IDENTIFYING INFLUENTIAL SPREADERS IN COMPLEX NETWORKS

Şirag Erkol

Submitted to the faculty of the Graduate School Bloomington
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the Luddy School of Informatics, Computing, and Engineering,
Indiana University
May 2023

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral committee

Filippo Radicchi, Ph.D. (Committee Chair)

Santo Fortunato, Ph.D.

Alessandro Flammini, Ph.D.

Roni Khardon, Ph.D.

Date of Defense: April 11, 2023

ACKNOWLEDGEMENT

I would first like to thank my advisor Filippo Radicchi. His immense support through the last six years have made my life at IU easy and fun. I am lucky that I had him as my advisor in this journey. His wisdom and guidance has helped me navigate in my research and take the next steps in my academic career. I would also like to thank Santo Fortunato for all his help and support. I am grateful for having the chance of learning from him and working with him. I thank both Filippo and Santo for their help in my research and this dissertation. I am also grateful to Alessandro Flammini and Roni Khardon for being a part of my committee.

I am grateful to my master's advisor Gönenç Yücel for all his help, and for introducing me to network science. I further thank Yaman Barlas, especially for creating an intellectually stimulating research environment during my time at Boğaziçi University.

I am thankful to my collaborators Dario Mazzilli, Satyaki Sikdar, Siddharth Patwardhan, and Ali Faqeeh for all the discussions we had during our research together.

I want to thank my friends for all the fun. Special thanks to Masis and Vartan for always being there.

Finally, I want to thank my family. Thank you mom and dad for your unconditional love and support. I am grateful to Narod for always being by my side, also thank you and Aren for Aram. And thank you Aram for being a bundle of joy! I could not have done this without any of you.

Şirag Erkol

IDENTIFYING INFLUENTIAL SPREADERS IN COMPLEX NETWORKS

Influence maximization is the problem of identifying the set of nodes that maximize the size of the outbreak of a spreading process occurring on the network. This problem is important for strategic decisions in marketing and political campaigns. Typically, the problem consists of finding small sets of initial spreaders in large static networks. Due to its computational complexity, the problem can not be solved exactly. Many methods have been proposed to approximate solutions to the influence maximization problem. Here, we first study the effectiveness of proposed methods on a large corpus of real-world networks. We show that simple heuristic methods with low computational complexity can provide comparable solutions to optimization algorithms with high computational burden. Furthermore, we propose a machine learning based approach that combines heuristic methods to increase the performance of provided solutions. Next, we tackle the problem of noise in network structure and dynamics data. We analyze both the individual and combined effects of structural and dynamical noise on the quality of solutions. We show that implementing artificial noise can improve the performance of optimization algorithms to identify influential spreaders. We further analyze the influence maximization problem on temporal networks. We show that losing the information on the ordering or the timing of the interactions significantly decreases the ability to identify influential spreaders. Furthermore, information of the network structure during the first phases of the spreading dynamics is important in order to successfully find influential spreaders, especially when the recovery probability is high.

Contents

List of Figures

List of Tables

1 Introduction

Every day, new pieces of information disseminate in social networks (3–7). Only a very small portion of them become widespread, while the others disseminate to a vanishing portion of the network and die out. What factors decide whether a piece of information will be viral or be able to reach only a few people? Many studies show that the quality of the information has an important affect on how far information will spread (3, 8). In models of information spreading, quality is typically quantified as the probability of spreading along an edge in social networks. However, the spreading probability is not the only determinant for the virality of a piece of information. The seed nodes where the information originates from are important too. Intuitively, a spreading process that starts from central nodes is expected to reach many more nodes compared to a spreading process that starts from peripheral nodes. Central nodes have many direct connections, which help the dissemination. On the other hand, peripheral nodes typically have only a few neighbors, meaning that the spreading process will have only a few chances to spread from the seed nodes to their neighbors, limiting the size of spreading, thus making the process much less likely to become widespread.

The problem of finding the best seed set that maximizes the outbreak size of a spreading process is called the influence maximization problem. In its most traditional form, the problem consists of finding a fixed number of seeds on a static network. It is also assumed that there is complete knowledge on the network topology and of the spreading dynamics. The function to optimize in influence maximization is the average value of the outbreak size for a fixed size of the seed set. The size of the seed set is a small fraction of the network size because in real-world applications the resources are typically limited. The influence maximization problem is NP-hard, effectively meaning that optimal solutions cannot be found except for very small systems. In its traditional form, the optimal solution to the influence maximization problem can be approximated using a greedy optimization method with an optimality gap of $1 - 1/e$ (9). The optimality gap comes from the fact that the

influence function is submodular. The greedy optimization uses information on both the topology of the network and the spreading dynamics, and it is the state-of-the-art method to solve the influence maximization problem. However, its applicability is limited by its high computational complexity. Some other methods, called heuristic methods, can be used to lower the computational burden for a trade-off with solution accuracy (10). Heuristic methods rely only on the knowledge of network topology to find seed nodes. They are much faster than greedy optimization, but provide worse solutions. The quality of the solutions they provide is an important point to investigate, as it can answer the necessity of using greedy optimization.

There are other experimental setups in which one can analyze the influence maximization problem. One of the strongest assumptions is that there is full knowledge of the network topology and of the spreading dynamics while solving the influence maximization problem. This assumption might be unrealistic in some cases (11–17), and methods to overcome these limitations are necessary. Also, the influence maximization problem on static networks has been widely studied in the literature (9, 18–22). However, many real-world networks are not static, they change in time (23). Studying the influence maximization problem on temporal networks is another necessary improvement for its applications in real-world scenarios.

In this thesis, we focus on the problem of influence maximization. We answer the questions asked above about the effectiveness of greedy optimization and heuristic methods, and the effect of incomplete knowledge of network topology and spreading dynamics on the solutions to the influence maximization problem. Finally, we study the problem of influence maximization on temporal networks.

This thesis consists of 7 chapters. After the introduction in Chapter 1, we introduce general methods used throughout the thesis in Chapter 2. These methods include defining complex networks, both static and temporal, introducing spreading models, defining the implementation and application of the influence maximization problem, and presenting methods used to find influential spreaders in complex networks. In the following 4 chapters,

we show the results of our studies. In Chapter 3, we present the results on our study of heuristic methods and their performances compared to greedy optimization on a big corpus of networks. We also introduce hybrid methods, which combine several heuristic methods and increase their performance (24). In Chapter 4, we present the results of our study on the effects of noise in network topology and spreading dynamics in influence maximization (25). In Chapter 5, we introduce the study of influence maximization on temporal networks (26). We present the effectiveness of the greedy algorithm in this non-traditional setting, and compare it to several approximation methods. In Chapter 6, we present the results of our analysis on the submodularity property of the influence function in the context of influence maximization on temporal networks (27). Finally, in Chapter 7, we conclude the thesis. Supplementary material is reported in the appendices.

2 Methods

2.1 Complex networks

An undirected network $G = (V, E)$ is composed of two sets: V is the set of nodes, and E is the set of edges. Nodes represent the entities in the network, and edges represent the interactions between those entities. The edge between nodes i and j is denoted as (i, j). A connected component in a network is a subset of nodes such that there exists a path between every pair of nodes in the subset. The connected component in the network that has the largest number of nodes is called the largest connected component.

Networks are commonly represented with an adjacency matrix A, such that

$$A_{ij} = \begin{cases} 1 \text{ if } (i,j) \in E \\ \\ 0 \text{ if } (i,j) \notin E. \end{cases} \tag{2.1}$$

Since the network is undirected, we have $A_{ij} = A_{ji}$.

2.1.1 Static networks

A static network is the simplest type of complex network. The topology of a static network does not change with time. Nodes do not leave the network or new nodes do not emerge, edges do not appear or disappear. The small/medium sized real-world static networks we use in this thesis are shown in Tables A.1-A.3. The sizes of networks range from 100 to 30,000 nodes, and their densities vary between 0.0001 and 0.25. They are from different domains, we have 63 social, 16 technological, 10 information, 8 biological, and 3 transportation networks. We also use a small corpus of large real-world networks, shown in Table 3.1. These networks range in size from 50,000 to slightly more than 1,000,000 nodes.

2.1.2 Temporal networks

Besides static networks, the interactions between nodes can also be represented as temporal networks, given that there is a time component. For example, a road network representing the connections between several points in a city will be better represented as a static network. However, if one considers the interactions between students in a school (28, 29), the timing of interactions can give important information about the dynamics. Such a system can be represented in a more detailed way without losing the time dimension using a temporal network.

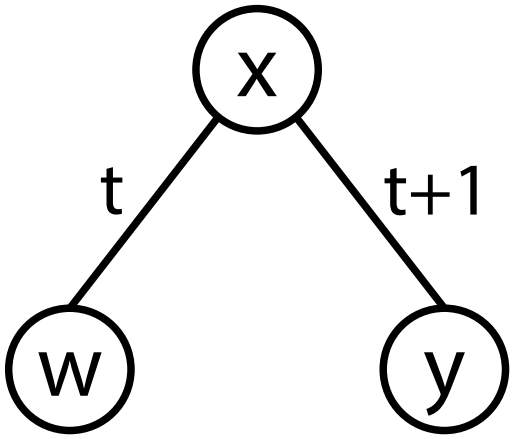

Figure 2.1: **A simple temporal network.** A toy example of a temporal network where the time stamps of edges are also presented.

For example, let us examine the toy network shown in Figure 2.1, where node w interacts with node x at time t, and node x interacts with node y at time $t+1$. If we disregard the time stamps and take this network as a static network, it would suggest that an information in node y can travel to node w, because there is a path through node x. However, if we take the timing of interactions into consideration, we can see that such a path does not exist because the interaction between x and y happens later in time. This simple example shows that losing the time dimension of a network data can result in different outcomes, meaning

the information on the timing of interactions can be crucial for the outcome of a dynamical process happening on a network.

A temporal network can be represented as $G = (V, E)$. The edges in such a network are represented using (i, j, t), where $i, j \in V$ and t represents the time of interaction between i and j.

The modeling scheme we follow in this thesis for temporal networks is a common one (23, 30, 31). Given the set of edges E, we slice the set of edges into time windows of equal length W. In a single time window, we aggregate all interactions to form a static network. Multiple interactions between two nodes in the same layer are represented as a single unweighted edge. Applying the aggregation to all the time windows gives T layers, and these layers in the given order represent the temporal network. All layers have N nodes, even though some nodes might have no interactions in certain layers. An example of a temporal network represented with layers of static networks is given in Figure 2.2. In Table 5.1, we present the real-world temporal networks we use in this thesis. These are small-sized temporal networks with the largest network having slightly more than 300 nodes. They consist of multiple layers, representing the changing nature of interactions in the network. In some networks, interactions correspond to physical proximity contacts, and in others the interactions correspond to emails exchanged between people.

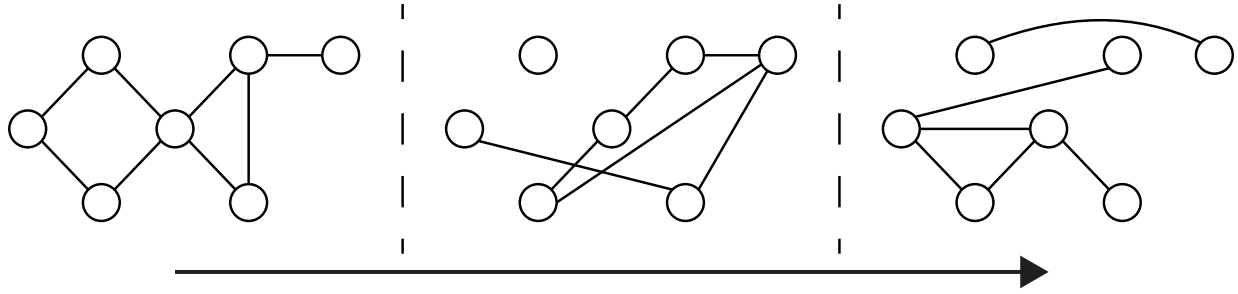

Figure 2.2: **Representation of a temporal network.** A toy example of a temporal network represented as layers of static networks. In each layer, the set of edges E can change, but the nodes stay the same. The time moves from left to right.

2.2 Spreading models

In real-world networks, we observe spreading of different phenomena, such as diseases (32, 33), information (8, 34), and misinformation (4, 5). In order to analyze the spreading behavior on networks, researchers have proposed different models, such as the Susceptible-Infected model and its variants (32), and the Linear Threshold model (LTM) (9, 35, 36). In the following sections, we introduce the spreading models that we made use of in this thesis.

2.2.1 Susceptible-Infected-Recovered model

In the Susceptible-Infected-Recovered (SIR) model (32), nodes can be in one of the three states S, I, or R. At the beginning of the dynamics, all nodes are in state S except for the nodes in the set of initial spreaders $\mathcal{X}$, which are in state I. At each step of the model, nodes in state I infect their neighbors in state S with probability λ, and nodes in state I recover by changing their state from I to R with probability μ. Once a node recovers, it does not participate in the spreading dynamics anymore. When there are no nodes left in state I, the dynamics stop and the number of nodes in state R give the outbreak size. This model is stochastic unless $\lambda, \mu \in \{0, 1\}$. In order to account for the stochasticity, the results are obtained by averaging over multiple realizations of the model for a given initial condition.

Using a mean-field approximation, we can define the probability of each node being in the states S, I, and R at any given time step. We can use the following equations to calculate these probabilities:

$$I_i^{(t)} = (1-\mu) I_i^{(t-1)} + \left(1 - I_i^{(t-1)} - R_i^{(t-1)}\right) \left[1 - \prod_j \left(1 - \lambda A_{ji} I_j^{(t-1)}\right)\right] \tag{2.2}$$

and

$$R_i^{(t)} = R_i^{(t-1)} + \mu I_i^{(t-1)} \ . \tag{2.3}$$

In the above two equations, $I_i^{(t)}$ and $R_i^{(t)}$ represent the probability of node i being in state I or R at time t respectively. $S_i^{(t)}$ can be found trivially since $S_i^{(t)} + I_i^{(t)} + R_i^{(t)} = 1$. Given an

initial spreader set $\mathcal{X}$, $I_i^{(0)} = 1$ for all $i \in \mathcal{X}$ and $S_i^{(0)} = 1$ for $i \notin \mathcal{X}$. Equation 2.2 means that for a node to be in the infected state at time t there are two options: Either it was infected previously and did not recover at time $t-1$, or it was susceptible and was infected by one of its infected neighbors at $t-1$. Equation 2.3 means that for a node to be in state R at time t, it either had to be in state R at time $t-1$ or it was in state I and recovered at $t-1$.

2.2.2 Susceptible-Infected-Recovered model on temporal networks

The spreading dynamics on temporal networks have a very important difference from the dynamics on static networks. Spreading happens between nodes through edges, and edges in static networks are ever-present. However, by their nature, temporal networks have edges that appear at certain periods and then disappear. This difference needs to be considered if a spreading model used on a static network were to be used on a temporal network.

Due to the reasons mentioned above, we use an SIR model that is adapted to temporal networks. We again have three states, S, I, and R, and two parameters λ and μ. We have a temporal network similar to the one presented in Section 2.1.2. Starting from the first layer we have initial spreaders in state I and all the other nodes are in state S. Furthermore, we assume that one time unit in the spreading process to be equal to one time unit of network evolution. This means that in the first layer, nodes in state I will try to infect their neighbors in state S, and will succeed with probability λ. After that, the nodes that attempted to infect others will recover and change their state to R with probability μ. This step is the same as in the original SIR model. However, after the infection and recovery attempts, we will move to the second layer before nodes attempt any further infections and recoveries. This is the point where temporality is introduced into the model. The model stops either when there are no nodes left in state I, or when the infection reaches the last layer in the temporal network. The second condition is another difference between the SIR model for static and temporal networks. In temporal networks, there is a time limit for interactions because they will end at some point, which will be the last layer of the temporal network. The total number of

nodes in state I and R at the end of the spreading dynamics gives the outbreak size. A toy example of SIR model applied on a temporal network can be seen in Figure 2.3.

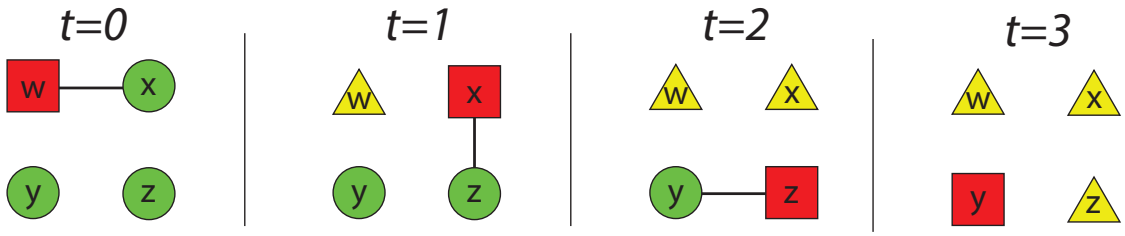

Figure 2.3: **SIR model on temporal networks.** Green circles represent susceptible nodes, red squares represent infected nodes, and yellow triangles represent recovered nodes. We set $\lambda = 1$ and $\mu = 1$. Node w infects node x in $t = 0$, and then recovers. After that the network changes and spreading dynamics happen on $t = 1$ for a single step again, *i.e.*, node x infects node z and then recovers. The spreading dynamics stops when it reaches $t = 3$.

