Diffusion Source Identification on Networks with Statistical Confidence

---

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Quinlan Dawkins

December 2021

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Quinlan Dawkins

This Thesis has been read and approved by the examing committee:

Advisor: Haifeng Xu

Advisor:

Committee Member: Tianxi Li

Committee Member: Jundong Li

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

December 2021

Abstract

Identifying the source of spread through a network, termed diffusion source identification, is a problem of fundamental importance in a broad class of applications such as 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 real-world network topologies. This work introduces a statistical framework for diffusion source identification for a broad class of diffusion processes on general network topologies. It also develops a confidence set inference approach inspired by statistical hypothesis testing. Our method efficiently produces a small subset of nodes, which provably covers the source node with any prespecified confidence level. Multiple Monte Carlo strategies are presented 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. The proposed method is evaluated by extensive synthetic experiments based on well-known random network models and real-world networks from several domains. The method is also demonstrated in a real-world diffusion example concerning the spread of COVID-19 between cities.

Acknowledgements

First, I would like to thank my advisor Haifeng Xu from the Department of Computer Science in the School of Engineering at the University of Virginia, as well as Professor Tianxi Li from the Department of Statistics in the School of Arts and Sciences at the University of Virginia. As the other contributors to this research, they were instrumental throughout the research process and offered critical guidance during the thesis writing process. In particular, Haifeng was a very helpful resource throughout time as a Master's Student.

Additionally I would like to thank my parents for the support and encouragement they offered me when writing the thesis, as well as throughout my time at the University of Virginia. Finally, I am grateful my younger brother and any friends who listened to me any time I went on about this research or my progress in writing this thesis.

This research was financially supported in part by the NSF grant DMS-2015298 and the Quantitative Collaborative grant from UVa Arts & Sciences via Haifeng Xu and Tianxi Li.

Contents

Chapter 1

Introduction

The spread of misinformation through a social network or a virus in a computer network has seen increasing relevance in recent years. The spreading of computer viruses through emails and computer networks have significant privacy and security implications on today's growing cyberspace [1, 2, 3]. Similarly, the COVID-19 epidemic presented another example of the importance of understanding the spread of infection through a network and how to take fast and effective counter-measures. Spreading via a *diffusion process* begins with a few source individuals and spreads quickly throughout the network. Reducing the loss from such an event relies on quickly identifying the sources so that counter-measures can be taken in a timely fashion, which is referred to as the *diffusion source identification* (DSI) problem.

Though there are examples of early practices 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 [4] which focused on the setting of infinite regular trees. Despite the significant interest and progress on this problem in recent years [5, 6, 7, 8, 9, 10], 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 [11]. Therefore, intuitive approximations have been used for general networks [12, 13]. 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 [4, 6, 9]. For general graphs and diffusion processes, this is an inherent difficulty with *single-point* estimation methods that we observe later.

To guarantee higher success probability, a typical approach, as in both machine learning theory [14] and data-driven applied models [15], 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 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 [8, 7, 10]. Though they enjoy good theoretical properties, the methods are applicable only on infinite trees.

This research 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 [4, 12] 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 process in early 2020. Figure 1.1 shows a travel mobility network between 49 major cities in China, constructed from the two-week travel volume [16, 17] 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?

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 5.2). Nevertheless, both of our 80% and 90% confidence sets cover Wuhan correctly, giving recommendations of 6 nodes

[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.

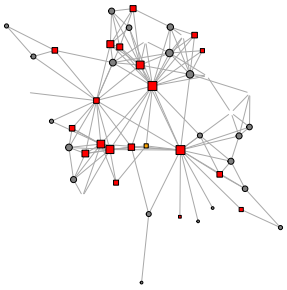

Figure 1.1: The mobility network and the COVID-19 infection status of major Chinese cities on Jan 24, 2020. Colored square nodes are cities with at least five confirmed cases.

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 and includes a more general definition of diffusion processes, 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 this research 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.

1.1 Literature

A primary motivational setting for the study of diffusion processes is in epidemiological spreading. The study of epidemic models often take a macroscopic view which focuses on the summary status of a population during the spreading of a disease [18], whereas diffusion processes on networks consider microscopic interactions between network members and describe the *geographic* properties of a population resulting from the diffusion process. The study of such models has been shown to be useful in many areas beyond epidemic models, such as information spreading in social networks which can have implications in politics, economics, and social engineering [19, 20]. Other examples of relevant applications of diffusion models on networks include simulating a failure or virus spreading in computer networks or tracking the movement of files in peer-to-peer content distribution networks [21]. Here we discuss some of the research on diffusion processes as context for the models studied in this research.

1.1.1 Cascading Models

Cascading, or epidemic models, are processes where the diffusion "cascades" through the network. Notable, additional infections increase the susceptibility of other nodes in the network, mimicking the behavior of an epidemic in a population, or the spreading of information in a social network [21]. Fundamental models in epidemic spreading, the "Susceptible-Infected" (SI) model, "Susceptible-Infected-Recovered" (SIR) model, and "Susceptible-Infected-Susceptible" (SIS) model were introduced by Kermack and McKendrick [22] by modelling the spread of an epidemic as the transition of individuals between states based on a set of differential equations. These models have served as the base for a number of epidemic spreading models such as discrete time and continuous time versions of the SEIR and SEIS where "E" denotes an exposed state [23]. The SWIR model includes a weakened state [24].

A close relative of the SIR model is the *Independent Cascades* (IC) model that is intended to describe the spread of influence through a social network [25]. The problem of influence maximization, which could be thought of as an inverse problem to DSI, requires identifying the set of source nodes that will maximize the expected influence through the network at the end of the

diffusion process. Variants of the IC model include community permeability and community embeddedness in an attempt to better model real world spreading in social networks [26]. This model has also been used in inferring back the network structure given diffusion information [21, 27], a similar problem to that of diffusion source identification.

Another popular model of study in computer science, particularly with the influence maximization problem, is the Linear Threshold model [28]. Threshold models impose a threshold on network nodes such that participation or infection only occurs once the surrounding influence has surpassed an individual's threshold [29]. There is a generalized version of the model that allows for influence to be determined by a general monotone function rather than a simple linear relationship [28].

In line with past studies on the DSI problem, in this research we focus on a variant of the SI model with additional experimental results for the Independent Cascades and Linear Threshold models.

1.1.2 The SI Model and DSI

The SI model with fixed infection size is the primary model studied in research on the diffusion source identification problem. Shah and Zaman [4] proposed a maximum likelihood *rumor center* estimator that can be located by an efficient message-passing algorithm with linear time complexity, but is restricted to infinite regular trees and the SI model with an exponential spreading time distribution. A result for diffusion processes with general spreading time distributions and general random trees was later presented in [5]. The rumor center result was extended by [9] to finite trees with an optimal result for regular trees and a near optimal result for general trees. Dong et al. [6], also focusing on the SI model, constructed a *maximum a posteriori* (MAP) estimator and proposed a concept called a local rumor center as generalization of rumor centrality to incorporate a priori knowledge into the DSI problem. Nguyen et al. [12] gave a result for approximate inference on multiple sources with results for the previously studied SI model as well as two other popular diffusion processes, the IC and LT models. In [30], source detection with partial observation of the activation time of the rumors in the diffusion is studied.

Rumor source obfuscation is studied in [31] for the setting of diffusions in anonymous messaging platforms. Their results shows that a certain family of messaging protocols denoted as *adaptive diffusions* are able to achieve perfect

obfuscation where a "snapshot adversary" cannot to better than random guessing in the best case. [32] studied adaptive diffusions on infinite regular trees in alternate settings where the adversary can take multiple snapshots and also with local information spreading, showing that in certain settings constant probability of source detection is achievable.

Confidence set methods have been studied in the context of this problem before. Bubeck et al. [8] showed that for uniform attachment and preferential attachment trees there exists an algorithm for finding a set of K nodes with a $1 - \epsilon$ guarantee of including the source for any ϵ. Notably the size of K is independent of the tree size and diffusion detection is more difficult in the preferential attachment case compared with the regular tree case. The confidence sets in [7] apply to a regular-tree structure and provides an upper bound on the number of elements needed to guarantee the inclusion of the source in the confidence set. Finally, [10] studies confidence sets for linear preferential attachment and uniform attachment models and provides upper bounds on the expected confidence set size.

Chapter 2

Preliminaries

We now define a general version of the *Diffusion Source Identification* (DSI) problem, introduced in the seminal work of Shah and Zaman [4]. Consider a network G with node set $V = \{1, \cdots, n\}$ and edge set E. We write $(u, v) \in E$ if there is an edge from node u to v. Additionally we let each edge $(u, v) \in E$ have non-negative weight $w_{uv} \in \mathbb{R}_+$. The network can be equivalently represented by its $n \times n$ *adjacency matrix* A, where $A_{uv} = w_{uv}$. Note that if A is symmetric, then G is undirected, and if $\forall u, v \in V$ the edge weight is $w_{uv} \in \{0, 1\}$, then G is unweighted. Note that G is assumed to be connected in this research as independent subgraphs can be analyzed independently under the DSI problem.

The diffusion model of choice for previous works is the "Susceptible-Infected" (SI) model [33, 4], however in this research we will be considering a more general definition of a diffusion process on a single *source node* $s^* \in V$. Let a diffusion process, $\sigma : \mathbb{R}^{n \times n} \times [n] \to \{0, 1\}^n$, be a probabilistic function on a graph $G = (V, E)$ and source s^* that generates a connected subset of V. In this research we are considering two-state diffusion models, however the statistical results shown later on are generalizable to models with a greater number of node statuses, or possibly to diffusion process without the requirement of connected infections. Note that in this research because confidence construction relies on simulating the diffusion process, we only consider efficiently simulatable diffusion processes.

Now, denote the *sample space*, $\mathcal{Y}_\sigma \subseteq \{0, 1\}^n$, as the subsets of V that are supported under diffusion process σ for *any* choice of source $s \in V$. With this, the DSI problem is formally defined as follows:

Definition 1 (Diffusion Source Identification)**.** *Given diffusion process* σ

and one sample $y \in \mathcal{Y}_\sigma$, identify the source node s^* of the diffusion process that generates y.

In the DSI problem, a node s cannot be the source for data y if $P(\sigma(G, s) = y) = 0$ for diffusion process σ. To capture the set of candidate source nodes for the DSI problem we denote the set $C_\sigma(y) = \{s : s \in y, P(\sigma(G, s) = y) > 0\}$ as the set of sources that are capable of generating the sample y under process σ.

Challenges

There are two notable challenges intrinsic to the DSI problem. First, for a given source s^*, there may be more than one execution path and choice of source for $\sigma(G, \cdot)$ that produces the same sample y. In fact, the problem of source identification is rooted in the fact that this information is lost by the time y is observed. Consequently we define the *diffusion path space* $\mathcal{Z}_\sigma$ and many-to-one mapping $\zeta : \mathcal{Z}_\sigma \to \mathcal{Y}_\sigma$ to capture the execution of σ.

$$\mathcal{Z}_\sigma = \{\boldsymbol{v} = \{(s^* = v_0, t_0 = 0), (v_1, t_1), \cdots, (v_T, t_T)\} : v_t \in V, \\ \text{and } (v_i, v_{i'}) \in E \text{ for some } i \neq i' \text{ with } t_i < t_{i'} \}$$

The second notable challenge is the inherent limited availability of data. In contrast to a learning problem, there is only access to a single data observation. For example, the Markovian nature of the SI model makes subsequent observations of a network state unhelpful in solving the DSI problem. If you combine this with the uncertainty in the generation of y, it makes achieving arbitrary accuracy in the DSI problem impossible for point estimate methods.

2.1 Diffusion processes

In this research we will consider four different diffusion processes: unweighted and undirected SI model with fixed infection size, weighted and directed SI model with fixed infection size, Independent Cascades (IC) model, and Linear Threshold (LT) model. As explained in section 1.1, the unweighted SI model is the most widely studied diffusion process in the source identification problem while the weighted version is a natural extension that better reflects real world network structures. The IC and LT models are popular choices in

the influence maximization problem.

2.1.1 Unweighted and Undirected SI Model

As the most widely studied diffusion process in the DSI problem, the unweighted SI model is the primary model of study in this research. In this model, there are a total of T infections starting with source node s^*, giving a total of $T+1$ infected nodes. Let $A_t \subseteq V$ be the set of active nodes at time step t. The next propagation edge is then chosen uniformly at random from the edge set $\{(u,v) : u \in A_t, v \in V \setminus A_t\}$. The active set for time $t+1$ is then $A_{t+1} = A_t \cup \{v_t\}$ for chosen edge (u_t, v_t). This proceeds with $A_0 = \{s^*\}$ and $\sigma_{SI}^T(G, s^*) = A_T$ is the resulting sample infected set. Note that this process is equivalent to a continuous time model with infections propagating across edges with spreading time sampled from an exponential distribution. The process is summarized in algorithm 1.

Algorithm 1 Unweighted and undirected SI model for fixed number of infections

1: **Input:** Infection count T, source s^*
2: $A_0 = \{s^*\}$
3: **for** $t = 1$ **to** T **do**
4: Select $(u_t, v_t) \in \{(u,v) : u \in A_t, v \in V \setminus A_t\}$ uniformly at random
5: $A_{t+1} = A_t \cup \{v_t\}$
6: **end for**
7: **return** A_T

For this model, the diffusion path space $\mathcal{Z}_T$ is the sequence of infection propagation edges.

$$\mathcal{Z}_T = \{\boldsymbol{v} = \{(s^* = v_0, t_0 = 0), (v_1, 1), \cdots, (v_T, T)\} : v_t \in V, v_{t_1} \neq v_{t_2} \text{ if } t_1 \neq t_2, \text{ and } (v_t, v_{t'}) \in E \text{ for some } t < t'\}$$

The diffusion sample space is the set of connected subgraphs of G of size $T+1$ and the set of candidate source nodes is $C_{SI}(y) = y$.

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

2.1.2 Weighted and Directed SI Model

For weighted graphs the probability of selecting an edge (u, v) is proportional to the edge weight w_{uv}. If $E_t = \{(u, v) : u \in A_t, v \in V \setminus A_t\}$ is the set of propagating edges for A_t, then the probability of selecting $(u, v) \in E_t$ is $P_{E_t}((u, v)) = w_{uv} / \sum_{(u',v') \in E_t} w_{u'v'}$. No changes are needed in the algorithm description to support directed graphs.

Algorithm 2 Weighted and directed SI model for fixed number of infections

1: **Input:** Infection count T, source s^*
2: $A_0 = \{s^*\}$
3: **for** $t = 1$ **to** T **do**
4: Select $(u_t, v_t) \in E_t = \{(u, v) : u \in A_t, v \in V \setminus A_t\}$ with probability

$$P_{E_t}((u, v)) = w_{uv} / \sum_{(u',v') \in E_t} w_{u'v'}$$

5: $A_{t+1} = A_t \cup \{v_t\}$
6: **end for**
7: **return** A_T

The diffusion path space $\mathcal{Z}_T$ and diffusion sample space $\mathcal{Y}_T$ are the same as in the unweighted SI model. If G is directed, however, it is possible to infect a node v such that there is no path from v back to the source s^*. Define G_y as the subgraph of G containing only the nodes in y. The set of candidate source nodes is then

$$C_{SI}(y) = \{v : \forall u \in y, \exists \text{ a path from } v \text{ to } u \text{ in } G_y\}$$

2.1.3 Independent Cascades Model

We consider the most basic version of the IC model. In this model, diffusion is based on a cascade of information throughout a network. The process begins

with a set of active nodes that remain active for a single time step during which it can affect all of their inactive neighbors. Because we are considering single source diffusion processes, the initial active set is $A_0 = \{s^*\}$. Infections propagate along edges according to their weights. This requires $w_{uv} \in [0, 1]$ where the probability of node activation is w_{uv}.

