Iterative Learning of Graph Connectivity from Partially-Observed Cascade Samples

Jiin Woo[1], Jungseul Ok[2], and Yung Yi[3]

[1]NAVER Corp
[2]POSTECH, AIGS/CSE
[3]KAIST, EE
ji.woo@navercorp.com, jungseul@postech.ac.kr, yiyung@kaist.ac.kr

Abstract

Graph learning is an inference problem of estimating connectivity of a graph from a collection of epidemic cascades, with many useful applications in the areas of online/offline social networks, p2p networks, computer security, and epidemiology. We consider a practical scenario when the information of cascade samples are partially observed in the independent cascade (IC) model. For the graph learning problem, we propose an efficient algorithm that solves a localized version of computationally-intractable maximum likelihood estimation through approximations in both temporal and spatial aspects. Our algorithm iterates the operations of recovering missing time logs and inferring graph connectivity, and thereby progressively improves the inference quality. We study the sample complexity, which is the number of required cascade samples to meet a given inference quality, and show that it is asymptotically close to a lower bound, thus near-order-optimal in terms of the number of nodes. We evaluate the performance of our algorithm using five real-world social networks, whose size ranges from 20 to 900, and demonstrate that our algorithm performs better than other competing algorithms in terms of accuracy while maintaining fast running time.

1 Introduction

Information spread is universal, where examples include propagation of infectious diseases, computer virus/spam infection in the Internet, technology diffusion, and tweeting/retweeting of popular topics. Inferring the "influential" connections of the underlying social network, referred to as *graph learning*, is important for understanding diffusion dynamics and controlling strategies for optimal dissemination or mitigation of spread in many applications, yet challenging due to restricted observation on the infection process. Cascade information such as infection time is often allowed to be logged for each node when the infection propagates through influential connections. There has been an extensive array of works which have studied the graph learning, e.g., [Gomez Rodriguez et al., 2010, Netrapalli and Sanghavi, 2012, Goyal et al., 2010, Pouget-Abadie and Horel, 2015, Daneshmand et al., 2014, Abrahao et al., 2013, Du et al., 2012], where they mainly consider the case when the observation is complete.

In practice, however, the observation is often incomplete because some nodes or individuals are reluctant to open their infection logs, or the logs are often partially stored due to its large scale. This incompleteness can produce non-trivial challenges to the graph learning since the order of infection can differ depending on the infection times of unobserved nodes and infection paths, which directly implies the influence of connections along the paths, become uncertain. Since the number of candidate infection paths to be considered increases, the loss of the observation not only degrades the confidence of the influential connections but also incurs a lot of computation cost. Figure 1 illustrates an example, where the incomplete information allows a much larger set of possible infection paths than the complete one. Hence, the connectivity inference with missing data clearly needs more computation and sample complexity than that with full observation.

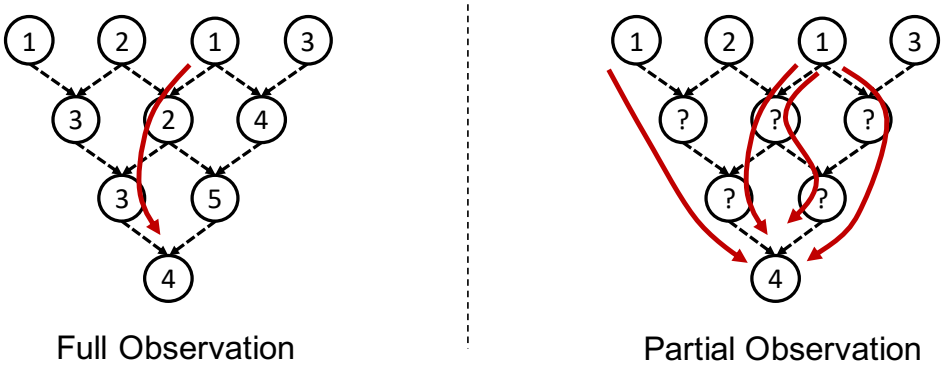


Figure 1: The numbers in each circle represent the time logs of infection. Missing time logs generate a variety of candidate infection scenarios under partial observation, while how infection occurs under full observation is easy to be inferred. The uncertainty of the infection paths degrades the confidence of influential connections.

In this paper, we propose a simple yet powerful graph learning algorithm, called Iterative Time Recovery and Edge Selection (`ITRES`), where the main idea is to approximate the likelihood into per-node localized functions, each of which can be optimized separately in a tractable manner. To infer the underlying true graph by maximizing these approximate likelihood functions, `ITRES` iterates two steps: (a) time-recovery, which recovers missing time logs using previously selected edges, and (b) edge selection, which selects influential connections using the recovered times. Through these iterative procedures improving the approximation step by step, `ITRES` progressively constructs the original graph connectivity. The time recovery in (a) helps to formulate the likelihood in a tractable form by ignoring uncertain and minor infection events but preserving important events by updating the estimated time logs of unobserved nodes with high confidence. The edge selection in (b) efficiently finds influential edges that maximize the likelihood with a simple greedy approach by considering the instant improvement of the likelihood. Despite its greedy behavior, the selected edges are guaranteed to be more influential than other edges mostly, which ensures `ITRES` to end up with inferring the true influence structure with high accuracy.

Other contributions of this paper include providing a theoretical analysis of `ITRES`, where we compute the number of cascade samples required to achieve $(1-\delta)$ probability of accurate estimation for any given $0 < \delta < 1$, referred to as *sample complexity*. We present a lower bound as a necessary sample complexity and prove that `ITRES` is asymptotically close to the lower bound in terms of the number of nodes. We then evaluate `ITRES` on five real-world graphs, whose size ranges from 20 to 900 nodes. We compare `ITRES` with one baseline algorithm proposed for the full observation scenario, `Greedy` [Netrapalli and Sanghavi, 2012], and two other competing algorithms proposed for the partial observation scenario, `NetInf` [Gomez Rodriguez et al., 2010] and `DMP` [Lokhov, 2016]. `ITRES` shows comparable or superior accuracy to the competing algorithms in short running times, which are four or five orders of magnitude faster than `DMP`. `ITRES` also provides the consistent performance in various observation scenarios even with imbalanced observation, while the others show significant performance degeneration in such unfavorable conditions.

1.1 Related Work

Many works have studied the graph inference problem which aims to recover the edges of influence from the observed infection time logs of cascades. Netrapalli and Sanghavi [2012] suggested a maximum likelihood estimator (MLE) and a greedy algorithm for recovering influential connections, which achieves an optimal cascade sample complexity for the IC model. Goyal et al. [2010] studied various diffusion models and developed an algorithm for learning influence parameters of the models. Pouget-Abadie and Horel [2015] investigated the graph learning from the sparse recovery perspective for a general discrete cascade model.

Daneshmand et al. [2014], Abrahao et al. [2013] have considered continuous-time cascade models for the graph learning problem. Kalimeris et al. [2018] showed that the sample complexity can be significantly reduced with hyperparameter assumptions on the cascade model. He and Liu [2017] proposed an algorithm that recovers a graph with limited cascade samples effectively by using commonality between highly correlated diffusion graphs.

There are several works which have studied other social network problem in various perspective when observed data is incomplete. He et al. [2016] have proposed algorithms inferring influence function with incomplete observations, Sun et al. [2017], Zong et al. [2012], Rozenshtein et al. [2016] have investigated the reconstruction of cascades from partial timestamps, and Zhu et al. [2016, 2017] have proposed an algorithm which locates diffusion sources when time logs are partially observed. Some researchers have studied the graph learning without the explicit time of infection. Gripon and Rabbat [2013] proposed an algorithm that recovers edges involved in the infection process from an unordered set of infected nodes. Amin et al. [2014] studied the problem of recovering connectivity only with initial seeds and finally-infected nodes for the IC model, without any time information of nodes. Although Gripon and Rabbat [2013] and Amin et al. [2014] have considered the scenarios when the observation is limited, they still assume complete observation on the infection status of all nodes.

The approaches proposed by Gomez Rodriguez et al. [2010], Wu et al. [2013], Lokhov [2016] are the closest to ours, where they consider the partial observation scenario when any hint of infection, even the infection status, is not observed for some nodes. Their algorithms recover the influence parameters of edges by maximizing the approximate likelihood for sparsely observed data. Gomez Rodriguez et al. [2010] suggested an approximate likelihood function, which is computed based on the most likely tracing diffusion trees from cascade logs, and proved that the greedy maximization of the function achieves near-optimal performance. Wu et al. [2013] investigated an expectation-maximization (EM) approach for the continuous independent cascade (CIC) model. Lokhov [2016] proposed an approximate gradient descent algorithm that efficiently finds the influence parameters with the gradients of the likelihood computed via mean-field-type approximation and dynamic message passing for the susceptible-infected (SI) model. The computation becomes tractable due to its alternative formulation, but it still suffers from slow convergence due to the complexity of the gradients.

In this paper, we suggest an algorithm that achieves both efficiency and high accuracy via proper approximation and greedy optimization. Also, we provide theoretical analysis, which proves its near-optimal sample complexity, as well as empirical results showing its superior accuracy and running time in various scenarios.

2 Model and Goal

Base and true graphs. Let a directed graph $G = (V, E)$ be a *base graph*, where V is a set of n nodes and E is a set of directed edges. Each edge $uv \in E$ is associated with a parameter $\theta_{uv} \in [0, 1]$, and represents a potential directed social relationship from node u to node v, where we say that node u (resp. node v) is a parent (resp. child) of node v (resp. node u). Let a subgraph $G^+ = (V, E^+)$ of G be a *true graph*, consisting of *influential* edge $uv \in E^+ \subset E$ only on which actual propagation can occur from u to v. We denote a set of parents of node v in G and G^+ by $V_v = \{u : uv \in E\}$ and $V_v^+ = \{u : uv \in E^+\}$, respectively.

