Diffusion Source Identification on Networks with Statistical Confidence

Quinlan Dawkins [1] **Tianxi Li** [2] **Haifeng Xu** [1]

Abstract

Diffusion source identification on networks is a problem of fundamental importance in a broad class of applications, including rumor controlling and virus identification. Though this problem has received significant recent attention, most studies have focused only on very restrictive settings and lack theoretical guarantees for more realistic networks. We introduce a statistical framework for the study of diffusion source identification and develop a confidence set inference approach inspired by hypothesis testing. Our method efficiently produces a small subset of nodes, which provably covers the source node with any prespecified confidence level without restrictive assumptions on network structures. Moreover, we propose multiple Monte Carlo strategies for the inference procedure based on network topology and the probabilistic properties that significantly improve the scalability. To our knowledge, this is the first diffusion source identification method with a practically useful theoretical guarantee on general networks. We demonstrate our approach via extensive synthetic experiments on well-known random network models, a large data set of hundreds of real-world networks, as well as a mobility network between cities concerning the COVID-19 spreading.

1 Introduction

One pressing problem today is the spreading of misinformation or malicious attacks/virus in various cyberspaces. For example, rumors and fake news on social networks may result in many serious political, economic, and social issues (Vosoughi et al., 2018). Viruses that spread via emails and computer communication may cause severe privacy and leakage problems (Newman et al., 2002; Halperin & Almogy, 2002; Xu & Ren, 2016). The negative impacts stem from a few source users/locations and then spread over the social networks via a *diffusion process* in such events. One crucial step to reduce the loss from such an event is to quickly identify the sources so that counter-measures can be taken in a timely fashion.

Though early practices have been done for this important problem with motivations from various domains, systematic research on this problem only began very recently, arguably starting from the seminal work of (Shah & Zaman, 2011), which proposed a *rumor center* estimator that can be located by an efficient message-passing algorithm with linear time complexity. Despite the significant interest and progress on this problem in recent years (Shah & Zaman, 2012; Dong et al., 2013; Khim & Loh, 2016; Bubeck et al., 2017; Yu et al., 2018; Crane & Xu, 2020), many challenges remain unaddressed. First, the theoretical understanding of these methods is currently only available under very restrictive and somewhat unrealistic structural assumptions of the networks such as *regular trees*. This is perhaps partially explained by the well-known computational hardness about the probabilistic inference of diffusion process in general graphs (Shapiro & Delgado-Eckert, 2012). Therefore, intuitive approximations have been used for general networks (Nguyen et al., 2016; Kazemitabar & Amini, 2020). However, such methods lack theoretical guarantees. Second, even for regular trees, the available performance guarantee is far from being useful in practice. Even in the most idealized situation of infinite regular trees, the correct probability of the rumor center is almost always below 0.3 (Shah & Zaman, 2011; Dong et al., 2013; Yu et al., 2018). For general graphs, as we show later, the correct rate of such a *single-point* estimation method only becomes too low to be practical.

To guarantee higher success probability, a typical approach, as in both machine learning theory (Valiant, 1984) and data-driven applied models (LeCun et al., 2015), is perhaps to obtain more data. However, a fundamental challenge in diffusion source identification (DSI) is that *the problem by nature has only one snapshot of the network information*, i.e., the earliest observation about the infection status

[1]Department of Computer Science, University of Virginia, Charlottesville, Virginia, USA [2]Department of Statistics, University of Virginia, Charlottesville, Virginia, USA. Correspondence to: Tianxi Li <tianxili@virginia.edu>, Haifeng Xu <hx4ad@virginia.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

of the network.[1] Therefore, compared to classic learning tasks, DSI poses a fundamentally different challenge for inference.It is the above crucial understanding that motivates our adoption of a different statistical inference technique, the confidence set. Previously systematic statistical studies adopt the confidence set approach for DSI on trees (Bubeck et al., 2017; Khim & Loh, 2016; Crane & Xu, 2020). Though they enjoy good theoretical properties, the methods are applicable only on infinite trees.

This paper aims to bridge the gap between practically useful algorithms and theoretical guarantees for the DSI problem. We introduce a new statistical inference framework which provably includes many previous methods (Shah & Zaman, 2011; Nguyen et al., 2016) as special cases. Our new framework not only highlights the drawback of the previous methods but, more importantly, also leads to the design of our confidence set inference approach with *finite-sample* theoretical guarantee on *any* network structures.

As a demonstration, consider the example of the COVID-19 spreading procedure in early 2020. Figure 1 shows a travel mobility network between 49 major cities in China, constructed from the two-week travel volume (Lab, 2020; Hu et al., 2020) before the virus caught wide attention. The square nodes (21 out of 49) are all cities with at least five confirmed cases of the virus on Jan 24, 2020. The DSI problem is: given only knowledge about the mobility network and which cities have detected a notable amount of confirmed cases (in this case, at least 5) , can we identify in which city the virus was first detected?

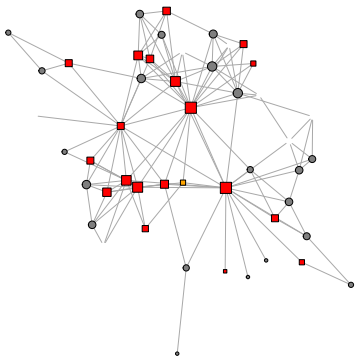

Figure 1: The mobility network and the COVID-19 infection status of major Chinese cities on Jan 24, 2020. Colored square nodes are cities with at least five confirmed cases.

This problem turns out to be too difficult for precise identification. None of the single-point source identification methods under evaluation can successfully identify Wuhan due to its relatively non-central position from the network (details in Section 5). Nevertheless, both of our 80% and 90% confidence sets cover Wuhan correctly, giving recommendations of 6 nodes and 11 nodes (out of 49 cities), respectively. In fact, the evaluation on all the whole week after the lockdown of Wuhan reveals that both confidence sets correctly cover Wuhan in all the seven days, while the single-point estimation methods are rarely effective. Such a result evidently shows the necessity of adopting confidence set approach and the effectiveness of our solution.Our contributions in this paper can be summarized in three-folds.

