

Deep Identification of Propagation Trees in Graph Diffusion

Zeeshan Memon^{1*}, Chen Ling^{1†}, Ruochen Kong¹, Vishwanath Seshagiri^{2†}, Andreas Züfle¹ and Liang Zhao¹

¹Emory University

²Meta

{zeeshan.memon, chen.ling, ruochen.kong, vishwanath.seshagiri, azufle, liang.zhao}@emory.edu

Abstract

Understanding how information or influence propagates through a network, such as during an epidemic outbreak or the spread of misinformation, is a fundamental yet challenging problem. While prior works have focused on cascade prediction (forecasting future infected nodes), network inference (recovering latent global diffusion graphs), or source localization (identifying diffusion’s origin), these approaches do not recover the actual “who-infected-whom” propagation tree for a specific diffusion instance. We introduce DIPT (Deep Identification of Propagation Trees), a probabilistic framework that infers propagation trees from final observed node diffusion states, without knowledge of the underlying diffusion mechanism. DIPT models local influence strengths between nodes and uses a discrete-continuous alternating optimization strategy to jointly learn the diffusion mechanism and infer the propagation structure. Empirical results across eight real-world datasets demonstrate that DIPT consistently outperforms existing approaches in reconstructing propagation trees.

1 Introduction

Graph inverse problems aim to uncover the underlying causes of observed phenomena in networks [Wang *et al.*, 2022; Ling *et al.*, 2022]. A prominent example is *diffusion source localization*, which identifies the origin(s) of spread from the final infection state. Recent GNN-based models have advanced this task [Ling *et al.*, 2022; Yan *et al.*, 2024; Wang *et al.*, 2023; Ling *et al.*, 2024]. However, source localization alone provides limited insight to understand the underlying diffusion dynamics. As shown in Fig. 1(a), source localization identifies only the origin nodes among the infected (pink). Hence, to capture the full transmission pathways revealing not just the sources, but also the sequence of infections and who influenced whom as shown in Fig. 1(b), without knowing the underlying diffusion mechanism is a

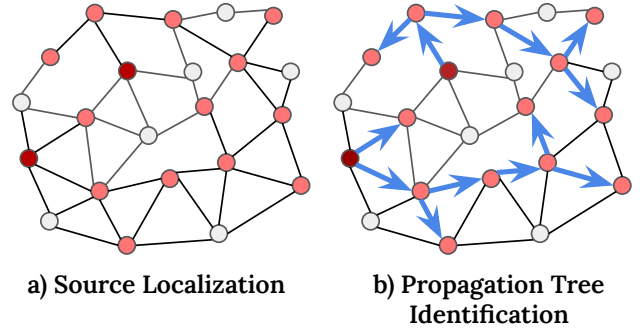


Figure 1: Given the diffused state (pink color nodes), source localization aims to identify the source nodes (red) only (a), while propagation tree identification further reveals how infection spreads, as shown by blue arrows (b).

challenging problem. This paper focuses on such a problem which we call propagation tree identification. Understanding diffusion pathways is crucial across different domains. In infectious disease modeling, identifying who infected whom, known as *contact tracing* [Mokbel *et al.*, 2020; Kleinman and Merkel, 2020], enables interventions to prevent further spread from initial carriers. In misinformation campaigns, understanding and tracing how false content propagates helps to mitigate network-wide spread [Zhou and Zafarani, 2019]. In phylogenetics, mapping genetic mutations reveals evolutionary lineages [Penny, 2004; Bouckaert *et al.*, 2014]. In most real world cases, transmission typically begins from a few sources and spreads in tree or forest-like structures, where each infected node may influence multiple others. There are research areas and works related yet distinct from propagation tree identification. The first type is prior-knowledge-based, such as Steiner-tree methods, which require the diffusion mechanism, edge costs, or influence scores to model propagation mechanism [Mishra *et al.*, 2023]. These methods typically optimize a maximum-likelihood objective over possible trees but lack learnable components, limiting their adaptability to real-world settings. In practice, such heuristics are often unavailable in domains like disease spread, e-commerce behavior, or rumor propagation. A separate line of work focuses on inferring global influence graphs from timestamped cascade data [Gomez-

*Extended version with appendices including proofs and theoretical analysis is available on arXiv: <https://arxiv.org/abs/2503.00646>.

†Work done while at Emory University.

Rodriguez *et al.*, 2012]. While these methods are learning-based, they operate on multiple cascades and infer global diffusion networks shared across cascades, rather than recovering the instance-specific propagation structure for a single observed diffusion outcome. In contrast, a third class of learning-based methods, closer to our setting, models cascade dynamics from final infection states. However, they typically aim to predict node states at fixed time points, without recovering the underlying “who-infected-whom” paths essential to propagation tree identification [Čutura *et al.*, 2021; Yan *et al.*, 2024]. Consequently, learning these propagation paths without assuming a specific diffusion mechanism remains underexplored due to the following nontrivial challenges: **1) Difficulty of inferring the unknown propagation mechanism.** Existing methods often rely on known diffusion models such as Linear Threshold (LT), Independent Cascade (IC), which make strong and often unrealistic assumptions about how information spreads. These models struggle to capture the complexity of real-world diffusion, where infection dynamics also depend on node attributes, e.g., a rumor’s transmission from user A to B might depend on their trust level, interest alignment, or social context. **2) Intractable search space.** The number of valid propagation trees leading to the observed infected nodes grows exponentially with the size of the network, making joint inference over source nodes and propagation edges highly intractable. **3) No or incomplete observation of propagation tree during training.** In realistic scenarios, only a partial observation of the diffusion process at edge level is available. For example, in epidemiological modeling, contact tracing rarely provides the full transmission path, making it challenging to learn from partial or no observations during training [Rodriguez *et al.*, 2014].

Our main contributions are:

- We propose a graph inverse problem called propagation tree identification, which aims to infer “who infected whom” given the final diffusion state, without assuming the underlying diffusion mechanism.
- We formulate a mathematical problem for the propagation tree identification which is a maximum-likelihood of a Bayesian network characterizing the diffusion process.
- We develop a discrete-continuous *alternating optimization framework* to jointly infer propagation trees (discrete problem) and learn diffusion process (continuous problem) without access to ground truth paths.

