Network Infusion to Infer Information Sources in Networks

Article · January 2015

DOI: 10.1109/TNSE.2018.2854218

CITATIONS
14

READS
205

4 authors:

Soheil Feizi
Massachusetts Institute of Technology
65 PUBLICATIONS **3,152** CITATIONS
SEE PROFILE

Ken R Duffy
National University of Ireland, Maynooth
134 PUBLICATIONS **2,542** CITATIONS
SEE PROFILE

Manolis Kellis
Massachusetts Institute of Technology
503 PUBLICATIONS **74,759** CITATIONS
SEE PROFILE

Muriel Médard
Massachusetts Institute of Technology
700 PUBLICATIONS **26,899** CITATIONS
SEE PROFILE

Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2014-028 December 2, 2014

Network Infusion to Infer Information Sources in Networks

Soheil Feizi, Ken Duffy, Manolis Kellis, and Muriel Medard

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Network Infusion to Infer Information Sources in Networks

Soheil Feizi*, Ken Duffy†, Manolis Kellis‡ and Muriel Médard§

December 2014

Abstract

Several models exist for diffusion of signals across biological, social, or engineered networks. However, the inverse problem of identifying the source of such propagated information appears more difficult even in the presence of multiple network snapshots, and especially for the single-snapshot case, given the many alternative, often similar, progression of diffusion that may lead to the same observed snapshots. Mathematically, this problem can be undertaken using a diffusion kernel that represents diffusion processes in a given network, but computing this kernel is computationally challenging in general. Here, we propose a path-based network diffusion kernel which considers edge-disjoint shortest paths among pairs of nodes in the network and can be computed efficiently for both homogeneous and heterogeneous continuous-time diffusion models. We use this network diffusion kernel to solve the inverse diffusion problem, which we term Network Infusion (NI), using both likelihood maximization and error minimization. The minimum error NI algorithm is based on an asymmetric Hamming premetric function and can balance between false positive and false negative error types. We apply this framework for both single-source and multi-source diffusion, for both single-snapshot and multi-snapshot observations, and using both uninformative and informative prior probabilities for candidate source nodes. We also provide proofs that under a standard susceptible-infected diffusion model, (1) the maximum-likelihood NI is mean-field optimal for tree structures or sufficiently sparse Erdös-Rényi graphs, (2) the minimum-error algorithm is mean-field optimal for regular tree structures, and (3) for sufficiently-distant sources, the multi-source solution is mean-field optimal in the regular tree structure. Moreover, we provide techniques to learn diffusion model parameters such as observation times. We apply NI to several synthetic networks and compare its performance to centrality-based and distance-based methods for Erdös-Rényi graphs, power-law networks, symmetric and asymmetric grids. Moreover, we use NI in two real-world applications. First, we identify the news sources for 3,553 stories in the Digg social news network, and validate our results based on annotated information, that was not provided to our algorithm. Second, we use NI to identify infusion hubs of human diseases, defined as gene candidates that can explain the connectivity pattern of disease-related genes in the human regulatory network. NI identifies infusion hubs of several human diseases including T1D, Parkinson, MS, SLE, Psoriasis and Schizophrenia. We show that, the inferred infusion hubs are biologically relevant and often not identifiable using the raw p-values.

Keywords. Network Infusion, Information Diffusion, Source Inference, Maximum Likelihood, Weighted Hamming Distance, Regulatory Network, Human Disease, Social Networks.

*Department of Electrical Engineering and Computer Science, MIT, Cambridge MA.
†Hamilton Institute, Maynooth University, Ireland, Ireland.
‡Department of Electrical Engineering and Computer Science, MIT, Cambridge MA.
§Department of Electrical Engineering and Computer Science, MIT, Cambridge MA.

1 Introduction

Networks provide an underlying framework over which different entities interact with each other. Through these interactions, network entities (nodes) can influence other entities (nodes) by propagating information/misinformation in the network. For instance, in a social network, a rumor formed by a person can spread to others through social interactions [1]. Similarly, a virus infection (either computer or biological) can propagate to different nodes in the network and become an epidemic [2]. Even a financial failure of an institution can have cascading effects on other financial entities and may lead to a financial crisis [3]. As a final example, in some human diseases, abnormal activities of few genes can cause their target genes and therefore some essential biological processes to fail to operate normally in the cell [4, 5].

In applications with underlying dynamic diffusion processes (e.g., an infection spread in an epidemic network), we wish to infer *source nodes* in the network by merely observing the information spread at single or multiple snapshots (Figure 1). In some other applications with static patterns in the network (e.g., patterns of disease-related genes in the regulatory network), we wish to infer *infusion hubs*, defined as nodes that explain the connectivity pattern of labeled nodes in the network optimally. Although these applications are inherently different, techniques to solve them are similar. In the sequel, we shall refer to the *source inference* problem to harmonize with the literature in the area, with the understanding that, we may be considering sources or infusion hubs.

The source inference problem seems on the surface difficult because real world diffusion dynamics are often unknown and there may be several diffusion processes that lead to the observed or similar samples of the information spread in the network. A standard continuous-time diffusion model for viral epidemics is known as the susceptible-infected-recovered (SIR) model [6], where infected nodes spread viruses to their neighbors probabilistically. Although the SIR diffusion model may be well-suited to model the forward problem of diffusion in the network, solving the inverse problem (the source inference problem) under this model is challenging in general except in few special cases [7–9], in great part owing to the presence of path multiplicity in the network. The case where multiple sources exist in the network has additional complexity, owing to combinatorial choices for source candidates (for more details, see Remarks 1, 2 and 3).

In this paper, we propose a computationally tractable general method for source inference called *Network Infusion* (NI). The key idea is to make source inferences based on a modified network diffusion kernel, which we term *path-based network diffusion*. Instead of the full network, our continuous-time network diffusion kernel considers k edge-disjoint shortest paths among pairs of nodes, neglecting other paths in the network, which leads to efficient kernel computation and NI algorithms. For instance, in a homogeneous diffusion setup, where node-to-node infection spread over all edges in the network has an exponential distribution, the proposed path-based network diffusion kernel can be characterized by an Erlang distribution. We use path-based network diffusion kernel to solve efficiently the inverse diffusion problem by maximizing the likelihood or minimizing the prediction error. The minimum error NI algorithm is based on an asymmetric Hamming premetric function and can balance between false positive and false negative error types. We apply this framework for both single-source and multi-source diffusion, for both single-snapshot and multi-snapshot observations, and using both uninformative and informative prior probabilities for candidate source nodes.

We prove that, under a standard susceptible-infected (SI) diffusion model,

- the maximum-likelihood NI algorithm is mean-field optimal for tree structures and sufficiently

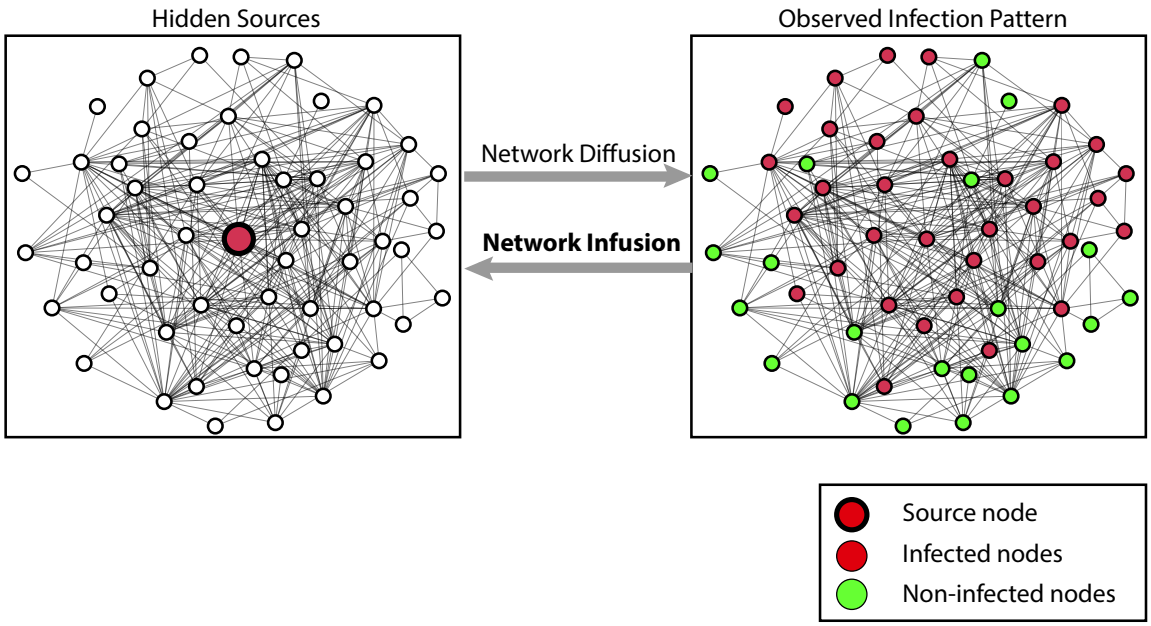

Figure 1: Network Infusion framework. NI aims to identify source node(s) by reversing information propagation in the network. The displayed network and infection pattern are parts of the Digg social news network.

sparse Erdös-Rényi graphs.

- the minimum-error NI algorithm is mean-field optimal for regular tree structures.
- the multi-source NI algorithm is mean-field optimal in the regular tree structure for sufficiently-distant sources.

The proposed path-based network diffusion kernel and NI algorithms can be extended to a complex heterogeneous diffusion setup as well, where edges propagate information/infection in the network according to different diffusion processes. In this setup, we show that, our network diffusion kernel can be characterized using the phase-type distribution of a Markov chain absorbing time (see Section 3.4). Moreover, we extend our NI algorithms to the cases with unknown or partially known diffusion model parameters such as observation times, by introducing techniques to learn these parameters from observed sample values (see Section 3.6).

We apply NI to several synthetic networks and compare its performance to degree centrality [10] and distance centrality [11] methods, under a standard SI diffusion model. We use four different network structures in our simulations: Erdös-Rényi graphs, power-law networks, symmetric and asymmetric grids. Our results indicate the superiority of proposed NI algorithms compared to existing methods, specially in sparse networks. Moreover, we apply NI to two real data applications: First, we identify the news sources for 3,553 stories in the Digg social news network, and validate our results based on annotated information, which was not provided to our algorithm. Second, we identify infusion hubs of several human diseases including Type 1 Diabetes (T1D), Systemic lupus erythematosus (SLE), Multiple sclerosis (MS), Parkinson, Psoriasis and Schizophrenia, and show

that, NI infers novel disease infusion hubs that are biologically relevant and often not identifiable using the raw p-values.

The rest of the paper is organized as follows. In Section 2, we introduce the problem, explain notation, and review prior work. In Section 3, we introduce NI methods and kernels for both homogeneous and heterogeneous models, and provide the main theorems and lemmas whose proofs are relegated to Appendix. We provide performance evaluation results over synthetic and real networks in Sections 4 and 5, respectively.

2 Problem Setup and Prior Work

In this section, we present the source inference problem and explain its underlying challenges. We also review prior work and present notation used in the rest of the paper.

2.1 Source Inference Problem Setup

Let $G = (V, E)$ be a binary, possibly directed, network with n nodes, where $G(i, j) = 1$ means that, there is an edge from node i to node j (i.e., $(i, j) \in E$). Suppose $\mathcal{N}(i)$ represents the set of neighbors of node i in the network. For the sake of description, we illustrate the problem setup and notation in the context of a virus infection spread in the network, with the understanding that, our framework can be used to solve a more general source or infusion hub inference problem. Consider source nodes $\mathcal{S} \subset V$ in the network. When a node gets infected, it starts to spread infection to its neighbors which causes the propagation of infection in the network. Let $\mathcal{T}_{(i,j)}$ be a random variable representing the virus traveling time over the edge $(i, j) \in E$ (i.e., the holding time variable of edge (i, j)). $\mathcal{T}_{(i,j)}$ variables are assumed to be mutually independent whose probability density functions are denoted by $f_{(i,j)}(.)$. Let $\mathcal{P}_{i\to j}$ denote a path (an ordered set of edges) connecting node i to node j in the network. $\mathcal{P}^{tot}_{i\to j}$ represents all paths between nodes i and j in the network. Similarly, we define $\mathcal{T}_{\mathcal{P}_{i\to j}}$ as a random variable representing the virus traveling time over the path $\mathcal{P}_{i\to j}$, with the following cumulative density function,

$$F_{\mathcal{P}_{i\to j}}(t) \triangleq Pr[\mathcal{T}_{\mathcal{P}_{i\to j}} \leq t]. \tag{2.1}$$

Let $\mathbf{y}(t) \in \{0, 1\}^n$ be the node infection vector at time t, where $y_i(t) = 1$ means that, node i is infected at time t. Suppose $\mathcal{T}_i$ is a random variable representing the time that node i gets infected. We assume that, if a node gets infected, it remains infected (i.e., there is no recovery). Suppose τ_i is a realization of the random variable $\mathcal{T}_i$. Thus, $y_i(t) = 1$ if $t \geq \tau_i$, otherwise $y_i(t) = 0$. If i is a source nodes, $\mathcal{T}_i = 0$ and $y_i(t) = 1$ for all $t \geq 0$. The set $V^t = \{i : y_i(t) = 1\}$ represents all nodes that are infected at time t. Thus, $\mathcal{S} = V^0$ represents the set of source nodes.

Definition 1 (SI Diffusion Model) *In a dynamic Susceptible-Infected (SI) diffusion setup, we have,*

$$\mathcal{T}_i \sim \min_{j\in\mathcal{N}(i)} (\mathcal{T}_j + \mathcal{T}_{(j,i)}). \tag{2.2}$$

Let $\{\mathbf{y}(t) : t \in (0, \infty)\}$ represent a continuous-time stationary stochastic process of diffusion in the network G. In the source inference problem, given the sample values at times $\{t_1, \ldots, t_z\}$ (i.e.,

$\{\mathbf{y}(t_1), \ldots, \mathbf{y}(t_z)\}$), as well as the underlying graph structure G, we wish to infer sources nodes that started the infection at time 0. We assume that, the number of sources to be inferred (i.e., m), and the observation time stamps (i.e., $\{t_1, \ldots, t_z\}$) are also given. We discuss the cases with unknown or partially known parameters in Section 3.6.

One way to formulate the source inference problem is to use a standard maximum a posteriori (MAP) estimation.

Definition 2 (MAP Source Inference) *The MAP source inference solves the following optimization:*

$$\begin{aligned} \arg\max \quad & Pr\big(\mathbf{y}(0)|\mathbf{y}(t_1), \ldots, \mathbf{y}(t_z)\big), \\ & \|\mathbf{y}(0)\|_{l_0} = m, \end{aligned} \tag{2.3}$$

where m is the number of source nodes in the network.

In some applications, there may be nonuniform prior probabilities for different candidate source nodes. The MAP source inference optimization takes into account these prior probabilities as well. If there is no informative prior probabilities for candidate source nodes, the MAP source Optimization (2.3) can be simplified to the following maximum likelihood (ML) source estimation:

Definition 3 (ML Source Inference) *The ML source inference solves the following optimization:*

$$\begin{aligned} \arg\max \quad & Pr\big(\mathbf{y}(t_1), \ldots, \mathbf{y}(t_z)|\mathbf{y}(0)\big), \\ & \|\mathbf{y}(0)\|_{l_0} = m, \end{aligned} \tag{2.4}$$

where its objective function is an ML function (score) of source candidates.

An alternative formulation for the source inference problem is based on minimizing the prediction error (instead of maximizing the likelihood). In Section 3.2, we shall propose a minimum prediction error formulation that uses an asymmetric Hamming pre-metric function and can balance between false positive and false negative error types by tuning a parameter.

In the following, we explain underlying challenges of the source inference problem.

Remark 1 *Suppose the underlying network G has 4 nodes and 3 undirected edges as depicted in Figure 2-a. Suppose the underlying diffusion is according to the SI model of Definition 1. Let the edge holding time variables $\mathcal{T}_{(i,j)}$ be mutually independent and be distributed according to an exponential distribution with parameter λ:*

$$f_{i,j}(\tau_{i,j}) = \lambda e^{-\lambda \tau_{i,j}}, \quad \forall (i,j) \in E. \tag{2.5}$$

Without loss of generality, let $\lambda = 1$. Suppose there is a single source in the network (i.e., $m = 1$), and we observe the infection pattern at a single snapshot at time t. Let the observed infection pattern at time t be $\mathbf{y}(t) = (1,1,1,0)$, implying that nodes $\{0,1,2\}$ are infected at time t, while node 3 is not infected at that time. Our goal is to find the most likely source node, according to the ML

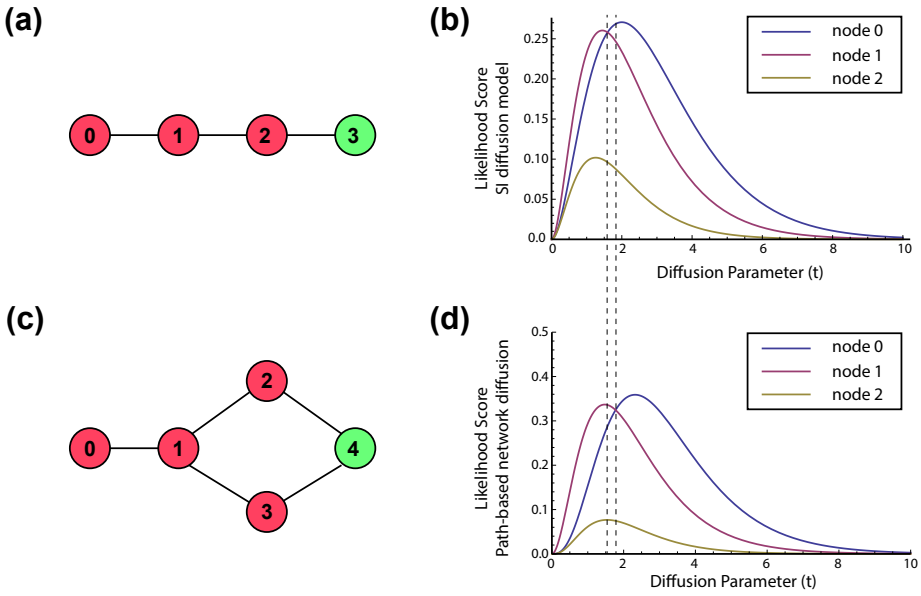

Figure 2: (a) A line graph considered in Remark 1. (b) Likelihood scores based on the SI diffusion model. (c) A graph considered in Remark 2. (d) Likelihood scores based on the path-based network diffusion kernel.