Cascade model. As a cascade process, we consider the discrete-time independent cascade (IC) model [Kempe et al., 2003], which is one of the most popular diffusion models in the literature. In this model, each node can be in any of three states: susceptible, active, and inactive. At the initial time, some nodes (which we call seeds) independently become active in random with probability θ_0 and other nodes start from the susceptible state. If a node u is active at time t, then it can activate (infect) any susceptible child v in the true graph G^+, i.e., $uv \in E^+$, with probability $\theta_{uv} \in (0, 1]$ at the next time $t + 1$. The active nodes are being in the active state for only one time slot, and they become inactive at the next time. Once a node becomes inactive, it maintains the state until the end of the cascade process.

Observation structure. Let $\mathcal{C}$ be a set of C independent and identically distributed cascade samples, where we index by $k \in \mathcal{C} = \{1, 2, \dots, C\}$. Each cascade sample is represented as the collection of infection time logs

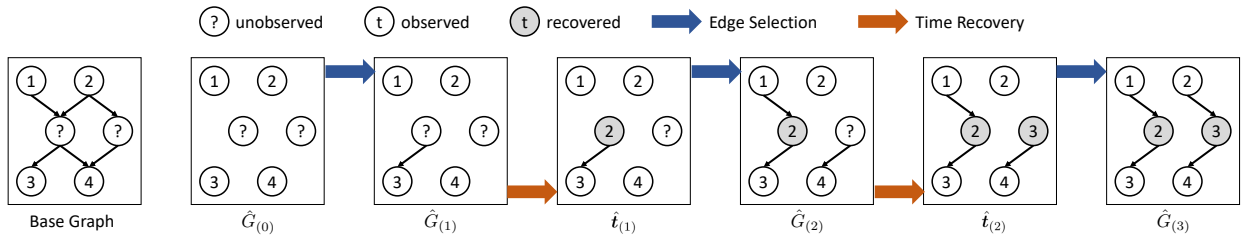


Figure 2: Algorithm overview with a simple example. Starting with an empty graph $\hat{G}_{(0)}$, our algorithm performs *time recovery* and *edge selection* alternatingly. The edge selection finds influential edges using observed/recovered time logs, and the time recovery recovers infection time logs using previously inferred edges. Finally, our algorithm outputs the true graph when it converges.

of all nodes, which we denote by $\boldsymbol{t}^k = \{t_v^k\}_{v \in V}$, where t_v^k is the infection time of v at the k-th cascade. Each infected node v has its time log $0 \leq t_v^k < \infty$ and we denote the time of susceptible nodes as ∞. Let O^k be a set of nodes whose time logs are observed for cascade k, and $\boldsymbol{t}_O^k = \{t_v^k\}_{v \in O^k}$ be the set of observed time logs for cascade k. We denote the total observed time logs as $\boldsymbol{t}_O^{\mathcal{C}} = (\boldsymbol{t}_O^k : k \in \mathcal{C})$.

Goal. In this paper, we focus on inferring the underlying true graph G^+ from the observations on the multiple cascades $\mathcal{C}$. We define the graph learning problem, where we aim to have the maximum likelihood estimator $\hat{G}$ from the following optimization problem:

$$\textbf{OPT:} \qquad \hat{G} = \operatorname*{arg\,max}_{G^+ \subseteq G} \ \mathbb{P}[\boldsymbol{t}_O^{\mathcal{C}} \mid G^+] \,. \tag{1}$$

The likelihood in (1) can be calculated by marginalizing out hidden time logs, denoted by $\boldsymbol{t}_H^{\mathcal{C}}$, as follows:

$$\mathbb{P}[\boldsymbol{t}_O^{\mathcal{C}} \mid G^+] = \textstyle\sum_{\boldsymbol{t}_H^{\mathcal{C}}} \mathbb{P}[\boldsymbol{t}_O^{\mathcal{C}}, \boldsymbol{t}_H^{\mathcal{C}} \mid G^+] \,, \tag{2}$$

where the summation is taken over all possible combinations of infection times for hidden logs. As the number of hidden logs increases, the marginalization requires exponentially many calculations and thus becomes intractable.

3 Algorithm

In this section, we present an efficient algorithm, called Iterative Time Recovery and Edge Selection (`ITRES`), which guarantees high inference accuracy while maintaining computational tractability through a smart approximation. We first present the rationale of `ITRES` (Section 3.1), followed by the algorithm description and the analysis of the sample complexity in Section 3.2 and Section 4, respectively.

3.1 Rationale

Our main idea is to reduce the complexity of the marginalization in (2) by approximating the intricate joint likelihood into a product of functions, each of which can be optimized separately by each node. Specifically, the approximation of the likelihood is temporally and spatially decomposed into per-node localized functions.

- **Temporal: Independence among time logs.** We first formulate the joint likelihood as the product of per-node marginals by applying the mean-field-type approach, which is similarly used in Lokhov [2016], as follows:

$$\mathbb{P}[\boldsymbol{t}_O^k \mid G^+] \approx \mathbb{P}[t_v^k \mid \boldsymbol{t}_{A_v}^k, G^+] \,, \tag{3}$$

where $A_v \subseteq O^k$ denotes a set of ancestor nodes who can reach to the node v through edges in G. Each of the marginals can be formulated in a simple form because it only takes infection events on its ancestors into account, and thus we can avoid the exhaustive computation over all possible realizations of the cascade. Note that this allows us to take account of all the possibilities although there are some redundant events.

- **Spatial: Only parents rather than all ancestors.** The per-node marginal probability (3) can require large computational cost in marginalizing out all missing logs on the entire ancestors. As an approximation of the marginalization, we focus on the parents, i.e., one-hop ancestors, recover their missing logs using the given graph structure and observations, and use them to obtain a closed-form approximation.
  To complete time values for the missing logs, we first construct a *feasible time set* for each unobserved node that contains all possible times of infection possibly propagated from its ancestors. Then, we narrow down the candidates by limiting possible paths of infection based on the true graph G^+. If the filtering process leaves a single feasible time for a node, we use it to complete the missing log of the node. Otherwise, we use the feasible time set to fill missing values leaving all possibilities open. To be specific, the infection time for each node s is characterized as follows:

$$\hat{\boldsymbol{t}}_s^k = \begin{cases} \{t_s^k\} \text{ s.t. } t_s^k \in \boldsymbol{t}_O^k & \text{if } s \in O^k \\ \{t_s^k : \mathbb{P}[t_s^k \mid \boldsymbol{t}_O^k, G^+] = 1\} & \text{if } s \in R^k \\ \{t_s^k : \mathbb{P}[t_s^k \mid \boldsymbol{t}_{A_s}^k] > 0\} & \text{otherwise.} \end{cases}$$

  where R^k is the set of nodes whose logs are solely characterized by the filtering process. Let $\hat{\boldsymbol{t}}^k = \{\hat{\boldsymbol{t}}_s^k\}_{s \in V}$ and $\hat{\boldsymbol{t}} = \{\hat{\boldsymbol{t}}^k\}_{k \in \mathcal{C}}$. With these conjectured times, we approximate the per-node likelihood for each recovered or observed node v as:

$$L_v(\hat{\boldsymbol{t}}^k; V_v^+) := \sum_{\{t_s^k\}_{s \in V_v^+ \cup \{v\}} : t_s^k \in \hat{\boldsymbol{t}}_s^k} (1 - \prod_{u \in V_v^+ : t_u^k = t_v^k - 1} (1 - \theta_{uv})) \cdot \prod_{w \in V_v^+ : t_w^k < t_v^k - 1} (1 - \theta_{wv}) \,.$$

The above ideas lead us to the following decomposed approximate log-likelihood function from **OPT** in (1):

$$\textbf{DEC-OPT:} \quad \hat{G} = \operatorname*{arg\,max}_{G^+ \subseteq G} \sum_{v \in V} M_v(\hat{\boldsymbol{t}}; V_v^+) \,, \tag{4}$$

where

$$M_v(\hat{\boldsymbol{t}}; V_v^+) = \sum_{k \in C} \mathbb{1}_{[v \in O^k \cup R^k]} \log L_v(\hat{\boldsymbol{t}}^k; V_v^+) \,,$$

and we propose an algorithm that aims at solving the decomposed optimization of **DEC-OPT**.