Similar to Equations 2.2-2.3, a mean-field approximation can be made:

$$I_i^{(t)} = (1-\mu)I_i^{(t-1)} + \left(1 - I_i^{(t-1)} - R_i^{(t-1)}\right)\left[1 - \prod_j \left(1 - \lambda A_{ji}^{(t-1)} I_j^{(t-1)}\right)\right] \tag{2.4}$$

and

$$R_i^{(t)} = R_i^{(t-1)} + \mu I_i^{(t-1)} \ . \tag{2.5}$$

The only difference in Equation 2.4 compared to Equation 2.2 is the term A_{ij} in Equation 2.2, which is replaced with the term $A_{ij}^{(t-1)}$ in Equation 2.4 as the adjacency matrix is time dependent and can change from one layer of the temporal network to the next. Equation 2.5 is the same as Equation 2.3.

2.2.3 Independent cascade model

The Independent Cascade model (ICM) (9, 37) is a special case of the SIR model. In ICM, μ is set to 1. This means that at a given time step, a node in state I tries to infect its neighbors in state S with probability λ, and then recovers immediately by changing its state from I to R. A simplified version of Equations 2.2-2.3 can be written by setting $\mu = 1$, *i.e.*,

$$I_i^{(t)} = \left(1 - I_i^{(t-1)} - R_i^{(t-1)}\right)\left[1 - \prod_j \left(1 - \lambda A_{ji} I_j^{(t-1)}\right)\right] \tag{2.6}$$

and

$$R_i^{(t)} = R_i^{(t-1)} + I_i^{(t-1)} \ . \tag{2.7}$$

Equation 2.6 tells that a node that is in state I at time t has to be in the susceptible state at time $t-1$ and infected by one of its neighbors in state I. Equation 2.7 means all nodes in states I and R at time $t-1$ will be in state R at time t.

2.3 Influence maximization

In a complex network, the spreading probability is an important determinant in terms of the outbreak size at the end of dynamics. For example, in an SIR model, if we have $\lambda = 1$, all nodes reachable from the initial spreaders will be infected. However, the spreading probability is not the only important factor on the spreading in a network. The positions of the initial spreaders are also important. Intuitively, if central nodes are selected as initial spreaders, the outbreak will reach many more nodes compared to initial spreaders selected among peripheral nodes. This intuition suggests that finding a good set of initial spreaders is also important if one wants to spread a piece of information in a network to as many nodes as possible.

The problem of selecting the initial spreaders that will maximize the outbreak size is called the influence maximization problem. Traditionally, the problem is formulated as finding a set $\mathcal{X}$ of initial spreaders called the seed set, where the size of the seed set $|\mathcal{X}|$ is fixed, such that the average value of the outbreak size is maximized for the given infection probability λ and recovery probability μ. This can be expressed as

$$\mathcal{X} = \arg\max_{\mathcal{A} \subset V} \langle O(\mathcal{A}) \rangle \tag{2.8}$$

where $\langle O(\mathcal{A}) \rangle$ is the average outbreak size of seed set $\mathcal{A}$, whose size is fixed. The problem

was first formulated by Domingos and Richardson (38). It was later generalized by Kempe *et al.* (9), who also showed that the problem is NP-hard, meaning it can be solved optimally only for small networks and small seed set size $|\mathcal{X}|$ in practice.

2.3.1 Critical threshold

The solutions for the influence maximization problem are dependent on λ and μ. For now, let's assume that we have ICM as the spreading model, *i.e.*, SIR model with $\mu = 1$. In this case, if we set $\lambda = 1$ or $\lambda = 0$, the solutions are trivial. Assuming that we have a connected network, in the former, no matter which nodes are in the seed set $\mathcal{X}$, the outbreak will reach to all nodes. In the latter, the outbreak size will be exactly equal to $|\mathcal{X}|$ as no further infection is possible, and this does not depend on the identities of nodes in $\mathcal{X}$. In these two extreme scenarios, the uncertainty of the prediction of the outbreak size is zero. The prediction of performances for different sets of $\mathcal{X}$ is non-trivial when the uncertainty of the outbreak size is maximal. The point where this value is maximal is called the critical threshold value λ_c (more specifically, it can be denoted as $\lambda_c(\mu)$ since it is a function of μ). Due to the non-triviality of the solutions to the influence maximization problem, we do most of our analysis at criticality unless we say otherwise. We call the point of criticality, *i.e.*, $\lambda = \lambda_c$, the critical regime. When we have $\lambda < \lambda_c$, we call this the subcritical regime, and $\lambda > \lambda_c$ is called the supercritical regime.

In order to find the critical threshold of a static network, we rely on the mapping between bond percolation and ICM (39), and apply the Newman-Ziff algorithm (40, 41). The algorithm starts from a configuration with no edges. After that, each edge is selected randomly and added to the network until all edges are added back to the network. During this process, the size of the largest connected component $C(p)$ is monitored as a function of p, where p is the fraction of edges added to the network, called the bond occupation probability. The process is repeated R times because the algorithm is stochastic. The percolation strength

P_∞ and the susceptibility S are estimated as

$$P_\infty(p) = \frac{1}{NR}\sum_{r=1}^{R} C_r(p) \tag{2.9}$$

$$S(p) = \frac{(1/N^2R)\sum_{r=1}^{R} C_r(p)^2 - P_\infty(p)^2}{P_\infty(p)} \tag{2.10}$$

where N is the number of nodes in the network, and $C_r(p)$ is the size of the largest connected component in the r^{th} realization for a given bond occupation probability p. The estimation of the critical threshold value λ_c is made by determining the p value where susceptibility reaches its maximum value, *i.e.*,

$$\lambda_c = \arg\max_p S(p). \tag{2.11}$$

The Newman-Ziff algorithm is used for finding the critical threshold in static networks. For temporal networks, a similar approach is used to find the critical threshold. Starting from each node in the first layer of a temporal network, SIR model is run to estimate the outbreak size for given λ and μ values. This process is repeated multiple times due to its stochastic nature. For a given μ value, the critical threshold $\lambda_c(\mu)$ is estimated by finding the λ value which maximizes the ratio of standard deviation and average of the outbreak size as

$$\lambda_c(\mu) = \arg\max_\lambda \frac{\sqrt{\frac{\sum_{i=1}^{N}(O_{\lambda,\mu}(i)-\bar{O}_{\lambda,\mu})^2}{N}}}{\bar{O}_{\lambda,\mu}} \tag{2.12}$$

where $O_{\lambda,\mu}(i)$ is the outbreak size of node i for given λ and μ values, and $\bar{O}_{\lambda,\mu} = \frac{\sum_{i=1}^{N} O_{\lambda,\mu}(i)}{N}$.

2.4 Methods for identifying influential spreaders

In the previous sections, we have established that both λ value for a given μ, and the set of initial spreaders $\mathcal{X}$ are important factors in the spreading dynamics. The influence maximization problem is maximally non-trivial to solve at criticality, thus we set $\lambda = \lambda_c$.

However, the question of finding the nodes of the seed set $\mathcal{X}$ is still open. There are many methods proposed in the literature to identify influential spreaders in complex networks. Some rely on optimization methods (42–55), while others rely on heuristic methods (56–79) that use the topology of networks. In the following sections, we present some of these methods that we have used in this thesis.

2.4.1 Submodularity and greedy optimization

In their work, Kempe *et al.* (9) have shown that the problem of influence maximization is NP-hard. In the same paper, they also show that greedy optimization is guaranteed to find solutions at worst $1 - 1/e \simeq 63\%$ of the optimum. This result applies for the SIR model and LTM on static networks. The optimality guarantee comes from the fact that the influence function for SIR model is submodular. A function is submodular when the inequality

$$f(\mathcal{A} \cup v) - f(\mathcal{A}) \geq f(\mathcal{B} \cup v) - f(\mathcal{B}) \tag{2.13}$$

holds $\forall \mathcal{A}, \mathcal{B}, v$ such that $\mathcal{A} \subseteq \mathcal{B} \subseteq V$, $v \in V$, and f is the influence function. In simpler words, for any set $\mathcal{B}$ and any of its subsets $\mathcal{A}$, the increase in the outbreak size by adding node v to set $\mathcal{A}$ has to be at least as big as the increase in the outbreak size by adding node v to set $\mathcal{B}$.

Given a submodular influence function, the greedy optimization can be used to find solutions that are at most 63% away from the optimum. Let $\mathcal{X}_k = \{x_1, x_2, \ldots, x_k\}$ be the seed set identified by the greedy algorithm at stage k. We initialize $\mathcal{X} = \emptyset$. Then, for $k > 0$, x_k is selected using

$$x_k = \arg \max_{v \notin \mathcal{X}_{k-1}} f(\mathcal{X}_{k-1} \cup v). \tag{2.14}$$

In other words, the set $\mathcal{X}$ is built by selecting the node that gives the maximum marginal gain of outbreak at each step. At each stage, the influence $f(\mathcal{X}_{k-1} \cup v)$ of nodes $v \notin \mathcal{X}_{k-1}$

is numerically estimated using independent simulations. The necessity to recalculate the marginal influence of each node at each step causes the algorithm to have a high computational complexity, which scales cubically with the number of nodes.

2.4.2 Bond percolation

The greedy optimization proposed in (9) requires calculating the marginal spreading of each node $v \notin \mathcal{X}_{k-1}$. This calculation has to be repeated after adding a node to the seed set since the marginal outbreak of each node could change. This creates a high computational burden, which some methods try to decrease using different approaches (19). One of those methods has been proposed by Chen *et al.* (18). Their proposed algorithm makes use of the direct mapping between bond percolation and SIR (39). For the ICM, the algorithm works as follows. First, multiple instances of bond percolation are created for a given spreading probability λ. This means that in an instance, each edge exists with probability λ. After creating the bond percolation instances, the influence of each node is estimated. The influence of a node in a single instance is equal to the size of the cluster it belongs to. This is because since each edge exists with probability λ, the existing edges in an instance can be interpreted as successful carriers of the spreading (if an edge is not present in a bond percolation instance, this means the spreading did not pass through that edge). In order to estimate the expected influence of a node in the network, we take the average of its cluster sizes over all bond percolation instances. Once the influence of each node is estimated, the node with the largest expected influence can be selected as the first seed. This first step requires a similar amount of computation as the simple greedy algorithm. However, to select the seeds after the first one, the same bond percolation instances can be used to decrease the amount of computation necessary. In order to select the second seed (and seeds after that), we remove the clusters that the previous seed belonged to in the bond percolation instances. Now, we can calculate the marginal influence of each node, *i.e.*, influence of a node when added on top of the seed set, by taking the average of the sizes of clusters the node belongs

to. The seed set can be filled by repeating this procedure. This method of using the same bond percolation instances to select multiple seeds makes the algorithm faster than simple greedy algorithm without any loss of accuracy.

2.4.3 Greedy optimization on temporal networks

The greedy algorithm provides solutions with an optimality guarantee in static networks when the spreading model is the SIR model. The same algorithm can be used for temporal networks, as the algorithm does not depend on the characteristics of edges or nodes to be implemented. However, when we use the network representation presented in Section 2.1.2 and the spreading model in Section 2.2.2, the influence function is no longer submodular, meaning there is no optimality guarantee on the performance of greedy algorithm. An example proving that the inequality in Equation 2.13 does not always hold in temporal networks is shown in Figure 2.4.

Figure 2.4 means that greedy algorithm does not have a performance guarantee when used in the given setting. However, it can still be used without an optimality guarantee, and its performance can be assessed and compared with other methods.

2.4.4 Heuristic methods

The optimization methods mentioned in the previous sections use information on the topology of the network and the spreading dynamics. There are also methods to find influential spreaders, using only the topological information of the network. Such methods are called heuristic methods. In general, these methods work by defining a measure, calculating this measure for each node, and selecting the top $|\mathcal{X}|$ nodes that have the maximum value for the measure.

Degree (D) - An example for a widely used heuristic method is the degree centrality. In this method, each node is assigned a score that corresponds to the number of neighbors it

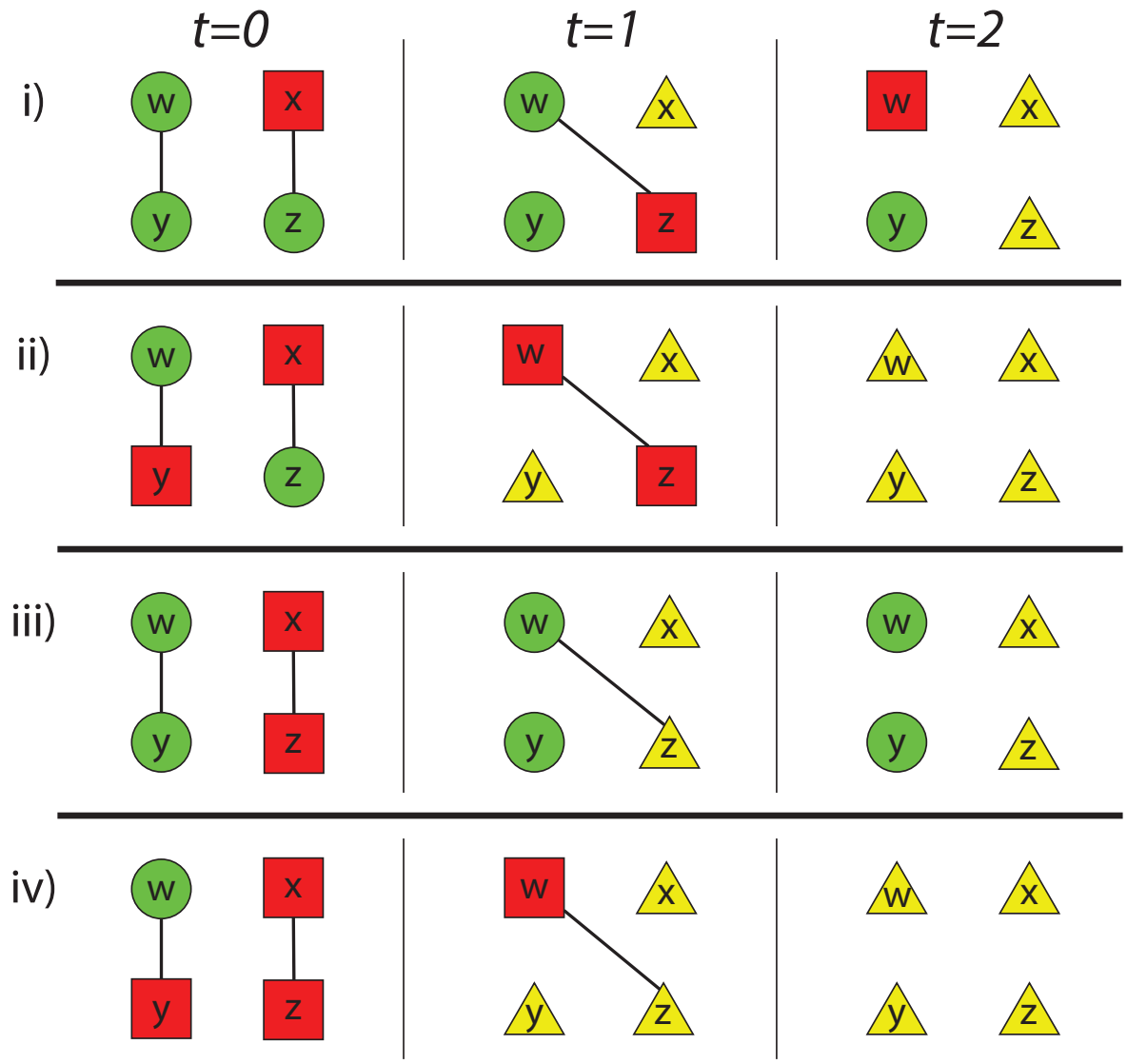

Figure 2.4: **Violation of the submodularity condition on temporal networks.** A toy example showing the violation of the inequality in Equation 2.13 in temporal networks. For simplicity, we consider a deterministic case of the SIR model with parameters $\lambda = \mu = 1$. Let $\mathcal{A} = \{x\}, \mathcal{B} = \{x, y\}$, and $v = z$. Green circles represent susceptible nodes, red squares represent infected nodes, and yellow triangles represent recovered nodes. We have the outbreak sizes in each panel as $f(\mathcal{A}) = 3$ in (i), $f(\mathcal{B}) = 4$ in (ii), $f(\mathcal{A} \cup v) = 2$ in (iii), and $f(\mathcal{B} \cup v) = 4$ in (iv), thus violating the condition for submodularity.

has. Formally, we can define the degree centrality of a node i as

$$D(i) = \sum_j A_{ij} \tag{2.15}$$

where A is the adjacency matrix of the network and A_{ij} is defined as in Equation 2.1.