Algorithm 3 Independent Cascades Model

Input: Infection count T, source s^*
$A_0 = \{s^*\}$, $I = \{\}$
while A_t is not empty **do**
  $A_{t+1} = \{\}$
  for each $(u, v) \in \{(u', v') : u' \in A_t, v' \in V \setminus (A_t \cup I)\}$ **do**
    Add v to A_{t+1} with probability w_{uv}
  end for
  $I = I \cup A_t$, $t = t + 1$
end while
return I

Note that this model appears to include three states: active, inactive, and uninfected, however the state assignment output by Algorithm 3 does not include any active nodes. The diffusion path space encodes the sequence of active nodes during the diffusion process, while the diffusion sample space is the set of connected subgraphs of G. The set of candidate nodes $C_{IC}(y)$ is the same as in the directed SI model.

$$\mathcal{Z}_{IC} = \{\boldsymbol{v} = \{(s^* = v_0, t_0 = 0), (v_1, t_1), \cdots, (v_T, t_T)\} : v_t \in V,$$
$$\text{if } t_1 \neq t_2, \text{ then } v_1 \neq v_2 \text{ and } (v_i, v_{i'}) \in E \text{ for some } i \neq i' \text{ with } t_i < t_{i'} \}$$

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

2.1.4 Linear Threshold Model

The Linear Threshold model is designed to mimic information spreading where an individual is only influenced once a certain proportion of its neighbors are influenced. Edge weights are restricted such that for each $v \in V$,

$\sum_{u \in V} w_{uv} \leq 1$. Once a node becomes active it remains active, and a node becomes active once the total influence of its active neighbors surpass a hidden threshold $\theta_t \in [0, 1]$ that is chosen uniformly at random.

Algorithm 4 Linear Threshold Model

Input: Infection count T, source s^*
$A_0 = \{s^*\}$
Select $\theta_v \in [0, 1]$ for each $v \in V$
while $A_t \neq A_{t-1}$ **do**
  $A_{t+1} = A_t$
  for each $v \in V \setminus A_t$ **do**
    Add v to A_{t+1} if $\theta_v \leq \sum_{u \in A_t} w_{uv}$
  end for
  $t = t + 1$
end while
return A_t

The diffusion path space, $\mathcal{Z}_{LT}$, encodes the sequence of newly activated nodes and thus takes the same form as $\mathcal{Z}_{IC}$. The diffusion sample space $\mathcal{Y}_{LT}$ and candidate set $C_{LT}(y)$ candidate are identical to $\mathcal{Y}_{IC}$ and $C_{IC}(y)$ for the independent cascades model as well.

Chapter 3

A General Statistical Framework

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. First, treat the diffusion process σ and network G as fixed with s^* as the only model parameter. The probability of generating data $y \in \mathcal{Y}_\sigma$ can then be represented by $\mathbb{P}_{s^*}(Y = y) = p(y|s^*)$. where random variable $Y = \sigma(G, s^*)$ denotes the diffusion sample resulting from σ with source s^*. 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* $\ell : \mathcal{Y}_\sigma \times \mathcal{Z}_\sigma \to [0, \infty)$, we want to find an estimator of s^* based on the following optimization problem:

$$\text{minimize}_s \quad \mathbb{E}_s \ell(y, Z) \tag{3.1}$$

in which $Z \in \mathcal{Z}_\sigma$ is the diffusion path random variable given diffusion process σ with source parameter s and where $\mathbb{E}_s$ denotes the expectation over Z. In other words, we look to select the s such 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 $\mathcal{Y}_\sigma \times \mathcal{Z}_\sigma$, not on $\mathcal{Y}_\sigma \times \mathcal{Y}_\sigma$. 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 includes a few previous methods as special cases. For both cases the diffusion process σ is the unweighted SI model with fixed infection size. 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, process* (3.1) *gives the rumor center of Shah and Zaman [4].*

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 [12][1].

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 and is unsuited for the Monte Carlo approach later described for confidence set construction. 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 following form*

$$\ell(y, z) = -\sum_{v: y_v = 1} \mathbb{I}(v \in z) h(t_z(v)), \tag{3.2}$$

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

The canonical form (3.2) is essentially a negative similarity function. It incorporates both the infection status and the infection order of z. This form of discrepancy only differentiates between deviations of z from y based

[1]However, different from our framework, [12] used an approximation metric to this discrepancy for DSI.

on infection order in z. The weight function h_t then is used to encode the importance of earlier deviations against later deviations. Conceptually, this canonical family is general enough to incorporate the needed information for the diffusion processes described in Section 2.1. In addition, as shown in Section 3.4, it admits fundamental properties that make the computation very efficient. As a special case, if the infection size is fixed for diffusion process σ (e.g. SI model with fixed infection size T), it holds 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. (3.2) with $f(t_z(v)) \equiv 2$ given a fixed diffusion sample size of T.

If we intuit that infections earlier in a diffusion process are more likely to match when the true source is tested, the following natural configuration for discrepancy function arises which we call the "**Averaged Deviation - inverse Time**" (ADiT), which takes the canonical family form (3.2) with the inverse time weights:

$$h(t_z(v)) = \frac{1}{t_z(v)}. \tag{3.3}$$

In this case a stronger signal is sent for deviations early in the diffusion process while later deviations are assigned less importance.

Remark 2. The design of discrepancy functions for general diffusion processes permits more depth than what is encoded in the canonical family of discrepancy functions. Notably, the kinds of loss functions that are effective and possible can vary significantly depending on the choice of diffusion process σ. The design of the canonical class as well as $\mathcal{Z}_T$ is motivated by their effectiveness in the SI model and their computational advantages, however different diffusion processes may need the consideration of different classes of discrepancy functions and definitions for $\mathcal{Z}_\sigma$ to include information beyond just infection time.

In Table 5.4 of Section 5.2, 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 and Zaman [4], Dong et al. [6], Yu et al. [9], 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. [8], Khim and Loh [7] and Crane and Xu [10]. Bubeck et al. [8] considered identifying the first node of a growing tree but not a diffusion process. Khim and Loh [7] 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 and Loh [7] 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 and Xu [10] 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 process 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

$$\psi_s = \mathbb{P}_s(T_s(\zeta(Z)) \geq T_s(y)). \tag{3.4}$$

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 [34]. For a level $1 - \alpha$ confidence set, we compute ψ_s for all nodes s and construct the confidence set by

$$S(y) = \{s : \psi_s(y) > \alpha\}. \tag{3.5}$$

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

Theorem 1. *The confidence set constructed by* (3.5) *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 depends on the network structure and the corresponding effectiveness of the discrepancy function (the test statistic). We will use the proposed ADiT as our primary choice of discrepancy in defining our test statistic. As shown in our empirical study, it gives excellent and robust performance across various network settings in the SI model. There is an extended discussion on choosing the discrepancy function included in Chapter 5.

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 computing the probability mass function for general diffusion processes is intractable, notable the IC model [11] and SI model. To overcome this barrier, we resort to the Monte Carlo (MC) method for approximate calculation, with details in Algorithm 5. This vanilla version turns out to be computationally inefficient. However, we will introduce techniques to significantly improve its computation efficiency afterwards.

Remark 3 (**Monte Carlo setup).** Note that we have two layers of Monte Carlo evaluations. The first layer is the loss function calculation in (3.6) and (3.7), while the second layer is the p-value evaluation (3.8). The number of samples used in the loss function calculation is denoted as m_l, while the number of samples used for p-value calculation is denoted as m_p. 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 [35, 36].

Remark 4 (**Choice of the sample numbers m_l and m_p).** In theory, the computation in Algorithm 5 is exact when $m_l, m_p \to \infty$. In practice, simple guidance about the choice of m_l and m_p can be derived as follows. The critical step in Algorithm 5 is Step 7 for the p-value calculation since the MC errors from previous steps are usually in a lower order, although there is a dependence on the choice of discrepancy function ℓ and . 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_p indicators. By the central limit theorem, the MC estimate at most misses the true p-value by roughly $2\sqrt{\alpha(1-\alpha)/m_p}$. For example, if we are aiming for a 90% confidence set where $\alpha = 0.1$, setting $m_p = 10000$ would indicate that the MC at most misses the targeting confidence level by 0.006%, which is usually good enough in most applications. For the main experiments, we use $m_p = 10000$ and $m_l = 10000$ and it has been sufficient in all situations. Notice that this recommendation is more conservative than the ones used in classical statistical inference problems [35]. In our experience, it might still be acceptable to use a smaller m_p, as $m_p = 2000$ is used in Appendix A.6 where acceptable confidence levels are observed.

Remark 5 (Time complexity of the vanilla MC, and its trivial parallelization). The time complexity of a standard sequential implementation of Algorithm 5 is dependent on the complexity of running diffusion process σ. Let $C = |C_y|$ be the number of candidate source nodes in C_y and let $N = \mathbb{E}|\zeta(z)|$ be the expected size of a diffusion sample. Then the number of calls to σ is $\tilde{O}((m_l + m_p)C)$ and loss calculation has an average complexity of $\tilde{O}(m_p m_l CN)$. For the SI model this comes out to a time complexity of $\tilde{O}((m_l + m_p)T^2 + m_p m_l T^2)$ [2] (1) the first term is due to the MC sampling [37]; (2) the second term is from the statistic calculation (3.7) given the MC samples. However, our algorithm can be trivially parallelized. In particular, the loop in Step 3 can be distributed across different $s \in C_y$ without any need for communication. This leads to a parallel time complexity $\tilde{O}((m_l + m_p)T + m_p m_l T)$ in the case of the SI model. It is worthwhile to compare this time cost with the rumor center of [4] 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 5 has a better dependence on the network density captured by d but has an additional quadratic dependence on the number of samples m_l and m_p.

3.4 Fast Loss Estimation for the Canonical Family

A major computational bottleneck of Algorithm 5 is the $O(m_p m_l N)$ ($O(m_p m_l T)$ for SI) time for estimating $\mathbb{E}_s(\ell(y, Z))$ in Equation (3.7) for every j since we have to compute $\hat{\psi}$ for m_p samples, and each $\hat{\psi}$ is the average over another m_l samples. Fortunately, it turns out that, for the canonical family of discrepancy functions this step can be done in $O((m_l+m_p)T)$ time, highlighting another advantage of this particular class of discrepancies.

Instead of computing $\hat{T}_s$ in Equation (3.7) by summing over the sample $i = m_p+1, \cdots, m_p+m_l$, 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

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

$\hat{T}_s(y)$ function in Equation (3.7):

$$\begin{aligned}\hat{T}_s(y) &= -\frac{1}{m_l}\sum_{v:y_v=1}\sum_{i=m_l+1}^{m_p+m_l}\sum_{k=1}^{T} h(k)\mathbb{I}(t_{z_i}(v)=k) \\ &= -\frac{1}{m_l}\sum_{v:y_v=1}\sum_{k=1}^{T} M_{v,k}h(k) \end{aligned} \tag{3.9}$$

where $M_{v,k}$ counts the total number of samples in $z_{m_p+1}, \cdots, z_{m_p+m_l}$ 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 $m_l N$ nonzero entries in M since each sample only has N nodes. These entries can be computed in $O(m_l N)$ 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 $O(m_l T)$ time. Finally, given the previous $\hat{h}(v)$, we can compute any $\hat{T}_s(y)$ in $O(T)$ time where $y = \zeta(z_1), \cdots, \zeta(z_{m_p})$, which thus in total takes an additional $O(m_p T)$ time. This overall takes $O((m_l + m_p)T)$ time. Note an additional advantage of this approach is the independence of M from sample data y. This allows for the efficient storage of M for each choice of source j if precomputing the samples for loss calculation.

Algorithm 5 Vanilla MC for Confidence Set Construction

1: **Input:** MC sample numbers m_l, m_p, confidence level $1-\alpha$
2: **Input:** Network G, diffusion process σ, data y, discrepancy function ℓ
3: **for** each candidate source node $s \in C_\sigma(y)$ **do**
4: Generate m_p+m_l samples $z_i \in \mathcal{Z}_\sigma, i = 1, \cdots, m_p+m_l$ from the diffusion process σ with source s on G.
5: Estimate expected loss $T_s(y)$ of data y as

$$\hat{T}_s(y) = \frac{1}{m_l} \sum_{i=m_p+1}^{m_p+m_l} \ell_s(y, z_i). \tag{3.6}$$

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

$$\hat{T}_s(\zeta(z_j)) = \frac{1}{m_l} \sum_{i=m_p+1}^{m_p+m_l} \ell(\zeta(z_j), z_i). \tag{3.7}$$

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

$$\hat{\psi}_s(y) = \frac{1}{m_p} \sum_{j=1}^{m_p} \mathbb{I}(\hat{T}_s(\zeta(z_j)) \geq \hat{T}_s(y)). \tag{3.8}$$

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

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

Chapter 4

Monte Carlo Acceleration

In section 3.4, we reduced the computation time for estimating a single p-value to $\tilde{O}(m_p + m_l)$ for sampling for a single node and $\tilde{O}((m_p + m_l)T)$ for p-value estimation. This is arguably the minimum possible in our framework for the unweighted SI model since even sampling $m_p + m_l$ samples already takes $\tilde{O}((m_p + m_l)T)$, and the weighted SI, IC, and LT models take even longer for sampling. 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 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 an appropriate symmetric reflection. We call such a property *node isomorphism*. Denote the node u's kth order neighborhood– the set of all nodes (at most) 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* ***kth-order isomorphic*** *if there exists a bijective mapping $\pi : V \to V$, such that: (1) $\pi(u) = v$; (2)*

$\pi(i) = i, \text{if } \; i \notin \{u, v\} \cup N_k(u) \cup N_k(v)$; *(3)* if $\; (u, v) \in E, \;$ then $(\pi(u), \pi(v)) \in E$,

For illustration, consider a simplified case of a node 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 this simplified case as it allows permutations to nodes in the kth order neighborhoods of u and v. 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 2. *Let u and v be kth-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, then define the permuted path as*

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

Then Z_π has the same distribution as a random diffusion path from source v for any choice of diffusion process.

Note that Theorem 2 is only a sufficient condition. For example, it is possible for two nodes u and v to share the same path diffusion distribution if after applying a permutation π, the network structure is preserved within the T neighborhood of u and v for the SI model with fixed T. Finding all pairs of nodes that share a distribution is difficult for general diffusion process, so we limit our focus to node isomorphisms.

4.1.1 Efficient Identification of Node Isomorphisms

To apply the MC samples of one node to its set of isomorphic counterparts according to Theorem 2, we need an efficient algorithm to identify isomorphic pairs and the corresponding permutations. Directly checking Definition 4 is costly and related to the widely studied problem of graph isomorphisms discussed below. To efficiently find node isomorphisms, we introduce two approaches for identifying permutations, the first of which is based on graph spectral properties.