3.2 ITRES: Description

Overview. To solve **DEC-OPT**, the times $\hat{\boldsymbol{t}}$ need to be filled based on the previously inferred graph $\hat{G}$ and the graph is estimated based on $\hat{\boldsymbol{t}}$ by separately maximizing $M_v(\hat{\boldsymbol{t}})$ for each node v. Hence, we propose ITRES (Iterative Time Recovery and Edge Selection) which iterates over multiple phases, where each phase consists of two steps: (a) *time recovery* and (b) *edge selection*, which updates the inference on times and accordingly select edges based on those inferred times, finally resulting in an estimated $\hat{G}$ close to the true graph G^+. The algorithm is sketched in what follows (see Appendix A for the full description):

ITRES (Iterative Time Recovery and Edge Selection)

Input: base graph $G = (V, E)$, observed infection times $\boldsymbol{t}_O^{\mathcal{C}}$
Output: inferred graph $\hat{G} = (V, \hat{E})$

Initialize: $\hat{G}_0$ consisting of the set of nodes V and empty edge set, i.e., $\hat{G}_0 = (V, \emptyset)$.
For each phase $i = 1, 2, \ldots$, the following two steps, which we denote by two functions ϕ and π, respectively, are sequentially executed. The algorithm stops and outputs $\hat{G}_{(i)}$ as a final inferred graph $\hat{G}$ until when there is no more change between $\hat{G}_{(i)}$ and $\hat{G}_{(i+1)}$:

- **Time recovery step.** It completes the infection time $\hat{\boldsymbol{t}}_{(i)}$ by inferring times from the observations $\boldsymbol{t}_O^{\mathcal{C}}$ and the inferred graph $\hat{G}_{(i-1)}$:
$$\hat{\boldsymbol{t}}_{(i)} = \phi\left(\hat{G}_{(i-1)}; G, \boldsymbol{t}_O^{\mathcal{C}}\right) ,$$
where $\hat{G}_{(i-1)} \subseteq G$ consists of the inferred edges at the $i-1$-th phase.
- **Edge selection step.** It infers the graph connectivity $\hat{G}_{(i)}$ by greedily maximizing M_v calculated based on the $\hat{\boldsymbol{t}}_{(i)}$ for each node v.:
$$\hat{G}_{(i)} = \pi\left(\hat{\boldsymbol{t}}_{(i)}; G\right) .$$

As the iteration continues, the time recovery effectively finds highly probable times for the unobserved nodes, which provides useful information to the next edge selection step. As more phases are taken, more accurate estimation becomes possible. We now elaborate on the time recovery and edge selection steps.

Time recovery step. In this step, we recover the infection times for some hidden logs using the inferred $\hat{G}_{(i-1)}$ in the previous phase. As we fill in times for the missing logs more and more, the uncertainty on infection paths is removed and it helps with inferring the true graph. However, the inference on unobserved data should be made carefully, so that we do not miss non-negligible candidate times.

We now explain how to find a highly probable time for each hidden node in a cascade k. First, we construct a feasible time set, which contains all possible infection times, for each node by propagating the times of its observed ancestor nodes. Any observed node v initializes the feasible time set as $\{t_v^k\}$ and other unobserved nodes start from an empty set. Then, for each t starting from $t = 1$, the unobserved nodes append t to their set, if they have a parent node $u \in V_v$ having $t - 1$ in its feasible time set. Finally, the feasible time sets of unobserved nodes consist of times of infection possibly propagated from their ancestors. The feasible time set is constructed independently of $\hat{G}_{(i-1)}$, and thus it can be reused over the iteration of `ITRES`.

We then select a highly probable time from the feasible time set using the previously inferred graph $\hat{G}_{(i-1)}$. Starting from an observed node v whose infection time is t_v^k, if there exists a node $u \in \hat{V}_v$ whose feasible time set contains $t_v^k - 1$, then u may be the possible infector of v. However, if there are no other possible infectors among $\hat{V}_v \setminus \{u\}$, then we may infer that u is an actual infector of v and u must be infected at $t_v^k - 1$. Thus, we recover the infection time of u as $t_v^k - 1$. In the example of Figure 3, the observed node a has only one candidate infector, the node b, since it is the only node who is infected right before the infection of a. Then, we recover the infection time of b as $t_b = 2$. Not only the observed nodes but unobserved nodes whose times are recovered, such as b, can be used for the time recovery of their parents. As shown in Figure 3, the node d becomes the only possible infector of b, which seems to be infected at 2, and we recover the time of d as $t_d = 1$. When more than two values are in conflict, or there is no reachable observed log via $\hat{G}_{(i-1)}$, we just set the infection time of the node as its feasible time set in $\hat{\boldsymbol{t}}_{(i)}$. In this manner, we narrow down the candidate times and finally complete $\hat{\boldsymbol{t}}_{(i)}$ according to the given structure of $\hat{G}_{(i-1)}$.

Edge selection step. This step aims to find true in-degree edges node-wisely solving (4) formulated based on the previously inferred times $\hat{\boldsymbol{t}}_{(i)}$ as follows:

$$\hat{V}_v = \underset{V_v^+ \subseteq V_v}{\arg\max}\ M_v(\hat{\boldsymbol{t}}_{(i)}; V_v^+) . \tag{5}$$

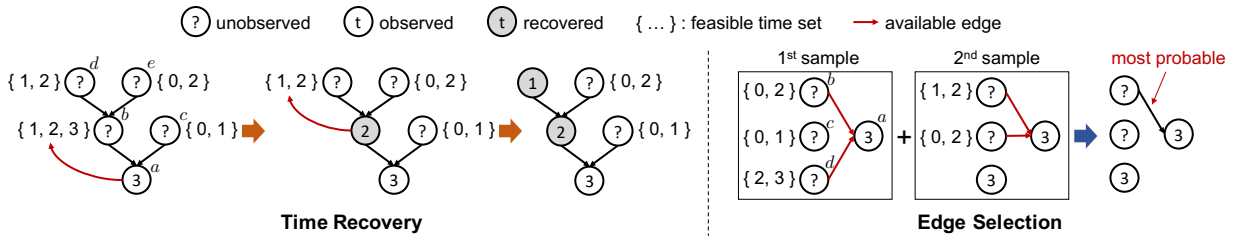


Figure 3: In the time recovery step, ITRES recovers the infection time of unobserved nodes from the observed/recovered neighbor nodes in the previously inferred graph. In the edge selection step, ITRES greedily adds edges which improves the approximate likelihood the most for each node.

To do so, we take a greedy approach that starts with an empty set $\hat{V}_v$ and iteratively adds a node $\hat{u}$ into $\hat{V}_v$ maximizing the instant improvement of the approximate as follows:

$$\hat{u} = \underset{u \in V_v \setminus \hat{V}_v}{\arg\max}\, M_v(\hat{\boldsymbol{t}}_{(i)}; \hat{V}_v \cup \{u\}) - M_v(\hat{\boldsymbol{t}}_{(i)}; \hat{V}_v)\,. \tag{6}$$

However, the calculation of M_v requires the knowledge of θ that is not available in practice. Hence, we further approximate (6) and find a parent $\hat{u}$ such that

$$\hat{u} = \underset{u \in V_v \setminus \hat{V}_v}{\arg\max} \sum_{k \in \mathcal{C}} x_{uv}^k \prod_{z \in \hat{V}_v} (1 - x_{zv}^k)\,, \tag{7}$$

where we define x_{uv}^k as an indicator such that $x_{uv}^k = 1$ if v is observed or recovered and $t_v^k - 1 \in \hat{\boldsymbol{t}}_u^k$, $x_{uv}^k = 0$ otherwise. The optimization in (7) implies that simply adding a node that seems to be infected right before v most frequently increases the likelihood the most. If the maximum value in (7) becomes zero for all remaining nodes in $V_v \setminus \hat{V}_v$, then the iteration ends and outputs $\hat{V}_v$.

We now explain the connection from (6) to (7). For a cascade k, if none of the nodes in the current estimated parents $\hat{V}_v$ is possible to be infected at $t_v^k - 1$, i.e., $\prod_{z \in \hat{V}_v}(1 - x_{zv}^k) = 1$, then $L_v(\hat{\boldsymbol{t}}_{(i)}^k; \hat{V}_v) = 0$. In this case, the improvement of the log-likelihood M_v by adding u to $\hat{V}_v$ is significantly large when $x_{uv}^k = 1$ since likelihood $L_v(\hat{\boldsymbol{t}}_{(i)}^k; V_v^+ \cup \{u\})$ becomes positive from zero with the addition of u. However, if there already exists a possible infector in the current parents, i.e., $\prod_{z \in V_v^+}(1 - x_{zv}^k) = 0$, then $L_v(\hat{\boldsymbol{t}}_{(i)}^k; \hat{V}_v)$ is positive and the likelihood improvement by u is relatively small regardless of x_{uv}^k. Since the log-likelihood improvement is overwhelmingly large when likelihood changes from zero to some positive constant, adding z who has the largest count of $x_{uv}^k \prod_{z \in V_v^+}(1 - x_{zv}^k)$ over all cascades among parent nodes in $V_v \setminus \hat{V}_v$ improves the likelihood the most.

Running time. To construct the feasible time set of a single hidden node for each cascade, it iterates over in-degree links of the node and the feasible time sets of its parents, which is bounded by $|V|$. Accordingly, constructing the entire feasible time sets over every hidden node for each cascade in $\mathcal{C}$ requires $O(|\mathcal{C}||E||V|)$. Similarly, to recover an infection time of a single hidden node for a cascade, it iterates over out-degree links of the node and the recovered or observed times of its children, which is bounded by $|V|$. Hence, the time recovery over every hidden node for every cascade requires $O(|\mathcal{C}||E||V|)$. In the edge selection step, to select true in-degree edges of a single node, it iterates over every cascade in $\mathcal{C}$ to calculate (7) for its parents, which is $|V|$ at maximum. Therefore, the edge selection phase requires $O(|\mathcal{C}||E||V|)$. Thus, a single iteration of ITRES requires $O(|\mathcal{C}||E||V|)$, while the original likelihood estimation in **OPT** requires exponentially many operations.

4 Analysis

A popular criterion of evaluating a graph inference algorithm is to analytically quantify the sample complexity, i.e., the number of cascades $C(n) = |\mathcal{C}|$ required to achieve a given target inference quality $0 < 1 - \delta < 1$ (equivalently the error probability δ). The inference quality is quantified by $Q(\hat{G}) := \mathbb{P}[\hat{G}(\boldsymbol{t}_O^{\mathcal{C}}) = G^+]$, where $\hat{G}$ is an estimation of the true graph. In our analysis, we make the following assumptions:

A1. The base graph G has D-regular in-degrees, and the true graph G^+ has d-regular in-degrees with homogeneous infection probability $\theta > 0$, where we denote the constant $\eta = d\theta$ to represent the strength of propagation.