1. We introduce an innovative statistical framework for the DSI problem. It includes several previous methods as special cases, but has the potential for more effective inference.

2. Under our framework, we propose a general way to construct the source node confidence set, whose validity can be guaranteed for finite sample size and any network structures. It is the first DSI method with a theoretical performance guarantee on general networks, to the best of our knowledge.

3. We propose techniques that dramatically improve the computational efficiency of our inference algorithm. En route, we develop a generalized importance sampling method, which may be of independent interest.

A high-level message in the paper is that the confidence set approach, which did not receive adequate attention in the machine learning literature, can be an important tool for inference tasks, especially for challenging problems with limited available data.

2 Preliminaries

We start by formalizing the *Diffusion Source Identification* (DSI) problem, introduced in the seminal work of Shah & Zaman (2011). Consider a network G with node set $V = \{1, \cdots, n\}$ and edge set E. For ease of presentation, we focus on unweighted and undirected networks but it is straightforward to generalize the model and our framework to weighted networks. We write $(u, v) \in E$ if node u and v are connected. The network can be equivalently represented by its $n \times n$ binary *adjacency matrix* A, where $A_{uv} = A_{vu} = 1$ if and only if $(u, v) \in E$.

There is a *source node* $s^* \in V$ on the network G initiating a diffusion of a certain effect (rumor, fake news or some virus) over the network G. We embed our inference of the diffusion procedure under the widely-adopted "Susceptible-Infected" (SI) model (Anderson & May, 1992; Shah & Zaman, 2011), though our approach can be easily tailored to other diffusion procedure as well.

[1] Since infected nodes are usually indistinguishable and equally infectious, any additional information in later observations only tells us which *new* or *additional* nodes are infected and is not helpful for us to infer the source node.

In the SI model, the source node s^* is the only "infected" node initially. The infection diffuses as follows: given the set of currently infected nodes after $t-1$ infections, the next infection happens by sampling uniformly at random one of the *edges* connecting an infected node and a susceptible node. Consequently, a full *diffusion path* with T infections can be represented by a sequence of $T+1$ nodes in the infection order. We define the *diffusion path space* to be

$$\mathcal{Z}_T = \{\boldsymbol{v} = \{s^* = v_0, v_1, \cdots, v_T\} : v_t \in V, v_{t_1} \neq v_{t_2} \text{ if } t_1 \neq t_2, \text{ and } (v_t, v_{t'}) \in E \text{ for some } t < t'\}$$

However, in practice, when the occurrence of the infection is noticed, we have already lost the information about the diffusion path. Instead, the available data only contain the *snapshot* of the current infection status on the network without the infection order. Formally, the data can be represented as an n-dimensional binary vector y with $y_i = \mathbb{I}(i \text{ is infected}) \in \{0,1\}$, where $\mathbb{I}$ is the standard indicator function. Therefore, the *sample space* of the DSI problem can be defined as

$$\mathcal{Y}_T = \{y \in \{0,1\}^n : \|y\|_1 = T, \text{such that } \{i : y_i = 1\} \text{ induces a connected subgraph of } G\}.$$

Equivalently, we will also think of any $y \in \mathcal{Y}_T$ as the a infected subset of nodes $V_I \subset V$ with size T. The DSI problem can then be defined as follows.

Definition 1 (Diffusion Source Identification)**.** *Given one sample $y \in \mathcal{Y}_T$, identify the source node s^* of the diffusion process that generates y.*

Challenges. The challenge of DSI intrinsically arises from the loss of information in the observed data. Specifically, by definition, we have a *many-to-one* mapping $\zeta : \mathcal{Z}_T \to \mathcal{Y}_T$, such that $\zeta(\cdot)$ maps a diffusion path to the corresponding infection snapshot of the network. Information about the infection order has been lost upon the observation of data y. Nevertheless, the DSI problem looks to identify the first node in the infection order, with access to only one snapshot of the infection status. Note that obtaining multiple snapshots over time does not reduce the difficulty of DSI. This is because, given the current snapshot, later observed data carry no additional information about the source node due to the Markov property of the SI model.

3 A General Statistical Framework for DSI with Confidence Guarantees

3.1 DSI as Parameter Estimation

We start by formulating DSI under a systematic statistical framework, which will help in our design of better inference methods later on. Treating the network G as fixed and s^* as the model parameter, the probability of generating data $y \in \mathcal{Y}_T$ can be represented by $\mathbb{P}_{s^*}(Y = y) = p(y|s^*)$. where random variable Y denotes the observed data. The identification of s^* can then be treated as a parameter estimation problem. Specifically, we consider the following general parameter estimation framework. Given any *discrepancy function* $\ell : \mathcal{Y}_T \times \mathcal{Z}_T \to [0, \infty)$, we want to find an estimator of s^* based on the following optimization problem:

$$\text{minimize}_s \ \mathbb{E}_s \ell(y, Z) \tag{1}$$

in which $Z \in \mathcal{Z}_T$ is the random diffusion path following the SI model starting from parameter s and $\mathbb{E}_s$ denotes the expectation over Z. That is, we look to select the s that the diffusion path Z it generates has the minimum expected discrepancy from our observed data y.

Remark 1. An important design here is that the discrepancy function ℓ is defined on $\mathcal{Y}_T \times \mathcal{Z}_T$, not on $\mathcal{Y}_T \times \mathcal{Y}_T$. That is, y will be compared with the random diffusion path while not merely the snapshot induced by the path. This is because Z contains richer information about the diffusion process. As we show later, this turns out to be very crucial for designing effective discrepancy functions.