Adaptive degree (AD) - Chen *et al.* (18) have proposed a variant of degree centrality, called adaptive degree. In adaptive degree, after adding a node to the set of seeds, it is removed from the network and degrees are calculated again to select the next spreader. This is repeated until the set is filled. The adaptive degree of a node i at step k can be defined as

$$AD_k(i) = \sum_{j \notin \mathcal{X}_{k-1}} A_{ij} \tag{2.16}$$

where $\mathcal{X}_{k-1}$ is the seed set at step $k-1$.

Betweenness (B) - Nodes that are connecting different clusters in a network can also have important structural roles. Betweenness centrality (80) is designed to find such nodes in a network. It is defined as the fraction of shortest paths that a node lies on. Formally, the betweenness centrality of node i can be defined as

$$B(i) = \sum_{\substack{jk \\ i \neq j \neq k}} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \tag{2.17}$$

where σ_{jk} is the number of shortest paths between two nodes j and k, and $\sigma_{jk}(i)$ is the number of shortest paths between j and k that pass through node i.

Closeness (C) - Besides nodes that act as bridges between communities, nodes that are close to all other nodes can also play an important role in the dynamics of a network. The idea is if a node is the closest one to all the other nodes on average, the effort to reach other nodes is minimal in the given network and thus can be effective in spreading processes. The average of shortest paths from node i to all the other nodes in a network gives its closeness centrality (81), and it is defined as

$$C(i) = \frac{N-1}{\sum_{j \neq i} d_{ij}} \tag{2.18}$$

where N is the number of nodes in the network, and d_{ij} is the length of the shortest path

between nodes i and j. Closeness centrality is actually defined as the reciprocal of average shortest path, in order to have higher values indicating nodes that are closer to all the other nodes.

Eigenvector (E) - The degree centrality indicates how many nodes can directly be reached. One important point it does not take into account is the structural importance of the neighboring nodes. Eigenvector centrality (82) is proposed to take this aspect into account. The idea is that if the neighbors of a node are structurally important nodes, then the node itself will also be important as it can directly affect its neighbors. The eigenvector centrality of a node can be found by solving

$$Ax = \lambda x \tag{2.19}$$

for the largest possible value of λ. The i^{th} element in vector x gives the eigenvector centrality of node i, *i.e.*, $E(i)$.

Katz (K) - A similar idea to eigenvector centrality is used for Katz centrality (83). Here, the importance of a node relies on the importance of its neighbors. The centrality value of node i is defined as

$$K(i) = \alpha \sum_j A_{ij} K(j) + \beta \tag{2.20}$$

where α is a constant such that $\alpha < 1/\lambda_{max}$, λ_{max} is the largest eigenvalue of the adjacency matrix, and β is a constant factor added to the initial centrality value for each node.

PageRank (PR) - In Katz centrality, the score of a node i is passed to all of its neighbors regardless of $D(i)$. Google's PageRank (84) offers a solution to this problem by dividing the score equally among the neighbors. PageRank is defined as

$$PR(i) = \frac{1-\alpha}{N} + \alpha \sum_j \frac{A_{ij} PR(j)}{D(j)} \tag{2.21}$$

where α is a constant generally set equal to 0.85.

Non-backtracking (NB) - Another centrality measure is the non-backtracking centrality

(85). The non-backtracking centrality can be calculated using

$$M = \begin{pmatrix} A & I - Z \\ I & \emptyset \end{pmatrix} \tag{2.22}$$

where I is the identity matrix, Z is a diagonal matrix with $Z_{ii} = D(i)$. For the largest eigenvalue of matrix M, we solve to find its eigenvector x. The i^{th} element of vector x gives the non-backtracking centrality of node i, *i.e.*, $NB(i)$. Note that $|x| = 2N$, but the last N elements of vector x can be ignored for our purposes.

K-shell (KS) - In most networks, the nodes can be divided into two groups, those that are in the core of the network and those that are at the periphery (86–88). A centrality measure called k-shell is used to quantify the coreness of nodes in a network (56). The algorithm recursively removes nodes with degree k at each turn, starting from $k = 1$, until no nodes are left with degree k. The nodes removed are assigned the k-shell value of k, *i.e.*, $KS(i) = k$. By incrementally increasing the k value, the scores are assigned to each node until all nodes are removed from the network.

LocalRank (LR) - Another heuristic proposed is the LocalRank (89). Degree centrality considers only the nodes that can be reached directly. The LocalRank measure considers a wider area. Formally, it can be defined as

$$Q(j) = \sum_j D^{(2)}(j) A_{ij} \tag{2.23}$$

$$LR(i) = \sum_j A_{ij} Q(j) \tag{2.24}$$

where $D^{(2)}(i)$ represents the number of nodes that can be reached in 2 steps from node i.

H-index (H) - There are also other measures that have been proposed for different tasks, which can be adapted to find influential spreaders in networks. One such example is the H-index (90, 91). This measure has been proposed to quantify the quality of a researcher

using the number of citations they received for each of their publications. It is defined as the maximum value for h such that a scientist has h publications with at least h citations. For networks, we can define a similar measure. Node i with H-index equal to h, $H(i) = h$, would have at least h neighbors with $D(j) \geq h$. The H-index score of a node $H(i)$ can be defined as h maximized over

$$\sum_j A_{ij} \Theta\Big(D(j) - h + 1 \Big) \geq h \tag{2.25}$$

where Θ is the Heaviside function, which returns $\Theta(x) = 1$ if $x > 0$ and 0 otherwise.

CoreHD (CD) - The problem of influence maximization has similarities with the optimal percolation problem (92), even though they are not exactly the same problem (93). Because of this similarity, it is natural to consider methods designed for optimal percolation to find influential spreaders. One such method is called CoreHD (94). The method is very similar to adaptive degree. In each step, the node with the greatest degree is selected as a seed, removed from the network, and the degrees are recalculated. The only difference of CoreHD from adaptive degree is that the input is not the whole network but its 2-core. The 2-core of a network is extracted by removing all nodes that have a single edge until all nodes left in the network have $D(i) > 1$. Formally, we can define the CoreHD score of a node i at step k as

$$CD_k(i) = \sum_{j \notin \mathcal{X}_{k-1}} A_{ij}^{(2-core)} \tag{2.26}$$

where $A^{(2-core)}$ is the adjacency matrix for the 2-core of the network.

Collective influence (CI) - Another method first proposed for the optimal percolation problem is collective influence (92). This method is an adaptive method that takes into account the nodes that can be reached from a node i in exactly l steps (and not less). At a

given time step k, the collective influence score of a node i is defined as

$$CI_k^l(i) = (AD_k(i) - 1) \sum_{j \in N^{(l)}(i)} (AD_k(j) - 1) \tag{2.27}$$

where $N^{(l)}(i)$ is the set of nodes that can be reached in exactly l steps from node i. When $l = 0$, we have $CI_k^0(i) = AD_k(i) - 1 \ \forall k$.

Explosive immunization (EI) - Last method used in this thesis is explosive immunization (EI) (95), which relies on the explosive percolation concept proposed by Achlioptas *et al.* (96). The algorithm starts with all nodes vaccinated, *i.e.*, an empty graph. After that, at each step, the node that is the most 'harmless' is added to the network. The idea is keeping the largest connected component as small as possible. In order to find the most harmless node, the algorithm uses two different approaches, one for below the immunization threshold q_c and one for above it. q_c represents the smallest fraction q of nodes vaccinated for which the size of the largest cluster of susceptible nodes is zero as $N \to \infty$. Above q_c, a node is selected such that the size of clusters to be connected is minimized. Below q_c, the goal is to find a node that will connect as few clusters as possible. At the end of this algorithm, a ranking of nodes is given, which can be used to create a seed set, consisting of the most 'harmful' nodes in the network.

Random (RN) - One final method to select the initial spreaders would be selecting them randomly. This method can give a lower bound on the achievable performance. Any proposed heuristic method should perform better than random selection, otherwise there is no reason to use it as it will have a computational complexity at least as high as random selection.

In Figure 2.5, we show a simple network. We can use this to illustrate the calculations of the introduced centrality measures, specifically for node x.

- $D(x) = 3$. Nodes w, y, and t are the neighbors of node x.
- $AD_k(x) = D(x) = 3$ for $k = 0$. For $k > 0$, $AD_k(x)$ depends on which neighbors of node x are in the seed set. For example, if only node w is in the seed set at step $k = 1$,

then $AD_1(x) = 2$. If node x is in the seed set itself, than $AD_k(x)$ is not defined.

- $B(x) = 3.5/6 = 0.58$. Node x lies on the unique shortest paths between node t and all the remaining three nodes. There are two shortest paths between nodes w and z, and node x lies on one of them. This makes the numerator of Equation 2.17 equal to $3 + 1/2 = 3.5$. The denominator calculates the total number of shortest paths between other nodes, which is 6 as there are 4 nodes other than node x.

- $C(x) = 4/5 = 0.8$. The numerator in Equation 2.18 is equal to $N - 1 = 4$. The denominator is the sum of shortest path lengths from node x to all the other nodes. We have $d_{xw} = 1$, $d_{xy} = 2$, $d_{xz} = 1$, and $d_{xt} = 1$, thus the denominator is equal to 5.

- $E(x) = 0.56$. Solving Equation 2.19 for the largest eigenvalue λ_{max}, we find the corresponding eigenvector.

- $K(x) = 0.49$. Iterating over Equation 2.20 for $\alpha = 0.1$ and $\beta = 1.0$, the Katz centrality can be calculated.

- $PR(x) = 0.29$. Iterating over Equation 2.21 for $\alpha = 0.85$ PageRank value of a node can be calculated.

- $NB(x) = 0.52$. Solving for the largest eigenvalue λ_{max} of the matrix in Equation 2.22, the corresponding eigenvector can be found, of which the first N elements give the non-backtracking centrality values.

- $KS(x) = 2$. Node t is removed from the network in the first iteration as $D(t) = 1$. All other nodes remain in the network. In the next iteration all nodes with $D = 2$ are removed, which includes node x.

- $LR(x) = 18$. In Equation 2.24, we have $LR(x) = Q(w) + Q(z) + Q(t)$. Using Equation 2.23, we can calculate $Q(w) = D^{(2)}(x) + D^{(2)}(y) = 4 + 3 = 7$, $Q(z) = D^{(2)}(x) + D^{(2)}(y) = 4 + 3 = 7$, and $Q(t) = D^{(2)}(x) = 4$, which makes $LR(x) = 7 + 7 + 4 = 18$.

- $H(x) = 2$. The maximum value that h can have is 2. Node x has 2 neighbors with $D \geq 2$, but does not have 3 neighbors with $D \geq 3$.

- $CD_k(x) = 2$ for $k = 0$, since node t is not in the 2-core. For $k > 0$, it is the same as calculating adaptive degree, but only for the 2-core of the network.

- $CI_k^1(x) = 4$ for $k = 0$. In Equation 2.27, we have $AD_0(x) - 1 = 2$. The term for the sum is equal to $AD_0(w) + AD_0(z) + AD_0(t) - 3 = 2 + 2 + 1 - 3 = 2$. For $k > 0$, the value for $CI_k^1(x)$ depends on which nodes are selected as seeds.

- $CI_k^2(x) = 2$ for $k = 0$. In Equation 2.27, we have $AD_0(x) - 1 = 2$. The term for the sum is equal to $AD_0(y) - 1 = 2 - 1 = 1$. For $k > 0$, the value for $CI_k^2(x)$ depends on which nodes are selected as seeds.

- $EI(x)$: The ultimate goal in explosive immunization is to break apart the network as much as possible. If node x is removed first, the network will have 2 components. If any other node is removed first, the network will remain connected. Thus, node x will be the first node in the ranking.

In order to give an example for how the seed sets are created using heuristic methods, we can use degree and adaptive degree. When we use degree, the first node selected will be node x, since it has the largest degree. For the second seed, there are three options, nodes w, y, and z. The tie is broken randomly, and one of these three nodes is chosen to be the second seed. The third and fourth seeds will be the remaining two of the three nodes. The last seed will be node t, as it has the lowest degree. There are six possible seed sets to be constructed with degree centrality in this network, the first and last seeds are certain, but the ordering of the middle three seeds is random. If we use adaptive degree for constructing a seed set, the first seed will again be node x, since $AD_0 = D$. For the second seed, there is only one option this time, because we have $AD_1(y) = 2, AD_1(w) = AD_1(z) = 1$, and $AD_1(t) = 0$. For the last three seeds, we will have a tie, which needs to be randomly broken. Thus, there are

six possible seed sets, where the first two seeds are certain, but the last three can be in any order.

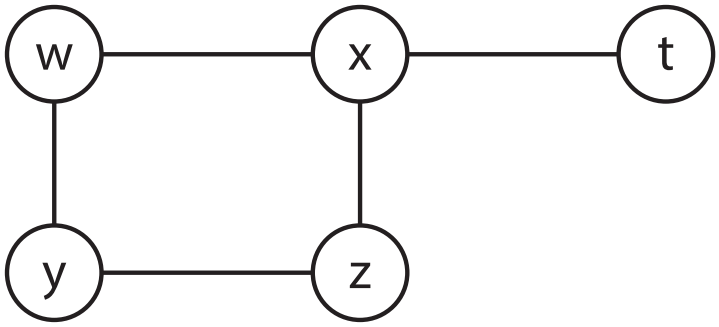

Figure 2.5: **A simple static network.** A toy network for illustrating the calculation of introduced centrality measures.

In Figure 2.6, we show the reference performances of several methods introduced. The optimal performance is achievable only using a brute-force search, *i.e.*, trying every possible combination for given seed set size $|\mathcal{X}|$. This is not possible as the number of combinations will be very large. The optimal performance can be approximated with greedy optimization, with a guarantee that the solution will be at worst $1-1/e$ of the optimal solution. This gives an upper bound on the performance of other methods. A random selection of nodes as seeds gives a lower bound for other methods as any method proposed should give better solutions than random selection. The worst case scenario is for $O(\mathcal{X}) = |\mathcal{X}|$, outbreak size of seed set equal to the size of the seed set. This happens when seeds can not infect any other nodes in the network. The area between the curves of greedy optimization and random selection is where performances of other methods are expected to be. The goal is to be as close to the curve of greedy optimization as possible.

2.4.5 Evaluating the performances of influential spreader identification methods

In order to evaluate the efficiency of solutions provided by a seed selection method, we need a quantifiable measure. To decrease the dependence on the seed set size, rather than using

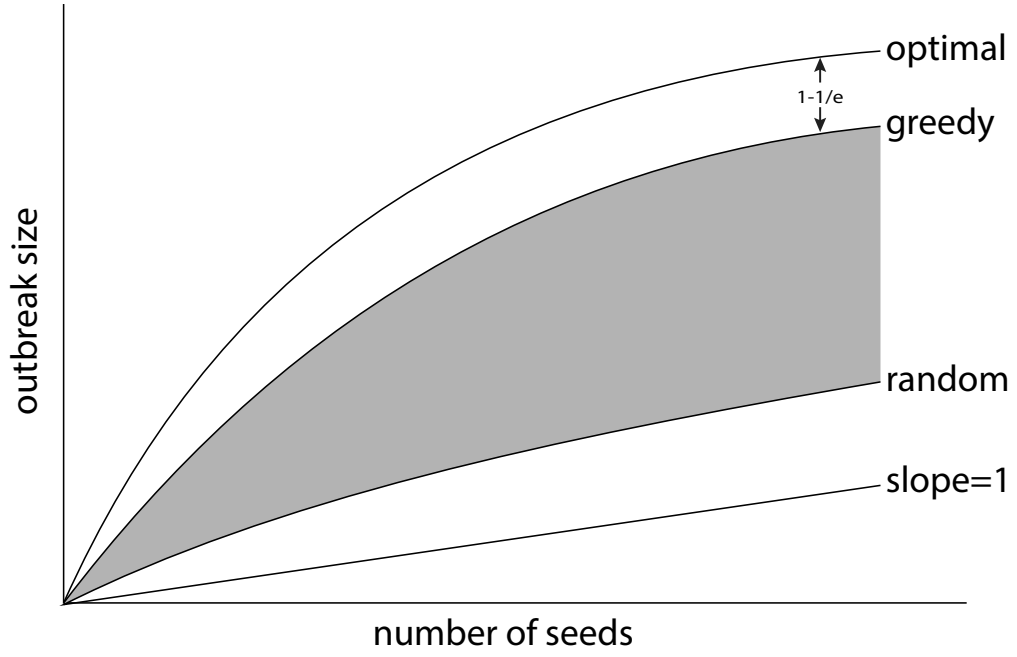

Figure 2.6: **Reference performances of methods for identifying influential spreaders.** A chart showing the expected performances of classes of seed selection methods. The curve at the top is the highest possible performance using the optimal set of seeds. The curve for greedy optimization shows the highest achievable performance in practice, which is at worst $1 - 1/e$ of the optimal performance. The curve for random selection shows the worst-case scenario, where the nodes in the seed set are selected randomly among the nodes in the network. Finally, the curve with slope equal to 1 shows when there is no node-to-node spreading, the outbreak size has to be at least as big as the size of the seed set. The grey shaded area shows the potential performances of heuristic methods, the closer to the curve of greedy optimization the better a heuristic method is.

the final outbreak size as a performance measure, we use the sum of outbreak sizes for $v \in \{1, \ldots, |\mathcal{X}|\}$, such that

$$q_m = \sum_{v=1}^{|\mathcal{X}|} \langle O(\mathcal{X}_m^v) \rangle \tag{2.28}$$

where $\langle O(\mathcal{X}_m^v) \rangle$ is the average outbreak size of the first v seeds found by method m. Using this measure, we then define the performance of a method relative to the performance of greedy optimization. We choose greedy optimization as a baseline because it gives an upper bound on the range of expected performance (see Figure 2.6). The relative performance of

a method m can be expressed as

$$g_m = \frac{q_m}{q_{GR}} \tag{2.29}$$

where q_{GR} is the performance of the seed set found by greedy optimization, *i.e.*, $\mathcal{X}_{(GR)}$, and defined as in Equation 2.28.