A2. For each cascade, each node in G independently reveals its infection time log with probability r, which we call *revelation probability*. The revelation probability models the degree of observation in practice, where nodes are often unaware of their infection (e.g. healthy carrier of an epidemic), or reluctant to open their logs and thus hides some information. Seeds are known regardless of revelation events for each cascade.

The above assumptions are for mathematical tractability because too general models make it challenging to obtain meaningful results. However, we believe that our analytical results are still highly valuable to study the performance of inference algorithms. We refer the readers to Section 5 for more practical and general scenarios over real-world graphs, e.g., heterogeneous revelation probabilities.

4.1 Sample Complexity Analysis

Sufficient sample complexity. We present our analysis of the sufficient sample complexity achieved by ITRES in Theorem 4.1. We introduce a constant T, the maximum diffusion time in cascades, which represents the limit of cascade length due to the time budget for observation or the underlying graph structure.

Theorem 4.1. *For a given base graph $G = (V, E)$, assume A1 and A2. Suppose that an induced subgraph of G with nodes which can reach to v within T hops is a directed acyclic tree rooted to v for every $v \in V$, $d \geq 2$, and $\theta_0 < \frac{D}{64dn}$. Then, for any $\delta \in (0, 1)$,* ITRES *obtains $Q(\hat{G}) \geq 1 - \delta$ if*

$$C(n) > K_1 \frac{d \log \frac{nD}{\delta}}{r\theta_0 \alpha(\eta)}, \tag{8}$$

where $\alpha(\eta) = \sum_{i=1}^{T} \eta^i$ and K_1 is some constant independent of all parameters.

The proof is presented in Section 4.2. Theorem 4.1 states that ITRES requires $O(\frac{d}{r} \log n)$ samples when nodes reveal their infection times with probability $0 < r \leq 1$. Under full observation, i.e., $r = 1$, this requires $O(d \log n)$ samples, which is proved as the optimal sample complexity by Netrapalli and Sanghavi [2012]. Under partial observation, i.e., $r < 1$, ITRES requires $\frac{1}{r}$ times more samples to achieve the same performance achieved under full observation. When we interpret this from the perspective of cost, assuming that a constant cost is incurred to make one node reveals its time log, $\Theta(rn)$ is the expected cost for a single cascade sample and the expected total observation cost becomes $O(dn \log n)$. Interestingly enough, this expected cost is equivalent to that under full observation $O(d \log n) \times \Theta(n) = O(dn \log n)$, and thus we see that ITRES requires the same revelation cost as that of the inference under full observation, which means that the incompleteness does not degrade performance if the same amount of data is given.

Necessary sample complexity. We now consider a lower bound of the sample complexity, which is necessarily required by any algorithm to achieve $(1 - \delta)$ inference quality.

Theorem 4.2. *For a given base graph $G = (V, E)$, assume A1 and A2. For any graph inference algorithm $\hat{G}$ and $\theta_0 \leq \frac{1}{6 \max(\eta^T, 1)}$, the necessary number of cascade samples to achieve $Q(\hat{G}) \geq 1 - \delta$ for any $\delta \in (0, 1)$ is the following:*

$$C(n) \geq K_2 \frac{(1 - \delta) d \log \frac{D}{d}}{r\beta(\eta, \theta_0)}, \tag{9}$$

where $\beta(\eta, \theta_0) = \theta_0(1 + \sum_{i=1}^{T} \eta^i (2i \log \frac{1}{\eta} + 2 \log \frac{1}{\theta_0} + 1))$ *and* K_2 *is some constant independent of all parameters.*

The proof is presented in Section 4.2. We see that the asymptotic order of the necessary sample complexity is $\Omega(\frac{d}{r} \log \frac{D}{d})$ for a given revelation degree r, which is close to the sufficient sample complexity in (8) when $D = \Theta(n)$ and d is a constant independent of other parameters. This implies that the performance of `ITRES` is order-wise optimal when the base graph is nearly a complete graph, i.e., negligible prior information about candidate edges, but the true graph is sparse. Note that our lower bound reproduces the result of Netrapalli and Sanghavi [2012] with the correlation decay and full observation, i.e., $\eta < 1$ and $r = 1$, respectively.

4.2 Proofs of Theorems

Proof of Theorem 4.1. We will show that

$$\mathbb{P}[\hat{V}_v = V_v^+] > 1 - \frac{\delta}{n} \tag{10}$$

for any $v \in V$ if cascade samples satisfying (8) are given since we can conclude the desired result using the union bound as follows:

$$\mathbb{P}[G^+ = \hat{G}] \geq 1 - \sum_{v \in V} \mathbb{P}[\hat{V}_v \neq V_v^+] > 1 - \delta \,.$$

The proof follows a similar approach to that of Netrapalli and Sanghavi [2012]. Fix an arbitrary node $v \in V$. Let $\hat{v}_m$ denote the node selected at the m-th iteration of the edge selection phase for $\hat{V}_v$. Allowing abuse of notation for simplicity, let $\hat{V}_m$ denotes the intermediate $\hat{V}_v$ at the end of m-th iteration with $\hat{V}_0 = \emptyset$, i.e., $\hat{V}_m := \{\hat{v}_1, ..., \hat{v}_m\}$.

The iteration will stop once all true parents are added into the estimated parent set, i.e., $V_v^+ \subseteq \hat{V}_m$, because there must be at least one true parent who actually infected v and has $t_v - 1$ in its feasible time set if v is infected. Hence, we write

$$\mathbb{P}[\hat{V}_v = V_v^+] = \mathbb{P}\left[\bigcap_{m=1}^{|V_v^+|} \{\hat{v}_m \in V_v^+\}\right] .$$

For any m, a true parent is added to $\hat{V}_v$, i.e., $\hat{v}_m \in V_v^+$, if there exists $u = \hat{v}_m \in V_v^+ \setminus \hat{V}_{m-1}$ such that

$$\sum_{k \in \mathcal{C}} x_{wv}^k \prod_{z \in \hat{V}_m} (1 - x_{zv}^k) < \sum_{k \in \mathcal{C}} x_{uv}^k \prod_{z \in \hat{V}_m} (1 - x_{zv}^k)$$

for all $w \in V_v \setminus V_v^+$. We can bound the expressions as follows:

$$x_{uv}^k \prod_{z \in \hat{V}_m} (1 - x_{zv}^k) \geq x_{uv}^k \prod_{z \in V_v^+ \setminus \{u\}} (1 - x_{zv}^k) \coloneqq y_{uv}^k \,,$$
$$x_{wv}^k \prod_{z \in \hat{V}_m} (1 - x_{zv}^k) \leq x_{wv}^k \,.$$

Then, for the proof of (10), it is sufficient to show that

$$\mathbb{P}\left[\sum_{k \in \mathcal{C}} x_{wv}^k < \mu < \sum_{k \in \mathcal{C}} y_{uv}^k \;\forall u \in V_v^+, \;\forall w \in V_v \setminus V_v^+\right] \geq 1 - \frac{\delta}{n} \tag{11}$$

for some $\mu > 0$. Using the union bound, we can obtain (11) by showing that

$$\mathbb{P}\left[\sum_{k \in \mathcal{C}} y_{uv}^k < \mu\right] < \frac{\delta}{nD} \text{ and } \mathbb{P}\left[\sum_{k \in \mathcal{C}} x_{wv}^k > \mu\right] < \frac{\delta}{nD} \tag{12}$$

for any $u \in V_v^+$ and $w \in V_v \setminus V_v^+$. Now, we omit k when we describe the random variables for one cascade, unless confusion arises since cascade events are identically and independently distributed. If there is some $\lambda > 0$ which satisfies

$$\mathbb{P}[y_{uv} = 1] > \lambda \;\text{ and }\; \mathbb{P}[x_{wv} = 1] < \frac{\lambda}{4}\,, \tag{13}$$

then we can derive (12) by applying the Chernoff bound Mitzenmacher and Upfal [2005] with the condition that $\mu = \frac{1}{2}\lambda|\mathcal{C}|$ and $|\mathcal{C}| > \frac{2\log\frac{nD}{\delta}}{(1-\log 2)\lambda}$.

Thus, we can get the sufficient number of cascades for (11) by finding a proper value for λ satisfying (13). To do so, we provide Lemma 4.1 of which proof is provided in Appendix B.

Lemma 4.1. *Under the assumptions A1 and A2, suppose that an induced subgraph of G with nodes which can reach to v within T hops is a directed acyclic tree rooted to v for every $v \in V$, $d \geq 2$, and $\theta_0 < \frac{D}{64dn}$. Then, for any $v \in V$, $u \in V_v^+$, and $w \notin V_v^+$,*

$$\mathbb{P}[y_{uv} = 1] > \frac{r\theta_0}{8d} \sum_{0<t\leq T} (\theta d)^t\,, \tag{14}$$

$$\mathbb{P}[x_{wv} = 1] < \frac{r\theta_0}{32d} \sum_{0<t\leq T} (\theta d)^t\,. \tag{15}$$

Using Lemma 4.1, we find $\lambda = \frac{r\theta_0}{8d}\sum_{0<t\leq T}(\theta d)^t$. We then have proven that (11) holds if the number of cascade samples $|\mathcal{C}|$ satisfies

$$|\mathcal{C}| > \frac{16}{1-\log 2}\frac{d\log\frac{nD}{\delta}}{r\theta_0\sum_{i=1}^{T}\eta^i}\,,$$

where $\eta = d\theta$. This completes the proof. □

Proof of Theorem 4.2. The estimation of G^+ can be interpreted as an information theoretic problem to recover G^+ from the observation $\boldsymbol{t}_O^{\mathcal{C}}$ in a noisy channel such that

$$G^+ \to \boldsymbol{t}_O^{\mathcal{C}} \to \hat{G}(\boldsymbol{t}_O^{\mathcal{C}})\,. \tag{16}$$

Suppose G^+ is chosen uniformly at random from a fixed collection of graphs $\mathcal{G}$. Then, from Fano's inequality, it follows that

$$\begin{aligned} H(G^+|\hat{G}) &\leq 1 + \mathbb{P}[G^+ \neq \hat{G}^+(\boldsymbol{t}_O^{\mathcal{C}})]\cdot\log\left(|\mathcal{G}|-1\right) \\ &\leq 1 + \delta\log\left(|\mathcal{G}|-1\right). \end{aligned}$$

From the data processing inequality with (16), we have

$$H(G^+|\hat{G}) = H(G^+) - I(G^+;\hat{G}) = \log|\mathcal{G}| - I(G^+;\boldsymbol{t}_O^{\mathcal{C}})$$

which implies

$$1 + \delta\log\left(|\mathcal{G}|-1\right) \geq \log|\mathcal{G}| - I(G^+;\boldsymbol{t}_O^{\mathcal{C}})\,. \tag{17}$$

We can relate δ to $C(n)$ by bounding $I(G^+;\boldsymbol{t}_O^{\mathcal{C}})$ with regard to $C(n)$ using Lemma 4.2.