Optimization 2.4. Let τ_i be a realization of the random variable $\mathcal{T}_i$ (i.e., the time that virus arrives at node i). If node 0 was the source node (i.e., $\tau_0 = 0$), we would have $\tau_1 \leq \tau_2 \leq t \leq \tau_3$ because of the underlying network structure. Thus,

$$\begin{aligned} Pr\big(\mathbf{y}(t) = (1,1,1,0)|\mathbf{y}(0) = (1,0,0,0)\big) &= \int_{\tau_1=0}^{t}\int_{\tau_2=\tau_1}^{t}\int_{\tau_3=t}^{\infty} e^{-\tau_1}e^{-(\tau_2-\tau_1)}e^{-(\tau_3-\tau_2)}d\tau_1 d\tau_2 d\tau_3 \\ &= \int_{\tau_1=0}^{t}\int_{\tau_2=\tau_1}^{t}\int_{\tau_3=t}^{\infty} e^{-\tau_3}d\tau_1 d\tau_2 d\tau_3 \\ &= \frac{1}{2}t^2e^{-t}. \end{aligned} \tag{2.6}$$

Similarly, we have,

$$\begin{aligned} Pr\big(\mathbf{y}(t) = (1,1,1,0)|\mathbf{y}(0) = (0,1,0,0)\big) &= t(1-e^{-t})e^{-t}, \\ Pr\big(\mathbf{y}(t) = (1,1,1,0)|\mathbf{y}(0) = (0,0,1,0)\big) &= e^{-t} - (1+te^{-2t}). \end{aligned} \tag{2.7}$$

These likelihood functions are plotted in Figure 2-b. For a given observation time stamp t, the ML source estimator selects the node with the maximum likelihood score as the source node, according to Optimization 2.4. Note that, an optimal source solution depends on the observation time parameter t (i.e., for $t \lesssim 1.6$, node 1 and for $t \gtrsim 1.6$, node 0 are ML optimal source nodes.)

Remark 2 *Suppose G is a network with 5 nodes and 5 edges as shown in Figure 2-c. Consider the same diffusion setup as the one of Remark 1. Let $\mathbf{y}(t) = (1,1,1,1,0)$; i.e., nodes $\{0,1,2,3\}$ are infected at time t while node 4 is not infected at that time. Similarly to Remark 1, let τ_i be a*

realization of the random variable $\mathcal{T}_i$, a variable representing the time that virus arrives at node i. If node 0 was the source node (i.e., $\tau_0 = 0$), we would have $\tau_1 \leq \min(\tau_2, \tau_3) \leq \tau_4$, $\max(\tau_2, \tau_3) \leq t$, and $\tau_4 > t$. Thus,

$$\begin{aligned} &Pr\big(\mathbf{y}(t) = (1,1,1,1,0) | \mathbf{y}(0) = (1,0,0,0,0)\big) = \\ &= \int_{\tau_1=0}^{t} \int_{\tau_2=\tau_1}^{t} \int_{\tau_3=\tau_1}^{t} \int_{\tau_4=\min(\tau_2,\tau_3)}^{\infty} e^{-\tau_1} e^{-(\tau_2-\tau_1)} e^{-(\tau_3-\tau_1)} e^{-(\tau_4-\min(\tau_2,\tau_3))} d\tau_1 d\tau_2 d\tau_3 d\tau_4 \\ &= 2e^{-t} - e^{-2t}\big(1 + (1+t)^2\big). \end{aligned} \tag{2.8}$$

In this case, likelihood computation is more complicated than the case of Remark 1, because both variables $\mathcal{T}_2$ and $\mathcal{T}_3$ depend on $\mathcal{T}_1$, and therefore, consecutive terms do not cancel as in (2.6). Moreover, note that, there are two paths from node 0 to node 4 that overlap at edge $(0,1)$. As we have mentioned earlier, such overlaps are source of difficulties in the source inference problem, which is illustrated by this simplest example, because the variable $\mathcal{T}_4$ depends on both variables $\mathcal{T}_2$ and $\mathcal{T}_3$ through a $\min(.,.)$ function which makes computation of the likelihood integral further complicated.

Remark 3 *Remarks 1 and 2 explain underlying source inference challenges for the case of having a single source node in the network. The case of having multiple source nodes has additional complexity because likelihood scores of Optimization (2.4) should be computed for all possible subsets of infected nodes. For the case of having m sources in the network, there are $\binom{|V^t|}{m}$ candidate source sets where for each of them, a likelihood score should be computed. If there are significant number of infected nodes in the network (i.e., $V^t = \mathcal{O}(n)$), there would be $\mathcal{O}(n^m)$ source candidate sets. This makes the multi-source inference problem computationally expensive for large networks, even for small values of m.*

Moreover, in Remarks 1 and 2, we assume that, the edge holding time distribution is known and follows an exponential distribution with the same parameter for all edges. This is the standard diffusion model used in the most of epidemic studies [12], because the exponential distribution has a single parameter and is memoryless. However, in some practical applications, the edge holding time distribution may be unknown and/or may vary for different edges. We discuss this case in Section 3.4.

In the next part of this section, we explain the prior work on information propagation and source inference.

2.2 Prior work

While our approach considers a general network diffusion setup and its inverse problem, most of the literature considers the applications to specific problems. The most common ones focus on studying different models of virus propagation in population networks. A standard information diffusion model in this setup is known as the susceptible-infected-recovered (SIR) model [6]. There are three types of nodes in this model: susceptible nodes which are capable of getting infected, infected nodes that spread virus in the network, and recovered nodes that are cured and can no longer become infected. Under the SIR diffusion model, infection spreads from sources to susceptible nodes probabilistically. References [2, 12–14] discuss the relationship among network structure,

infection rate, and the size of the epidemics under this diffusion model. Learning different diffusion parameters of this model have been considered in references [15–17]. Some other diffusion methods use random walks to model information spread and label propagation in networks [18–20]. In these methods, a random walker goes to a neighbor node with a probability inversely related to node degrees. Therefore, high degree nodes may be less influential in information spread in the network which may be counter-intuitive in some applications.

Although there are several works on understanding mechanisms of diffusion processes in different networks, there is somehow little work on studying the inverse diffusion problem to infer information sources, in great part owing to the presence of path multiplicity in the network [7], that we described in Remarks 1, 2, and 3. Recently, reference [9] considers the inverse problem of a diffusion process in a network under a discrete time memoryless diffusion model, and when time steps are known. Their discrete time diffusion model is different than the continuous time dynamic diffusion setup considered in this paper.

For the case of having a single source node in the network, some methods infer the source node based on distance centrality [11], or degree centrality [10] measures of the infected subgraph. These methods are efficient to apply to large networks. However, because they do not assume any particular diffusion model, their performance lacks provable guarantees in general. For tree structures and under a homogeneous SI diffusion model, reference [8] computes a maximum likelihood solution for the source inference problem and provides provable guarantees for its performance. Over tree structures, their solution is in fact equivalent to the distance centrality of the infected subgraph. Our approach considers a diffusion model that is less restrictive than the SI model considered in [8], and that can be computed efficiently for large complex networks, similarly to the distance-based and degree-based centrality methods. Moreover, unlike distance-based and degree-based centrality methods, we provide provable performance guarantees for our approach under a continuous-time dynamic SI diffusion setup of reference [8]. The existent source inference methods only consider the case of having a single-source in the network. As we have explained in Remark 3, the case of having more than one source in the network has additional combinatorial complexity. In Section 3.3, we show that, our framework can be used efficiently for multi-source inference problem under some general conditions. Furthermore, the existent methods only consider homogeneous diffusion setups, where all edges propagate information/infection in the network according to the same diffusion process. Our framework can be extended to solve the source inference problem even under a complex heterogeneous diffusion setup, that we explain in Section 3.4.

3 Main Results

In this section, first, we introduce a path-based network diffusion kernel which is used in proposed source inference methods. Then, we present global and local NI algorithms to infer single and multiple sources in the network, respectively. Finally, we present NI algorithms for heterogeneous diffusion, multi-snapshot, and non-parametric cases. We provide the main theorems and lemmas in this section while all proofs are relegated to Appendix. For the sake of description, we shall, as before, have a recurrent example of the virus infection spread in the network, with the understanding that, our framework can be used to solve a more general source or infusion hub inference problem.

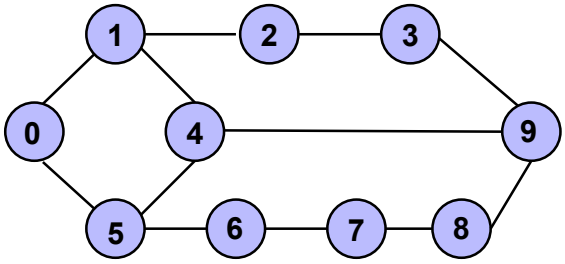

Figure 3: An example graph with overlapping shortest paths between nodes 0 and 9.

3.1 Path-based Network Diffusion Kernel

In this section, we consider the case when there exists a single source node in the network. The multi-source inference problem is considered in Section 3.3. Suppose the network structure $G = (V, E)$ is given, and we observe a snapshot $\mathbf{y}(t)$ from the *real* diffusion dynamics at time t. In general, there may be several diffusion processes that lead to the observed infection snapshot in the network, either exactly or approximately. Suppose $\hat{\mathbf{y}}(t')$ is the sample generated at time t' using a certain diffusion model. One way to characterize the error of this diffusion model to explain the observed diffusion sample is to use an asymmetric Hamming premetric function as follows:

$$\min_{t'} h_\alpha(\mathbf{y}(t), \hat{\mathbf{y}}(t')) \triangleq (1-\alpha) \sum_{i:y_i(t)=1} \mathbb{1}_{\hat{y}_i=0} + \alpha \sum_{i:y_i(t)=0} \mathbb{1}_{\hat{y}_i(t')=1}, \tag{3.1}$$

where $0 \leq \alpha \leq 1$. This error metric assigns weight α to false positive and weight $1-\alpha$ to false negatives error types. To solve the inverse problem, one may select a diffusion process which approximates the observed diffusion pattern closely and also leads to a tractable source inference method. Although the SI diffusion model may be well-suited to model the forward problem of information diffusion in the network, solving the inverse problem (the source inference problem) under this model is challenging in general, in great part owing to the presence of path multiplicity in the network, as we explain in Remarks 1, 2 and 3. Here, we present a path-based network diffusion kernel that is distinct from the standard SI diffusion models, but its order of diffusion approximates well many of them. We will show that, this kernel leads to an efficient source inference method with theoretical performance guarantees, under some general conditions, even if the underlying diffusion model is different than the one considered in the method itself.

In our diffusion model, instead of the full network, we consider up to k edge-disjoint shortest paths among pairs of nodes, neglecting other paths in the network. Suppose $\mathcal{P}^1_{i\to j}$, $\mathcal{P}^2_{i\to j}$, ... represent different paths between nodes i and j in the network. The length of a path $\mathcal{P}^r_{i\to j}$ is denoted by $|\mathcal{P}^r_{i\to j}|$. Let $E^r_{i\to j}$ be the set of edges of the path $\mathcal{P}^r_{i\to j}$. We say two paths are edge-disjoint if the set of their edges do not overlap. Let $\{\mathcal{P}^1_{i\to j}, \mathcal{P}^2_{i\to j}, \ldots, \mathcal{P}^k_{i\to j}\}$ represent k disjoint shortest paths between nodes i and j. We choose these paths iteratively so that,

- $|\mathcal{P}^1_{i\to j}| \leq |\mathcal{P}^2_{i\to j}| \leq \ldots \leq |\mathcal{P}^k_{i\to j}|$,
- paths are disjoint. I.e., for $1 < r \leq k$, $E^r_{i\to j} \bigcap \left(\bigcup_{a=1}^{r-1} E^a_{i\to j} \right) = \varnothing$,

- $\mathcal{P}^r_{i\to j}$ is a shortest path between nodes i and j in the network $G' = (V, E - \bigcup_{a=1}^{r-1} E^a_{i\to j})$.

In some cases, the shortest path solutions may not be unique. That is, there are at least two shortest paths connecting nodes i to j in the network. If these shortest paths do not overlap, the resulting path length vector $(|\mathcal{P}^1_{i\to j}|, \ldots, |\mathcal{P}^k_{i\to j}|)$ is the same irrespective of the selection order. Thus, the tie breaking can be done randomly. However, in the case of having overlapping shortest paths, one way to break the tie among these paths is to choose the one which leads to a shorter path in the next step. For example, consider the network depicted in Figure 3. There are two paths of length 3 between nodes 0 and 9. Choosing the path $0-5-4-9$ leads to the next independent path $0-1-2-3-9$ with length 4, while choosing the path $0-1-4-9$ leads to the next path $0-5-6-7-8-9$ of length 5. Thus, the algorithm chooses the path $0-5-4-9$. If next paths have the same length, tie would be broken considering more future steps. In practice, this case has negligible effect in the performance of the source inference method. Methods based on message-passing or dynamic programming can be used to select optimal k shortest paths in the network as well [21, 22]. In this paper, we break ties randomly among paths with the same length.

Recall that, $\mathcal{T}_{\mathcal{P}^r_{i\to j}}$ represents the virus traveling time over the path $\mathcal{P}^r_{i\to j}$ whose cumulative density function is denoted by $F_{\mathcal{P}^r_{i\to j}}(.)$ according to Equation (2.1).

Definition 4 (Path-based network diffusion kernel) *Let $p_{i,j}(t)$ be the probability of node j being infected at time t if node i is the source node. Thus,*

$$\begin{aligned} p_{i,j}(t) =& Pr\big[y_j(t) = 1 | y_i(0) = 1\big] \\ \triangleq& 1 - \prod_{r=1}^{k} 1 - F_{\mathcal{P}^r_{i\to j}}(t), \end{aligned} \tag{3.2}$$

where k is the number of independent shortest paths between nodes i and j. $P(t) = [p_{i,j}(t)]$ is called a path-based network diffusion kernel.

The path-based diffusion kernel indicates that, node j gets infected at time t if the infection reaches to it over at least one of the k independent shortest paths connecting that node to the source node. The path-based network diffusion kernel provides a non-dynamic diffusion basis for the network and is based on two important assumptions that, edge holding time variables $\mathcal{T}_{(i,j)}$ are mutually independent, and the paths are disjoint. A path-based network diffusion kernel with $k = 1$ only considers the shortest paths in the network and has the least computational complexity among other path-based network diffusion kernels. Considering more paths among nodes in the network (i.e., $k > 1$) can provide a better characterization of network diffusion processes with the cost of increased kernel computational complexity (see Proposition 2). A path-based network diffusion kernel lies at the heart of our algorithms to solve the inverse diffusion problem. We show that, even if the underlying diffusion model is according to a SI model of Definition 1, using a k-path network diffusion kernel to solve the inverse diffusion problem provides a robust source estimation under some general conditions.

In the following, we highlight properties and relaxations of the path-based network diffusion kernel:

Remark 4 *The path-based network diffusion kernel provides a non-dynamic diffusion model, where nodes become infected independently based on their distances (path lengths) to source nodes. Unlike*

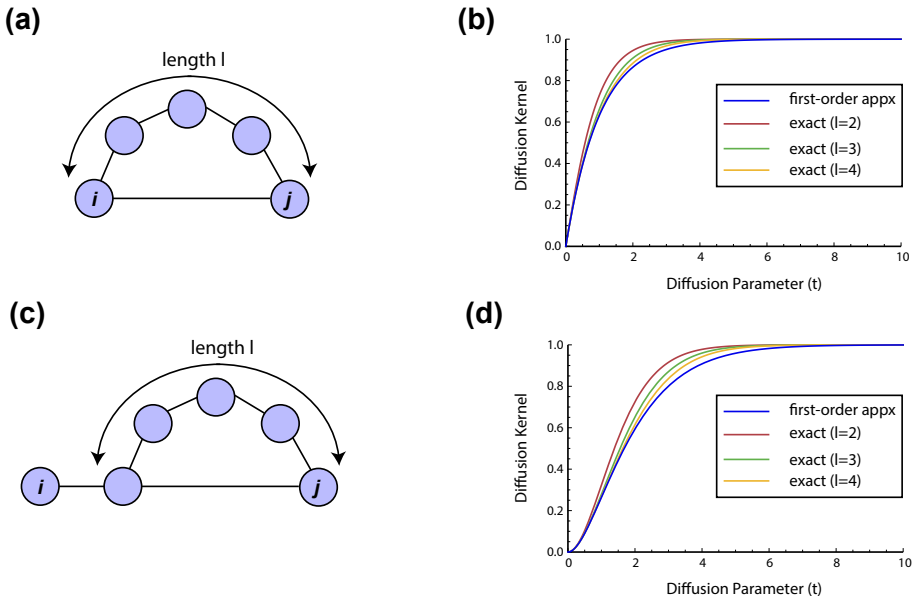

Figure 4: (b),(d) Tightness of the first order approximation of the path-based network diffusion kernel over example networks depicted in panels (a) and (c), respectively.

the dynamic SI model, in the path network diffusion model, it is possible (though unlikely) to have $y_i(t) = 1$ *while* $y_j(t) = 0$*, for all neighbors of node* i *(i.e.,* $j \in \mathcal{N}(i)$*). The key idea is that, to infer the source node in the network, full characterization of diffusion dynamics, in many cases, may not be necessary as long as the diffusion model approximates the observed samples closely (e.g., according to an error metric of (3.1)). For instance, consider the setup of Remark 1 where the underlying diffusion model is according to a SI model. In that example, we compute source likelihood scores in (2.6) and (2.7) by integrating likelihood conditional density functions. The likelihood computation under this model becomes challenging for complex networks. However, according to the path-based network diffusion model of Definition 4, these likelihood scores are decoupled to separate terms and can be computed efficiently as follows:*

$$\begin{aligned} Pr\big(\mathbf{y}(t) = (1,1,1,0)|\mathbf{y}(0) = (1,0,0,0)\big) &= F(1,t)F(2,t)\bar{F}(3,t), \\ Pr\big(\mathbf{y}(t) = (1,1,1,0)|\mathbf{y}(0) = (0,1,0,0)\big) &= F(1,t)^2\bar{F}(2,t), \\ Pr\big(\mathbf{y}(t) = (1,1,1,0)|\mathbf{y}(0) = (0,0,1,0)\big) &= F(1,t)F(2,t)\bar{F}(1,t), \end{aligned} \tag{3.3}$$

where $F(l,t)$ *is the Erlang cumulative distribution function over a path of length* l*, that we shall show in (3.6). Figure 2-d shows likelihood scores of infected nodes computed according to (3.3). This example illustrates that, for a wide range of parameter* t*, both models lead to the same optimal solution. Moreover, unlike the SI model, likelihood functions can be computed efficiently using the path-based network diffusion kernel, even for large complex networks.*