An important note is that the spreading model used can be stochastic. This is also true for seed selection methods, *e.g.*, if there is a tie for the score of multiple nodes, it is broken randomly. In the presence of stochasticity, the outbreak size $O(\mathcal{X}_m^v)$ needs to be calculated by averaging over multiple runs such that

$$\langle O(\mathcal{X}_m^v) \rangle = \frac{1}{R}\frac{1}{K}\sum_{k=1}^{K}\sum_{r=1}^{R} O^r(\mathcal{X}_m^{(v,k)}). \tag{2.30}$$

where K and R are the number of runs to decrease the statistical fluctuations arising from the stochasticity of the method and spreading model, and $O^r(\mathcal{X}_m^{(v,k)})$ is the outbreak size of the first v nodes found with method m in the k^{th} run for the seed selection method and r^{th} run for the spreading dynamic.

3 Systematic comparison between methods for the detection of influential spreaders in complex networks

3.1 Introduction

Kempe *et al.* have shown that the influence maximization problem can be solved using greedy optimization (9). The greedy optimization method uses information on the topology of the network and the dynamics of spreading model. After the seminal work of Kempe *et al.* other similar optimization techniques have been proposed with the main goal of increasing the computational efficiency while keeping the accuracy of the algorithm the same with simple greedy optimization (19, 97, 98). These optimization methods suffer from the limitation of not being applicable to large networks, because they require knowledge of the spreading model, which is often obtained through a large number of numerical simulations.

On large networks, heuristic methods can be used to find solution for the influence maximization problem. There are many examples of heuristic methods in the literature (10, 56, 99). Heuristic methods use information on the topology of the network, but they neglect the information on spreading dynamics. These methods are generally much faster than greedy optimization, but they also have lower effectiveness. The limitations of heuristic methods are two-fold. First, they do not account for the combined effects of nodes, as the seed set is built by combining the best individual spreaders, and their influence might overlap. Second, because they are pure topological methods and ignore the spreading dynamics, heuristic methods lack sensitivity to the variations in the parameters of the spreading model. Given the vast number of heuristic methods proposed in the literature to identify influential nodes in networks, one important question is how different those methods are from each other in terms of performance. Even more important, how far the performance of best heuristic methods are from the optimum, or at least from the largest achievable performance provided by the greedy algorithm? There is no clear answer for these questions in the literature, so we decided to fill this gap of knowledge.

In order to answer these questions, we report a systematic test on the performances of 16 heuristic methods and analyze their performances relative to an upper baseline found by the greedy algorithm and a lower baseline found by random selection. We do our analysis on a large corpus of 100 real-world networks, and quantify the performances of various heuristic methods using ICM as the spreading model. We show that many heuristic methods achieve comparable performances to greedy optimization. When 5% of nodes are selected as seeds, best performing methods achieve a performance within 90% of the performance of greedy optimization, meaning the room for potential improvement is small. We show that one way to achieve better performances is by relying on hybrid methods that combine multiple centrality metrics. We validate our results on hybrid methods on a small set of large-scale real-world networks.

3.2 Methods

3.2.1 Networks

We use a corpus of 100 real-world networks, undirected and unweighted, to study the performance of the heuristic methods. The networks in the corpus range in size from 100 to 30,000 nodes. These networks allow us to apply greedy optimization to find influential spreaders. We consider networks from different domains. In total, our corpus consists of 63 social, 16 technological, 10 information, 8 biological, and 3 transportation networks. Further details on the corpus of networks can be found in Tables A.1-A.3. Following the analysis on the corpus of 100 real-world networks, we validate our findings on 9 large real-world networks, ranging in size from 50,000 to more than 1,000,000 nodes. Details on the large networks are provided in Table 3.1.

3.2.2 Spreading dynamics

We use the ICM to simulate the spreading process as explained in Section 2.2.3. Since the model has a stochastic nature, all the results are obtained by taking the average value of 50

Network	N	E	λ_c	Ref.	url
Slashdot	51,083	116,573	0.0262	(100, 101)	url
Gnutella, Aug. 31, 2002	62,561	147,878	0.0956	(102, 103)	url
Epinions	75,877	405,739	0.0062	(101, 104)	url
Flickr	105,722	2,316,668	0.0142	(101, 105)	url
Gowalla	196,591	950,327	0.0073	(101, 106)	url
EU email	224,832	339,925	0.0119	(101, 103)	url
Web Stanford	255,265	1,941,926	0.0598	(107)	url
Amazon, Mar. 2, 2003	262,111	899,792	0.0940	(108)	url
YouTube friend. net.	1,134,890	2,987,624	0.0063	(101, 109)	url

Table 3.1: **Large-scale real-world networks.** From left to right, we report the name of the network, number of nodes in the giant component, number of edges in the giant component, critical value λ_c of the spreading probability, references to studies where the network was first analyzed, and the url where network data were downloaded.

independent numerical simulations for every given initial condition.

3.2.3 Methods for identifying influential spreaders

We consider 18 methods in total for identifying influential spreaders in networks. Each method outputs a ranking from the most influential node to the least influential node. This ranking is used to construct the seed set in a sequential manner. The computational complexity of the various methods differ based on the input of information and the way this information is handled for calculations. We report the computational complexity of each method in Table 3.2, where we also classify the methods into four groups.

The first group, *i.e.*, baseline methods, consists of greedy optimization and random selection. The greedy algorithm (Section 2.4.1) is the best performing method available, thus providing an upper bound on performance. For the greedy method, we rely on the algorithm by Chen *et al.* as explained in Section 2.4.2. On the other hand, we use random selection to have a lower bound on performance. This method ignores any available information on the network and selects random nodes from the network to construct the seed set.

The remaining 16 methods to identify influential spreaders, which are presented in Section 2.4.4, are purely topological methods in the sense that they rely only on topological

Group	Method	Abbreviation	Complexity
Baseline	Greedy	GR	cubic
	Random	RN	constant
Local	Degree	D	linear
	Adaptive Degree	AD	linear
Global	Betweenness	B	quadratic
	Closeness	C	quadratic
	Eigenvector	E	linear
	Katz	K	linear
	PageRank	PR	linear
	Non-backtracking	NB	linear
	Adaptive NB	ANB	quadratic
Intermediate	k-shell	KS	linear
	LocalRank	LR	linear
	H-index	H	linear
	CoreHD	CD	linear
	Collective Influence, $\ell = 1$	CI1	linear
	Collective Influence, $\ell = 2$	CI2	linear
	Expl. Immunization	EI	linear

Table 3.2: **Methods for the selection of influential spreaders.** We list basic details of all the methods for the detection of influential spreaders in complex networks. Each row of the table refers to a specific method. From left to right, we report the full name of the method, the abbreviation of the method name, and the computational complexity of the method. Computational complexities reported in the table are obtained under the realistic assumption that methods are applied to sparse networks where the number of edges scales linearly with the network size. Methods are further grouped into different categories: Baseline, local, global, and intermediate, depending on their properties.

information and not on dynamical information, such as the value of the spreading probability. These methods propose that the influence of a node is proportional to a centrality metric.

We call the second group of methods used the local methods. The values for these methods are computed relying only on information about their nearest neighbors. Degree (D) and adaptive degree (AD) centralities belong to this group.

The third group includes global methods. The computation of these methods rely on the complete knowledge of the whole network structure. Betweenness (B), closeness (C),

eigenvector (E), Katz (K), PageRank (PR), and non-backtracking (NB) centralities belong to this group. Additionally, we consider the adaptive variant of non-backtracking centrality (ANB).

The fourth and final group consist of methods that rely on intermediate topological information, such as the next nearest neighbors. K-shell (KS), LocalRank (LR), and H-index (H) belong in this group. We also classify CoreHD (CD), collective influence (CI), and explosive immunization (EI) in this group. These methods have been proposed for the optimal percolation problem, which has a similar nature to the influence maximization problem. We note that we consider two variants of CI with $l = 1$ (CI1) and $l = 2$ (CI2).

3.2.4 Evaluating the performance of the methods

All methods for identifying influential spreaders are potentially subjected to statistical fluctuations. This is because they may generate a different ranking at each run due to the presence of ties, and the fact that these ties are broken randomly. In order to account for these fluctuations, we run each method for $R = 10$ independent times. We indicate with $\mathcal{X}_m^{(t,r)}$ the top tN nodes identified by method m in its r^{th} instance on a network with N nodes. For every $\mathcal{X}_m^{(t,r)}$, we run the ICM 50 different times, and measure the average value of the outbreak size $O(\mathcal{X}_m^{(t,r)})$. We repeat this procedure for every instance r of the method and take the average value over R separate sets to calculate

$$V_m^{(t)} = \frac{1}{R} \sum_{r=1}^{R} O(\mathcal{X}_m^{(t,r)}). \tag{3.1}$$

In Figure 3.1, we show the relative size of the outbreak $V_m^{(t)}/N$ as a function of seed set size t.

As a measure for the performance of the method m for identifying TN top influential spreaders, we calculate the area under the curve of Figure 3.1 up to a pre-imposed T value

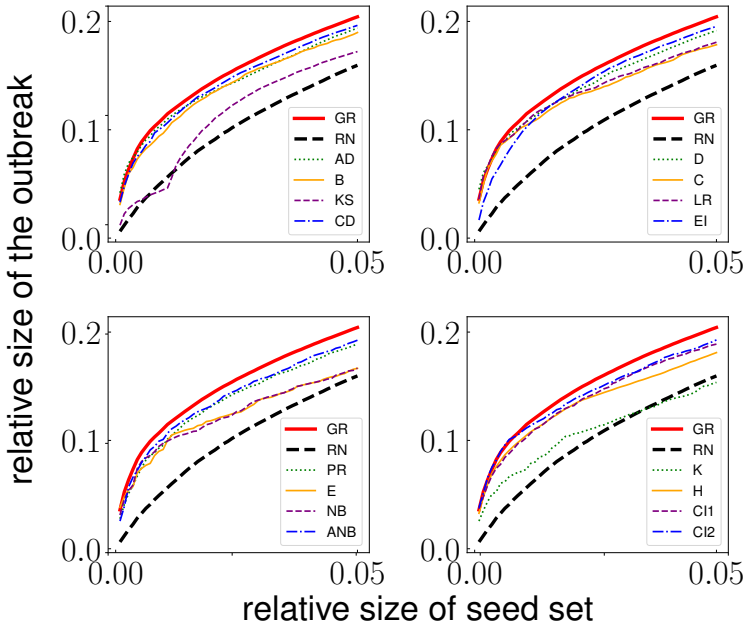

Figure 3.1: **Outbreak size as a function of seed set size.** Relative size of the outbreak as a function of the relative size of the seed set for an email communication network (1). The relative measures are obtained by dividing the outbreak size and seed set size by the number of nodes in the network. Relative measures allow us to compare results across networks with different sizes. The outbreak sizes are calculated with ICM dynamics at the critical threshold of the network, *i.e.*, $\lambda = \lambda_c$. Each panel in the figure includes the performances curves of 4 different heuristic methods along with the curves of greedy (GR) algorithm and random selection (RN). Similar plots for the same network for $\lambda = 0.5\lambda_c$ and $\lambda = 2\lambda_c$ can be found in Figures A.1 and A.2.

$$q_m^{(T)} = \frac{1}{N} \int_0^T dt V_m^{(t)} \tag{3.2}$$

As the size of the seed set is linearly dependent on the network size N, we can aggregate results obtained over the entire corpus of real-world networks in order to find an average performance of each method. Specifically, we set $T = 0.05$ for the main analysis. We also report the results for $T = 0.1$ in Figure A.3. There is no significant difference observed between the two different T values. Note that the problem of influence maximization is meaningful mostly for small T, as the seeding on networks is performed on a small portion of the system. We also perform the analysis considering $V_m^{(T)}$ as the main measure of performance, and report the results in Figure A.4. We do not observe any significant difference

from the results for $q_m^{(T)}$ with $T = 0.05$.

As mentioned before, the greedy algorithm provides an upper bound on the performance of other methods. We therefore use it to normalize the performance of the other methods. We consider two metrics of performance. The first measure is based on the comparison of outbreak sizes by a method m and by the greedy algorithm. In order to calculate this measure, we compute

$$g_m^{(T)} = \frac{q_m^{(T)}}{q_{GR}^{(T)}} , \tag{3.3}$$

where $q_{GR}^{(T)}$ is the value from Equation 3.2 for the greedy algorithm. We evaluate the performances relative to the greedy algorithm for each method over the whole corpus of networks. We summarize the results in Figure 3.2, where we display the cumulative distribution of this value. In order to obtain a single value for the performance of the method m over the entire corpus of networks, we define $\langle g_m^{(T)} \rangle$ as the average of the performance metric over all networks in the corpus.

The second metric we use to evaluate the performance of the various methods is named precision. This metric ignores the outbreak size and only focuses on the nodes identified as initial spreaders by a method. In the context of influence maximization, this second metric is not as important as the first one, since the only goal of influence maximization is maximizing the outbreak size, regardless of the specific nodes selected. However, this metric is useful for the characterization of the topological properties of the initial spreaders. For a given network, we evaluate the frequency $f_m^{(T,i)}$ of every node i appearing in the set of top TN initial spreaders according to the method m over $R = 10$ runs. Then, we compute the precision of the method m relative to the greedy algorithm as

$$p_m^{(T)} = \frac{1}{TN} \sum_{i=1}^{N} f_m^{(T,i)} f_{GR}^{(T,i)} . \tag{3.4}$$

We note that Equation 3.4 can be used to calculate the self-consistency of the greedy

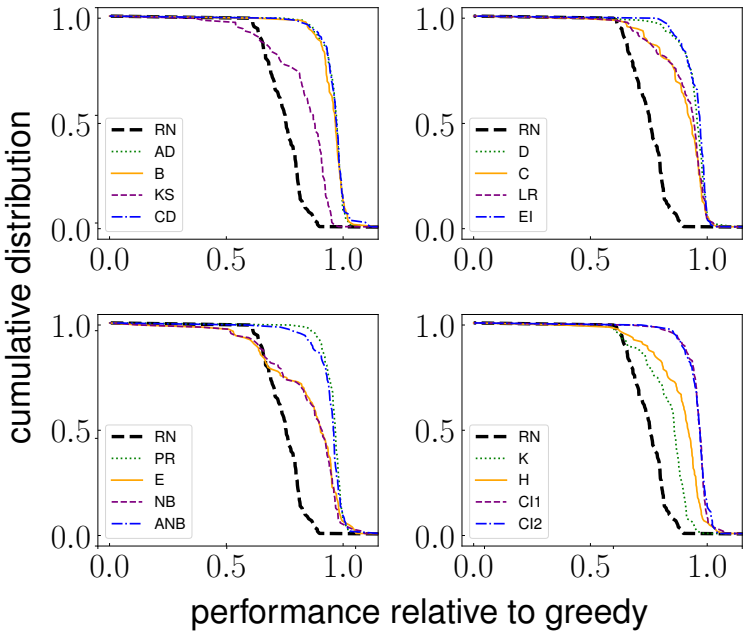

Figure 3.2: **Cumulative distribution of relative performance.** Cumulative distribution of the relative performance $g_m^{(T)}$ for $T = 0.05$. The metric of relative performance is defined in Equation 3.3. The distribution considers all 100 networks in the corpus. The outbreak size is calculated using ICM at criticality, *i.e.*, $\lambda = \lambda_c$. Similar plots for subcritical ($\lambda = 0.5\lambda_c$) and supercritical ($\lambda = 2\lambda_c$) regimes can be found in Figures A.5 and A.6.

algorithm by setting $m = GR$. This can be necessary because it is expected that $p_{GR}^{(T)} < 1$ since there is stochasticity in the selection of spreaders made by the greedy algorithm. In Figure 3.3, we display the cumulative distribution of the precision metric over the whole corpus of networks. We define the overall precision $\langle p_m^{(T)} \rangle$ of the method m as the average value of precision over the entire corpus. The overall precision value $\langle p_m^{(T)} \rangle$ tells us on average how similar the nodes selected by method m are to those selected by greedy algorithm.