Lemma 4.2. *Under the assumption A2, the mutual information $I(G^+;\boldsymbol{t}_O^{\mathcal{C}})$ satisfies*

$$I(G^+;\boldsymbol{t}_O^{\mathcal{C}}) \leq C(n)\cdot nr\cdot H_{\max}\,, \tag{18}$$

where $H_{\max} = \max_{v\in o\subseteq V} H(t_v^k|O^k = o)$.

The proof of Lemma 4.2 is provided in Appendix B.
Then, this leads to the desired lower bound on $C(n)$:

$$C(n) \geq \frac{(1-\delta)(\log|\mathcal{G}|-1)}{nrH_{\max}}\,. \tag{19}$$

Now, we obtain $|\mathcal{G}|$ and $H_{\max}$ for our model assumption. Let a graph collection $\mathcal{G}_{D,d}$ as the set of all possible subgraphs with in-degrees d and corresponding positive edge influence probability where the base graph G has D-regular in-degrees as stated in $A1$. Then, we may obtain the size of the graph collection $|\mathcal{G}_{D,d}|$ as follows:

$$|\mathcal{G}_{D,d}| = \binom{D}{d}^n = (1+o(1))\left(\frac{D}{d}\right)^{nd}\,. \tag{20}$$

With the definition of entropy and basic algebraic manipulations, we can derive the upper bound of $H_{\max}$ as follows:

$$\begin{aligned} H_{\max} &\leq \sum_{t=1}^{T} -2(d\theta)^t\theta_0 \log 2(d\theta)^t\theta_0 - \left(1-\sum_{t=0}^{T}(d\theta)^t\theta_0\right)\log\left(1-\sum_{t=0}^{T}(d\theta)^t\theta_0\right) \\ &= \theta_0\left[\sum_{t=1}^{T}\{2t\eta^t\log\frac{1}{\eta}+2\eta^t\log\frac{1}{\theta_0}+\eta^t\}+1\right] \\ &\coloneqq \beta(\eta,\theta_0)\,. \end{aligned} \tag{21}$$

The detailed derivation of the upper bound is provided in Appendix B. Hence, we complete the proof of Theorem 4.2 by substituting (20) and (21) into (19). □

5 Evaluation Results

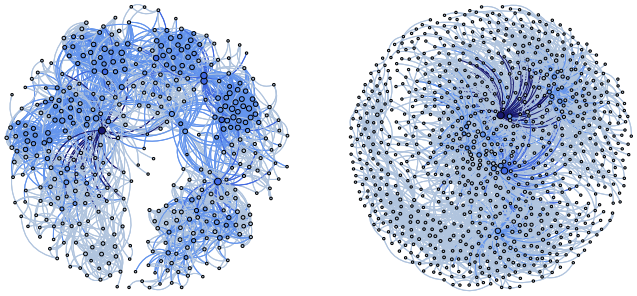
5.1 Setup

Graphs and cascades. We use a variety of types of social networks from Rossi and Ahmed [2015], ranging from small ones such as karate club network (karate), firm-hi-tech social network (firm-hitech), and tortoise social network (tortoise) to large ones such as human contact network in Dublin infections (infect-dublin) and Wikipedia who-votes-on-whom network (wiki-vote). The statistics of these five networks are presented in Table 1.

Table 1: The statistics for the networks.

Statistic	Num. nodes	Num. edges	Avg. degree	Num. triangles	Directed
infect-dublin	410	2,765	13	21,300	N
wiki-vote	889	2,914	6	64,000	N
firm-hitech	33	124	9	454	Y
tortoise	20	26	2	24	N
karate	34	78	4	135	N

For each real graph, we randomly generate five different true graphs in the following manner: Each edge uv is chosen to be a true edge with probability 0.5, by which we construct the set of true edges E^+. Each true edge $uv \in E^+$ is assigned its activation probability θ_{uv} which is a uniformly chosen random value in $(0,1]$.



(a) infect-dublin (b) wiki-vote

Figure 4: Examples of two real social networks: (a) human contact network in Dublin infections (infect-dublin) and (b) Wikipedia who-votes-on-whom network (wiki-vote).

For all edges that are not true, we set their activation probabilities to be 0. For each cascade, each node is independently and randomly determined to be a seed with probability θ_0 at $t = 0$. We generate cascades along the true edges from the seed nodes according to the IC diffusion model, where $\theta_0 = 0.01$ is set for the large networks (infect-dublin and wiki-vote) and $\theta_0 = 0.05$ for the small networks (karate, firm-hitech, tortoise).

Partial observation. There would exist different scenarios of partial observation in practice. These scenarios cover from evenly observed samples, such as the case when every node reveals their time logs randomly but with regular probability, as well as very unevenly observed samples, such as the case where some nodes never reveal their time logs over the whole cascades. To model these different partial observation scenarios, we introduce per-node revelation probability r_v for each node v. The parameter r_v represents each individual's willingness to reveal its infection time, and we assume that it is cascade-invariant, where the revelation of each node implies the observation of its infection time log. The following three scenarios are used in our evaluation: (a) *heterogeneous random,* where, for each cascade, each node v reveals its time log with its revelation probability r_v chosen uniformly at random over $[0, 1]$, (b) *homogeneous random* (r), where r_v is the same across all nodes for some $0 < r \leq 1$, i.e., $r_v = r$, and (c) *fixed* (f), where each node can have either $r_v = 0$ or 1. Each node is determined to have $r_v = 1$ with probability f and it always reveals its infection time for all cascades, otherwise, it has $r_v = 0$, which means that the node never tells its infection time at any cascade.

Table 2: Running time of algorithms. (min)

Graph	`Greedy`	`NetInf`	`DMP`	`ITRES`
infect-dublin	1.136	0.862	> 7200	2.868
wiki-vote	1.424	5.322	> 7200	4.582
firm-hitech	0.006	< 0.001	998.092	0.018
tortoise	¡ 0.001	< 0.001	52.434	¡ 0.001
karate	0.002	< 0.001	1183.060	0.010

Tested algorithms. We use the following three algorithms for comparison:

- `Greedy` [Netrapalli and Sanghavi, 2012]: This algorithm greedily infers edges based on the time difference on each edge. It achieves optimal sample complexity with fully observed samples. We use this algorithm just as a base-line one, which is expected to perform suboptimally with partially observed samples.

Table 3: Accuracy comparison on various real graphs.
(the best results are marked in **bold**.)

Graph	r	Greedy	NetInf	DMP	ITRES
infect-dublin	0.5	0.5750	0.6651	-	**0.9149**
	0.8	0.6499	0.7366	-	**0.9223**
	1	**0.9768**	0.8346	-	0.9763
wiki-vote	0.5	0.6357	0.7020	-	**0.9109**
	0.8	0.6769	0.7581	-	**0.9157**
	1	**0.9784**	0.8396	-	0.9773
firm-hitech	0.5	0.8270	0.6986	0.8811	**0.8986**
	0.8	0.8500	0.7568	0.8906	**0.9419**
	1	**0.9608**	0.8338	0.8932	**0.9608**
tortoise	0.5	0.8692	0.6269	0.7384	**0.8769**
	0.8	**0.9115**	0.6692	0.7808	0.9077
	1	**0.9308**	0.7269	0.8077	**0.9308**
karate	0.5	0.8397	0.6936	**0.8782**	0.8769
	0.8	0.8821	0.7538	0.8910	**0.9346**
	1	**0.9603**	0.8269	0.9000	0.9577

- NetInf [Gomez Rodriguez et al., 2010]: This algorithm maximizes the approximate likelihood computed based only on the most-likely cascade tree.
- DMP [Lokhov, 2016]: This algorithm maximizes a mean-field approximated likelihood with gradient descent by computing its gradient via message passing. In the paper, the SI model was used, but we modified the likelihood formulation of DMP in the IC model for a fair comparison.