2 Previous Work

Propagation Path Reconstruction. Early methods estimate global edge weights from historical interactions and use heuristics like LeaderRank with probabilistic inference to recover global directed graphs [Gomez-Rodriguez *et al.*, 2012; Zhu *et al.*, 2016]. Deep models such as I3T and DeepIS [Tai *et al.*, 2023; Xia *et al.*, 2021] combine GNNs, sequence encoders (e.g., Bi-LSTM), and attention to capture structural and temporal diffusion features. Others improve scalability through community-aware path modeling [Zhu *et al.*, 2023], though these methods focus on forward

diffusion from known sources. A separate line of direction frames path reconstruction as a Steiner tree problem, recovering cascades via minimal-cost trees over observed infections [Mishra *et al.*, 2023; Jang *et al.*, 2021; Rozenshtein *et al.*, 2016]. While effective, these methods depend on heuristics like hand-crafted cost functions, assumptions about edge weights or infection timestamps, limiting their adaptability in realistic settings. Recent generative approaches instead aim to infer diffusion paths from final observed node states. [Qiu *et al.*, 2023] reconstructs diffusion histories from a single final snapshot by estimating posterior barycenters with MCMC and GNN-guided proposals. [Čutura *et al.*, 2021] use a Variational Autoencoder to model node trajectories, but performance degrades as time horizons grow. DDMSL [Yan *et al.*, 2024] employs discrete diffusion models for source localization and snapshot prediction, yet both fail to reconstruct explicit transmission edges—i.e., the “who-infected-whom” links critical for full path recovery.

Source Localization. Classical approaches estimate one or more sources under SI/SIR models using full or partial observations [Prakash *et al.*, 2012; Wang *et al.*, 2017; Zhu *et al.*, 2016; Zhu and Ying, 2016; Zang *et al.*, 2015; Zhu *et al.*, 2017]. GNN-based models [Dong *et al.*, 2019] improved accuracy, but often lack uncertainty quantification and incur high inference costs on large graphs. More recent methods [Ling *et al.*, 2022; Wang *et al.*, 2022; Wang *et al.*, 2023; Xu *et al.*, 2024] learn latent source distributions without strict diffusion assumptions. However, these are limited to localizing sources or reconstructing discrete-time infection states, without recovering full propagation edges.

Alternating Optimization in Graph Problems. Alternating optimization is used for tasks involving latent variables. [Xiao *et al.*, 2021] apply it in GNNs to learn personalized propagation strategies, while GLEM [Zhao *et al.*, 2022] alternates between language models and GNNs for scalable learning on text-attributed graphs. MLCO [Wang *et al.*, 2021] uses bi-level optimization for joint structure and reasoning. These works demonstrate the effectiveness of alternating updates in jointly refining model parameters and latent structures. Inspired by this, our method uses an alternating optimization to jointly infer propagation trees and learn diffusion dynamics.

3 Methodology

3.1 Problem Formulation

Given a graph $G = (V, E, \mathbf{y})$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, and $\mathbf{y} \in \{0, 1\}^{|V|}$ is a binary infection state vector indicating whether node v is infected ($\mathbf{y}_v = 1$) or not ($\mathbf{y}_v = 0$), the objective is to reconstruct a propagation tree $\mathcal{T}$ given the observed infections. Information spreads through directed propagation trees rooted at source nodes $\mathbf{s} \in \{0, 1\}^{|V|}$, where $\mathbf{s}_v = 1$ denotes that node v is a source. Each node $v \in V$ is associated with a feature

vector $\mathbf{F}_v \in \mathbb{R}^d$, forming a node feature matrix $\mathbf{F} \in \mathbb{R}^{|V| \times d}$ where d is the feature dimension.

3.2 Objective Function

We formulate propagation tree identification as a maximum a posteriori (MAP) estimation problem. The goal is to jointly infer the source distribution $\tilde{s}$ and propagation tree $\tilde{\mathcal{T}}$ that maximize the joint probability $P(s, \mathbf{y} \mid \mathcal{T}, G)$ of observed infections:

$$\tilde{s}, \tilde{\mathcal{T}} = \arg \max_{s, \mathcal{T}} P(s, \mathbf{y} \mid \mathcal{T}, G) \quad (1)$$

However, (1) cannot be solved directly. Since the infection state $\mathbf{y}$ is influenced by the graph topology G , source nodes, s , and the propagation tree $\mathcal{T}$, we can decompose the problem. This reformulation simplifies the Maximum A Posteriori (MAP) estimation as follows:

$$P(s, \mathbf{y} \mid \mathcal{T}, G) = P(\mathbf{y} \mid s, \mathcal{T}, G) \cdot P(s) \quad (2)$$

where $P(s)$ represents prior distribution over source nodes and $P(\mathbf{y} \mid s, \mathcal{T}, G)$ represents likelihood of observed infections given the source nodes and propagation tree. We model the likelihood and prior using parameterized functions: $P_\psi(\mathbf{y} \mid s, \mathcal{T}, G)$ captures the diffusion process by learning edge-level influence, and $P_\phi(s)$ encodes a learnable prior over source node distributions. This leads to the following training objective:

$$\max_{\mathcal{T}, \phi, \psi} P_\psi(\mathbf{y} \mid s, \mathcal{T}, G) \cdot P_\phi(s) \quad (3)$$

As $\mathcal{T}$ is unobserved and intractable, we optimize the objective in Eq. 3 by alternately updating (1) the propagation structure $\mathcal{T}$ given current model parameters (Section 3.4), and (2) the model parameters (ϕ, ψ) given the inferred propagation tree. This alternating optimization is summarized in Algorithm 1 and its convergence is discussed in **Theorem 2** in Appendix D.2.