Let A be the network adjacency matrix with $|V| = n$ and $A = U\Lambda U^T$ be the eigen-decomposition of A. Note that requiring $A = A_{\tilde{\pi}}$ where $A_{\tilde{\pi}}$ is the

adjacency matrix resulting from applying π to V is equivalent to the third condition in Definition 4. Given an arbitrary permutation from $\pi : V \to V$, let P be the permutation matrix (by permutation rows of I according to π). Then $PAP^T = A_{\tilde{\pi}}$ is the matrix of permuting the rows and columns of A according to π. Notice that $PP^T = I$, because PIP^T is applying the same permutation to the rows and columns of I, which always give I itself. In particular, this property indicates that

$$PU\Lambda(PU)^T$$

is also the eigen-decomposition of PAP^T. In particular, let W be the permutation matrix of $\tilde{\pi}$ in Definition 4. We then have

$$U\Lambda U^T = A = WAW^T = WU\Lambda(WU)^T.$$

Assume that among all eigenvalues, there are Q eigenvalues with multiplicity 1. Without loss of generality, denote them by $\lambda_1, \cdots, \lambda_Q$, and assume they are sorted by non-increasing magnitude: $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_Q| > 0$. Notice that for these eigenvalues, the corresponding eigenvectors are unique up to a sign. Therefore, we have

$$U_q = \tau_q P U_q, \tau_q \in \{\pm 1\}$$

where U_q is the qth column of U and the eigenvector for λ_q. Also notice that, if it happens that u and v are kth-order isomorphic but $\tau_q = -1$, notice that by the above relation, we know that $U_{i,q} = 0$ if i is not within k-hops to either of the nodes. So this condition provides a convenient to way to perform an adaptive check of node isomorphisms – we can start from a lower order first and stop earlier if we find they are isomorphic in a lower order.

Note that this approach contains a potentially significant computational cost for large graphs when performing the eigen-decomposition of A and subsequent comparisons of columns of U. This can be avoided, however, because a kth-order permutation only affects edges the $k+1$ neighborhoods of u and v. This leads to the following proposition.

Proposition 2. *Let G' be a subgraph of G induced by V' such that $\{u, v\} \cup N_{k+1}(u) \cup N_{k+1}(v) \subseteq V' \subseteq V$. Then u and v are kth order isomorphic on G if and only if the pair is kth order isomorphic on G'.*

This allows us to adaptively check an isomorphism candidate pair u, v on the subgraph of G induced by y_{k+1}, the $k+1$ neighborhood of diffusion sample

y. A description of this algorithm, along with an alternate degree based approach to identifying node isomorphisms is described in Appendix A.3.

4.1.2 Relation With the Graph Automorphism Problem

The graph isomorphism problem is a well studied problem in graph theory concerning the existence of a structure preserving mapping between two graphs G_1 and G_2. While it is known to reside in NP, it is one of the few problems where it is not known whether is lies in P or NP-Complete [38, 39, 40]. Efficient algorithms have been shown to exist for specific graph structures, such as trees, planar graphs, and graphs with bounded degree or eigenvalue multiplicity, however results for general network structures are not known [40, 41, 42, 43, 44]. The problem of identifying node isomorphisms seen in this research is closely related to the graph automorphism problem, which is a sister problem to finding graph isomorphisms.

Definition 5. *Given graph $G = (V, E)$, does there exists a non-trivial permutation $\pi : V \to V$ such that if $(u, v) \in E$, then $(\pi(u), \pi(v)) \in E$?*

The focus of this research is not to provide a heavy analysis of this problem, but instead to include a practical approach to finding node isomorphisms that is effective within the context of the DSI problem. There exist practical tools for solving the graph automorphism problem such as nauty [45], bliss [46], and saucy [47], however we observe that in the case of real world networks, high order node isomorphisms are extremely uncommon. This motivate focusing on finding lower order node isomorphisms quickly on the infected subset of a graph G.

4.2 Surjective Importance Sampling and Single-Degree Nodes in the SI Model

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, a certain mapping is needed to connect the two diffusion processes. The mapping is dependent on the choice of diffusion process, so here we will be considering the SI model, however alternate versions of this result exist for the IC and LT models. Importantly the method of surjective importance sampling is a general approach. The following theorem formulates this intuition.

Theorem 3. *Let u be a single-degree node in the graph G with the only neighbor node v_0. If a path $z \in \mathcal{Z}_\sigma$ 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*

$$f_u(z) = \{u, v_0, s_1, \cdots, s_{K-1}, s_{K+1}, \cdots, s_T\}. \tag{4.1}$$

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

$$\begin{aligned} \frac{p\left(f_u(z)|u\right)}{p\left(z|v_0\right)} &= \frac{1}{\mathbb{P}\left(u|v_0, s_1 \cdots s_{K-1}\right)} \\ &\quad \times \frac{1}{\prod_{k=1}^{K-1}\left(1 - \mathbb{P}\left(s_k|v_0, s_1 \cdots s_{k-1}\right)\right)} \end{aligned} \tag{4.2}$$

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 (4.2) 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 3, 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 [48]. However, 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 process the *surjective importance sampling* and give its property in the following theorem. We believe that this theorem could be of general interest beyond the context of diffusion source identification.

Theorem 4 (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*

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

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 4 when ϕ is the identity mapping. Theorem 3 and 4 together 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 3.

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 (3.7) in Algorithm 5 can be replaced by $\hat{T}_u\left(\zeta\left(f_u\left(z_j\right)\right)\right)$ equalling the following

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

and Equation (3.8) can be replaced by $\hat{\psi}_u(y)$ equalling the following

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

where $z_j, j = 1, \cdots, m_l + m_p$ 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 A.2).

Chapter 5

Experiments

In this chapter a suite of experimental results are presented to provide an experimental verification of the confidence set method and an analysis of the model on different networks, diffusion processes, and discrepancy functions.

5.1 Evaluation Setup Details

The implementation of our methods are provided in the form of a Python package [1]. The code is structured as a python package with three primary sub-packages, each corresponding to a particular programmable facet of the confidence set model. "diffusion_source.graphs" provides classes for a superset of the graphs tested in this section, as well as a wrapper for importing arbitrary graph structures. "diffusion_source.infection_model" provides an abstract implementation of the confidence set method that accommodates the general definition for σ. Additionally, explicit implementations for the SI, IC, and LT models are included. Finally "diffusion_source.discrepancies" provides implementations of the set of the discrepancy functions studied in this research. Additional information about usage can be found with the code base, however the code importantly provides an abstract design that allows the results presented in this research to easily be extended to alternate choices of diffusion process, network structure, and discrepancy function.

[1] The code can be found on Github at https://github.com/lab-sigma/Diffusion-Source-Identification.

5.1.1 Implementation Challenges and Minor Optimizations

Here we note some minor optimizations found in the implementation not discussed in the main body. First, to verify the statistical properties of the confidence set we require a large number of repetitions of the Monte Carlo procedure. This poses a substantial compute challenge in generating $O(KN(m_l + m_p))$ samples where N is the average infection sample size and K is the number of trials. One optimization is to leverage the property of the canonical family of loss functions to precompute the infection frequency matrix M described in section 3.4 for each choice of source node s. While storing the full set of m_l samples used in loss calculation is too space inefficient in practice, storing M is possible for moderately sized networks ($\sim$ 3gb for 3182 node network with time depth). This relegates the m_l factor to $O(nm_l)$ number of samples in a preprocessing step where $|V| = n$. This strategy is impractical for real world use, but is very effective when studying different choices of discrepancy function.

Another difficulty was observed when moving from the unweighted to weighted SI model. There is a linear complexity discrepancy during the edge sampling step due to having a weighted distribution (step 4 in Algorithm 1 and Algorithm 2). If we restrict the edge weights to integer values, the difficulty of sampling the edge can be recouped up to a factor of the average node degree d. This proves to be adequate for integer weight networks, however without parallelization, sampling for general weighted networks proves to be slow in practice. A different problem is observed for the IC and LT models where the runtime depends heavily on how large the sample infected set $|y| = T$ is. As a result, to provide accurately distributed data, the IC and LT models are only studied on smaller networks.

5.1.2 System Used in Evaluation

All experiments were executed on the UVa research compute system Rivanna. Executions of each algorithm were submitted as individual jobs with the Slurm workload manager and run in parallel. Rivanna is a heterogeneous cluster with 595 nodes and over 22598 cores. More information can be found at Rivanna's overview page.

5.1.3 Network Details

Three random network models were chosen for evaluation of the unweighted SI model: 4-regular trees, the preferential attachment model [49], and the small-world (S-W) network model [50]. 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 [51]. The small-world property is believed to be prevalent in social networks [50]. The network size is $N = 1365$ (the size of regular tree with degree 4 and depth 6). Details are shown in table 5.1

Table 5.1: Properties of each of the synthetic unweighted networks used in the evaluation.

Name	N	Avg. Degree	Max Degree
4-reg. tree	1365	0.999	4
Pref. Att.	1365	0.999	70
S-W	1365	2	7

Evaluations of weighted diffusion processes are on a set of five real world networks based on global airport flight traffic from (3182 nodes) [52], a statistician citation network (2654 nodes) [53], and three smaller hiring networks (Business School, Computer Science, and History) [54]. We consider both an undirected and directed version of these networks. The undirected version is constructed by using the weights in the upper triangle of the adjacency matrix in each network, unless the weight $w_{ij} = 0$ and $w_{ji} > 0$ with $i > j$. Properties for each of these networks are included in tables 5.2 and 5.3.

5.2 Evaluation of the SI model on Synthetic Networks

In this section, we evaluate our proposed methods on the synthetic network models with the unweighted SI model. In these experiments, both Monte Carlo numbers, m_p and m_l, are 10000. Source nodes are randomly sampled, and the reported results are an averaged across 200 replications.

Table 5.2: Properties of each of the weighted networks used in the evaluation. (undirected)

Name	N	Avg. Degree	Max Degree
AFT	3182	5.90	248
Stat. Cit.	2654	7.55	298
BSchool	113	26.1	111
Comp. Sci	206	13.3	172
History	145	15.6	124

Table 5.3: Properties of each of the weighted networks used in the evaluation. (directed)

Name	N	Avg. Deg.	Max in Deg.	Max out Deg.
AFT	3182	11.6	238	239
Stat. Cit.	2654	8.13	143	239
BSchool	113	30.4	111	51
Comp. Sci	206	14.2	172	33
History	145	16.7	124	33

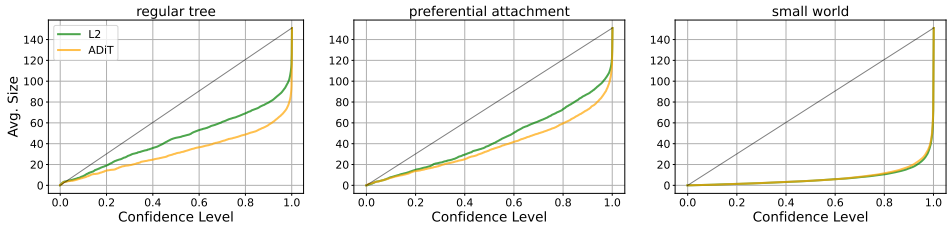

Figure 5.1: Average confidence set size as a function of confidence level for synthetic networks in the unweighted SI model.

5.2.1 Confidence validity evaluation on the SI model

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 [4, 7, 8, 9], as well as estimator using our proposed framework with discrepancy functions ℓ_{se} and ADiT. The result is shown in the Table 5.4. 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 5.4: 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 5.5 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 substan-

Table 5.5: The average coverage rate of the confidence sets across 200 replications. The standard error for the coverage rate is about 2.1% and 2.8% for 90% and 80% confidence sets, respectively.

	4-REG. TREE	PREF. ATT.	S-W
EUCLIDEAN-90%	93%	88%	91%
SIZE	79.4	87.8	16.3
ADiT-90%	91.5%	89%	90.5%
SIZE	58.1	72.7	17.98
EUCLIDEAN-80%	82.5%	80%	80.5%
SIZE	69.1	73.2	10.6
ADiT-80%	81.5%	80.5%	83.5%
SIZE	48.8	59.7	11.5

tially 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.

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 5.1. Notably, the relation is sub-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 5.4 and support our argument that, in general, inferring the source by a single-point estimator is hopeless. Figure 5.2 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.

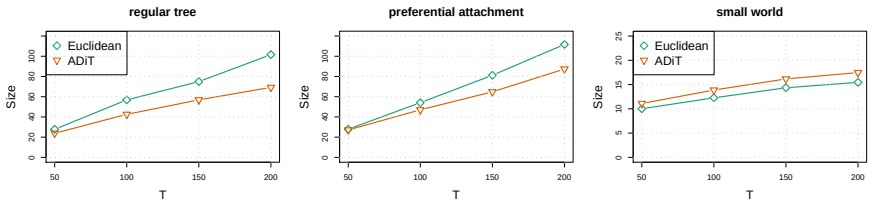

Figure 5.2: The average size of 90% confidence sets for T values.

Table 5.6: 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

5.2.2 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 5.6. The timing included is only the sequential version 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 7 in Appendix A.4). 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 5.7 includes the timing results of the parallel version implementation based on 20 cores in the same settings as Table 5.6. 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 5.7: 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 [55] 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 A.5.

5.3 Comparison of Different Discrepancies and Diffusion Processes

Here, we present a comparison of our model for different diffusion processes and choices of discrepancy functions. All figures were generated with $m_l = 10000$, $m_p = 2000$, and $K = 1000$ replications unless noted otherwise. Weighted SI model evaluations are on all five of the previously described weighted networks. IC and LT results focus on the three hiring networks. In addition to L2 and ADiT loss functions, we consider two additional losses. First is "**Averaged Deviation Time**" (ADT) which is simply the inverse of ADiT.

$$h(t_z(v)) = t_z(v). \tag{5.1}$$

This loss is included in all cases. The other loss is included for the case of IC and LT because the canonical form of the L2 loss depends on a fixed infection size, which does not apply to those two models. This loss is non-canonical but still admits efficient computation strategies based on the stored matrix M. We denote it as "Z_-" and it represents the count of how many nodes are found in the diffusion path but not in the sample data.

$$Z_-(y, z) = \sum_{v:\zeta(z)_v=1} \mathbb{I}(v \notin y) \tag{5.2}$$

For correctness we will denote the canonical "L2" loss in the IC and LT models as Z_+.

$$Z_+(y, z) = -\sum_{v:y_v=1} \mathbb{I}(v \in z) \tag{5.3}$$

We start by observing experimental coverage levels in the unweighted SI model. The trends shown in Figure 5.3 indicate that the true coverage for closely matches the theoretical lower bound. Points where the observed confidence drops below the nominal level can be explained by the variance introduced by approximating the coverage with repetitions of the model. Figure 5.4 shows the same coverage relationship on the airport flight traffic and statistician citation networks.

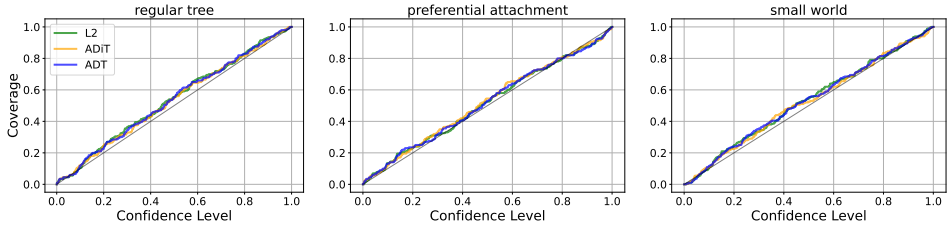

Figure 5.3: The relationship between confidence level $(1 - \alpha)$ and true coverage level in the Unweighted SI model ($K = 200$, $m_l = m_p = 10000$).