Remark 5 *The path-based diffusion kernel considers only the top* k *shortest paths among nodes, neglecting other paths in the networks. The effect of long paths is dominated by the one of short ones leading to a tight approximation. Suppose* $\mathcal{P}^1_{i\to j}$ *and* $\mathcal{P}^2_{i\to j}$ *represents two paths between nodes*

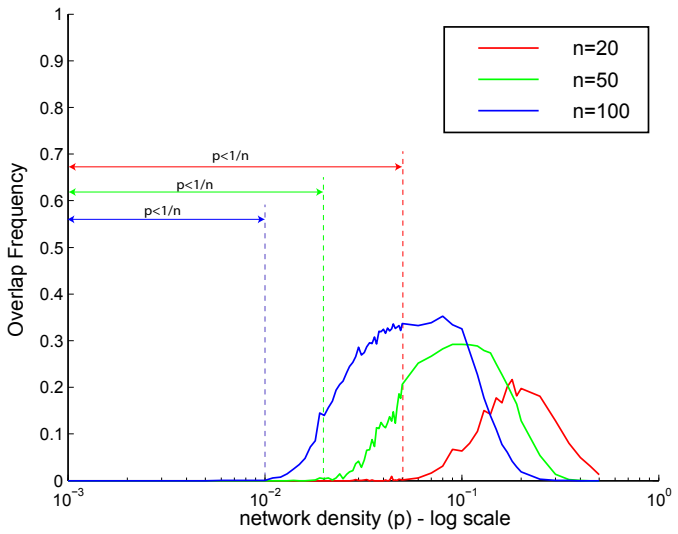

Figure 5: The frequency of having overlapping shortest paths between two randomly selected nodes over an Erdös-Rényi graph with parameter p. In sparse graph ($p \leq \frac{1}{n}$), the overlap frequency is small. Experiments are repeated 20,000 times for each case.

i and j where $|\mathcal{P}^1_{i\to j}| \ll |\mathcal{P}^2_{i\to j}|$ (i.e., the path $\mathcal{P}^2_{i\to j}$ is much longer than the path $\mathcal{P}^1_{i\to j}$). Thus, for a wide range of parameter t, we have $F_{\mathcal{P}^1_{i\to j}}(t) \gg F_{\mathcal{P}^2_{i\to j}}(t)$, and therefore,

$$(1 - F_{\mathcal{P}^1_{i\to j}}(t))(1 - F_{\mathcal{P}^2_{i\to j}}(t)) \approx 1 - F_{\mathcal{P}^1_{i\to j}}(t). \tag{3.4}$$

Note that, for very small or large t values (i.e., $t \to 0$ or $t \to \infty$), both $F_{\mathcal{P}^1_{i\to j}}(.)$ and $F_{\mathcal{P}^2_{i\to j}}(.)$ go to 0 and 1, respectively, and thus the approximation (3.4) remains tight. For an example network depicted in Figure 4-a, we illustrate the tightness of the first order approximation (i.e., $k = 1$) for different lengths of the path $\mathcal{P}^2_{i\to j}$. In general, for large k values, the gap between the approximate and the exact kernels becomes small with the cost of increased kernel computational complexity (see Proposition 2). The same approximation holds for overlapping paths with different lengths as it is illustrated in Figures 4-c,d.

Remark 6 *Finally, path-based network diffusion kernel only considers independent shortest paths among nodes and therefore ignores the effects of non-disjoint paths in the network. This is a critical relaxation because, as we explain in Remark 2, overlapping paths and dependent variables make the likelihood computation and therefore source inference challenging. In general, if there are many overlapping shortest paths among nodes in the network, this approximation might not be tight. However, in network structures whose paths do not overlap significantly (for example tree structures), this approximation is tight. In the following proposition, we show that, in a common model for sparse random networks [23], shortest paths among nodes are extremely unlikely to overlap with each other, leading to a tight kernel approximation.*

Proposition 1 *Let $G = (V, E)$ be an undirected Erdös-Rényi graph with n nodes where $Pr[(i, j) \in E] = p$. Consider two nodes i and j where $d(i, j) \leq l_0$. If $p < \frac{c}{n}$ where $c = 1/n^{2l_0}$, the probability of having overlapping shortest paths between nodes i and j goes to zero asymptotically.*

Proof *The proof is presented in Appendix 7.1.* ■

Figure 5 illustrates Proposition 1 for Erdös-Rényi graphs with different number of nodes and different parameters p. As illustrated in this figure, shortest paths are less likely to overlap in sparse networks. Moreover, in very dense networks, which practically might be less interesting, the shortest path overlap probability decreases as well, because most node pairs are connected by one-hop or two-hop paths.

One main advantage of using the path-based network diffusion kernel compared to other diffusion models such as the SI diffusion model is its efficient computation even for large and complex networks:

Proposition 2 (Computational complexity of path-based network diffusion kernel) *Let $G = (V, E)$ be a directed network with n nodes and $|E|$ edges. Then, computation of the k-path network diffusion kernel of Definition 4 has a worst case computational complexity $\mathcal{O}(k|E|n + kn^2 \log(n))$.*

Proof The proof is presented in Appendix 7.2. ■

Remark 7 *To solve the source inference problem, one only needs to compute rows of the path-based network diffusion kernel which correspond to infected nodes ($i \in V^t$). Thus, time complexity of kernel computation can be reduced to $\mathcal{O}(k|E||V^t| + k|V^t|n\log(n))$, where $|V^t|$ is the number of observed infected nodes in the network at time t.*

Computation of the path-based network diffusion kernel depends on edge holding time distributions. If virus travelling time variables $\mathcal{T}_{(i,j)}$ are i.i.d. for all edges in the network, the underlying diffusion process is called *homogeneous*. On the other hand, if holding time distributions differ among edges in the network, the resulting diffusion process is *heterogeneous*. In this section, we consider a homogeneous diffusion setup, where the holding time distribution is an exponential distribution with the same parameter λ for all edges. Without loss of generality, we assume that $\lambda = 1$. The case of heterogeneous diffusion is considered in Section 3.4. Under the setup considered in this section, the virus traveling time over each path in the network has an Erlang distribution, because it is the sum of independent exponential variables. Thus, we have,

$$F_{\mathcal{P}^r_{i\to j}}(t) = Pr[\mathcal{T}_{\mathcal{P}^r_{i\to j}} \leq t] = \frac{\gamma(|\mathcal{P}^r_{i\to j}|, \lambda t)}{(|\mathcal{P}^r_{i\to j}| - 1)!}, \tag{3.5}$$

where $\gamma(.)$ is the lower incomplete gamma function. $|\mathcal{P}^r_{i\to j}|$ (the path length connecting node i to j) is also called the Erlang's shape parameter. Because $F_{\mathcal{P}^r_{i\to j}}(t)$ is only a function of the path length and parameter t, to simplify notation, we define,

$$F(l, t) \triangleq F_{\mathcal{P}^r_{i\to j}}(t), \tag{3.6}$$

where $l = |\mathcal{P}^r_{i\to j}|$. The k-path network diffusion kernel of Definition 4 using the Erlang distribution is called a path-based Erlang network diffusion kernel. If only one shortest path among nodes is considered (i.e., $k = 1$), the diffusion kernel of Definition 4 is called the shortest path network diffusion kernel.

Definition 5 (Shortest path Erlang diffusion kernel) *Let $p_{i,j}(t)$ be the probability of node j being infected at time t if node i is the source node. Suppose edge holding time variables are distributed independently according to an exponential distribution with the parameter $\lambda = 1$. The shortest path Erlang network diffusion kernel is defined as follows:*

$$p_{i,j}(t) = Pr\big[y_j(t) = 1 | y_i(0) = 1\big] = F(d_{i,j}, t), \tag{3.7}$$

where $d_{i,j}$ is the length of the shortest path connecting node i to node j, and $F(d_{i,j}, t)$ represents the Erlang cumulative distribution function of (3.6).

The shortest path Erlang diffusion kernel can be viewed as the first order approximation of the underlying diffusion process. It has the least computational complexity among other path-based network diffusion kernels which makes it suitable to be used over large and complex networks. Moreover, this kernel has a single parameter t which can be learned reliably using the observed samples (see Section 3.6).

Proposition 3 *The shortest path Erlang network diffusion kernel of Definition 5 has following properties:*

- *$p_{i,j}(t)$ is a decreasing function of $d_{i,j}$, the length of the shortest path between nodes i and j.*
- *$p_{i,j}(t)$ is a decreasing function of t and,*

$$\frac{\partial p_{i,j}(t)}{\partial t} = F(d_{i,j} - 1, t) - F(d_{i,j}, t), \tag{3.8}$$

where $F(0, t) = 1$.

Proof The proof is presented in Appendix 7.3. ■

3.2 Global Source Inference Using NI

In this section, we describe a source inference method called Network Infusion (NI) which aims to solve the inverse diffusion problem over a given network using observed infection patterns. The method described in this section finds a single node as the source of the global information spread in the network. In Section 3.3, we consider the case when more than one source node exists, where each source causes a local infection propagation in the network. In this section, we also assume that, the infection pattern is observed at a single snapshot at time t (i.e., $\mathbf{y}(t)$ is given). The case of having multiple snapshots is considered in Section 3.5.

Recall that, V^t is the set of observed infected nodes at time t, and $P(t) = [p_{i,j}(t)]$ represents the path-based network diffusion kernel according to Definition 4. The ML Optimization 2.4 can be re-written as follows:

Algorithm 1 (Maximum Likelihood NI) *Suppose $G = (V, E)$ is a binary graph with n nodes. Let $P(t) = [p_{i,j}(t)]$ be the path-based network diffusion kernel according to Definition 4. Then, a maximum-likelihood NI algorithm infers the source node by solving the following optimization:*

$$\arg\max_{i\in V^t} \mathcal{L}(i,t) = \arg\max_{i\in V^t} \sum_{j\in V^t} \log\big(p_{i,j}(t)\big) + \sum_{j\notin V^t} \log\big(1-p_{i,j}(t)\big), \tag{3.9}$$

where $\mathcal{L}(i,t)$ is the log-likelihood function of node i at time t.

Under the path-based network diffusion kernel, the joint diffusion probability distribution can be de-coupled into individual marginal distributions, which leads to a tractable ML Optimization (3.9), even for complex networks. Note that, in Optimization (3.9), we assume that, the parameter t (the time at which the observation is made) is known. If this parameter is unknown, it can be learned using the observed infection pattern. We discuss this case in Section 3.6.

Remark 8 *To have a well-defined log-likelihood objective function, $p_{i,j}(t)$ should be non-zero for infected nodes i and j (i.e., $p_{i,j}(t) \neq 0$ when $i,j \in V^t$). If infected nodes form a strongly connected sub-graph over the network, this condition is always satisfied. In practice, if $p_{i,j}(t) = 0$ for some $i,j \in V^t$ (i.e., i and j are disconnected in the graph), we assume that, $p_{i,j}(t) = \epsilon$ (e.g., $\epsilon = 10^{-6}$). Note that, for $j \notin V^t$, for any value of $t > 0$, $p_{i,j}(t) < 1$. Therefore, the second term in the summation of Optimization (3.9) is always well-defined.*

NI Algorithm 1 aims to infer the source node by maximizing the likelihood score assigned to each node. An alternative approach is to infer the source node by minimizing the expected prediction error of the observed infection pattern. We describe this approach in the following:

Let $h_\alpha(\mathbf{y},\mathbf{x})$ be a weighted Hamming premetric between two binary sequences $\mathbf{x}$ and $\mathbf{y}$ defined as follows:

$$h_\alpha(\mathbf{y},\mathbf{x}) \triangleq (1-\alpha) \sum_{i:y_i=1} \mathbb{1}_{x_i=0} + \alpha \sum_{i:y_i=0} \mathbb{1}_{x_i=1}, \tag{3.10}$$

where $0 \leq \alpha \leq 1$.

If $\alpha = 1/2$, $h_\alpha(.,.)$ is a metric distance. If $\alpha \neq 1/2$, $h_\alpha(.,.)$ is a premetric (not a metric) because it does not satisfy the symmetric property of distance metrics (i.e., there exist $\mathbf{x}$ and $\mathbf{y}$ such that $h_\alpha(\mathbf{x},\mathbf{y}) \neq h_\alpha(\mathbf{y},\mathbf{x})$), and it does not satisfy the triangle inequality as well (i.e., there exist $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ such that $h_\alpha(\mathbf{x},\mathbf{y}) > h_\alpha(\mathbf{y},\mathbf{z}) + h_\alpha(\mathbf{z},\mathbf{x})$).

Remark 9 *$h_\alpha(\mathbf{y},\mathbf{x})$ generalizes Hamming distance between binary sequences $\mathbf{y}$ and $\mathbf{x}$ using different weights for different error types. Suppose $\mathbf{x}$ is a prediction of the sequence $\mathbf{y}$. There are two types of possible errors: (1) if $y_i = 1$ and the prediction is zero (i.e., $x_i = 0$, false negative error), (2) if $y_i = 0$ and the prediction is one (i.e., $x_i = 1$, false positive error). $h_\alpha(\mathbf{y},\mathbf{x})$ combines these errors by assigning weight $1-\alpha$ to false negatives (i.e., missing ones), and weight α to false positives (i.e., missing zeros). Having different weights for different error types can be useful, specially if the sequence $\mathbf{y}$ is sparse. Suppose $\mathbf{y}$ has κ ones (positives), and $n-\kappa$ zeros (negatives). Therefore, there are κ possible type 1 errors and $n-\kappa$ type 2 errors in prediction. In this case, to have balance between the number of true negative and false positive errors, one can choose $\alpha = \kappa/n$ in calculation of $h_\alpha(\mathbf{y},\mathbf{x})$.*

Now we introduce a NI algorithm which infers the source node by minimizing the prediction error.

Algorithm 2 (Minimum Error NI) *Suppose $G = (V, E)$ is a binary graph with n nodes. Let $P(t) = [p_{i,j}(t)]$ be the path-based network diffusion kernel according to Definition 4 and $P_i(t) = [p_{i,j}(t)]$ be the kernel vector of node i. Then, the minimum error NI algorithm infers the source node by solving the following optimization:*

$$\begin{aligned}\arg\min_{i\in V^t} \mathcal{H}_\alpha(i,t) &= \arg\min_{i\in V^t} \mathbb{E}[h_\alpha(\mathbf{y}(t), \mathbf{x}_i(t))] \\ &= \arg\min_{i\in V^t} (1-\alpha) \sum_{j\in V^t} 1 - p_{i,j}(t) + \alpha \sum_{j\notin V^t} p_{i,j}(t),\end{aligned} \tag{3.11}$$

where $\mathbf{x}_i(t)$ is a binary prediction vector of node i at time t with probability distribution $P_i(t)$, and $\mathcal{H}_\alpha(i,t)$ is the expected prediction error of node i at time t.

Similarly to Maximum Likelihood NI (NI-ML) Algorithm, we assume that, the parameter t (the time at which observation is made) is known. We discuss the case when this parameter is unknown in Section 3.6.

Remark 10 *According to Remark 9, to have balance between false positive and false negative error types, one can use $\alpha = |V^t|/n$ where $|V^t|$ is the number of infected nodes (positives) at time t. However, in general, this parameter can be tuned in different applications using standard machine learning techniques such as cross validations [24].*

The proposed NI methods based on maximum likelihood (NI-ML, Algorithm 1) and minimum error (NI-ME, Algorithm 2) are efficient to solve even for large complex networks:

Proposition 4 *Suppose the underlying network $G = (V, E)$ has n nodes and $|E|$ edges. Let V^t represent the set of infected nodes at time t. Then, a worst case computational complexity of NI Algorithms 1 and 2 is $\mathcal{O}\big(|V^t|(k|E| + kn\log(n))\big)$.*

Proof The proof is presented in Appendix 7.4. ■

In the rest of this section, we analyze the performance of NI Algorithms 1 and 2 under a standard SI diffusion model of Definition 1.

Theorem 1 *Let $G = (V, E)$ be an undirected tree with countably infinite nodes. Suppose node s is the source node, t is the infection observation time, and the underlying diffusion process is according to the SI model of Definition 1. Then, we have,*

$$\mathbb{E}[\mathcal{L}(s,t)] \geq \mathbb{E}[\mathcal{L}(i,t')], \quad \forall i, \forall t', \tag{3.12}$$

where $\mathbb{E}[\mathcal{L}(i,t')]$ is the expected log-likelihood score of node i with parameter t'.

Proof The proof is presented in Appendix 7.5. ■

In the setup of Theorem 1, similarly to the setup of reference [8], we assume that, the set of vertices is countably infinite to avoid boundary effects. Theorem 1 provides a mean-field (expected) optimality for Algorithm 1. In words, it considers the case when we have sufficient samples from independent infection spreads in the network starting from the same source node. Note that, in (3.12), i can be equal to s (the source node), and/or t' can be equal to t as well. If i is equal to s, we have the following:

Proposition 5 *Under the conditions of Theorem 1, we have,*

$$\mathbb{E}[\mathcal{L}(s,t)] \geq \mathbb{E}[\mathcal{L}(s,t')], \quad \forall t', \tag{3.13}$$

where the equality holds iff $t = t'$.

