Reconstructing Graph Diffusion History from a Single Snapshot

Ruizhong Qiu
rq5@illinois.edu
University of Illinois
Urbana-Champaign, IL, USA

Dingsu Wang
dingsuw2@illinois.edu
University of Illinois
Urbana-Champaign, IL, USA

Lei Ying
leiying@umich.edu
University of Michigan
Ann Arbor, MI, USA

H. Vincent Poor
poor@princeton.edu
Princeton University
Princeton, NJ, USA

Yifang Zhang
zhang303@illinois.edu
C3.ai Digital Transformation Institute
Urbana, IL, USA

Hanghang Tong
htong@illinois.edu
University of Illinois
Urbana-Champaign, IL, USA

ABSTRACT

Diffusion on graphs is ubiquitous with numerous high-impact applications, ranging from the study of residential segregation in socioeconomics and activation cascading in neuroscience, to the modeling of disease contagion in epidemiology and malware spreading in cybersecurity. In these applications, complete diffusion histories play an essential role in terms of identifying dynamical patterns, reflecting on precaution actions, and forecasting intervention effects. Despite their importance, complete diffusion histories are rarely available and are highly challenging to reconstruct due to ill-posedness, explosive search space, and scarcity of training data. To date, few methods exist for diffusion history reconstruction. They are exclusively based on the maximum likelihood estimation (MLE) formulation and require to know true diffusion parameters. In this paper, we study an even harder problem, namely *reconstructing Diffusion history from A single SnapsHot* (DASH), where we seek to reconstruct the history from only the final snapshot without knowing true diffusion parameters. We start with theoretical analyses that reveal a fundamental limitation of the MLE formulation. We prove: (a) estimation error of diffusion parameters is unavoidable due to NP-hardness of diffusion parameter estimation, and (b) the MLE formulation is sensitive to estimation error of diffusion parameters. To overcome the inherent limitation of the MLE formulation, we propose a novel *barycenter formulation*: finding the barycenter of the posterior distribution of histories, which is provably stable against the estimation error of diffusion parameters. We further develop an effective solver named *DIffusion hiTting Times with Optimal proposal* (DITTO) by reducing the problem to estimating posterior expected hitting times via the Metropolis–Hastings Markov chain Monte Carlo method (M–H MCMC) and employing an unsupervised graph neural network to learn an optimal proposal to accelerate the convergence of M–H MCMC. We conduct extensive experiments to demonstrate the efficacy of the proposed method. Our code is available at https://github.com/q-rz/KDD23-DITTO.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599488

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; • **Computing methodologies** → *Neural networks*.

KEYWORDS

Graph Diffusion, History Reconstruction, Markov Chain Monte Carlo (MCMC), Graph Neural Network (GNN)

ACM Reference Format:
Ruizhong Qiu, Dingsu Wang, Lei Ying, H. Vincent Poor, Yifang Zhang, and Hanghang Tong. 2023. Reconstructing Graph Diffusion History from a Single Snapshot. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3580305.3599488

1 INTRODUCTION

Diffusion on graphs is ubiquitous in various domains owing to its generality in representing complex dynamics among interconnected objects. It appears in numerous high-impact applications, ranging from the study of residential segregation in socioeconomics [73] and activation cascading in neuroscience [3], to the modeling of disease contagion in epidemiology [51] and malware spreading in cybersecurity [88]. At the core of these applications are the *complete diffusion histories* of the underlying diffusion process, which could be exploited to identify dynamical patterns [19], reflect on precaution actions [11], forecast intervention effects [84], etc.

Despite their importance, complete diffusion histories are rarely available in real-world applications because diffusion may not be noticed in early stages, collecting diffusion histories may incur unaffordable costs, and/or tracing node states may raise privacy concerns [16, 74]. It is thus highly desirable to develop learning-based methods to automatically reconstruct diffusion histories from limited observations. However, diffusion history reconstruction faces critical challenges. **(i) Ill-posed inverse problem.** Since different histories can result in the same observation, it is difficult to distinguish which history is preferred. Hence, it is crucial to design a formulation with desired inductive bias. **(ii) Explosive search space.** The number of possible histories grows exponentially with the number of nodes, so history estimation is an extremely high-dimensional combinatorial problem. **(iii) Scarcity of training data.** Conventional methods for time series imputation such as supervised learning (e.g., [9, 18, 43, 58]) require training data to learn

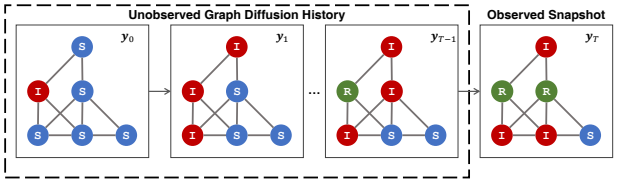


Figure 1: Illustration of the DASH problem. This is an SIR diffusion process on a graph, where each square box represents a snapshot $\boldsymbol{y}_t$ at each time t. In the DASH problem, only the final snapshot $\boldsymbol{y}_T$ is observed, and we need to reconstruct all the unobserved snapshots $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}$.

from, but true diffusion histories are rarely available. Thus, they turn out to be inefficacious or even inapplicable for this problem.

Compared to extensive research on forward problems on diffusion (e.g., node immunization [41]; see Sec. 6 for a survey), few works exist for diffusion history reconstruction. The sparse literature on diffusion history reconstruction is exclusively based on the maximum likelihood estimation (MLE) formulation [16, 74], which relies on two assumptions: (1) knowing true diffusion parameters and/or (2) knowing partial diffusion histories. However, neither diffusion parameters or partial diffusion history is available in many real-world applications. For example, when an epidemic is noticed for the first time, we might only know who are *currently* infected but have no historical infection information (e.g., who the patient zero is, when outbreaks happen, and where super-spreaders locate).

To address these limitations, we study a novel and more realistic setting: reconstructing a complete Diffusion history from A single SnapsHot (DASH). Importantly, we remove both assumptions of existing works. That is, we do not require knowing true diffusion parameters, and we only have access to the current snapshot.

Under such a realistic setting, we start with theoretical analyses revealing a fundamental limitation of the MLE formulation for the DASH problem, which motivates us to propose a novel *barycenter formulation* that addresses the limitation. We then develop an effective solver named *DIffusion hiTting Times with Optimal proposal* (DITTO) for the barycenter formulation. Our method is unsupervised and thus does not require real diffusion histories as training data, which is desirable due to the scarcity of training data. Extensive experiments demonstrate the efficacy and the scalability of our method DITTO. The main contributions of this paper are:

- **Problem definition.** Motivated by real-world challenges, we propose and study a new problem DASH. This challenging problem assumes that only the current snapshot is observed, while previous works rely on partial diffusion histories and/or require to know true diffusion parameters.
- **Theoretical insights.** We reveal a fundamental limitation of the MLE formulation for DASH. We prove: (a) estimation error of diffusion parameters is inevitable due to NP-hardness of diffusion parameter estimation, and (b) the MLE formulation is sensitive to estimation error of diffusion parameters.
- **Methodology.** We propose a novel *barycenter formulation* for DASH, which is *provably stable* against the estimation error of diffusion parameters. We further develop an effective method DITTO, which reduces DASH to estimating hitting times via MCMC and employs an unsupervised GNN to learn an optimal proposal to accelerate the convergence of MCMC. DITTO has time complexity $\mathrm{O}(T(n \log n + m))$ scaling near-linearly w.r.t. the output size $\Theta(Tn)$, where T is the timespan, and n, m are the numbers of nodes and edges, respectively.
- **Evaluation.** We conduct extensive experiments with both synthetic and real-world datasets, and DITTO consistently achieves strong performance on all datasets. For example, DITTO is 10.06% better than the best baseline on the Covid dataset in terms of normalized rooted mean squared error.

Table 1: Main notations.

Symbol	Definition
$\mathcal{T}$	the set of time
$\mathcal{X}$	the set of diffusion states
$\mathcal{V}$	the set of nodes
$\mathcal{E}$	the set of edges
$\mathcal{N}_u$	the set of neighbors of node u
T	the timespan of interest
$n^{\mathcal{X}_0}$	the number of nodes with state in $\mathcal{X}_0$
$y_{t,u}$	the state of node u at time t
$\boldsymbol{y}_t$	a snapshot of diffusion at time t
$\boldsymbol{Y}$	a complete diffusion history
$\widehat{\boldsymbol{Y}}$	the reconstructed diffusion history
S, I, R	states in the SIR model
β^{I}, β^{R}	the infection rate and the recovery rate
$\boldsymbol{\beta}$	true diffusion parameters
$\widehat{\boldsymbol{\beta}}$	estimated diffusion parameters
$P_{\boldsymbol{\beta}}$	the probability measure of the diffusion model
$P_{\boldsymbol{\beta}} \mid \boldsymbol{y}_T$	the posterior given the observed snapshot
$\mathrm{supp}(P)$	the set of possible histories
$\mathrm{supp}(P \mid \boldsymbol{y}_T)$	the set of histories consistent with the snapshot $\boldsymbol{y}_T$
$\searrow$	the one-sided limit from above
∂	the partial derivative operator
∇	the gradient operator
$\mathbb{E}$	the expectation operator
O, Θ	the asymptotic notations

2 PROBLEM DEFINITION

In this section, we formally define the DASH problem. Our notation conventions are as follows. We use the calligraphic font for sets (e.g., $\mathcal{E}$); lightface uppercase letters for constants (e.g., T) and probability distributions (e.g., P); lightface lowercase letters for indices (e.g., t) and scalar-valued functions (e.g., ψ); boldface lowercase letters for vectors (e.g., $\boldsymbol{\beta}$); boldface uppercase letters for matrices (e.g., $\boldsymbol{Y}$); the monospaced font for states (e.g., S); and the hat notation for estimates (e.g., $\widehat{\boldsymbol{\beta}}$). Notations are summarized in Table 1.

2.1 Preliminaries

2.1.1 Diffusion on Graphs. We study discrete-time diffusion processes, where $\mathcal{T} := \{0, 1, \ldots, T\}$ denotes the set of time, and $\mathcal{X}$ denotes the set of diffusion states. The graph is *undirected*, with node set $\mathcal{V}$ and edge set $\mathcal{E}$, where the number of nodes is $|\mathcal{V}| = n$, and the number of edges is $|\mathcal{E}| = m$. For each node $u \in \mathcal{V}$, let $\mathcal{N}_u := \{v \in \mathcal{V} : (u, v) \in \mathcal{E}\}$ denote the neighbors of node u.

Let $y_{t,u} \in \mathcal{X}$ denote the state of a node $u \in \mathcal{V}$ at a time $t \in \mathcal{T}$. A *diffusion process* [89] on a graph $(\mathcal{V}, \mathcal{E})$ is a spatiotemporal stochastic process $\langle y_{t,u} \rangle_{t \in \mathcal{T}, u \in \mathcal{V}}$ where $y_{t,u}$ depends only on $\{y_{t-1,v} : v \in \{u\} \cup \mathcal{N}_u\}$ for every node $u \in \mathcal{V}$ at every time $t > 0$. Hence, a diffusion process is necessarily a Markov process. A *snapshot* at a time $t \in \mathcal{T}$ is a vector $\boldsymbol{y}_t := (y_{t,u})_{u \in \mathcal{V}} \in \mathcal{X}^{\mathcal{V}}$ containing

all node states at the time t. A *diffusion history* (or a *history* for short) is a matrix $\boldsymbol{Y} := (\boldsymbol{y}_0, \dots, \boldsymbol{y}_T)^\mathsf{T} = (y_{t,u})_{t\in\mathcal{T}, u\in\mathcal{V}} \in \mathcal{X}^{\mathcal{T}\times\mathcal{V}}$ containing snapshots at all times. A history $\boldsymbol{Y}$ is said to be *feasible* iff it happens with nonzero probability.

2.1.2 Graph Diffusion Models. In this work, we focus on two classic graph diffusion models, namely the Susceptible–Infected (SI) model and the Susceptible–Infected–Recovered (SIR) model [48]. The SI model can be considered as a special case of the SIR model, so we start with the SIR model.

The SIR model describes the contagion process of infectious diseases from which recovery provides permanent protection against re-infection. In the SIR model, the states are $\mathcal{X} := \{\mathtt{S}, \mathtt{I}, \mathtt{R}\}$. The probability measure $P_{\boldsymbol{\beta}}$ for the SIR model is parameterized by two parameters $\boldsymbol{\beta} := (\beta^\mathtt{I}, \beta^\mathtt{R})^\mathsf{T} \in (0,1)^2$, where $\beta^\mathtt{I}$ and $\beta^\mathtt{R}$ are called the *infection rate* and the *recovery rate*, respectively. Let $n^{\mathcal{X}_0}(\boldsymbol{y}_t)$ denote the number of nodes with state in $\mathcal{X}_0$ at time t, e.g., $n^{\mathtt{IR}}(\boldsymbol{y}_t)$ meaning the number of infected or recovered nodes at time t. Please refer to Appendix A for the definition of $P_{\boldsymbol{\beta}}$ in the SIR model.

Let $\mathrm{supp}(P) := \{Y \in \mathcal{X}^{\mathcal{T}\times\mathcal{V}} : P_{\boldsymbol{\beta}}[Y] > 0\}$ denote the set of possible histories. For a snapshot $\boldsymbol{y}_T \in \mathcal{X}^{\mathcal{V}}$, let $P_{\boldsymbol{\beta}}|\boldsymbol{y}_T := P_{\boldsymbol{\beta}}[\,\cdot \mid \boldsymbol{y}_T]$ denote the posterior given the snapshot $\boldsymbol{y}_T$, and let $\mathrm{supp}(P \mid \boldsymbol{y}_T) := \{Y \in \mathcal{X}^{\mathcal{T}\times\mathcal{V}} : P_{\boldsymbol{\beta}}[Y \mid \boldsymbol{y}_T] > 0\}$ denote the set of possible histories consistent with the snapshot $\boldsymbol{y}_T$. Since the set of possible histories does not depend on $\boldsymbol{\beta}$, we omit $\boldsymbol{\beta}$ in its notation.

For the SI model, it is defined by letting $\beta^\mathtt{R} := 0$ and removing the state $\mathtt{R}$ from the SIR model. It describes the contagion process of infectious diseases that cannot recover.

2.2 Problem Statement

The problem we study is reconstructing the complete diffusion history Y from a single snapshot $\boldsymbol{y}_T$ without knowing true diffusion parameters $\boldsymbol{\beta}$. As is discussed above, it is often impractical to obtain true diffusion histories for supervised methods to learn from or for statistical methods to accurately estimate diffusion parameters from. Hence, we assume that neither a database of diffusion histories nor true diffusion parameters are known. Instead, what we know is the final snapshot $\boldsymbol{y}_T$. Since a single snapshot cannot provide any information about the underlying diffusion model, then we have to assume that the underlying diffusion model is known as domain knowledge while its diffusion parameters are not assumed to be known. This is a common assumption in previous work [16, 74]. Such domain knowledge is usually available in practice. For instance, if we know that recovery from a disease will probably provide lifelong protection (e.g., chickenpox), then it will be reasonable to assume that the underlying diffusion model is the SIR model while we do not know its diffusion parameters. In this work, we consider the SI model and the SIR model.

Given a timespan T of interest and the snapshot $\boldsymbol{y}_T$ at time T, our task is to reconstruct the diffusion history $\boldsymbol{y}_0, \dots, \boldsymbol{y}_{T-1}$. Since we assume no extra diffusion histories for training, our method is of unsupervised learning. This setting is more realistic than that of previous works (e.g., [16, 74]), where diffusion histories for training and/or true diffusion parameters are assumed to be available.

Under this realistic setting, the diffusion history reconstruction problem involves two aspects: (i) estimating diffusion parameters from a single snapshot and (ii) reconstructing the diffusion history in the presence of estimation error of diffusion parameters. As we will show later in Theorem 2, it is NP-hard to estimate diffusion parameters from a snapshot. Thus, diffusion parameter estimation itself is a non-trivial problem here. Furthermore, its NP-hardness implies that estimation error of diffusion parameters is unavoidable. Hence, it is important for the diffusion history reconstruction method to be stable against estimation error of diffusion parameters.

We assume that the source nodes are unknown, but the initial distribution $P[\boldsymbol{y}_0]$ is known as a domain knowledge, such as how many nodes (roughly) are initially infected, which areas the source nodes probably locate in, and whether high-density communities are more suitable for epidemics to break out. The knowledge of $P[\boldsymbol{y}_0]$ is necessary because without it the diffusion parameters will be uncertain in the unsupervised setting. For instance, given the timespan T and the snapshot $\boldsymbol{y}_T$, a smaller number of initially infected nodes suggests a higher infection rate, while a larger number of initially infected nodes implies a lower infection rate.

For computational consideration, the initial distribution should be efficiently computable up to a normalizing constant, i.e., $P[\boldsymbol{y}_0] \propto p(\boldsymbol{y}_0)$ for all possible $\boldsymbol{y}_0$ where $p : \mathcal{X}^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$ is an efficiently computable function. In this work, we define the initial distribution w.r.t. the number $n_0^\mathtt{I}$ of initially infected nodes. We assume no initially recovered nodes, because they could be removed from the graph. Thus, we define $P_{\boldsymbol{\beta}}[\boldsymbol{y}_0] \propto \exp(-\gamma|n^\mathtt{I}(\boldsymbol{y}_0) - n_0^\mathtt{I}| - \gamma n^\mathtt{R}(\boldsymbol{y}_0))$, where $\gamma > 0$ is a hyperparameter. If we are more certain on $n_0^\mathtt{I}$, we should use a larger γ. We do not consider $n_0^\mathtt{I}$ a hard constraint because it is typically a rough estimate rather than the exact number.

We formally state our problem definition as Problem 1 below. See Fig. 1 for an illustration of the DASH problem.

Problem 1 (DASH). *Under SI/SIR model, reconstruct the complete Diffusion history from A single SnapsHot without knowing true diffusion parameters.* ***Input:*** *(i) graph $(\mathcal{V}, \mathcal{E})$; (ii) timespan T of interest; (iii) final snapshot $\boldsymbol{y}_T \in \mathcal{X}^{\mathcal{V}}$; (iv) initial distribution $P[\boldsymbol{y}_0]$.* ***Output:*** *reconstructed complete diffusion history $\widehat{Y} \in \mathcal{X}^{\mathcal{T}\times\mathcal{V}}$.*

3 REVISITING DIFFUSION HISTORY MLE

In this section, we theoretically reveal a fundamental limitation of the MLE formulation for diffusion history reconstruction. In Section 3.1, we show that estimation error of diffusion parameters is unavoidable due to the NP-hardness of diffusion parameter estimation. Then in Section 3.2, we prove that the MLE formulation for diffusion history reconstruction is sensitive to estimation error of diffusion parameters. Therefore, the performance of the MLE formulation can be drastically degraded by such estimation error of diffusion parameters. Please refer to Appendix B for proofs.

3.1 NP-Hardness of Diffusion Parameter Estimation

In this subsection, we show that estimation error of diffusion parameters is unavoidable due to the NP-hardness of diffusion parameter estimation. To estimate diffusion parameters $\boldsymbol{\beta}$, a conventional approach is maximum likelihood estimation (MLE) [66]. Given the observed snapshot $\boldsymbol{y}_T$, diffusion parameter MLE is formulated as:

$$\max_{\widehat{\boldsymbol{\beta}}} P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]. \tag{1}$$

To optimize Eq. (1), one may consider using gradient-based methods. Typically, gradient-based methods first evaluate the likelihood function and then differentiate it to get the gradient. However, due to the explosive search space of possible histories, it is intractable to compute the likelihood $P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]$. In fact, we prove that computing the likelihood $P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]$ (or even approximating it) is already NP-hard, as is stated in Theorem 1.