The observed true coverage levels for confidence construction in the case of the SI model matches the results shown in the previous section. Figure 5.1 shows a sublinear relationship between the confidence level and the confidence set size for L2 and ADiT in the unweighted case. As a comparison, Figure 5.5 shows a similar sublinear relationship between the confidence level and the

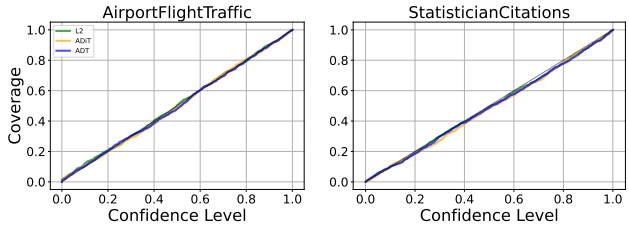

Figure 5.4: The relationship between confidence level $(1 - \alpha)$ and true coverage level in the unweighted SI model (undirected).

confidence set size. This is unsurprising given the similarity between the unweighted and weighted diffusions.

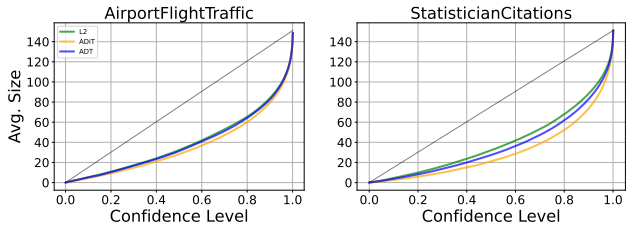

Figure 5.5: Average confidence sets size as a function of confidence level for two networks in the weighted SI model (undirected).

In figure 5.5, ADiT outperforms L2 while ADT falls between the two. In all cases we observe a sublinear function of the confidence set size, meaning the Monte Carlo algorithm is able to provide a better confidence set size than uniformly random sampling from the infected sample y. Inference in the small world networks appears to be easier than other network models. If the intuition for designing ADiT is to put more importance on nodes infected early in the diffusion procedure, then a reasonable question is why ADT, which applies the inverse relationship, outperforms L2. This suggests that the incorporation of infection order information into the discrepancy function is critical to designing effective loss functions.

Now, looking at the results for coverage in the directed case of the SI model shown in figure 5.6, confidence set performance appears poor for low confidence levels. This is contrasted by the true coverage levels for these confidence sets, shown in figure 5.7. If we instead plot the confidence set

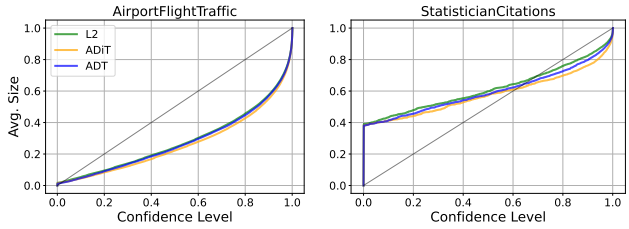

Figure 5.6: Average confidence sets size as a function of confidence level for two networks in the weighted SI model (directed).

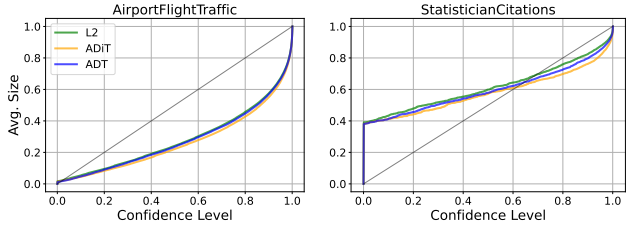

Figure 5.7: True confidence set coverage function of confidence level for two networks in the weighted SI model (directed).

size against the true confidence level, as in figure 5.8, then we see a more useful relation. This can be explained by considering a difficulty in directed networks not found in undirected ones, specifically for the fixed infection size SI model. If starting from a particular source is unable to infect the full set of $T = 150$ nodes, then the resulting infected set is guaranteed to include all nodes it is able to infect. Moreover, all candidate nodes (which might be a single one) will produce identical infected sets, making useful inference impossible. In the case of the flight traffic network, most nodes are strongly connected, while that is not the case in the statistician citation network.

Moving on to the undirected IC and LT models, we observe a similar confidence set size relation. The confidence set size results for the IC and LT models in figure 5.9 for the BSchool hiring network then seem to suggest that the discrepancy functions considered thus far are not well suited for those models.

Comparing the confidence set size to the true confidence level shown in figure 5.10, however, we observe an interesting trend. While the confidence

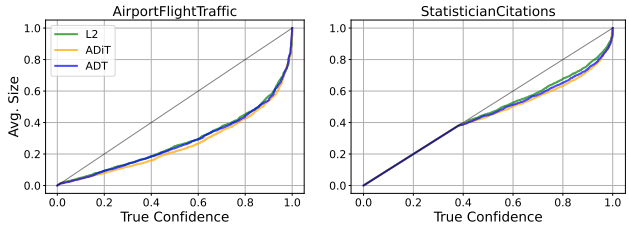

Figure 5.8: Average confidence sets size as a function of confidence level for two networks in the weighted SI model (undirected).

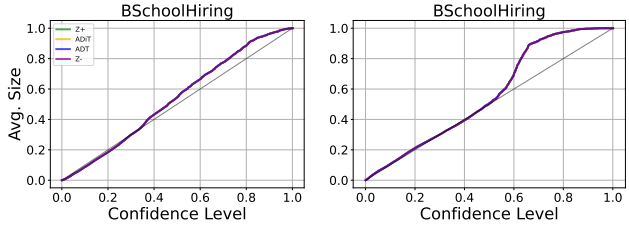

Figure 5.9: Average confidence set size as a function of confidence level for the IC (left) and LT (right) models on the business school hiring network.

set size appears worse than linear, the true confidence level for the $1-\alpha$ level confidence sets for smaller values of α are larger than the nominal level for both model which reflects the relation seen in the confidence set sizes for each model. This is reflected in figure 5.11, which shows that the confidence set is approximately equal to $(1-\alpha)$ times the infected set size. For the IC and LT models we see that all of the loss functions shown here display nearly identical performance. This shows that the confidence set matches the performance of randomly selecting $(1-\alpha)$ of the infected nodes in these two models. Results for the directed case show a similar trend are left to appendix A.6.. This is a potential future direction of study to better understand the difficulty of inference in the DSI problem for the case of IC and LT.

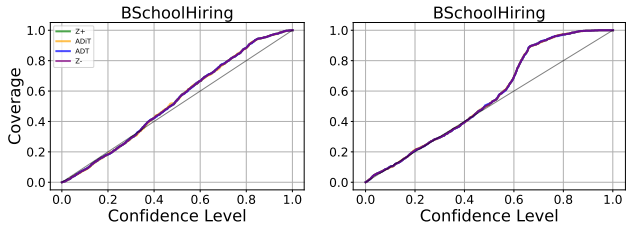

Figure 5.10: True coverage level as a function of confidence level for the IC (left) and LT (right) models on the business school hiring network.

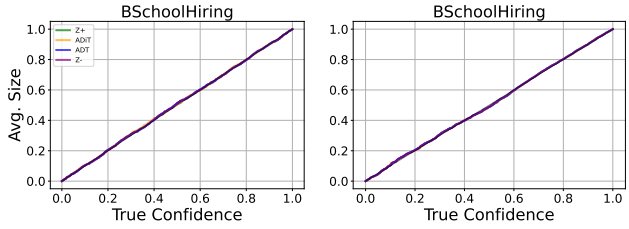

Figure 5.11: Confidence set size compared against the true coverage level for the IC (left) and LT (right) models on the business school hiring network.

Chapter 6

Future Directions and Conclusion

The problem of discrepancy function design for general networks remains an open problem and a potential focus for future work. Considering a more general class of loss functions may be necessary to achieve good performance across different diffusion processes and network structures. One limitation of this model is in generalizing to multiple sources. The study of IC and LT models in other problems (e.g. influence maximization) typically allow diffusions to begin from a set of sources, however the current confidence set algorithm requires enumeration of the combination of source nodes to achieve the desired statistical confidence which is impractical. Another potential direction for further study is in considering cases with limited information about either the infected status or topology of the network. This might better match real world settings where full information about network topology or the status of an epidemic is usually unavailable.

This research contributes 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 and any diffusion procedure. 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.

Bibliography

[1] Mark EJ Newman, Stephanie Forrest, and Justin Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3): 035101, 2002.

[2] Avner Halperin and Gal Almogy. System and method of virus containment in computer networks, December 19 2002. US Patent App. 10/058,809.

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

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

[5] Devavrat Shah and Tauhid Zaman. Finding rumor sources on random graphs. 2012.

[6] Wenxiang Dong, Wenyi Zhang, and Chee Wei Tan. Rooting out the rumor culprit from suspects. In *2013 IEEE International Symposium on Information Theory*, pages 2671–2675. IEEE, 2013.

[7] Justin Khim and Po-Ling Loh. Confidence sets for the source of a diffusion in regular trees. *IEEE Transactions on Network Science and Engineering*, 4(1):27–40, 2016.

[8] Sébastien Bubeck, Luc Devroye, and Gábor Lugosi. Finding adam in random growing trees. *Random Structures & Algorithms*, 50(2):158–172, 2017.

[9] Pei-Duo Yu, Chee Wei Tan, and Hung-Lin Fu. 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*, pages 175–192. Springer, 2018.

[10] Harry Crane and Min Xu. Inference on the history of a randomly growing tree. *arXiv preprint arXiv:2005.08794*, 2020.

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

[12] Hung T Nguyen, Preetam Ghosh, Michael L Mayo, and Thang N Dinh. Multiple infection sources identification with provable guarantees. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1663–1672, 2016.

[13] S Jalil Kazemitabar and Arash A Amini. Approximate identification of the optimal epidemic source in complex networks. In *International Conference on Network Science*, pages 107–125. Springer, 2020.

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

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[16] China Data Lab. Baidu Mobility Data, 2020. URL `https://doi.org/10.7910/DVN/FAEZIO`.

[17] Tao Hu, Weihe Wendy Guan, Xinyan Zhu, Yuanzheng Shao, Lingbo Liu, Jing Du, Hongqiang Liu, Huan Zhou, Jialei Wang, Bing She, et al. Building an open resources repository for covid-19 research. *Data and Information Management*, 4(3):130–147, 2020.

[18] D. J. DALEY and D. G. KENDALL. Stochastic Rumours. *IMA Journal of Applied Mathematics*, 1(1):42–55, 03 1965. ISSN 0272-4960. doi: 10.1093/imamat/1.1.42. URL `https://doi.org/10.1093/imamat/1.1.42`.

[19] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[20] Kristina Lerman and Rumi Ghosh. Information contagion: an empirical study of the spread of news on digg and twitter social networks. *CoRR*, abs/1003.2664, 2010. URL http://arxiv.org/abs/1003.2664.

[21] Praneeth Netrapalli and Sujay Sanghavi. Finding the graph of epidemic cascades. *CoRR*, abs/1202.1779, 2012. URL http://arxiv.org/abs/1202.1779.

[22] W.O. Kermack and A.G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, 1991. doi: https://doi.org/10.1098/rspa.1927.0118. URL https://royalsocietypublishing.org/doi/10.1098/rspa.1927.0118.

[23] Joan L. Aron and Ira B. Schwartz. Seasonality and period-doubling bifurcations in an epidemic model. *Journal of Theoretical Biology*, 110(4):665–679, 1984. ISSN 0022-5193. doi: https://doi.org/10.1016/S0022-5193(84)80150-2. URL https://www.sciencedirect.com/science/article/pii/S0022519384801502.

[24] Deokjae Lee, Wonjun Choi, J. Kertész, and B. Kahng. Universal mechanism for hybrid percolation transitions. *Scientific Reports*, 7(1), Jul 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-06182-3. URL http://dx.doi.org/10.1038/s41598-017-06182-3.

[25] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 137–146, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581137370. doi: 10.1145/956750.956769. URL https://doi.org/10.1145/956750.956769.

[26] Letizia Milli and Giulio Rossetti. Community-aware content diffusion: Embeddednes and permeability. In Hocine Cherifi, Sabrina Gaito, José Fernendo Mendes, Esteban Moro, and Luis Mateus Rocha, editors,

Complex Networks and Their Applications VIII, pages 362–371, Cham, 2020. Springer International Publishing. ISBN 978-3-030-36687-2.

[27] Mateusz Wilinski and Andrey Y. Lokhov. Prediction-centric learning of independent cascade dynamics from partial observations, 2021.

[28] Paulo Shakarian, Abhinav Bhatnagar, Ashkan Aleali, Elham Shaabani, and Ruocheng Guo. *The Independent Cascade and Linear Threshold Models*, pages 35–48. Springer International Publishing, Cham, 2015. ISBN 978-3-319-23105-1. doi: 10.1007/978-3-319-23105-1_4. URL https://doi.org/10.1007/978-3-319-23105-1_4.

[29] Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978. ISSN 00029602, 15375390. URL http://www.jstor.org/stable/2778111.

[30] Roxana Alexandru and Pier Dragotti. Diffusion source detection in a network using partial observations. page 20, 09 2019. doi: 10.1117/12.2529418.

[31] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Spy vs. spy: Rumor source obfuscation. *CoRR*, abs/1412.8439, 2014. URL http://arxiv.org/abs/1412.8439.

[32] Miklós Z. Rácz and Jacob Richey. Rumor source detection with multiple observations under adaptive diffusions. *CoRR*, abs/2006.11211, 2020. URL https://arxiv.org/abs/2006.11211.

[33] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

[34] Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[35] Karl-Heinz Jockel. Finite sample properties and asymptotic efficiency of monte carlo tests. *The annals of Statistics*, pages 336–347, 1986.

[36] Julian Besag and Peter Clifford. Generalized monte carlo significance tests. *Biometrika*, 76(4):633–642, 1989.

[37] Karl Bringmann and Konstantinos Panagiotou. Efficient sampling methods for discrete distributions. *Algorithmica*, 79(2):484–508, 2017.

[38] Johannes Köbler, Uwe Schöning, and Jacobo Torán. The graph isomorphism problem: Its structural complexity. 1993.

[39] Jacobo Torán. On the hardness of graph isomorphism. *SIAM J. Comput.*, 33:1093–1108, 2004.

[40] Harm Derksen. The graph isomorphism problem and approximate categories, 2010.

[41] R.C. Busacker, R.G. Busacker, T.L. Saaty, and B. Grob. *Finite Graphs and Networks: An Introduction with Applications*. Number v. 10 in Finite Graphs and Networks: An Introduction with Applications. McGraw-Hill, 1965. ISBN 9780070093058. URL `https://books.google.com/books?id=AedLAAAAMAAJ`.

[42] J.E. Hopcroft and R.E. Tarjan. A v log v algorithm for isomorphism of triconnected planar graphs. *Journal of Computer and System Sciences*, 7(3):323–331, 1973. ISSN 0022-0000. doi: https://doi.org/10.1016/S0022-0000(73)80013-3. URL `https://www.sciencedirect.com/science/article/pii/S0022000073800133`.

[43] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982. ISSN 0022-0000. doi: https://doi.org/10.1016/0022-0000(82)90009-5. URL `https://www.sciencedirect.com/science/article/pii/0022000082900095`.

[44] László Babai, D. Yu. Grigoryev, and David M. Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, page 310–324, New York, NY, USA, 1982. Association for Computing Machinery. ISBN 0897910702. doi: 10.1145/800070.802206. URL `https://doi.org/10.1145/800070.802206`.

[45] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014. ISSN 0747-7171. doi: https://doi.org/10.1016/j.jsc.2013.09.003. URL `https://www.sciencedirect.com/science/article/pii/S0747717113001193`.

[46] Tommi Junttila and Petteri Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In *Proceedings of the Meeting*

on Algorithm Engineering Expermiments, page 135–149, USA, 2007. Society for Industrial and Applied Mathematics.

[47] Paolo Codenotti, Hadi Katebi, Karem A. Sakallah, and Igor L. Markov. Conflict analysis and branching heuristics in the search for graph automorphisms. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 907–914, 2013. doi: 10.1109/ICTAI.2013.139.