Notice that our framework include a few previous methods as special cases. Due to space limit, all formal proofs in this paper have been deferred to the Appendix. Instead, intuition and explanations are provided as needed.

Proposition 1. *1. If $\ell_{rc}(y, z) = 1 - \mathbb{I}(y = \zeta(z))$, when the network is an infinite regular tree, procedure (1) gives the rumor center of Shah & Zaman (2011).*

2. If $\ell_{se}(y, z) = \|y - \zeta(z)\|_2^2$, the squared Euclidean distance between y and $\zeta(z)$, the discrepancy is equivalent to the symmetric difference in (Nguyen et al., 2016).[2]

Proposition 1 also reveals some key drawbacks of the rumor center method and its variants. First, the discrepancy function ℓ_{rc} only takes two values, and it treats all configurations z with $\zeta(z) \neq y$ equally. Therefore, such a function may not be sufficiently sensitive for general networks. From this perspective, ℓ_{se} is potentially better. Second, and importantly, both of the above discrepancy functions only depend on $\zeta(z)$, failing to leverage the *diffusion order* of the z. Ignoring such information may also undermine the performance. To overcome these drawbacks, we propose the following family of discrepancy functions as a better alternative. We call this family the *canonical family* of discrepancy functions.

Definition 2 (Canonical Discrepancy Functions)**.** *Consider a class of discrepancy functions ℓ that can be written in the*

[2] However, different from our framework, (Nguyen et al., 2016) used an approximation metric to this discrepancy for DSI.

following form

$$\ell(y,z) = -\sum_{v:y_v=1} \mathbb{I}(v \in z) h(t_z(v)), \tag{2}$$

*in which $t_z(v)$ is the infection order of node v in path z and h is a **non-increasing weight function**. When $v \notin z$, we define $t_z(v) = \infty$.*

The canonical form (2) is essentially a negative similarity function. It incorporates both the infection status and the infection order of z. The weight function h incorporates the diffusion order such that if z deviates from y at an early stage, the deviation is treated as a stronger signal for their discrepancy, compared with the case when they only deviates at a later stage of the diffusion. Conceptually, this canonical family is general enough to incorporate the needed information for the diffusion process. In addition, as shown in Section 3.4, it admits fundamental properties that make the computation very efficient. As a special case, we demonstrate that ℓ_{se} is equivalent to a discrepancy function with $h(t_z(v)) \equiv 2$, as follows

$$\|y - \zeta(z)\|_2^2 = \sum_{i=1}^{n} \mathbb{I}(y_i \neq \zeta(z)_i) = 2T - 2 \sum_{v:y_v=1} \mathbb{I}(v \in z).$$

Therefore L_2 is equivalent to Eq. (2) with $f(t_z(v)) \equiv 2$.

In this paper, we are particularly interested in the following natural configuration as the discrepancy function, which we call the "**A**veraged **D**eviation - **i**nverse **T**ime" (ADiT), which takes the canonical family form (2) with the inverse time weights:

$$h(t_z(v)) = \frac{1}{t_z(v)}. \tag{3}$$

In Table 1 of Section 5, we show the simulation performance of the single-point estimation by our framework compared to other methods. Though our methods demonstrate improvements, the accuracy is **universally low** in all situations for all methods. Such an observation indicates that it is generally impossible to recover the source node by a single estimator with high accuracy. Indeed, as shown in Shah & Zaman (2011); Dong et al. (2013); Yu et al. (2018), even in the ideal infinite regular tree for which the rumor center is proved to be optimal in the MLE sense, the probability of correct source node identification turns out to still be low (≤ 0.3). Such a low accuracy is far from useful in real-world applications, suggesting the necessity of developing alternative forms of inference, which is we embark on in the next section.

3.2 Confidence Set

As mentioned previously, single point estimators suffer from low success rates, rendering them unsatisfactory in real-world applications. To identify the source node with a nontrivial performance guarantee, we propose constructing a small subset of nodes that provably contains the source nodes with any pre-defined confidence. This insight motivates our use of the *confidence set* as the DSI method.

Definition 3. *Let Y be the random infection status of the stochastic diffusion process starting from s^*. A level $1-\alpha$ confidence set of the source node is a **random** set $S(Y) \subset V$ depending on Y for which*

$$\mathbb{P}(s^* \in S(Y)) \geq 1 - \alpha.$$

Surprisingly, the idea of using confidence set to infer the diffusion source – though arguably a natural one in statistics – has not been explored much in the context of DSI. The most relevant to ours are probably Bubeck et al. (2017); Khim & Loh (2016) and Crane & Xu (2020). Bubeck et al. (2017) considered identifying the first node of a growing tree but not a diffusion process. Khim & Loh (2016) extended the idea to the SI model but only for infinite regular tree and asymptotic setting. Despite its theoretical merits, this method is not practical. For example, even consider the situation of an infinite 4-regular tree as the network structure, applying the method of Khim & Loh (2016) would indicate a confidence set of size $4^{11} \approx 5 \times 10^6$, regardless of the infected size T. This is far too large for almost any applications, let alone the fact that infinite regular tree itself is unrealistic. Crane & Xu (2020) makes the inference more effective, but still rely on the tree-structure assumption.

We instead take a completely different yet natural approach based on our statistical framework for the problem. To ensure the validity of the inference for any network structures, we will rely on the general statistical inference strategy for the confidence set construction. We first introduce a testing procedure for the hypothesis $H_0 : s^* = s$ against the alternative hypothesis $H_1 : s^* \neq s$. Given a discrepancy function ℓ, data y and the node s under evaluation, define the testing statistic to be our loss $T_s(y) = \mathbb{E}_s \ell(y, Z)$ for any data y. Then the *p-value* of the test is defined to be

$$\psi_s = \mathbb{P}_s(T_s(\zeta(Z)) \geq T_s(y)). \tag{4}$$

where the probability $\mathbb{P}_s$ is over the randomness of the path Z generated from the random diffusion process starting from s. The p-value is the central concept in statistical hypothesis testing, and it gives a probabilistic characterization of how extreme the observed y deviates from the expected range for random paths that are truly from s (Lehmann & Romano, 2006). For a level $1-\alpha$ confidence set, we compute ψ_s for all nodes s and construct the confidence set by

$$S(y) = \{s : \psi_s(y) > \alpha\}. \tag{5}$$

The following result guarantees the validity of the confidence set constructed above.