Theorem 1 (NP-hardness of snapshot probability). *Under the SIR model, approximating the probability of a snapshot*[1] *is NP-hard, even if the initial probability $P[\boldsymbol{y}_0]$ (including its normalizing constant) for each possible $\boldsymbol{y}_0$ can be computed in polynomial time.*

Theorem 1 implies that there probably do not exist tractable algorithms to approximate the likelihood $P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]$, unless P = NP. The intuition behind Theorem 1 is that possible diffusion histories form an explosively large search space. Since gradient-based methods require computing the likelihood $P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]$, this diminishes the applicability of such methods for diffusion parameter MLE.

Although approximating the likelihood $P_{\widehat{\boldsymbol{\beta}}}[Y]$ is intractable, one may also wonder whether there exists an efficient algorithm that can give the optimal $\widehat{\boldsymbol{\beta}}$ *without* computing $P_{\widehat{\boldsymbol{\beta}}}[Y]$. Unfortunately, we prove that computing MLE diffusion parameters (even if a small relative error is allowed) is also NP-hard, as is stated in Theorem 2.

Theorem 2 (NP-hardness of diffusion parameter MLE). *Under the SIR model, diffusion parameter MLE*[2] *is NP-hard, even if the initial probability $P[\boldsymbol{y}_0]$ (up to a normalizing constant*[3]*) for each possible $\boldsymbol{y}_0$ can be computed in polynomial time.*

Both Theorem 1 and Theorem 2 suggest that there do not exist tractable algorithms to estimate diffusion parameters $\boldsymbol{\beta}$ accurately from a single snapshot $\boldsymbol{y}_T$, unless P = NP. Hence, estimation error of diffusion parameters is unavoidable in the DASH problem. Consequently, a good method for the DASH problem should be stable against such estimation error of diffusion parameters, which motivates us to utilize posterior expected hitting times in Section 4.2.

3.2 Sensitivity to Estimation Error of Diffusion Parameters

In this subsection, we reveal a fundamental limitation of the MLE formulation for DASH.The MLE formulation reconstructs the history Y by maximizing its likelihood $P_{\widehat{\boldsymbol{\beta}}}[Y]$ among all possible histories that are consistent with the observed history $\boldsymbol{y}_T$:

$$\max_{Y \in \operatorname{supp}(P|\boldsymbol{y}_T)} P_{\widehat{\boldsymbol{\beta}}}[Y]. \tag{2}$$

As is shown in Section 3.1, estimation error of diffusion parameters is unavoidable. Thus, it is crucial to analyze the sensitivity of the MLE formulation to such estimation error of diffusion parameters. We prove that unfortunately, the MLE formulation is sensitive to estimation error of diffusion parameters when diffusion parameters are small, as is stated in Theorem 3.

[1]See Problem 2 in Appendix B.1 for the precise definition.
[2]See Problem 3 in Appendix B.2 for the precise definition.
[3]The normalizing constant does not affect the result of this problem.

Theorem 3 (Sensitivity to estimation error of diffusion parameters). *Under the SIR model with small true $\boldsymbol{\beta}$ (i.e., $\boldsymbol{\beta} \searrow \mathbf{0}$), for every possible history Y, we have:*

$$\frac{\partial}{\partial \beta^{\mathrm{I}}} P_{\boldsymbol{\beta}}[Y] = \Theta\Big(\frac{1}{\beta^{\mathrm{I}}}\Big) P_{\boldsymbol{\beta}}[Y] \quad \text{if } n^{\mathrm{IR}}(\boldsymbol{y}_T) > n^{\mathrm{IR}}(\boldsymbol{y}_0); \tag{3}$$

$$\frac{\partial}{\partial \beta^{\mathrm{R}}} P_{\boldsymbol{\beta}}[Y] = \Theta\Big(\frac{1}{\beta^{\mathrm{R}}}\Big) P_{\boldsymbol{\beta}}[Y] \quad \text{if } n^{\mathrm{R}}(\boldsymbol{y}_T) > n^{\mathrm{R}}(\boldsymbol{y}_0). \tag{4}$$

Theorem 3 shows that the relative error of the likelihood induced by estimation error of diffusion parameters is inversely proportional to true diffusion parameters. The conditions $n^{\mathrm{IR}}(\boldsymbol{y}_T) > n^{\mathrm{IR}}(\boldsymbol{y}_0)$ and $n^{\mathrm{R}}(\boldsymbol{y}_T) > n^{\mathrm{R}}(\boldsymbol{y}_0)$ mean that infection and recovery do happen during the timespan T of interest, which is almost always the case in practice. Hence, the conditions are quite mild and realistic.

Infection and recovery rates in many real-world data are small [33, 62, 86]. Hence, the likelihood under estimated diffusion parameters $\widehat{\boldsymbol{\beta}}$ has a large relative error and is ill-conditioned. Moreover, since the error of the likelihood is proportional to the likelihood itself, the MLE history under *estimated* $\widehat{\boldsymbol{\beta}}$ may have larger decrease in likelihood than other histories and thus may not be the MLE history under *true* diffusion parameters. Therefore, sensitivity is indeed a fundamental limitation of the MLE formulation in practice.

To address this limitation, we instead solve the DASH problem from a new perspective and propose a so-called *barycenter formulation* utilizing posterior expected hitting times, which, as we will prove, is stable against estimation error of diffusion parameters.

4 PROPOSED METHOD: DITTO

In this section, we propose a method called *DIffusion hiTting Times with Optimal proposal* (DITTO) for solving the DASH problem. In Sec. 4.1, we employ *mean-field approximation* to estimate diffusion parameters. In Sec. 4.2, we propose the *barycenter formulation* that is provably stable, and reduce the DASH problem to estimating the posterior expected *hitting times*. In Sec. 4.3, we propose to use an unsupervised graph neural network to learn an optimal proposal in Metropolis–Hastings Markov chain Monte Carlo (M–H MCMC) algorithm to estimate the posterior expected *hitting times*. The overall procedure of DITTO is presented in Algorithm 1.

4.1 Mean-Field Approximation for Diffusion Parameter Estimation

Previous works [16, 74] assume diffusion parameters are known, but in our setting we have to estimate the unknown diffusion parameters $\boldsymbol{\beta}$. Theorem 2 shows it is intractable to estimate $\boldsymbol{\beta}$ via MLE. To develop a tractable estimator, we employ *mean-field approximation* [89] to compute the so-called *pseudolikelihood* for each node u at each time t. In mean-field approximation, the state $y_{t,u}$ is assumed to only depend on $y_{t-1,v}$ of neighbors $v \in \mathcal{N}_u$, but the dependence between $y_{t,u}$ and $y_{t,v}$ is ignored. Then, the joint pseudolikelihood factorizes into peudolikelihoods of each single node.

Let $\widehat{\boldsymbol{\beta}}$ denote the estimator of diffusion parameters $\boldsymbol{\beta}$. Let $f^{x}_{t,u;\widehat{\boldsymbol{\beta}}}$ denote the pseudolikelihood for node $u \in \mathcal{V}$ to be in state $x \in \mathcal{X}$ at time $t \in \mathcal{T}$. If we assume that the set of n_0^{I} initially infected nodes is uniformly drawn from all $\binom{n}{n_0^{\mathrm{I}}}$ possible sets, then the probability that a node is initially infected is $\binom{n-1}{n_0^{\mathrm{I}}-1}/\binom{n}{n_0^{\mathrm{I}}} = n_0^{\mathrm{I}}/n$. Thus, we set

Algorithm 1 Proposed method: DITTO

Input: (i) the graph $(\mathcal{V}, \mathcal{E})$; (ii) the timespan T of interest and the observed snapshot $\boldsymbol{y}_T$; (iii) the initial distribution $P[\boldsymbol{y}_0]$ and the (rough) number $n_0^{\mathtt{I}}$ of initial infections; (iv) the batch sizes K, L, the MCMC steps S, and the moving average hyperparameter η.

Output: the reconstructed diffusion history $\widehat{Y}$.

initialize the diffusion parameter estimates $\widehat{\boldsymbol{\beta}}$
while $\widehat{\boldsymbol{\beta}}$ not converged **do**
  initialize pseudolikelihoods $f^{\mathtt{S}}_{0,u;\widehat{\boldsymbol{\beta}}}, f^{\mathtt{I}}_{0,u;\widehat{\boldsymbol{\beta}}}, f^{\mathtt{R}}_{0,u;\widehat{\boldsymbol{\beta}}}$ for all $u \in \mathcal{V}$ by Eq. (5)
  for $t = 0, \dots, T-1$ **do**
    compute pseudolikelihoods $f^{\mathtt{S}}_{t+1,u;\widehat{\boldsymbol{\beta}}}, f^{\mathtt{I}}_{t+1,u;\widehat{\boldsymbol{\beta}}}, f^{\mathtt{R}}_{t+1,u;\widehat{\boldsymbol{\beta}}}$ for all $u \in \mathcal{V}$ by Eq.'s (6)(7)(8)
  end for
  update $\widehat{\boldsymbol{\beta}} \leftarrow \text{GradientDescent}\Big(-\frac{1}{n}\sum_{u\in\mathcal{V}} \log f^{y_{T,u}}_{T,u;\widehat{\boldsymbol{\beta}}}\Big)$
end while
initialize the proposal $Q_{\boldsymbol{\theta}}$
while $Q_{\boldsymbol{\theta}}$ not converged **do**
  sample K histories $Y^{(1)}, \dots, Y^{(K)} \sim P_{\widehat{\boldsymbol{\beta}}}$
  update $\boldsymbol{\theta} \leftarrow \text{GradientDescent}\Big(-\frac{1}{K}\sum_{i=1}^{K} \log Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T^{(i)})[Y^{(i)}]\Big)$
end while
sample L histories $Y^{(0,1)}, \dots, Y^{(0,L)} \sim Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$
initialize the hitting time estimates: for each $u \in \mathcal{V}$
$\widehat{h}^{\mathtt{I}}_u \leftarrow \frac{1}{L}\sum_{i=1}^{L} h^{\mathtt{I}}_u(Y^{(0,i)}), \quad \widehat{h}^{\mathtt{R}}_u \leftarrow \frac{1}{L}\sum_{i=1}^{L} h^{\mathtt{R}}_u(Y^{(0,i)})$
for $s = 1, \dots, S$ **do**
  sample L histories $\widetilde{Y}^{(s,1)}, \dots, \widetilde{Y}^{(s,L)} \sim Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$
  generate $\xi^{(s,1)}, \dots, \xi^{(s,L)} \sim \text{Uniform}[0,1)$
  update MCMC by the M–H rule: for each $i = 1, \dots, L$

$$Y^{(s,i)} \leftarrow \begin{cases} \widetilde{Y}^{(s,i)} & \text{if } \xi^{(s,i)} < \dfrac{P_{\widehat{\boldsymbol{\beta}}}[\widetilde{Y}^{(s,i)}]Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y^{(s-1,i)}]}{P_{\widehat{\boldsymbol{\beta}}}[Y^{(s-1,i)}]Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[\widetilde{Y}^{(s,i)}]} \\ Y^{(s-1,i)} & \text{otherwise} \end{cases}$$

  update the hitting time estimates: for each $u \in \mathcal{V}$
$\widehat{h}^{\mathtt{I}}_u \leftarrow \eta\widehat{h}^{\mathtt{I}}_u + \frac{1-\eta}{L}\sum_{i=1}^{L} h^{\mathtt{I}}_u(Y^{(s,i)}),\ \widehat{h}^{\mathtt{R}}_u \leftarrow \eta\widehat{h}^{\mathtt{R}}_u + \frac{1-\eta}{L}\sum_{i=1}^{L} h^{\mathtt{R}}_u(Y^{(s,i)})$
end for
reconstruct the diffusion history $\widehat{Y}$: for each $u \in \mathcal{V}$

$$\widehat{y}_{t,u} \leftarrow \begin{cases} \mathtt{S} & \text{for } 0 \le t < \text{round}(\widehat{h}^{\mathtt{I}}_u) \\ \mathtt{I} & \text{for } \text{round}(\widehat{h}^{\mathtt{I}}_u) \le t < \text{round}(\widehat{h}^{\mathtt{R}}_u) \\ \mathtt{R} & \text{for } \text{round}(\widehat{h}^{\mathtt{R}}_u) \le t \le T \end{cases}$$

return $\widehat{Y}$

the pseudolikelihoods at $t = 0$ as follows:

$$f^{\mathtt{S}}_{0,u;\widehat{\boldsymbol{\beta}}} := 1 - \frac{n_0^{\mathtt{I}}}{n}, \qquad f^{\mathtt{I}}_{0,u;\widehat{\boldsymbol{\beta}}} := \frac{n_0^{\mathtt{I}}}{n}, \qquad f^{\mathtt{R}}_{0,u;\widehat{\boldsymbol{\beta}}} := 0. \tag{5}$$

The pseudolikelihoods at time $1, \dots, T$ are computed inductively on t assuming that neighbors are independent. A node is susceptible at time $t+1$ iff it is susceptible at time t and is not infected by its infected neighbors at time $t+1$:

$$f^{\mathtt{S}}_{t+1,u;\widehat{\boldsymbol{\beta}}} := f^{\mathtt{S}}_{t,u;\widehat{\boldsymbol{\beta}}} \prod_{v\in\mathcal{N}_u} (1 - f^{\mathtt{I}}_{t,v;\widehat{\boldsymbol{\beta}}} \cdot \widehat{\beta}^{\mathtt{I}}). \tag{6}$$

A node is infected at time $t+1$ iff it is infected at time t, or it is susceptible at time t but is infected by one of its infected neighbors and does not recover immediately at time $t+1$:

$$f^{\mathtt{I}}_{t+1,u;\widehat{\boldsymbol{\beta}}} := \Big(f^{\mathtt{I}}_{t,u;\widehat{\boldsymbol{\beta}}} + f^{\mathtt{S}}_{t,u;\widehat{\boldsymbol{\beta}}}\Big(1 - \prod_{v\in\mathcal{N}_u}(1 - f^{\mathtt{I}}_{t,v;\widehat{\boldsymbol{\beta}}} \cdot \widehat{\beta}^{\mathtt{I}})\Big)\Big)(1 - \widehat{\beta}^{\mathtt{R}}). \tag{7}$$

A node is recovered at time $t+1$ iff it is recovered at time t, or it recovers just at time $t+1$:

$$f^{\mathtt{R}}_{t+1,u;\widehat{\boldsymbol{\beta}}} := f^{\mathtt{R}}_{t,u;\widehat{\boldsymbol{\beta}}} + \Big(f^{\mathtt{I}}_{t,u;\widehat{\boldsymbol{\beta}}} + f^{\mathtt{S}}_{t,u;\widehat{\boldsymbol{\beta}}}\Big(1 - \prod_{v\in\mathcal{N}_u}(1 - f^{\mathtt{I}}_{t,v;\widehat{\boldsymbol{\beta}}} \cdot \widehat{\beta}^{\mathtt{I}})\Big)\Big)\widehat{\beta}^{\mathtt{R}}. \tag{8}$$

Finally, we estimate $\widehat{\boldsymbol{\beta}}$ by maximizing the joint log-pseudolikelihood of the observed snapshot $\boldsymbol{y}_T$, which decomposes into the sum of log-pseudolikelihoods of each single node:

$$\max_{\widehat{\boldsymbol{\beta}}} \sum_{u\in\mathcal{V}} \log f^{y_{T,u}}_{T,u;\widehat{\boldsymbol{\beta}}}. \tag{9}$$

The objective Eq. (9) can be optimized by gradient descent methods. From now on, we let $\widehat{\boldsymbol{\beta}}$ denote the estimated diffusion parameters.

4.2 Barycenter Formulation with Provable Stability

Since diffusion parameter estimation is NP-hard by Theorem 2, it is impossible to avoid estimation error of diffusion parameters. Meanwhile, as is shown by Theorem 3, the MLE formulation for diffusion history reconstruction is sensitive to estimation error of diffusion parameters. To avoid this inherent limitation of the MLE formulation, we propose an alternative formulation that is stable against estimation error of diffusion parameters.

Since DASH is an ill-posed inverse problem, it is crucial to design an appropriate formulation with desired inductive bias. Here we propose a so-called *barycenter formulation* that can capture the desired information of the posterior distribution of histories and is provably stable against estimation error of diffusion parameters.

Consider the *hitting times* at which node states change. For a node u in a history Y, we define $h^{\mathtt{I}}_u(Y)$ and $h^{\mathtt{R}}_u(Y)$ to be the first time when the node u becomes infected/recovered, respectively:

$$h^{\mathtt{I}}_u(Y) := \min\big\{T+1,\ \min\{t \ge 0 : y_{t,u} = \mathtt{I} \text{ or } \mathtt{R}\}\big\}, \tag{10}$$

$$h^{\mathtt{R}}_u(Y) := \min\big\{T+1,\ \min\{t \ge 0 : y_{t,u} = \mathtt{R}\}\big\}. \tag{11}$$

Note that a node can become infected and recover at the same time, so the definition of $h^{\mathtt{I}}_u$ includes the case where $y_{t-1,u} = \mathtt{S}$, $y_{t,u} = \mathtt{R}$.

Our key theoretical result is: (unlike the likelihood $P_{\widehat{\boldsymbol{\beta}}}[Y],$) the posterior expected hitting times are stable against estimation error of diffusion parameters, as is stated in Theorem 4.

Theorem 4 (Stability against estimation error of diffusion parameters). *Under SIR model with small true $\boldsymbol{\beta}$ (i.e., $\boldsymbol{\beta} \searrow \mathbf{0}$), if $n^{\mathtt{I}}(\boldsymbol{y}_0)$ and $n^{\mathtt{R}}(\boldsymbol{y}_0)$ are fixed, then for any possible snapshot $\boldsymbol{y}_T$,*

$$\nabla_{\boldsymbol{\beta}} \mathop{\mathbb{E}}_{Y\sim P_{\boldsymbol{\beta}}|\boldsymbol{y}_T} [h^{\mathtt{I}}_u(Y)] = \mathrm{O}(1), \quad \nabla_{\boldsymbol{\beta}} \mathop{\mathbb{E}}_{Y\sim P_{\boldsymbol{\beta}}|\boldsymbol{y}_T} [h^{\mathtt{R}}_u(Y)] = \mathrm{O}(1). \tag{12}$$

In stark contrast with the MLE formulation, Theorem 4 shows posterior expected hitting times are stable even when $\boldsymbol{\beta}$ is close to zero. Such stability guarantee motivates us to utilize hitting times to design an objective function that is stable against estimation error of diffusion parameters.