[48] Pierre L'Ecuyer and Art B Owen. *Monte Carlo and Quasi-Monte Carlo Methods 2008*. Springer, 2009.

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

[50] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

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

[52] Open Flights. Airport Flight Traffic, 2020. URL `https://openflights.org/data.html`.

[53] Pengsheng Ji and Jiashun Jin. Coauthorship and citation networks for statisticians. *The Annals of Applied Statistics*, 10(4), Dec 2016. ISSN 1932-6157. doi: 10.1214/15-aoas896. URL `http://dx.doi.org/10.1214/15-AOAS896`.

[54] Aaron Clauset, Samuel Arbesman, and Daniel B. Larremore. Systematic inequality and hierarchy in faculty hiring networks. *Science Advances*, 1(1):e1400005, 2015. doi: 10.1126/sciadv.1400005. URL `https://www.science.org/doi/abs/10.1126/sciadv.1400005`.

[55] Amir Ghasemian, Homa Hosseinmardi, Aram Galstyan, Edoardo M Airoldi, and Aaron Clauset. Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117(38):23393–23400, 2020.

Appendix A

Appendix

Appendix to "Diffusion Source Identification on Networks with Statistical Confidence"

A.1 Proofs

A.1.1 Proof of Proposition 1

Both the two claims are straightforward to show by using the definition. First, we have

$$\mathbb{E}_s \ell_{rc}(y, z) = 1 - \mathbb{E}\mathbb{I}(y = \zeta(z)) = 1 - \mathbb{P}_s(\zeta(Z) = y).$$

So minimizing the loss is equivalent to the MLE, which is equivalent to the rumor center in infinite regular trees.

Secondly, notice that that both y and $\zeta(Z)$ are n dimensional binary vectors. So

$$\|y - \zeta(Z)\|_2^2 = \sum_i I(y_i \neq \zeta(Z)_i)$$

which is the symmetric difference between the set $\{i : y_i = 1\}$ and $\{i : \zeta(Z)_i = 1\}$.

A.1.2 Proof of Theorem 1

Define $q^{\alpha}_{T,s^*} = \inf_t\{t : \mathbb{P}_{s^*}(T_{s^*}(\zeta(Z)) \geq t) \leq \alpha\}$. Notice that q^{α}_{T,s^*} can be seen as one generalized definition for the right quantile of the distribution of the

random variable $T_{s^*}(\tilde{Y})$, where $\tilde{Y} := \zeta(Z)$ is a random infection status of the network generated by the diffusion process starting from s^*.

Now assume Y is a random infection status from the diffusion process from s^*. We have two cases. First, let $\mathbb{P}_{s^*}(T_{s^*}(\zeta(Z)) \geq q^{\alpha}_{T,s^*}) \leq \alpha$:

According to the definition of the p-value, we have

$$\begin{aligned}
\mathbb{P}_{s^*}\left(s^* \in S(Y)\right) &= \mathbb{P}_{s^*}\left(\psi_{s^*}(Y) > \alpha\right) \\
&= \mathbb{P}_{s^*}\left(\mathbb{P}_{s^*}\left(T_{s^*}(\tilde{Y}) \geq T_{s^*}(Y)\right) > \alpha\right) \\
&= \mathbb{P}_{s^*}\left(T_{s^*}(Y) < q^{\alpha}_{T,s^*}\right) \\
&= 1 - \mathbb{P}_{s^*}\left(T_{s^*}(Y) \geq q^{\alpha}_{T,s^*}\right).
\end{aligned}$$

Note that since s^* is the true source node, $T_{s^*}(Y)$ and $T_{s^*}(\tilde{Y})$ are following exactly the same distribution, thus

$$\mathbb{P}_{s^*}\left(s^* \in S(Y)\right) = 1 - \mathbb{P}_{s^*}\left(T_{s^*}(Y) \geq q^{\alpha}_{T,s^*}\right) \geq 1 - \alpha.$$

Now, let $\mathbb{P}_{s^*}(T_{s^*}(\zeta(Z)) \geq q^{\alpha}_{T,s^*}) > \alpha$

$$\begin{aligned}
\mathbb{P}_{s^*}\left(s^* \in S(Y)\right) &= \mathbb{P}_{s^*}\left(\psi_{s^*}(Y) > \alpha\right) \\
&= \mathbb{P}_{s^*}\left(\mathbb{P}_{s^*}\left(T_{s^*}(\tilde{Y}) \geq T_{s^*}(Y)\right) > \alpha\right) \\
&= \mathbb{P}_{s^*}\left(T_{s^*}(Y) \leq q^{\alpha}_{T,s^*}\right) \\
&= 1 - \mathbb{P}_{s^*}\left(T_{s^*}(Y) > q^{\alpha}_{T,s^*}\right).
\end{aligned}$$

We then get a similar relationship.

$$\mathbb{P}_{s^*}\left(s^* \in S(Y)\right) = 1 - \mathbb{P}_{s^*}\left(T_{s^*}(Y) > q^{\alpha}_{T,s^*}\right) \geq 1 - \alpha.$$

A.1.3 Proof of Theorem 3

Given a path generated by the diffusion process starting from v_0 **containing** u, denoted by

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

we match it to the path $f_u(Z)$ defined as

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

We start from the probability mass of z starting from v_0. By using the Markov property, we have

$$\begin{aligned} p(z|v_0) =& \mathbb{P}(s_1|v_0)\mathbb{P}(s_2|v_0, s_1)\cdots\mathbb{P}(s_{K-1}|v_0, s_1, \cdots, s_{K-2}) \\ & \times \mathbb{P}(s_{K+1}|v_0, \cdots, u)\cdots\mathbb{P}(s_T|v_0, \cdots, s_{T-1}) \\ & \times \mathbb{P}(u|v_0, s_1, \cdots, s_{K-1}) \end{aligned} \tag{A.1}$$

In contrast, for the path $f_u(z)$, we have

$$\begin{aligned} \mathbb{P}(f_u(z)|u) =& \mathbb{P}(v_0|u)\mathbb{P}(s_1|u, v_0)\mathbb{P}(s_2|u, v_0, s_1)\cdots\mathbb{P}(s_{K-1}|u, v_0, s_1, \cdots, s_{K-2}) \\ & \times \mathbb{P}(s_{K+1}|u, v_0, \cdots, s_{K-1})\cdots\mathbb{P}(s_T|u, v_0, \cdots, s_{T-1}) \\ =& \mathbb{P}(s_1|u, v_0)\mathbb{P}(s_2|u, v_0, s_1)\cdots\mathbb{P}(s_{K-1}|u, v_0, s_1, \cdots, s_{K-2}) \\ & \times \mathbb{P}(s_{K+1}|u, v_0, \cdots, s_{K-1})\cdots\mathbb{P}(s_T|u, v_0, \cdots, s_{T-1}). \end{aligned} \tag{A.2}$$

Notice that the conditional probability $\mathbb{P}(s_{k+1}|v_0, \cdots, u, s_{K+1}, \cdots s_k), k > K$ only depends on the infection status before the kth infection and is invariant to the infection order. This property indicates that all terms after the $K+1$th (in the second rows) of (A.1) and (A.2) are equal.

Next, we compare the terms in the first line in each of (A.1) and (A.2). Notice that for each $k < K$, the term $\mathbb{P}(s_k|v_0, s_1, \cdots, s_{k-1})$ is identical for each available connections given the infected nodes $v_0, s_1, \cdots, s_{k-1}$ while the term $\mathbb{P}(s_k|u, v_0, s_1, \cdots, s_{k-1})$ is identical on all available edges given the infected nodes $u, v_0, s_1, \cdots, s_{k-1}$. The only difference in the two infected sets is on u. Since u has only one connection to v_0, at each point, the number of available infecting edges is one more in the former case. Therefore, we have

$$\mathbb{P}(s_k|u, v_0, s_1, \cdots, s_{k-1}) = \frac{1}{1 - \mathbb{P}(s_k|v_0, s_1, \cdots, s_{k-1})}\mathbb{P}(s_k|v_0, s_1, \cdots, s_{k-1}), k < K.$$

In addition, notice that in the third line of (A.1), there is one extra term that does not appear in (A.2). Combining the aforementioned three relations, we final obtain probability mass factor to be

$$\frac{\mathbb{P}(f_u(\pi)|S_0 = u)}{\mathbb{P}(\pi|S_0 = v_0)} = \frac{1}{\mathbb{P}(u|v_0, s_1, \cdots, s_{K-1})}\prod_{i=1}^{K-1}\frac{1}{1 - \mathbb{P}(s_i|v_0, s_1, \cdots, s_{i-1})} \tag{A.3}$$

Moreover, if z does not contain u, we set the ratio to be 0.

A.1.4 Proof of Theorem 4

Since Z_i's are a random sample from p_1, under the current assumption, by the strong law of large numbers, we have

$$\mathbb{P}\left(\hat{\eta} \to \mathbb{E}_1[\frac{g(\phi(Z))}{|\phi^{-1}(\phi(Z))|}\frac{p_2(\phi(Z))}{p_1(Z)}]\right) = 1.$$

Notice that ϕ is a surjection. Therefore, the term $\mathbb{E}_1[\frac{g(\phi(Z))}{|\phi^{-1}(\phi(Z))|}\frac{p_2(\phi(Z))}{p_1(Z)}]$ can be rewritten as

$$\begin{aligned}
\mathbb{E}_1[\frac{g(\phi(Z))}{|\phi^{-1}(\phi(Z))|}\frac{p_2(\phi(Z))}{p_1(Z)}] &= \sum_{z\in\mathcal{C}_1}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}\frac{p_2(\phi(z))}{p_1(z)}p_1(z) \\
&= \sum_{z\in\mathcal{C}_1}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z)) \\
&= \sum_{z\in\mathcal{C}_1'}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z)) + \sum_{z\in\mathcal{C}_1/\mathcal{C}_1'}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z)) \\
&= \sum_{z\in\mathcal{C}_1'}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z)) \\
&= \sum_{\tilde{z}\in\mathcal{C}_2}\sum_{z:\phi(z)=\tilde{z}}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z)) \\
&= \sum_{\tilde{z}\in\mathcal{C}_2}\sum_{z:\phi(z)=\tilde{z}}\frac{g(\tilde{z})}{|\phi^{-1}(\tilde{z})|}p_2(\tilde{z}) \\
&= \sum_{\tilde{z}\in\mathcal{C}_2}\sum_{z\in\phi^{-1}(\tilde{z})}\frac{g(\tilde{z})}{|\phi^{-1}(\tilde{z})|}p_2(\tilde{z}) \\
&= \sum_{\tilde{z}\in\mathcal{C}_2}|\phi^{-1}(\tilde{z})|\frac{g(\tilde{z})}{|\phi^{-1}(\tilde{z})|}p_2(\tilde{z}) \\
&= \sum_{\tilde{z}\in\mathcal{C}_2}g(\tilde{z})p_2(\tilde{z}) \\
&= \mathbb{E}_2[g(Z)].
\end{aligned}$$

A.1.5 Proof of Corollary 1

We will use $f_u(z)$ in place of ϕ to apply Theorem 4. The only remaining step is to find $|f_u(f_u^{-1}(z))|$. By the definition of f_u in Theorem 4, it is easy to

see that the other $T-1$ nodes (except u and v_0) and their order uniquely determine to the mapped path. Therefore, $|f_u(f_u^{-1}(z))|$ would always be T in this situation.

A.1.6 Proof of Theorem 2

Notice that, given σ, the probability mass function of a diffusion path only depends on the network A and the source node. Condition 1 of Definition 4 indicates that Z_π starts from v. Condition 2 and condition 3 of Definition 4 together indicate that Z_π is also a valid diffusion path. Condition 3, in particular, indicates that

$$p_u(Z) = p_v(Z_\pi).$$

Now define can define π^{-1} to be the inverse of π. For any $\tilde{Z}$ from v, for the same reason, $\tilde{Z}_{\pi^{-1}}$ is also a valid diffusion path starting from u. In particular, we have $Z = (Z_\pi)_{\pi^{-1}}$. Therefore, Z_π has the same sample space and probability mass function as the random diffusion path from v.

A.1.7 Proof of Proposition 2

For convenience, let

$$N_{k+1}(u,v) = \{u,v\} \cup N_{k+1}(u) \cup N_{k+1}(v)$$

and

$$N_k(u,v) = \{u,v\} \cup N_k(u) \cup N_k(v)$$

Now, let u and v be kth order isomorphic on G' with bijective mapping π' satisfying Definition 4. Define $\pi : V \to V$ such that $\pi(i) = \pi'(i)$ for $i \in V'$ and $\pi(i) = i$ otherwise. Now, because π' is a valid kth-order isomorphic permutation for u and v on G', we get $\pi(u) = \pi'(u) = v$ and $\pi(v) = \pi'(v) = u$ showing Condition 1. Second, because π' is a valid kth order permutation on V', we get

$$\pi(i) = \pi'(i) = i \text{ for } i \in V' \setminus N_k(u,v)$$

and we get $\pi(i) = i$ for $i \in V \setminus V'$, showing Condition 2. Finally, consider three cases for edge $(i,j) \in E$. First, if $i,j \in V'$, then $(\pi(i),\pi(j)) \in E$ by the validity of π'. Second, if $i,j \in V \setminus V'$, then $(\pi(i),\pi(j)) = (i,j) \in E$ holds

as well. Finally, let $i \in V'$ and $j \in V \setminus V'$. In this case, if $i \in N_k(u, v)$, then it must hold that $i \in$, which contradicts $N_{k+1}(u, v) \subseteq V'$. Then, because π' corresponds to a kth order mapping in V' and $i \notin N_k(u, v)$, it must hold that $(\pi(i), \pi(j)) = (i, j) \in E$. This proves Condition 3 and shows that u and v are kth order isomorphic on G.

Now instead let u and v be kth order isomorphic on G with bijective mapping π. Define $\pi' : V' \to V'$ such that $\pi'(i) = \pi(i)$. Then, Conditions 1 holds from $\pi(u) = v$ and $\pi(v) = u$ and $u, v \in V'$. Condition 2 holds because π is kth order in G, and $N_k(u, v) \subseteq V'$. Finally, if $(i, j) \in E'$, then $(\pi'(i), \pi'(j)) = (\pi(i), \pi(j)) \in E$. This shows that u and v are kth order isomorphic on G'.

A.2 Loss Function Computation Acceleration for Surjective Importance Sampling

The calculation strategy for canonical discrepancy functions can also be further generalized to the weighted averaging scenario used for the single-degree

nodes in Section 4.2. Specifically, there we need to calculate terms like

$$\begin{aligned}
&\frac{1}{m_l}\sum_{i=m_p+1}^{m_p+m_l}\ell\left(y,f_u(z_i)\right)\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}= \\
&\qquad=-\frac{1}{m_l}\sum_{i=m_p+1}^{m_p+m_l}\sum_{v:y_v=1}\mathbb{I}(v\in f_u(z_i))h(t_{f_u(z_i)}(v))\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T} \\
&\qquad=-\frac{1}{m_l}\sum_{v:y_v=1}\sum_{i=m_p+1}^{m_p+m_l}\mathbb{I}(v\in f_u(z_i))h(t_{f_u(z_i)}(v))\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T} \\
&\qquad=-\frac{1}{m_l}\sum_{v:y_v=1}\sum_{i=m_p+1}^{m_p+m_l}\mathbb{I}(v\in f_u(z_i))\left[\sum_{k=1}^{T}\mathbb{I}(t_{f_u(z_i)}(v)=k)h(k)\right]\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T} \\
&\qquad=-\frac{1}{m_l}\sum_{v:y_v=1}\sum_{i=m_p+1}^{m_p+m_l}\sum_{k=1}^{T}\mathbb{I}(v\in f_u(z_i))\mathbb{I}(t_{f_u(z_i)}(v)=k)h(k)\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T} \\
&\qquad=-\frac{1}{m_l}\sum_{v:y_v=1}\sum_{k=1}^{T}h(k)\left[\sum_{i=m_p+1}^{m_p+m_l}\mathbb{I}(t_{f_u(z_i)}(v)=k)\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}\right].
\end{aligned}$$

Therefore, to use this strategy in Section 4.2, when general MC samples from v_0, in addition to caching M, we also want to cache the matrix adjusted by the factor

$$M_{v,k}^{(v_0\to u)}=\sum_{i=m_p+1}^{m_p+m_l}\mathbb{I}(t_{f_u(z_i)}(v)=k)\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}.$$

A.3 Algorithms for Node Isomorphism Identification

First, details for applying the spectral analysis to isomorphism identification is shown in Algorithm 1.