Theorem 1. *The confidence set constructed by (5) is a valid $1-\alpha$ confidence set.*

Notice that Theorem 1 is a general result, independent of the network structure or the specific test statistic we use. However, the validity of the confidence set only gives one aspect of the inference. We would like to have small confidence sets in practice since such a small set would narrow down our investigation more effectively. The confidence set size would depend on the network structure and the corresponding effectiveness of the discrepancy function (the test statistic). We will use the proposed ADiT to define our test statistic. As shown in our empirical study, it gives excellent and robust performance across various settings.

3.3 Algorithmic Construction of Confidence Sets

The exact evaluation of the statistic $T_s(y)$ and p-value ψ_s is infeasible for general graphs since the probability mass function of the SI model is intractable. To overcome this barrier, we resort to the Monte Carlo (MC) method for approximate calculation, with details in Algorithm 1. This vanilla version turns out to be computationally inefficient. However, we will introduce techniques to significantly improve its computation efficiency afterwards.

Remark 2 (Monte Carlo setup). Note that we have two layers of Monte Carlo evaluations. The first layer is the loss function calculation in (6) and (7), while the second layer is the p-value evaluation (8). The first layer shares the same m samples. This is different from the classical Monte Carlo, but would not break the validity for p-value calculation. The properties of p-value calculation by Monte Carlo method have been studied in detail by (Jockel, 1986; Besag & Clifford, 1989).

Remark 3 (Choice of the sample number m). In theory, the computation in Algorithm 1 is exact when $m \to \infty$. In practice, simple guidance about the choice of m can be derived as follows. The critical step in Algorithm 1 is Step 7 for the p-value calculation since the MC errors from previous steps are usually in a lower order. For the correctness, we only need to worry about the evaluation at node s^* when the true p-value is close to α. Step 7 averages over m indicators. By the central limit theorem, the MC estimate at most misses the true p-value by roughly $2\sqrt{\alpha(1-\alpha)/m}$. For example, if we are aiming for a 90% confidence set where $\alpha = 0.1$, setting $m = 10000$ would indicate that the MC at most misses the targeting confidence level by 0.006%, which is usually good enough in most applications. In our experiments, we use this $m = 10000$ and it has been sufficient in all situations. Notice that this recommendation is more conservative than the ones used in classical statistical inference problems (Jockel, 1986). In our experience, it might still be acceptable to use a smaller m.

Algorithm 1 Vanilla MC for Confidence Set Construction

1: **Input:** MC sample number m, confidence level α
2: **Input:** Network G, data y, discrepancy function ℓ
3: **for** each infected node $s \in y$ **do**
4: Generate $2m$ samples $z_i \in \mathcal{Z}, i = 1, \cdots, 2m$ from the T-round diffusion process with source s on G.
5: Estimate expected loss $T_s(y)$ of data y as

$$\hat{T}_s(y) = \frac{1}{m} \sum_{i=m+1}^{2m} \ell_s(y, z_i). \tag{6}$$

6: For path $z_j, j = 1, \cdots, m$, estimate $T_s(\zeta(z_j))$ as

$$\hat{T}_s(\zeta(z_j)) = \frac{1}{m} \sum_{i=m+1}^{2m} \ell(\zeta(z_j), z_i). \tag{7}$$

7: Estimate the p-value $\psi_s(y)$ as

$$\hat{\psi}_s(y) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}(\hat{T}_s(\zeta(z_j)) \geq \hat{T}_s(y)). \tag{8}$$

8: **end for**
9: **return** level $1-\alpha$ confidence set:

$$\mathcal{C}_\alpha(y) = \{s \in V_I : \hat{\psi}_s(y) > \alpha\}.$$

Remark 4 (Time complexity of the vanilla MC, and its trivial parallelization). The time complexity of a standard sequential implementation of Algorithm 1 is $\tilde{O}(mT^2 + m^2T^2)$;[3] (1) the first term is due to the MC sampling (Bringmann & Panagiotou, 2017); (2) the second term is from the statistic calculation (7) given the MC samples. However, our algorithm can be trivially parallelized. In particular, the for-loop in Step 3 can be distributed across different $s \in V_I$ with any communication. This leads to a parallel time complexity $\tilde{O}(mT + m^2T)$. It is worthwhile to compare this time cost with the rumor center of (Shah & Zaman, 2011) which has $\tilde{O}(dT)$ linear complexity and d is the maximum node degree. But the algorithm has to be sequential (thus non-parallelizable). In summary, Algorithm 1 has a better dependence on the network density captured by d but has an additional quadratic dependence on the number of samples m.

[3] As a convention, the $\tilde{O}$ notation omits logrithmic terms.

3.4 Fast Loss Estimation for the Canonical Family

A major computational bottleneck of Algorithm 1 is the $O(m^2T)$ time for estimating $\mathbb{E}_s(\ell(y, Z))$ in Equation (7) for every j since we have to compute $\hat{\psi}$ for m samples, and each $\hat{\psi}$ is the average over another m samples. Fortunately, it turns out that, for canonical discrepancy family, this step can be done in $O(mT)$ time, highlighting another advantage of our proposed family of cost functions.