Proof The proof is presented in Appendix 7.5. ■

Remark 11 *In this remark, we highlight the difference between parameters t and t' in Theorem 1. The parameter t is the time at which we observe the infection pattern in the network. If this parameter is known, it can be used to compute likelihood scores according to Optimization (3.9). However, this parameter may be unknown and one may use an estimate of this parameter in Optimization (3.9) (i.e., using t' instead of t). Theorem 1 indicates that even if different parameters $t' \neq t$ are used to compute source likelihood scores for different nodes, the likelihood score obtained by the source node s and the true parameter t is optimal in expectation. This theorem and corresponding Proposition 5 provide a theoretical basis to estimate the underlying true parameter t by maximizing the likelihood score for each node over different values of t' (for more details, see Section 3.6).*

Next, we show the mean-field optimality of Algorithm 1 for sparse Erdös-Rényi graphs utilizing their local tree structure. For a given $\epsilon > 0$ and t, we define,

$$l_0 \triangleq \arg\min_d F(d,t) < \epsilon, \tag{3.14}$$

where $F(.,.)$ is the Erlang distribution of Equation (3.5).

Proposition 6 *Let $G = (V,E)$ be an undirected Erdös-Rényi graph with n nodes, where $Pr[(i,j) \in E] = p$. If $p < \frac{c}{n}$ where $c = 1/n^{2l_0}$ and l_0 is defined according to (3.14), the mean-field optimality inequality (3.12) holds for asymptotically large graphs with high probability.*

Proof The proof follows from Proposition 1 and Theorem 1. ■

In the following, we present the mean-field optimality of minimum error NI algorithm (NI-ME) over regular tree structures:

Theorem 2 *Let $G = (V,E)$ be a regular undirected tree with countably infinite nodes. Suppose node s is the source node, t is the observation time, and the underlying diffusion process is according to the SI model of Definition 1. Then, for any value of $0 < \alpha < 1$ and $t' > 0$, we have,*

$$\mathbb{E}[H_\alpha(s,t')] < \mathbb{E}[H_\alpha(i,t')], \quad \forall i \neq s, \forall t' > 0, \tag{3.15}$$

where $\mathbb{E}[H_\alpha(i,t')]$ is the expected prediction error of node i using parameter t'. Equality (3.15) holds iff $s = i$.

Proof The proof is presented in Appendix 7.6. ■

Remark 12 *The mean field optimality of NI-ME algorithm holds for all values of* $0 < \alpha < 1$ *under the setup of Theorem 2. In practice and under more general conditions, we find that,* α *selection according to Remarks 9 and 10 leads to a robust performance, owing to the balance between true negative and false positive errors (see Sections 4 and 5).*

Remark 13 *The NI-ML mean-field optimality of Theorem 1 holds even if different* t' *values are used for different nodes. However, the mean-field optimality of the NI-ME method of Theorem 2 holds if the same* t' *parameter is used for all nodes. Interestingly, even if the parameter used in the NI-ME algorithm is difference than the true observation time parameter (i.e.,* $t' \neq t$*), the optimality argument of Theorem 2 holds which indicates the robustness of the method with respect to this parameter. Moreover, the NI-ME optimality of inequality (3.15) is strict, while the one of NI-ML method according to the inequality (3.12) may have multiple optimal solutions.*

3.3 Localized NI

In this section, we consider the multi-source inference problem, where there exists m sources in the network. We consider this problem when sources are sufficiently distant from each other and only a single snapshot, at time t, is available (i.e., $\mathbf{y}(t)$ is given). For simplicity of the analysis, we consider $k = 1$ in the path-based network diffusion kernel (i.e., only shortest paths are considered).

Let $G = (V, E)$ be the underlying network where $d_{i,j}$ represents the length of the shortest path between node i and node j. Define $D(i, R) \triangleq \{j \in V | d_{i,j} < R\}$ as a disk with radius R centered at node i, which we refer to as the R-neighborhood of node i in the network. Similarly, the union of disks with radius R centered at nodes of the set $V_1 \subset V$ is defined as $D(V_1, R) \triangleq \{j \in V | \exists i \in V_1, d_{i,j} < R\}$. We define the following distances in the network:

$$\begin{aligned} d_0 &\triangleq \arg\max_d F(d,t) > \frac{1}{2}, \\ d_1^\epsilon &\triangleq \arg\min_d F(d,t) < \frac{\epsilon}{nm}, \end{aligned} \tag{3.16}$$

where $F(d, t)$ is defined according to (3.6).

Definition 6 (ϵ-Coherent Sources) *Let* $G = (V, E)$ *be a binary network. Sources* $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ *are* ϵ*-coherent if,*

$$d(s_a, s_b) > 2(d_0 + d_1^\epsilon), \quad \forall 1 \leq a, b \leq m, a \neq b, \tag{3.17}$$

where d_0 *and* d_1 *are defined according to (3.16).*

Intuitively, sources are incoherent if they are sufficiently distant from each other in the network so that their infection effects at time t do not overlap in the network (for instance, viruses released from them, with high probability, have not visited the same nodes.). This assumption is a critical condition to solve the multi-source NI problem efficiently.

Definition 7 (Multi-Source Network Diffusion Kernel) *Suppose* $G = (V, E)$ *is a possibly directed binary graph and there exist* m *source nodes* $\mathcal{S} = \{s_1, \ldots, s_m\}$ *in the network that are* ϵ*-coherent. We say a node* $j \notin \mathcal{S}$ *gets infected at time* t *if it gets a virus from at least one of the sources. Thus, we have,*

$$Pr[y_j(t) = 1] \triangleq 1 - \prod_{s \in \mathcal{S}} \bar{p}_{s,j}(t), \tag{3.18}$$

where $\bar{p}_{s,j}(t) = 1 - p_{s,j}(t)$.

Using multi-source network diffusion kernel of Definition 7, the log-likelihood function $\mathcal{L}(\mathcal{S}, t)$ and the Hamming error function $\mathcal{H}_\alpha(\mathcal{S}, t)$ are defined as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{S}, t) &\triangleq \sum_{j \in V^t} \log(1 - \prod_{s \in \mathcal{S}} \bar{p}_{s,j}(t)) + \sum_{j \notin V^t} \log(\prod_{s \in \mathcal{S}} \bar{p}_{s,j}(t)), \\ \mathcal{H}_\alpha(\mathcal{S}, t) &\triangleq (1-\alpha) \sum_{j \in V^t} \prod_{s \in \mathcal{S}} \bar{p}_{s,j}(t) + \alpha \sum_{j \notin V^t} \big(1 - \prod_{s \in \mathcal{S}} \bar{p}_{s,j}(t)\big). \end{aligned} \tag{3.19}$$

Similarly to Algorithms 1 and 2, NI aims to find a set of m sources which maximizes the log-likelihood score, or minimizes the weighted Hamming error. However, unlike the single source case, these optimizations are computationally costly because all $\binom{|V^t|}{m}$ possible source combinations should be evaluated. If the number of infected nodes is significant ($|V^t| = \mathcal{O}(n)$), even for small constant number of sources, one needs to compute the likelihood or error scores for approximately $\mathcal{O}(n^m)$ possible source subsets, which may be computationally overly challenging for large networks.

One way to solve this combinatorial optimization is to take an iterative approach, where, at each step, one source node is inferred. However, at each step, using single source NI methods may not lead to an appropriate approximation because single source NI methods aim to find the source node which explains the entire infection pattern in the network, while in the multi-source case, the entire infection pattern are caused by multiple sources. To avoid this problem, at each step, we use a localized version of NI methods developed in Algorithms 1 and 2, where sources explain the infection pattern only around their neighborhood in the network.

Definition 8 *The localized likelihood function of node i in its d_0 neighborhood is defined as,*

$$\mathcal{L}_{d_0}(i, t) \triangleq \sum_{\substack{j \in V^t \\ j \in D(i, d_0)}} \log\big(p_{i,j}(t)\big) + \sum_{\substack{j \notin V^t \\ j \in D(i, d_0)}} \log\big(1 - p_{i,j}(t)\big), \tag{3.20}$$

where only nodes in the d_0 neighborhood of node i is considered in likelihood computation.

A similar argument can be expressed for the localized Hamming prediction error. For large d_0 values, the localized likelihood score is similar to the global likelihood score of (3.9). Using localized likelihood function is important in the multi-source NI problem because source candidates cannot explain the infection pattern caused by other sources. In the following, we propose an efficient localized NI method to solve the multi-source inference problem by maximizing localized likelihood scores of source candidates using a greedy approach. A similar algorithm can be designed for the localized minimum error NI.

Algorithm 3 (Localized NI-ML Algorithm) *Suppose $\mathcal{S}_r$ is the set of inferred sources at iteration r. The localized NI-ML algorithm has the following steps:*

- *Step 0:* $\mathcal{S}_0 = \varnothing$.

- *Step r+1:*
  - *Likelihood computation: compute s_{r+1} using the following optimization,*

$$s_{r+1} = \arg \max_{i \in V^t - D(\mathcal{S}_k, d_1^\epsilon)} \mathcal{L}_{d_0}(i,t). \tag{3.21}$$

  - *Update the source set: add s_{r+1} to the list of inferred sources,*

$$\mathcal{S}_{r+1} = \mathcal{S}_r \cup s_{r+1},$$

- *Termination: stop if $r = m$.*

In the following, we show that, if sources are sufficiently incoherent (i.e., sufficiently distant from each other in the network), the solution of localized NI Algorithm 3 approximates the exact solution closely.

Theorem 3 *Let $G = (V, E)$ be a regular undirected tree with countably infinite nodes. Suppose sources are ϵ-coherent according to Definition 6, and the underlying diffusion process is according to the SI model of Definition 1. Suppose $\mathcal{S}_r$ is the set of sources inferred by localized NI Algorithm 3 till iteration r. If $\mathcal{S}_k \subset \mathcal{S}$, then with probability at least $1 - \epsilon$, there exists a source node that has not been inferred yet whose localized likelihood score is optimal in expectation:*

$$\exists s \in \mathcal{S} - \mathcal{S}_r, \quad , \mathbb{E}[\mathcal{L}_{d_0}(s,t)] \geq \mathbb{E}[\mathcal{L}_{d_0}(i,t)] \quad , \forall i \in V^t - D(\mathcal{S}_k, d_1^\epsilon). \tag{3.22}$$

Proof The proof is presented in Appendix 7.7. ■

Proposition 7 *A worst case computational complexity of localized NI Algorithm 3 is $\mathcal{O}\big(|V^t|(k|E| + kn\log(n) + mn)\big)$.*

Proof The proof is presented in Appendix 7.8. ■

3.4 NI for Heterogeneous Network Diffusion

In previous sections, we have assumed that, the infection spread in the network is homogeneous; i.e., virus traveling time variables $\mathcal{T}_{(i,j)}$ are i.i.d. for all edges in the network. This can be an appropriate model for the binary (unweighted) graphs. However, if edges have weights, the infection spread in the network may be heterogeneous; i.e., the infection spread is faster over strong connections compared to the one of weak edges.

Suppose $G = (V, E, W)$ represents a weighted graph, where $w(i,j) > 0$ if $(i,j) \in E$, and $w(i,j) = 0$ otherwise. One way to model a heterogeneous diffusion in the network is to assume that, edge holding time variables $\mathcal{T}_{(i,j)}$ are distributed independently according to an exponential distribution with parameter $\lambda_{i,j} = w(i,j)$. According to this model, the average holding time of edge (i,j) is $1/w_{i,j}$, indicating the fast spread of infection over strong connections in the network.

Recall that $\mathcal{T}_{\mathcal{P}^r_{i\to j}}$ represents the virus traveling time variable from node i to node j over the path $\mathcal{P}^r_{i\to j}$. To simplify notations and highlight the main idea, consider the path $\mathcal{P}^r_{0\to l} = \{0 \to 1 \to 2 \ldots \to l\}$. The virus traveling time from node 0 to node l over this path ($\mathcal{T}_{\mathcal{P}^r_{0\to l}}$) is a *hypoexponential*

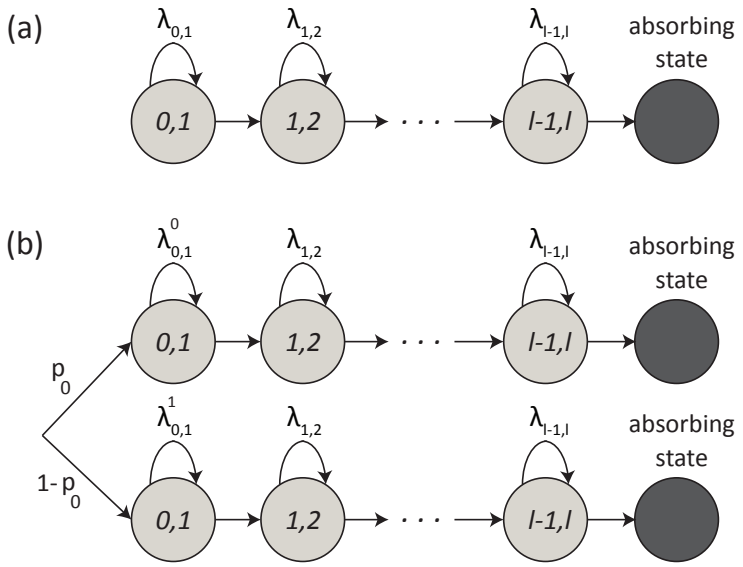

Figure 6: (a) A Markov chain of a hypo-exponential distribution. (b) A Markov chain of a mixed hypo-exponential distribution.

variable whose distribution is a special case of the *phase-type distribution*. For this path, we consider a Markov chain with $l+1$ states, where the first l states are transient, and the state $l+1$ is an absorbing state. Each transient state of this Markov chain corresponds to an edge (i,j) over this path whose holding time is characterized by an exponential distribution with rate $\lambda_{i,j} = w(i,j)$ (Figure 6-a). In this setup, the virus traveling time from node 0 to node l over the path $\mathcal{P}^r_{0\to l}$ is equal to the time from the start of the process until reaching to the absorbing state of the corresponding Markov chain. The distribution of this absorbing time can be characterized as a special case of the phase-type distribution. A subgenerator matrix of the Markov chain of Figure 6-a is defined as follows:

$$\begin{bmatrix} -\lambda_{0,1} & \lambda_{0,1} & 0 & \dots & 0 & 0 \\ 0 & -\lambda_{1,2} & \lambda_{1,2} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & -\lambda_{l-3,l-2} & \lambda_{l-3,l-2} & 0 \\ 0 & 0 & \dots & 0 & -\lambda_{l-2,l-1} & \lambda_{l-2,l-1} \\ 0 & 0 & \dots & 0 & 0 & -\lambda_{l-1,l} \end{bmatrix}. \tag{3.23}$$

For simplicity, denote the above matrix by $\Theta \equiv \Theta(\lambda_{0,1}, \dots, \lambda_{l-1,l})$. Define $\boldsymbol{\alpha} = (1, 0, \dots, 0)$ as the probability of starting in each of the l states. Then, the Markov chain absorbtion time is distributed according to $PH(\boldsymbol{\alpha}, \Theta)$, where $PH(.,.)$ represents a phase-type distribution. In this special case, this distribution is also called a hypoexponential distribution. A similar subgenerator matrix Θ can be defined for a general path $\mathcal{P}^r_{i\to j}$ connecting nodes i to j. Thus, we have,

$$F_{\mathcal{P}^r_{i\to j}}(t) = Pr[\mathcal{T}_{\mathcal{P}^r_{i\to j}} \le t] = 1 - \boldsymbol{\alpha} e^{t\Theta} \mathbf{1}, \tag{3.24}$$

where $\mathbf{1}$ is a column vector of ones of the size $|\mathcal{P}^r_{i\to j}|$, and e^X is the matrix exponential of X. For an unweighted graph where all edges have the same rate λ, (3.26) is simplified to (3.5). For the weighted graph $G = (V, E, W)$, we compute k shortest paths among pairs of nodes over the graph $G' = (V, E, W')$, where $w'(i,j) = 1/w(i,j)$ if $(i,j) \in E$, otherwise $w'(i,j) = \infty$. Then, the path network diffusion kernel for a weighted graph $G = (V, E, W)$ can be defined according to Definition (4). Using this kernel, NI algorithms introduced in Sections 3.2 and 3.3 can then be used to infer the source node under the heterogeneous diffusion in the network.

Note that, this framework can be extended to a more complex diffusion setup as well. We provide an example of such diffusion setup in the following:

Example 1 *Consider the the path $\mathcal{P}^r_{0\to l} = \{0 \to 1 \to 2 \to \ldots \to l\}$. Suppose the edge $(0,1)$ spreads the infection with rates $\lambda^0_{0,1}$ and $\lambda^1_{0,1}$ with probabilities p_0 and $1-p_0$, respectively. Suppose other edges (i,j) of this path spread the infection with rate $\lambda_{i,j}$. Figure 6-b illustrates the corresponding Markov chain for this path. The subgenerator matrices of this Markov chain can be characterized as follows:*

$$\Theta_i = \begin{bmatrix} -\lambda^i_{0,1} & \lambda^i_{0,1} & 0 & \ldots & 0 & 0 \\ 0 & -\lambda_{1,2} & \lambda_{1,2} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & -\lambda_{l-3,l-2} & \lambda_{l-3,l-2} & 0 \\ 0 & 0 & \ldots & 0 & -\lambda_{l-2,l-1} & \lambda_{l-2,l-1} \\ 0 & 0 & \ldots & 0 & 0 & -\lambda_{l-1,l} \end{bmatrix}, \tag{3.25}$$

for $i = 0, 1$. Then, for this path, we have,

$$F_{\mathcal{P}^r_{0\to l}}(t) = Pr[\mathcal{T}_{\mathcal{P}^r_{0\to l}} \le t] = 1 - \boldsymbol{\alpha}\big(p_0 e^{t\Theta_0} + (1-p_0)e^{t\Theta_1}\big)\mathbf{1}. \tag{3.26}$$

To compute the path-based network diffusion kernel, we compute shortest paths among pairs of nodes over the graph $G' = (V, E, W')$, where $w'(0,1) = p_0/\lambda^0_{0,1} + (1-p_0)/\lambda^1_{0,1}$, $w'(i,j) = 1/w(i,j)$ if $(i,j) \in E$ and $(i,j) \neq (0,1)$, and $w'(i,j) = \infty$ otherwise. This example illustrates that NI framework can be used even under a complex heterogeneous diffusion setup.

3.5 NI with Multiple Snap-shots

In this section, we consider the NI problem when multiple snapshots from infection patterns are available. To simplify notation and highlight the main ideas, we consider the single source case with two samples $\mathbf{y}(t_1)$ and $\mathbf{y}(t_2)$ at times t_1 and t_2, respectively. All arguments can be extended to a more general setup as well.