Algorithm 1 (Algorithm to check isomorphism up to kth order)**.** *Assume we have $U^{(Q)}=(U_1\cdots,U_Q)$ available. To check isomorphism up-to order k for node u and v:*

1. *Check if* $|U^{(Q)}|_{u\cdot} = |U^{(Q)}|_{v\cdot}$. *If not, return FALSE. Else, identify the potential* τ_q*'s based on the sign-matching. Create* $\tilde{U}^{(Q)} = (\tau_1 U_1 \cdots, \tau_Q U_Q)$. *Proceed to Step 2 with* $k = 0, \tilde{\pi} = \{u \to v\}$,

2. *Run the following recursive checking step*

   (a) *If* $k <= K$*: Take* ne_u^k *and* ne_v^k, *run* $\pi^k =$`DistanceMatching`$(ne_u^k, ne_v^k, \tilde{U}^{(Q)})$. *Check the following cases:*

   - *If fail, break and return FALSE.*
   - *If success, set* $\tilde{\pi} = [\tilde{\pi}, \pi^k]$. *Take set* $D = [n] \backslash \big(\cup_{k=0}^{k} [ne_u^k \cup ne_v^k]\big)$. *Check if* $\|U_{D,q}\| = 0$ *for all* q *with* $\tau_q = -1$. *If success, break and go to step 3. Else, go to Step 2 with* $k := k + 1$.

   (b) *If* $k > K$*: Break and return FALSE.*

3. *Final check: check the definition based on the current* $\tilde{\pi}$. *If success, return TRUE with* $\tilde{\pi}$. *Else, return FALSE.*

Algorithm 2 (`DistanceMatching`(ne_1, ne_2, U))**.** *Given two subsets of nodes* ne_1, ne_2 *and a matrix* U *with rows corresponding to the nodes. Set the mapping* $\pi : ne_1 \to ne_2$ *to be empty.*

1. *If* $|ne_1| \neq |ne_2|$, *return Failure. Else, go to Step 2.*

2. *For each* $u \in ne_1$*: find* $v \in ne_2$ *such that* $\|U_u - U_v\| = 0$. *If success, add the mapping* $\{u \to v\}$ *to* π, *and remove* u *from* ne_1 *and* v *from* ne_2. *Else, break and return Failure*

3. *Return Success with* π.

Now, consider another natural necessary condition for node isomorphisms based on node degrees.

Proposition 3. *If* u *and* v *are* k*th-order isomorphic, we must have* $d_u = d_v$ *and* $D_k(u) = D_k(v)$ *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* $N_k(u)$ *and* $N_k(v)$. *Furthermore,* u *and* v *have the same* $k+1$*th-order neighbor sets. That is,* $N_{k+1}(u) \setminus N_k(u) = N_{k+1}(v) \setminus N_k(v)$.

Based on the properties in Proposition 3, Algorithm 6 finds all first-order isomorphic pairs and their associated permutations in the network. Next, we provide a proof of Proposition 3.

Proof of Proposition 3. $d_u = d_v$ because π gives a 1-1 mapping from $N_k(u)$ to $N_k(v)$. Furthermore, we have $\pi(N_k(u)) = N_k(v)$. By condition 3 of Definition 4, we also have $D_k(u) = D_k(v)$.

For the last one, we can prove by contradiction. Suppose there exists a node w, such that $w \in N_{k+1}(u)$ but $w \notin N_{k+1}(v)$. Since $\pi(N_k(u)) = N_k(v)$ while $\pi(w) = w$, so after applying the permutation to the network, we have $\pi(w)$ disconnected from $\pi(N_k(w))$. This contradicts condition 3 of Definition 4. □

Based on Proposition 3 we can efficiently identify isomorphism using pre-screening steps. And while Algorithm 6 can be extended to higher-order neighborhoods, identifying more isomorphic pairs, the complexity of identifying pairs with this method 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 and is the principle motivator for considering Algorithm 6. That is, for real world network examples, high order node isomorphisms are relatively uncommon, and the degree based approach for isomorphism searching can outperform the spectral approach in some cases.

A.4 Parallel Algorithm for Confidence Set Construction

As discussed in Section 3.3, our confidence set construction algorithm can be implemented in parallel, further boosting its speed. In the main paper, we only include the details and timing for the sequential version. The parallelized algorithm is described in Algorithm 7.

As can be seen, Algorithm 7 needs MC sampling for only one node in each group, and the calculations for other nodes can be done using pooled MC methods. When additional cores are available, the for-loops in the algorithm can be further parallelized.

A.5 Evaluation on 381 real-world networks

The data set from Ghasemian et al. [55] contains 550 networks. We focus on 381 networks with more than 200 nodes for stable evaluation. The removed

ones are either too small or have certain pathological structures to stable computation. The 381 networks are from six domains (71 biological, 110 economic, 9 informational, 105 social, 56 technological, and 30 transportation networks).

Though there are no real diffusion labels observed on these networks, we can generate a synthetic diffusion process based on our model. We generate a diffusion process with $T = \min(0.2N, 150)$. By doing this, we can evaluate the confidence set properties on these networks and the timing. The average coverage probability of the 90% confidence set and the relative size of $|\mathcal{C}|/T$ are shown in Table A.1. The results match what we observed previously on the simulated networks, and the ADiT is more effective than the Euclidean loss, indicating the valid method on the real-world network. More importantly, we also evaluate the timing improvement based on the pooled MC methods. We calculate the improvement percentage of the pooled MC strategies (over the vanilla MC) on each network. The results are summarized in Figure A.1. As can be seen, the pooled MC is very effective on economic networks and social networks, resulting in an average improvement of 40%. It is moderately effective on biological and informational networks with 10%-20% improvement. The technological networks and transportation networks are suitable structures for the strategy. The economic, social, and biological networks are the three largest domains in the data set. These results demonstrate the pooled MC's potential as a general computational strategy.

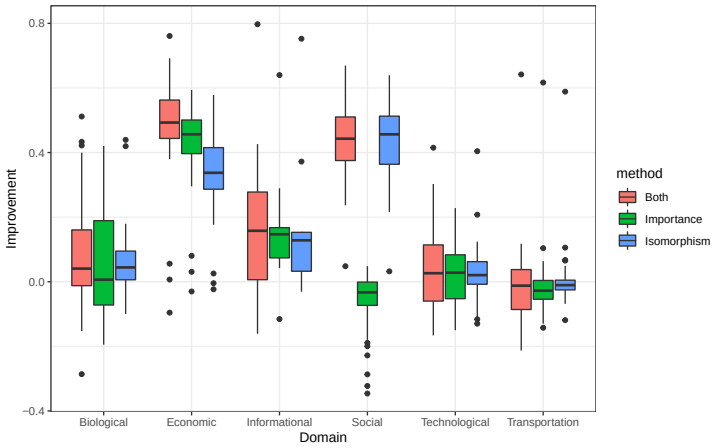

Figure A.1: The timing improvement proportion by the pooled MC over the vanilla MC on 381 real-world networks.

Table A.1: The average coverage rate of the 90% confidence sets and the relative size $|\mathcal{C}|/T$ on the 381 real-world networks.

	Bio.	Econ.	Info.	Social	Tech.	Transport.
Euclidean-90%	90.0%	90.1%	88.9%	89.8%	89.1%	90.9%
$\lvert\mathcal{C}\rvert/T$	0.60	0.41	0.66	0.44	0.57	0.38
ADiT-90%	90.2%	89.9%	87.6%	89.3%	89.3%	90.1%
$\lvert\mathcal{C}\rvert/T$	0.55	0.36	0.61	0.40	0.52	0.35

A.6 Coverage Rates and Confidence Sizes for Multiple Models and Networks

The remainder of the figures discussed in section 5.3 are provided here.

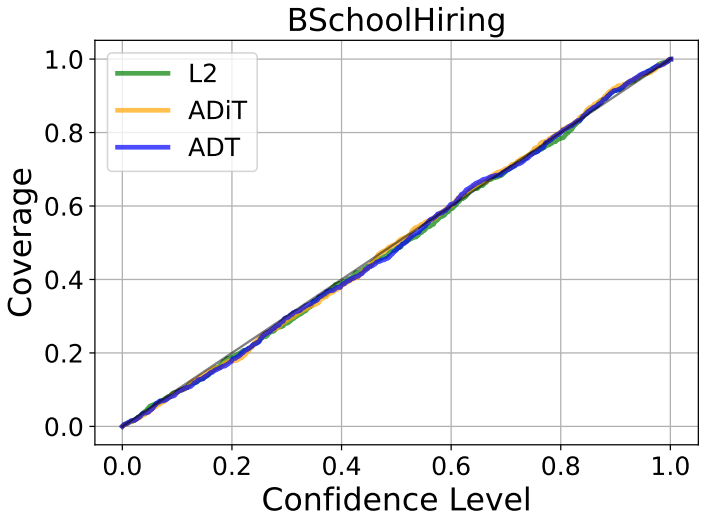

Figure A.2: Weighted SI model true confidence set coverage as a function of confidence level on the business school hiring network (undirected).

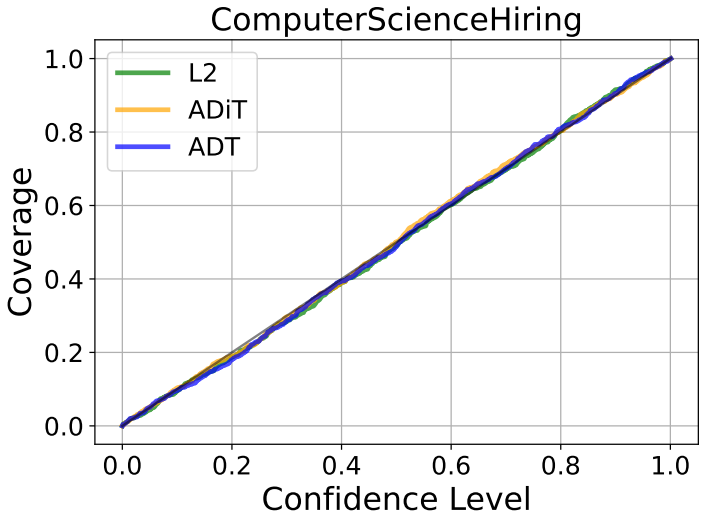

Figure A.3: Weighted SI model true confidence set coverage as a function of confidence level on the computer science hiring network (undirected).

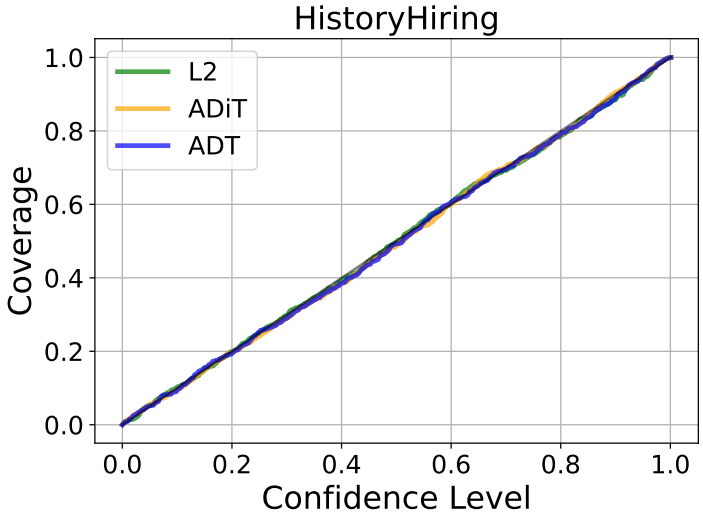

Figure A.4: Weighted SI model true confidence set coverage as a function of confidence level on the history hiring network (undirected).

Algorithm 6 Identification of First-Order Isomorphic Pairs

Input: Graph $G = (V, E)$
Initialize $L = \emptyset$ to store the list of isomorphic node pairs
for every node $u \in V$ **do**
  Compute $N_{1-4}(u)$ as the set of all neighbors of u within 4 hops
    // $N_{1-4}(u)$ includes all nodes that are possible to be isomorphic to u
  for $v \in N_{1-4}(u)$ **do**
    if $d_v == d_u$ **then**
      Compute $D_1(u) = \{d_{u'} : u' \in N_1(u)\}$, the multi-set of degrees of nodes in $N_1(u)$
      Similarly, compute $D_1(v)$, the multi-set of degrees of all one-hop neighbors of v
      if $D_1(u) == D_1(v)$ **then**
        Compute $\tilde{N}_2(u) = N_2(u) - N_1(u) - N_1(v) - \{u, v\}$
          // $\tilde{N}_2(u)$ contains all (exactly) two-hop neighbors of u, but with all (exactly) one-hop neighbors of u, v removed
        Compute $\tilde{N}_2(v) = N_2(v) - N_1(u) - N_1(v) - \{u, v\}$
        if $\tilde{N}_2(u) == \tilde{N}_2(v)$ **then**
          Do exhaustive search to check whether u, v are isomorphic by enumerating all possible matchings of their neighbors under the constraints of $\tilde{N}_2(u)$ and the matching $D_1(u), D_1(v)$, and if so, add (u, v) to list L
            // Usually, not many pairs need to go through this step
        end if
      end if
    end if
  end for
end for

Algorithm 7 Parallel Confidence Set Construction

Input: MC sample numbers m_l and m_p, confidence level α, Network G, data y, discrepancy function ℓ
Compute $S = \{g_1, g_2, \cdots, g_M\}$, the isomorphic groups for infected nodes with degree at least 2.
for each $g \in S$ **do**
    Extend g by including all of its single-degree neighbor.
end for
for each infected isomorphic group $g \in S$ **in parallel do**
    Select any $s \in g$ with degree at least 2
    Generate $m_p + m_l$ samples $z_i \in \mathcal{Z}, i = 1, \cdots, m_p + m_l$ from the T-step diffusion process from source s.
    Calculate the p-value for s following (3.6), (3.7), and (3.8)
    for each $v \in g$ that is isomorphic to s **do**
        Calculate $\hat{\psi}_v(y)$ according to Theorem 2.
    end for
    for each single-degree node $v \in g$ **do**
        Calculate the p-value $\hat{\psi}_v(y)$ according to the surjective importance sampling in Section 4.2.
    end for
end for
return the level $1 - \alpha$ confidence set:

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

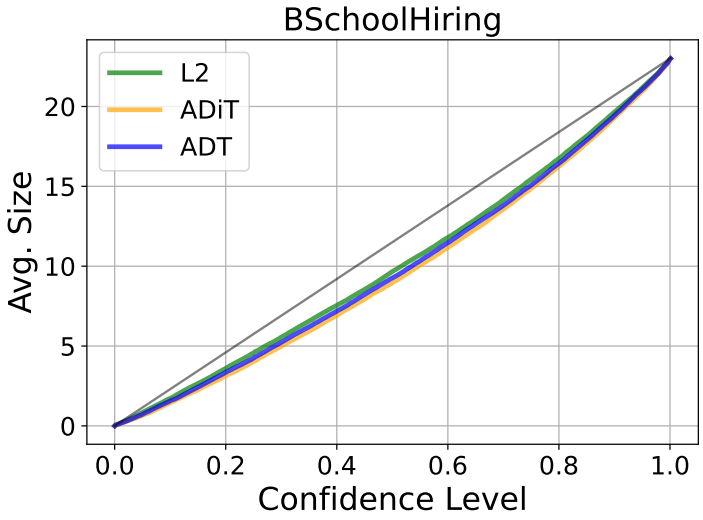

Figure A.5: Weighted SI model confidence set size as a function of confidence level on the business school hiring network (undirected).