Instead of computing $\hat{T}_s$ in Equation (7) by summing over the sample $i = m+1, \cdots, 2m$, we can compute $\hat{T}_s$ directly using only the "summary information" of these samples that can be computed and cached in advance. This insight is possible due to the following alternative representation of the $\hat{T}_s(y)$ function in Equation (7):

$$\begin{aligned}\hat{T}_s(y) &= -\frac{1}{m}\sum_{v:y_v=1}\sum_{i=m+1}^{2m}\sum_{k=1}^{T} h(k)\mathbb{I}(t_{z_i}(v)=k)\\ &= -\frac{1}{m}\sum_{v:y_v=1}\sum_{k=1}^{T} M_{v,k}h(k) \qquad (9)\end{aligned}$$

where $M_{v,k}$ counts the total number of samples in $z_{m+1}, \cdots, z_{2m}$ in which node v is the k'th infected node in the infection path. Let $M \in \mathbb{R}^{n\times T}$ be the matrix containing the entries $M_{v,k}$. Note that, there are at most mT nonzero entries in M since each sample only has T nodes. These entries can be computed in $O(mT)$ time simply by updating the corresponding $M_{v,k}$ entries during sampling. With these non-zero $M_{v,k}$ entries, we can then compute $\hat{h}(v) = \sum_{k=1}^{T} M_{v,k}h(k)$ for all the v that showed up in our samples in $O(mT)$ time. Finally, given the previous $\hat{h}(v)$, we can compute any $\hat{T}_s(y)$ in $O(T)$ time where $y = \zeta(z_1), \cdots, \zeta(z_m)$, which thus in total takes an additional $O(mT)$ time. This overall takes $O(mT)$ time.

4 Monte Carlo Acceleration via Pooled Sampling

In subsection 3.4, we reduced the computation time for estimating a single p-value to $\tilde{O}(mT)$, which is arguably the minimum possible in our framework since even sampling m samples already takes $\tilde{O}(mT)$. In this section, we will introduce efficient strategies to reduce another major computational cost in our algorithm – the MC sampling. Our techniques will "borrow" MC samples of one node for the inference task of another node by leveraging the network structure and properties of the SI model. Consequently, we only need to generate MC samples for a subset of the infected nodes, which may effectively reduce the computational cost.

4.1 Surjective Importance Sampling for Single-Degree Nodes

A node with only one connection in the network is called a *single-degree node*. Suppose node $u \in V_I$ is a single degree node with the only neighbor v_0 that is also infected. Since any diffusion process starting from u must pass v_0, we can then use the distribution of paths from v_0 to infer the distribution of paths from u. However, the converse is not true — a diffusion path from v_0 may not pass u, and even if it passes u, this may not occur as the first infection. Therefore, certain mapping is needed to connect the two diffusion processes. The following theorem formulates this intuition.

Theorem 2. *Let u be a single-degree node in the graph G with the only neighbor node v_0. If a path $z \in \mathcal{Z}_T$ starting from v_0 contains u*

$$z = \{v_0, s_1, s_2, \cdots, s_{K-1}, u, s_{K+1}, \cdots, s_T\},$$

*define z's **matching path** from u as*

$$f_u(z) = \{u, v_0, s_1, \cdots, s_{K-1}, s_{K+1}, \cdots, s_T\}. \qquad (10)$$

In this case, the likelihood ratio between z and $f_u(z)$ is

$$\begin{aligned}\frac{p\left(f_u(z)|u\right)}{p\left(z|v_0\right)} &= \frac{1}{\mathbb{P}\left(u|v_0, s_1\cdots s_{K-1}\right)}\\ &\quad\times\frac{1}{\prod_{k=1}^{K-1}\left(1-\mathbb{P}\left(s_k|v_0, s_1\cdots s_{k-1}\right)\right)} \qquad (11)\end{aligned}$$

If the path z from v_0 that does not contain u, we define the ratio $p\left(f_u(z)|u\right)/p\left(z|v_0\right)$ to be 0.

Notice that all terms on the right-hand side of (11) are available when we sample a path from the diffusion process starting at v_0, thus given a sampled path z, computing the likelihood ratio only introduces negligible computational cost. Intuitively, according to Theorem 2, when the MC samples of v_0 are available, they can be used to compute the p-value for node u based on a similar idea to importance sampling (L'Ecuyer & Owen, 2009). However, the regular importance sampling cannot be directly applied because the likelihood ratio is only available between z and $f_u(z)$ under the mapping of f_u. Therefore, we need a generalized version of the importance sampling. We name this procedure the *surjective importance sampling* and give its property in the following theorem. We believe that this theorem could be of general interest beyond our context.

Theorem 3 (Surjective Importance Sampling)**.** *Suppose p_1 and p_2 are two probability mass functions for discrete random vector Z defined on $\mathcal{C}_1$ and $\mathcal{C}_2$. Let $\mathbb{E}_1$ and $\mathbb{E}_2$ denote the expectation with respect to p_1 and p_2, respectively. Given surjection $\phi : \mathcal{C}_1' \to \mathcal{C}_2$, defined on a subset $\mathcal{C}_1' \subset \mathcal{C}_1$, we define the inverse mapping by $\phi^{-1}(\tilde{z}) = \{z \in \mathcal{C}_1' :$*

$\phi(z) = \tilde{z}\}$ for any $\tilde{z} \in \mathcal{C}_2$. For a given bounded real function of interest, g, define

$$\eta = \mathbb{E}_2[g(Z)] \text{ and } \hat{\eta} = \frac{1}{m}\sum_{i=1}^{m} \frac{g(\phi(Z_i))}{|\phi^{-1}(\phi(Z_i))|}\frac{p_2(\phi(Z_i))}{p_1(Z_i)}$$

where $Z_1, Z_2, \cdots, Z_m$ is a size-m i.i.d. sample from distribution p_1, and if $Z_i \notin \mathcal{C}_1'$, we define $p_2(\phi(Z_i)) = 0$. We have

$$\lim_{m\to\infty} \hat{\eta} = \eta \qquad \text{a.s.}$$

Notice that the standard importance sampling is a special case of Theorem 3 when ϕ is the identity mapping. Theorem 2 and 3 toghether would serve as a cornerstone for our use of the MC samples from v_0 to make inference of u.