3.3 Results

3.3.1 Individual methods

Using the metrics defined above, we test various methods of identifying influential spreaders with ICM dynamics over our corpus of real networks. Here, we remark that the identity and performance of true influential spreaders depend on the spreading probability λ of ICM, thus the performance of seed selection methods needs to be evaluated for different values

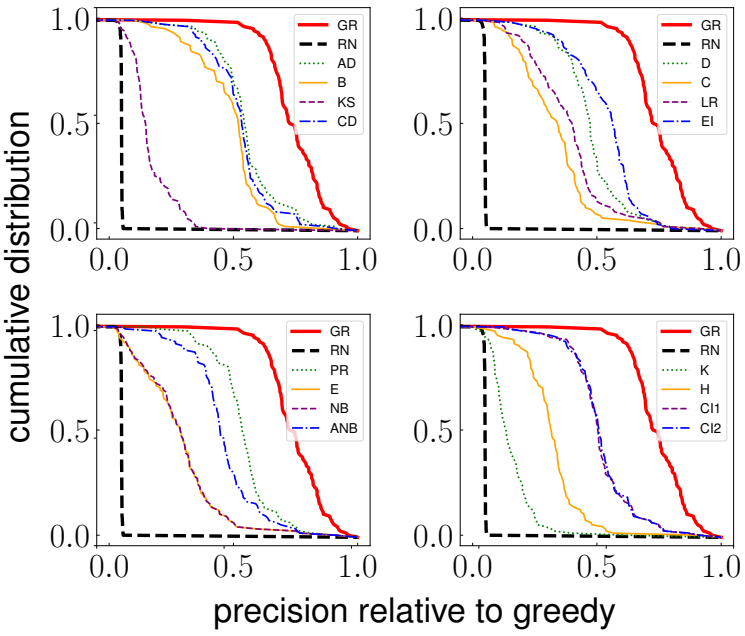

Figure 3.3: **Cumulative distribution of precision.** Cumulative distribution of the precision metric $p_m^{(T)}$ for $T = 0.05$ as defined in Equation 3.4. The distribution covers all 100 networks in the corpus. Results for greedy algorithm are obtained for ICM dynamics at criticality, *i.e.*, $\lambda = \lambda_c$. Similar plots for subcritical ($\lambda = 0.5\lambda_c$) and supercritical ($\lambda = 2\lambda_c$) regimes can be found in Figures A.7 and A.8.

of λ. The problem is maximally non trivial at $\lambda = \lambda_c$, so we focus our attention on ICM dynamics at the critical threshold λ_c, which we call the critical regime. Further, we consider two regimes below and above criticality, subcritical regime at $\lambda = \lambda_c/2$, and supercritical regime at $\lambda = 2\lambda_c$.

The results of our analysis are summarized in Figure 3.4. Here, every method is used to identify the top TN nodes in the networks, with $T = 0.05$ and N being the number of nodes in the network. In the figure, we represent each method m with the value pair $(\langle g_m \rangle, \langle p_m \rangle)$. We remark that both metrics are relative to the greedy algorithm. By definition, $\langle g_{GR} \rangle = 1$, but self-consistency of greedy algorithm is $\langle p_{GR} \rangle < 1$, meaning there is some variability in the seed sets identified by greedy algorithm. This variability is due to the existence of quasi-degenerate solutions, *i.e.*, different sets of seeds with similar outbreak sizes. Also, the statistical fluctuations from the stochasticity of the greedy algorithm might be exacerbating the degeneracy of solutions. The other important reference point is given by

random selection. By definition, we have $\langle p_{RN} \rangle \simeq T = 0.05$. $\langle g_{RN} \rangle$ values strongly depend on the dynamical regime.

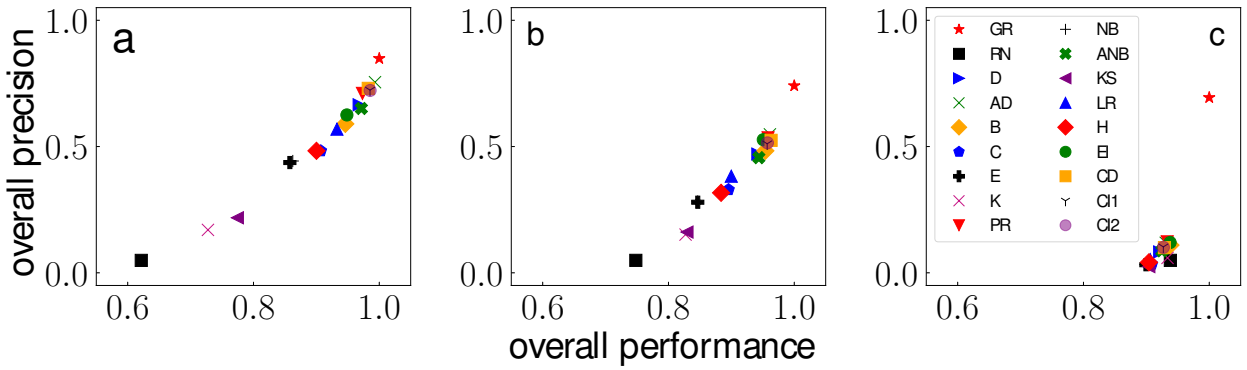

Figure 3.4: **Overall performance and overall precision of methods for the identification of influential spreaders in real networks.** Results are based on the systematic analysis of the corpus of 100 real-world networks. We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = 2\lambda_c$, (c) $\lambda = 2\lambda_c$. Each point in a panel corresponds to a single method. Every method is used to identify top TN nodes as spreaders with $T = 0.05$. Methods are characterized by the metrics of performance defined. Both metrics relate the performance of a method m to the performance of the greedy algorithm. Overall performance $\langle g_m \rangle$ shows the outbreak size of a method m relative to the outbreak size from greedy algorithm. Overall precision $\langle p_m \rangle$ quantifies the overlap between the seed sets identified by a method m and the greedy algorithm.

In the subcritical regime (Figure 3.4a), the two metrics $\langle g_m \rangle$ and $\langle p_m \rangle$ are highly correlated with each other. Adaptive degree (AD) outperforms all other methods in both metrics. Other methods that perform well are Degree (D), Adaptive Non-Backtracking (ANB), and PageRank (PR) centralities, and CoreHD (CD) and Collective Influence (CI) algorithms. In the critical regime (Figure 3.4b), the results are very similar to the subcritical regime. The most significant change is the decrease in the range of values for overall performance. In the supercritical regime (Figure 3.4c), the precision and performance of methods are no longer properly distinguishable.

One feature standing out from Figure 3.4 is that the overall performance is high in general. Most of the values for overall performance are above 0.9 for all regimes, and even the performance of random selection is always above 0.6. This observation is significant to weight the importance of greedy algorithms. While the solutions of greedy algorithms are

guaranteed to be not too far away from the true optimum, their performance can be almost achieved by simple, computationally much cheaper, and much more easily implemented purely topological methods.

The similarity in the performance between the various methods can be deduced from a pairwise comparison of seed sets identified by those methods across the entire corpus of real-world networks. The results of this analysis are summarized in Figure 3.5. We observe that the seed sets found by the top-performing methods have high similarity relative to the similarity with other methods. Methods with lower performance tend to select nodes that are not generally selected by other methods.

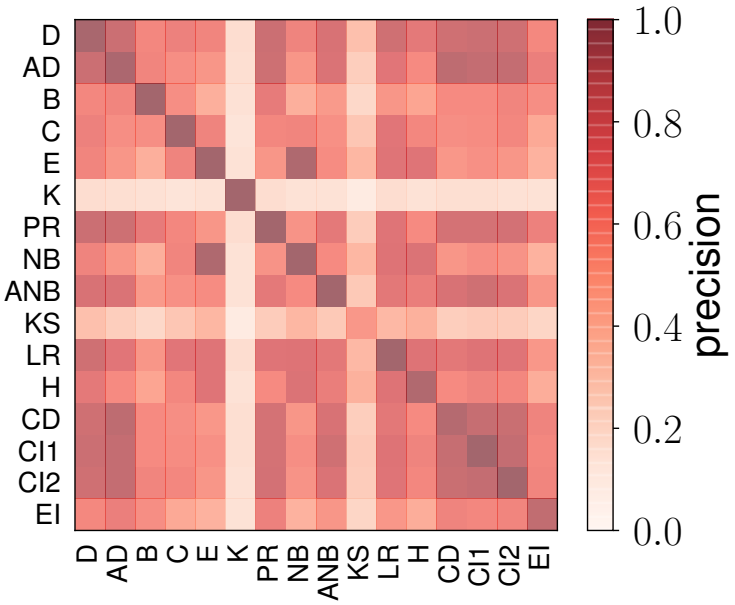

Figure 3.5: **Pairwise comparison among methods for the identification of influential spreaders.** For every pair of methods m_1 and m_2, we evaluate $p^{(T)}_{m_1,m_2}$ among the two seed sets of size TN using Equation 3.4. We then estimate the average precision value over the entire corpus of networks. In the figure, the darker colors represent higher values of precision.

We repeat the same analysis for different subsets of the corpus. Each subset consists of networks from similar domains (*e.g.*, social, technological, information, biological). The results of this analysis are reported in Figures 3.6-3.9. We did not find any significant differences from the outcomes of our main analysis.

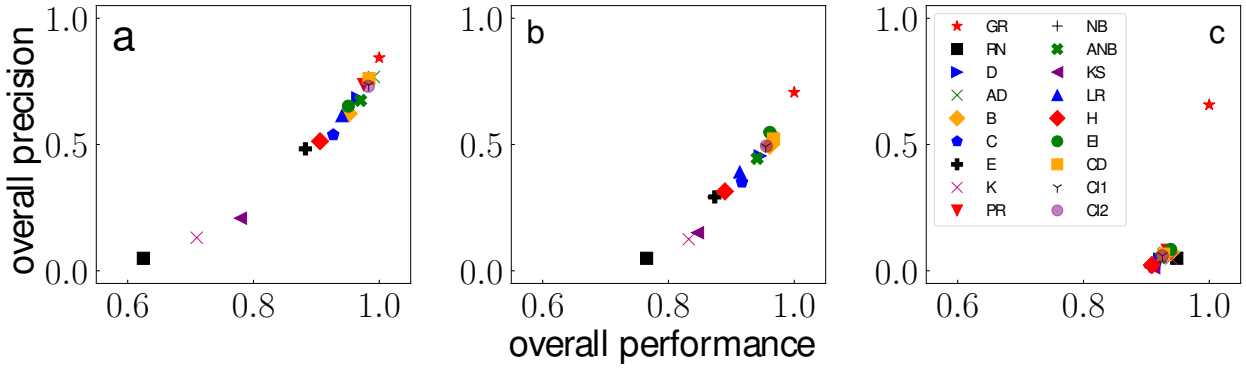

Figure 3.6: **Overall performance and overall precision of methods for the identification of influential spreaders in real social networks.** We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = \lambda_c$, (c) $\lambda = 2\lambda_c$.

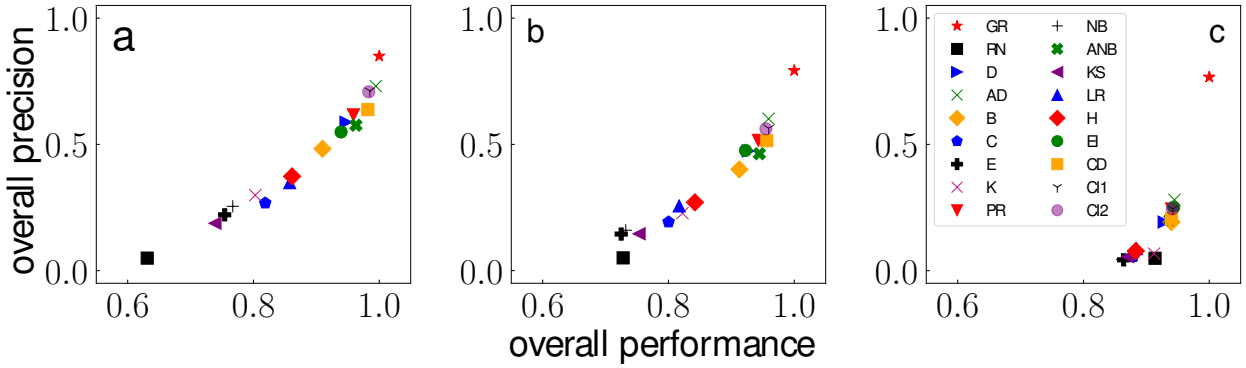

Figure 3.7: **Overall performance and overall precision of methods for the identification of influential spreaders in real technological networks.** We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = \lambda_c$, (c) $\lambda = 2\lambda_c$.

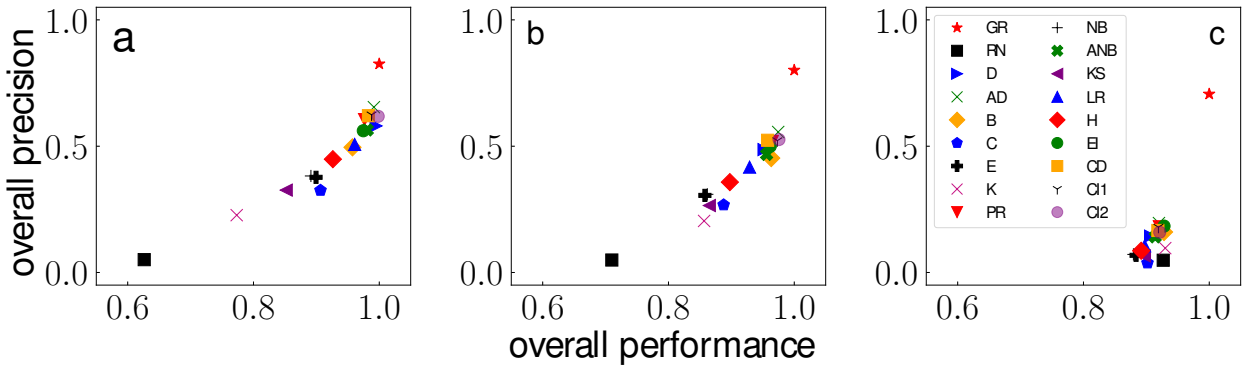

Figure 3.8: **Overall performance and overall precision of methods for the identification of influential spreaders in real information networks.** We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = \lambda_c$, (c) $\lambda = 2\lambda_c$.

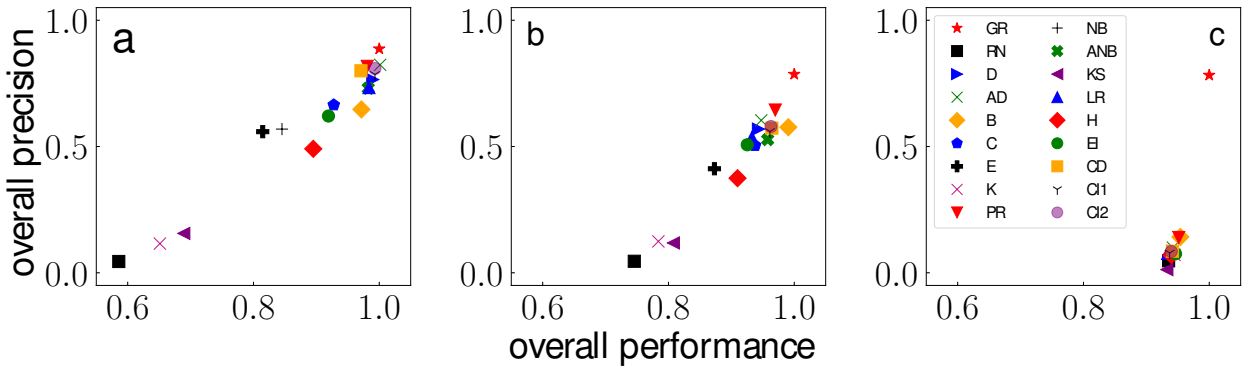

Figure 3.9: **Overall performance and overall precision of methods for the identification of influential spreaders in real biological networks.** We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = \lambda_c$, (c) $\lambda = 2\lambda_c$.

Furthermore, we consider synthetic networks created with the Barabási-Albert model. The networks considered consist of 5000, 10000, and 20000 nodes, and 10 of each has been created. Results are reported in Figure 3.10, and they are very similar to those obtained on our corpus of real-world networks. In summary, the main results hold for a variety of different settings.

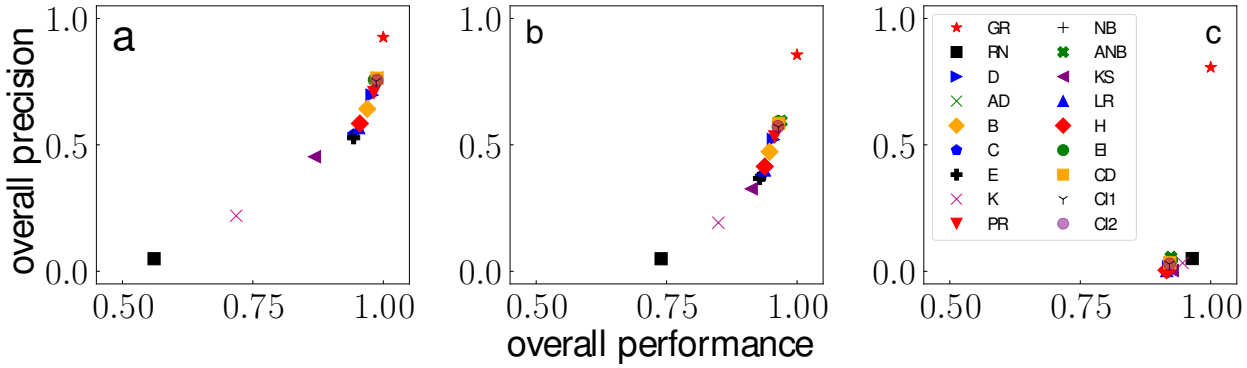

Figure 3.10: **Overall performance and overall precision of methods for the identification of influential spreaders in networks created with the Barabási-Albert model.** We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = \lambda_c$, (c) $\lambda = 2\lambda_c$.