Recall that V^t denotes the set of infected nodes at time t. Let $E^t = \{(i,j)|(i,j) \in E, \{i,j\} \subset V^t\}$ represent edges among infected nodes in the network. The infection subgraph $G^t = (V^t, E^t)$ is connected if there is no infection recovery and the underlying diffusion is according to a dynamic process. We define an infection contraction operator $g(.)$ as follows:

$$g(v) = \begin{cases} v & v \in V \setminus V^t \\ x & o.w. \end{cases} \tag{3.27}$$

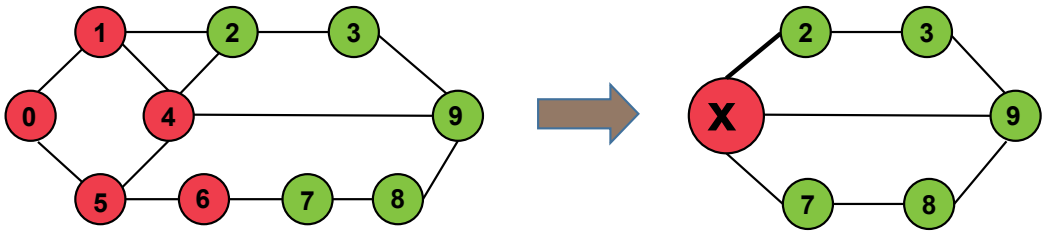

Figure 7: An example illustrating the infusion contraction graph of Definition 9.

where $x \notin V$. In other words, $g(.)$ maps all infected nodes to a new node x, while it maps all other nodes to themselves (Figure 7).

Definition 9 (Infusion Contraction Graph) *Suppose $G = (V, E, W)$ is a weighted graph whose infected subgraph at time t is represented as $G^t = (V^t, E^t, W^t)$. An infusion contraction graph $G_c^t = (V_c^t, E_c^t, W_c^t)$ is defined as follows:*

- *$(i, j) \in E_c^t$ for $i, j \neq x$ iff $(i, j) \in E$. In this case, $w_c^t(i, j) = w(i, j)$.*
- *$(i, x) \in E_c^t$ for $i \neq x$ iff there exists $j \in V^t$ such that $(i, j) \in E$. In this case, $w_c^t(i, x) = \sum_{\substack{j \in V^t \\ (i,j) \in E}} w(i, j)$.*

Figure 7 illustrates the infusion contraction graph for an example graph. Intuitively, the infusion contraction graph considers the infected subgraph as one node and adjusts weights of un-infected nodes connected to the infected ones accordingly.

Now we consider the source inference problem when two snapshots at times t_1 and t_2 are given. Recall that V^{t_1} and V^{t_2} denote the set of infected nodes at times t_1 and t_2, respectively. Without loss of generality, we assume $0 < t_1 < t_2$. Using the probability chain-rule, we can re-write the likelihood scores of Optimization (2.4) as,

$$Pr\big(\mathbf{y}(t_1), \mathbf{y}(t_2) | \mathcal{S} = \{s\}\big) = \underbrace{Pr\big(\mathbf{y}(t_1) | \mathcal{S} = \{s\}\big)}_{\text{term I}} \underbrace{Pr\big(\mathbf{y}(t_2) | \mathbf{y}(t_1), \mathcal{S} = \{s\}\big)}_{\text{term II}}. \tag{3.28}$$

Term (I) is the likelihood score of the single source NI Optimization (3.9). We consider different possibilities for Term II as follows:

- If $y_j(t_1) = 1$ and $y_j(t_2) = 1$, $Pr(y_j(t_2) | \mathbf{y}(t_1), \mathcal{S} = \{s\}) = 1$, because if a node gets infected, it remains infected (there is no recovery). Thus, if $y_j(t_1) = 1$ and $y_j(t_2) = 0$, $Pr(y_j(t_2) | \mathbf{y}(t_1), \mathcal{S} = \{s\}) = 0$.
- Now we consider the case $y_j(t_1) = 0$. Let $G_c^{t_1}$ be the infusion contraction graph of Definition 9. Suppose that all infected nodes at time t_1 are mapped to the node x. The second term can be approximated as follows:

$$Pr(y_j(t_2) = 1 | \mathbf{y}(t_1), \mathcal{S} = \{s\}) \approx p_{x,j}(t_2 - t_1), \tag{3.29}$$

where $p_{x,j}(.)$ is the path-based network diffusion kernel over the graph G_c^t. In other words, all infected nodes at time t_1 can be viewed as a single source x in the infusion contraction graph. Note that, this approximation is tight when the underlying edge holding time distribution is an exponential distribution which has the memory-less property. Moreover, note that, to compute the diffusion kernel in this case, we use the infusion contraction graph because infected nodes are not incoherent, and therefore, the multi-source diffusion kernel of Definition 7 cannot be used.

Under approximation (3.29), the second term of (3.28) leads to a similar expression for all source candidates, and therefore, the optimization is simplified to a single snapshot one. In practice, one may compute average source likelihood scores using all snapshots to decrease the variance of the source likelihood scores.

3.6 Non-parametric NI

In some real-world applications, only the network structure G and infection patterns $\{\mathbf{y}(t_1), \ldots, \mathbf{y}(t_z)\}$ are known and therefore to use NI algorithms, we need to learn the parameters such as observation times $\{t_1, \ldots, t_z\}$, and the number of sources m in the network. In the following, we introduce efficient techniques to learn these parameters:

- *Observation time parameters:* In the maximum likelihood NI Algorithm 1, according to Remark 11, the true parameter t is the one that maximizes the expected source likelihood score according to Theorem 1. Thus, in the case of unknown parameter t, we solve the following optimization:

$$(s, t) = \arg\max_{i,t'} \mathcal{L}(i, t'), \tag{3.30}$$

where $\mathcal{L}(i, t')$ is the log-likelihood function of node i using parameter t'. One way to solve Optimization (3.30) approximately is to quantize the range of parameter t (i.e., $t \in (0, t_{max})$) to b bins and evaluate the objective function in each case. Because we assume that the λ parameter of the edge holding time distribution is equal to one, one appropriate choice for t_{max} is the diameter of the infected subgraph, defined as the longest shortest path among pairs of infected nodes. The number of quantization levels b determines the resolution of the inferred parameter t and therefore the tightness of the approximation. If t_{max} is large and the true t parameter is small, to have a tight approximation, the number of quantization levels b should be large which may be computationally costly. In this case, one approach to estimate parameter t is to use the first moment approximation of the Erlang network diffusion kernel over source neighbors. Suppose μ is the fraction of the infected neighbors of source s. Since infection probabilities of source neighbors approximately come from an exponential distribution, for a given parameter t, $\mu \approx 1 - e^{-t}$. Therefore,

$$t \approx -ln(1 - \mu). \tag{3.31}$$

In the minimum error NI Algorithm 2, according to Remark 13, the prediction error of all infected nodes should be computed using the same parameter t. In the setup of Theorem

2, any value of parameter t leads to an optimal solution in expectation. In general, we suggest the following approach to choose this parameter: First, for each node, we minimize the prediction error for different values of the parameter t as follows,

$$t_i^* = \arg\max_{t'} \mathcal{H}_\alpha(i, t'). \tag{3.32}$$

This Optimization can be solved approximately similarly to the case of maximum likelihood NI Optimization (3.30). For the small t values, we use (3.31) to obtain t_i^*. Then, to obtain a fixed t parameter for all nodes, we average t_i^* parameters of the nodes with the minimum prediction error. In the cases of multi-source and multi-snapshot NI, one can use similar approaches to estimate time stamp parameters.

- *The number of sources:* In NI algorithms presented in Section 3.3, we assume that, the number of sources in the network (i.e., the parameter m) is known. In the case of unknown parameter m, if sources are sufficiently incoherent according to Definition 6, one can estimate m as follows: because sources are incoherent, their caused infected nodes do not overlap with each other with high probability. Thus, the number of connected components of the infected sub-graph (or the number of infected clusters in the network) can provide a reliable estimate of the number of sources in this case.

- *Regularization parameter of NI-ME:* The minimum error NI Algorithm 2 has a regularization parameter α which balances between false positive and false negative error types. In the setup of Theorem 2, any value of $0 < \alpha < 1$ leads to an optimal expected weighted Hamming error solution of (3.15). However, in general, we choose this parameter according to Remarks 9 and 10 to have a balance between the number of false negative and false positive errors.

4 NI Over Synthetic Networks

In this section, we compare the performance of proposed NI algorithms with other source inference methods over four different synthetic network structures. In our simulations, we assume that, there exists a single source in the network, and the underlying diffusion is according to the SI model of Definition 1. In the SI model, edge holding time variables $\mathcal{T}_{(i,j)}$ are i.i.d. having an exponential distribution with parameter $\lambda = 1$. Note that, to generate simulated diffusion patterns, we do not use our path-based network diffusion kernel to have a fair performance comparison of our methods with the one of other source inference techniques. We use the following methods in our performance assessment:

- **Distance Centrality:** This method infers the source node with the minimum shortest path distance from all infected nodes. Suppose G is the underlying network. The distance centrality of node i corresponding to infected nodes at time t is defined as follows:

$$D_t(i, G) \triangleq \sum_{j \in V^t} d(i, j), \tag{4.1}$$

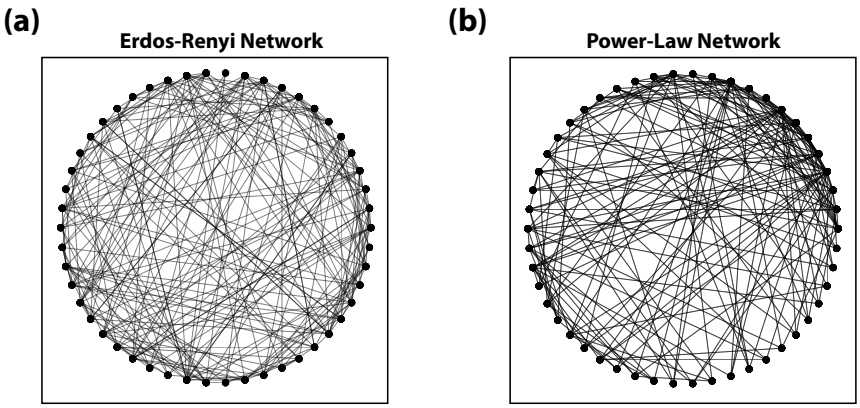

Figure 8: Examples of (a) Erdös-Rényi, and (b) power law networks, with 50 nodes.

where $d(i,j)$ represents the length of the shortest path between nodes i and j. A source node is inferred using the following optimization:

$$s = \arg\min_{i \in V^t} D_t(i, G). \tag{4.2}$$

If there is no path between two nodes i and j, $d(i,j) = \infty$. This makes the distance centrality measure sensitive to noise specially in real world applications. To avoid this issue, for disconnected nodes i and j, we assign $d(i,j) = M$, where M is a large number compared to shortest path distances in the network. In our simulations, we set M as 5 times larger than the network diagonal (i.e., the longest shortest path in the network).

- **Degree Centrality:** This methods infers the source node with highest direct connections to other infected nodes. Degree centrality of node i corresponding to infected nodes V^t is defined as follows:

$$C_t(i, G) \triangleq \sum_{j \in V^t} G(i,j). \tag{4.3}$$

Note that, unlike the distance centrality method which considers both direct and indirect interactions among infected nodes, degree centrality only considers direct interactions. To infer the source node using the degree centrality approach, one needs to solve the following optimization:

$$s = \arg\min_{i \in V^t} C_t(i, G). \tag{4.4}$$

- **Network Infusion:** We use NI methods based on maximum likelihood (denoted as NI-ML) described in Algorithm 1, and minimum error (denoted as NI-ME) described in Algorithm 2. To have a fair comparison with other methods, we assume that, the observation time parameter t is unknown and is estimated using the techniques presented in Section 3.6.

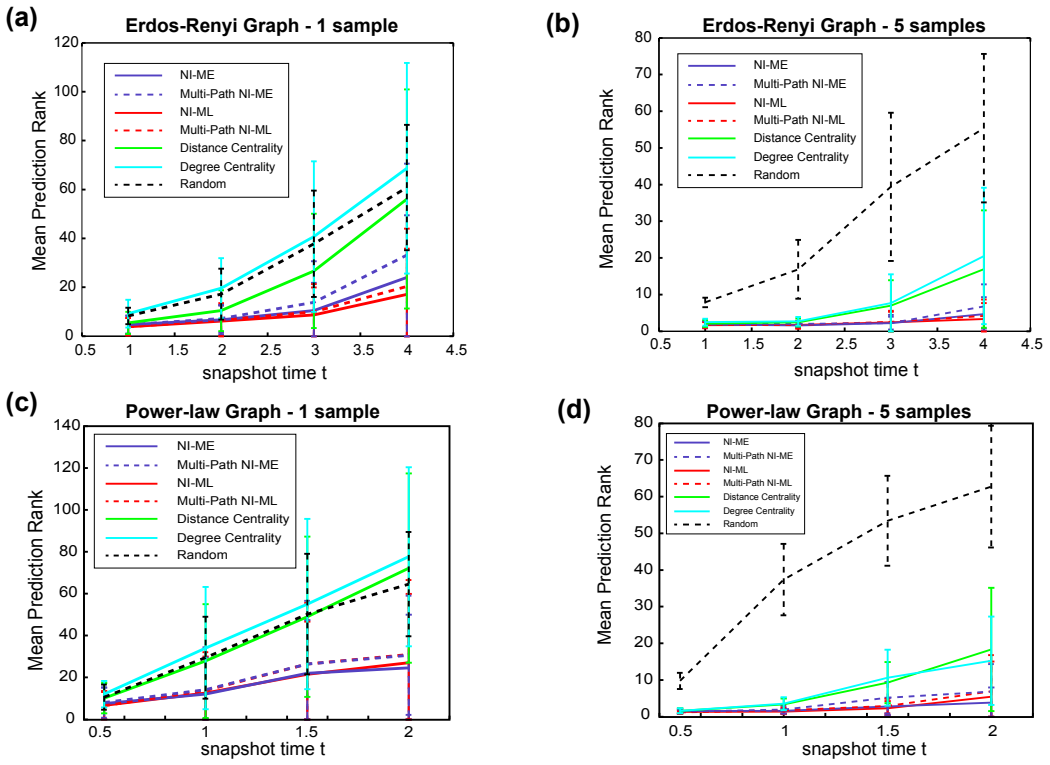

Figure 9: Performance of source inference methods over (a,b) Erdös-Rényi, and (c,d) power law networks graphs with 250 nodes and an average density of 1%. Experiments have been repeated 100 times.

In our simulations, we use four types of input networks:

- **Erdös-Rényi graphs:** In this case, G is a symmetric random graph where, $Pr[G(i,j) = 1] = p$. Networks have 250 nodes. An example network with fewer number of nodes is shown in Figure 8-a). In our simulations, we use $p = 0.01$.

- **Power law graphs:** We construct G as follows [25]; we start with a random subgraph with 5 nodes. At each iteration, a node is added to the network connecting to θ existent nodes with probabilities proportional to their degrees. This process is repeated till the number of nodes in the network is equal to $n = 250$. In our simulations, we use $\theta = 2$ which results in networks with the average density approximately 0.01 (see an example with fewer number of nodes in Figure 8-b).

- **Grid networks:** In this case, G is an undirected square grid network with 250 nodes. Figure 10-a shows an example of such networks with fewer number of nodes. We assume that, the source node is located at the center of the grid to avoid boundary effects.

- **Asymmetric grid networks:** In this case, G is an undirected graph with 250 nodes. It has 6 branches connected to the central node, three branches on the right with heavier connectivity among their nodes, and three branches on the left with sparse connectivity. Figure 11-a shows

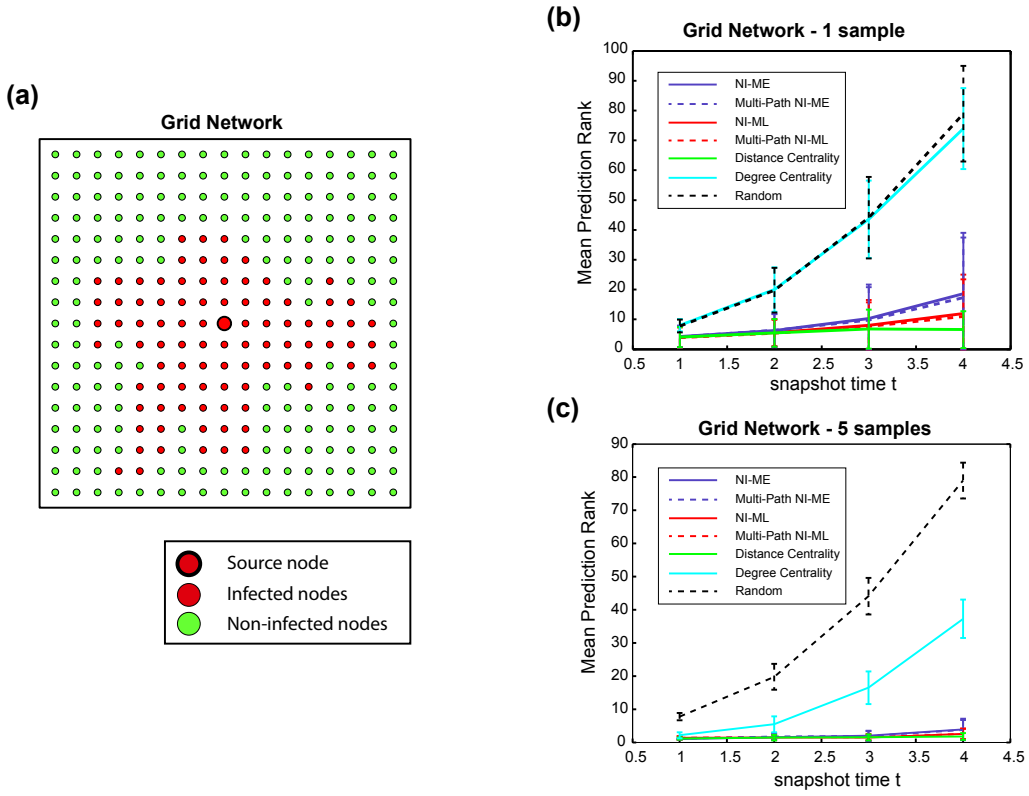

Figure 10: (a) An example of a grid network. Performance of source inference methods over grid networks with 250 nodes with (a) one and (b) five independent samples. Experiments have been repeated 100 times.

an example of such networks with fewer number of nodes. This is an adversarial example to highlight cases where NI methods fail to converge to the optimal value.