Corollary 1. *For a single degree node u and its neighbor v_0, let $z_i, i = 1, \cdots, m$ be the m i.i.d. paths generated from the diffusion process with source v_0. For any bounded function g, we have*

$$\lim_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} g\left(f_u(z_i)\right) \frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)} \frac{1}{T} = \mathbb{E}_u[g(Z)] \qquad \text{a.s.}$$

in which $f_u(z_i)$ and the likelihood ratio is given by Theorem 2.

Based on Corollary 1, when $g(z) = \ell(y, z)$ or $g(z) = \mathbb{I}(T_u(\zeta(z)) \geq T_u(y))$, $\mathbb{E}[g]$ corresponds to the test statistic $T_u(y)$ or the p-value $\psi_u(y)$. Consequently, the MC sampling for u can be avoided. Instead, to find the p-value for u, Equation (7) in Algorithm 1 can be replaced by $\hat{T}_u\left(\zeta\left(f_u\left(z_j\right)\right)\right)$ equalling the following

$$\frac{1}{m}\sum_{i=m+1}^{2m} \ell\left(\zeta\left(f_u\left(z_j\right)\right), f_u(z_i)\right) \frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)} \frac{1}{T}$$

and Equation (8) can be replaced by $\hat{\psi}_u(y)$ equalling the following

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left(\hat{T}_u\left(\zeta\left(f_u(z_j)\right)\right) \geq \hat{T}_u(y)\right) \frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)} \frac{1}{T},$$

where $z_j, j = 1, \cdots, 2m$ are the MC samples generated from v_0. The same operation can be used for $\hat{T}_u\left(y\right)$). The computational strategy for canonical discrepancy functions can also be extended in this setting (see Appendix C).

4.2 Permuted Sampling for Isomorphic Nodes

When the network structure is in some sense "symmetric" for two nodes, the inference properties of the MC samples from one node can be viewed as stochastically equivalent to the MC samples from the other node after the symmetric reflection. We call such a property *isomorphism*. Denote the node u's kth order neighborhood– the set of all nodes (exactly) k hops away from u– by $N_k(u)$. The following definition for isomorphism rigorously formulates the aforementioned idea.

Definition 4. *Any two nodes u, v in a network are **first-order isomorphic** if there exists a permutation $\pi : \{1, 2, \cdots, n\} \to \{1, 2, \cdots, n\}$, such that: (1) $\pi(u) = v$; (2) $\pi(i) = i$, if $i \notin \{u, v\} \cup N_1(u) \cup N_1(v)$; (3) $A = A_{\tilde{\pi},\tilde{\pi}}$, where $A_{\tilde{\pi},\tilde{\pi}}$ is the resulting matrix by applying permutation π on the rows and columns of A simultaneously.*

For illustration, consider a simplified case of the isomorphism where u and v have exactly the same connections. In this case, π only swaps u and v and remains the identity mapping for all other nodes. For this pair of u, v, the diffusion process properties would be the same if we swap the positions of u and v. Definition 4 is more general than the above simplified case as it allows permutation to the first-order neighbors. Under this definition of isomorphism, the following theorem shows that we can use the MC samples from one node to make inference of its isomorphic nodes after applying the permutation.

Theorem 4. *If u and v are first-order isomorphic under the permutation π. If $Z = \{u, v_1, v_2, \cdots, v_{T-1}\}$ is a random diffusion path from source u. Define the permuted path*

$$Z_\pi = \{\pi(u), \pi(v_1), \cdots, \pi(v_{T-1})\}.$$

Then Z_π has the same distribution as a random diffusion path from source v.

To use the MC samples of one node to its isomorphic nodes according to Theorem 4, we need an efficient algorithm to identify all isomorphic pairs and the corresponding permutations. Directly checking Definition 4 is costly. To speed up the computation, we identify *necessary* conditions for isomorphism in Proposition 2.

Proposition 2. *If u and v are first-order isomorphic, we must have $d_u = d_v$ and $D_1(u) = D_1(v)$ where d_u and d_v are the degrees of u and v, $D_1(u)$ and $D_2(v)$ are the degree sequence (sorted in ascending order) of $N_1(u)$ and $N_1(v)$. Furthermore, u and v have the same second-order neighbor sets. That is, $N_2(u) = N_2(v)$.*

Based on Proposition 2 we can efficiently identify isomorphism using pre-screening steps. This turns out to significantly speed up our computation. Details of the algorithm are described by Algorithm 2 in Appendix B. With the isomorphic relations available, we can partition the nodes into isomorphic groups. Then MC sampling is only needed for one node in each group, and the MC samples can be shared within the group according to Theorem 4. Specifically, suppose $Z_1, \cdots Z_{2m}$ are sampled from the diffusion process

from u. If u and v are isomorphic with permutation π, we can use $(Z_1)_\pi, (Z_2)_\pi, \cdots, (Z_{2m})_\pi$ as the MC samples of v in Algorithm 1.

Remark 5. Definition 4 can be extended to higher-order neighborhoods, identifying more isomorphic pairs. However, the complexity of identifying such pairs increases exponentially with the order of neighbors, which may overwhelm the saved time on the MC side. The first-order isomorphism turns out to give the most desirable tradeoff in terms of computational efficiency.