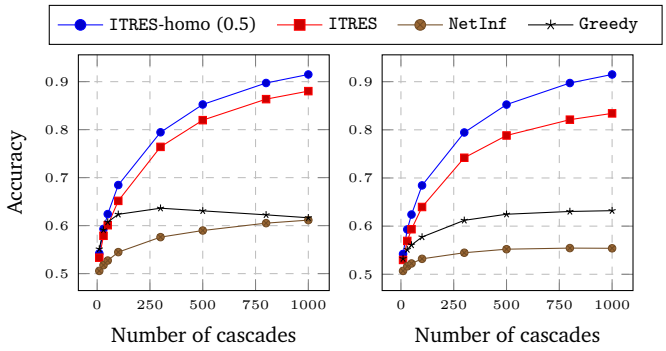
As an evaluation metric of the algorithm, we define *accuracy* to be the ratio of correctly inferred edges among all edges E in a given base graph. Since the accuracy implies the "distance" between the true graph G^+ and the inferred graph $\hat{G}$, it shows how close our inference is to the true graph. Throughout this section, we present the inference accuracy averaged over five true graphs randomly generated for each real graph.

5.2 Results

Performance for homogeneous observation. We first present the performance comparison results under homogeneous random observation for three revelation probabilities $r = 0.5, 0.8, 1.0$. For the small networks, we use 500 randomly generated IC cascade samples with initially infected seeds with a probability of 0.05 and for the large networks, 1000 cascade samples were generated with 0.01 initial activation probability. Note that we show the performance of DMP only for small networks because we cannot obtain the results in large networks within a reasonable time (longer than five days with 24 2.0 GHz CPU cores).[1] DMP takes significantly large computation time since it iterates too much before convergence. For example, in inferring the small karate network with only 34 nodes with 500 homogeneous random observation samples with $r = 0.5$, it takes about 20 hours, while greedy-style approaches such as ITRES, Greedy, and NetInf take just less than one minute. We provide the running time of each algorithm for revelation probability $r = 0.5$ in Table 2.

The inference accuracies of tested algorithms are provided in Table 3. As shown in the table, ITRES outperforms other algorithms for both full observation ($r = 1.0$) and partial observation ($r = 0.5, 0.8$.)

[1] We run DMP with initialized θ uniformly random in $[0, 1]$ for each edge. We use the step size of 0.05 for the gradient descent method and stops when the absolute difference of θ becomes lower than 10^{-3}.



(a) Heterogeneous Random (b) Fixed (0.5)

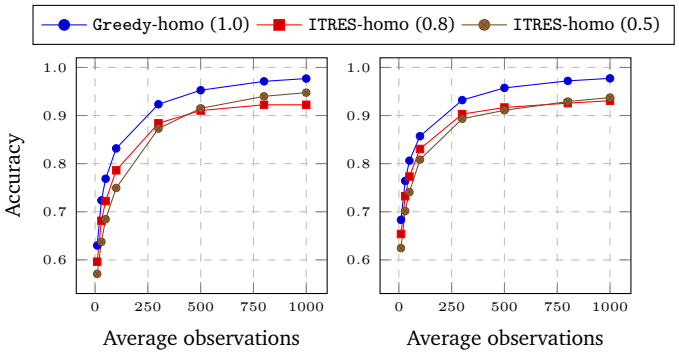
Figure 5: Impact of imbalanced observation on the infect-dublin network (ITRES, NetInf, and Greedy).

With fully observed cascade samples, ITRES runs similarly to Greedy and performs the best among other algorithms with fully observed samples. However, when some parts of time logs are missing, Greedy loses its consistency and shows worse performance than ITRES. Especially, ITRES learns the infect-dublin graph about 30% more accurately than Greedy, when samples are partially observed. ITRES also outperforms NetInf and DMP which are devised considering partial observations. Thus, we can say that ITRES maintains high performance regardless of data completeness.

Impact of imbalanced observation. We now investigate the impact of imbalanced observation scenarios, where we consider the cases of heterogeneous random and fixed, wherein some nodes never or rarely reveal their infection time over every cascade while other nodes always reveal their infection time. We compare the performance of ITRES, NetInf, and Greedy on infect-dublin network, for both heterogeneous random and fixed observation scenarios. Intuitively, evenly observed logs, such as homogeneous random observation, is of great advantage to graph inference algorithms since it enables algorithms to evenly capture every edges' activation over cascades. As a node reveals more of its infection time, there are more chances to capture the hints implying the infections along true edges around the node. In heterogeneous random or fixed scenarios, where some nodes hardly or never reveal their infection times, selecting true edges around the nodes can be significantly challenging. However, Figure 5 shows that ITRES successfully recovers most of the edges despite such imbalanced observation compared to NetInf and Greedy.

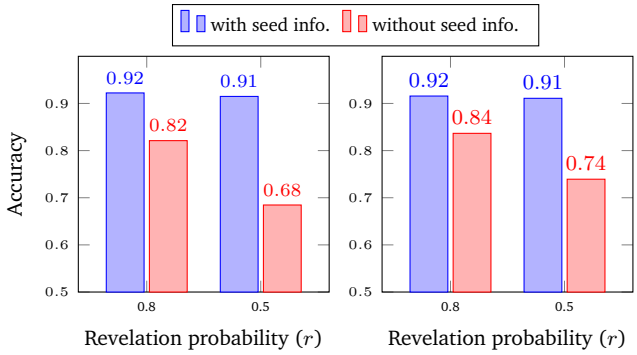
Under the heterogeneous random observation, as shown in Figure 5a, ITRES shows much higher accuracy than other algorithms, and nearly 90% of edges are recovered when enough cascade samples are given, while NetInf and Greedy show limited performance although enough cascades are given. This is because other algorithms seem to fail to infer edges around the nodes who are rarely observed, while our algorithm succeeds to find true edges by recovering the missing times of the nodes. We also see that ITRES loses only a small amount of accuracy compared to the homogeneous random observation ($r = 0.5$). We also validate our algorithm under the fixed observation ($f = 0.5$), which is more difficult since half of the nodes are never observed in this case. However, as shown in Figure 5b, ITRES still shows the high accuracy of more than 80%, significantly outperforming NetInf and Greedy. As with the heterogeneous random scenario, NetInf and Greedy show limited performance even enough cascades are given, while ITRES shows better performance as many cascades are given.

Full vs. partial observation with the same observation cost. In this section, we address the following question: *given a budget for observation cost, which is better, a small number of cascades with full observation, or a large number of cascades with partial observation?* For a fair comparison, we equalize the observation cost for both cases by making the average number of observed nodes the same. For example, if M fully-observed cascade samples are given, then the average observation cost is $M \cdot n$, which equals to the cost of $\frac{M}{r}$ samples with $n \cdot r$ observed logs for each cascade, where time logs are observed under the homogeneous random scenario with r. To evaluate our algorithm from this perspective of observation cost, we compare



(a) infect-dublin (b) wiki-vote

Figure 6: Full vs. partial observation with the same observation cost (ITRES and Greedy).



(a) infect-dublin (b) wiki-vote

Figure 7: Performance of ITRES without seed information.

ITRES against Greedy, which is known to require (asymptotically) minimum samples under full observation Netrapalli and Sanghavi [2012]. The accuracy comparison is conducted where Greedy is given M fully observed samples and ITRES is given $\frac{M}{r}$ homogeneously observed samples with $r = 0.5, 0.8$ for infect-dublin and wiki-vote networks. The result is shown in Figure 6. On infect-dublin and wiki-vote networks, ITRES achieves comparable performance to Greedy. Even with half of the samples missing, the loss of inference accuracy is less than 5% with ITRES. This shows that ITRES maximally utilizes the amount of data, even when it is incomplete. This corresponds to the sample complexity analysis provided in Section 4, which implies that ITRES requires $\frac{1}{r}$ times more cascade samples when samples are partially observed with the ratio of r to achieve the same accuracy achieved under full observation.

Impact of seed information. The absence of seed information makes the task of inferring graphs highly challenging since the direction of infection becomes much more ambiguous when any of the unobserved nodes might be a source of the infection. We quantify this degradation to see the impact of seed information in Figure 7 for 1000 partially observed samples with $r = 0.5$ and $r = 0.8$. For both infect-dublin and wiki-vote networks, ITRES correctly infers more than 80% of edges without seed information but decreases by 10% when compared to edges with seed information. When the portion of missing observation data grows, e.g., $r = 0.5$, the gap due to the absence of seed information also grows.

6 Conclusion

We consider the graph learning where the information of cascade samples are only partially observed under the independent cascade model. We develop an algorithm that maximizes approximate likelihood with a greedy approach and analyze the sample complexity of the algorithm. We evaluate the performance of our algorithm over various types of real-world graphs under diverse scenarios.

References

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

Kareem Amin, Hoda Heidari, and Michael Kearns. Learning from contagion (without timestamps). In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, page II–1845–II–1853. JMLR.org, 2014.

Hadi Daneshmand, Manuel Gomez-Rodriguez, Le Song, and Bernhard Schölkopf. Estimating diffusion network structures: Recovery conditions, sample complexity and soft-thresholding algorithm. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, page II–793–II–801. JMLR.org, 2014.

Nan Du, Le Song, Alex Smola, and Ming Yuan. Learning networks of heterogeneous influence. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, page 2780–2788. Curran Associates Inc., 2012.

Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1019–1028. Association for Computing Machinery, 2010.

Amit Goyal, Francesco Bonchi, and Laks V.S. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, page 241–250. Association for Computing Machinery, 2010.

Vincent Gripon and Michael G. Rabbat. Reconstructing a graph from path traces. *2013 IEEE International Symposium on Information Theory*, pages 2488–2492, 2013.

Xinran He and Yan Liu. Not enough data? joint inferring multiple diffusion networks via network generation priors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, page 465–474. Association for Computing Machinery, 2017.

Xinran He, Ke Xu, David Kempe, and Yan Liu. Learning influence functions from incomplete observations. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 2073–2081. Curran Associates Inc., 2016.

Dimitris Kalimeris, Yaron Singer, Karthik Subbian, and Udi Weinsberg. Learning diffusion using hyperparameters. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2420–2428. PMLR, 2018.

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*, page 137–146. Association for Computing Machinery, 2003.