In multi-path NI methods, we consider top 10 independent shortest paths to form the k-path network diffusion kernel (i.e., $k = 10$). However, in the grid network, to enhance the computation time, we consider $k = 2$ because different nodes have at most 2 independent shortest paths connected to the source node. Parameter t is the time at which we observe the infection spread, and it determines the fraction of infected nodes in the network. If t is very large compared to the graph diameter, almost all nodes in the network become infected. On the other hand, for very small values of t, the source inference problem becomes trivial. In our simulations, we consider the cases where the number of infected nodes in the network is less than 75% of the total number of nodes, and greater than at least 10 nodes.

For evaluation, we sort infected nodes as source candidates according to the score obtained by different methods. High performing methods should assign the highest scores to the source node. The source node should appear on the top of the inferred source candidates. Ideally, if a method assigns the highest score to the source node, the rank of the prediction is one. We use the rank of true sources averaged over different runs of simulations. More formally, suppose $r(\mathcal{M}, s)$ is the rank of the source node s inferred by using the method $\mathcal{M}$. In an exact prediction, $r(\mathcal{M}, s) = 1$, while an average rank of a method based on random guessing is $r(\mathcal{M}, s) = |V^t|/2$. If $r(\mathcal{M}, s)$ is

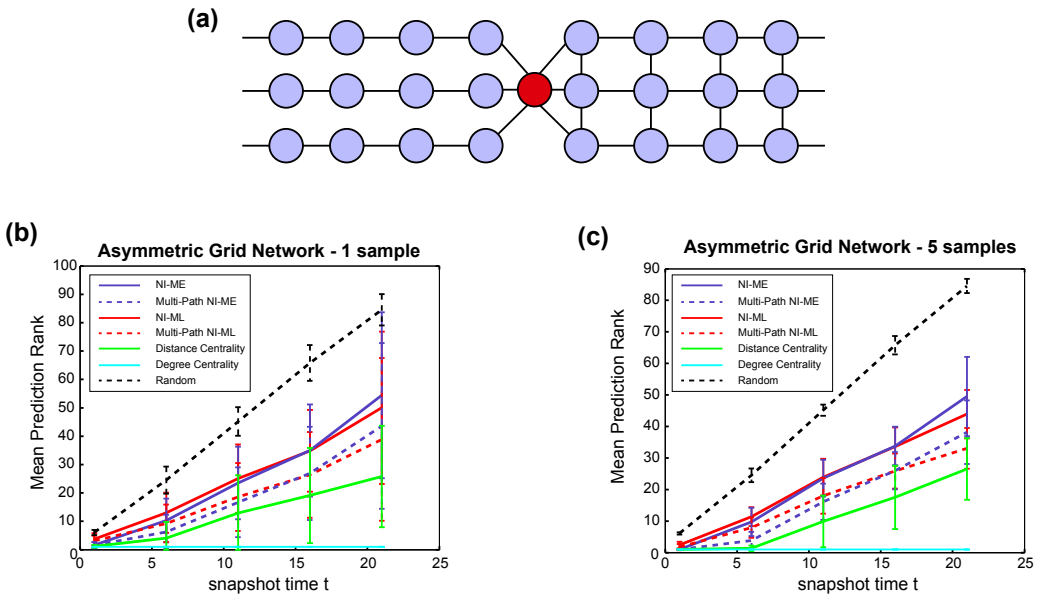

Figure 11: (a) An asymmetric grid network. Performance of source inference methods over asymmetric grid networks with 250 nodes with (a) one and (b) five independent samples. Node with red color is the source node. Experiments have been repeated 100 times.

close to one, it means that, the true source is among top predictions of the method $\mathcal{M}$. In each case, we run simulations 100 times.

Figure 9 compares the performance of different source inference methods over both Erdös-Rényi and power law networks, and in different ranges of the parameter t. In both network models and in all diffusion rates, NI Algorithms based on maximum likelihood (NI-ML) and minimum error (NI-ME) outperform other methods. Panels (a) and (c) illustrate the performance when only one sample from the infection pattern at time t is available, for Erdös-Rényi and power law networks, respectively. Panels (b) and (d) illustrate the performance of different methods when five independent samples from the infection pattern at time t are given, illustrating the mean-field optimality of NI methods according to Theorems 1 and 2. Because the underlying networks are sparse, according to Proposition 1, the performance of maximum likelihood and minimum error NI methods, both shortest-path and multi-path versions, are close to each other in both network models. Notably, unlike NI methods, the performance of other source inference methods such as distance centrality and degree centrality does not tend to converge to the optimal value even for higher sample sizes.

Figure 10 compares the performance of different source inference methods over grid networks in different ranges of the parameter t. In the case of having a single sample from the infection pattern, in small ranges of the parameter t, when the fraction of infected nodes is less than approximately $1/4n$, NI methods and distance centrality have approximately the same performance, significantly outperforming the degree centrality method. In higher diffusion rates, distance centrality outperforms NI-ML method and NI-ML method outperforms NI-ME method. Again in this range, the performance of the degree centrality method is significantly worst than other methods. Having five

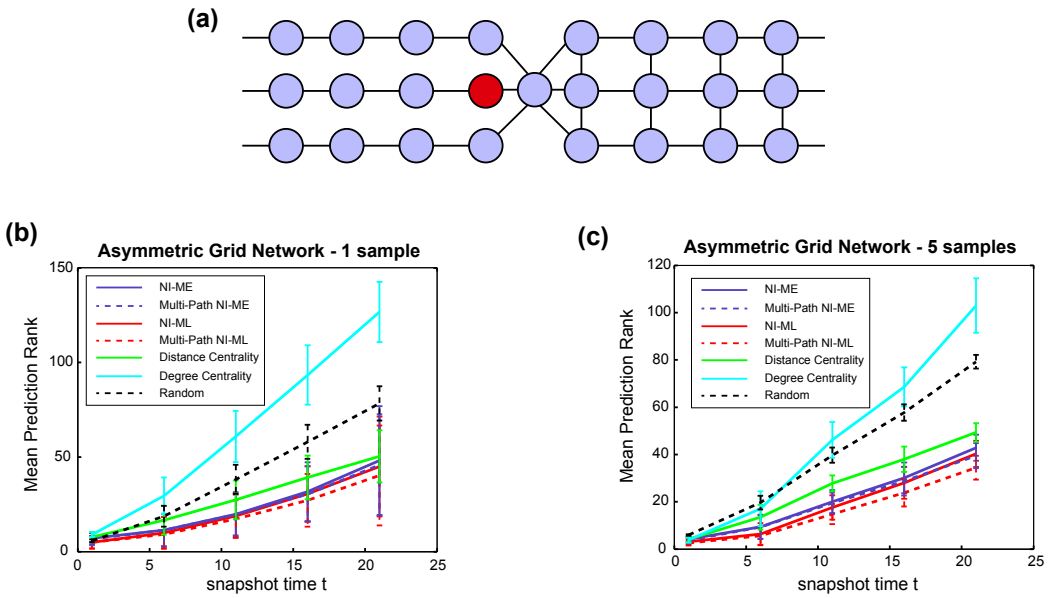

Figure 12: (a) An asymmetric grid network. Performance of source inference methods over asymmetric grid networks with 250 nodes with (a) one and (b) five independent samples. Node with red color is the source node. Experiments have been repeated 100 times.

independent samples from node infection patterns, the performance of NI methods and distance centrality converges to the optimal one.

Unlike tree and sparse Erdös-Rényi networks, there are multiple overlapping paths among nodes in the grid structures. However, as we illustrate in Figure 10, the performance of NI methods converge to the optimal value, similarly to the case of sparse graphs. The main reason is that, grid structures are symmetric and even though paths among nodes overlap significantly, considering shortest paths among nodes approximates the true underlying diffusion based on an SI dynamics closely. In order to have an adversarial example where the performance of NI methods do not converge to the optimal one, we design an asymmetric grid structure illustrated in Figure 11. The three branches on the right side of the central node have strong connectivity among themselves, forming a grid, while the ones on the left is sparsely connected to each other, forming a tree. Therefore, the spread of infection will be faster on the right side compared to the left side, under a dynamic SI model. However, considering only the shortest path among nodes does not capture this asymmetric structure. Therefore, the performance of shortest path NI methods diverges from the optimal value as diffusion rate increases (see Figure 11). In this case, considering more paths among nodes to form the k-path network diffusion kernel according to Definition 4 improves the performance of NI methods significantly, because higher order paths partially capture the asymmetric diffusion spread in the network. Note that, the degree centrality method has the best performance in this case, because the source node has the highest degree in the network by design. If we select another node to be the source node as illustrated in Figure 12-a, the performance of the degree centrality method becomes worst significantly, indicating its sensitivity to the source location. In the setup of Figure 12, multi-path NI methods outperform the one of other methods, although their performance do

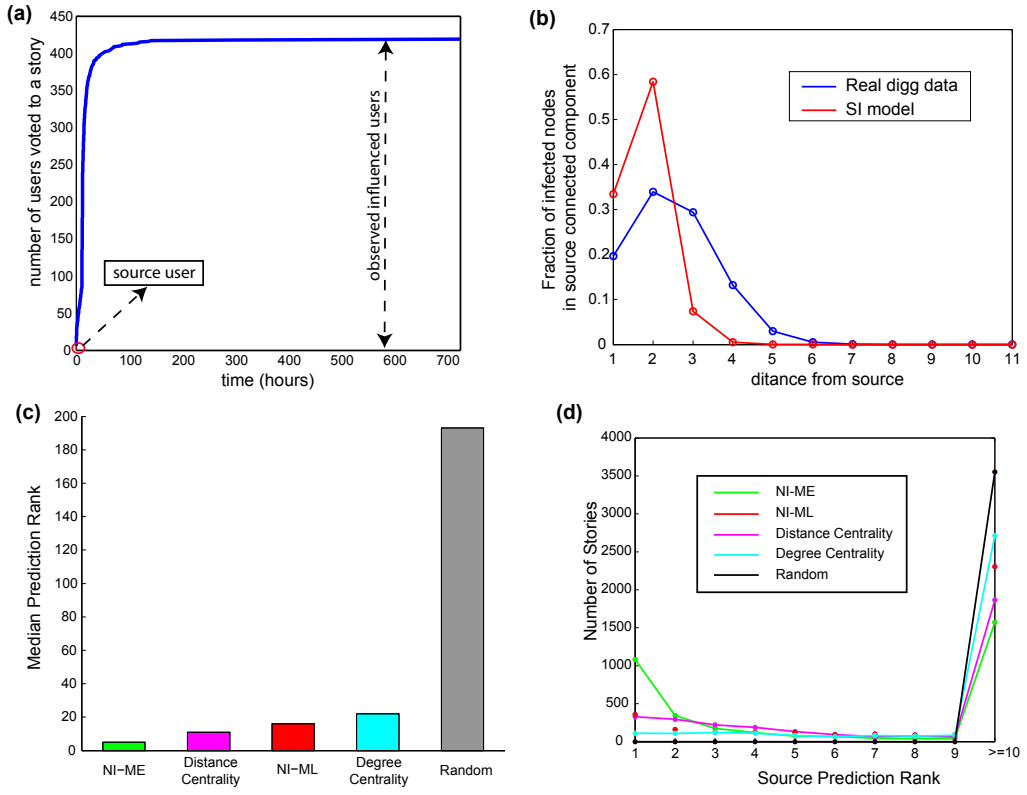

Figure 13: (a) An example of number of votes of a story over time. (b) Fraction of infected nodes in the connected component of sources in both Digg data and using simulated SI model. (c) Median rank of true sources of 3,553 different Digg stories inferred by different source inference methods. (d) Histogram of source ranks of 3,553 Digg stories inferred by different methods.

not converge to the optimal value.

5 NI Over Real Networks

In this section, we evaluate the performance of different source inference methods in two real data applications: the first application is to infer news sources over Digg social news aggregator [26]. In this application, the underlying diffusion process is based on the real dynamics over the Digg friendship network, and also true sources of different news stories are known. These characteristics make this application an ideal framework to assess the robustness and performance of different source inference methods. The second application is to infer infusion hubs of different human diseases, defined as gene candidates that explain the connectivity pattern of disease related genes over the gene regulatory network significantly. We show that, the inferred infusion hubs of different diseases are biologically relevant and often not identifiable using the raw p-values.

5.1 Inference of News Sources Over Social Networks

In this section, we evaluate the performance of NI and other source inference techniques in identifying news sources over the social news aggregator Digg (http://digg.com). Digg allows its users to submit and rate different news links. Highly rated news links are *promoted* to the front page of Digg. Digg also allows its users to become friends with other users and follow their activities over the Digg network. Digg's friendship network is asymmetric, i.e., user A can be a follower (friend) of user B but not vice versa. Reference [26] have collected voting activities and friendship connections of Digg's users over a period of a month in 2009 for $3,553$ promoted stories. We use this data to form a friendship network of active Digg users with $24,219$ nodes and more than $350K$ connections. We consider users as active if they have voted for at least 10 stories in this time period. Figure (1) demonstrates a small part of the Digg friendship network.

For each story, we have voting time stamps of different users where the first voting user is the source of that story. Figure (13-a) demonstrates the number of votes of a particular story in different times. If friends of a user A vote for a specific story, it is more likely that user A also votes for that story and that is how information propagates over the Digg friendship network. NI aims to inverse this information propagation process to infer the news source by observing the voting pattern in a single snapshot in the steady state. Here, we only consider the shortest path among nodes to compute the path-based network diffusion kernel used in NI Algorithms 1 and 2.

This application provides an ideal real data framework to assess the performance and robustness of different source inference methods because the true sources (i.e., the user who started the news) are known for different stories and also the underlying diffusion processes are based on real dynamics over the Digg friendship network. These real dynamics can in fact be significantly different from the standard SI diffusion model (Figure 13-b). Moreover, not all of the voting pattern is derived by the source users and there are disconnected voting activities over the Digg friendship network. Thus, performance assessment of different source inference methods in this application can provide a measure of robustness of different methods under real-world circumstances.

Figure (13-c) demonstrates the median rank of true sources of $3,553$ news stories inferred by different methods. The median source rank of NI-ME (Algorithm 2) is 4 which is 7, 13 and 18 ranks better than the ones of distance centrality, NI-ML, and degree centrality methods, respectively. In fact, the NI-ME algorithm infers news sources optimally (in its first prediction rank) for approximately 31.3% of stories, while this number for distance centrality, NI-ML, and degree centrally methods are 9.7%, 8.9% and 3.2%, respectively (Figure 13-d).

5.2 Infusion Hubs of Human Diseases

The genome sequence can play an important role in identifying an individual's susceptibility to a particular disease. One of the main goals of genome-wide association studies (GWAS) is to discover marker single nucleotide variants (SNV) that are associated with complex diseases [27]. Recently, researchers have hypothesized that some human diseases can be related to single nucleotide variants (SNVs) sitting in enhancer-like regions of the genome and are typically enriched in transcription factor binding sites [4]. This leads to the hypothesis that, disease-associated SNVs of particular diseases may be disrupting gene regulatory processes. Thus, using regulatory networks can help us to identify direct and indirect target genes as well as higher-order regulatory pathways of these disease-causing SNVs [5].

We use human gene regulatory network constructed by integration of genome-wide datasets

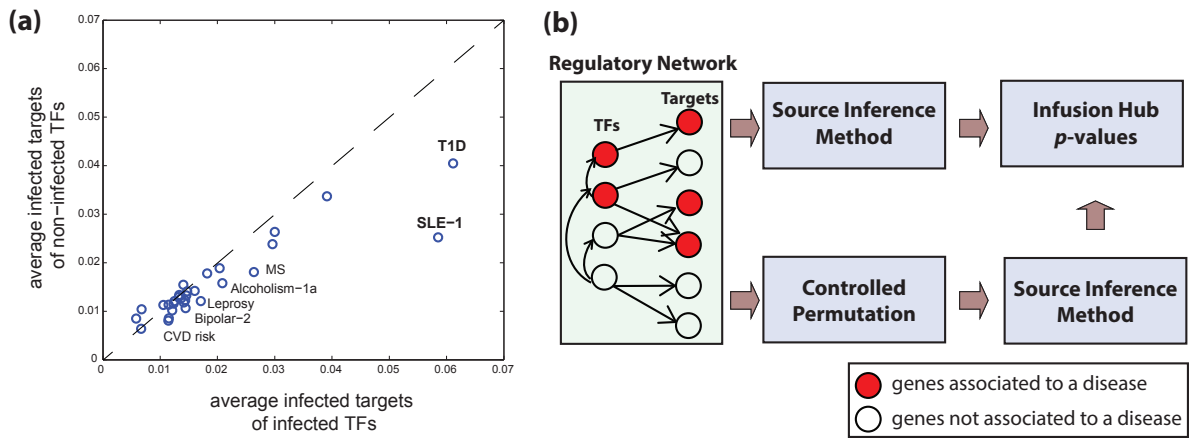

Figure 14: (a) Enrichment of regulatory interactions for different human diseases. (b) The proposed framework to compute significant infusion hubs of human diseases in the regulatory network.

of transcription factor binding, gene expression, and regulatory motif information of ENCODE consortium [28]. The human gene regulatory network is a directed binary network which connects 2,757 transcription factors (TFs) to 19,221 target genes using more than $2.5M$ regulatory connections. For different diseases, we compute gene p-values by meta-analysis of gene set enrichments using p-values of single nucleotide variants (SNVs) [29]. We say a gene is related to a disease if it has a p-value less than 0.01 for that disease. Over the regulatory network, for some diseases including Type 1 Diabetes (T1D), Systemic lupus erythematosus (SLE), Multiple sclerosis (MS), etc., we find that, if a TF is disease-related, its targets are more likely to be disease-related as well (Figure 14-a), indicating the enrichment of regulatory connections for these diseases.