3.3 Estimating the Diffusion Process

We model the likelihood term $P_\psi(\mathbf{y} \mid s, \mathcal{T}, G)$ in (3) using a recursive diffusion process along the inferred propagation tree $\mathcal{T}$. The joint infection probability is factorized as a product of local conditionals over the tree structure:

$$P_\psi(\mathbf{y} \mid s, \mathcal{T}, G) = \prod_{v \in V \setminus s} P_\psi(y_v \mid y_{\text{Pa}(v, \mathcal{T})}), \quad (4)$$

where $V \setminus s$ denotes the set of non-source nodes, and $\text{Pa}(v, \mathcal{T})$ denotes the parent of node v in the tree $\mathcal{T}$. This factorization imposes a tree-structured Markov property in which each node is conditionally dependent only on its parent. Each local conditional is parameterized by a learnable *edge transmission* produced from node features:

$$m_\psi(\text{Pa}(v) \rightarrow v) := f_\psi(\mathbf{F}_v, \mathbf{F}_{\text{Pa}(v)}) \in (0, 1), \quad (5)$$

where $\mathbf{F}_v$ and $\mathbf{F}_{\text{Pa}(v)}$ denote the feature vectors of node v and its parent, respectively, and $m_\psi(\text{Pa}(v) \rightarrow v)$ is the edge transmission probability. The marginal infection probability of a child depends recursively on the marginal of its parent (we suppress the explicit conditioning on $(s, \mathcal{T}, G)$ for brevity):

$$\begin{aligned} P_\psi(y_v=1) &= P_\psi(y_{\text{Pa}(v)}=1) m_\psi(\text{Pa}(v) \rightarrow v), \\ P_\psi(y_u=1) &= 1, \quad \forall u \in s. \end{aligned}$$

Algorithm 1: DIPT Training via Alternating Optimization

Input: $G = (V, E)$, features F , infections $\mathbf{y}$, steps K , epochs E , weights λ, μ

Output: ψ, ϕ

Init ψ, ϕ ; **for** $e = 1$ **to** E **do**

$I_{uv} = f_\psi(\mathbf{F}_u, \mathbf{F}_v), \forall (u, v) \in E$; // (5)

Sample $s \sim P_\phi(s)$; // Sec. 3.5

$P_{\text{inf}}^{(0)}(v) = P_\phi(s_v)$; // (8)

for $k = 1$ **to** K **do**

$P_{\text{inf}}^{(k)}(v) = \max(P_{\text{inf}}^{(k-1)}(v), \max_{u \in \mathcal{C}(v)} P_{\text{inf}}^{(k-1)}(u) I_{uv}), \forall v$ // (9)

$p_v = \arg \max_{u \in \mathcal{C}(v)} P_{\text{inf}}^{(K)}(u) I_{uv}$ for $y_v=1, v \notin s$;

$\mathcal{T}^* = \{(p_v, v)\}$; // (10)

$P_\psi(\mathbf{y} \mid s, \mathcal{T}^*, G) = \prod_{v \notin s} P_\psi(y_v \mid y_{p_v})$; // (4)

Compute $\mathcal{L}_{\text{diff}}, \mathcal{L}_{\text{ELBO}}$; **if** $\mathcal{T}_{\text{obs}}$:

$\mathcal{L}_{\text{obs}} = -\sum_{(u,v) \in \mathcal{T}_{\text{obs}}} \log P_\psi(y_v \mid y_u)$; // (13)

$\mathcal{L} = \lambda \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{ELBO}} + \mu \mathcal{L}_{\text{obs}}$; // (14)

Update ψ, ϕ via $\nabla \mathcal{L}$

return ψ, ϕ

To learn the diffusion process, we optimize the following negative log-likelihood over observed infection states:

$$\begin{aligned} \mathcal{L}_{\text{diff}} = - \sum_{v \in V \setminus s} & \left[y_v \log(P_\psi(y_{\text{Pa}(v)}) m_\psi(\text{Pa}(v) \rightarrow v)) \right. \\ & \left. + (1 - y_v) \log(1 - P_\psi(y_{\text{Pa}(v)}) m_\psi(\text{Pa}(v) \rightarrow v)) \right]. \end{aligned} \quad (6)$$

This loss enables the model to learn localized influence patterns across the tree structure and generalize the underlying diffusion dynamics from observed infection outcomes.

3.4 Optimizing Propagation Trees

Since the true propagation tree is unobserved during training, we estimate $\mathcal{T}^*$ by maximizing (3) while keeping the model parameters fixed with respect to $\mathcal{T}$. The inferred tree consists of directed edges linking seed nodes s to observed infected nodes $\mathbf{y}$:

$$\mathcal{T}^* = \arg \max_{\mathcal{T}} P_\psi(\mathbf{y} \mid s, \mathcal{T}, G) \cdot P_\phi(s). \quad (7)$$

To approximate this structure, we first compute edge-level influence scores using f_ψ for all edges in G ((5)), forming a sparse influence matrix¹ with entries $I_{uv} = f_\psi(\mathbf{F}_u, \mathbf{F}_v) = m_\psi(u \rightarrow v)$, where each entry represents the learned transmission strength along edge (u, v) . Propagation scores are then updated iteratively in a monotonic increasing way. At iteration $k = 0$, initial infection probabilities are given by the source prior:

$$P_{\text{inf}}^{(0)}(v) = P_\phi(s_v). \quad (8)$$

¹We do not store a full influence matrix $\mathbf{I} \in \mathbb{R}^{|V| \times |V|}$; influence scores are computed only for existing edges $(u, v) \in E$.

At iteration $k \geq 1$, the probability of node v being infected is updated by

$$P_{\text{inf}}^{(k)}(v) = \max \left(P_{\text{inf}}^{(k-1)}(v), \max_{u \in \mathcal{C}(v)} P_{\text{inf}}^{(k-1)}(u) I_{uv} \right), \quad (9)$$

where $\mathcal{C}(v)$ denotes the set of candidate parents of v , i.e., its infected neighbors in G . Because improvements can only be inherited from nodes that improved in the previous iteration, this induces a topological order on updates. Consequently, the set of parent selections defines a directed diffusion tree. Monotonicity (Theorem 1, Appendix D.1) and acyclicity (Theorem 3, Appendix D.4) proofs are discussed in the extended version. After K iterations, the propagation tree $\mathcal{T}^*$ is constructed by assigning to each infected node v the parent that maximizes its inherited likelihood under the monotone update from neighbors that became infected in earlier iterations²,

$$p_v = \arg \max_{u \in \mathcal{C}(v): t(u) < t(v)} P_{\psi}(y_u) m_{\psi}(u \rightarrow v), \quad (10)$$

which, by construction, respects the topological order induced by the updates and yields an acyclic propagation tree.