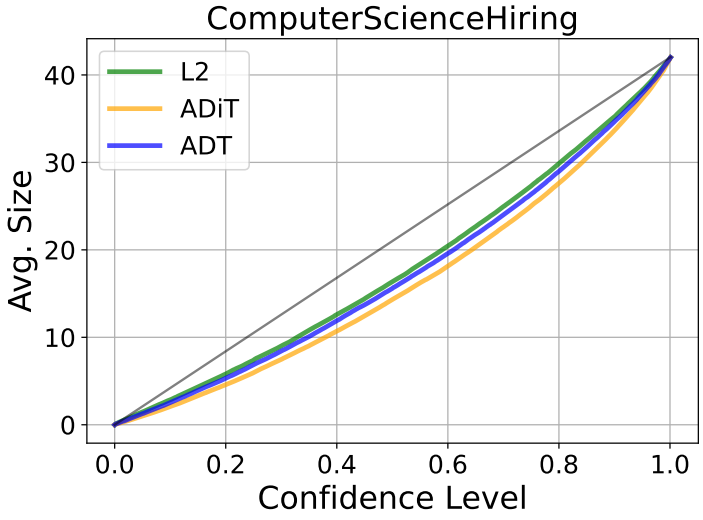

Figure A.6: Weighted SI model confidence set size as a function of confidence level on the computer science hiring network (undirected).

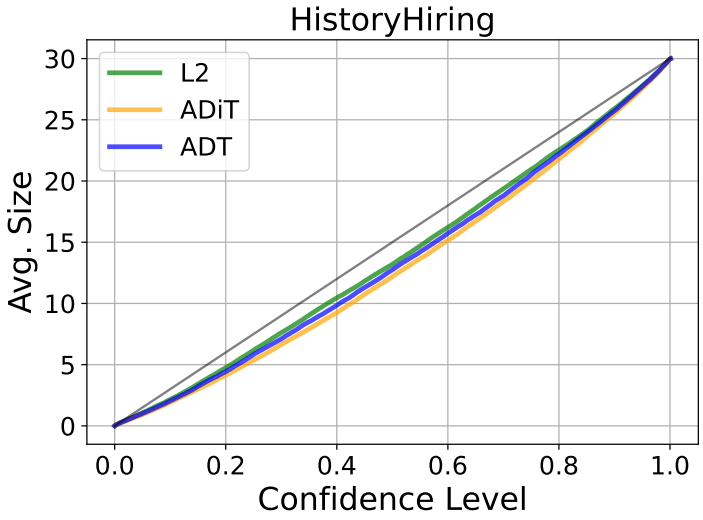

Figure A.7: Weighted SI model confidence set size as a function of confidence level on the history hiring network (undirected).

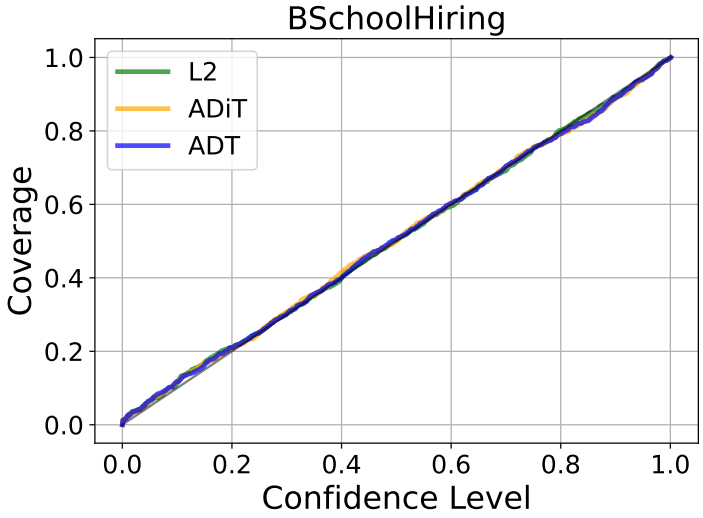

Figure A.8: Weighted SI model true confidence set coverage as a function of confidence level on the business school hiring network (directed).

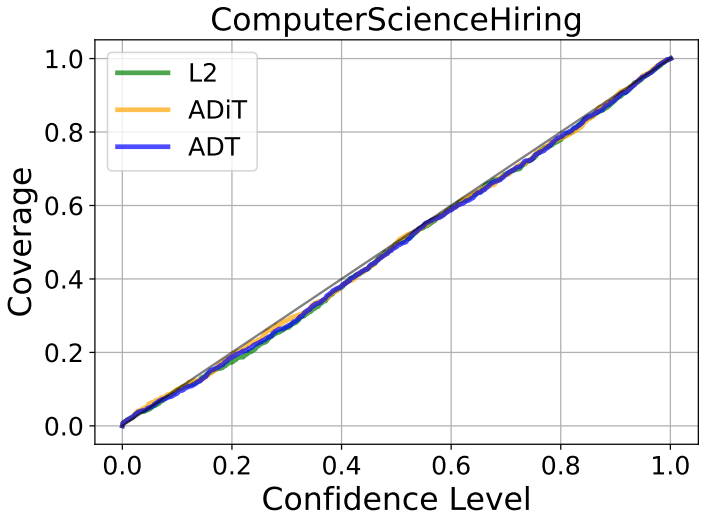

Figure A.9: Weighted SI model true confidence set coverage as a function of confidence level on the computer science hiring network (directed).

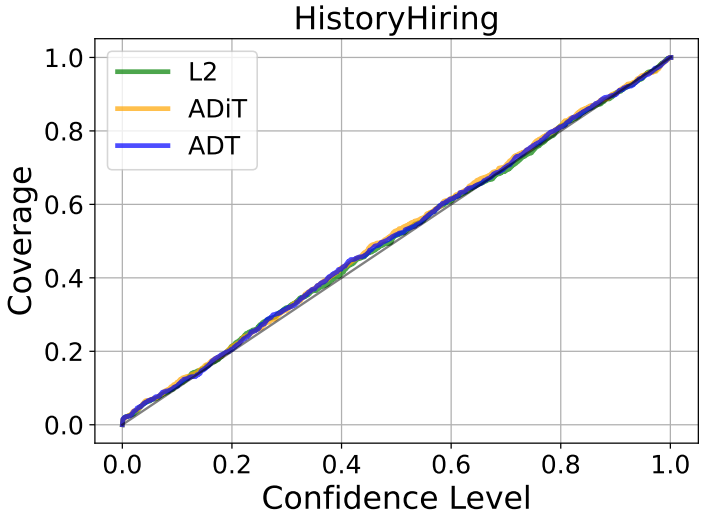

Figure A.10: Weighted SI model true confidence set coverage as a function of confidence level on the history hiring network (directed).

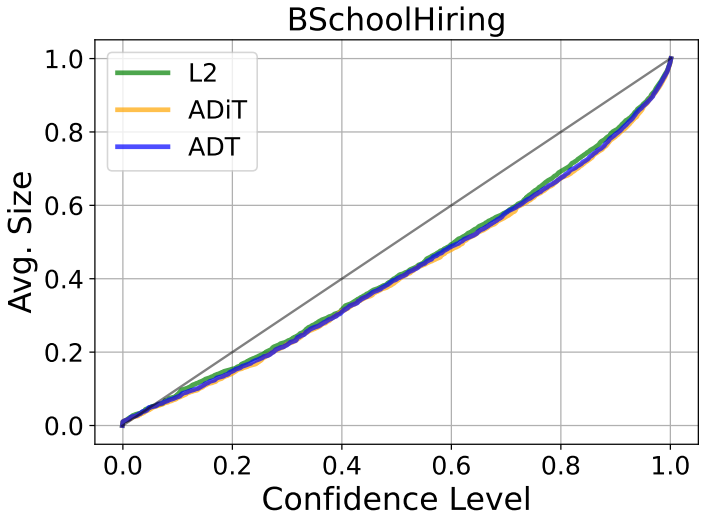

Figure A.11: Weighted SI model confidence set size as a function of confidence level on the business school hiring network (directed).

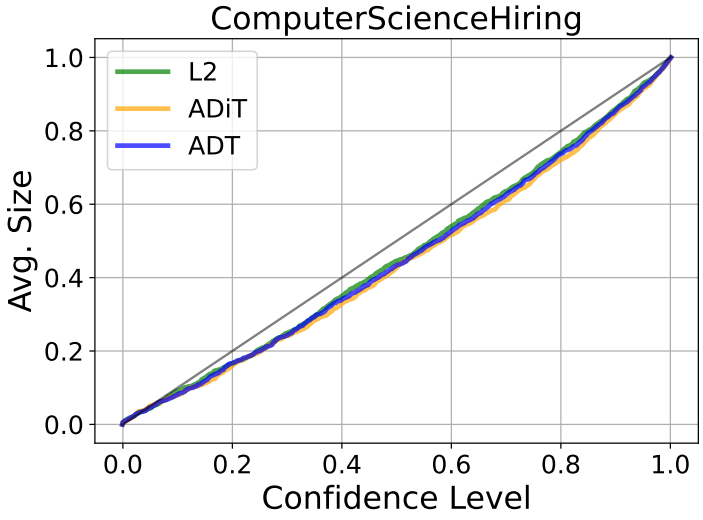

Figure A.12: Weighted SI model confidence set size as a function of confidence level on the computer science hiring network (directed).

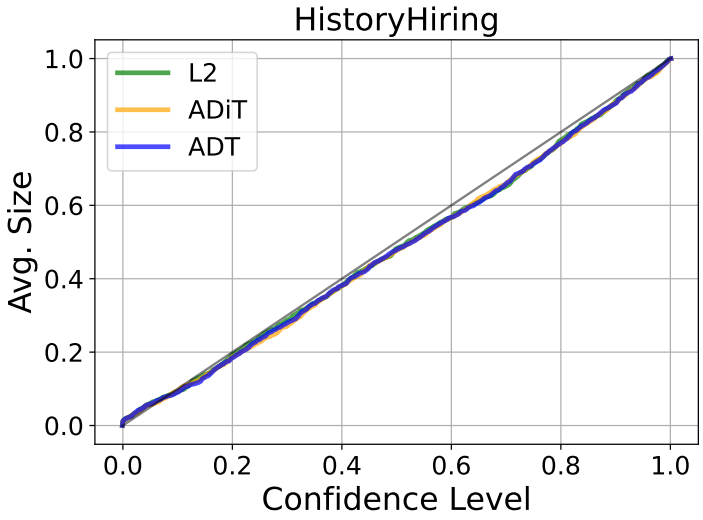

Figure A.13: Weighted SI model confidence set size as a function of confidence level on the history hiring network (directed).

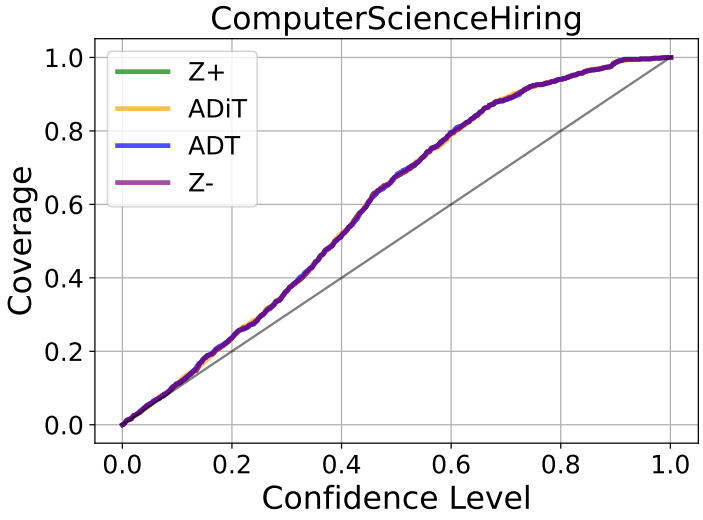

Figure A.14: IC model true confidence set coverage as a function of confidence level on the computer science hiring network (undirected).

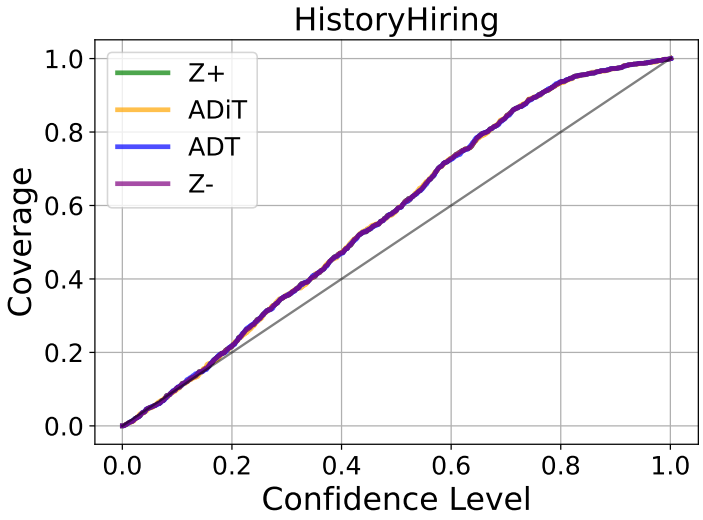

Figure A.15: IC model true confidence set coverage as a function of confidence level on the history hiring network (undirected).

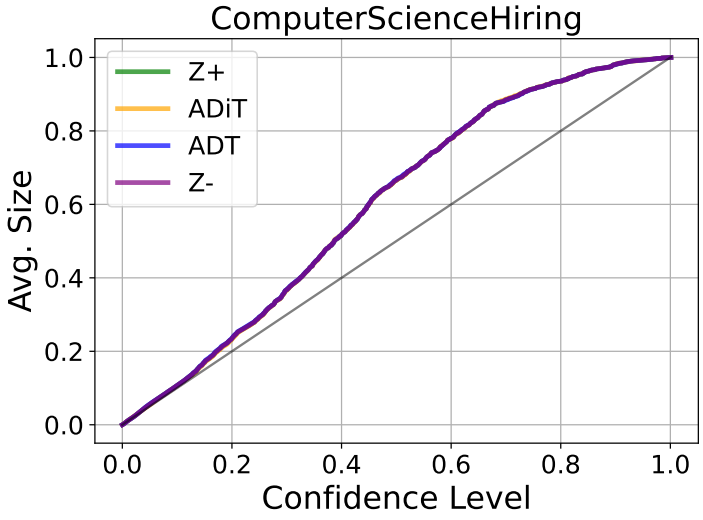

Figure A.16: IC model confidence set size as a function of confidence level on the computer science hiring network (undirected).

Figure A.17: IC model confidence set size as a function of confidence level on the history hiring network (undirected).