Here, we use source inference methods to identify *infusion hubs* of different human diseases (Figure 14-b). Infusion hubs are defined as gene candidates that explain the connectivity pattern of disease related genes in human gene regulatory network significantly. Figure 14-b illustrates our framework to infer significant infusion hubs of different diseases using source inference methods. First, we assign infusion hub scores to different disease-related genes by performing source inference methods such as NI, degree centrality and distance centrality. We use single-source NI algorithms because most disease-related genes form a giant connected component in the network. Also, similarly to the social network application, we consider the shortest path among nodes in the NI algorithms (i.e., $k = 1$). To compute the statistical significance of these scores, we randomize disease patterns using controlled permutations. In the permutation step, we keep the network structure and the number of disease-related TFs and target genes the same as the real ones. Moreover, we keep in- and out- degree distributions of disease-related TFs and targets approximately the same as the real ones, to eliminate potential degree biases of infusion hubs. Then, we compute p-values of genes to be an infusion hub of a particular disease by comparing the actual and permutation scores. Genes with p-values less than 0.01 are called significant infusion hubs for that particular disease.

Figure 15-a illustrates the number of significant infusion hubs of different diseases using various sources inference methods. The minimum error NI method (NI-ME) infers the most number of

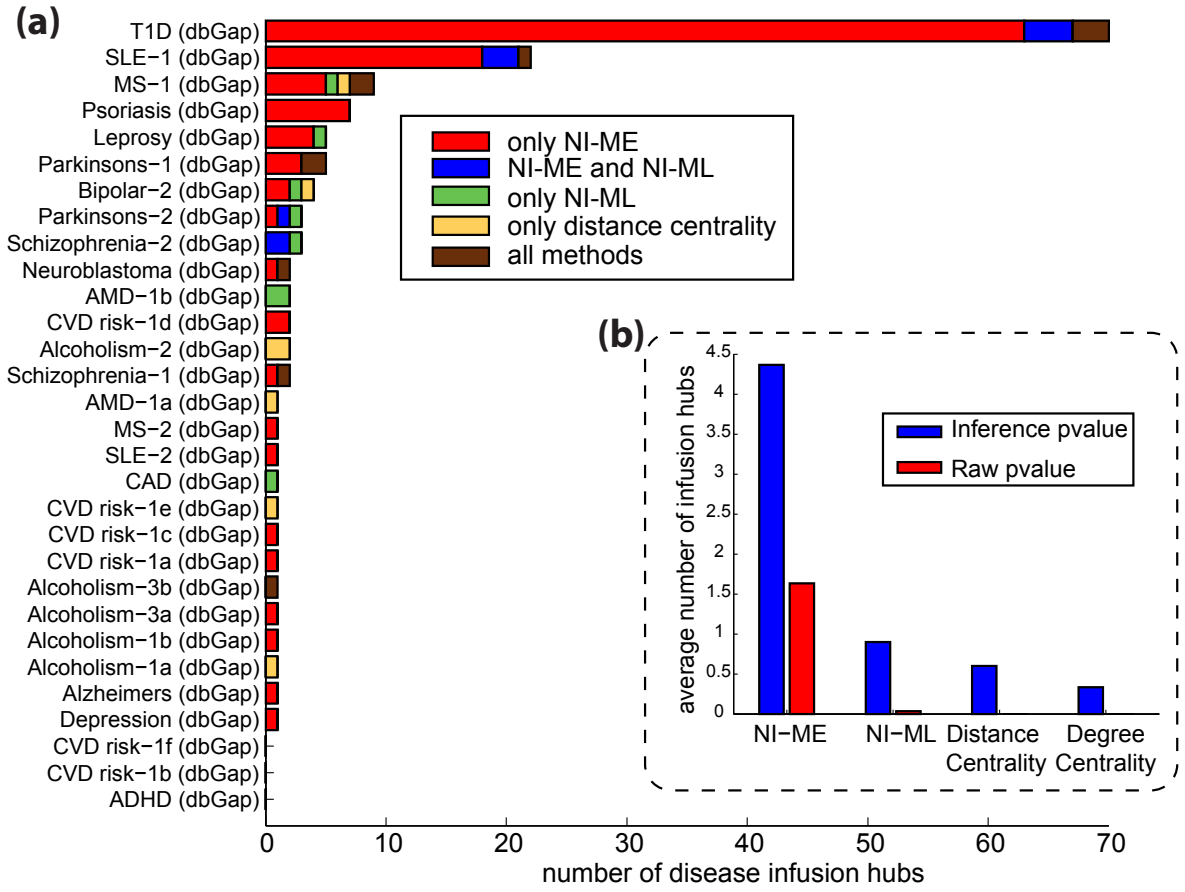

Figure 15: (a) Number of significant infusion hubs inferred by different methods (a) for individual diseases, and (b) on average.

significant disease infusion hubs on average compared to other methods (Figure 15-b). Moreover, most of the disease hubs cannot be inferred by merely using the raw gene p-values. Furthermore, diseases with the strongest regulatory enrichment (Figure 14-a) such as T1D, SLE and MS have the highest number of significant infusion hubs compared to other diseases. For some diseases, we use multiple available datasets that is clarified in their names.

Table 1 presents common infusion hubs across different diseases inferred by the NI-ME method (the overlap of infusion hubs inferred by other methods is not significant). Examples of these common infusion hubs are gene BCL3 across diseases Parkinson's and Alzheimers, gene EGR1 across diseases Schizophrenia and MS, gene TCF12 across diseases Parkinson's and Neuroblastoma, etc. Most of these inferred individual or common hubs are in fact biologically relevant [30, 31], and mostly unidentifiable merely by using the raw gene p-values. Thus, the proposed framework to infer infusion hubs of human diseases can potentially open new directions in drug designs and therapies for these diseases. It is important to note that, further experiments should be performed to determine the involvement of these gene candidates in human diseases which is beyond the scope of this paper.

Infusion Hub	Diseases	Infusion Hub	Diseases
BCL3	Parkinson's-Alzheimers	EGR1	Schizophrenia-MS
TCF12	Parkinson's-Neuroblastoma	PBX2-VARS2	T1D-MS
HLA-DQB1	MS-T1D-Leprosy	ATF6B-PHF1	SLE-T1D
PATZ1	CVD-MS	TRIM10-TRIM15-ZNRD1	SLE-Psoriasis

Table 1: Common infusion hubs of diseases inferred by the NI-ME algorithm.

6 Conclusion

In this paper, we proposed a computationally tractable general method for source inference which we termed Network Infusion (NI). Our source inferences are based on a continuous-time path-based network diffusion, which considers k edge-disjoint shortest paths among pairs of nodes, neglecting other paths in the network. We used this kernel to solve efficiently the inverse diffusion problem by maximizing the likelihood or minimizing the prediction error. The minimum error NI algorithm is based on an asymmetric Hamming premetric function, and balances between false positive and false negative error types. We applied NI framework for both single-source and multi-source diffusion, for both single-snapshot and multi-snapshot observations, and using both uninformative and informative prior probabilities for candidate source nodes. We provided proofs that under a standard susceptible-infected diffusion model, the maximum-likelihood NI is mean-field optimal for tree structures or sufficiently sparse Erdös-Rényi graphs, the minimum-error algorithm is mean-field optimal for regular tree structures, and for sufficiently-distant sources, the multi-source solution is mean-field optimal in the regular tree structure. We showed that, NI can be used in a complex heterogeneous diffusion setup as well, where edges propagate information/infection in the network according to different diffusion processes. In this setup, our network diffusion kernel is characterized using the phase-type distribution of a Markov chain absorbing time. Moreover, we proposed extensions to NI algorithms for cases with unknown or partially known model parameters such as observation times, the number of sources, etc., by introducing techniques to learn these parameters from observed infection samples.

We applied NI to several synthetic networks and compared its performance to centrality-based and distance-based methods, for Erdös-Rényi graphs, power-law networks, symmetric and asymmetric grids. Our results illustrated the superiority of proposed NI algorithms compared to existing methods, specially in sparse networks. Moreover, we used NI in two applications. First, we identified the news sources for 3,553 stories in the Digg social news network, and validated our results based on annotated information, that was not provided to our algorithm. In this real-world application with unknown underlying dynamics, we found that, the minimum error NI algorithm outperformed other methods significantly and led to a robust source inference solution, owing to the balance between false positive and false negative error types. Second, we applied NI to identify infusion hubs of human diseases, defined as gene candidates that explain the connectivity pattern of disease-related genes in the human regulatory network significantly. In this application, again the NI-ME algorithm outperformed other methods and identified infusion hubs of several human diseases including T1D, Parkinson, MS, SLE, Psoriasis and Schizophrenia. We showed that, the inferred infusion hubs are biologically relevant and often not identifiable using the raw p-values.

7 Appendix

In this section, we present proofs of the main results of the paper.

7.1 Proof of Proposition 1

First, we compute the probability that a node $v \in V$ belongs to a cycle of length at most l. Such a cycle is determined by the $l-1$ other vertices. By choosing them in order, there are less than n^{l-1} choices for those other vertices, while the cycle appears with probability p^l in the graph. Thus, the probability that v is involved in a cycle of length l is at most $n^{l-1}p^l \leq c^l/n$. To have an overlapping shortest path between nodes i and j, at least one of the nodes over that path should belong to a cycle of length at most $2l_0$. This happens with probability less than $l_0 c^{2l_0}/n$, which goes to zero asymptotically if $c < 1/n^{2l_0}$.

7.2 Proof of Proposition 2

To compute the k-path network diffusion kernel, we need to compute k-independent shortest paths among nodes. Note that ties among paths with the same length is broken randomly as explained in Section 3.1. Computation of these paths among one node and all other nodes using the Dijkstra's algorithm costs $\mathcal{O}(k|E| + kn\log(n))$. Thus, computational complexity of forming the entire kernel has complexity $\mathcal{O}(k|E|n + kn^2\log(n))$.

7.3 Proof of Proposition 3

To prove these properties, we use an alternative definition of the Erlang cumulative distribution function as follows:

$$F(d_{i,j}, t) = 1 - \sum_{k=0}^{d_{i,j}-1} \frac{1}{k!} e^{-t} t^k. \tag{7.1}$$

For larger values of $d_{i,j}$, there are more positive terms in the summation of equation (7.1). Thus, this is a decreasing function of $d_{i,j}$.

To prove the second part, we have,

$$\begin{aligned}
\frac{\partial p_{i,j}(t)}{\partial t} &= \sum_{k=0}^{d_{i,j}-1} \frac{1}{k!}(-e^{-t}t^k + kt^{k-1}e^{-t}), \\
&= \sum_{k=0}^{d_{i,j}-1} \frac{1}{k!} e^{-t}t^k + \sum_{k=0}^{d_{i,j}-2} \frac{1}{k!} e^{-t}t^k, \\
&= F(d_{i,j}-1, t) - F(d_{i,j}-1, t).
\end{aligned}$$

This completes the proof.

7.4 Proof of Proposition 4

Computation of the k-path network diffusion kernel for one node has complexity $\mathcal{O}(k|E|+kn\log(n))$, according to Proposition 2. We need to compute the kernel for V^t nodes. Moreover, Optimizations 3.9 and 3.11 have complexity $\mathcal{O}(|V^t|n)$. Thus, the total computational complexity is $\mathcal{O}\big(|V^t|(k|E| + kn\log(n))\big)$.

7.5 Proofs of Theorem 1 and Proposition 5

First, we prove the following lemma:

Lemma 1 *Let x, y and z be positive numbers such that $0 < x, y < z$. Define*

$$f(x,y) \triangleq x \log \frac{x}{y} + (z-x) \log \frac{z-x}{z-y}. \tag{7.2}$$

Then, $f(x,y) \geq 0$ where equality holds iff $x = y$.

Proof We have,

$$\frac{\partial f}{\partial y} = -\frac{x}{y} + \frac{z-x}{z-y},$$
$$\frac{\partial^2 f}{\partial y^2} = \frac{x}{y^2} + \frac{z-x}{(z-y)^2} > 0.$$

Thus, $f(.,.)$ is a convex function where its minimum is equal to 0 and occurs at $x = y$. This completes the proof of Lemma 1. ■

Recall that $\mathcal{L}(i,t')$ is the likelihood score of node i using diffusion parameter t':

$$\mathcal{L}(i,t') = \sum_{j \in V^t} \log\big(p_{i,j}(t')\big) + \sum_{j \notin V^t} \log\big(1 - p_{i,j}(t')\big). \tag{7.3}$$

First, we prove Proposition 5. Let s be the source node and t be the infection observation time (t' can be different than t, see explanations of Remark 11). Thus, we can write,

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(s,t)] - \mathbb{E}[\mathcal{L}(s,t')] &= \sum_{j \in V} p_{s,j}(t) \log\big(p_{s,j}(t)\big) + \big(1 - p_{s,j}(t)\big) \log\big(1 - p_{s,j}(t)\big) \\
&\quad - \sum_{j \in V} p_{s,j}(t) \log\big(p_{s,j}(t')\big) + \big(1 - p_{s,j}(t)\big) \log\big(1 - p_{s,j}(t')\big) \\
&= \sum_{j \in V} p_{s,j}(t) \log \frac{p_{s,j}(t)}{p_{s,j}(t')} + \big(1 - p_{s,j}(t)\big) \log \frac{1 - p_{s,j}(t)}{1 - p_{s,j}(t')} \\
&\overset{(I)}{\geq} \sum_{j \in V} p_{s,j}(t) \log \frac{\sum_{j \in V} p_{s,j}(t)}{\sum_{j \in V} p_{s,j}(t')} + \sum_{j \in V} 1 - p_{s,j}(t) \log \frac{\sum_{j \in V} 1 - p_{s,j}(t)}{\sum_{j \in V} 1 - p_{s,j}(t')} \\
&\overset{(II)}{\geq} 0.
\end{aligned} \tag{7.4}$$

Inequality (I) follows from the log-sum inequality. Inequality (II) follows from Lemma 1. In particular, according to Lemma 1, the equality condition (II) holds iff $\sum_{j \in V} p_{s,j}(t) = \sum_{j \in V} p_{s,j}(t')$ which indicates $t = t'$. This completes the proof of Proposition 5.

In the next step, we prove Theorem 1.

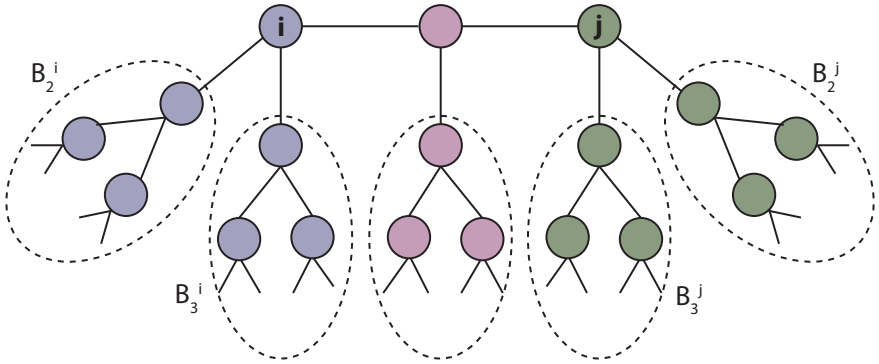

Figure 16: An example of a regular tree used in the proof of Theorem 2.

$$
\begin{aligned}
\mathbb{E}[\mathcal{L}(s,t)] - \mathbb{E}[\mathcal{L}(i,t')] &= \sum_{j\in V} p_{s,j}(t)\log\big(p_{s,j}(t)\big) + \big(1-p_{s,j}(t)\big)\log\big(1-p_{s,j}(t)\big) \\
&\quad - \sum_{j\in V} p_{s,j}(t)\log\big(p_{i,j}(t')\big) + \big(1-p_{s,j}(t)\big)\log\big(1-p_{i,j}(t')\big) \\
&= \sum_{j\in V} p_{s,j}(t)\log\frac{p_{s,j}(t)}{p_{i,j}(t')} + \big(1-p_{s,j}(t)\big)\log\frac{1-p_{s,j}(t)}{1-p_{i,j}(t')} \\
&\overset{(III)}{\geq} \sum_{j\in V} p_{s,j}(t)\log\frac{\sum_{j\in V} p_{s,j}(t)}{\sum_{j\in V} p_{i,j}(t')} + \sum_{j\in V} 1-p_{s,j}(t)\log\frac{\sum_{j\in V} 1-p_{s,j}(t)}{\sum_{j\in V} 1-p_{i,j}(t')} \\
&\overset{(IV)}{\geq} 0.
\end{aligned}
\tag{7.5}
$$

Inequality (III) follows from the log-sum inequality. Inequality (IV) follows from Lemma 1. This completes the proof of Theorem 1.

7.6 Proof of Theorem 2

To prove Theorem 2, first we prove regular trees are distance symmetric according to the following definition:

Definition 10 *A graph $G = (V, E)$ is distance symmetric if for any pair of nodes $i, j \in V$, there exists a graph partition $\{V_1, V_2, V_3\}$ where,*

- *$\forall r \in V_1$, $d(i,r) = d(j,r)$. I.e., distances of nodes in V_1 from both nodes i and j are the same.*
- *There exists a bijective mapping function $\zeta(.)$ between nodes V_2 and V_3 (i.e., $g : V_2 \to V_3$) such that for any $r \in V_2$, $d(i,r) = d(j,\zeta(r))$.*

In the following Lemma, we show that regular trees are in fact distance symmetric:

Lemma 2 *Let $G = (V, E)$ be a regular tree where V is countably infinite set of vertices and E is the set of edges. Then, G is distance-symmetric according to Definition 10.*

Proof Consider two distinctive nodes i and j. In the following, we construct graph partitions V_1, V_2 and V_3 satisfying conditions of Definition 10. Let degree of nodes in the regular tree be k. Thus, there are k branches connected to each nodes i and j, denoted by $\{B_1^i, \ldots, B_k^i\}$ and $\{B_1^j, \ldots, B_k^j\}$, respectively. Without loss of generality, assume $i \in B_1^j$ and $j \in B_1^i$. Add $\{B_2^i, \ldots, B_k^i\}$ and $\{B_1^j, \ldots, B_k^j\}$ to the partition sets V_2 and V_3, respectively. A mapping function $\zeta(.)$ of Definition 10 can be constructed between these sets by mapping nodes over branches B_l^i $(l \neq 1)$ to nodes over branches B_l^j $(l \neq 1)$ in a symmetric way (see Figure 16).