3.5 Learning the Prior of Seed Nodes

The intrinsic patterns of the prior over seed nodes, $P(s)$, are hard to model and often high dimensional, which leads to intractability. To tackle it, we map s to a latent embedding z in lower dimensional space representing the abstract semantics. A variational inference framework is used to learn an approximation of $P(s)$, capturing its structure and variability. It consists of a generative process $P_{\phi_D}(s | z)$ ³ that reconstructs s from z , and a simple prior $P(z)$, typically a standard Gaussian $\mathcal{N}(0, I)$, which regularizes the latent space. The variational posterior $q_{\phi_E}(z | s)$ approximates the intractable $P(z | s)$ and serves as the encoder. Variational inference is introduced to efficiently approximate the intractable posterior $P(z | s)$ by optimizing the Evidence Lower Bound (ELBO), ensuring that the latent variables z capture the variability of s while maintaining regularization through the prior $P(z)$. The objective is to maximize the Evidence Lower Bound for $P(s)$:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_{\phi_E}(z|s)} [\log P_{\phi_D}(s | z)] - \text{KL}(q_{\phi_E}(z | s) \| P(z)). \quad (11)$$

The first term ensures accurate reconstruction of s , while the second term regularizes the latent distribution to match the prior $P(z)$. Learning prior over source nodes directly addresses the challenge of intractable search space (Challenge 2). In real-world diffusion, source nodes are often sparse, learning a prior provides a compact representation of source configurations, enabling the model to learn a distribution over plausible sparse source sets. Sampling from this learned prior avoids the need to enumerate an exponential number of candidate sources, thereby narrowing the joint search space over source nodes and propagation trees.

²Here $t(v)$ denotes the iteration index at which node v first becomes infected, i.e., the first iteration where $P_{\text{inf}}^{(k)}(v)$ strictly increases.

³ ϕ_D and ϕ_E represents decoder and encoder respectively

3.6 Incorporating Partial Propagation Path Observations

The propagation tree is unobservable during training in our problem setting, so the diffusion process is inferred solely from the observed infection states $\mathbf{y}$. However, in some scenarios when partial observations of the propagation tree $\mathcal{T}_{\text{obs}} \subset \mathcal{T}$ are available, DIPT can also incorporate such supervision to improve training. In such setting, the objective function in Eq. 1 is extended to:

$$P(s, \mathbf{y} | \mathcal{T}_{\text{obs}}, \mathcal{T}_{\text{unobs}}, G) = P(\mathbf{y} | s, \mathcal{T}_{\text{obs}} \cup \mathcal{T}_{\text{unobs}}, G) \cdot P(\mathcal{T}_{\text{obs}} | s, \mathbf{y}, G) \cdot P(s) \quad (12)$$

The unobserved portion $\mathcal{T}_{\text{unobs}}$ is inferred as before, while observed edges $(u, v) \in \mathcal{T}_{\text{obs}}$ provide a supervised loss signal and maximize the conditional probability $P_{\psi}(\mathbf{y}_u | \mathbf{y}_v)$:

$$\mathcal{L}_{\text{obs}} = - \sum_{(u,v) \in \mathcal{T}_{\text{obs}}} \log P_{\psi}(\mathbf{y}_u | \mathbf{y}_v) \quad (13)$$

This objective encourages the model to assign higher likelihood to edges consistent with known propagation tree edges using direct supervision. When supervision is available, the training loss becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ELBO}} + \lambda \cdot \mathcal{L}_{\text{diff}} + \mu \cdot \mathcal{L}_{\text{obs}} \quad (14)$$

where λ and μ are hyperparameters. This formulation enables DIPT to transition smoothly between unsupervised and partially supervised training.

3.7 Test Time Inference of Propagation Tree

At inference time, we aim to infer the optimal propagation tree given an observed infection state $\mathbf{y}$. A learnable latent vector $\hat{z}$ is initialized, which is optimized while freezing model parameters ϕ and ψ . This enables test-time optimization of $P_{\phi}(s)$ through z rather than sampling directly. The latent vector $\hat{z}$ is initialized as the empirical average $\bar{z}$ over the training set and optimized to maximize the expected likelihood of observed infections:

$$\mathcal{L}_{\text{inf}} = \max_{\hat{z}} \mathbb{E}_{s \sim P_{\phi_D}(s|\hat{z})} [P_{\psi}(\mathbf{y} | s, \mathcal{T}^*, G)] - \gamma \cdot \|\hat{z} - \bar{z}\|^2 \quad (15)$$

The regularization term constrains $\hat{z}$ to remain close to the training distribution, mitigating test-time drift. The propagation tree $\mathcal{T}^*$ is inferred from $s \sim P_{\phi_D}(s | \hat{z})$ using the same propagation tree inference process described in training (see Eq. (8)–(11)). The constrained objective in (15) cannot be computed directly, so we provide a practical version of the inference objective: since the diffused observation y fits the Gaussian distribution and the seed set s fits the Bernoulli distribution (see Appendix B.1), we can simplify (15) as:

$$\mathcal{L}_{\text{inf}}^* = \min_{\hat{z}} \left[- \sum_{i=1}^N \log (f_{\phi_D}(\hat{z})^{s_i} (1 - f_{\phi_D}(\hat{z}))^{1-s_i}) + \frac{1}{2} \|\hat{\mathbf{y}} - \mathbf{y}\|^2 + \lambda \|\hat{z} - \bar{z}\|^2 \right]. \quad (16)$$

Here, the first term captures the Bernoulli likelihood of the seed nodes, and the second term represents the Gaussian likelihood. The test-time inference process does not introduce

any new training objective. Instead, it optimizes the latent source representation $\hat{z}$ given the observed diffused state y , while keeping the learned diffusion model fixed. This enables instance-specific adaptation through the same likelihood computation and propagation tree inference steps used during training (see Algorithm 2 in Appendix B), and a **theoretical analysis** of the error correction of propagation tree is provided in Appendix D.3.

