becomes larger and our model is capable to identify more accurate sources in such networks. Our model’s

performance on *Jazz* under SI is slightly lower than desired, which might be due to the high average node degree of *Jazz* compared to other datasets – nodes with many connections in the graph increase the randomness of the infection and make the uncertainty more complicated. Moreover, DDSL relies on the centrality to generate uncertainty sources, higher node degrees smooth the centrality features and consequently impact the generation process. (2) For SIR, DDSL outperforms all three baselines especially on *Jazz*, *CML* and *NS*. This indicates our model’s performance is more robust under SIR. Furthermore, all three baselines perform worse under SIR than SI. This is because the diffusion process of SIR is more complex than SI’s.

TABLE II  
PERFORMANCE OVER BASELINES UNDER SI MODEL

Methods	Jazz				Network Science			
	PR	RE	F1	AUC	PR	RE	F1	AUC
LPSI	0.105	0.478	0.171	0.484	0.423	0.604	0.497	0.837
GCNSI	0.158	0.436	0.232	0.642	0.137	0.224	0.171	0.475
SL-VAE	0.719	0.947	0.818	0.978	0.599	0.935	0.729	0.949
DDSL	0.571	0.892	0.781	0.975	<b>0.602</b>	<b>0.922</b>	<b>0.742</b>	0.934

Methods	Cora-ML				Power Grid			
	PR	RE	F1	AUC	PR	RE	F1	AUC
LPSI	0.155	0.595	0.246	0.667	0.454	0.495	0.473	0.933
GCNSI	0.118	0.361	0.178	0.538	0.141	0.347	0.209	0.504
SL-VAE	0.571	0.899	0.697	0.941	0.589	0.932	0.721	0.944
DDSL	<b>0.593</b>	<b>0.947</b>	<b>0.744</b>	<b>0.950</b>	<b>0.611</b>	0.921	<b>0.735</b>	0.933

TABLE III  
PERFORMANCE OVER BASELINES UNDER SIR MODEL

Methods	Jazz				Network Science			
	PR	RE	F1	AUC	PR	RE	F1	AUC
LPSI	0.115	0.363	0.169	0.501	0.136	0.432	0.207	0.561
GCNSI	0.141	0.373	0.205	0.641	0.104	0.351	0.161	0.543
SL-VAE	0.503	0.787	0.613	0.789	0.571	0.942	0.709	0.951
DDSL	<b>0.571</b>	<b>0.888</b>	<b>0.762</b>	<b>0.946</b>	<b>0.582</b>	0.922	<b>0.734</b>	0.927

Methods	Cora-ML				Power Grid			
	PR	RE	F1	AUC	PR	RE	F1	AUC
LPSI	0.107	0.477	0.175	0.498	0.486	0.472	0.478	0.582
GCNSI	0.115	0.338	0.172	0.532	0.113	0.237	0.153	0.503
SL-VAE	0.582	0.919	0.711	0.930	0.580	0.933	0.714	0.947
DDSL	<b>0.593</b>	<b>0.924</b>	<b>0.755</b>	0.929	<b>0.597</b>	0.917	<b>0.732</b>	0.929

### C. Ablation Study

To investigate the contributions of each module in DDSL, we design four variants: (1) *DDSL w/o cent* which removes the centrality information from our model; (2) *DDSL w/o dmi* which removes the dissemination monotone increment constraint from the invertible GNN; (3) *DDSL w/o diff* which removes generated source vector from our model and only use the invertible GNN, and (4) *DDSL w/o sp* which removes the source prominence from the invertible GNN. The performance changes on four real-world datasets under SI propagation are shown in Fig. 2. In general, the removal of any module in our model decreases the 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. Specifically, we found: (1) *DDSL w/o diff* and *DDSL w/o cent* have the worst performance among four variants, this implies that centrality reflects a node’s importance within a network, directly impacting the model’s ability to identify the source node. Meanwhile, incorporating uncertainty into the model captures the complexity and diversity of the source node’s propagation behavior, allowing for more accurate source identification. The model’s significant performance drop in the absence of these factors further demonstrates their necessity in graph source localization; (2) the performance of *DDSL w/o sp* is better than *DDSL w/o dmi*, which suggests source prominence is more beneficial than the monotone increasing, which nevertheless, still brings a slight improvement for source localization.

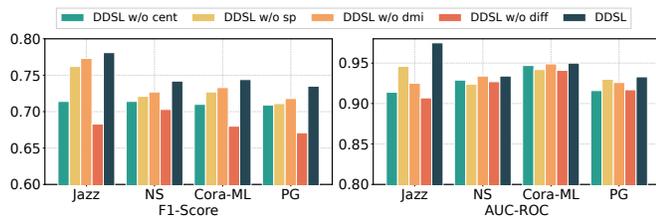


Fig. 2. Ablation Study of DDSL under SI.