Our idea is to define a distance metric d for histories based on the hitting times, and find a history $\widehat{Y}$ that is close to all possible histories w.r.t. the distance metric d:

$$\min_{\widehat{Y}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T} [d(\widehat{Y}, Y)^2]. \tag{13}$$

We call it the *barycenter formulation*, because the optimal history $\widehat{Y}$ for this formulation is the barycenter of the posterior distribution $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$ w.r.t. the distance metric d. We define the distance metric d as the Euclidean distance with hitting times as coordinates:

$$d(\widehat{Y}, Y) := \sqrt{\sum_{u\in\mathcal{V}} \big((h_u^{\mathtt{I}}(\widehat{Y}) - h_u^{\mathtt{I}}(Y))^2 + (h_u^{\mathtt{R}}(\widehat{Y}) - h_u^{\mathtt{R}}(Y))^2\big)}. \tag{14}$$

Then, our barycenter formulation instantiates as:

$$\begin{aligned}&\min_{\widehat{Y}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}\Big[\sum_{u\in\mathcal{V}}\big((h_u^{\mathtt{I}}(\widehat{Y})-h_u^{\mathtt{I}}(Y))^2 + (h_u^{\mathtt{R}}(\widehat{Y})-h_u^{\mathtt{R}}(Y))^2\big)\Big]\\ =&\min_{\widehat{Y}} \sum_{u\in\mathcal{V}}\Big(\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[(h_u^{\mathtt{I}}(\widehat{Y})-h_u^{\mathtt{I}}(Y))^2] + \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[(h_u^{\mathtt{R}}(\widehat{Y})-h_u^{\mathtt{R}}(Y))^2]\Big).\end{aligned} \tag{15}$$

According to bias–variance decomposition, we can further decompose Eq. (15) for $x = \mathtt{I}, \mathtt{R}$ as:

$$\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[(h_u^x(\widehat{Y}) - h_u^x(Y))^2] = \Big(h_u^x(\widehat{Y}) - \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^x(Y)]\Big)^2 + \mathop{\mathrm{Var}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^x(Y)]. \tag{16}$$

Since the variances are constant w.r.t. the history estimator $\widehat{Y}$, Eq. (15) is thus equivalent to minimizing the squared biases:

$$\min_{\widehat{Y}} \sum_{u\in\mathcal{V}}\Big(\Big(h_u^{\mathtt{I}}(\widehat{Y}) - \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^{\mathtt{I}}(Y)]\Big)^2 + \Big(h_u^{\mathtt{R}}(\widehat{Y}) - \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^{\mathtt{R}}(Y)]\Big)^2\Big). \tag{17}$$

Therefore, the optimal estimates are simply rounding each expected hitting time to the nearest integer:

$$h_u^x(\widehat{Y}) := \mathrm{round}\Big(\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^x(Y)]\Big), \qquad x = \mathtt{I}, \mathtt{R}. \tag{18}$$

Now our problem reduces to estimating the expected hitting times over the posterior $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$. Owing to the stability of expected hitting times, the optimal estimates $h_u^{\mathtt{I}}(\widehat{Y})$ and $h_u^{\mathtt{R}}(\widehat{Y})$ are also stable against estimation error of diffusion parameters. Finally, we reconstruct the history $\widehat{Y}$ according to the estimated hitting times in Eq. (18):

$$\widehat{y}_{t,u} := \begin{cases} \mathtt{S}, & \text{for } 0 \le t < h_u^{\mathtt{I}}(\widehat{Y}); \\ \mathtt{I}, & \text{for } h_u^{\mathtt{I}}(\widehat{Y}) \le t < h_u^{\mathtt{R}}(\widehat{Y}); \\ \mathtt{R}, & \text{for } h_u^{\mathtt{R}}(\widehat{Y}) \le t \le T. \end{cases} \tag{19}$$

4.3 Metropolis–Hastings MCMC for Posterior Expectation Estimation

So far we have reduced our problem to estimating the posterior expected hitting times $\mathbb{E}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^{\mathtt{I}}(Y)]$ and $\mathbb{E}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^{\mathtt{R}}(Y)]$. However, due to the explosive search space of possible histories, it is intractable to compute the posterior probability $P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T]$, as is proven in Theorem 1. Therefore, it is non-trivial to design Monte Carlo samplers to estimate the posterior expectation.

To tackle this difficulty, we employ the Metropolis–Hastings Markov chain Monte Carlo (M–H MCMC) algorithm [40, 60] to estimate posterior expectation. The basic idea of M–H MCMC is to construct a Markov chain whose stationary distribution is the desired posterior distribution. This algorithm requires a so-called *proposal* distribution over possible histories. In our method, we design a proposal that differs for different $\boldsymbol{y}_T$, so we write it as $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[\cdot]$. Let $\mathrm{supp}(Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)) := \{Y \in \mathcal{X}^{\mathcal{T}\times\mathcal{V}} : Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] > 0\}$ denote the set of histories that can be generated from $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$. In each step of M–H MCMC, we sample a new history $\widetilde{Y} \sim Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$. Let Y denote the current history in MCMC. Then according to the M–H rule, the new history $\widetilde{Y}$ is accepted with probability

$$\min\Big\{1, \frac{P_{\widehat{\boldsymbol{\beta}}}[\widetilde{Y}]Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[\widetilde{Y}]}\Big\}. \tag{20}$$

This defines a Markov chain of histories. After a sufficient number of steps, this Markov chain provably converges to the desired posterior distribution $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$ [40]. The convergence rate of MCMC depends critically on the design of the proposal $Q_{\boldsymbol{\theta}}$. If $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$ approximates $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$ better, then the rate of convergence will be higher [59]. Since hand-craft proposals may fail to approximate the posterior distribution and thus adversely affect the convergence rate, we propose to use a graph neural network (GNN) to learn an optimal proposal. The backbone of $Q_{\boldsymbol{\theta}}$ is an Anisotropic GNN with edge gating mechanism [7, 44, 65]. The GNN takes the observed snapshot $\boldsymbol{y}_T$ as input and predicts a proposal $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$ corresponding to $\boldsymbol{y}_T$. The neural architecture of $Q_{\boldsymbol{\theta}}$ is detailed in Appendix C.1. We want $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$ to approximate $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$, so we adopt the expected squared difference of their log-likelihoods as the objective function:

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}[(\log Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T])^2]. \tag{21}$$

However, it is intractable to compute $P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T]$, so we cannot implement this objective function directly. To address this, we derive an equivalent objective (Theorem 5) that is tractable to evaluate.

Theorem 5 (An equivalent objective). *If the GNN $Q_{\boldsymbol{\theta}}$ is sufficiently expressive and has the same set of possible histories as the posterior (i.e., $\mathrm{supp}(Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)) = \mathrm{supp}(P \mid \boldsymbol{y}_T)$ for any snapshot $\boldsymbol{y}_T$), then the original objective Eq. (21) is equivalent to*

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big], \tag{22}$$

for any strictly convex *function $\psi : \mathbb{R}_+ \to \mathbb{R}$.*

Here, the intractable term $P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T]$ in Eq. (21) is replaced with a tractable term $P_{\widehat{\boldsymbol{\beta}}}[Y]$. In this work, we use $\psi(w) := -\log w$, and the objective Eq. (22) instantiates as

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[-\log\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{23}$$

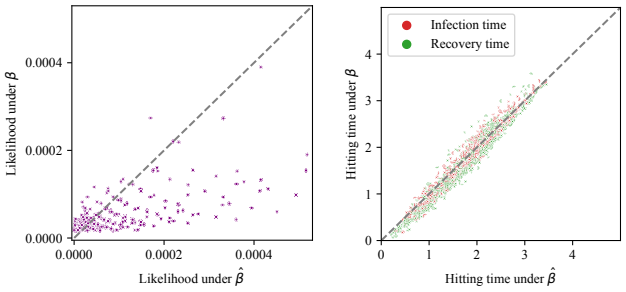
$$\iff \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}[-\log Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] + \log P_{\widehat{\boldsymbol{\beta}}}[Y]] \tag{24}$$

$$\iff \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}[-\log Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]]. \tag{25}$$

We train the GNN $Q_{\boldsymbol{\theta}}$ using the objective Eq. (25). Notably, since DITTO does not require real diffusion histories as training data, it does not suffer from the scarcity of training data in practice. After training, we use this GNN as the proposal in the M–H MCMC algorithm to estimate the posterior expected hitting times for diffusion

Table 2: Summary of datasets.

Dataset	#Nodes	#Edges	Timespan	Graph	Diffusion
BA	1,000	3,984	10	Synthetic	Synthetic
ER	1,000	3,987	10	Synthetic	Synthetic
Oregon2	11,461	32,730	15	Real	Synthetic
Prost	15,810	38,540	15	Real	Synthetic
BrFarmers	82	230	16	Real	Real SI
Pol	18,470	48,053	40	Real	Real SI
Covid	344	2,044	10	Real	Real SIR
Hebrew	3,521	18,064	9	Real	Real SIR



(a) $P_{\boldsymbol{\beta}}[Y]$ **vs** $P_{\widehat{\boldsymbol{\beta}}}[Y]$ **in the MLE formulation.** **(b)** $\mathbb{E}_{P_{\boldsymbol{\beta}}|\boldsymbol{y}_T}[h_u^x(Y)]$ **vs** $\mathbb{E}_{P_{\widehat{\boldsymbol{\beta}}}|\boldsymbol{y}_T}[h_u^x(Y)]$ **in the barycenter formulation.**

Figure 2: Sensitivity of the MLE formulation vs stability of the barycenter formulation.

history reconstruction. The sampling scheme of the proposal $Q_{\boldsymbol{\theta}}$ is detailed in Appendix C.2, which is designed to satisfy the condition $\text{supp}(Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)) = \text{supp}(P \mid \boldsymbol{y}_T)$ in Theorem 5.

4.4 Complexity Analysis

Proposition 6. *(i) The time complexity of each iteration of diffusion parameter estimation is* $O(T(n+m))$. *(ii) The time complexity to sample a history from the proposal is* $O(T(n \log n + m))$.

According to Proposition 6, if we optimize $\widehat{\boldsymbol{\beta}}$ for I iterations, optimize $Q_{\boldsymbol{\theta}}$ for J iterations with K samples per iteration, and run MCMC for S iterations with L samples per iteration, then the overall time complexity of DITTO is $O(T(n+m)I + T(n \log n + m)(JK + SL))$. If hyperparameters are considered as constants, then the overall time complexity $O(T(n \log n + m))$ is nearly linear w.r.t. the output size $\Theta(Tn)$ and the input size $\Theta(n+m)$.

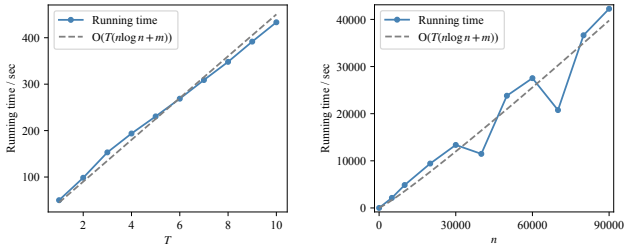
5 EXPERIMENTS

We conduct extensive experiments on both synthetic and real-world datasets to answer the following research questions:

RQ1: What is the quality of estimated diffusion parameters $\widehat{\boldsymbol{\beta}}$?
RQ2: How does DITTO perform for real-world diffusion?
RQ3: How does DITTO compare to MLE-based methods?
RQ4: How stable is DITTO against estimation error of $\widehat{\boldsymbol{\beta}}$?
RQ5: How is the scalability of DITTO?
RQ6: How does the performance of DITTO vary with timespan?
RQ7: In M–H MCMC, how does our learned proposal $Q_{\boldsymbol{\theta}}$ compare to a random proposal?

5.1 Experimental Setting

5.1.1 Datasets. We use 3 types of datasets. **(D1) Synthetic graphs and diffusion:** Barabási–Albert (BA) [4] and Erdős–Rényi (ER) [25]



(a) Running time vs T. **(b) Running time vs** n.

Figure 3: Running time (training time + testing time).

random graphs with synthetic SI and SIR diffusion. **(D2) Synthetic diffusion on real graphs:** Oregon2 [55] and Prost [68] with synthetic SI and SIR diffusion. **(D3) Real diffusion on real graphs:** BrFarmers [69, 83] and Pol [20] with SI-like real diffusion, and Covid and Hebrew [5] with SIR-like real diffusion. Please refer to Appendix D.1 for detailed description of datasets.

5.1.2 Baselines. We consider 2 types of baselines. **(B1) Supervised methods for time series imputation:** DASH can be alternatively formulated as time series imputation, so we compare DITTO with latest imputation methods including GCN [49], [93], BRITS [9], GRIN [18], and SPIN [58]. **(B2) MLE-based methods for diffusion history reconstruction:** DHREC [74] and CRI [17]. Please refer to Appendix D.2 for description of baselines.

5.1.3 Evaluation Metrics. To measure the similarity between the true history and the reconstructed history, we use the macro F1 score (F1; the higher, the better) and the normalized rooted mean squared error (NRMSE; the lower, the better) of hitting times, where

$$\text{NRMSE}(\boldsymbol{Y}, \widehat{\boldsymbol{Y}}) := \sqrt{\frac{\sum_{u \in \mathcal{V}} ((h_u^{\text{I}}(Y) - h_u^{\text{I}}(\widehat{Y}))^2 + (h_u^{\text{R}}(Y) - h_u^{\text{R}}(\widehat{Y}))^2)}{2n(T+1)^2}}. \quad (26)$$

In Sec. 5.4, we also use the performance gap to the ideal performance as a metric. The smaller gap, the better. Let s and s^* denote the actual performance and the ideal performance, respectively. For F1, $\text{Gap}(s, s^*) := (s^* - s)/s^*$. For NRMSE, $\text{Gap}(s, s^*) := (s - s^*)/s^*$.

5.1.4 Reproducibility. Please refer to Appendix D.3.

5.2 Quality of Estimated Diffusion Parameters

To answer RQ1, we compare the performance of supervised methods under true $\boldsymbol{\beta}$ with their performance under estimated $\widehat{\boldsymbol{\beta}}$. We use two strongest imputation methods GRIN and SPIN. Since true diffusion parameters of real diffusion are not available, we only use datasets D1 and D2 in this experiment. Results are shown in Table 3. Whether trained with true $\boldsymbol{\beta}$ or estimated $\widehat{\boldsymbol{\beta}}$, the performance has no significant difference. Results suggest that mean-field approximation is sufficient to estimate diffusion parameters accurately, and the estimated diffusion parameters can help supervised methods achieve strong performance when the diffusion model is known.

5.3 Performance for Real-World Diffusion

For real-world diffusion, the diffusion model is not exactly SI/SIR, and true diffusion parameters are unknown. Hence, it is important to test how DITTO generalizes from SI/SIR to real-world diffusion. To answer RQ2 and RQ3, we comprehensively compare DITTO with

Table 3: Comparison between estimated $\widehat{\beta}$ and true β to justify the mean-field approximation. "OOM" indicates "out of memory."

Method	Training	BA-SI		ER-SI		Oregon2-SI		Prost-SI		BA-SIR		ER-SIR		Oregon2-SIR		Prost-SIR	
		F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓
GRIN	w/ true β	.8404	.2123	.8317	.2166	.8320	.2249	.8482	.2155	.7867	.1692	.7626	.2484	.8024	.1651	.8067	.1652
	w/ estimated $\widehat{\beta}$	.8456	.2071	.8324	.2160	.8370	.2199	.8504	.2128	.7833	.1717	.7757	.1939	.8030	.1633	.8068	.1644
SPIN	w/ true β	.8414	.2117	.8310	.2167	OOM		OOM		.7832	.1663	.7647	.2321	OOM		OOM	
	w/ estimated $\widehat{\beta}$	.8477	.2047	.8315	.2170					.7869	.1611	.7800	.1909				

Table 4: Results for real-world diffusion. "OOM" indicates "out of memory."

Type	Method	BrFarmers		Pol		Covid		Hebrew	
		F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓	F1↑	NRMSE↓
Supervised (w/ estimated $\widehat{\beta}$)	GCN	.5409	.6660	.4458	.4946	.3162	.5214	.3350	.6070
	GIN	.4548	.6565	.5203	.4767	.3226	.4951	.3704	.7816
	BRITS	.5207	.3995	OOM		.3524	.5333	.3120	.6584
	GRIN	.8003	.2425	.6518	.3731	.5448	.3040	.5916	**.2212**
	SPIN	**.8268**	**.2084**	OOM		.5917	.2932	.5178	.3330
MLE	DHREC	.6131	.4150	.7023	.3398	.3540	.6023	.6251	.4169
	CRI	.6058	.4444	.7468	.2942	.4170	.5487	.5344	.3552
Barycenter	DITTO (ours)	.8206	.2142	**.7471**	**.2903**	**.6240**	**.2637**	**.6411**	.2983

Table 5: Comparison with MLE-based methods on synthetic SI and SIR diffusion. *We use GRIN trained with true β as the ideal performance and calculate *Gap* w.r.t. this ideal performance.

Type	Method	BA-SI				ER-SI				Oregon2-SI				Prost-SI			
		F1↑	Gap↓	NRMSE↓	Gap↓	F1↑	Gap↓	NRMSE↓	Gap↓	F1↑	Gap↓	NRMSE↓	Gap↓	F1↑	Gap↓	NRMSE↓	Gap↓
Ideal	GRIN	.8404*	—	.2123*	—	.8317*	—	.2166*	—	.8320*	—	.2249*	—	.8482*	—	.2155*	—
MLE	DHREC	.6026	28.30%	.4644	118.75%	.6281	24.48%	.4495	107.53%	.6038	27.43%	.4101	82.35%	.6558	22.68%	.4138	92.02%
	CRI	.7502	10.73%	.3012	41.87%	.7797	6.25%	.2744	26.69%	.8183	1.65%	.2438	8.40%	.8083	4.70%	.2491	15.59%
Barycenter	DITTO (ours)	**.8384**	**0.24%**	**.2139**	**0.75%**	**.8269**	**0.58%**	**.2225**	**2.72%**	**.8280**	**0.48%**	**.2289**	**1.78%**	**.8327**	**1.83%**	**.2317**	**7.52%**
Type	**Method**	BA-SIR				ER-SIR				Oregon2-SIR				Prost-SIR			
		F1↑	Gap↓	NRMSE↓	Gap↓	F1↑	Gap↓	NRMSE↓	Gap↓	F1↑	Gap↓	NRMSE↓	Gap↓	F1↑	Gap↓	NRMSE↓	Gap↓
Ideal	GRIN	.7867*	—	.1692*	—	.7626*	—	.2484*	—	.8024*	—	.1651*	—	.8067*	—	.1652*	—
MLE	DHREC	.5080	35.43%	.4722	179.08%	.5500	27.88%	.4423	78.06%	.6044	24.68%	.4478	171.23%	.6268	22.30%	.4326	161.86%
	CRI	.5994	23.81%	.3356	98.35%	.6129	19.63%	.3109	25.16%	.5761	28.20%	.3576	116.60%	.5738	28.87%	.3406	106.17%
Barycenter	DITTO (ours)	**.7783**	**1.07%**	**.1633**	**−3.49%**	**.7734**	**−1.42%**	**.1679**	**−32.41%**	**.7928**	**1.20%**	**.1707**	**3.39%**	**.7929**	**1.71%**	**.1690**	**2.30%**

(B1) supervised methods for time series imputation and (B2) MLE-based methods for diffusion history reconstruction on real-world diffusion in D3. Since true diffusion parameters of real diffusion are not available, we estimate diffusion parameters by DITTO. For B1, we use estimated diffusion parameters to generate training data. For B2, we feed estimated diffusion parameters to MLE-based methods.

The results for real diffusion are shown in Table 4. DITTO generalizes well to real diffusion and consistently achieves strong performance on all datasets. For instance, DITTO is 10.06% better in NRMSE than the best baseline for the Covid dataset. In contrast, the performance of supervised methods degrades drastically when real diffusion deviates from SI/SIR models. This is because the training data generated by the SI/SIR model follow a different distribution from the real diffusion. Only for BrFarmers do supervised methods achieve good performance, because the diffusion in BrFarmers is very close to the SI model [83]. For MLE-based methods, their performance varies largely across datasets. This is because real diffusion may not be close to the SI/SIR model, so the likelihood as their objective function may fail.