5 Experimental Studies

In this section, we evaluate our proposed methods on well-studied random network models. We generate networks from three random network models: random 4-regular trees, the preferential attachment model (Barabási & Albert, 1999) and the small-world (S-W) network model (Watts & Strogatz, 1998). In network science, the preferential attachment model is usually used to model the scale-free property of networks that is conjectured by many as ubiquity in real-world networks (Barabási, 2013). The small-world property is believed to be prevalent in social networks (Watts & Strogatz, 1998). The network size is $N = 1365$ (the size of regular tree with degree 4 and depth 6). The networks are sparse, with an average degree below 4. The Monte Carlo size m is 10000. Source nodes are randomly sampled, and the reported results are an averaged across 100 replications. All source code of this paper can be found in hyperlink https://github.com/lab-sigma/Diffusion-Source-Identification.

5.1 Confidence validity evaluation

First, we set the infection size $T = 150$. We start with evaluating the performance of the single-point source estimation accuracy from the rumor center and distance center of (Shah & Zaman, 2011; Khim & Loh, 2016; Bubeck et al., 2017; Yu et al., 2018), as well as estimator using our proposed framework with discrepancy functions ℓ_{se} and ADiT. The result is shown in the Table 1. Though the two estimators based on our framework are better, the overall message from the table is not promising. All of the methods, including ours, give poor accuracy that is too low to be useful in applications. Such a negative result convincingly shows that the DSI problem is generally too difficult for the single-point estimation strategy to work, and exploring the alternative confidence set inference is necessary.

Table 2 shows the coverage rate of the confidence sets, with the squared Euclidean distance and the ADiT as the discrepancy functions. Notably, the proposed confidence set procedure delivers the desired coverage (up to the simulation error). Meanwhile, the size of the confidence set varies substantially depending on the network structure. For regular trees and scale-free networks, the ADiT works much better than the Euclidean distance, indicating that the diffusion order is informative in this type of network structure. For the small-world networks, the two are very similar. This may indicate that for well-connected networks, the diffusion order is less informative. In general, we believe the adaptivity of the ADiT- based confidence set is always preferable.

Table 1: The correct rate of single-point estimation methods across 200 replications.

	REG. TREE	PREF. ATT.	S-W
RUMOR CENTER	0	0	0.004
DIST. CENTER	0	0	0
EUCLIDEAN (OURS)	0	0	0.099
ADIT (OURS)	0	0	0.128

Table 2: The average coverage rate of the confidence sets across 200 replications. The standard error for the coverage rate is about 3% and 4% for 90% and 80% confidence sets, respectively.

	REG. TREE	PREF. ATT.	S-W
EUCLIDEAN-90%	90.4%	90.8%	90.2%
SIZE	74.9	81.2	14.3
ADIT-90%	86.2%	90.7%	91.5%
SIZE	56.9	64.8	16.2
EUCLIDEAN-80%	84%	82%	81.1%
SIZE	50.0	57.5	10.2
ADIT-80%	77.4%	82.7%	79.9%
SIZE	47.5	51.0	9.2

To obtain a comprehensive view of the tradeoff between the set size and confidence level, we show the relationship between the confidence set's average size and the confidence level in Figure 2. The relation is slightly sup-linear. In connection with the single-point estimation results, notice that for small-world networks, the confidence set with a confidence level 20% has average size of around 1. In contrast, the regular tree and preferential attachment network are more difficult, and to guarantee at 10%, the average size of the confidence set is already about 5. These observations verify the results in Table 1 and support our argument that, in general, inferring the source by a single-point estimator is hopeless. Figure 3 shows the variation of the size with respect to T. It can be seen that the size, within the current range, follows a roughly linear trend with T. Again, though the ADiT is slightly worse than the Euclidean loss in small-world networks, the difference is negligible. In the other two settings, the improvement of ADiT is significant.

5.2 Computational Improvement by the pooled MC

Finally, we also evaluate the timing improvements achieved by the pooled MC strategies. The power of the pooled MC strategies depends on network structures, as expected. The timing comparison for the pooled MC strategies is included in Table 3. The timing included is only the sequential ver-

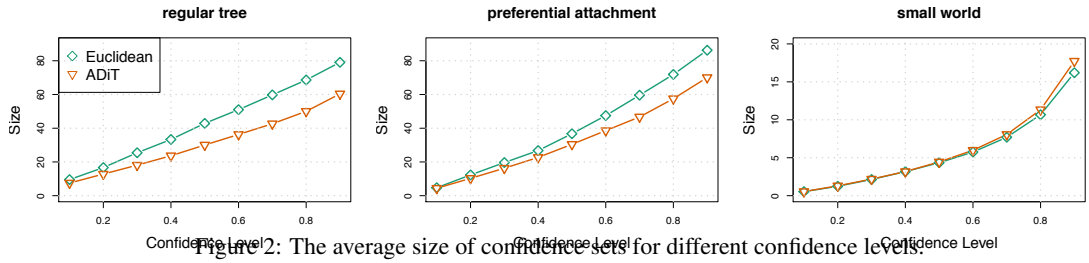

Figure 2: The average size of confidence sets for different confidence levels.

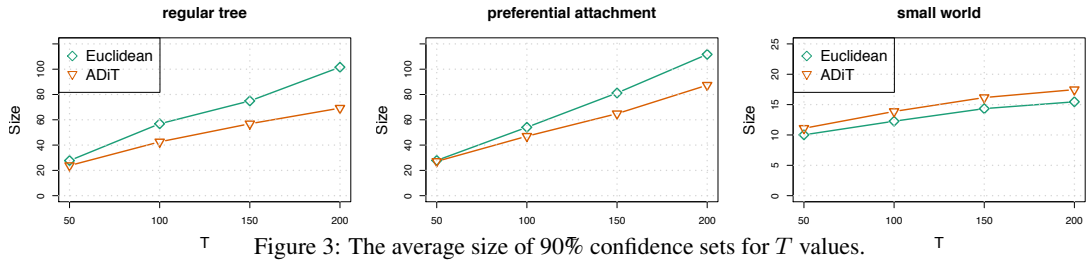

Figure 3: The average size of 90% confidence sets for T values.