Andrey Y. Lokhov. Reconstructing parameters of spreading models from partial observations. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 3467–3475. Curran Associates Inc., 2016.

Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge university press, 2005.

Praneeth Netrapalli and Sujay Sanghavi. Learning the graph of epidemic cascades. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, page 211–222. Association for Computing Machinery, 2012.

Jean Pouget-Abadie and Thibaut Horel. Inferring graphs from cascades: A sparse recovery framework. In *Proceedings of the 24th International Conference on World Wide Web*, page 625–626. Association for Computing Machinery, 2015.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, page 4292–4293. AAAI Press, 2015. URL `http://networkrepository.com`.

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

Y. Sun, C. Qian, N. Yang, and P. S. Yu. Collaborative inference of coexisting information diffusions. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1093–1098, 2017.

Xiaojian Wu, Akshat Kumar, Daniel Sheldon, and Shlomo Zilberstein. Parameter learning for latent network diffusion. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, page 2923–2930. AAAI Press, 2013.

Kai Zhu, Zhen Chen, and Lei Ying. Locating the contagion source in networks with partial timestamps. *Data Min. Knowl. Discov.*, 30(5):1217–1248, 2016.

Kai Zhu, Zhen Chen, and Lei Ying. Catch'em all: Locating multiple diffusion sources in networks with partial observations. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1676–1683. AAAI Press, 2017.

B. Zong, Y. Wu, A. K. Singh, and X. Yan. Inferring the underlying structure of information cascades. In *2012 IEEE 12th International Conference on Data Mining*, pages 1218–1223, 2012.

A Algorithm Description

Algorithm 1: ITRES (Iterative Time Recovery and Edge Selection)

Input: base graph $G = (V, E)$, observed infection times $\boldsymbol{t}_O^{\mathcal{C}}$
Output: inferred graph $\hat{G} = (V, \hat{E})$

% Construct feasible time sets
for $k \in \mathcal{C}$ **do**
  Initialize $F_v^k = \{t_v^k\}$ if $v \in O^k$ otherwise $F_v^k = \emptyset$;
  for $0 \leq t < T - 1$ **do**
    for $v \in V \setminus O^k$ **do**
      $F_v^k = F_v^k \cup \{t + 1\}$ if $\exists u \in V_v$ s.t. $t \in F_u^k$;
Initialize $\hat{G}_0 = (V, \emptyset)$, $\hat{\boldsymbol{t}}_0 = \{F_v^k\}_{v \in V, k \in \mathcal{C}}$, $i = 0$;
while $\hat{G}_{(i)} \neq \hat{G}_{(i-1)}$ **do**
  % Time recovery
  for $k \in \mathcal{C}$ **do**
    $\hat{\boldsymbol{t}}^k = \{F_v^k\}_{v \in V}$;
    for $t \in \{T - 1, ..., 1, 0\}$ **do**
      $A(j) = \{u \in V | j \in F_u^k\}$;
      $B_u(j) = \{v \in V | uv \in \hat{E}_{(i)}, \hat{\boldsymbol{t}}_v^k = \{j + 1\}\}$;
      for $u \in A(t)$ **do**
        if $|B_u(t) \setminus \cup_{z \in A(t) \setminus \{u\}} B_z(t)| > 0$ **then**
          $\hat{\boldsymbol{t}}^k = \{t\}$;
  $\hat{\boldsymbol{t}}_{(i+1)} = \{\hat{\boldsymbol{t}}^k\}_{k \in \mathcal{C}}$;
  % Edge selection
  for $v \in V$ **do**
    $\hat{V}_v = \emptyset$;
    while $\sum_{u \in V_v \setminus \hat{V}_v} \sum_{k \in \mathcal{C}} x_{uv}^k \prod_{z \in \hat{V}_v} (1 - x_{zv}^k) > 0$ **do**
      $u = \arg\max_{u \in V_v \setminus \hat{V}_v} \sum_{k \in \mathcal{C}} x_{uv}^k \prod_{z \in \hat{V}_v} (1 - x_{zv}^k)$;
      $\hat{V}_v = \hat{V}_v \cup \{u\}$;
  $\hat{G}_{(i+1)} = (V, \hat{E}_{(i)})$ with $\hat{E}_{(i)} = \{uv \in E : u \in \hat{V}_u\}$;
  $i = i + 1$;
return $\hat{G} = \hat{G}_{(i)}$

B Proof of Lemmas

Proof of Lemma 4.1. We omit k when we describe the random variables for one cascade, unless confusion arises, since cascade events are identically and independently distributed. To prove (14), we characterize

$\mathbb{P}[y_{uv}=1]$ for any $u \in V_v^+$ as follows:

$$\begin{aligned}\mathbb{P}[y_{uv}=1] &= \sum_{0<t\le T} \mathbb{P}[x_{uv}=1, x_{zv}=0\ \forall z \in V_v^+ \setminus \{u\}, t_v=t] \\ &\ge \sum_{0<t\le T} \mathbb{P}[x_{uv}=1, t_v=t] \times \left(1 - \sum_{z\in V_v^+\setminus\{u\}} \mathbb{P}[x_{zv}=1 \mid x_{uv}=1, t_v=t]\right),\end{aligned}$$

where the last inequality follows from the union bound. Then, it is sufficient to show

$$\mathbb{P}[x_{uv}=1, t_v=t] \ge \frac{r\theta_0}{4d}(\theta d)^t \tag{22}$$

$$\mathbb{P}[x_{zv}=1 \mid x_{uv}=1, t_v=t] < \frac{1}{2d} \tag{23}$$

for any $z \in V_v^+ \setminus \{u\}$.

To obtain the lower bound in (22), which implies that any true parent u has enough chance to be regarded as a possible infector of v, we characterize the probabilty as follows:

$$\begin{aligned}\mathbb{P}[x_{uv}=1, t_v=t] &\ge \mathbb{P}[v\in O, u \overset{t}{\to} v] \\ &= \mathbb{P}[v\in O]\ \mathbb{P}[u \overset{t}{\to} v | t_u = t-1, t_v > t-1]\mathbb{P}[t_u = t-1 | t_v > t-1]\ \mathbb{P}[t_v > t-1], \end{aligned} \tag{24}$$

where $a \overset{t}{\to} b$ denote the event that a infects b at time t for any distinct nodes a and b. The first inequality follows from the fact that the feasible time set of u always includes $t-1$ when u actually activates v at t and the next equality follows from the independence between cascade events and revelation events. Now, we can derive (22) from the lower bound of each term in (24). From the model assumption A1 and A2, we obatin

$$\mathbb{P}[v\in O] = r\ ,\ \mathbb{P}[u \overset{t}{\to} v | t_u = t-1, t_v > t-1] = \theta\,. \tag{25}$$

Next, noting that u is infected at $t-1$ if one of u's ancestor node which has a path of length $t-1$ to u initiates and propagates the infection through the path to u, we write the event $\{t_u = t-1 | t_v > t-1\}$ as

$$\bigcup_{a_0 \in A_{uv}^+(t-1)} \{t_{a_0}=0, a_0 \overset{1}{\to} \dots \overset{t-1}{\to} u\},$$

where $A_{uv}^+(l)$ denote a set of true ancestor nodes that have a path of length l to u not including v in the true graph G^+.

The propagation from a_0 to u successfully occurs if (a) every intermediate node between a_0 to u remains susceptible unitl the infection originated from a_0 arrives and (b) every edge in the propagation path is activated. Thus, we formulate the probability of the propagation event for each a_0 as follows:

$$\begin{aligned}\mathbb{P}[t_{a_0}=0, a_0 \overset{1}{\to} \dots \overset{t-1}{\to} u] &\ge \theta^{t-1} \cdot \mathbb{P}[t_{a_0}=0, t_a \ne 0\ \forall a \in \{\cup_{i=0}^{t-1} A_{uv}^+(i)\} \setminus \{a_0\}] \\ &\ge \theta^{t-1}\theta_0(1-\theta_0)^{2d^{t-1}} > \frac{1}{2}\theta^{t-1}\theta_0\,,\end{aligned} \tag{26}$$

where the first inequality follows from the fact that (a) always occurs if none of ancestor nodes of u within $(t-1)$ hops except a_0 is initially infected and (b) occurs when each of $(t-1)$ edges on the path is independently activated with probability θ, the second inequality follows from the assumption that the true graph is a locally tree with d in-degrees within $t \le T$ hops, which implies $|\{\cup_{i=0}^{t-1} A_{uv}^+(i)\}| \le \sum_{i=1}^{t-1} d^i \le 2d^{t-1}$, and the last inequality holds since θ_0 is small enough to claim that $(1-\theta_0)^{2d^{t-1}} \ge 1 - 2d^{t-1}\theta_0 > \frac{1}{2}$. Note that we have small θ_0 under the assumptions in A1 and Theorem 4.1 as follows:

$$\theta_0 < \frac{D}{64dn} \le \frac{1}{64dD^{T-1}} \tag{27}$$

since the locally tree structure rooted to v within T hops has at least D^T distinct nodes.

Because the propagation events for each a_0 are disjoint if a_0 is the only seed in $A^+_{uv}(t-1)$, we obtain the lower bound of the third probability in (24) by suming (26) over $a_0 \in A^+_{uv}(t-1)$ as follows:

$$\mathbb{P}[t_u = t-1 | t_v > t-1] > \frac{1}{2}(d\theta)^{t-1}\theta_0\,, \tag{28}$$

where the inequality holds since the true graph is a locally tree with d in-degrees and $|A^+_{uv}(t-1)| = d^{t-1}$ by A1.