5.4 Comparison with MLE-Based Methods

To further answer RQ3, we compare DITTO to MLE-based methods also with synthetic diffusion on both synthetic graphs in D1 and real graphs in D2. Note that here we do not directly compare with supervised methods for the following reasons. (1) Table 3 shows that if supervised methods know the diffusion model of test data, then they can generate training data that follow the same distribution as the test data. Thus, they are expected to perform well. (2) Meanwhile, Table 4 shows the superiority of supervised methods comes only from knowing the underlying diffusion model (including its true parameters), which is almost impossible in practice due to the scarcity of training data. As long as they have no access to the true diffusion model, their performance drop drastically. Therefore, it is meaningless to compare with supervised methods for synthetic diffusion. Instead, we train GRIN with true diffusion parameters and use its results as the *ideal* performance. Then for MLE-based methods and DITTO, we compare their performance gaps to this ideal performance. We do not use SPIN here because it has similar performance with GRIN, while SPIN is out of memory on D2.

The results are shown in Table 5. DITTO consistently achieves the strongest performance and significantly outperforms state-of-the-art MLE-based methods for all datasets. Notably, DITTO even outperforms GRIN for BA-SIR and ER-SIR. In contrast, the performance of MLE-based methods vary largely due to their instability to estimation error of diffusion parameters. For instance, DITTO has only 1.07% gap in F1 for BA-SIR, while MLE-based methods have at least 23.81% gap in F1. Results demonstrate the superior performance of DITTO over state-of-the-art MLE-based methods.

5.5 Additional Experiments

5.5.1 Stability against Estimation Error of Diffusion Parameters. To answer RQ4 and demonstrate the stability of our barycenter

formulation, we visualize the likelihoods of histories and the posterior expected hitting times under true and estimated diffusion parameters. Since the number of possible histories under the SIR model is roughly $O(\binom{T+3}{2}^n)$, it is intractable to compute them on large graphs. Thus, we use a graph with $n = 6$ and $T = 4$ so that likelihoods and posterior expected hitting times can be computed exactly. We visualize them under the SIR model with $\boldsymbol{\beta} = (0.3, 0.2)^\mathsf{T}$ and $\widehat{\boldsymbol{\beta}} = (0.2, 0.3)^\mathsf{T}$. Fig. 2 displays the results. Fig. 2a shows that the likelihoods of histories change drastically in the presence of parameter estimation error. In contrast, Fig. 2b shows that posterior expected hitting times are almost identical under $\boldsymbol{\beta}$ and $\widehat{\boldsymbol{\beta}}$. Therefore, our barycenter formulation is more stable than the MLE formulation against estimation error of diffusion parameters, which agrees with our theoretical analyses in Theorem 3 and Theorem 4.

5.5.2 Scalability. To answer RQ5, we evaluate the scalability of DITTO by varying T and n. We generate BA graphs with attachment 4 to obtain scale-free networks with various sizes. Fig. 3a shows running times under $n = 1,000$ and $T = 1, \ldots, 10$, and Fig. 3b shows running times under $T = 10$ and n up to 90k. Results demonstrate that the running times of DITTO scale near-linearly w.r.t. T and n, which agrees with its time complexity $O(T(n \log n + m))$.

5.5.3 Effect of Timespan & Ablation Study. In Appendix E, we answer RQ6 and RQ7. Appendix E.1 compares DITTO and MLE-based methods under various timespans. It demonstrates that DITTO can better handle the higher uncertainty induced by larger timespan than MLE-based methods. Appendix E.2 is an ablation study on the effect of the number of training steps. It shows that the learned proposal performs better than untrained proposal.

6 RELATED WORK

Diffusion on graphs are deterministic or stochastic processes where information or entities on nodes transmit through edges [2, 37, 46, 48, 53, 72, 81, 92]. In this section, we review related work on graph diffusion, which can be grouped into forward and inverse problems.

Forward problems on graph diffusion. The vast majority of research on diffusion or dynamic graphs [29, 30] are devoted to forward problems. Pioneering works derive epidemic thresholds for random graphs from probabilistic perspectives [6] or for arbitrary graphs from spectral perspectives [31, 63, 85, 89]. Later, observational studies investigate influence patterns of diffusion processes [8, 34, 39, 54]. On the algorithmic side, researchers have made tremendous effort to diffusion-related optimization problems, such as influence maximization [13–15, 21, 34, 36, 46, 67] and node immunization [41, 47, 64, 80, 82]. Recently, differential equations of graph diffusion has also been applied to the design of graph convolutional networks to alleviate oversmoothing [10, 12, 50, 79, 90, 94].

Inverse problems on graph diffusion. Compared with forward problems on graph diffusion, the inverse problems are in general more difficult due to the challenge of ill-posedness. The inverse problems split into two categories by whether diffusion histories are known. In the one category where diffusion histories are known, the problems are relatively more tractable because the search space is smaller. These problems include estimating diffusion parameters [32, 35, 38, 61, 77, 99], recovering graph topology [32, 35, 38, 61, 99], and inferring diffusion paths [1, 22, 26, 71, 77, 78].

The other category where diffusion histories are unknown is much less studied. In this category, the problems are often harder because the number of possible histories is explosively large. Among them, most works focus on the *diffusion source localization* problem. Only recently has research emerged on the even harder problem *diffusion history reconstruction. (1) Diffusion source localization.* The source localization problem aims to find the source nodes of diffusion. Early works focus on designing localization algorithms based on graph theory and network science [27, 28, 52, 75, 76, 91, 95–98]. These methods may not generalize well to various diffusion models. Later works propose data-driven methods that utilize graph neural networks to learn to identify sources from data [23, 56, 87]. *(2) Diffusion history reconstruction.* Compared with source localization, diffusion history reconstruction is even harder because the search space of possible histories is larger. Existing methods for diffusion history reconstruction are exclusively based on the MLE formulation, including DHREC [74], CRI [17], and SSR [16]. These methods assume that true diffusion parameters [16, 74] and/or partial diffusion histories are known [16], or cannot reconstruct a complete diffusion history [17]. Meanwhile, diffusion history reconstruction can be alternatively formulated as a time series imputation problem. State-of-the-art methods include BRITS [9] for multivariate time series, and GRIN [18] and SPIN [58] for graph time series. They are all supervised and thus suffer from the scarcity of training data of real diffusion histories. Furthermore, since the true diffusion model is unknown for real-world diffusion, it is difficult to synthesize training data that follow the same distribution as the true diffusion model. Therefore, they have limited applicability in practice.

7 CONCLUSION

In this work, we have studied a challenging problem: reconstructing diffusion history from a single snapshot. To address the sensitivity of the MLE formulation, we have proposed a barycenter formulation that is provably stable against the estimation error of diffusion parameters. We have further developed an effective solver named DITTO for the barycenter formulation, which is based on Metropolis–Hastings MCMC with a learned optimal proposal. Our method is unsupervised, which is desirable in practice due to the scarcity of training data. Extensive experiments have shown that DITTO consistently achieve strong performance for both synthetic and real-world diffusion.

ACKNOWLEDGMENTS

This work was supported in part by NSF (1947135 and 2134079), the NSF Program on Fairness in AI in collaboration with Amazon (1939725), DARPA (HR001121C0165), NIFA (2020-67021-32799), DHS (17STQAC00001-06-00), ARO (W911NF2110088), C3.ai Digital Transformation Institute, and IBM-Illinois Discovery Accelerator Institute. The work of Lei Ying was supported in part by NSF (2134081). The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Bruno Abrahao, Flavio Chierichetti, Robert Kleinberg, and Alessandro Panconesi. 2013. Trace complexity of network inference. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 491–499.

[2] Joan L Aron and Ira B Schwartz. 1984. Seasonality and period-doubling bifurcations in an epidemic model. *Journal of Theoretical Biology* 110, 4 (1984), 665–679.

[3] Andrea Avena-Koenigsberger, Bratislav Misic, and Olaf Sporns. 2018. Communication dynamics in complex brain networks. *Nature Reviews Neuroscience* 19, 1 (2018), 17–33.

[4] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

[5] Alon Bartal, Nava Pliskin, and Oren Tsur. 2020. Local/Global contagion of viral/non-viral information: Analysis of contagion spread in online social networks. *PLOS One* 15, 4 (2020), e0230811.

[6] Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. 1992. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy* 100, 5 (1992), 992–1026.

[7] Xavier Bresson and Thomas Laurent. 2018. An experimental study of neural networks for variable graphs. In *Workshop of the 6th International Conference on Learning Representations*.

[8] Linda Briesemeister, Patrick Lincoln, and Phillip Porras. 2003. Epidemic profiles and defense of scale-free networks. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*. 67–75.

[9] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. BRITS: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, Vol. 31.

[10] Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. 2021. Grand: Graph neural diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139. PMLR, 1407–1418.

[11] Chen Chen, Hanghang Tong, B Aditya Prakash, Charalampos E Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. 2015. Node immunization on large graphs: Theory and algorithms. *IEEE Transactions on Knowledge and Data Engineering* 28, 1 (2015), 113–126.

[12] Qi Chen, Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. 2022. Optimization-induced graph implicit nonlinear diffusion. In *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162. PMLR, 3648–3661.

[13] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincon, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. 2011. Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 379–390.

[14] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1029–1038.

[15] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 199–208.

[16] Zhen Chen, Hanghang Tong, and Lei Ying. 2019. Inferring full diffusion history from partial timestamps. *IEEE Transactions on Knowledge and Data Engineering* 32, 7 (2019), 1378–1392.

[17] Zhen Chen, Kai Zhu, and Lei Ying. 2016. Detecting multiple information sources in networks under the SIR model. *IEEE Transactions on Network Science and Engineering* 3, 1 (2016), 17–31.

[18] Andrea Cini, Ivan Marisca, and Cesare Alippi. 2022. Filling the G_ap_s: Multivariate time series imputation by graph neural networks. In *Proceedings of the Tenth International Conference on Learning Representations*.

[19] James Coleman, Elihu Katz, and Herbert Menzel. 1957. The diffusion of an innovation among physicians. *Sociometry* 20, 4 (1957), 253–270.

[20] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. 2011. Political polarization on Twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 5. 89–96.

[21] Samik Datta, Anirban Majumder, and Nisheeth Shrivastava. 2010. Viral marketing for multiple products. In *Proceedings of the 2010 IEEE International Conference on Data Mining*. IEEE, 118–127.

[22] Quinlan E Dawkins, Tianxi Li, and Haifeng Xu. 2021. Diffusion source identification on networks with statistical confidence. In *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139. PMLR, 2500–2509.

[23] Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. 2019. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 569–578.

[24] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks* 107 (2018), 3–11.

[25] Paul Erdős and Alfréd Rényi. 1959. On random graphs I. *Publicationes Mathematicae Debrecen* 6 (1959), 290–297.

[26] David Fajardo and Lauren M Gardner. 2013. Inferring contagion patterns in social contact networks with limited infection data. *Networks and Spatial Economics* 13, 4 (2013), 399–426.

[27] Mehrdad Farajtabar, Manuel Gomez Rodriguez, Mohammad Zamani, Nan Du, Hongyuan Zha, and Le Song. 2015. Back to the past: Source identification in diffusion networks from partially observed cascades. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, Vol. 38. PMLR, 232–240.

[28] Soheil Feizi, Muriel Médard, Gerald Quon, Manolis Kellis, and Ken Duffy. 2018. Network infusion to infer information sources in networks. *IEEE Transactions on Network Science and Engineering* 6, 3 (2018), 402–417.

[29] Dongqi Fu, Liri Fang, Ross Maciejewski, Vetle I. Torvik, and Jingrui He. 2022. Meta-learned metrics over multi-evolution temporal graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Washington DC, USA, 367–377.

[30] Dongqi Fu and Jingrui He. 2021. SDG: A simplified and dynamic graph neural network. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2273–2277.

[31] Ayalvadi Ganesh, Laurent Massoulié, and Don Towsley. 2005. The effect of network topology on the spread of epidemics. In *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2. IEEE, 1455–1466.

[32] Lauren M Gardner, David Fajardo, and S Travis Waller. 2014. Inferring contagion patterns in social contact networks using a maximum likelihood approach. *Natural Hazards Review* 15, 3 (2014), 04014004.

[33] KL Goh and Shu-Dong Xiao. 2009. Inflammatory bowel disease: A survey of the epidemiology in Asia. *Journal of Digestive Diseases* 10, 1 (2009), 1–6.

[34] Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters* 12, 3 (2001), 211–223.

[35] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. 2012. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data* 5, 4 (2012), 1–37.

[36] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2011. A Data-Based Approach to Social Influence Maximization. *Proceedings of the VLDB Endowment* 5, 1 (2011).

[37] Mark Granovetter. 1978. Threshold models of collective behavior. *American Journal of Sociology* 83, 6 (1978), 1420–1443.

[38] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. 2004. Information diffusion through blogspace. In *Proceedings of the 13th International Conference on World Wide Web*. 491–501.

[39] Habiba Habiba and Tanya Berger-Wolf. 2011. Working for influence: Effect of network density and modularity on diffusion in networks. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 933–940.

[40] Wilfred Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109.

[41] Yukio Hayashi, Masato Minoura, and Jun Matsukubo. 2004. Oscillatory epidemic prevalence in growing scale-free networks. *Physical Review E* 69, 1 (2004), 016112.

[42] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37. PMLR, 448–456.

[43] Baoyu Jing, Hanghang Tong, and Yada Zhu. 2021. Network of tensor time series. In *Proceedings of the Web Conference 2021*. 2425–2437.

[44] Chaitanya K Joshi, Quentin Cappart, Louis-Martin Rousseau, and Thomas Laurent. 2020. Learning TSP requires rethinking generalization. arXiv:2006.07054 (2020).

[45] Richard M Karp. 1972. Reducibility among combinatorial problems. *Complexity of Computer Computations* (1972), 85–103.

[46] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 137–146.

[47] David Kempe, Jon Kleinberg, and Éva Tardos. 2005. Influential nodes in a diffusion model for social networks. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*. Springer, 1127–1138.

[48] William Ogilvy Kermack and Anderson Gray McKendrick. 1927. A contribution to the mathematical theory of epidemics. In *Proceedings of the Royal Society of London, Series A, Containing Papers of a Mathematical and Physical Character*, Vol. 115. The Royal Society London, 700–721.

[49] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=SJU4ayYgl

[50] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. arXiv:1911.05485 (2019).

[51] Alden S Klovdahl. 1985. Social networks and the spread of infectious diseases: The AIDS example. *Social Science & Medicine* 21, 11 (1985), 1203–1216.

[52] Theodoros Lappas, Evimaria Terzi, Dimitrios Gunopulos, and Heikki Mannila. 2010. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1059–1068.

[53] Deokjae Lee, Wonjun Choi, Janos Kértesz, and Byungnam Kahng. 2017. Universal mechanism for hybrid percolation transitions. *Scientific Reports* 7, 1 (2017), 1–7.

[54] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web* 1, 1 (2007), 5–es.

[55] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. 177–187.

[56] Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. 2022. Source localization of graph diffusion via variational autoencoders for graph inverse problems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1010–1020.

[57] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

[58] Ivan Marisca, Andrea Cini, and Cesare Alippi. 2022. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. In *Advances in Neural Information Processing Systems*.

[59] Kerrie L Mengersen and Richard L Tweedie. 1996. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics* 24, 1 (1996), 101–121.

[60] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092.

[61] Seth Myers and Jure Leskovec. 2010. On the convexity of latent social network inference. In *Advances in Neural Information Processing Systems*, Vol. 23.

[62] Richard J O'Brien, Lawrence J Geiter, and Dixie E Snider Jr. 1987. The epidemiology of nontuberculous mycobacterial diseases in the United States: results from a national survey. *American Review of Respiratory Disease* 135, 5 (1987), 1007–1014.

[63] B Aditya Prakash, Deepayan Chakrabarti, Nicholas C Valler, Michalis Faloutsos, and Christos Faloutsos. 2012. Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowledge and Information Systems* 33, 3 (2012), 549–575.

[64] B Aditya Prakash, Hanghang Tong, Nicholas Valler, Michalis Faloutsos, and Christos Faloutsos. 2010. Virus propagation on time-varying networks: Theory and immunization algorithms. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 99–114.

[65] Ruizhong Qiu, Zhiqing Sun, and Yiming Yang. 2022. DIMES: A differentiable meta solver for combinatorial optimization problems. In *Advances in Neural Information Processing Systems*.

[66] Roger Ratcliff and Francis Tuerlinckx. 2002. Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review* 9, 3 (2002), 438.

[67] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 61–70.

[68] Luis EC Rocha, Fredrik Liljeros, and Petter Holme. 2011. Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. *PLoS Computational Biology* 7, 3 (2011), e1001109.

[69] EM Rogers, JR Ascroft, and N Röling. 1970. *Diffusion of innovations in Brazil, Nigeria, and India*. Technical Report. Michigan State University, East Lansing, MI, USA.

[70] Ryan Rossi and Nesreen Ahmed. 2015. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[71] Polina Rozenshtein, Aristides Gionis, B Aditya Prakash, and Jilles Vreeken. 2016. Reconstructing an epidemic over time. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1835–1844.

[72] Zhongyuan Ruan, Gerardo Iniguez, Márton Karsai, and János Kertész. 2015. Kinetics of social contagion. *Physical Review Letters* 115, 21 (2015), 218702.

[73] Thomas C Schelling. 1978. *Micromotives and Macrobehavior*. W.W. Norton & Co.

[74] Emre Sefer and Carl Kingsford. 2016. Diffusion archeology for diffusion progression history reconstruction. *Knowledge and Information Systems* 49, 2 (2016), 403–427.

[75] Devavrat Shah and Tauhid Zaman. 2011. Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory* 57, 8 (2011), 5163–5181.

[76] Devavrat Shah and Tauhid Zaman. 2012. Rumor centrality: A universal source detector. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*. 199–210.

[77] Yichen Song, Aiping Li, Jiuming Huang, Yong Quan, and Lu Deng. 2017. History path reconstruction analysis of topic diffusion on microblog. In *Proceedings of the International Conference on Intelligent and Interactive Systems and Applications*. Springer, 150–157.

[78] Yanchao Sun, Cong Qian, Ning Yang, and S Yu Philip. 2017. Collaborative inference of coexisting information diffusions. In *Proceedings of the 2017 IEEE International Conference on Data Mining*. IEEE, 1093–1098.

[79] Matthew Thorpe, Hedi Xia, Tan Nguyen, Thomas Strohmer, Andrea Bertozzi, Stanley Osher, and Bao Wang. 2022. GRAND++: Graph neural diffusion with a source term. In *Proceedings of the International Conference on Learning Representations*.

[80] Hanghang Tong, B Aditya Prakash, Charalampos Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. 2010. On the vulnerability of large graphs. In *Proceedings of the 2010 IEEE International Conference on Data Mining*. IEEE, 1091–1096.

[81] János Török and János Kertész. 2017. Cascading collapse of online social networks. *Scientific Reports* 7, 1 (2017), 1–8.

[82] Gaurav Tuli, Chris J Kuhlman, Madhav V Marathe, S Ravi, and Daniel J Rosenkrantz. 2012. Blocking complex contagions using community structure. In *Proceedings of the Workshop on Multiagent Interaction Networks*. CiteSeer.

[83] T W Valente. 1995. *Network Models of the Diffusion of Innovations (2nd Edition)*. Hampton Press, Cresskill, NJ.

[84] Thomas W Valente. 2005. Network models and methods for studying the diffusion of innovations. *Models and Methods in Social Network Analysis* 28 (2005), 98–116.