3.3.2 Hybrid methods

We also test hybrid methods for the identification of top spreaders in networks. These methods are linear combinations of the individual methods considered so far. Specifically, we select

a certain number of individual methods to form a hybrid method $\mathcal{H} = \{m_1, m_2, \ldots, m_{\mathcal{H}}\}$. We associate each node i in a given network a score $s^i_{(\mathcal{H})}$, which is a linear combination of scores associated with individual methods, such that

$$s^i_{(\mathcal{H})} = \sum_{m \in \mathcal{H}} c_m s^i_m. \tag{3.5}$$

In Equation 3.5, s^i_m is the normalized score of node i in the network according to method m. The normalization has the purpose of making the scores of different methods have comparable magnitudes with each other. L^2-norm is used for normalization. We perform supervised learning where we use solutions of the greedy algorithm to estimate the linear coefficients c_m. Linear regression is used to find the best linear fit between $s^i_{(\mathcal{H})}$ and $f^{(T,i)}_{GR}$, *i.e.*, the fraction of times that node i in the network is identified in the top TN nodes by the greedy algorithm. The estimates for the coefficients are obtained using five-fold cross validation. The model is trained on 80% of the networks in the corpus. After training, the hybrid method $\mathcal{H}$ is tested on the remaining 20% of the corpus, where overall performance and overall precision are measured. We replicate this process for 1,000 times to quantify the uncertainty associated with both the estimates of the linear coefficients as well as the measured values of the performance metrics.

We test several hybrid methods that consist of two or three individual methods. In general, we combine methods that differ on the basis of their classification as local, global, and intermediate methods, shown in Table 3.2. Results for some of the tested hybrid methods are shown in Table 3.3. Our first observation is that, relative to the individual methods, there is an increase in the values of overall precision $\langle p_m \rangle$. This suggests that learned coefficients from the training set can be used meaningfully to mimic greedy optimization and identify influential spreaders. The overall performance $\langle g_m \rangle$ of hybrid methods also increases. We observe improvements of 2-5%, especially in the critical and supercritical regimes. Another point is that, when individual methods that provide similar information are combined together, one of the two gets the bigger part of the weight compared to the other.

For example, $\mathcal{H} = \{AD, B\}$ learned from the training set is almost equivalent to pure AD in both the subcritical and critical regimes. Finally, we observe that the coefficients of the linear combinations can be negative. For example, for the hybrid method $\mathcal{H} = \{AD, PR, LR\}$ we find $c_{LR} < 0$. This fact helps this hybrid method to outperform almost all other methods considered in the analysis, in both critical and supercritical regimes. We note that finding $c_{LR} < 0$ does not suggest the LR centrality is negatively correlated with node influence. This is only observed when LR is used in combination with other methods; LR is instead positively correlated with node influence when used as the only metric for the identification of influential spreaders, as it can be seen in Figure 3.4.

In order to validate the effectiveness of hybrid methods for identifying influential spreaders, we apply the top-performing hybrid method $\mathcal{H} = \{AD, PR, LR\}$ to large real-world networks. We report the results in Table 3.4. The networks considered for this analysis are too large for the application of greedy optimization, so we use the top-performing individual method AD as the baseline for comparing the performance of the hybrid method by taking the ratio $\langle g_H \rangle / \langle g_{AD} \rangle$. When applying the hybrid method to large networks, we use the linear coefficients learned from small/medium networks reported in Table 3.3. In general, we observe that hybrid method improves on the performance of AD. In the subcritical regime, the improvements are negligible. Instead, in both the critical and supercritical regimes, there is a significant increase in overall performance, although the variance in the supercritical regime is also high. On average, the overall performance increases by 2-5%, in line with the results observed on the corpus of small/medium networks. This provides additional support to the robustness and generality of our finding that linear combinations of methods can help in finding more influential seed sets. It is necessary to note that hybrid methods use larger amount of information compared to individual methods, such as AD. The amount of information used might be at the root of increase in performance. On the other hand, the improvement in effectiveness does not cause drawbacks in efficiency. Linear coefficients are already established for three dynamical regimes. The computational complexity of hy-

brid methods are equal to the maximum of the computational complexities of the individual methods that they are composed of, making them applicable to large networks.

3.4 Conclusion

In this chapter, we comparatively analyzed the performances of heuristics for identifying influential spreaders in networks. We used the independent cascade model for simulating spreading dynamics, and studied 16 methods for the identification of influential spreaders that are widely adopted to approximate solutions to the influence maximization problem. The analysis was performed on a large corpus of 100 real-world networks from different domains such as social networks, transportation networks, and biological networks. We used the greedy algorithm as a baseline to analyze the performances of the 16 methods. Furthermore, we used random selection as a method for identifying influential spreaders in networks to have a lower bound on the performance. Overall, the results indicate that the performance of some simple heuristic methods are not too far from those obtained by the greedy algorithm, which is a computationally expensive method. Adaptive degree centrality turned out to be a powerful network centrality metric for the identification of influential spreaders. Adaptive degree centrality is a simple enough method to be applied on large real-world networks. The performance of adaptive degree centrality was on average 96% of the performance of the upper baseline, *i.e.*, greedy algorithm, when 5% of nodes are selected as seeds. There are several other methods that have comparable performances to adaptive degree, even though adaptive degree comes out at the top when we consider both performance and computational efficiency. When we consider the overlap between the seeds identified by the greedy algorithm and other methods, we observe that as this overlap increases, the performance of the method also increases. In general, the overall precision values of the various methods, which quantifies the overlap of seeds from a method m and greedy algorithm, turns out to be relatively low. This is expected given the NP-complete nature of the influence maximization problem.

We further test hybrid methods, which linearly combine several individual methods. The analysis shows that it is possible to achieve better performances by using hybrid methods, especially when one combines individual methods that use information on different levels of network. We found that it was possible to increase the performance up to 98% of the upper baseline greedy by combining different methods. We have also tested this on larger networks, and the results suggest the same outcome as found in the corpus of 100 small/medium-sized real-world networks. Combinations of a few individual methods can increase the performance without significantly increasing the computational cost.

Method	Features	Subcritical	Critical	Supercritical
AD	c_{AD}	1.000	1.000	1.000
	$\langle g_m \rangle$	0.993	0.961	0.931
	$\langle p_m \rangle$	0.755	0.548	0.119
CD	c_{CD}	1.000	1.000	1.000
	$\langle g_m \rangle$	0.983	0.963	0.929
	$\langle p_m \rangle$	0.730	0.525	0.100
B	c_B	1.000	1.000	1.000
	$\langle g_m \rangle$	0.946	0.954	0.938
	$\langle p_m \rangle$	0.590	0.483	0.110
AD,B	c_{AD}	0.718	0.590	0.023
	c_B	-0.027	0.046	0.069
	$\langle g_m \rangle$	0.987	0.964	0.936
	$\langle p_m \rangle$	0.755	0.551	0.116
AD,PR,LR	c_{AD}	1.189	1.044	0.115
	c_{PR}	-0.266	0.145	0.772
	c_{LR}	-0.336	-0.632	-0.771
	$\langle g_m \rangle$	0.991	0.980	0.971
	$\langle p_m \rangle$	0.806	0.616	0.300
PR,LR,CD	c_{PR}	0.006	0.386	0.803
	c_{LR}	-0.419	-0.702	-0.771
	c_{CD}	1.028	0.898	0.088
	$\langle g_m \rangle$	0.985	0.979	0.971
	$\langle p_m \rangle$	0.784	0.597	0.293
AD,B,LR	c_{AD}	1.096	1.047	0.343
	c_B	-0.010	0.067	0.083
	c_{LR}	-0.466	-0.565	-0.395
	$\langle g_m \rangle$	0.993	0.976	0.952
	$\langle p_m \rangle$	0.810	0.625	0.220
PR,LR,EI	c_{PR}	0.304	0.583	0.740
	c_{LR}	0.101	-0.251	-0.733
	c_{EI}	0.235	0.277	0.121
	$\langle g_m \rangle$	0.973	0.964	0.970
	$\langle p_m \rangle$	0.698	0.589	0.304

Table 3.3: **Hybrid methods for the identification of influential spreaders in networks.** The table is organized in blocks, each corresponding to a specific method. For every method m, either individual or hybrid, we report performance values for the three different dynamical regimes in terms of overall performance $\langle g_m \rangle$ and overall precision $\langle p_m \rangle$. The top three blocks correspond to the best individual methods in the three regimes according to overall performance metric. The remaining blocks are for hybrid methods. In each block, the first rows report values of the coefficient c_m of the individual method m in the definition of the hybrid method. We report the averages for the coefficient values over $1,000$ iterations of the learning algorithm. The bottom two rows in each block correspond instead to the values of the performance metrics.

	$\langle g_{\mathcal{H}} \rangle / \langle g_{AD} \rangle$		
Network	Subcrit.	Critical	Supercrit.
Slashdot	1.003	1.017	1.062
Gnutella, Aug. 31, 2002	1.009	1.040	1.039
Epinions	1.012	1.057	1.130
Flickr	1.007	1.082	1.242
Gowalla	1.011	1.024	1.066
EU email	1.002	1.009	0.923
Web Stanford	1.009	1.031	1.035
Amazon, Mar. 2, 2003	1.008	1.025	0.994
YouTube friend. net.	1.004	1.013	0.952
Average on large networks	1.007 ± 0.001	1.033 ± 0.007	1.050 ± 0.030
Average on the corpus of 100 networks	1.001 ± 0.002	1.021 ± 0.003	1.043 ± 0.005

Table 3.4: **Identification of influential spreaders in large networks.** We compare the performance of the hybrid method $\mathcal{H} = \{\text{AD,PR,LR}\}$ with the individual method AD. For the hybrid method, we use the values of the coefficients reported in Table 3.3. From left to right, we report the name of the network, value of the ratio $\langle g_{\mathcal{H}} \rangle / \langle g_{AD} \rangle$ between the performance metric of the hybrid method $\mathcal{H} = \{\text{AD,PR,LR}\}$ and the one of the individual method AD for the subcritical, critical, and supercritical regimes. The bottom two lines in the table report, for each dynamical regime, average values and standard errors of the mean for the ratios $\langle g_{\mathcal{H}} \rangle / \langle g_{AD} \rangle$ over the set of large networks and over the corpus of 100 networks considered.

4 Influence maximization in noisy networks

4.1 Introduction

Most studies on influence maximization rely on one strong assumption: prior knowledge of system structure and dynamics is complete and free of errors. In real-life applications of influence maximization, we should recognize that this assumption is at best optimistic. The presence or absence of a connection in a network is generally established from the results of some experimental observation, and therefore potentially affected by experimental errors (110). Similarly, the type of process that is driving the spreading might be known, but the exact value of the rates at which the spreading occurs might be unknown. There are techniques to approximate the spreading rates from empirical observations of spreading events (111). However, these techniques rely on the assumption that the structural information is complete and free of errors. Furthermore, the influence maximization problem is about controlling the fate of a future or an ongoing spreading process, so posterior estimates of spreading rates are not very helpful.

Several studies have considered the reliability of network centrality measures in the presence of noisy or incomplete structural information (12, 13, 112). In the context of the influence maximization problem, there has been previous studies testing the robustness of centrality metrics in noisy structural data (56). However, there are no previous studies that attempted to understand how incomplete or erroneous information on both the structure of the network and the spreading dynamics affects our ability to solve the influence maximization problem. Note that one may naively expect that noise does not dramatically modify the overall trend of a geometric centrality metric as shown in (56). However, the distortions that noise can create in the solutions of a combinatorial optimization problem such as influence maximization are far less predictable. Here, we aim to fill this gap of knowledge.

4.2 Methods

4.2.1 Networks

We conduct our analysis on the affects of noise in influence maximization problem on static networks. We use 14 undirected and unweighted real-world networks in our analysis. The networks used are shown in Table 4.1.

Network	N	Ref.
URV email	1133	(1)
US Air Transportation	500	(113)
Tennis	4342	(114)
C. Elegans, neural	297	(2)
High school, 2012	180	(28)
Air traffic	1226	(101)
Open flights	2939	(101, 115)
UC Irvine	1899	(101, 116)
Petster, hamster	1858	(101)
Political blogs	1224	(117)
Political books	105	(117)
US Power grid	4941	(2)
S 838	512	(118)
Yeast, protein	2284	(119)

Table 4.1: **Real-world networks.** From left to right, we report the name of the network, the size of the network, and the references(s) where the network was first analyzed.

4.2.2 Spreading dynamics

We use the ICM to simulate the spreading process as explained in Section 2.2.3. Due to the stochastic nature of the spreading model, we measure the results for every given initial condition by averaging over 20 independent numerical simulations.

4.2.3 Influence maximization

We use greedy optimization to select influential spreaders and find the seed set $\mathcal{X}$ for a given size $|\mathcal{X}|$ for the seed set. We use the algorithm by Chen *et al.* (18) as described in Section

2.4.2 for greedy optimization.

4.2.4 Modeling structural and dynamical errors

The process of selecting the top spreaders relies on prior knowledge of the network structure and spreading dynamics. This means that the seed set $\mathcal{X}$ depends on the information at our disposal about the structure of the network, *i.e.*, the adjacency matrix A. The seed set $\mathcal{X}$ also depends on our prior knowledge about the dynamical process, *i.e.*, the ICM with spreading probability λ. It is common practice to assume the prior knowledge on structure and dynamics as complete and free of errors. This is equivalent to assuming that the inputs A and λ of the algorithm for identifying the influential spreaders are equal to their true values, namely A_{true} and λ_{true}, respectively. However, in practical situations this may not be the case. Prior knowledge may be affected by errors, so the actual information used to solve the influence maximization problem is given by A_{err} and λ_{err}, respectively. There are many different ways to model errors on structural and dynamical information. In this work, we use simple, yet realistic, models.

Errors on our knowledge of spreading dynamics are implemented by simply setting $\lambda_{true} \neq \lambda_{err}$. We assume to know that spreading occurs following the rules of ICM, but the exact value of the spreading probability is not known. Instead of using raw values for the spreading probabilities, we rescale them as $\phi_{true} = \lambda_{true}/\lambda_c$ and $\phi_{err} = \lambda_{err}/\lambda_c$, where λ_c is the critical threshold for the true network. This transformation is used to simplify the presentation of results. It also allows us to use the same reference values for all networks to distinguish between dynamical regimes of spreading. Depending on whether ϕ_{true} is smaller than, equal to, or larger than 1, we say the network is respectively in subcritical, critical, or supercritical regime. Similarly, the value of ϕ_{err} tells which dynamical regime we hypothesize we are in. Previous studies have shown that the identities of seeds are affected by the regime of spreading (93). We expect therefore that the error in the estimation of the spreading probability may strongly affect our ability to properly predict the top spreaders.

Errors on our knowledge of the network structure are implemented using a model similar to the one in (110). The total number of nodes N is unaffected. Errors happen on edges. This means that the total number of edges M_{err} in the altered network can be different from the true number of edges M_{true}. We consider two potential sources of errors. The first source is responsible for making true edges disappear. Given the true adjacency matrix A_{true}, each edge is removed with probability $0 \leq \epsilon_{del} \leq 1$ in A_{err}. The number of edges deleted is equal to zero for $\epsilon_{del} = 0$, and equals M_{true} for $\epsilon_{del} = 1$. The second source of error generates false edges. Every pair of nodes that is not connected according to A_{true} appears as connected in A_{err} with probability $\epsilon_{add} M_{true}/[N(N-1)/2 - M_{true}]$, where $0 \leq \epsilon_{add} \leq 1$. When $\epsilon_{add} = 0$, no false edges are added. For $\epsilon_{add} = 1$, the expected number of false edges equals to M_{true}. We define both parameters ϵ_{del} and ϵ_{add} in the range $[0, 1]$, and their maximum values correspond to an expected alteration, *i.e.*, deletion or addition, of 100% of the true number of edges.

Please note that removing true edges is not the inverse operation of adding false edges. For instance, even the addition of a very small number of edges among pairs of non-connected nodes is able to decrease substantially the average path length of networks that do not originally satisfy the small-world property (2). For example, in networks with strong spatial embedding, random edges are likely to behave as shortcuts between spatially far regions of the system. On the other hand, removing a handful of true edges does not change the average path length dramatically. Also, we do not expect the two sources of structural errors to be equally likely in real-world networks. Their likelihood depends on the type of network, and the way the network is constructed from empirical observations (110). As the two sources of structural error can not be treated on the same footing, we always consider them separately in our analysis, *i.e.*, the condition $\epsilon_{del}\epsilon_{add} = 0$ is always satisfied.