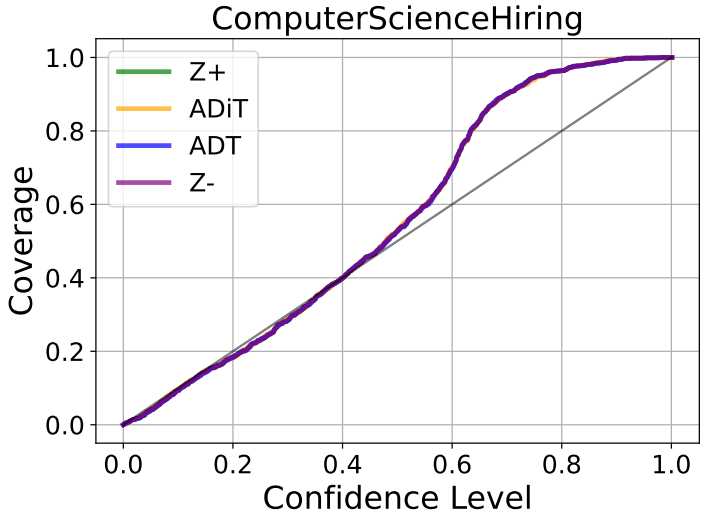

Figure A.18: LT model true confidence set coverage as a function of confidence level on the computer science hiring network (undirected).

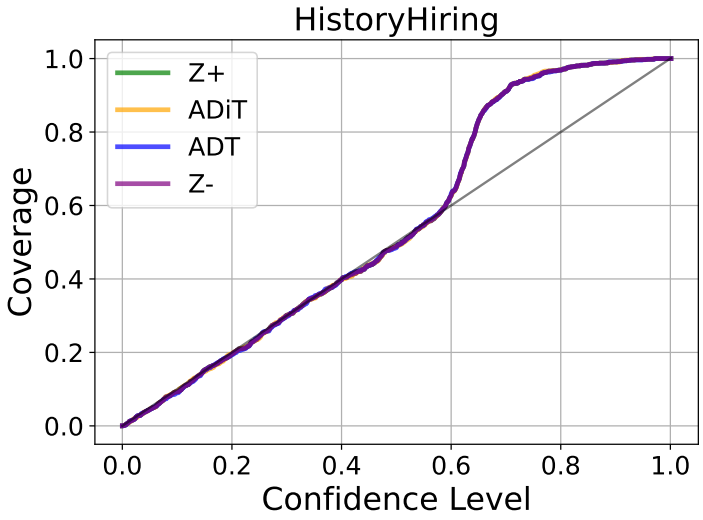

Figure A.19: LT model true confidence set coverage as a function of confidence level on the history hiring network (undirected).

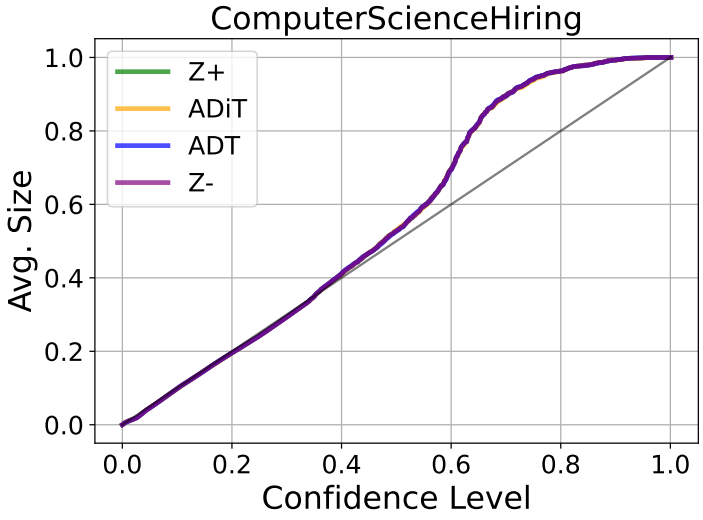

Figure A.20: LT model confidence set size as a function of confidence level on the computer science hiring network (undirected).

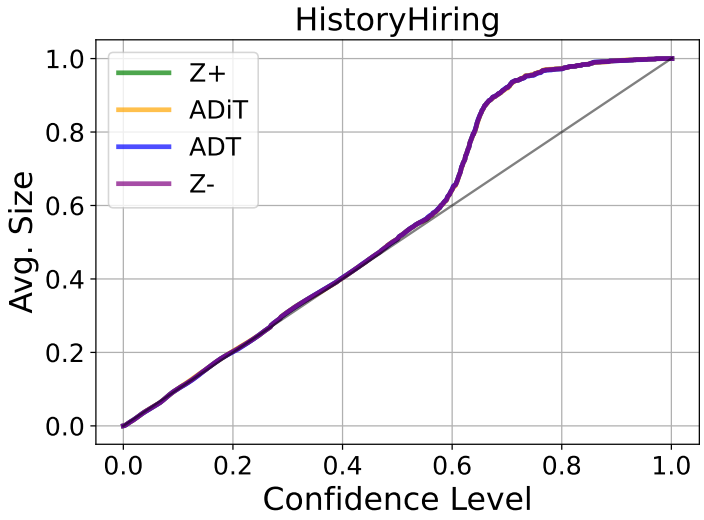

Figure A.21: LT model confidence set size as a function of confidence level on the history hiring network (undirected).

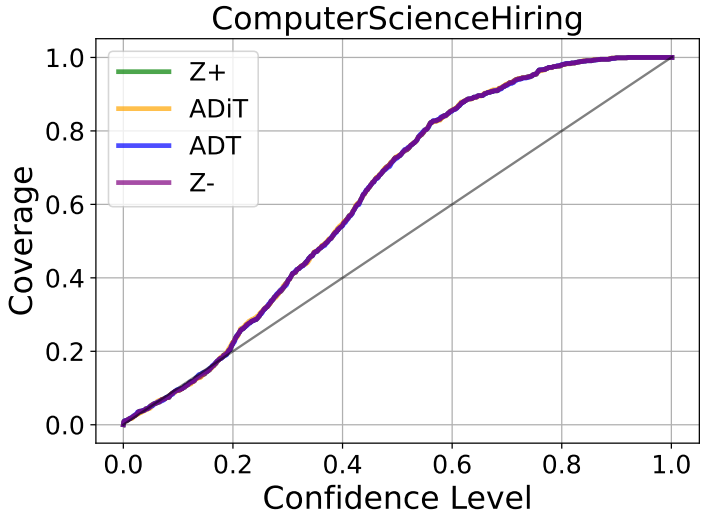

Figure A.22: IC model true confidence set coverage as a function of confidence level on the computer science hiring network (directed).

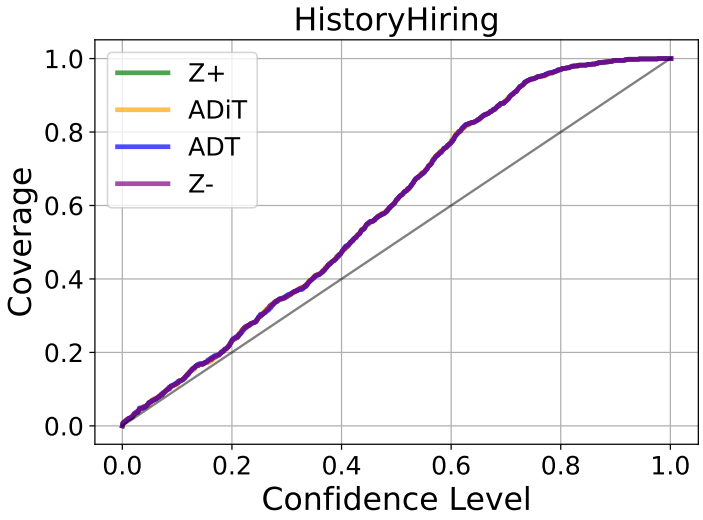

Figure A.23: IC model true confidence set coverage as a function of confidence level on the history hiring network (directed).

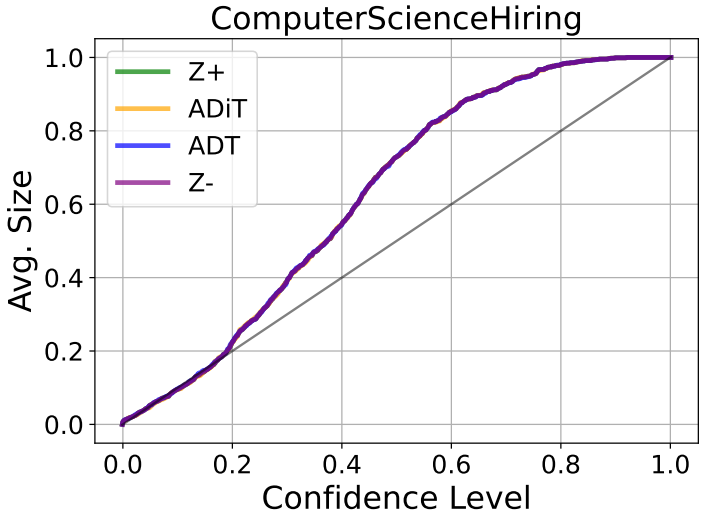

Figure A.24: IC model confidence set size as a function of confidence level on the computer science hiring network (directed).

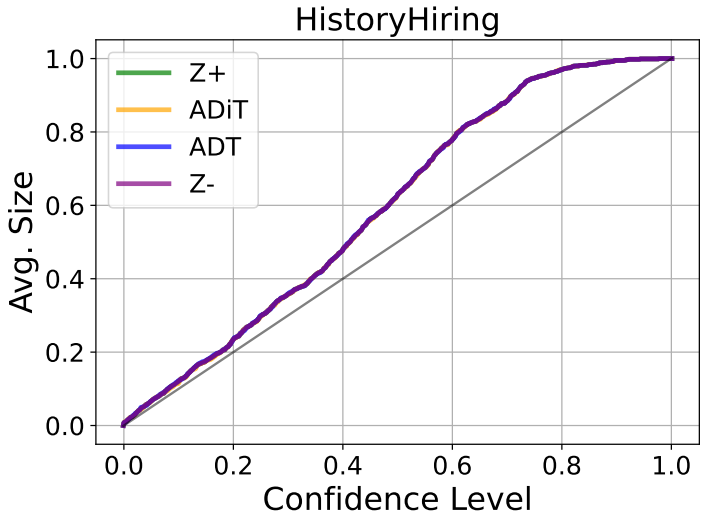

Figure A.25: IC model confidence set size as a function of confidence level on the history hiring network (directed).

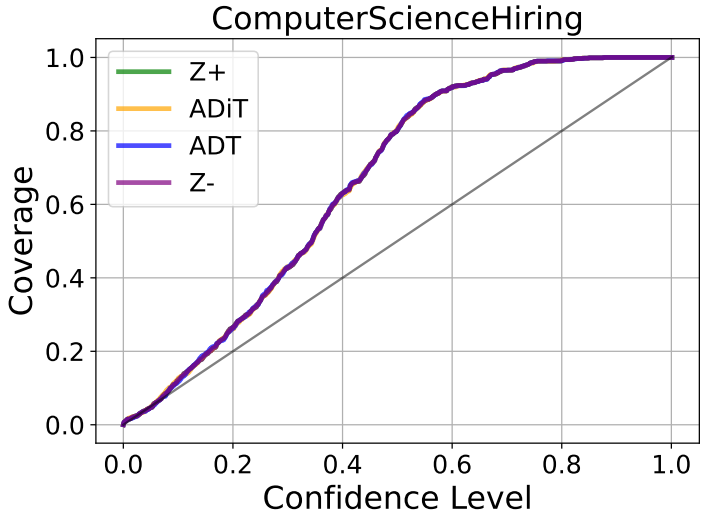

Figure A.26: LT model true confidence set coverage as a function of confidence level on the computer science hiring network (directed).

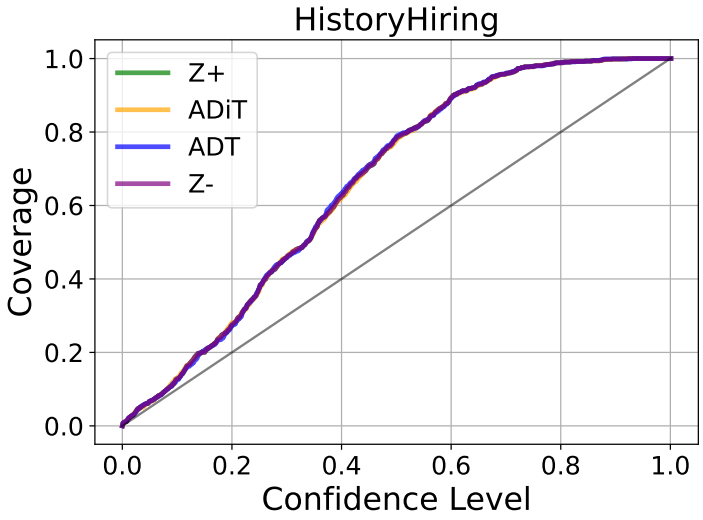

Figure A.27: LT model true confidence set coverage as a function of confidence level on the history hiring network (directed).

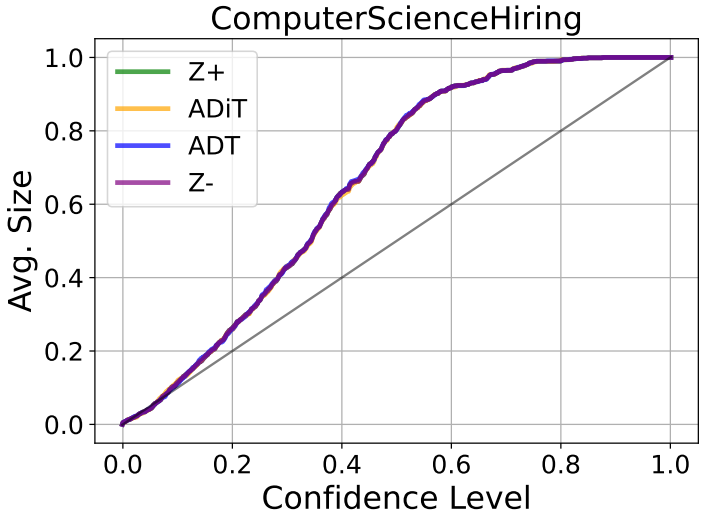

Figure A.28: LT model confidence set size as a function of confidence level on the computer science hiring network (directed).

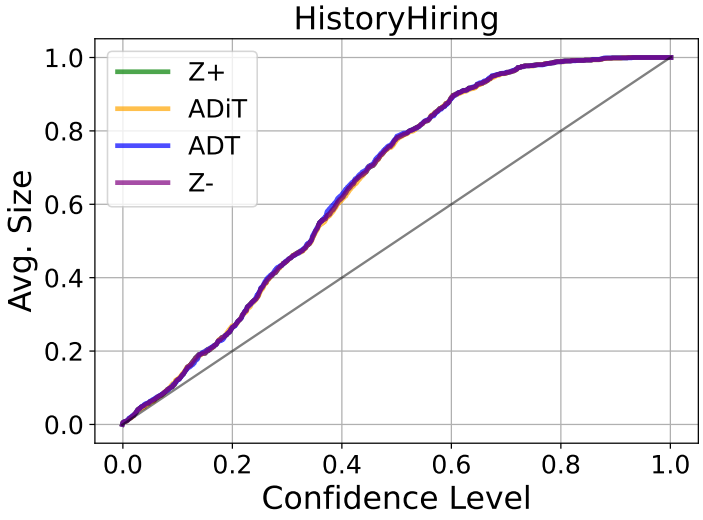

Figure A.29: LT model confidence set size as a function of confidence level on the history hiring network (directed).