[85] Nicholas C Valler, B Aditya Prakash, Hanghang Tong, Michalis Faloutsos, and Christos Faloutsos. 2011. Epidemic spread in mobile ad hoc networks: Determining the tipping point. In *Proceedings of the International Conference on Research in Networking*. Springer, 266–280.

[86] Mark PJ Vanderpump. 2011. The epidemiology of thyroid disease. *British Medical Bulletin* 99, 1 (2011).

[87] Junxiang Wang, Junji Jiang, and Liang Zhao. 2022. An Invertible Graph Diffusion Neural Network for Source Localization. In *Proceedings of the ACM Web Conference 2022*. 1058–1069.

[88] Pu Wang, Marta C González, César A Hidalgo, and Albert-László Barabási. 2009. Understanding the spreading patterns of mobile phone viruses. *Science* 324, 5930 (2009), 1071–1076.

[89] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. 2003. Epidemic spreading in real networks: An eigenvalue viewpoint. In *Proceedings of the 22nd International Symposium on Reliable Distributed Systems*. IEEE, 25–34.

[90] Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. 2021. Dissecting the diffusion process in linear graph convolutional networks. In *Advances in Neural Information Processing Systems*, Vol. 34. 5758–5769.

[91] Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. 2017. Multiple source detection without knowing the underlying propagation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[92] Duncan J Watts. 2002. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences* 99, 9 (2002), 5766–5771.

[93] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks?. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ryGs6iA5Km

[94] Jialin Zhao, Yuxiao Dong, Ming Ding, Evgeny Kharlamov, and Jie Tang. 2021. Adaptive Diffusion in Graph Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 34. 23321–23333.

[95] Kai Zhu, Zhen Chen, and Lei Ying. 2016. Locating the contagion source in networks with partial timestamps. *Data Mining and Knowledge Discovery* 30, 5 (2016), 1217–1248.

[96] Kai Zhu, Zhen Chen, and Lei Ying. 2017. Catch'em all: Locating multiple diffusion sources in networks with partial observations. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.

[97] Kai Zhu and Lei Ying. 2014. Information source detection in the SIR model: A sample-path-based approach. *IEEE/ACM Transactions on Networking* 24, 1 (2014), 408–421.

[98] Kai Zhu and Lei Ying. 2016. Information source detection in networks: Possibility and impossibility results. In *Proceedings of IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.

[99] Bo Zong, Yinghui Wu, Ambuj K Singh, and Xifeng Yan. 2012. Inferring the underlying structure of information cascades. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*. IEEE, 1218–1223.

CONTENTS

A PRELIMINARIES ON THE SIR MODEL

In this section, we introduce the detailed definition of the SIR diffusion model. According to the Markov property, the probability of a history Y can be factorized in the temporal order:

$$P_{\boldsymbol{\beta}}[Y] = P[\boldsymbol{y}_0] \prod_{t=0}^{T-1} P_{\boldsymbol{\beta}}[\boldsymbol{y}_{t+1} \mid \boldsymbol{y}_t]. \tag{27}$$

The initial distribution $P[\boldsymbol{y}_0]$ is not defined in the SIR model, so it does not depend on diffusion parameters $\boldsymbol{\beta}$. For the transition probabilities $P_{\boldsymbol{\beta}}[\boldsymbol{y}_{t+1} \mid \boldsymbol{y}_t]$, they can be further factorized into the transition probability of each single node because every node is assumed to be independent with other nodes at the same time:

$$P_{\boldsymbol{\beta}}[\boldsymbol{y}_{t+1} \mid \boldsymbol{y}_t] = \prod_{u \in \mathcal{V}} P_{\boldsymbol{\beta}}[y_{t+1,u} \mid \boldsymbol{y}_t]. \tag{28}$$

If a node u is susceptible at time $t+1$, then it has to be susceptible at time t, and all its infected neighbors failed to infect it:

$$P_{\boldsymbol{\beta}}[y_{t+1,u} = \mathtt{S} \mid \boldsymbol{y}_t] := \begin{cases} \prod\limits_{v \in \mathcal{N}_u \wedge y_{t,v} = \mathtt{I}} (1-\beta^{\mathtt{I}}), & \text{if } y_{t,u} = \mathtt{S}; \\ 0, & \text{if } y_{t,u} = \mathtt{I} \text{ or } \mathtt{R}. \end{cases} \tag{29}$$

If a node u is infected at time $t+1$, then either it is infected by its infected neighbors and does not recover immediately, or it is already infected and has not recovered yet:

$$P_{\boldsymbol{\beta}}[y_{t+1,u} = \mathtt{I} \mid \boldsymbol{y}_t] := \begin{cases} \left(1 - \prod\limits_{v \in \mathcal{N}_u \wedge y_{t,v} = \mathtt{I}} (1-\beta^{\mathtt{I}})\right)(1-\beta^{\mathtt{R}}), & \text{if } y_{t,u} = \mathtt{S}; \\ 1-\beta^{\mathtt{R}}, & \text{if } y_{t,u} = \mathtt{I}; \\ 0, & \text{if } y_{t,u} = \mathtt{R}. \end{cases} \tag{30}$$

If a node u is recovered at time $t+1$, then either it recovers just at time $t+1$, or it is already recovered previously:

$$P_{\boldsymbol{\beta}}[y_{t+1,u} = \mathtt{R} \mid \boldsymbol{y}_t] := \begin{cases} \left(1 - \prod\limits_{v \in \mathcal{N}_u \wedge y_{t,v} = \mathtt{I}} (1-\beta^{\mathtt{I}})\right)\beta^{\mathtt{R}}, & \text{if } y_{t,u} = \mathtt{S}; \\ \beta^{\mathtt{R}}, & \text{if } y_{t,u} = \mathtt{I}; \\ 1, & \text{if } y_{t,u} = \mathtt{R}. \end{cases} \tag{31}$$

B PROOFS

B.1 Proof of Theorem 1

The precise definition of approximating the probability of a snapshot is stated in Problem 2.

Problem 2 (Approximating the probability of a snapshot). *Under the SIR model, given a graph $(\mathcal{V}, \mathcal{E})$, diffusion parameters $\boldsymbol{\beta}$, a timespan T, a snapshot $\boldsymbol{y}_T$, an initial distribution $P[\boldsymbol{y}_0]$, and a relative error tolerance $0 < \epsilon < 1$, find a number p such that*

$$(1-\epsilon)P_{\boldsymbol{\beta}}[\boldsymbol{y}_T] < p < (1+\epsilon)P_{\boldsymbol{\beta}}[\boldsymbol{y}_T]. \tag{32}$$

Now we prove Theorem 1.

Proof of Theorem 1. By reduction from the Minimum Dominating Set (MDS) problem. Suppose that we are to find the minimum dominating set of a graph $(\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = n$. We will construct an instance of Problem 2 that can be utilized to solve the MDS problem.

The graph for Problem 2 is the same graph $(\mathcal{V}, \mathcal{E})$. We choose the diffusion parameters $\beta^{\mathtt{I}} := 1$ and $\beta^{\mathtt{R}} := 0$, and choose the timespan $T = 1$. We consider the snapshot $\boldsymbol{y}_1$ to be $y_{1,u} := \mathtt{I}$ for all nodes $u \in \mathcal{V}$. Pick a relative error tolerance $0 < \epsilon < 1$ arbitrarily. Initially, we define every node to be independently infected with probability $\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}$:

$$P[\boldsymbol{y}_0] := \left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n^{\mathtt{I}}(\boldsymbol{y}_0)} \left(1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-n^{\mathtt{I}}(\boldsymbol{y}_0)}. \tag{33}$$

Then run the oracle for Problem 2 to get the output number p, which satisfies

$$(1-\epsilon)P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] < p < (1+\epsilon)P_{\boldsymbol{\beta}}[\boldsymbol{y}_1]. \tag{34}$$

We claim that the minimum dominating set is of size s iff the output p satisfies

$$\frac{(1-\epsilon)\left(\frac{1+\epsilon}{1-\epsilon}2^n\right)^{n-s}}{\left(1+\frac{1+\epsilon}{1-\epsilon}2^n\right)^n} < p < \frac{(1-\epsilon)\left(\frac{1+\epsilon}{1-\epsilon}2^n\right)^{n-s+1}}{\left(1+\frac{1+\epsilon}{1-\epsilon}2^n\right)^n}. \tag{35}$$

Since the intervals in Eq. (35) have no overlap for different s, then we can uniquely determine the minimum size s from the output p.

To prove the claim, note that $\beta^{\mathtt{I}} := 1$ implies that $P_{\boldsymbol{\beta}}[\boldsymbol{y}_1 \mid \boldsymbol{y}_0] > 0$ iff the infected nodes in $\boldsymbol{y}_0$ is a dominating set. Let s denote the size of the minimum dominating set and c_k denote the number of dominating sets of size k. Hence,

$$P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] = \sum_{k=s}^{n} c_k \left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^k \left(1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-k}. \tag{36}$$

Since s is the size of the minimum dominate set, then we have $c_s \geq 1$. Thus,

$$P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] \geq c_s \left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s \left(1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s} \tag{37}$$

$$\geq \left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s \left(1 - \frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s}. \tag{38}$$

Hence,

$$p > (1-\epsilon)P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] \tag{39}$$

$$\geq (1-\epsilon)\left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s\left(1-\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s} \tag{40}$$

$$= \frac{(1-\epsilon)\left(\frac{1+\epsilon}{1-\epsilon}2^n\right)^{n-s}}{\left(1+\frac{1+\epsilon}{1-\epsilon}2^n\right)^n}. \tag{41}$$

Furthermore, since $c_k \leq \binom{n}{k}$ and $\left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)\left(1-\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{-1} = \frac{1}{\frac{1+\epsilon}{1-\epsilon}2^n} < 1$, then:

$$P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] \leq \sum_{k=s}^{n}\binom{n}{k}\left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s\left(1-\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s} \tag{42}$$

$$\leq \left(\sum_{k=s}^{n}\binom{n}{k}\right)\left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s\left(1-\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s} \tag{43}$$

$$\leq \left(\sum_{k=0}^{n}\binom{n}{k}\right)\left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s\left(1-\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s} \tag{44}$$

$$= 2^n\left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s\left(1-\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s}. \tag{45}$$

Hence,

$$p < (1+\epsilon)P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] \tag{46}$$

$$\leq (1+\epsilon)\cdot 2^n\left(\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^s\left(1-\frac{1}{1+\frac{1+\epsilon}{1-\epsilon}2^n}\right)^{n-s} \tag{47}$$

$$= \frac{(1-\epsilon)\left(\frac{1+\epsilon}{1-\epsilon}2^n\right)^{n-s+1}}{\left(1+\frac{1+\epsilon}{1-\epsilon}2^n\right)^n}. \tag{48}$$

Combining Eq. (41) and Eq. (48) yields our claim Eq. (35).

The numbers involved can be stored in $\mathrm{poly}\left(n, \log\frac{1}{\epsilon}\right)$ bits and be computed using high-precision arithmetics within $\mathrm{poly}\left(n, \log\frac{1}{\epsilon}\right)$ time. Therefore, this gives a polynomial-time reduction from the MDS problem to Problem 2, so the NP-hardness of the MDS problem [45] implies that Problem 2 is NP-hard. □

B.2 Proof of Theorem 2

The precise definition of diffusion parameter MLE is stated in Problem 3.

Problem 3 (Diffusion parameter MLE). *Under the SIR model, given a graph $(\mathcal{V}, \mathcal{E})$, a timespan T, a snapshot $\boldsymbol{y}_T$, an initial distribution $P[\boldsymbol{y}_0]$, and a relative error tolerance $0 < \epsilon < 1$, find $\widehat{\boldsymbol{\beta}}$ where*

$$\exists \boldsymbol{\beta} \in \operatorname*{argmax}_{\boldsymbol{\beta}} P_{\boldsymbol{\beta}}[\boldsymbol{y}_T] : (1-\epsilon)\boldsymbol{\beta} < \widehat{\boldsymbol{\beta}} < (1+\epsilon)\boldsymbol{\beta}. \tag{49}$$

Before proving Theorem 2, we give a technical lemma.

Lemma 7. *For $r \geq 1$, $c > 0$, and $0 \leq x \leq \frac{c}{r^2}$,*

$$(1+x)^r \leq 1 + \left(r + \frac{\mathrm{e}^c - 1}{2}\right)x. \tag{50}$$

In particular, for $r \geq 1$ and $0 \leq x \leq \frac{1}{r^2}$,

$$(1+x)^r \leq 1 + \left(r + \frac{\mathrm{e} - 1}{2}\right)x \leq 1 + (r+1)x. \tag{51}$$

Proof. Define an auxiliary function:

$$\phi(z) := 1 + cz + \frac{\mathrm{e}^c - 1}{2}cz^2 - \mathrm{e}^{cz}. \tag{52}$$

Its first order derivative $\phi'(z) = c + (\mathrm{e}^c - 1)cz - c\mathrm{e}^{cz}$ is concave, so for every $0 \leq z \leq 1$,

$$\phi'(z) \geq \min\{\phi'(0), \phi'(1)\} = \min\{0, 0\} = 0. \tag{53}$$

This suggests that $\phi(z)$ is increasing over $0 \leq z \leq 1$. Therefore, since $1 + \left(r + \frac{\mathrm{e}^c-1}{2}\right)x - (1+x)^r$ is concave w.r.t. $x \geq 0$ for $r \geq 1$, then for every $0 \leq x \leq \frac{c}{r^2}$,

$$1 + \left(r + \frac{\mathrm{e}^c - 1}{2}\right)x - (1+x)^r \tag{54}$$

$$\geq \min\left\{1 + \left(r + \frac{\mathrm{e}^c - 1}{2}\right)0 - (1+0)^r,\right. \tag{55}$$

$$\left. 1 + \left(r + \frac{\mathrm{e}^c - 1}{2}\right)\frac{c}{r^2} - \left(1 + \frac{c}{r^2}\right)^r\right\} \tag{56}$$

$$= \min\left\{0,\ 1 + \frac{c}{r} + \frac{\mathrm{e}^c - 1}{2}\frac{c}{r^2} - \left(1 + \frac{c}{r^2}\right)^r\right\} \tag{57}$$

$$\geq \min\left\{0,\ 1 + \frac{c}{r} + \frac{\mathrm{e}^c - 1}{2}\frac{c}{r^2} - (\mathrm{e}^{c/r^2})^r\right\} \tag{58}$$

$$= \min\left\{0,\ \phi\left(\frac{1}{r}\right)\right\} \geq \min\{0,\ \phi(0)\} = \min\{0, 0\} = 0. \tag{59}$$

□

Now we are ready to prove Theorem 2.

Proof of Theorem 2. By reduction from the Minimum Dominating Set (MDS) problem. Suppose that we are to find the minimum dominating set of a graph $(\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = n$ and $\mathcal{E}$ contains no self-loops. If $|\mathcal{E}| = 0$, then the only dominating set is $\mathcal{V}$. Thus, we can assume that $|\mathcal{E}| \geq 1$ from now on, which implies $n \geq 2$ and that the size of the minimum dominating set is at most $n-1$. We will construct an instance of Problem 3 that can be utilized to solve the MDS problem.

We create two auxiliary vertices $\mathcal{V}' := \{a, b\}$ and create edges between them and all nodes in $\mathcal{V}$, i.e., the graph for Problem 3 is $(\mathcal{V} \cup \mathcal{V}',\ \mathcal{E} \cup (\mathcal{V} \times \mathcal{V}'))$. We choose timespan $T = 1$. We consider the snapshot $\boldsymbol{y}_1$ to be $y_{1,a} = \mathtt{S}$, $y_{1,b} = \mathtt{R}$, and $y_{1,u} = \mathtt{R}$ for all nodes $u \in \mathcal{V}$. We choose the relative error tolerance

$$\epsilon := \min\left\{\frac{1}{16(n+1)n8^n - 1},\ \frac{\frac{1}{\sqrt[n]{2}} - \frac{1}{\sqrt[n-1]{2}}\left(1 + \frac{1}{n2^n}\right)^{\frac{1}{n-1}}}{1 - \frac{1}{\sqrt[n-1]{2}}\left(1 + \frac{1}{n2^n}\right)^{\frac{1}{n-1}}}\right\}. \tag{60}$$

We define the initial distribution as

$$P[\boldsymbol{y}_0] \propto \frac{1}{n^{\mathtt{I}}(\boldsymbol{y}_0)}\left(\frac{1}{2n4^n}\right)^{n^{\mathtt{I}}(\boldsymbol{y}_0)} \tag{61}$$

iff $y_{0,a} = y_{0,b} = \mathtt{S}$, $\{u \in \mathcal{V} : y_{0,u} = \mathtt{I}\}$ is a dominating set of $(\mathcal{V}, \mathcal{E})$, and $y_{0,u} = \mathtt{R}$ for all other nodes; otherwise, $P[\boldsymbol{y}_0] := 0$. Then run the oracle for Problem 3 to get diffusion parameter estimates $\widehat{\boldsymbol{\beta}} = [\widehat{\beta}^{\mathtt{I}}, \widehat{\beta}^{\mathtt{R}}]^{\mathsf{T}}$. We claim that the minimum dominating set is of size s iff the output $\widehat{\beta}^{\mathtt{I}}$ satisfies

$$1 - \frac{1}{\sqrt[s+1]{2}} < \widehat{\beta}^{\mathtt{I}} < 1 - \frac{1}{\sqrt[s]{2}}. \tag{62}$$

Since the intervals in Eq. (62) have no overlap for different s, then we can uniquely determine the minimum size s from the output $\widehat{\boldsymbol{\beta}}$.