4 Experimental Setup

4.1 Datasets and Evaluation Metrics

We evaluate DIPT and baselines on eight real-world graph datasets. Memetracker [Rodriguez *et al.*, 2011] captures real diffusion via hyperlink cascades across online articles. We use a subnetwork with 583 sites and 6.7k cascades, extracting source nodes as the earliest 5% per cascade and each cascade is converted into a propagation tree. Cora-ML [Rossi and Ahmed, 2015], CiteSeer [Wu *et al.*, 2016], and Power Grid [Watts and Strogatz, 1998] provide the node features and topology of the citation networks and infrastructure grids respectively. Cascades are obtained by simulating diffusion by randomly selecting 10% of nodes as sources and running the SI model for 200 steps until convergence. Twitter and Weibo are large-scale real-world datasets of information diffusion in social networks, with 616k and 490k nodes respectively, where each cascade records user-to-user retweet paths over time. For our problem setting, we extract the final diffusion state and treat the observed retweet tree as ground truth. Similarly, Pol [Conover *et al.*, 2011] is a temporal retweet network about a U.S. political event. Further dataset preprocessing details appear in Appendix B.3.

Infectious Disease Simulated Spread (IDSS) Dataset.

To evaluate performance in large-scale, realistic epidemics, we simulate disease spread using a spatial SIR model [Kermack and McKendrick, 1932] over 3,143 U.S. counties, incorporating real mobility data [Kang *et al.*, 2020; SafeGraph, Inc., 2021]. Infections seed via air travel and evolve under a time-varying reproduction number. Each run generates a propagation forest over 500–1,000 counties and 12k–48k infections. Full details and data are in Appendix A. The simulation code and datasets are available at <https://github.com/RuoChenKong/Compartmental-Infectious-Disease-Simulation>.

Evaluation Metrics. We evaluate DIPT on two tasks: propagation tree identification and source localization. For tree identification, we use *Jaccard Index* and *Path Precision*. Jaccard captures structural overlap between predicted and ground-truth edges, while Path Precision reflects correctness along true diffusion paths. For source localization, we use standard classification metrics to assess the accuracy of source localization.

4.2 Comparison Methods and Experimental Settings

Most prior approaches focus on predicting global diffusion graphs from cascades, using *non-learnable heuristics*, MLP-

like models, or classical methods that assume timestamped infections and fixed diffusion mechanisms. These differ fundamentally from DIPT, which reconstructs instance-level propagation trees from a single final infection state. We therefore compare against learnable diffusion models which can capture infection dynamics from final diffusion observations, as:

- *Generative Diffusion Models.* **DITTO** [Qiu *et al.*, 2023] reconstructs diffusion histories from a single final snapshot by estimating posterior barycenters with MCMC and GNN-guided proposals. **DDMIX** [Čutura *et al.*, 2021] uses a VAE to reconstruct diffusion trajectories by learning latent node states. **DDMSL** [Yan *et al.*, 2024] models invertible diffusion to recover node-level timestamps. These methods are adapted to approximate propagation paths by combining random forward and backward walks (MCMC in the case of DITTO) between nodes active in adjacent diffusion steps. This enables a fair comparison to DIPT in terms of reconstructing propagation tree.
- *Source Localization Methods.* We also benchmark DIPT on source localization task against four methods: (1) **LPSI** [Wang *et al.*, 2017]: labels-based inference without explicit dynamics. (2) **OJC** [Zhu *et al.*, 2017]: SIR-based inference from partial observations. (3) **GCNSI** [Dong *et al.*, 2019]: GCN-based prediction of multiple sources. (4) **SLVAE** [Ling *et al.*, 2022]: variational modeling of source distributions.

Experimental configurations, implementation details, including tuned hyperparameters and training setup are provided in Appendix B.2.

4.3 Experimental Results

Propagation Trees Prediction Performance

Table 1 compares DIPT against three baselines for propagation tree identification. DIPT consistently outperforms all three across eight datasets on path precision and Jaccard index. The gap is especially large on denser graphs like Power Grid, IDSS, Memetracker, where baselines struggle due to complex propagation dynamics and real-world noise. On sparser datasets (Cora-ML, CiteSeer), the margin is smaller but still favorable. This demonstrates DIPT’s robustness in synthetic and real-world settings, enabled by edge-level influence modeling (Eq. 5). DIPT achieves its best results on Power Grid (68% path precision, 51.5% Jaccard), and handles the challenging IDSS dataset with 42.1% path precision—far surpassing DDMSL (12.1%) and DDMIX (10.9%). It also generalizes well to large-scale graphs, achieving strong performance on Twitter (58.1%, 41.8%) and Weibo (55.2%, 39.7%), where baselines underperform. Figure 2 illustrates predicted trees on Memetracker⁴ DDMSL and DDMIX exhibit more incorrect edge predictions (orange edges) in dense subgraphs. DIPT better handles such dense regions by learning influence between nodes, leading to more accurate edge selection. Additional visualizations are in Appendix F. On

⁴Every infected (pink) node has a parent. Some appear without incoming edges because their parent lies outside the displayed subgraph. Due to space constraints, nodes are subsampled for visualization

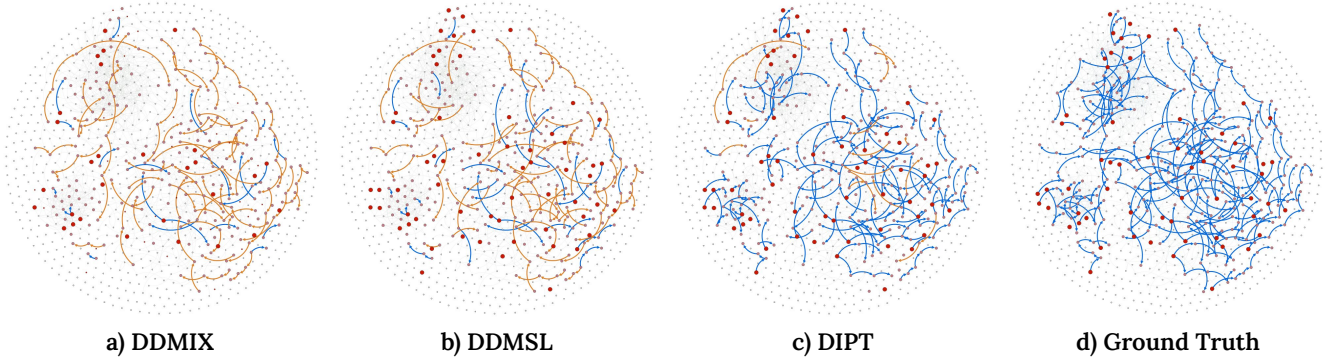


Figure 2: Predicted vs. ground truth propagation trees on Memetracker. Correctly predicted edges are shown in blue, incorrect ones in orange. Source nodes are marked in red, and infected nodes in pink.