Now consider the branch connecting nodes i and j in the graph. Let $\{1, 2, \ldots, l\}$ be nodes over the shortest path connecting node i to node j. Therefore, $d(i, j) = l + 1$, the distance between nodes i and j.

- If l is odd, add non-partitioned nodes connected to nodes $\{1, \ldots, l/2\}$ to the partition set V_2. Similarly, add remaining nodes connected to nodes $\{l/2 + 1, \ldots, l\}$ to the partition set V_3.
- If l is odd, add non-partitioned nodes connected to nodes $\{1, \ldots, \lfloor l/2 \rfloor\}$ and $\{\lceil l/2 \rceil + 1, \ldots, l\}$ to partition sets V_2 and V_3, respectively. Non-partitioned nodes connected to the node $\lceil l/2 \rceil$ are assigned to the partition set V_1.

A mapping function $\zeta(.)$ of Definition 10 can be constructed between newly added nodes to partition sets V_2 and V_3 in a symmetric way. Moreover, nodes in the partition set V_1 have the same distance from both nodes i and j. This completes the proof. ■

Without loss of generality, suppose node 0 is the source node and we observe the infection pattern at time t according to the SI diffusion model. Thus, $Pr[y_j(t) = 1] = p_{0,j}(t)$, defined according to equation (3.5). Suppose we use parameter t' in Network Infusion Algorithm 2. According to equation (3.11), we have,

$$\begin{aligned} \mathbb{E}[\mathcal{H}_\alpha(i, t')] &= \sum_{j \in V} (1 - \alpha) p_{0,j}(t) \big(1 - p_{i,j}(t')\big) + \alpha \big(1 - p_{0,j}(t)\big) p_{i,j}(t') \\ &= \sum_{j \in V} p_{0,j}(t) \big(1 - p_{i,j}(t')\big) + \alpha \big(p_{i,j}(t') - p_{0,j}(t)\big). \end{aligned} \tag{7.6}$$

Thus, we have,

$$\begin{aligned}
\mathbb{E}[\mathcal{H}_\alpha(i,t')] - \mathbb{E}[\mathcal{H}_\alpha(0,t')] &= \sum_{j\in V} p_{0,j}(t)\big(p_{0,j}(t') - p_{i,j}(t')\big) + \alpha\big(p_{i,j}(t') - p_{0,j}(t')\big) \qquad (7.7)\\
&= \sum_{j\in V} \big(p_{0,j}(t') - p_{i,j}(t')\big)(p_{0,j}(t) - \alpha)\\
&\overset{(a)}{=} \sum_{j\in V_2} \big(p_{0,j}(t') - p_{i,j}(t')\big)(p_{0,j}(t) - \alpha)\\
&+ \sum_{j'=g(j)\in V_3} \big(p_{0,j'}(t') - p_{i,j'}(t')\big)(p_{0,j'}(t) - \alpha)\\
&\overset{(b)}{=} \sum_{j\in V_2} \big(p_{0,j}(t') - p_{i,j}(t')\big)(p_{0,j}(t) - \alpha)\\
&+ \sum_{j\in V_2} \big(p_{i,j}(t') - p_{0,j}(t')\big)(p_{i,j}(t) - \alpha)\\
&= \sum_{j\in V_2} \big(p_{0,j}(t') - p_{i,j}(t')\big)\big(p_{0,j}(t) - p_{i,j}(t)\big).
\end{aligned}$$

Equality (a) comes from partitioning nodes to sets V_1, V_2 and V_3 according to Definition 10. The terms correspond to nodes in the partition set V_1 is equal to zero. Equality (b) comes from the fact that $d(0,j') = d(i,j)$ and $d(0,j) = d(i,j')$. Thus, $p_{0,j'}(.) = p_{i,j}(.)$ and $p_{i,j'}(.) = p_{0,j}(.)$.

Therefore, if $t' = t$, we have,

$$\mathbb{E}[\mathcal{H}_\alpha(i,t')] - \mathbb{E}[\mathcal{H}_\alpha(0,t')] = \sum_{j\in V_2} \big(p_{0,j}(t) - p_{i,j}(t)\big)^2, \qquad (7.8)$$

which is strictly positive if $i \neq 0$.

Now we consider the case that $t' \neq t$. Suppose $d_{0,j} < d_{i,j}$. Then, according to Proposition 3, $p_{0,j}(t) > p_{0,j}(t)$ for any value of $t > 0$. Therefore, $\big(p_{0,j}(t') - p_{i,j}(t')\big)\big(p_{0,j}(t) - p_{i,j}(t)\big) > 0$. The same argument holds if $d_{0,j} > d_{i,j}$. If $d_{0,j} = d_{i,j}$, then $\big(p_{0,j}(t') - p_{i,j}(t')\big)\big(p_{0,j}(t) - p_{i,j}(t)\big) = 0$. This completes the proof of Theorem 2.

7.7 Proof of Theorem 3

To simplify notation, we prove this Theorem for a specific case where there are three sources in the network ($m = 3$). Also, we drop ϵ from d_1^ϵ. All arguments can be extended to a general case. Let $\mathcal{S} = \{0,1,2\}$ be the sources. Suppose at the first step of the Algorithm 3, we have inferred the source node 0. We show that at the next step, we have,

$$\mathbb{E}[\mathcal{L}_{d_0}(s,t)] \geq \mathbb{E}[\mathcal{L}_{d_0}(i,t)], \qquad (7.9)$$

for $s \in \{1,2\}$ and for all $i \in V^t - D(0,d_1)$.

Consider a node i in the d_1-neighborhood of the source node 1 (i.e., $i \in D(1,d_1)$. Consider a node j in d_0-neighborhood of node i (Figure 17). According to Equation (3.18), we have,

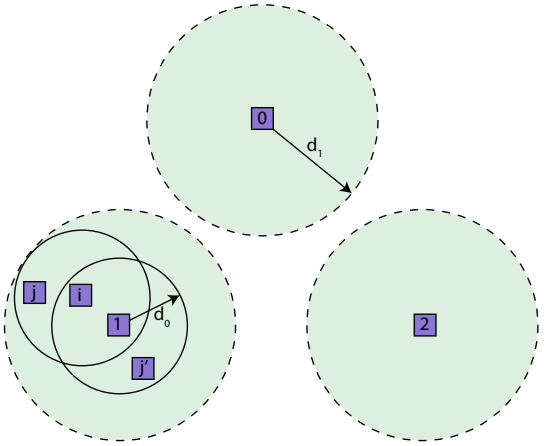

Figure 17: An illustrative figure of the proof of Theorem 3.

$$
\begin{aligned}
&Pr(y_j(t)=1) \geq p_{1,j} \qquad (7.10)\\
&Pr(y_j(t)=1) \overset{(I)}{\leq} \sum_{s=1}^{3} p_{s,j}\\
&\qquad\qquad \overset{(II)}{\leq} p_{1,j} + \frac{\epsilon}{n},
\end{aligned}
$$

where Inequality (I) comes from the union bound of probabilities, and Inequality (II) uses incoherent source property of Definition 6.

Consider a node i in the d_1-neighborhood of the source node 1 (i.e., $i \in D(1,d_1)$). For this node, we have,

$$
\begin{aligned}
\mathbb{E}[\mathcal{L}_{d_0}(i,t)] &= \sum_{j\in D(i,d_0)} Pr\big(y_j(t)=1\big)\log\big(p_{i,j}(t)\big) + Pr\big(y_j(t)=0\big)\log\big(1-p_{i,j}(t)\big) \qquad (7.11)\\
&\overset{(III)}{=} \sum_{j\in D(i,d_0)} (p_{1,j}(t)+\frac{\epsilon_j}{n})\log\big(p_{i,j}(t)\big) + (1-p_{1,j}(t)-\frac{\epsilon_j}{n})\log\big(1-p_{i,j}(t)\big)\\
&= \sum_{j\in D(i,d_0)} p_{1,j}(t)\log\big(p_{i,j}(t)\big) + (1-p_{1,j}(t))\log\big(1-p_{i,j}(t)\big)\\
&+ \underbrace{\sum_{j\in D(i,d_0)} \frac{\epsilon_j}{n}\log\frac{p_{i,j}(t)}{1-p_{i,j}(t)}}_{\text{term IV}}\\
&\asymp \sum_{j\in D(i,d_0)} p_{1,j}(t)\log\big(p_{i,j}(t)\big) + (1-p_{1,j}(t))\log\big(1-p_{i,j}(t)\big),
\end{aligned}
$$

where Equality (I) comes from Equation (7.10), and term (IV) goes to zero for sufficiently large n and a fixed t.

Similarly to the proof of Theorem 1, we have

$$\sum_{j\in D(i,d_0)\cap D(1,d_0)} p_{1,j}(t)\log\big(p_{1,j}(t)\big)+(1-p_{1,j}(t))\log\big(1-p_{1,j}(t)\big) \tag{7.12}$$
$$\geq \sum_{j\in D(i,d_0)\cap D(1,d_0)} p_{1,j}(t)\log\big(p_{i,j}(t)\big)+(1-p_{1,j}(t))\log\big(1-p_{i,j}(t)\big).$$

Note that this inequality holds for nodes in the d_0-neighborhood of both nodes i and 1. Now consider a node $j \in D(i,d_0)-D(1,d_0)$, and a node $j' \in D(1,d_0)-D(i,d_0)$ (see Figure 17). Owing to the symmetric structure of the network, similarly to Lemma 2, there is a one-to-one map among nodes j and j' such that $d(i,j)=d(1,j')$. For such node pairs j and j', we have,

$$\begin{aligned} &p_{1,j'}(t)\log(p_{1,j'}(t))+\big(1-p_{1,j'}(t)\big)\log\big(1-p_{1,j'}(t)\big) \qquad (7.13)\\ &-p_{1,j}(t)\log(p_{i,j}(t))-\big(1-p_{1,j}(t)\big)\log\big(1-p_{i,j}(t)\big)\\ &=p_{1,j'}(t)\log(p_{1,j'}(t))+\big(1-p_{1,j'}(t)\big)\log\big(1-p_{1,j'}(t)\big)\\ &-p_{1,j}(t)\log(p_{1,j'}(t))-\big(1-p_{1,j}(t)\big)\log\big(1-p_{1,j'}(t)\big)\\ &=(p_{1,j'}(t)-p_{1,j}(t))\log\frac{p_{1,j'}(t)}{1-p_{1,j'}(t)}\\ &\geq 0, \qquad (7.14)\end{aligned}$$

where the inequality comes from the fact that $d(1,j')<d(1,j)$. Thus, we have,

$$\sum_{j\in D(1,d_0)-D(1,d_0)} p_{1,j}(t)\log\big(p_{1,j}(t)\big)+(1-p_{1,j}(t))\log\big(1-p_{1,j}(t)\big) \tag{7.15}$$
$$\geq \sum_{j\in D(i,d_0)-D(1,d_0)} p_{1,j}(t)\log\big(p_{i,j}(t)\big)+(1-p_{1,j}(t))\log\big(1-p_{i,j}(t)\big).$$

Combining Inequalities (7.12) and (7.15), we have,

$$\mathbb{E}[\mathcal{L}_{d_0}(1,t)]\geq\mathbb{E}[\mathcal{L}_{d_0}(i,t)], \tag{7.16}$$

for any node i in the d_1-neighborhood of the source node 1. The same arguments can be repeated for nodes in the d_1-neighborhood of the source node 2. There are some remaining nodes that are not in the d_1-neighborhood of the sources. As the last step of the proof, we show that the probability of having an infected remaining node is small. Consider node j such that $d(j,\mathcal{S})>d_1$. according to Equation (3.18) and using the probability union bound, we have

$$Pr(y_j(t)=1)\leq\frac{\epsilon}{n}. \tag{7.17}$$

Let p_e denote the probability of at least one such infected node exists. We have,

$$\begin{aligned} p_e &\leq 1-\big(1-\frac{\epsilon}{n}\big)^n \qquad (7.18)\\ &\asymp \epsilon,\end{aligned}$$

for sufficiently large n. This completes the proof of Theorem 3.

7.8 Proof of Proposition 7

Computation of the k-path network diffusion kernel for infected nodes has computational complexity $\mathcal{O}\big(V^t(k|E| + kn\log(n))\big)$ according to Proposition 2. Moreover, solving Optimization (3.21) for m iterations costs $\mathcal{O}(|V^t|nm)$. Thus, the total computational complexity of Algorithm 3 is $\mathcal{O}\big(|V^t|(k|E| + kn\log(n) + mn)\big)$.

8 Acknowledgments

Authors thank Gerald Quon for discussions and an early processing of disease datasets.

References

[1] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi, "Spread of (mis) information in social networks," *Games and Economic Behavior*, vol. 70, no. 2, pp. 194–227, 2010.

[2] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Physical review letters*, vol. 86, no. 14, p. 3200, 2001.

[3] D. Hirshleifer and S. Hong Teoh, "Herd behaviour and cascading in capital markets: A review and synthesis," *European Financial Management*, vol. 9, no. 1, pp. 25–66, 2003.

[4] M. T. Maurano, R. Humbert, E. Rynes, R. E. Thurman, E. Haugen, H. Wang, A. P. Reynolds, R. Sandstrom, H. Qu, J. Brody *et al.*, "Systematic localization of common disease-associated variation in regulatory dna," *Science*, vol. 337, no. 6099, pp. 1190–1195, 2012.

[5] A. B. Glinskii, J. Ma, S. Ma, D. Grant, C.-U. Lim, S. Sell, and G. V. Glinsky, "Identification of intergenic trans-regulatory rnas containing a disease-linked snp sequence and targeting cell cycle progression/differentiation pathways in multiple common human disorders," *Cell Cycle*, vol. 8, no. 23, pp. 3925–3942, 2009.

[6] N. T. Bailey *et al.*, *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975.

[7] M. E. Newman, "Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality," *Physical review E*, vol. 64, no. 1, p. 016132, 2001.

[8] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *Information Theory, IEEE Transactions on*, vol. 57, no. 8, pp. 5163–5181, 2011.

[9] E. Sefer and C. Kingsford, "Diffusion archaeology for diffusion progression history reconstruction."

[10] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.

[11] M. Newman, *Networks: an introduction*. Oxford University Press, 2010.

[12] M. E. Newman, "Spread of epidemic disease on networks," *Physical review E*, vol. 66, no. 1, p. 016128, 2002.

[13] C. Moore and M. E. Newman, "Epidemics and percolation in small-world networks," *Physical Review E*, vol. 61, no. 5, p. 5678, 2000.

[14] A. Ganesh, L. Massoulié, and D. Towsley, "The effect of network topology on the spread of epidemics," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2. IEEE, 2005, pp. 1455–1466.

[15] N. Demiris and P. D. O'neill, "Bayesian inference for epidemics with two levels of mixing," *Scandinavian journal of statistics*, vol. 32, no. 2, pp. 265–280, 2005.

[16] P. D. ONeill, "A tutorial introduction to bayesian inference for stochastic epidemic models using markov chain monte carlo methods," *Mathematical biosciences*, vol. 180, no. 1, pp. 103–114, 2002.

[17] H. Okamura, K. Tateishi, and T. Dohi, "Statistical inference of computer virus propagation using non-homogeneous poisson processes," in *Software Reliability, 2007. ISSRE'07. The 18th IEEE International Symposium on.* IEEE, 2007, pp. 149–158.

[18] S. Mostafavi, A. Goldenberg, and Q. Morris, "Labeling nodes using three degrees of propagation," *PloS one*, vol. 7, no. 12, p. e51947, 2012.

[19] Y. Bengio, O. Delalleau, and N. Le Roux, "Label propagation and quadratic criterion," *Semi-supervised learning*, pp. 193–216, 2006.

[20] M. E. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, p. 026126, 2003.

[21] J. Hershberger, M. Maxel, and S. Suri, "Finding the k shortest simple paths: A new algorithm and its implementation," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4, p. 45, 2007.

[22] C. De Bacco, S. Franz, D. Saad, and C. H. Yeung, "Shortest node-disjoint paths on random graphs," *arXiv preprint arXiv:1401.8096*, 2014.

[23] P. Erdős and A. Rényi, "On the strength of connectedness of a random graph," *Acta Mathematica Hungarica*, vol. 12, no. 1, pp. 261–267, 1961.

[24] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, no. 2, 1995, pp. 1137–1145.

[25] W. Aiello, F. Chung, and L. Lu, "A random graph model for power law graphs," *Experimental Mathematics*, vol. 10, no. 1, pp. 53–66, 2001.

[26] K. Lerman, R. Ghosh, and T. Surachawala, "Social contagion: An empirical study of information spread on digg and twitter follower graphs," *arXiv preprint arXiv:1202.3162*, 2012.

[27] P. R. Burton, D. G. Clayton, L. R. Cardon, N. Craddock, P. Deloukas, A. Duncanson, D. P. Kwiatkowski, M. I. McCarthy, W. H. Ouwehand, N. J. Samani *et al.*, "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," *Nature*, vol. 447, no. 7145, pp. 661–678, 2007.

[28] S. Feizi, G. Quon, M. Mendoza, M. Médard, and M. Kellis, "Constructing regulatory networks in human, fly and worm by integrative inference of genome-wide functional and physical associations," *in preparation*.

[29] A. V. Segrè, L. Groop, V. K. Mootha, M. J. Daly, D. Altshuler, D. Consortium, M. Investigators *et al.*, "Common inherited variation in mitochondrial genes is not enriched for associations with type 2 diabetes or related glycemic traits," *PLoS genetics*, vol. 6, no. 8, p. e1001058, 2010.

[30] A. J. de Ruijter, R. J. Meinsma, P. Bosma, S. Kemp, H. N. Caron, and A. B. van Kuilenburg, "Gene expression profiling in response to the histone deacetylase inhibitor bl1521 in neuroblastoma," *Experimental cell research*, vol. 309, no. 2, pp. 451–467, 2005.

[31] T. Valentino, D. Palmieri, M. Vitiello, G. Pierantoni, A. Fusco, and M. Fedele, "Patz1 interacts with p53 and regulates expression of p53-target genes enhancing apoptosis or cell survival based on the cellular context," *Cell death & disease*, vol. 4, no. 12, p. 963, 2013.