Our choice for modeling structural noise is heavily inspired by (110). This model is simple enough, yet can naturally describe the sources of uncertainty in empirically constructed social networks. Alternative models for implementing structural noise can be considered. For example, a model shuffling the true edges with certain probability would provide a way

to introduce structural noise without altering the degree of nodes in the network. In general, the choice of the noise model should depend on the specific question one wants to address, or the specific system considered. The questions we investigate here can be fully addressed with our particular choice for the noise model.

4.2.5 Measuring performance

Given the inputs A_{err} and ϕ_{err}, we use greedy optimization to identify the set $\mathcal{X}_{err}$ of top spreaders. As the greedy optimization is stochastic, we apply the algorithm $R = 10$ times to find R potentially different sets of $\mathcal{X}_{err}$. For each set, we use numerical simulations of the ICM relying on A_{true} and ϕ_{true} to evaluate their performance. We define $\mathcal{X}_{err} = \{x_1, x_2, \ldots, x_{|\mathcal{X}_{err}|}\}$, and $\mathcal{X}_{err}^{(r)} = \cup_{i=1}^{r} x_i$. Note that by definition $\mathcal{X}_{err}^{(|\mathcal{X}_{err}|)} = \mathcal{X}_{err}$. The overall performance of the set $\mathcal{X}_{err}$ is computed according to the equation

$$q(\mathcal{X}_{err}) = \frac{1}{N|\mathcal{X}_{err}|} \sum_{1}^{|\mathcal{X}_{err}|} O(\mathcal{X}_{err}^{(r)}) \tag{4.1}$$

where $O(\mathcal{X}_{err}^{(r)})$ is the average size of the outbreak for the seed set $\mathcal{X}_{err}^{(r)}$. For given A_{err}, the overall performance is finally obtained by taking average over all R realizations of the sets of top spreaders. To account for the stochastic nature of structural noise, we repeat the entire procedure 10 times, and quantify the spreading performance of top spreaders as the average value over these independent realizations. Note that the sum on the r.h.s. of Equation 4.1 allows us to estimate not only the overall performances of $\mathcal{X}_{err}^{(r)}$, but also the way the set is constructed. The pre-factor on the r.h.s. of Equation 4.1 is used only to confine values of performance metric to the interval $[0, 1]$.

4.3 Results

In Figure 4.1, we display the results obtained for the e-mail communication network originally considered in (1). In the majority of cases, the performance of the seed sets appears robust against structural noise. However, the overall ability to properly select seeds may be seriously

affected by noise both at the structural and dynamical levels. Major issues seem to arise when $\phi_{true} > 1$ and $\phi_{err} \leq 1$ (see Figure 4.1d).

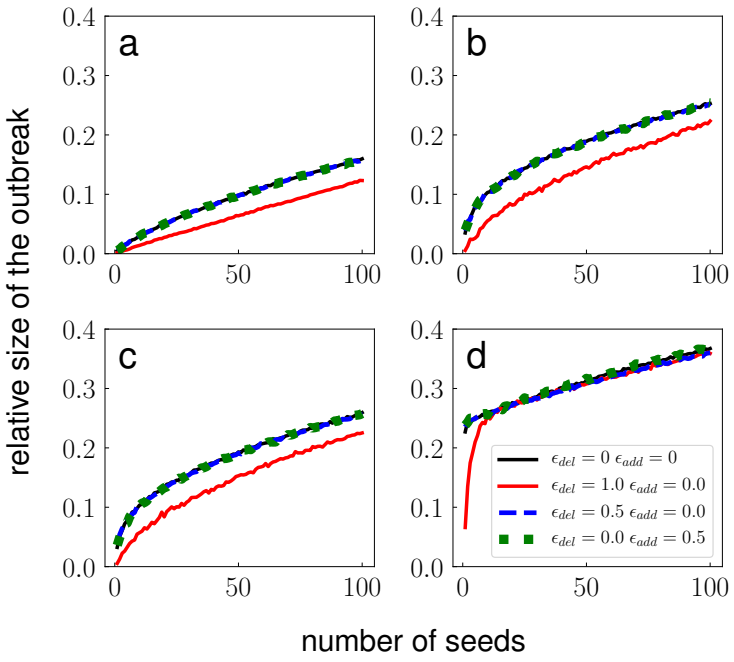

Figure 4.1: **Influence maximization in presence of structural and dynamical noise.** The true network structure analyzed here is given by the email communication network (1). (a) Relative size of the outbreak O/N as a function of the number of seeds found by greedy optimization. The true spreading probability of the ICM is such that $\phi_{true} = 0.5$. Prior dynamical knowledge used by the greedy algorithm is not affected by noise, i.e., $\phi_{err} = \phi_{true} = 0.5$. The different curves correspond to different level of noise that affect prior structural information. We consider various combinations of the parameters ϵ_{del} and ϵ_{add}. (b) Same as in panel a, but for $\phi_{true} = 1.0$ and $\phi_{err} = 0.5$. (c) Same as in panel a, but for $\phi_{true} = \phi_{err} = 1.0$. (b) Same as in panel a, but for $\phi_{true} = 1.5$ and $\phi_{err} = 1.0$.

In order to systematically analyze the observed trend, we set $|\mathcal{X}_{err}| = 100$, and quantify the performance of selected seed sets using Equation 4.1 for different levels of noise. The results of this analysis are shown in Figure 4.2. The various panels refer to different dynamical regimes identified by different values of ϕ_{true}. In every panel, we present three curves, each representing different values of ϕ_{err}. Each curve shows the spreading performance $q(\mathcal{X}_{err})$ as a function of structural noise parameters ϵ_{del} and ϵ_{add}. Note that even though the two sources of structural noise are never considered together, we present them in the same plot for the sake of compactness. In general, we observe that maximum performance is achieved

at $\epsilon_{del} = \epsilon_{add} = 0$ only for $\phi_{true} = \phi_{err}$. The two sources of structural noise affect the choice of seeds differently. In the first case for $\epsilon_{del} = 0$ and $0 \leq \epsilon_{add} \leq 1$, we roughly see that $q(\mathcal{X}_{err})$ is a monotonic function of ϵ_{add}, decreasing if $\phi_{err} \geq \phi_{true}$, and increasing otherwise. When we have $\epsilon_{add} = 0$ and $0 \leq \epsilon_{del} \leq 1$ instead, $q(\mathcal{X}_{err})$ is not a monotonic function of structural error. The trend changes depending on whether the system is in subcritical or supercritical regime. If $\phi_{true} \leq 1$, $q(\mathcal{X}_{err})$ is concave, otherwise it is convex. In the second regime, it becomes possible to obtain higher performance by adding further structural noise. If $\phi_{err} < \phi_{true}$, the best performance is achieved at $\epsilon_{del} = 1$, essentially using no structural information and sampling the seeds randomly.

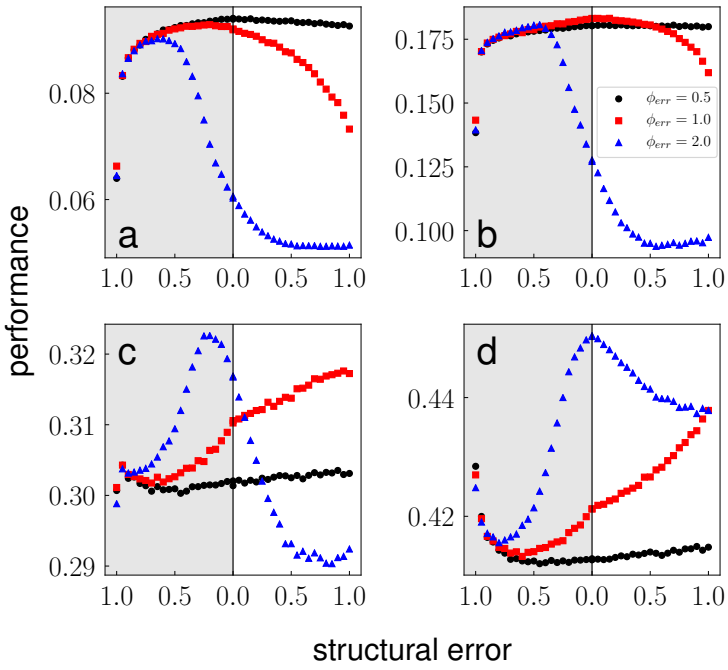

Figure 4.2: **Performance of the top spreaders in the presence of structural and dynamical noise.** We consider the same network as in Fig. 4.1. (a) We compute Eq. 4.1 for the set of top spreaders of size $|\mathcal{X}_{err}| = 100$, and we plot the value of the performance as a function of the noise level in prior structural information. Performance is measured for $\phi_{true} = 0.5$. The shaded part of the plot serves to report results valid for $0 \leq \epsilon_{del} \leq 1$ and $\epsilon_{add} = 0$. The non-shaded part of the graph instead represents results for $0 \leq \epsilon_{add} \leq 1$ and $\epsilon_{del} = 0$. (b) Same as in panel a, but for $\phi_{true} = 1.0$. (c) Same as in panel a, but for $\phi_{true} = 1.5$. (d) Same as in panel a, but for $\phi_{true} = 2.0$.

We next analyze the phenomenon systematically. For different combinations of ϕ_{true} and

ϕ_{err}, we computed ϵ^*_{del} and ϵ^*_{add}, *i.e.*, the values of the structural noise parameters where $q(\mathcal{X}_{err})$ reaches its maximum. The results of this analysis are shown in Figure 4.3. As shown in Figure 4.3a, when $\phi_{err} \simeq \phi_{true}$, $\epsilon^*_{del} \simeq 0$. However, when the error on dynamical parameter increases, this error is compensated by making further mistakes in the structure. When structural noise is allowed through the addition of false edges, noise is only helpful at the supercritical regime (see Figure 4.3b).

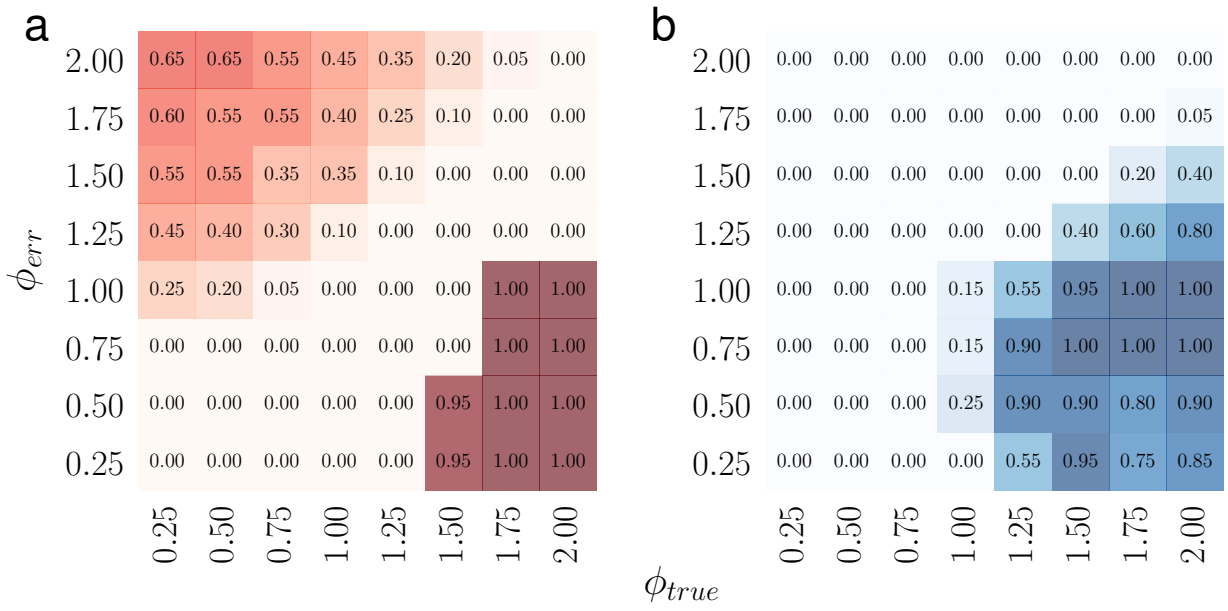

Figure 4.3: **Best values of the structural errors in the presence of dynamical uncertainty.** We consider the same network as in Fig. 4.1. (a) We set $\epsilon_{add} = 0$, and, for given dynamical parameters ϕ_{true} and ϕ_{err}, we determine ϵ^*_{del}, *i.e.*, the value of the error parameter ϵ_{del} that leads to the maximum performance in the prediction of top spreaders. Best estimates of ϵ^*_{del} are reported in the cells of the table. The intensity of the background color is proportional to the value of ϵ^*_{del}. (b) Same as in panel a, but for the other source of structural error. Here, we set $\epsilon_{del} = 0$ and focus on ϵ^*_{add}, *i.e.*, the value of the error parameter ϵ_{add} that allows to identify the best performing set of top spreaders.

So far, the reported results are only for a specific network. However, the main findings are not sensitive to the choice of network, in the sense that the qualitative results are very similar for all real-world networks we analyzed (see Appendix B). The only major difference arises when we consider networks characterized by strong spatial embedding (*i.e.*, a specific modular structure identified by very loose intermodule connections (120)). In such networks,

the strong asymmetry of altering the network structure by adding or removing edges is apparent. This is visible in Figure 4.4, where we report the analogue of Figure 4.2 for the power grid network (2). In the figure, we see that $q(\mathcal{X}_{err})$ does not behave smoothly around $\epsilon_{del} = \epsilon_{add} = 0$. However, the general findings valid for the two different sources of structural noise are almost identical to those valid for networks with no spatial embedding.

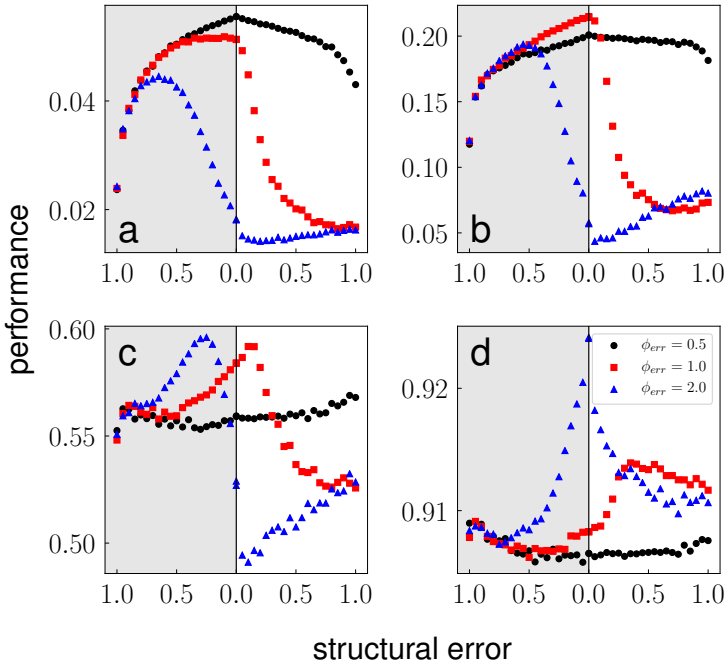

Figure 4.4: **Performance of the top spreaders on a spatially embedded network in presence of structural and dynamical noise.** Same analysis as in Fig. 4.2, but for a different network. Here, the true network structure is given by the US power grid network (2).

Same qualitative results also hold for different values of $|\mathcal{X}_{err}|$, as long as the value is large enough compared to the size of the network N. In Appendix B, we reports the results for $|\mathcal{X}_{err}| = 10$. For the e-mail network considered here in the main text, which has a size of $N = 1133$, the clear pattern of Figure 4.2 becomes much noisier. For networks with smaller size, we observe that the pattern is already clear even for $|\mathcal{X}_{err}| = 10$.