Method	Cora-ML		Memetracker		CiteSeer		Power Grid		IDSS		Twitter		Weibo		Pol	
	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.
DDMIX	0.327	0.195	0.062	0.041	0.236	0.133	0.081	0.031	0.109	0.057	0.211	0.122	0.203	0.118	0.198	0.115
DITTO	0.371	0.218	0.091	0.052	0.312	0.178	0.106	0.050	0.115	0.061	0.258	0.154	0.210	0.124	0.242	0.148
DDMSL	0.412	0.259	0.119	0.063	0.405	0.253	0.130	0.069	0.121	0.064	0.312	0.187	0.216	0.130	0.298	0.176
DIPT	0.622	0.452	0.602	0.430	0.593	0.421	0.680	0.515	0.421	0.266	0.581	0.418	0.552	0.397	0.565	0.404

Table 1: Performance comparison for Propagation Tree Identification. Best results are in bold. Prec. = Path Precision, Jac. = Jaccard Index.

Method	Cora-ML		Memetracker		CiteSeer		Power Grid		IDSS		Twitter		Weibo		Pol	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
LPSI	0.301	0.592	0.014	0.529	0.306	0.598	0.474	0.934	0.037	0.540	0.201	0.578	0.227	0.552	0.193	0.575
OJC	0.121	0.534	0.026	0.517	0.117	0.530	0.153	0.501	0.028	0.520	0.084	0.491	0.077	0.470	0.082	0.490
GCNSI	0.401	0.687	0.035	0.422	0.387	0.680	0.330	0.639	0.045	0.430	0.287	0.603	0.265	0.567	0.261	0.600
SLVAE	0.764	0.831	0.488	0.624	0.749	0.825	0.797	0.879	0.494	0.630	0.253	0.578	0.209	0.547	0.209	0.576
DDMIX	0.221	0.247	0.022	0.417	0.215	0.250	0.280	0.340	0.029	0.425	0.176	0.464	0.155	0.439	0.168	0.470
DITTO	0.384	0.660	0.115	0.545	0.369	0.660	0.488	0.700	0.151	0.555	0.230	0.595	0.209	0.555	0.239	0.585
DDMSL	0.750	0.873	0.515	0.641	0.742	0.870	0.831	0.866	0.527	0.645	0.313	0.625	0.381	0.622	0.317	0.630
DIPT	0.839	0.881	0.518	0.629	0.832	0.880	0.828	0.864	0.525	0.630	0.421	0.648	0.439	0.658	0.429	0.655

Table 2: Source Localization performance comparison based on F1 score and AUC (best and second-best in bold).

average, DIPT improves path precision and Jaccard index by 3.5 $\times$ and 4.37 $\times$, respectively, over all three baselines.

Source Localization Accuracy

We evaluate DIPT against six source localization methods, with results summarized in Table 2. DIPT performs competitively with DDMSL and SLVAE across all datasets and significantly outperforms LPSI, OJC, and GCNSI. DIPT achieves the highest F1 (0.839) and AUC (0.881) scores on Cora-ML and performs strongly on CiteSeer, consistent with its superior propagation tree accuracy on these less dense graphs. On IDSS, which exhibits varying seed distributions due to mobility patterns, DIPT matches DDMSL’s performance (F1: 0.525, AUC: 0.630). It also outperforms all baselines on large-scale Twitter and Weibo graphs, achieving F1 scores of 0.421 and 0.439, respectively, demonstrating strong generalization to large networks. These results show that de-

spite solving the harder task of full propagation tree inference, DIPT achieves source localization accuracy comparable to specialized methods. Precision and Recall metrics are included in Appendix E.2.

Impact of Partial Observations

As DIPT is primarily designed to learn propagation trees without edge supervision, Section 3.6 explores how incorporating partial tree information during training affects performance. Table 3 shows that providing 10%, 20%, and 30% of edge annotations leads to average path precision gains of 7.11%, 11.84%, and 15.16%, respectively. Similar improvements are observed on Twitter and Weibo; for example, Twitter precision rises from 0.581 (unsupervised) to 0.595 (10% supervision), highlighting DIPT’s adaptability across graph scales. These gains stem from the supervised loss in Eq. 13, which enhances learning of influence patterns. Since infer-

Data %	Cora-ML		Memetracker		CiteSeer		Power Grid		IDSS		Twitter		Weibo		Pol	
	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.
10%	0.671	0.504	0.633	0.463	0.662	0.495	0.683	0.518	0.437	0.279	0.595	0.430	0.567	0.410	0.582	0.422
20%	0.707	0.546	0.651	0.482	0.671	0.504	0.718	0.560	0.451	0.291	0.612	0.445	0.579	0.425	0.599	0.439
30%	0.720	0.562	0.662	0.494	0.695	0.532	0.759	0.611	0.453	0.292	0.628	0.461	0.597	0.442	0.618	0.455

Table 3: Performance of DIPT with varying proportions of partially observed propagation tree data during training.

Ablation	Cora-ML		Memetracker		CiteSeer		Power Grid		IDSS		Twitter		Weibo		Pol	
	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.	Prec.	Jac.
DIPT (a)	0.388	0.235	0.407	0.255	0.422	0.267	0.437	0.276	0.329	0.206	0.368	0.248	0.362	0.243	0.372	0.252
DIPT (b)	0.519	0.350	0.489	0.326	0.511	0.343	0.607	0.435	0.371	0.227	0.495	0.335	0.489	0.330	0.502	0.341
DIPT	0.622	0.452	0.602	0.430	0.593	0.421	0.680	0.515	0.421	0.266	0.581	0.418	0.552	0.397	0.565	0.404

Table 4: Performance of DIPT under different ablation settings.