To prove the claim, note that initially infected nodes fail to infect a but succeed in infecting b. Let s denote the size of the minimum dominating set, and $c_k \le \binom{n}{k}$ denote the number of dominating sets of size k. Ignoring the normalizing constant of $P[\boldsymbol{y}_0]$, we have:

$$P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] \propto \sum_{k=s}^{n} c_k \cdot \frac{1}{k}\Big(\frac{1}{2n4^n}\Big)^k (1-\beta^{\mathrm{I}})^k (1-(1-\beta^{\mathrm{I}})^k)(\beta^{\mathrm{R}})^{k+1}. \tag{63}$$

Thus, $\widehat{\beta^{\mathrm{R}}} = 1$ is the maximizer for $P_{\boldsymbol{\beta}}[\boldsymbol{y}_1]$. Plugging this into $P_{\boldsymbol{\beta}}[\boldsymbol{y}_1]$ gives

$$P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] \propto \sum_{k=s}^{n} c_k \cdot \frac{1}{k}\Big(\frac{1}{2n4^n}\Big)^k (1-\beta^{\mathrm{I}})^k (1-(1-\beta^{\mathrm{I}})^k). \tag{64}$$

To simplify notation, we change the variable to $\alpha := 1-\beta^{\mathrm{I}}$. Then, $P_{\boldsymbol{\beta}}[\boldsymbol{y}_1] \propto p(\alpha)$ where

$$p(\alpha) := \sum_{k=s}^{n} c_k \cdot \frac{1}{k}\Big(\frac{1}{2n4^n}\Big)^k \alpha^k (1-\alpha^k). \tag{65}$$

By calculus, its first order derivative is

$$p'(\alpha) = \sum_{k=s}^{n} c_k \Big(\frac{1}{2n4^n}\Big)^k \alpha^{k-1}(1-2\alpha^k), \tag{66}$$

and then its second order derivative is

$$p''(\alpha) = \sum_{k=s}^{n} c_k \Big(\frac{1}{2n4^n}\Big)^k \alpha^{k-2}(k-1-2(2k-1)\alpha^k). \tag{67}$$

Adding a node to a dominating set always yields a dominating set, so we have $c_k \ge 1$ for all $s \le k \le n$. Since $s \le n-1$, then for each $0 \le \alpha \le 1/\sqrt[s]{2}$,

$$\begin{aligned} p'(\alpha) &\ge \Big(\frac{1}{2n4^n}\Big)^s \alpha^{s-1}\Big(1-2\Big(\frac{1}{\sqrt[s]{2}}\Big)^s\Big) \\ &\quad + \sum_{k=s+1}^{n} \Big(\frac{1}{2n4^n}\Big)^k \alpha^{k-1}\Big(1-2\Big(\frac{1}{\sqrt[s]{2}}\Big)^k\Big) && (68)\\ &= 0 + \sum_{k=s+1}^{n} \Big(\frac{1}{2n4^n}\Big)^k \alpha^{k-1}\Big(1-\frac{2}{2^{k/s}}\Big) > 0. && (69)\end{aligned}$$

Let κ_α denote the minimum integer k such that $k-1-2(2k-1)\alpha^k > 0$. For each $1/\sqrt[s]{2} \le \alpha \le 1$, note that $\kappa_\alpha \ge s+1$ and $s-1-2(2s-1)(1/\sqrt[s]{2})^s = -s$, so we have:

$$\begin{aligned} p''(\alpha) &= \sum_{k=s}^{\kappa_\alpha - 1} c_k \Big(\frac{1}{2n4^n}\Big)^k \alpha^{k-2}(k-1-2(2k-1)\alpha^k) \\ &\quad + \sum_{k=\kappa_\alpha}^{n} c_k \Big(\frac{1}{2n4^n}\Big)^k \alpha^{k-2}(k-1-2(2k-1)\alpha^k) && (70)\\ &\le \Big(\frac{1}{2n4^n}\Big)^s \alpha^{s-2}\Big(s-1-2(2s-1)\Big(\frac{1}{\sqrt[s]{2}}\Big)^s\Big) \\ &\quad + \sum_{k=\kappa_\alpha}^{n} \binom{n}{k}\Big(\frac{1}{2n4^n}\Big)^{s+1} \alpha^{k-2} n && (71)\\ &= \alpha^{s-2}\Big(\frac{1}{2n4^n}\Big)^s \Big(-s + \frac{1}{2\cdot 4^n}\sum_{k=\kappa_\alpha}^{n}\binom{n}{k}\alpha^{k-s}\Big) && (72)\\ &\le \alpha^{s-2}\Big(\frac{1}{2n4^n}\Big)^s \Big(-s + \frac{1}{2\cdot 4^n}\sum_{k=\kappa_\alpha}^{n}\binom{n}{k}\cdot 1^{k-s}\Big) && (73)\\ &\le \alpha^{s-2}\Big(\frac{1}{2n4^n}\Big)^s \Big(-s + \frac{1}{2\cdot 4^n}\cdot 2^n\Big) && (74)\\ &\le \alpha^{s-2}\Big(\frac{1}{2n4^n}\Big)^s \Big(-1+\frac{1}{4}\Big) < 0. && (75)\end{aligned}$$

This implies $p'(\alpha)$ is strictly decreasing over $1/\sqrt[s]{2} \le \alpha \le 1$. Combining with Eq. (69), we know that $p(\alpha)$ is strictly unimodal over $0 \le \alpha \le 1$. Furthermore, since $c_k \ge 1$ for all $s \le k \le n$, then:

$$p'(1) = -\sum_{k=s}^{n} c_k\Big(\frac{1}{2n4^n}\Big)^k \le -\sum_{k=s}^{n}\Big(\frac{1}{2n4^n}\Big)^k < 0. \tag{76}$$

Hence, the minimizer β^{I} is the unique solution to $p'(1-\beta^{\mathrm{I}}) = 0$ over $0 < \beta^{\mathrm{I}} < 1 - 1/\sqrt[s]{2}$.

Next, we give tighter bounds for β^{I} to prove the claim Eq. (62). Let

$$\alpha_+ := \frac{1}{\sqrt[s]{2}}\Big(1 + \frac{1-\frac{1}{\sqrt[s]{2}}}{8(s+2)n4^n\binom{n}{s}}\Big) > \frac{1}{\sqrt[s]{2}}. \tag{77}$$

Note that $1-2\alpha_+^k > 0$ for all $k \ge s+1$, because:

$$\begin{aligned} \alpha_+ &< \frac{1}{\sqrt[s]{2}}\Big(1+\frac{1-0}{8(s+2)(s+1)4^0}\Big) && (78)\\ &< \frac{1}{\sqrt[s]{2}}\Big(1+\frac{\log 2}{s(s+1)}\Big) \le \frac{1}{\sqrt[s]{2}}\exp\Big(\frac{\log 2}{s(s+1)}\Big) = \frac{1}{\sqrt[s+1]{2}}. && (79)\end{aligned}$$

Since $1 \le s \le n-1$, and $1 \le c_k \le \binom{n}{k}$ for $s \le k \le n$, then by Lemma 7,

$$\begin{aligned} p'(\alpha_+) &\ge -\binom{n}{s}\Big(\frac{1}{2n4^n}\Big)^s \alpha_+^{s-1}(2\alpha_+^s - 1) \\ &\quad + \Big(\frac{1}{2n4^n}\Big)^{s+1}\alpha_+^s(1-2\alpha_+^{s+1}) && (80)\\ &= \frac{\alpha_+^{s-1}}{(2n4^n)^s}\Big(-\binom{n}{s}(2\alpha_+^s-1) + \frac{\alpha_+(1-2\alpha_+^{s+1})}{2n4^n}\Big) && (81)\\ &= \frac{\alpha_+^{s-1}}{(2n4^n)^s}\Big(-\binom{n}{s}\Big(\Big(1+\frac{1-\frac{1}{\sqrt[s]{2}}}{8(s+2)n4^n\binom{n}{s}}\Big)^s - 1\Big) \\ &\quad + \frac{\alpha_+\Big(1-\frac{1}{\sqrt[s]{2}}\Big(1+\frac{1-\frac{1}{\sqrt[s]{2}}}{8(s+2)n4^n\binom{n}{s}}\Big)^{s+1}\Big)\Big)}{2n4^n} && (82)\\ &\ge \frac{\alpha_+^{s-1}}{(2n4^n)^s}\Big(-\binom{n}{s}\Big(\Big(1+\frac{(s+1)\big(1-\frac{1}{\sqrt[s]{2}}\big)}{8(s+2)n4^n\binom{n}{s}}\Big)-1\Big) \\ &\quad + \frac{\frac{1}{\sqrt[s]{2}}\Big(1-\frac{1}{\sqrt[s]{2}}\Big(1+\frac{(s+2)\big(1-\frac{1}{\sqrt[s]{2}}\big)}{8(s+2)n4^n\binom{n}{s}}\Big)\Big)}{2n4^n}\Big) && (83)\\ &= \frac{\frac{1}{\sqrt[s]{2}}\big(1-\frac{1}{\sqrt[s]{2}}\big)\alpha_+^{s-1}}{(2n4^n)^{s+1}}\Big(1-\frac{\sqrt[s]{2}(s+1)}{4(s+2)} - \frac{1}{8\sqrt[s]{2}n4^n\binom{n}{s}}\Big) && (84)\\ &> \frac{\frac{1}{\sqrt[s]{2}}\big(1-\frac{1}{\sqrt[s]{2}}\big)\alpha_+^{s-1}}{(2n4^n)^{s+1}}\Big(1-\frac{1}{2}-\frac{1}{8}\Big) > 0. && (85)\end{aligned}$$

This implies the maximizer $\beta^{\mathrm{I}} < 1-\alpha_+$. Hence,

$$\widehat{\beta^{\mathrm{I}}} < (1+\epsilon)\beta^{\mathrm{I}} < (1+\epsilon)(1-\alpha_+) \tag{86}$$

$$\le \left(1+\frac{1}{16(n+1)n8^n-1}\right)\left(1-\frac{1}{\sqrt[s]{2}}\left(1+\frac{1-\frac{1}{\sqrt[s]{2}}}{8(s+2)n4^n\binom{n}{s}}\right)\right) \tag{87}$$

$$\le \left(1+\frac{1}{8\sqrt[s]{2}(s+2)n4^n\binom{n}{s}-1}\right)\left(1-\frac{1}{\sqrt[s]{2}}\left(1+\frac{1-\frac{1}{\sqrt[s]{2}}}{8(s+2)n4^n\binom{n}{s}}\right)\right) \tag{88}$$

$$= 1-\frac{1}{\sqrt[s]{2}}. \tag{89}$$

For the lower bound, let

$$\alpha_- := \frac{1}{\sqrt[s]{2}}\left(1+\frac{1}{n2^n}\right)^{\frac{1}{s}} > \frac{1}{\sqrt[s]{2}}. \tag{90}$$

Since $s \le n-1$, we have:

$$\alpha_- \le \frac{1}{\sqrt[s]{2}}\left(1+\frac{1}{(s+1)2^{s+1}}\right)^{\frac{1}{s}} \tag{91}$$

$$< \frac{1}{\sqrt[s]{2}}\left(1+\frac{\log 2}{s+1}\right)^{\frac{1}{s}} < \frac{1}{\sqrt[s]{2}}\left(\mathrm{e}^{\frac{\log 2}{s+1}}\right)^{\frac{1}{s}} = \frac{1}{\sqrt[s+1]{2}}. \tag{92}$$

Besides that, note that

$$\alpha_-^s(2\alpha_-^s-1) = \frac{1}{2n2^n}\left(1+\frac{1}{n2^n}\right) > \frac{1}{2n2^n}. \tag{93}$$

Since $1 \le s \le n-1$, and $1 \le c_k \le \binom{n}{k}$ for $s \le k \le n$, then:

$$p'(\alpha_-) \le -\left(\frac{1}{2n4^n}\right)^s \alpha_-^{s-1}(2\alpha_-^s-1) + \sum_{k=s+1}^{n}\binom{n}{k}\left(\frac{1}{2n4^n}\right)^k \alpha_-^{k-1}(1-2\alpha_-^k) \tag{94}$$

$$\le -\left(\frac{1}{2n4^n}\right)^s \alpha_-^{s-1}(2\alpha_-^s-1) + \sum_{k=s+1}^{n}\binom{n}{k}\left(\frac{1}{2n4^n}\right)^{s+1}\alpha_-^{k-1} \tag{95}$$

$$= \frac{\alpha_-^{-1}}{(2n4^n)^s}\left(-\alpha_-^s(2\alpha_-^s-1) + \frac{1}{2n4^n}\sum_{k=s+1}^{n}\binom{n}{k}\alpha_-^k\right) \tag{96}$$

$$\le \frac{\alpha_-^{-1}}{(2n4^n)^s}\left(-\alpha_-^s(2\alpha_-^s-1) + \frac{1}{2n4^n}\sum_{k=0}^{n}\binom{n}{k}\alpha_-^k\right) \tag{97}$$

$$= \frac{\alpha_-^{-1}}{(2n4^n)^s}\left(-\alpha_-^s(2\alpha_-^s-1) + \frac{1}{2n4^n}(1+\alpha_-)^n\right) \tag{98}$$

$$< \frac{\alpha_-^{-1}}{(2n4^n)^s}\left(-\alpha_-^s(2\alpha_-^s-1) + \frac{1}{2n4^n}2^n\right) \tag{99}$$

$$< \frac{\alpha_-^{-1}}{(2n4^n)^s}\left(-\frac{1}{2n2^n} + \frac{1}{2n4^n}2^n\right) = 0. \tag{100}$$

This implies the maximizer $\beta^{\mathrm{I}} > 1-\alpha_-$. Hence,

$$\widehat{\beta^{\mathrm{I}}} > (1-\epsilon)\beta^{\mathrm{I}} > (1-\epsilon)(1-\alpha_-) \tag{101}$$

$$\ge \left(1-\frac{\frac{1}{\sqrt[s]{2}}-\frac{1}{\sqrt[n-1]{2}}\left(1+\frac{1}{n2^n}\right)^{\frac{1}{n-1}}}{1-\frac{1}{\sqrt[n-1]{2}}\left(1+\frac{1}{n2^n}\right)^{\frac{1}{n-1}}}\right)\left(1-\frac{1}{\sqrt[s]{2}}\left(1+\frac{1}{n2^n}\right)^{\frac{1}{s}}\right) \tag{102}$$

$$\ge \left(1-\frac{\frac{1}{\sqrt[s+1]{2}}-\frac{1}{\sqrt[s]{2}}\left(1+\frac{1}{n2^n}\right)^{\frac{1}{s}}}{1-\frac{1}{\sqrt[s]{2}}\left(1+\frac{1}{n2^n}\right)^{\frac{1}{s}}}\right)\left(1-\frac{1}{\sqrt[s]{2}}\left(1+\frac{1}{n2^n}\right)^{\frac{1}{s}}\right) \tag{103}$$

$$= 1-\frac{1}{\sqrt[s+1]{2}}. \tag{104}$$

Combining Eq. (89) and Eq. (104) yields our claim Eq. (62).

The numbers involved can be stored in $\mathrm{poly}(n)$ bits and be computed using high-precision arithmetics within $\mathrm{poly}(n)$ time. Therefore, this gives a polynomial-time reduction from the MDS problem to Problem 3, so the NP-hardness of the MDS problem [45] implies that Problem 3 is NP-hard. □

B.3 Proof of Theorem 3

Before proving Theorem 3, we give an auxiliary lemma.

Lemma 8. *Under the SIR model, for each possible history $\boldsymbol{Y} \in \mathrm{supp}(P)$, there exists a number $\omega_{\boldsymbol{Y}} > 0$ independent of $\boldsymbol{\beta}$ such that*

$$P_{\boldsymbol{\beta}}[Y] = \omega_{\boldsymbol{Y}}(\beta^{\mathrm{I}})^{n^{\mathrm{IR}}(\boldsymbol{y}_T)-n^{\mathrm{IR}}(\boldsymbol{y}_0)}(\beta^{\mathrm{R}})^{n^{\mathrm{R}}(\boldsymbol{y}_T)-n^{\mathrm{R}}(\boldsymbol{y}_0)}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{105}$$

Proof. Fix the history $\boldsymbol{Y}$, and we will omit $\boldsymbol{Y}$ in some notations. Under the SIR model, we have the factorization:

$$P_{\boldsymbol{\beta}}[Y] = P[\boldsymbol{y}_0]\prod_{t=1}^{T}\prod_{u\in\mathcal{V}}P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}] \tag{106}$$

$$= P[\boldsymbol{y}_0]\prod_{u\in\mathcal{V}}\left(\prod_{t=1}^{T}P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right). \tag{107}$$

Let $\mathcal{U}_{x_0}^{x_1}$ denote the set of nodes with initial state x_0 and final state x_1:

$$\mathcal{U}_{x_0}^{x_1} := \{u \in \mathcal{V} : y_{0,u} = x_0,\ y_{T,u} = x_1\}. \tag{108}$$

Then, the node set $\mathcal{V}$ can be decomposed disjointly into

$$\mathcal{V} = \mathcal{U}_{\mathrm{S}}^{\mathrm{S}} \cup \mathcal{U}_{\mathrm{S}}^{\mathrm{I}} \cup \mathcal{U}_{\mathrm{S}}^{\mathrm{R}} \cup \mathcal{U}_{\mathrm{I}}^{\mathrm{I}} \cup \mathcal{U}_{\mathrm{I}}^{\mathrm{R}} \cup \mathcal{U}_{\mathrm{R}}^{\mathrm{R}}. \tag{109}$$

Besides that, for each node $u \in \mathcal{U}_{\mathrm{S}}^{\mathrm{I}} \cup \mathcal{U}_{\mathrm{S}}^{\mathrm{R}}$, let $\mathcal{I}_u$ denote the set of neighbors that may have infected u:

$$\mathcal{I}_u := \{v \in \mathcal{N}_u : y_{h_u^{\mathrm{I}}-1,u} = \mathrm{I}\} \ne \varnothing. \tag{110}$$

Now we calculate $\prod_{t=1}^{T}P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]$ according to the decomposition Eq. (108). For each node $u \in \mathcal{U}_{\mathrm{S}}^{\mathrm{S}}$, it is never infected, so:

$$\prod_{t=1}^{T}P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}] = \prod_{t=1}^{T}\prod_{v\in\mathcal{N}_u \wedge y_{t-1,v}=\mathrm{I}}(1-\beta^{\mathrm{I}}) \tag{111}$$

$$= \prod_{t=1}^{T}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)) = 1+\mathrm{O}(\|\boldsymbol{\beta}\|). \tag{112}$$

For each node $u \in \mathcal{U}_{\mathtt{S}}^{\mathtt{I}}$, it is infected but never recovered, so:

$$\prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]$$

$$= \left(\prod_{t=1}^{h_u^{\mathtt{I}}-1} \prod_{v\in\mathcal{N}_u \wedge y_{t-1,v}=\mathtt{I}} (1-\beta^{\mathtt{I}})\right)\left(1 - \prod_{v\in\mathcal{N}_u \wedge y_{h_u^{\mathtt{I}}-1,v}=\mathtt{I}} (1-\beta^{\mathtt{I}})\right)\left(\prod_{t=h_u^{\mathtt{I}}}^{T}(1-\beta^{\mathtt{R}})\right) \tag{113}$$

$$= \left(\prod_{t=1}^{h_u^{\mathtt{I}}-1} \prod_{v\in\mathcal{N}_u \wedge y_{t-1,v}=\mathtt{I}} (1-\beta^{\mathtt{I}})\right)\left(1-(1-\beta^{\mathtt{I}})^{|\mathcal{I}_u|}\right)\left(\prod_{t=h_u^{\mathtt{I}}}^{T}(1-\beta^{\mathtt{R}})\right) \tag{114}$$

$$= (1+\mathrm{O}(\|\boldsymbol{\beta}\|)) \cdot (|\mathcal{I}_u|\beta^{\mathtt{I}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|))) \cdot (1+\mathrm{O}(\|\boldsymbol{\beta}\|)) \tag{115}$$

$$= |\mathcal{I}_u|\beta^{\mathtt{I}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{116}$$

For each node $u \in \mathcal{U}_{\mathtt{S}}^{\mathtt{R}}$, it is infected and recovered, so:

$$\prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]$$

$$= \left(\prod_{t=1}^{h_u^{\mathtt{I}}-1} \prod_{v\in\mathcal{N}_u \wedge y_{t-1,v}=\mathtt{I}} (1-\beta^{\mathtt{I}})\right)\left(1 - \prod_{v\in\mathcal{N}_u \wedge y_{h_u^{\mathtt{I}}-1,v}=\mathtt{I}} (1-\beta^{\mathtt{I}})\right)\left(\prod_{t=h_u^{\mathtt{I}}}^{h_u^{\mathtt{R}}}(1-\beta^{\mathtt{R}})\right)\beta^{\mathtt{R}} \tag{117}$$

$$= (1+\mathrm{O}(\|\boldsymbol{\beta}\|)) \cdot (|\mathcal{I}_u|\beta^{\mathtt{I}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|))) \cdot (1+\mathrm{O}(\|\boldsymbol{\beta}\|)) \cdot \beta^{\mathtt{R}} \tag{118}$$

$$= |\mathcal{I}_u|\beta^{\mathtt{I}}\beta^{\mathtt{R}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{119}$$