#### IV. CONCLUSION

We propose DDSL, a novel DDPM-based source localization framework to address the uncertainty problem in reverse graph diffusion. We designed an invertible GNN that incorporates the source prominence and monotone increasing into graph neighborhood aggregation. Moreover, a DDPM-based source generator is presented to generate effective diverse source vectors to tackle the uncertainty problem. Experiments show the efficacy of our model compared to strong baselines. For future works, we plan to extend our method to more complex propagation models other than SI and SIR.

#### ACKNOWLEDGEMENT

Research supported in part by the Joint Funds of the National Natural Science Foundation of China (Grant No.U2336204, 62176043 and 62072077).

#### REFERENCES

- [1] L. Ying and K. Zhu, “Diffusion source localization in large networks,” *Synthesis Lectures on Communication Networks*, vol. 11, no. 1, pp. 1–95, 2018.
- [2] L. Liu, J. Chen, Z. Cheng, W. Tai, and F. Zhou, “Towards trustworthy rumor detection with interpretable graph structural learning,” in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 4089–4093.
- [3] D. Shah and T. Zaman, “Rumors in a network: Who’s the culprit?” *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5163–5181, 2011.
- [4] K. Zhu and L. Ying, “Information source detection in the sir model: A sample-path-based approach,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 408–421, 2014.
- [5] B. A. Prakash, J. Vreeken, and C. Faloutsos, “Spotting culprits in epidemics: How many and which ones?” in *ICDM*, 2012, pp. 11–20.

- [6] A. Agaskar and Y. M. Lu, “A fast monte carlo algorithm for source localization on graphs,” in *Wavelets and Sparsity XV*, vol. 8858. SPIE, 2013, pp. 429–434.
- [7] P. C. Pinto, P. Thiran, and M. Vetterli, “Locating the source of diffusion in large-scale networks,” *Physical Review Letters*, vol. 109, no. 6, p. 068702, 2012.
- [8] E. Seo, P. Mohapatra, and T. Abdelzaher, “Identifying rumors and their sources in social networks,” in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*, vol. 8389. SPIE, 2012, pp. 417–429.
- [9] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2017.
- [10] F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, “A survey of information cascade analysis: Models, predictions, and recent advances,” *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–36, Mar. 2021.
- [11] C. Ling, J. Liang, J. Wang, and L. Zhao, “Source localization of graph diffusion via variational autoencoders for graph inverse problems,” in *SIGKDD*, 2022, pp. 1010–1020.
- [12] B. Huang, W. Yu, R. Xie, J. Xiao, and J. Huang, “Two-stage denoising diffusion model for source localization in graph inverse problems,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2023, pp. 325–340.
- [13] M. Dong, B. Zheng, N. Q. V. Hung, H. Su, and G. Li, “Multiple rumor source detection with graph convolutional networks,” in *CIKM*, 2019, pp. 569–578.
- [14] Q. Guo, C. Zhang, H. Zhang, and L. Fu, “Igcnet: Infected graph convolutional network based source identification,” in *IEEE GLOBECOM*. IEEE, 2021, pp. 1–6.
- [15] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen, “Invertible residual networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 573–582.
- [16] J. Wang, J. Jiang, and L. Zhao, “An invertible graph diffusion neural network for source localization,” in *The Web Conference*, 2022, pp. 1058–1069.
- [17] Z. Wang, C. Wang, J. Pei, and X. Ye, “Multiple source detection without knowing the underlying propagation model,” in *AAAI*, 2017, pp. 217–223.
- [18] S. Dhamal, K. Prabuchandran, and Y. Narahari, “Information diffusion in social networks in two phases,” *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 4, pp. 197–210, 2016.
- [19] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou, “Identifying propagation sources in networks: State-of-the-art and comparative studies,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 465–481, 2016.
- [20] Y. Wang, A. V. Vasilakos, J. Ma, and N. Xiong, “On studying the impact of uncertainty on behavior diffusion in social networks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 2, pp. 185–197, 2014.
- [21] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *NeurIPS*, 2020, pp. 6840–6851.
- [22] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” *arXiv:1312.6114*, 2013.
- [23] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, “Regularisation of neural networks by enforcing lipschitz continuity,” *Machine Learning*, vol. 110, pp. 393–416, 2021.
- [24] S. S. Ali, T. Anwar, and S. A. M. Rizvi, “A revisit to the infection source identification problem under classical graph centrality measures,” *Online Social Networks and Media*, vol. 17, p. 100061, 2020.
- [25] P.-D. Yu, C. W. Tan, and H.-L. Fu, “Epidemic source detection in contact tracing networks: Epidemic centrality in graphs and message-passing algorithms,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 2, pp. 234–249, 2022.
- [26] P. M. Gleiser and L. Danon, “Community structure in Jazz,” *Advances in Complex Systems*, vol. 6, no. 04, pp. 565–573, 2003.
- [27] M. E. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.
- [28] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the construction of internet portals with machine learning,” *Information Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.
- [29] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.