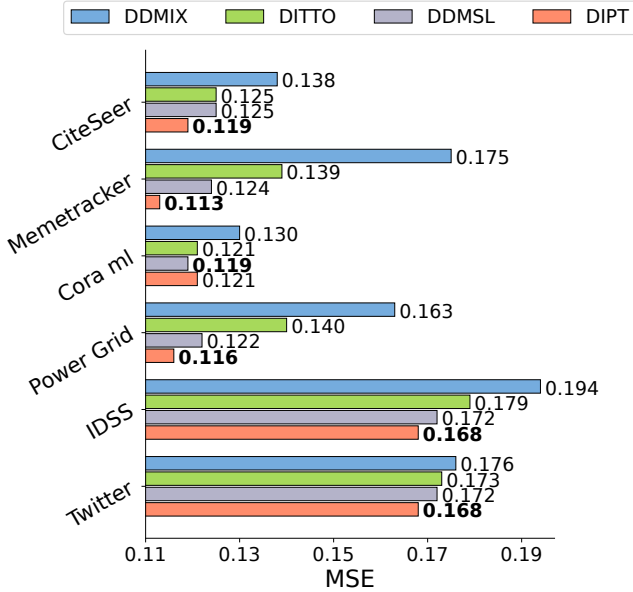


Figure 3: Comparison of DIPT, DITTO, DDMSL, and DDMIX on diffusion reconstruction (MSE; lower is better).

ence maximizes infection likelihood, supervision scales non-linearly with annotation.

Diffusion State Reconstruction

Both DDMSL and DDMIX reconstruct node states at discrete time steps during the diffusion process, capturing an infection sequence that reflects the order in which nodes became infected (i.e., who was infected after whom). In contrast, DIPT not only recovers this infection order (induced from the inferred propagation tree) but also identifies the source of each infection (i.e., who infected whom), thereby reconstructing the propagation tree. To ensure a fair comparison, we evaluate all methods on node state reconstruction accuracy with respect to the infection order. As shown in Fig. 3,

DIPT achieves an average mean squared error (MSE) that is 19.67% lower than DDMIX, 5.2% lower than DITTO and 3.85% lower than DDMSL. These results highlight that, beyond recovering propagation tree, DIPT accurately models node-level infection dynamics as well.

Ablation Results

We conduct an ablation study to assess the contribution of each core component in DIPT. In the first variant, DIPT(a), the learned edge influence (Eq. 5) is replaced with cosine similarity between node features. In DIPT(b), the model directly optimizes the joint objective in Eq. 3 without alternating between tree inference and parameter updates during training; however, inference still uses the propagation tree inference described in Section 3.7. Results are reported in Table 4. Both ablations lead to notable performance drops. In DIPT(a), accuracy degrades significantly because edge-level influence is no longer learned—scores remain static, undermining the model’s ability to capture propagation structure. On Cora-ML, DIPT(a) even underperforms DDMSL in path precision (Table 1), though it remains competitive on denser datasets where simple similarity still provides some signal. DIPT(b) performs better than DIPT(a) and continues to outperform all baselines, but its performance lags behind full DIPT. This highlights that while both components are important, learning local node influence is more critical than alternating optimization.

Additional Experimental Results. Results on generalization, feature robustness, intermediate-observation accuracy, and inference time are in Appendix E.

5 Conclusion

We introduce DIPT, a probabilistic framework that infers propagation trees from observed diffusion data via alternating optimization, jointly learning edge-level influence and propagation structure without observing paths during training. Overall, experiments on eight real-world datasets show DIPT consistently outperforms strong baselines on propagation path identification (58.2% avg. path precision).

Ethical Statement

This work uses only publicly available datasets previously released for research purposes. The IDSS dataset is synthetic and contains no personally identifiable information. The work does not involve new data collection from human subjects.

Acknowledgements

This work was supported by the National Science Foundation (NSF) Grant Nos. 2302968, 2109647, 2414115, 2403312, 2113350, and 2411530; the National Institutes of Health (NIH) Grants R01AG089806 and R01CA297856; and a Sony Research Award.