Lastly, the probabilty that v is not infected until $t-1$ in (24) is bounded as:

$$\mathbb{P}[t_v > t-1] = 1 - \mathbb{P}[t_v \le t-1] \ge 1 - \sum_{i=0}^{t-1}(d\theta)^i\theta_0 > \frac{1}{2} \tag{29}$$

since $\mathbb{P}[t_v = i] \le \sum_{u\in V_v^+}\theta \cdot \mathbb{P}[t_u = i-1] \le (d\theta)^i\theta_0$ and θ_0 is small enough as claimed in (27). By substituting (25), (28), and (29) to (24), we conclude the proof of (22).

Next, to obtain the upper bound in (23), we first characterize the probability as follows:

$$\begin{aligned}\mathbb{P}[x_{zv} = 1 \mid x_{uv} = 1, t_v = t] &= \mathbb{P}[x_{zv} = 1 | t_v = t] \\ &\le \mathbb{P}[\exists a_0 \in A_z(t-1) \text{ s.t. } t_{a_0} = 0]\,,\end{aligned}$$

where the first equality follows from the fact that $x_{uv} = 1$ and $x_{zv} = 1$ are independent given that $t_v = t \le T$ since the parents u and z share no ancestor within $(T-1)$ hops and thus there is no overlap between possible infection paths to u and z of length $t-1$ ($\le T-1$) in the local tree structure assumed in Theorem 4.1 and the next inequality holds since infection originated from any seed in $A_z(t-1)$, which denotes a set of ancestor nodes that have a path of length $t-1$ to z in the base graph G, is necessarily observed among the ancestors of z for $t-1$ to be added to the feasible time set of z. Then, we derive the upper bound in (23) as follows:

$$\mathbb{P}[\exists a_0 \in A_z(t-1) \text{ s.t. } t_{a_0} = 0] \le D^{t-1}\theta_0 < \frac{1}{2d}\,,$$

where the first inequality follows from the union bound and the tree structure assumed in A1 and Theorem 4.1 implying that $|A_z(t-1)| = D^{t-1}$ and the last inequality holds since θ_0 is small enough as claimed in (27).

To prove (15), we first characterize the upper bound for any $w \notin V_v^+$ when t_v is fixed as t:

$$\begin{aligned}\mathbb{P}[x_{wv} = 1, t_v = t] &\le \mathbb{P}[t_v = t, v \in O, \exists a_0 \in A_w(t-1) \text{ s.t. } t_{a_0} = 0] \\ &\le |A_w(t-1)|\mathbb{P}[t_{a_0} = 0]\mathbb{P}[t_v = t, v \in O] \\ &\le D^{t-1}r(d\theta)^t(\theta_0)^2\,,\end{aligned}$$

where the first inequality holds since infection originated from any seed in $A_w(t-1)$ is necessarily observed among the ancestors of w within $(t-1)$ hops for $x_{wv} = 1$, the next inequality follows from the fact that w is false parent of v and infection events on w's ancestors and v are independent, and the last inequality follows from the fact that $\mathbb{P}[t_v = t-1] \le (d\theta)^{t-1}\theta_0$ and the assumptions about the the independence between cascade and revelation and the tree structure with D in-degrees implying $|A_w(t-1)| = D^{t-1}$ as stated in A1, A2 and Theorem 4.1.Then, we can draw the desired result completing the proof of (13) by summing the above upper bounds over $1 \le t \le T$ with the fact that $\sum a_t \cdot b_t \le \sum a_t \sum b_t$ where $a, b > 0$ and θ_0 is small enough as claimed in (27).

Proof of Lemma 4.2. Using the chain rule, we first rewrite the mutual information as follows:

$$\begin{aligned}I(G^+; \boldsymbol{t}_O^{\mathcal{C}}) &= \sum_{k=1}^{C(n)} I(\boldsymbol{t}_O^k; G^+ | \boldsymbol{t}_O^1, ..., \boldsymbol{t}_O^{k-1}) \\ &= \sum_{k=1}^{C(n)} I(\boldsymbol{t}_O^k; G^+ | \boldsymbol{t}_O^1, ..., \boldsymbol{t}_O^{k-1}, O^k)\end{aligned} \tag{30}$$

since O^k is partial information of $\boldsymbol{t}_O^k$. Indeed, we have

$$I(G^+;\boldsymbol{t}_O^k,O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1}) = I(G^+;\boldsymbol{t}_O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1}) + \underbrace{I(G^+;O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1},\boldsymbol{t}_O^k)}_{=0}$$
$$= \underbrace{I(G^+;O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1})}_{=0} + I(G^+;\boldsymbol{t}_O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1},O^k)\,.$$

Using the expression in (30), we obtain an upper bound of $I(G^+;\boldsymbol{t}_O^{\mathcal{C}})$:

$$\begin{aligned} I(G^+;\boldsymbol{t}_O^{\mathcal{C}}) &= \sum_{k=1}^{C(n)} \{H(\boldsymbol{t}_O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1},O^k) - H(\boldsymbol{t}_O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1},O^k,G^+)\} \\ &= \sum_{k=1}^{C(n)} \{H(\boldsymbol{t}_O^k|\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1},O^k) - H(\boldsymbol{t}_O^k|O^k,G^+)\} \\ &\le \sum_{k=1}^{C(n)} H(\boldsymbol{t}_O^k|O^k) - H(\boldsymbol{t}_O^k|O^k,G^+) = \sum_{k=1}^{C(n)} I(\boldsymbol{t}_O^k;G^+|O^k)\,, \end{aligned} \tag{31}$$

where the second equality follows from the fact that $\boldsymbol{t}_O^k$ is conditionally independent to $\boldsymbol{t}_O^1,...,\boldsymbol{t}_O^{k-1}$ given O^k and G^+, and the last inequality follows from the fact that $H(X|Y) \le H(X)$.

Using the basic properties of entropy, we further bound the last term in (31) as follows:

$$\begin{aligned} I(\boldsymbol{t}_O^k;G^+|O^k) &= \sum_{o\subseteq V} \mathbb{P}[O^k=o] I(\boldsymbol{t}_O^k;G^+|O^k=o) \\ &\le \sum_{o\subseteq V} \mathbb{P}[O^k=o] H(\boldsymbol{t}_O^k|O^k=o) \\ &\le \sum_{o\subseteq V} \mathbb{P}[O^k=o] \sum_{v\in o} H(t_v^k|O^k=o) \\ &\le \sum_{0\le m\le n} \mathbb{P}[|O^k|=m]\cdot m\cdot H_{\max}\,, \end{aligned} \tag{32}$$

where for the first, second inequalities and third inequalities, we use the positivity of entropy, i.e., $I(X;Y) \le H(X)$, the subadditivity of entropy and $H_{\max} = \max_{v\in o\subseteq V} H(t_v^k|O^k=o)$, respectively. Therefore, combining (31) and (32), we have

$$I(G^+;\boldsymbol{t}_O^{\mathcal{C}}) \le \sum_{k=1}^{C(n)} \mathbb{E}[|O^k|]\cdot H_{\max}\,.$$

Upper bound of $H_{\max}$. Since cascade events are identically and independently distributed, we omit k when we describe the random variables for one cascade, unless confusion arises. Using the definition of entropy, we write the entropy of t_v for any node $v \in o \subseteq V$ as follows:

$$\begin{aligned} H(t_v|O=o) = (&\sum_{0<t\le T} -\mathbb{P}[t_v=t|O=o]\log\mathbb{P}[t_v=t|O=o]) \\ &- \mathbb{P}[t_v>T|O=o]\log\mathbb{P}[t_v>T|O=o]\,. \end{aligned} \tag{33}$$

Using the properties of t_v, we obtain

$$\mathbb{P}[t_v=t] \;\le\; \sum_{z\in V_v} \theta_{zv}\mathbb{P}[t_z=t-1] \;\le\; d\theta\mathbb{P}[t_z=t-1]\,,$$

where the first inequality follows from the fact that v is activated at t when at least one of its neighbor z is activated at $t-1$ and activates v, and the second one follows from the assumption which the underlying true graph has d in-degrees and θ activation probability for each node and edge. Thus, the recursive upper bound implies

$$\mathbb{P}[t_v = t] \;\leq\; (d\theta)^t \theta_0 \,. \tag{34}$$

Then, we obtain an upper bound of (33) as follows:

$$\begin{aligned} H(t_v|O=o) &\leq \sum_{t=1}^{T} -2(d\theta)^t\theta_0 \log 2(d\theta)^t\theta_0 - \left(1 - \sum_{t=0}^{T}(d\theta)^t\theta_0\right) \log \left(1 - \sum_{t=0}^{T}(d\theta)^t\theta_0\right) \\ &\leq \sum_{t=1}^{T} -2(d\theta)^t\theta_0 \log 2(d\theta)^t\theta_0 + \sum_{t=0}^{T}(d\theta)^t\theta_0 \\ &= \theta_0[\sum_{t=1}^{T}\{2t\eta^t \log\frac{1}{\eta} + 2\eta^t \log\frac{1}{\theta_0} + \eta^t\} + 1] \,, \end{aligned} \tag{35}$$

where the first inequality follows from the fact that $-x\log x$ is increasing when $0 < x \leq \frac{1}{e}$ and $\mathbb{P}[t_v = t|O=o] \leq 2(d\theta)^t\theta_0 \leq \frac{1}{e}$ for small $\theta_0 \leq \frac{1}{2e\max(\eta^T,1)}$ and $-x\log x$ is decreasing when $\frac{1}{e} \leq x < 1$ and $\mathbb{P}[t_v > T] \geq 1 - \mathbb{P}[t_v \leq T]$. The second inequality follows from $-(1-x)\log(1-x) \leq x$. The last inequality follows from basic algebraic manipulations.