For each node $u \in \mathcal{U}_{\mathtt{I}}^{\mathtt{I}}$, it is never recovered, so:

$$\prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}] = \prod_{t=1}^{T}(1-\beta^{\mathtt{R}}) = (1+\mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{120}$$

For each node $u \in \mathcal{U}_{\mathtt{I}}^{\mathtt{R}}$, it is eventually recovered, so:

$$\prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}] = \left(\prod_{t=1}^{h_u^{\mathtt{R}}}(1-\beta^{\mathtt{R}})\right)\beta^{\mathtt{R}} \tag{121}$$

$$= (1+\mathrm{O}(\|\boldsymbol{\beta}\|)) \cdot \beta^{\mathtt{R}} = \beta^{\mathtt{R}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{122}$$

For each node $u \in \mathcal{U}_{\mathtt{R}}^{\mathtt{R}}$, its state does not change, so:

$$\prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}] = \prod_{t=1}^{T} 1 = 1. \tag{123}$$

Finally, note that

$$|\mathcal{U}_{\mathtt{S}}^{\mathtt{I}} \cup \mathcal{U}_{\mathtt{S}}^{\mathtt{R}}| = n^{\mathtt{IR}}(\boldsymbol{y}_T) - n^{\mathtt{IR}}(\boldsymbol{y}_0), \tag{124}$$

$$|\mathcal{U}_{\mathtt{S}}^{\mathtt{R}} \cup \mathcal{U}_{\mathtt{I}}^{\mathtt{R}}| = n^{\mathtt{R}}(\boldsymbol{y}_T) - n^{\mathtt{R}}(\boldsymbol{y}_0). \tag{125}$$

Therefore,

$$P_{\boldsymbol{\beta}}[Y] = P[\boldsymbol{y}_0] \prod_{u\in\mathcal{V}} \left(\prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right) \tag{126}$$

$$= P[\boldsymbol{y}_0] \prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{S}}\cup\mathcal{U}_{\mathtt{S}}^{\mathtt{I}}\cup\mathcal{U}_{\mathtt{S}}^{\mathtt{R}}\cup\mathcal{U}_{\mathtt{I}}^{\mathtt{I}}\cup\mathcal{U}_{\mathtt{I}}^{\mathtt{R}}\cup\mathcal{U}_{\mathtt{R}}^{\mathtt{R}}} \left(\prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right) \tag{127}$$

$$\begin{aligned} = P[\boldsymbol{y}_0] &\left(\prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{S}}} \prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right)\left(\prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{I}}} \prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right) \\ &\left(\prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{R}}} \prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right)\left(\prod_{u\in\mathcal{U}_{\mathtt{I}}^{\mathtt{I}}} \prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right) \\ &\left(\prod_{u\in\mathcal{U}_{\mathtt{I}}^{\mathtt{R}}} \prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right)\left(\prod_{u\in\mathcal{U}_{\mathtt{R}}^{\mathtt{R}}} \prod_{t=1}^{T} P_{\boldsymbol{\beta}}[y_{t,u} \mid \boldsymbol{y}_{t-1}]\right) \end{aligned} \tag{128}$$

$$\begin{aligned} = P[\boldsymbol{y}_0] &\left(\prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{S}}} (1+\mathrm{O}(\|\boldsymbol{\beta}\|))\right)\left(\prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{I}}} (|\mathcal{I}_u|\beta^{\mathtt{I}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)))\right) \\ &\left(\prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{R}}} (|\mathcal{I}_u|\beta^{\mathtt{I}}\beta^{\mathtt{R}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)))\right)\left(\prod_{u\in\mathcal{U}_{\mathtt{I}}^{\mathtt{I}}} (1+\mathrm{O}(\|\boldsymbol{\beta}\|))\right) \\ &\left(\prod_{u\in\mathcal{U}_{\mathtt{I}}^{\mathtt{R}}} (\beta^{\mathtt{R}}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)))\right)\left(\prod_{u\in\mathcal{U}_{\mathtt{R}}^{\mathtt{R}}} \prod_{t=1}^{T} 1\right) \end{aligned} \tag{129}$$

$$= \left(P[\boldsymbol{y}_0] \prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{I}}\cup\mathcal{U}_{\mathtt{S}}^{\mathtt{R}}} |\mathcal{I}_u|\right)(\beta^{\mathtt{I}})^{|\mathcal{U}_{\mathtt{S}}^{\mathtt{I}}\cup\mathcal{U}_{\mathtt{S}}^{\mathtt{R}}|}(\beta^{\mathtt{R}})^{|\mathcal{U}_{\mathtt{S}}^{\mathtt{R}}\cup\mathcal{U}_{\mathtt{I}}^{\mathtt{R}}|}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)) \tag{130}$$

$$= \omega_Y(\beta^{\mathtt{I}})^{n^{\mathtt{IR}}(\boldsymbol{y}_T)-n^{\mathtt{IR}}(\boldsymbol{y}_0)}(\beta^{\mathtt{R}})^{n^{\mathtt{R}}(\boldsymbol{y}_T)-n^{\mathtt{R}}(\boldsymbol{y}_0)}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)), \tag{131}$$

where $\omega_Y \coloneqq P[\boldsymbol{y}_0] \prod_{u\in\mathcal{U}_{\mathtt{S}}^{\mathtt{I}}\cup\mathcal{U}_{\mathtt{S}}^{\mathtt{R}}} |\mathcal{I}_u| > 0$ because $\mathcal{I}_u \neq \emptyset$. □

Now we are ready to prove Theorem 3.

Proof of Theorem 3. By Lemma 8,

$$P_{\boldsymbol{\beta}}[Y] = \omega_Y(\beta^{\mathtt{I}})^{n^{\mathtt{IR}}(\boldsymbol{y}_T)-n^{\mathtt{IR}}(\boldsymbol{y}_0)}(\beta^{\mathtt{R}})^{n^{\mathtt{R}}(\boldsymbol{y}_T)-n^{\mathtt{R}}(\boldsymbol{y}_0)}(1+\mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{132}$$

Thus,

$$\log P_{\boldsymbol{\beta}}[Y] = \log\omega_Y + (n^{\mathtt{IR}}(\boldsymbol{y}_T) - n^{\mathtt{IR}}(\boldsymbol{y}_0))\log\beta^{\mathtt{I}} \tag{133}$$

$$+ (n^{\mathtt{R}}(\boldsymbol{y}_T) - n^{\mathtt{R}}(\boldsymbol{y}_0))\log\beta^{\mathtt{R}} + \log(1+\mathrm{O}(\|\boldsymbol{\beta}\|)) \tag{134}$$

$$= \log\omega_Y + (n^{\mathtt{IR}}(\boldsymbol{y}_T) - n^{\mathtt{IR}}(\boldsymbol{y}_0))\log\beta^{\mathtt{I}} \tag{135}$$

$$+ (n^{\mathtt{R}}(\boldsymbol{y}_T) - n^{\mathtt{R}}(\boldsymbol{y}_0))\log\beta^{\mathtt{R}} + \mathrm{O}(\|\boldsymbol{\beta}\|). \tag{136}$$

Note that $P_{\boldsymbol{\beta}}[Y]$ is a polynomial of $\boldsymbol{\beta}$, so it is differentiable w.r.t. $\boldsymbol{\beta}$. Hence,

$$\frac{\partial}{\partial\beta^{\mathtt{I}}}\log P_{\boldsymbol{\beta}}[Y] = \frac{n^{\mathtt{IR}}(\boldsymbol{y}_T) - n^{\mathtt{IR}}(\boldsymbol{y}_0)}{\beta^{\mathtt{I}}} + \mathrm{O}(1), \tag{137}$$

$$\frac{\partial}{\partial\beta^{\mathtt{R}}}\log P_{\boldsymbol{\beta}}[Y] = \frac{n^{\mathtt{R}}(\boldsymbol{y}_T) - n^{\mathtt{R}}(\boldsymbol{y}_0)}{\beta^{\mathtt{R}}} + \mathrm{O}(1). \tag{138}$$

It follows that

$$\frac{\partial}{\partial\beta^{\mathtt{I}}}P_{\boldsymbol{\beta}}[Y] = \left(\frac{\partial}{\partial\beta^{\mathtt{I}}}\log P_{\boldsymbol{\beta}}[Y]\right)P_{\boldsymbol{\beta}}[Y] \tag{139}$$

$$= \left(\frac{n^{\mathtt{IR}}(\boldsymbol{y}_T) - n^{\mathtt{IR}}(\boldsymbol{y}_0)}{\beta^{\mathtt{I}}} + \mathrm{O}(1)\right)P_{\boldsymbol{\beta}}[Y], \tag{140}$$

$$\frac{\partial}{\partial\beta^{\mathtt{R}}}P_{\boldsymbol{\beta}}[Y] = \left(\frac{\partial}{\partial\beta^{\mathtt{R}}}\log P_{\boldsymbol{\beta}}[Y]\right)P_{\boldsymbol{\beta}}[Y] \tag{141}$$

$$= \left(\frac{n^{\mathtt{R}}(\boldsymbol{y}_T) - n^{\mathtt{R}}(\boldsymbol{y}_0)}{\beta^{\mathtt{R}}} + \mathrm{O}(1)\right)P_{\boldsymbol{\beta}}[Y], \tag{142}$$

Therefore, if $n^{\mathtt{IR}}(\boldsymbol{y}_T) > n^{\mathtt{IR}}(\boldsymbol{y}_0)$, then:

$$\frac{\partial}{\partial \beta^{\mathtt{I}}} P_{\boldsymbol{\beta}}[Y] = \Theta\Big(\frac{1}{\beta^{\mathtt{I}}}\Big) P_{\boldsymbol{\beta}}[Y]; \tag{143}$$

if $n^{\mathtt{R}}(\boldsymbol{y}_T) > n^{\mathtt{R}}(\boldsymbol{y}_0)$, then:

$$\frac{\partial}{\partial \beta^{\mathtt{R}}} P_{\boldsymbol{\beta}}[Y] = \Theta\Big(\frac{1}{\beta^{\mathtt{R}}}\Big) P_{\boldsymbol{\beta}}[Y]. \tag{144}$$

□

B.4 Proof of Theorem 4

Proof. Since $n^{\mathtt{I}}(\boldsymbol{y}_0)$ and $n^{\mathtt{R}}(\boldsymbol{y}_0)$ are fixed a.s., we write $n_0^{\mathtt{IR}} := n^{\mathtt{IR}}(\boldsymbol{y}_0)$ and $n_0^{\mathtt{R}} := n^{\mathtt{R}}(\boldsymbol{y}_0)$ are constants. Fix a snapshot $\boldsymbol{y}_T$, then $n_T^{\mathtt{IR}} := n^{\mathtt{IR}}(\boldsymbol{y}_T)$ and $n_T^{\mathtt{R}} := n^{\mathtt{R}}(\boldsymbol{y}_T)$ are also fixed. Then by Lemma 8, for any history $Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T)$,

$$P_{\boldsymbol{\beta}}[Y] = \omega_Y (\beta^{\mathtt{I}})^{n_T^{\mathtt{IR}} - n_0^{\mathtt{IR}}} (\beta^{\mathtt{R}})^{n_T^{\mathtt{R}} - n_0^{\mathtt{R}}} (1 + \mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{145}$$

Thus, the probability of the snapshot $\boldsymbol{y}_T$ is

$$P_{\boldsymbol{\beta}}[\boldsymbol{y}_T] = \sum_{Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T)} P_{\boldsymbol{\beta}}[Y] \tag{146}$$

$$= \sum_{Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T)} \omega_Y (\beta^{\mathtt{I}})^{n_T^{\mathtt{IR}} - n_0^{\mathtt{IR}}} (\beta^{\mathtt{R}})^{n_T^{\mathtt{R}} - n_0^{\mathtt{R}}} (1 + \mathrm{O}(\|\boldsymbol{\beta}\|)). \tag{147}$$

Fix a node $u \in \mathcal{V}$. Categorize the possible histories according to the hitting times $0 \le t \le T+1$ of the node u:

$$\mathcal{Y}_t^{\mathtt{I}} := \{Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T) : h_u^{\mathtt{I}}(Y) = t\}, \tag{148}$$

$$\mathcal{Y}_t^{\mathtt{R}} := \{Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T) : h_u^{\mathtt{R}}(Y) = t\}. \tag{149}$$

Let

$$\omega_t^{\mathtt{I}} := \sum_{Y \in \mathcal{Y}_t^{\mathtt{I}}} \omega_Y, \quad \omega_t^{\mathtt{R}} := \sum_{Y \in \mathcal{Y}_t^{\mathtt{R}}} \omega_Y, \quad \omega := \sum_{Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T)} \omega_Y. \tag{150}$$

Then by cancellation, the expected hitting time to state $\mathtt{I}$ is

$$\mathop{\mathbb{E}}_{Y \sim P_{\boldsymbol{\beta}} \mid \boldsymbol{y}_T} [h_u^{\mathtt{I}}(Y)] \tag{151}$$

$$= \sum_{t=0}^{T+1} t \sum_{Y \in \mathcal{Y}_t^{\mathtt{I}}} P_{\boldsymbol{\beta}}[Y \mid \boldsymbol{y}_T] \tag{152}$$

$$= \frac{\sum_{t=0}^{T+1} t \sum_{Y \in \mathcal{Y}_t^{\mathtt{I}}} P_{\boldsymbol{\beta}}[Y]}{P_{\boldsymbol{\beta}}[\boldsymbol{y}_T]} \tag{153}$$

$$= \frac{\sum_{t=0}^{T+1} t \sum_{Y \in \mathcal{Y}_t^{\mathtt{I}}} \omega_Y (\beta^{\mathtt{I}})^{n_T^{\mathtt{IR}} - n_0^{\mathtt{IR}}} (\beta^{\mathtt{R}})^{n_T^{\mathtt{R}} - n_0^{\mathtt{R}}} (1 + \mathrm{O}(\|\boldsymbol{\beta}\|))}{\sum_{Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T)} \omega_Y (\beta^{\mathtt{I}})^{n_T^{\mathtt{IR}} - n_0^{\mathtt{IR}}} (\beta^{\mathtt{R}})^{n_T^{\mathtt{R}} - n_0^{\mathtt{R}}} (1 + \mathrm{O}(\|\boldsymbol{\beta}\|))} \tag{154}$$

$$= \frac{\sum_{t=0}^{T+1} t \sum_{Y \in \mathcal{Y}_t^{\mathtt{I}}} \omega_Y (1 + \mathrm{O}(\|\boldsymbol{\beta}\|))}{\sum_{Y \in \mathrm{supp}(P \mid \boldsymbol{y}_T)} \omega_Y (1 + \mathrm{O}(\|\boldsymbol{\beta}\|))} \tag{155}$$

$$= \frac{\sum_{t=0}^{T+1} t\omega_t^{\mathtt{I}} + \mathrm{O}(\|\boldsymbol{\beta}\|)}{\omega + \mathrm{O}(\|\boldsymbol{\beta}\|)} \tag{156}$$

$$= \Big(\sum_{t=0}^{T+1} t\omega_t^{\mathtt{I}} + \mathrm{O}(\|\boldsymbol{\beta}\|)\Big)\Big(\frac{1}{\omega} + \mathrm{O}(\|\boldsymbol{\beta}\|)\Big) \tag{157}$$

$$= \frac{1}{\omega} \sum_{t=0}^{T+1} t\omega_t^{\mathtt{I}} + \mathrm{O}(\|\boldsymbol{\beta}\|). \tag{158}$$

Similarly, the expected hitting time to state $\mathtt{R}$ is

$$\mathop{\mathbb{E}}_{Y \sim P_{\boldsymbol{\beta}} \mid \boldsymbol{y}_T} [h_u^{\mathtt{R}}(Y)] = \frac{1}{\omega} \sum_{t=0}^{T+1} t\omega_t^{\mathtt{R}} + \mathrm{O}(\|\boldsymbol{\beta}\|). \tag{159}$$

Therefore,

$$\nabla_{\boldsymbol{\beta}} \mathop{\mathbb{E}}_{Y \sim P_{\boldsymbol{\beta}} \mid \boldsymbol{y}_T} [h_u^{\mathtt{I}}(Y)] = \nabla_{\boldsymbol{\beta}} \Big(\frac{1}{\omega} \sum_{t=0}^{T+1} t\omega_t^{\mathtt{I}} + \mathrm{O}(\|\boldsymbol{\beta}\|)\Big) = \mathrm{O}(1), \tag{160}$$

$$\nabla_{\boldsymbol{\beta}} \mathop{\mathbb{E}}_{Y \sim P_{\boldsymbol{\beta}} \mid \boldsymbol{y}_T} [h_u^{\mathtt{R}}(Y)] = \nabla_{\boldsymbol{\beta}} \Big(\frac{1}{\omega} \sum_{t=0}^{T+1} t\omega_t^{\mathtt{R}} + \mathrm{O}(\|\boldsymbol{\beta}\|)\Big) = \mathrm{O}(1). \tag{161}$$

□

B.5 Proof of Theorem 5

Proof. The sufficient expressiveness of $Q_{\boldsymbol{\theta}}$ implies that there exists a parameter sequence $\{\boldsymbol{\theta}_k\}_{k \ge 1}$ such that

$$\lim_{k \to +\infty} Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y] = P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T], \qquad \forall \boldsymbol{y}_T. \tag{162}$$

Since $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$ and $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$ share a common finite support, then $\max_{Y \in \mathrm{supp}(P_{\widehat{\boldsymbol{\beta}}})} \big|\log Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T]\big| < \infty$. It follows from the dominated convergence theorem that

$$0 \le \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} [(\log Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T])^2] \tag{163}$$

$$\le \lim_{k \to +\infty} \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} [(\log Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T])^2] \tag{164}$$

$$= \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} \Big[\lim_{k \to +\infty} (\log Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T])^2\Big] \tag{165}$$

$$= \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} \Big[\Big(\log \lim_{k \to +\infty} Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T]\Big)^2\Big] \tag{166}$$

$$= \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} [(\log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T])^2] \tag{167}$$

$$= \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} [0^2] = 0. \tag{168}$$

This implies

$$\lim_{k \to +\infty} \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} [(\log Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T])^2] \tag{169}$$

$$= \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y \sim P_{\widehat{\boldsymbol{\beta}}}} [(\log Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] - \log P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T])^2] = 0. \tag{170}$$

Hence, $\{\boldsymbol{\theta}_k\}_{k \ge 1}$ is asymptotically optimal for the original objective Eq. (21). Meanwhile, for any other parameter sequence $\{\widetilde{\boldsymbol{\theta}}_k\}$ where $Q_{\widetilde{\boldsymbol{\theta}}_k}(\boldsymbol{y}_T)$ do not converge to $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$, then they have nonzero objectives and are thus non-optimal. Therefore, any asymptotically optimal parameter sequence $\{\boldsymbol{\theta}_k\}_{k \ge 1}$ for the objective Eq. (21) must converge to $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$.