References

- [Bouckaert *et al.*, 2014] Remco Bouckaert, Joseph Heled, Denise Kühnert, Tim Vaughan, Chieh-Hsi Wu, Dong Xie, Marc A Suchard, Andrew Rambaut, and Alexei J Drummond. Beast 2: a software platform for bayesian evolutionary analysis. *PLoS computational biology*, 10(4):e1003537, 2014.
- [Conover *et al.*, 2011] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *Proceedings of the international aai conference on web and social media*, volume 5, pages 89–96, 2011.
- [Čutura *et al.*, 2021] Gojko Čutura, Boning Li, Ananthram Swami, and Santiago Segarra. Deep demixing: Reconstructing the evolution of epidemics using graph neural networks. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 2204–2208. IEEE, 2021.
- [Dong *et al.*, 2019] Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. Multiple rumor source detection with graph convolutional networks. In *CIKM*, pages 569–578, 2019.
- [Gomez-Rodriguez *et al.*, 2012] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–37, 2012.
- [Jang *et al.*, 2021] Hankyu Jang, Shreyas Pai, Bijaya Adhikari, and Sriram V Pemmaraju. Risk-aware temporal cascade reconstruction to detect asymptomatic cases: For the cdc mind healthcare network. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 240–249. IEEE, 2021.
- [Kang *et al.*, 2020] Yuhao Kang, Song Gao, Yunlei Liang, Mingxiao Li, Jinneng Rao, and Jake Kruse. Multiscale dynamic human mobility flow dataset in the us during the covid-19 epidemic. *Scientific data*, 7(1):390, 2020.
- [Kermack and McKendrick, 1932] William Ogilvy Kermack and Anderson G McKendrick. Contributions to the mathematical theory of epidemics. ii.—the problem of endemicity. *Proceedings of the Royal Society of London. Series A, containing papers of a mathematical and physical character*, 138(834):55–83, 1932.
- [Kleinman and Merkel, 2020] Robert A Kleinman and Colin Merkel. Digital contact tracing for covid-19. *Cmaj*, 192(24):E653–E656, 2020.
- [Ling *et al.*, 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.
- [Ling *et al.*, 2024] Chen Ling, Tanmoy Chowdhury, Jie Ji, Sirui Li, Andreas Züfle, and Liang Zhao. Source localization for cross network information diffusion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5419–5429, 2024.
- [Mishra *et al.*, 2023] Ritwick Mishra, Jack Heavey, Gursharn Kaur, Abhijin Adiga, and Anil Vullikanti. Reconstructing an epidemic outbreak using steiner connectivity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11613–11620, 2023.
- [Mokbel *et al.*, 2020] Mohamed Mokbel, Sofiane Abbar, and Rade Stanojevic. Contact tracing: Beyond the apps. *SIGSPATIAL Special*, 12(2):15–24, 2020.
- [Penny, 2004] David Penny. Inferring phylogenies.—joseph felsenstein. 2003. sinauer associates, sunderland, massachusetts., 2004.
- [Prakash *et al.*, 2012] B Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. Spotting culprits in epidemics: How many and which ones? In *ICDM*, pages 11–20, 2012.
- [Qiu *et al.*, 2023] Ruizhong Qiu, Dingsu Wang, Lei Ying, H Vincent Poor, Yifang Zhang, and Hanghang Tong. Reconstructing graph diffusion history from a single snapshot. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1978–1988, 2023.
- [Rodriguez *et al.*, 2011] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697*, 2011.
- [Rodriguez *et al.*, 2014] Manuel Gomez Rodriguez, Jure Leskovec, David Balduzzi, and Bernhard Schölkopf. Uncovering the structure and temporal dynamics of information propagation. *Network Science*, 2(1):26–65, 2014.
- [Rossi and Ahmed, 2015] Ryan Rossi and Nesreen Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.
- [Rozenshtein *et al.*, 2016] Polina Rozenshtein, Aristides Gionis, B Aditya Prakash, and Jilles Vreeken. Reconstructing an epidemic over time. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1835–1844, 2016.
- [SafeGraph, Inc., 2021] SafeGraph, Inc. SafeGraph Foot Traffic Data. <https://www.safegraph.com/guides/foot-traffic-data>, 2021. Accessed: 2026-05-17.

- [Tai *et al.*, 2023] Yu Tai, Hui He, Weizhe Zhang, Hongwei Yang, Xinglong Wu, and Yan Wang. Predicting information diffusion using the inter-and intra-path of influence transitivity. *Information Sciences*, 651:119705, 2023.
- [Wang *et al.*, 2017] Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. Multiple source detection without knowing the underlying propagation model. In *AAAI*, 2017.
- [Wang *et al.*, 2021] Runzhong Wang, Zhigang Hua, Gan Liu, Jiayi Zhang, Junchi Yan, Feng Qi, Shuang Yang, Jun Zhou, and Xiaokang Yang. A bi-level framework for learning to solve combinatorial optimization on graphs. *Advances in neural information processing systems*, 34:21453–21466, 2021.
- [Wang *et al.*, 2022] 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.
- [Wang *et al.*, 2023] Zhen Wang, Dongpeng Hou, Chao Gao, Xiaoyu Li, and Xuelong Li. Lightweight source localization for large-scale social networks. In *Proceedings of the ACM Web Conference 2023*, pages 286–294, 2023.
- [Watts and Strogatz, 1998] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440–442, 1998.
- [Wu *et al.*, 2016] Jian Wu, Chen Liang, Huaiyu Yang, and C Lee Giles. Citeseerx data: Semanticizing scholarly papers. In *Proceedings of the International Workshop on Semantic Big Data*, pages 1–6, 2016.
- [Xia *et al.*, 2021] Wenwen Xia, Yuchen Li, Jun Wu, and Shenghong Li. Deepis: Susceptibility estimation on social networks. In *WSDM*, pages 761–769, 2021.
- [Xiao *et al.*, 2021] Teng Xiao, Zhengyu Chen, Donglin Wang, and Suhang Wang. Learning how to propagate messages in graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1894–1903, 2021.
- [Xu *et al.*, 2024] Xovee Xu, Tangjiang Qian, Zhe Xiao, Ni Zhang, Jin Wu, and Fan Zhou. Pgsi: A probabilistic graph diffusion model for source localization. *Expert Systems with Applications*, 238:122028, 2024.
- [Yan *et al.*, 2024] Xin Yan, Hui Fang, and Qiang He. Diffusion model for graph inverse problems: Towards effective source localization on complex networks. *Advances in Neural Information Processing Systems*, 36, 2024.
- [Zang *et al.*, 2015] Wenyu Zang, Peng Zhang, Chuan Zhou, and Li Guo. Locating multiple sources in social networks under the sir model: A divide-and-conquer approach. *Journal of Computational Science*, 10:278–287, 2015.
- [Zhao *et al.*, 2022] Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*, 2022.
- [Zhou and Zafarani, 2019] Xinyi Zhou and Reza Zafarani. Network-based fake news detection: A pattern-driven approach. *ACM SIGKDD explorations newsletter*, 21(2):48–60, 2019.
- [Zhu and Ying, 2016] Kai Zhu and Lei Ying. Information source detection in the sir model: A sample-path-based approach. *IEEE/ACM TON*, 24(1):408–421, 2016.
- [Zhu *et al.*, 2016] Hengmin Zhu, Xicheng Yin, Jing Ma, and Wei Hu. Identifying the main paths of information diffusion in online social networks. *Physica A: Statistical Mechanics and its Applications*, 452:320–328, 2016.
- [Zhu *et al.*, 2017] Kai Zhu, Zhen Chen, and Lei Ying. Catch’em all: Locating multiple diffusion sources in networks with partial observations. In *AAAI*, 2017.
- [Zhu *et al.*, 2023] Hengmin Zhu, Xinyi Yang, and Jing Wei. Path prediction of information diffusion based on a topic-oriented relationship strength network. *Information Sciences*, 631:108–119, 2023.