Table 3: The timing comparison of the sequential running time for the proposed pooled MC strategies (in sec.).

	REG. TREE	PREF. ATT.	S-W
VANILLA MC	2606	3129	3209
IMPORT. SAMPL.	1679	1730	3253
ISOMORPHISM	1657	1988	3138
BOTH	1219	1360	3114

sion of our method for a fair comparison with the rumor center. As can be seen, with both of the pooled MC strategies used, we can reduce the timing by about 60% for tree structure and the preferential attachment networks, but the effects on small-world networks are negligible.

Meanwhile, notice that our inference procedure can be parallelized. We give a parallel algorithm in the Appendix section (see Algorithm 3 in Appendix D). It needs MC sampling for only one node in each group, and the calculations for other nodes can be done using pooled MC methods. Table 4 includes the timing results of the parallel version implementation based on 20 cores in the same settings as Table 3. With 20 cores, the time needed for a confidence set construction is around 1 minute for cases when the pooled MC methods are effective. For reference, the average timing for finding the rumor center is about 2 seconds. However, with the extra computational cost, our method provides *confidence sets at all specified levels* within one run, with guaranteed accuracy for any network structures. We believe it is generally a wise tradeoff.

Table 4: Comparison of the parallel running time for the proposed pooled MC strategies (in sec.) on 20 cores.

	REG. TREE	PREF. ATT.	S-W
VANILLA MC	150.8	176.0	184.9
IMPORT. SAMPL.	116.7	96.1	185.9
ISOMORPHISM	111.0	130.3	184.3
BOTH	60.4	76.5	183.4

To obtain a better sense of its practical effectiveness, we also evaluate the timing improvement brought by the pooled MC on real-world network structures. In particular, we take 381 network data studied in (Ghasemian et al., 2020) from 6 domains (biological, economic, informational, social, technological and transportation networks). The pooled MC can give more than 40% computational improvement on economic and social networks, and deliver 10% to 20% improvement on biological and informational networks. Details can be found in Appendix E.

6 Summary

We have introduced a statistical inference framework for diffusion source identification on networks. Compared with previous methods, our framework is more general and renders salient insights about the problem. More importantly, within this framework, we can construct the confidence set for the source node in a more natural and principled way such that the success rate can be guaranteed on any network structure. To our knowledge, our method is the first DSI method with theoretical guarantees for general network structures. We also propose efficient computational strategies that are potentially useful in other problems as well.

Acknowledgements

The work is supported by the Quantitative Collaborative Award from the University of Virginia. T. Li is supported by the NSF grant DMS-2015298. H. Xu is supported by a GIDI award from the UVA Global Infectious Diseases Institute. We thank the anonymous reviewers for helpful feedback.

References

Anderson, R. M. and May, R. M. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

Barabási, A.-L. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Besag, J. and Clifford, P. Generalized monte carlo significance tests. *Biometrika*, 76(4):633–642, 1989.

Bringmann, K. and Panagiotou, K. Efficient sampling methods for discrete distributions. *Algorithmica*, 79(2): 484–508, 2017.

Bubeck, S., Devroye, L., and Lugosi, G. Finding adam in random growing trees. *Random Structures & Algorithms*, 50(2):158–172, 2017.

Crane, H. and Xu, M. Inference on the history of a randomly growing tree. *arXiv preprint arXiv:2005.08794*, 2020.

Dong, W., Zhang, W., and Tan, C. W. Rooting out the rumor culprit from suspects. In *2013 IEEE International Symposium on Information Theory*, pp. 2671–2675. IEEE, 2013.

Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M., and Clauset, A. Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117(38):23393–23400, 2020.

Halperin, A. and Almogy, G. System and method of virus containment in computer networks, December 19 2002. US Patent App. 10/058,809.

Hu, T., Guan, W. W., Zhu, X., Shao, Y., Liu, L., Du, J., Liu, H., Zhou, H., Wang, J., She, B., et al. Building an open resources repository for covid-19 research. *Data and Information Management*, 4(3):130–147, 2020.

Jockel, K.-H. Finite sample properties and asymptotic efficiency of monte carlo tests. *The annals of Statistics*, pp. 336–347, 1986.

Kazemitabar, S. J. and Amini, A. A. Approximate identification of the optimal epidemic source in complex networks. In *International Conference on Network Science*, pp. 107–125. Springer, 2020.

Khim, J. and Loh, P.-L. Confidence sets for the source of a diffusion in regular trees. *IEEE Transactions on Network Science and Engineering*, 4(1):27–40, 2016.

Lab, C. D. Baidu Mobility Data, 2020. URL https://doi.org/10.7910/DVN/FAEZIO.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

L'Ecuyer, P. and Owen, A. B. *Monte Carlo and Quasi-Monte Carlo Methods 2008*. Springer, 2009.

Lehmann, E. L. and Romano, J. P. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

Newman, M. E., Forrest, S., and Balthrop, J. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101, 2002.

Nguyen, H. T., Ghosh, P., Mayo, M. L., and Dinh, T. N. Multiple infection sources identification with provable guarantees. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1663–1672, 2016.

Shah, D. and Zaman, T. Rumors in a network: Who's the culprit? *IEEE Transactions on information theory*, 57 (8):5163–5181, 2011.

Shah, D. and Zaman, T. Finding rumor sources on random graphs. 2012.

Shapiro, M. and Delgado-Eckert, E. Finding the probability of infection in an sir network is np-hard. *Mathematical biosciences*, 240(2):77–84, 2012.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vosoughi, S., Roy, D., and Aral, S. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

Xu, Y. and Ren, J. Propagation effect of a virus outbreak on a network with limited anti-virus ability. *PloS one*, 11 (10):e0164415, 2016.

Yu, P.-D., Tan, C. W., and Fu, H.-L. Rumor source detection in finite graphs with boundary effects by message-passing algorithms. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 175–192. Springer, 2018.