Next, we will show that any asymptotically optimal parameter sequence $\{\boldsymbol{\theta}_k\}_{k\geq 1}$ for the objective Eq. (22) must also converge to $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$. For any $\boldsymbol{y}_T$, since $\operatorname{supp}(Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)) = \operatorname{supp}(P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T)$, then:

$$\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}\mid \boldsymbol{y}_T}\Big[\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big] \tag{171}$$

$$= \sum_{Y\in\operatorname{supp}(P_{\widehat{\boldsymbol{\beta}}}\mid \boldsymbol{y}_T)} P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T]\cdot\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]} \tag{172}$$

$$= \frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\cdot\sum_{Y\in\operatorname{supp}(P_{\widehat{\boldsymbol{\beta}}}\mid \boldsymbol{y}_T)} Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] \tag{173}$$

$$= \frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\cdot\sum_{Y\in\operatorname{supp}(Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T))} Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y] \tag{174}$$

$$= \frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\cdot 1 = \frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}. \tag{175}$$

By Jensen's inequality,

$$\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}\mid \boldsymbol{y}_T}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{176}$$

$$\geq \psi\Big(\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}\mid \boldsymbol{y}_T}\Big[\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big]\Big) \tag{177}$$

$$= \psi\Big(\frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\Big). \tag{178}$$

Since convexity implies continuity, then by the law of total expectation,

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{179}$$

$$= \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\boldsymbol{y}_T\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}\mid \boldsymbol{y}_T}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big]\Big] \tag{180}$$

$$\geq \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\boldsymbol{y}_T\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\Big)\Big] \tag{181}$$

$$= \mathop{\mathbb{E}}_{\boldsymbol{y}_T\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\Big)\Big] \tag{182}$$

$$= \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\Big)\Big]. \tag{183}$$

Thus, for any parameter sequence $\{\boldsymbol{\theta}_k\}$ such that $Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)$ converge to $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$,

$$\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\Big)\Big] \leq \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{184}$$

$$\leq \lim_{k\to+\infty} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{185}$$

$$= \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\lim_{k\to+\infty}\psi\Big(\frac{Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{186}$$

$$= \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{\lim_{k\to+\infty} Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{187}$$

$$= \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{P_{\widehat{\boldsymbol{\beta}}}[Y \mid \boldsymbol{y}_T]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{188}$$

$$= \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\Big)\Big], \tag{189}$$

which implies

$$\lim_{k\to+\infty} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] = \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big]. \tag{190}$$

This suggests that the sequence $\{\boldsymbol{\theta}_k\}_{k\geq 1}$ is asymptotically optimal for the objective Eq. (22). Meanwhile, for any other parameter sequence $\{\widetilde{\boldsymbol{\theta}}_k\}$ where $Q_{\widetilde{\boldsymbol{\theta}}_k}(\boldsymbol{y}_T)$ converge to a distribution other than $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$ for some $\boldsymbol{y}_T$ with $\#\operatorname{supp}(P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T) > 1$, then $\frac{\lim_{k\to+\infty} Q_{\widetilde{\boldsymbol{\theta}}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]} \,\Big|\, \boldsymbol{y}_T$ is non-degenerate. By Fatou's lemma and Jensen's inequality with strict convexity,

$$\lim_{k\to+\infty} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{191}$$

$$\geq \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\lim_{k\to+\infty}\psi\Big(\frac{Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{192}$$

$$= \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{\lim_{k\to+\infty} Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big] \tag{193}$$

$$> \psi\Big(\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\frac{\lim_{k\to+\infty} Q_{\boldsymbol{\theta}_k}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big]\Big) \tag{194}$$

$$= \psi\Big(\mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\frac{1}{P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_T]}\Big]\Big) \tag{195}$$

$$= \min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{Y\sim P_{\widehat{\boldsymbol{\beta}}}}\Big[\psi\Big(\frac{Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[Y]}{P_{\widehat{\boldsymbol{\beta}}}[Y]}\Big)\Big]. \tag{196}$$

This suggests that $\{\widetilde{\boldsymbol{\theta}}_k\}_{k\geq 1}$ is not asymptotically optimal. Note that those $\boldsymbol{y}_T$ with $\#\operatorname{supp}(P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T) = 1$ has no influence, because in that case $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$ is degenerate and thus does not depend on $\boldsymbol{\theta}$. Hence, any asymptotically optimal parameter sequence $\{\boldsymbol{\theta}_k\}_{k\geq 1}$ for the objective Eq. (22) must also converge to $P_{\widehat{\boldsymbol{\beta}}} \mid \boldsymbol{y}_T$. Therefore, the objectives Eq. (21) and Eq. (22) are equivalent. □

B.6 Proof of Proposition 6

Proof of (i). Since there are $T+1$ times and n nodes, then there are $O(Tn)$ pseudolikelihoods to be computed in total. The time complexity to compute the pseudolikelihood of a node at a time is at most proportional to the number of edges connecting to that node, and there are m edges in total, so the total time complexity to compute all pseudolikelihoods is $O(T(n+m))$. Furthermore, since the backpropagation algorithm has the same complexity as the forward computation, the overall time complexity of an iteration is still $O(T(n+m))$. □

Proof of (ii). To predict the probabilities $q^{\mathrm{I}}_{t,u}$ and $q^{\mathrm{R}}_{t,u}$, the time complexity is $O(T(n+m))$ due to the graph neural network $Q_{\boldsymbol{\theta}}$. To generate a snapshot, the two main steps are sorting probabilities and maintaining counters. Sorting the n probabilities $q^{\mathrm{I}}_{t,u}$ of all nodes $u\in\mathcal{V}$ takes $O(n\log n)$ time at each t. Aggregating and updating the counters $\rho_{t,u}$ for all nodes $u\in\mathcal{V}$ at each t take $O(n+m)$ time

in total, because each edge is involved in O(1) operations. Since sampling a history needs to generate T snapshots, then the total time complexity is $O(T(n \log n + m))$. □

C THE PROPOSAL IN M–H MCMC

In this section, we detail our design of the proposal $Q_{\boldsymbol{\theta}}$ in M–H MCMC.

C.1 Neural Architecture

The backbone of $Q_{\boldsymbol{\theta}}$ is an Anisotropic GNN with edge gating mechanism [7, 44, 65]. Let $\boldsymbol{g}_u^{\ell,1}$ and $\boldsymbol{g}_{u,v}^{\ell,2}$ denote the node and edge embeddings at layer ℓ associated with node u and edge (u, v), respectively. The embeddings at the next layer is propagated with an anisotropic message passing scheme:

$$\boldsymbol{g}_u^{\ell+1,1} := \boldsymbol{g}_u^{\ell,1} + a(\mathrm{BN}(\boldsymbol{W}^{\ell,1}\boldsymbol{g}_u^{\ell,1} + \mathop{\mathfrak{A}}_{v\in\mathcal{N}_u}(\sigma(\boldsymbol{g}_{u,v}^{\ell,2}) \odot (\boldsymbol{W}^{\ell,2}\boldsymbol{g}_v^{\ell,1})))), \tag{197}$$

$$\boldsymbol{g}_{u,v}^{\ell+1,2} := \boldsymbol{g}_{u,v}^{\ell,2} + a(\mathrm{BN}(\boldsymbol{W}^{\ell,3}\boldsymbol{g}_{u,v}^{\ell,2} + \boldsymbol{W}^{\ell,4}\boldsymbol{g}_u^{\ell,1} + \boldsymbol{W}^{\ell,5}\boldsymbol{g}_v^{\ell,1})). \tag{198}$$

where $\boldsymbol{W}^{\ell,1}, \ldots, \boldsymbol{W}^{\ell,5} \in \mathbb{R}^{d\times d}$ are learnable parameters of layer ℓ, a denotes the activation function (we use SiLU [24] in this paper), BN denotes the Batch Normalization operator [42], $\mathfrak{A}$ denotes the aggregation function (we use mean pooling in this work), σ denotes the sigmoid function, and $\odot$ denotes the Hadamard product. A Multi-Layer Perceptron (MLP) is appended after the GNN to produce the final outputs. The node inputs $\boldsymbol{g}_u^{0,1}$ are initialized by feeding $y_{T,u}$ into a linear layer. The edge inputs $\boldsymbol{g}_{u,v}^{0,2}$ are learnable parameters.

C.2 Sampling Scheme

To satisfy the condition in Theorem 5 and not to generate waste samples, we require that $\mathrm{supp}(Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)) = \mathrm{supp}(P \mid \boldsymbol{y}_T)$ for any snapshot $\boldsymbol{y}_T$. A sufficient condition for this requirement is that $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)[\boldsymbol{y}_t \mid \boldsymbol{y}_{t+1}] > 0$ iff $P_{\widehat{\boldsymbol{\beta}}}[\boldsymbol{y}_{t+1} \mid \boldsymbol{y}_t] > 0$ for every $t = 1, \ldots, T$, as long as $P[\boldsymbol{y}_0]$ is nowhere vanishing. Hence, we design a sampling scheme according to this sufficient condition. We sample a history in the reverse temporal order. Provided that $\boldsymbol{y}_{t+1}$ is already generated, we next describe how to generate $\boldsymbol{y}_t$.

Given the observed snapshot $\boldsymbol{y}_T$, the GNN $Q_{\boldsymbol{\theta}}(\boldsymbol{y}_T)$ predicts two probabilities $0 < q_{t,u}^{\mathtt{I}}, q_{t,u}^{\mathtt{R}} < 1$ for each node $u \in \mathcal{V}$ at time t. There are two stages at time t. In the first stage, for each node $u \in \mathcal{V}$, if $y_{t+1,u} = \mathtt{R}$, we randomly change its state to $\mathtt{I}$ with probability $q_{t,u}^{\mathtt{I}}$. Let $y'_{t,u}$ denote the state of node u after the first stage. In the second stage, we sort the nodes $\mathcal{V}$ into $v_1, \ldots, v_n$ so that $q_{t,v_1}^{\mathtt{I}} \geq \cdots \geq q_{t,v_n}^{\mathtt{I}}$. We also maintain a counter $\rho_{t,u}$ for each node u to indicate the remaining chance to be infected from a neighbor. The counters $\rho_{t,u}$ are initialized as one plus the degree of the node u. Then, we decide whether to change the state of v_i to $\mathtt{S}$ sequentially in the order $v_1, \ldots, v_n$. If $y'_{t,v_i} \neq \mathtt{I}$, then we set $y_{t,v_i} := y'_{t,v_i}$. If $y'_{t,v_i} = \mathtt{I}$ and $\min_{u\in\{v_i\}\cup\mathcal{N}_{v_i}} \rho_{t,u} \leq 1$, then we set $y_{t,v_i} = \mathtt{I}$. Otherwise, we set $y_{t,v_i} := \mathtt{S}$ with probability $q_{t,v_i}^{\mathtt{I}}$ or $y_{t,v_i} := \mathtt{I}$ with probability $1 - q_{t,v_i}^{\mathtt{I}}$. If $y'_{t,v_i} = \mathtt{I}$ and $y_{t,v_i} = \mathtt{S}$, then we decrease $\rho_{t,u}$ by 1 for all $u \in \{v_i\} \cup \mathcal{N}_{v_i}$. If $y'_{t,v_i} = \mathtt{I}$ and $y_{t,v_i} = \mathtt{I}$, then we set $\rho_{t,u} := +\infty$ for all $u \in \{v_i\} \cup \mathcal{N}_{v_i}$. After the second stage, we get the states $\boldsymbol{y}_t$ at time t, and then we use it to generate $\boldsymbol{y}_{t-1}$, and so on.

D DETAILED EXPERIMENTAL SETTING

In this section, we give detailed description of datasets, baselines, and reproducibility.

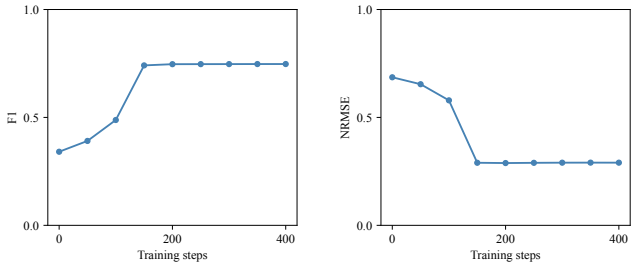
D.1 Datasets

We use 3 types of datasets. **(D1) Synthetic graphs and diffusion.** We generate random undirected graphs with 1,000 nodes using the Barabási–Albert (BA) model [4] with attachment 4 and the Erdős–Rényi (ER) model [25] with edge probability 0.008. We simulate SI and SIR diffusion for $T = 10$ with the infection rate 0.1 and the recovery rate 0.1. We randomly select 5% nodes as diffusion sources. **(D2) Synthetic diffusion on real graphs.** We use two real graphs and simulate diffusion on them. Oregon2[4] [55] is a collection of graphs representing the peering information in autonomous systems, and we use the graph of May 26, 2001, the largest one. Prost [68] is a bipartite graph representing prostitution reviews in an online forum. We simulate SI and SIR diffusion for $T = 15$ with the infection rate 0.1 and the recovery rate 0.05. We randomly select 10% nodes as diffusion sources. **(D3) Real diffusion on real graphs.** To evaluate how DITTO generalizes to real-world diffusion, we use 4 real-world diffusion datasets. BrFarmers[5] [69, 83] incorporates diffusion of technological innovations among Brazilian farmers. It is an SI-like diffusion, where an infection means that a farmer hears about the new technology from a friend and adopts it. Pol[6] [20, 70] is a temporal retweet network about a U.S. political event. It is an SI-like diffusion, because when a user retweets or is retweeted, they must have known about the event. Covid[7] is a dataset of Covid Community Levels from Feb 23, 2022 to Dec 21, 2022 released by CDC, where nodes are counties. We build a graph by connecting each node with its 10 nearest neighbors according to latitudes and longitudes. It is an SIR-like diffusion as follows. When a county becomes medium or high level for the first time, the county is "infected." After the last time a county becomes low level and does not change again, the county is "recovered." Hebrew [5] is a temporal retweet network among Hebrew tweets about an Israeli election event. It is an SIR-like diffusion as follows. For users who retweet at most once and are never retweeted, they never become influential in this event, so they are "susceptible." For users who retweet at least twice or are retweeted by others, they actively involve or influence other users in the event, so they are "infected." For "infected" users, after the last time that they are retweeted, they are no longer influential, so they become "recovered."

D.2 Baselines

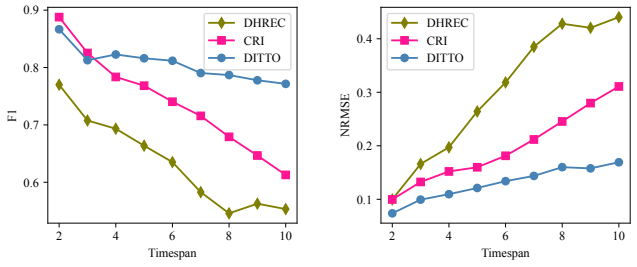
We compare DITTO with 2 types of baselines. **(B1) Supervised methods for time series imputation.** Diffusion history reconstruction can be alternatively formulated as time series imputation on graphs. Therefore, we also compare DITTO with state-of-the-art time series imputation methods, including BRITS [9] for multivariate time series, and GRIN [18] and SPIN [58] for graph time series. Since these methods are all based on supervised learning, we use our estimated diffusion parameters to simulate diffusion histories as

[4] http://snap.stanford.edu/data/Oregon-2.html
[5] https://usccana.github.io/netdiffuseR/reference/brfarmers.html
[6] https://networkrepository.com/rt-pol.php
[7] https://data.cdc.gov/Public-Health-Surveillance/United-States-COVID-19-Community-Levels-by-County/3nnm-4jni



(a) F1 vs training steps. **(b) NRMSE vs training steps.**

Figure 5: Performance vs training steps.



(a) F1 vs timespan T. **(b) NRMSE vs timespan T.**

Figure 4: Performance vs timespan T.

training data. We follow the hyperparameters of baselines, except adjusting the batch size to fit in memory. **(B2) MLE-based methods for diffusion history reconstruction.** To date, few works have studied diffusion history reconstruction, and all of them are based on the MLE formulation. We compare DITTO with state-of-the-art methods DHREC [74] and CRI [17]. DHREC reduces the MLE formulation to the Prize Collecting Dominating Set Vertex Cover (PCDSVC) problem and uses a greedy algorithm to solve PCDSVC. It requires the knowledge of the diffusion model parameter. Therefore, we feed our estimated diffusion parameters to it. CRI designs a heuristic method based on clustering and reverse infection. It can estimate infection times but cannot estimate recovery times.

D.3 Reproducibility & Implementation Details

Experiments were run on Intel Xeon CPU @ 2.20GHz and NVIDIA Tesla P100 16GB GPU. Our source code is publicly available at https://github.com/q-rz/KDD23-DITTO. All datasets are publicly available. For each dataset, we will either provide a link to it or include it in our code repository.

For DITTO, we optimize $\widehat{\boldsymbol{\beta}}$ for $I = 500$ iterations. The proposal $Q_{\boldsymbol{\theta}}$ is a 3-layer GNN followed by a 2-layer MLP with hidden size 16. We train $Q_{\boldsymbol{\theta}}$ for $J = 500$ steps for D1 and D2, $J = 2,000$ for BrFarmers, $J = 300$ for Pol, $J = 250$ for Covid, and $J = 200$ for Hebrew. We use batch size $K = 10$ for D1, BrFarmers, and Covid, and $K = 2$ for D2, Pol, and Hebrew. We use the AdamW [57] optimizer with learning rate 0.003 for $\widehat{\boldsymbol{\beta}}$ and 0.001 for $Q_{\boldsymbol{\theta}}$. After training, we run M–H MCMC for $S = 10$ iterations with $L = 100$ samples per iteration and $\eta = 0.5$ for moving average. For the initial distribution $P[\boldsymbol{y}_0]$, we use the coefficient $\gamma = 1$ in MCMC. For supervised imputation methods in B1, we use the estimated diffusion parameters (unless specified) to generate training data. For GRIN and SPIN, we train them for 1,000 steps with batch size 1. We follow the other hyperparameters of these methods. For MLE-based methods in B2, we feed estimated diffusion parameters to them.

E ADDITIONAL EXPERIMENTS

In this section, we present addtional experimental results to answer RQ6 and RQ7.

E.1 Effect of Timespan

As the timespan T increases, the search space of possible histories grows exponentially and thus the uncertainty of the history grows accordingly. Hence, it is helpful to investigate the effect of timespan T. To answer RQ6, we vary the timespan T from 2 to 10 and compare the performance of DITTO and MLE-based methods. The results are shown in Fig. 4. As is expected, the performance of all methods degrades as T increases. Nonetheless, the performance of DITTO degrades slower than that of MLE-based methods. The results demonstrate that DITTO can better handle the uncertainty induced by the increase in the timespan T.

E.2 Ablation Study

To answer RQ7, we conduct ablation study on the effect of the number of training steps. We vary the number of training steps from 0 to 400 for the Pol dataset and compare the performance in terms of F1 and NRMSE. The results are shown in Fig. 5. When the number of training steps is less than 200, as the number of training steps increases, the performance of DITTO improves accordingly. When the number of training steps is more than 200, the performance does not change because the proposal already converges. The results suggest that the learned proposal in DITTO is indeed beneficial for M–H MCMC.

F LIMITATIONS & FUTURE WORK

One limitation of DITTO is that the history $\widehat{Y}$ reconstructed by Eq. (19) is not necessarily feasible under the SIR model. However, the perfect feasibility under the SIR model has limited significance in practice, because the SIR model is often considered as an over-simplification of real-world diffusion. Meanwhile, an alternative solution is to use the samples generated by M–H MCMC, as they are guaranteed to be feasible. Another potential limitation is that our theoretical analyses are based on small diffusion parameters, which is indeed the case for most real-world data [33, 62, 86]. Meanwhile, there might exist situations where infection rates are large. The analyses under large diffusion parameters is beyond the scope of our work. To our best knowledge, no literature has studied diffusion history reconstruction with large infection rates, which leads to an interesting future research direction that is worth an independent investigation. Besides that, there are a number of other directions that are worth future study, including extending to diffusion models other than the SI/SIR model, incorporating node and edge attributes to allow heterogeneity, and improving the expressiveness of the proposal to further accelerate the convergence of M–H MCMC.