One may explain the results with the following naive argument. For simplicity, let us consider only the case where noise randomly deletes true edges. In our prior knowledge, every true edge becomes invisible with probability ϵ_{del}. We also believe that the ICM has

spreading probability ϕ_{err}. In summary, our prior knowledge forces us to think that the effective spreading probability on a random edge that we are considering in the true but unknown network is $(1-\epsilon_{del})\phi_{err}$, rather than the true value ϕ_{true}. Best predictions should be obtained for $(1-\epsilon_{del})\phi_{err} \simeq \phi_{true}$. If $\phi_{err} > \phi_{true}$, one can correct the mistake by choosing the appropriate $\epsilon_{del} \in [0,1]$. If $\phi_{err} < \phi_{true}$, there is no way to satisfy the previous equation. The best performance can be naively expected to be at $\epsilon_{del} = 0$, as this value corresponds to the noise level that minimizes the difference between effective and true spreading probability. However, this is not what we observe in our numerical results, where the best performance is actually achieved at $\epsilon_{del} = 1$. This apparent paradox can be solved by accounting for structural correlations. As well known, the true top spreaders in ICM depend on the critical regime (93). In the subcritical regime, central nodes are generally better locations for seeds. In the supercritical regime, peripheral nodes are selected first instead. In both these regimes, seeds are generally placed on nodes that are not directly connected, as a source of spreading that is too redundant is generally not optimal. If the probability ϵ_{del} of random deletion of edges is not very high, then the ranking based on degree centrality of the nodes is basically unaffected. However, pairs of truly connected nodes appear as disconnected in the noisy version of the network regardless of their degree. As a results, for $\phi_{err} < 1$, many high-degree nodes are chosen as seeds. However, they may behave poorly as seed sets, as they constitute a source of spreading that is too redundant to be optimal. A visual intuition of this structural explanation is provided in Figure 4.5. In the figure, we consider the true value of the average degree of the set of top spreaders identified using noisy information. Each panel corresponds to a different value of ϕ_{err}. Note that the value of average degree measured at $\epsilon_{del} = \epsilon_{add} = 0$ is the one that corresponds to the true set of optimal seeds for the dynamical regime $\phi_{true} = \phi_{err}$. As expected, for subcritical regime, the identified set of top spreaders has high average degree and the structural noise does not affect much the value of this variable, except when $\epsilon_{del} \simeq 1$. Instead in the supercritical regimes, the best performances is achieved for sets with low values of the average degree, comparable with

the average degree of the network. Structural noise changes dramatically the set of seeds, especially in the region $0 < \epsilon_{del} < 1$. However, in the regime of very large noise, the average degree of the seed set is basically equal to the average degree of the network as nodes are chosen using almost no structural information.

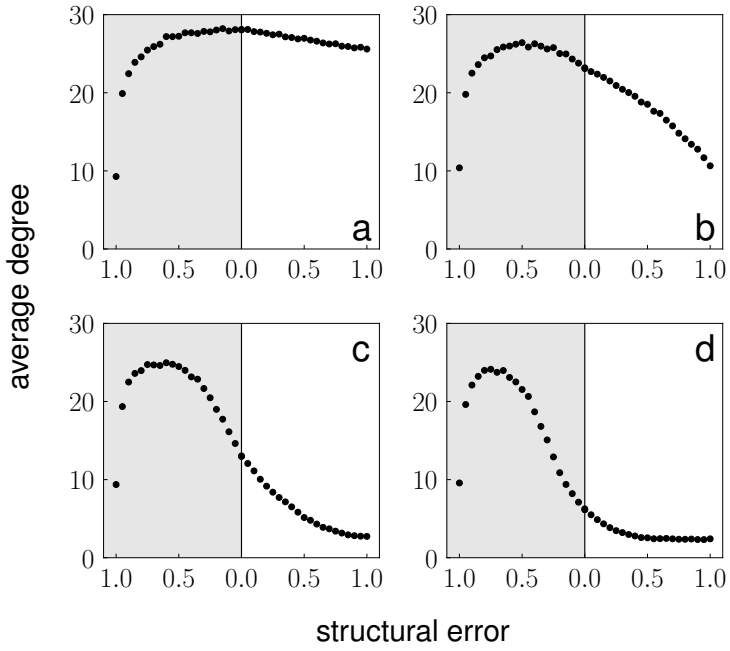

Figure 4.5: **Average degree of the set of top spreaders.** We consider the same network as in Fig. 4.1. (a) Average degree of the set of $|\mathcal{X}_{err}| = 100$ of top spreaders identified for $\phi_{err} = 0.5$ and different values of the structural errors ϵ_{del} and ϵ_{add}. As in Figure 4.2, we use left part of the plot, highlighted with a gray-shaded background, to report results valid for $0 \leq \epsilon_{del} \leq 1$ and $\epsilon_{add} = 0$. The non-shaded part of the graph instead represents results for $0 \leq \epsilon_{add} \leq 1$ and $\epsilon_{del} = 0$. (b) Same as in panel a, but for $\phi_{err} = 1.0$. (c) Same as in panel a, but for $\phi_{err} = 1.5$. (d) Same as in panel a, but for $\phi_{err} = 2.0$.

4.4 Conclusion

In this chapter, we considered a simple yet practically relevant scenario. We assumed that prior information used in identifying influential spreaders for the influence maximization problem is affected by some noise, and we studied how the quality of the solutions found using noisy information deteriorates as a function of the noise intensity. The main finding is that the quality of the solution always decreases monotonically with noise, if structural

and dynamical noise are considered independently. However, when both sources of noise act simultaneously, one of them can compensate the disruptive effect of the other. Noise affecting the dynamical information may be suppressed by additional noise at the structural level, or vice versa. This is particularly apparent when structural noise is such that random edges of the original network disappear with a certain probability. As this is a plausible model of error that may affect our knowledge of the true network structure (110), our results may be important for real-world applications. More in general, the approach presented here may be used to understand how incomplete and/or erroneous information on network structure and spreading dynamics affects our ability to solve optimization problems in a meaningful way.

5 Influence maximization on temporal networks

5.1 Introduction

The problem of influence maximization has been traditionally studied on static networks. However, many real-world networks display nontrivial edge temporal variability (23). If structural variations happen on a timescale comparable with the one of the spreading dynamics, then the two processes interact in a highly nontrivial manner (121–124). Most of the work in the area of spreading processes on time-varying networks has been focusing on the characterization of their critical properties. Some attention has been devoted to the problem of influence maximization. Specifically, Osawa *et al.* studied the influence maximization problem on temporal networks for the susceptible-infected (SI) model (125). They proposed an alternative to the greedy algorithm, showing that the proposed method is effective in correctly identifying top spreaders in networks with community structure. Michalski *et al.* model the temporal network as a multilayer network and the spreading dynamics using the linear threshold (LT) model (126). They analyze the solutions to the influence maximization problem under different granularities for the temporal network, including the time-aggregated versions of the networks. Murata *et al.* propose heuristic methods to solve the influence maximization problem on temporal networks (127). Han *et al.* (128) and Zhuang *et al.* (129) propose a method where it is assumed that only the topology of the first layers of a temporal network is known. The topology of the succeeding layers can only be discovered by partial probing of nodes in the network, and the partial topological information gathered is used to select influential spreaders. Gayraud *et al.* focus on the independent cascade (IC) and LT models in a theoretical study about the properties of the influence maximization problem (130). At odds with many of the influence maximization problems considered in the literature, they show that their optimization problem does not necessarily involve the maximization of a submodular function. They further demonstrate that delaying the activation of some initial spreaders may increase their effective influence.

In this chapter, we introduce a discrete-time version of the susceptible-infected-recovered (SIR) model on temporal networks (32). We systematically study the influence maximization problem associated with SIR spreading on 12 real-world temporal networks. We test the performance of different approximate algorithms aimed at the identification of the best spreaders in the network. Approximations rely on different levels of dynamical and topological information.

We note that our modeling framework is very similar to the one previously used by Valdano *et al.* for susceptible-infected-susceptible (SIS) model on temporal networks (124). The properties of the two spreading models are, however, rather different, especially because the SIR model displays a sensitivity to the temporal ordering of the network edges that is stronger than the one observed for the SIS model. Furthermore, both the SI model and ICM previously studied (130) can be seen as two extreme cases of our model. We extend and generalize those analyses in several respects. First, being able to tune our model between the two extremes, we are essentially able to change the effective level of submodularity of the function that we want to optimize. Second, we do not focus on specific values of the spreading probability for every network. Rather, we tune each network close to its own critical point and study the influence maximization problem near criticality. While studying the phase diagram of the SIR model, we show that the system behavior can be reasonably well predicted by a mean-field approximation and that such an approximation can be effectively used to solve the influence maximization problem. Finally, we do not assume that the optimization problem is solved using full information about the system. Rather, we systematically test the performances of approximations obtained under partial knowledge of the network topology and dynamics. Our results, obtained on a corpus of real-word temporal networks that is significantly larger than those typically considered in previous studies, provide clear indications on the type of ingredients that one needs to rely on when available topological information is incomplete or noisy.

5.2 Methods

5.2.1 Networks

We use 12 empirical datasets containing time-stamped social interactions among pairs of individuals. Datasets refer to two types of interactions. In some of the datasets, interactions correspond to physical proximity contacts, *e.g.*, among school students (131, 132), conference attendees (133), and hospital staff and patients (134). In the other type, interactions stand for emails exchanged between coworkers (135). In both cases, we treat the contacts as undirected. All datasets considered in the study are listed in Table 5.1.

Dataset	W	T	N	Ref.
Email, dept. 1	2,880,000	18	309	(135)
Email, dept. 2	2,880,000	18	162	(135)
Email, dept. 3	2,880,000	18	89	(135)
Email, dept. 4	2,880,000	18	142	(135)
High school, 2011	14,400	11	126	(28)
High school, 2012	14,400	21	180	(28)
High school, 2013	14,400	14	327	(29)
Hospital ward	14,400	20	75	(134)
Hypertext, 2009	14,400	11	113	(133)
Primary school	7,200	11	242	(131, 132)
Workplace	28,800	20	92	(136)
Workplace-2	28,800	20	217	(137)

Table 5.1: **Real-world temporal networks.** From left to right, we report the name of the dataset, the length W of the temporal window used to slice the data (time is expressed in seconds), the number T of network layers resulting after slicing and cleaning data, the number of nodes N in the network, and the reference to the paper(s) where the data were first considered.

We follow the modeling scheme presented in Section 2.1.2 to create the temporal networks. In this scheme, all layers contain exactly N nodes, where N is the number of distinct individuals involved in at least one social interaction in the dataset. By construction, some nodes may have degree equal to zero in one or more temporal layers. In order to avoid the presence of layers that are too sparsely connected, we exclude from our analysis network

layers containing a number of null-degree nodes greater than $0.9N$. In some datasets, a large portion of layers, corresponding to periods of inactivity, is excluded from the analysis (see Appendix C). For example, in the high school datasets, there are no recorded night-time interactions, and the layers corresponding to such time frames are excluded as they do not contain edges. After the cleaning procedure, we end up with T layers chronologically ordered from 1 to T.

We choose different W values of temporal window length depending on the dataset on purpose. Our goal is simply ending up with a similar number T of layers across datasets. Also, the threshold value used for disregarding low-density layers is arbitrarily chosen. We are aware that both ingredients affect the construction of the network layers and the outcome of the spreading process taking place on them. We stress, however, that the goal here is not understanding the dynamics of a specific temporal network and/or a specific choice of the parameters of the spreading model. Rather, we want to study the general problem of influence maximization on temporal networks and compare different strategies to solve the problem. As far as we are concerned, the real-world datasets considered here just provide useful data for the construction of temporal network topologies, and influence maximization strategies are compared one against the other on the same test sets.

At the end of the procedure described above, we have a sequence $\{A^{(1)}, \ldots, A^{(t)}, \ldots, A^{(T)}\}$ of T adjacency matrices at our disposal. The adjacency matrix $A^{(t)}$ fully encodes the information about the topology of the tth temporal network layer, with its generic element $A_{ij}^{(t)} = A_{ji}^{(t)} = 1$ if a connection exists between nodes i and j at stage t of the network dynamics, and $A_{ij}^{(t)} = A_{ji}^{(t)} = 0$ otherwise.

5.2.2 Spreading dynamics

We study the spreading dynamics taking place on temporal networks using the spreading model described in Section 2.2.2. We assume that the characteristic timescales of the spreading process and of the network evolution are identical (124). We consider the discrete-time

version of the SIR model to mimic spreading dynamics (32). In the SIR model, the state $\sigma_i^{(t)}$ for node i at time t can be $\sigma_i^{(t)} = S, I$, or R. Every node i that is infected at time t, *i.e.*, $\sigma_i^{(t)} = I$, attempts to infect all its susceptible neighbors, *i.e.*, all j such that $A_{ij}^{(t)} = 1$ and $\sigma_j^{(t)} = S$. Infection is successfully transmitted with probability λ. In case of a successful attempt, the state of the newly infected node j changes as $\sigma_j^{(t)} = S \to \sigma_j^{(t+1)} = I$, meaning that node j can spread the infection from time $t+1$ on. After all spreading attempts have been performed, each infected node i may recover with probability μ. A successful recovery attempt changes the state of node i as $\sigma_i^{(t)} = I \to \sigma_i^{(t+1)} = R$. A recovered node no longer participates in the dynamics, thus it can not spread nor receive infection. After all recovery attempts have been performed, time increases as $t \to t+1$.

Two standard models of spreading are obtained as special cases of our more general model. If $\mu = 1$, the model reduces to the ICM. If $\mu = 0$ and no nodes are initially in the recovered state, then the SIR model reduces to the discrete-time version of the SI model.

Starting from a given initial configuration $\vec{\sigma}^{(1)} = [\vec{\sigma}_1^{(1)}, \ldots, \vec{\sigma}_N^{(1)}]^{\mathrm{T}}$, we follow the dynamics of the model until the last iteration of spreading is performed, reaching the final configuration $\vec{\sigma}^{(T+1)}$. Even for fixed values of λ and μ, the outcome of the spreading model is highly sensitive to the initial conditions (see Figure 5.1).

We restrict our attention to special types of initial configurations where all nodes are in state S, except for a set $\mathcal{X} = \{x_1, x_2, \ldots, x_{|\mathcal{X}|}\}$ of seed nodes that are in state I, *i.e.*, $\sigma_i^{(1)} = I$ if $i \in \mathcal{X}$ and $\sigma_i^{(1)} = S$ if $i \notin \mathcal{X}$. Given the parameters of the SIR model and the topology of the temporal network, we estimate the relative outbreak size $O(\mathcal{X})$ generated by the seed set $\mathcal{X}$ in a single realization of the SIR model. $O(\mathcal{X})$ is defined as the total number of nodes found either in the states I or R in the stage $T+1$ of the process, divided by the network size N, *i.e.*,

$$O(\mathcal{X}) = \frac{1}{N} \sum_{i=1}^{N} \left[\mathbb{1}_{\sigma_i^{(T+1)}, I} + \mathbb{1}_{\sigma_i^{(T+1)}, R} \right], \tag{5.1}$$

where $\mathbb{1}_{x,y}$ is the identity operator, *i.e.*, $\mathbb{1}_{x,y} = 1$ if $x = y$ and $\mathbb{1}_{x,y} = 0$ otherwise. $O(\mathcal{X})$

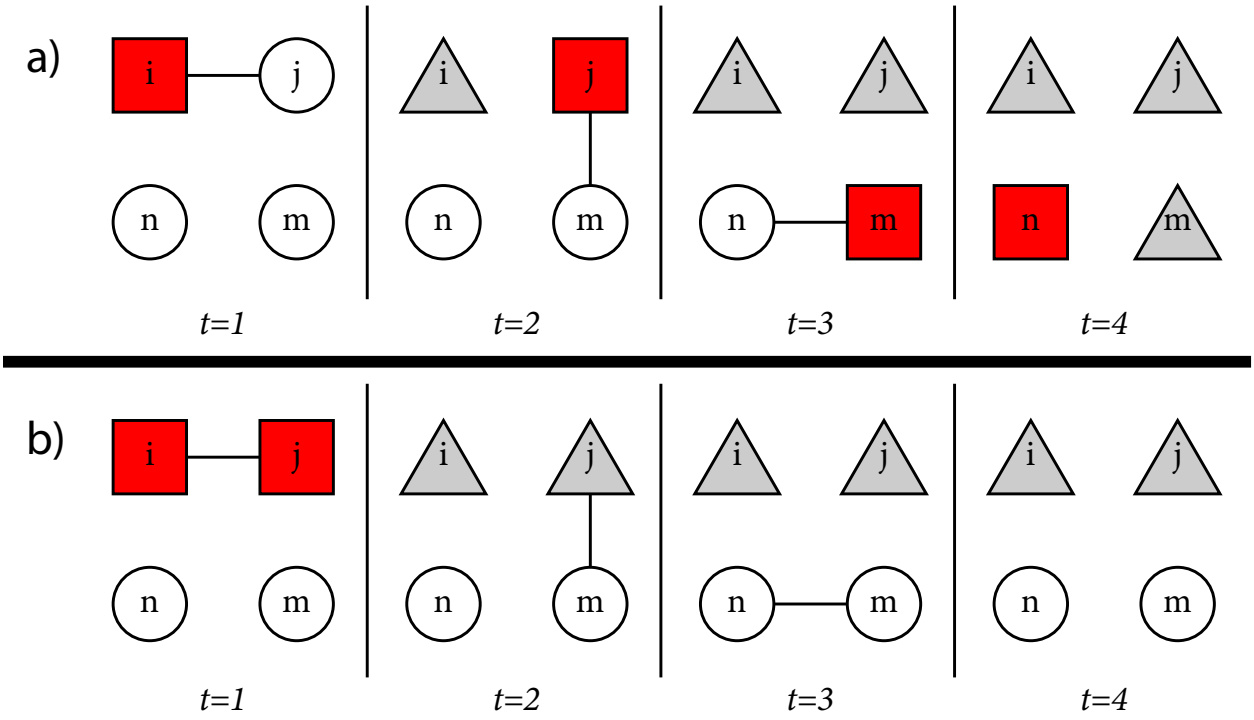

Figure 5.1: **SIR model on temporal networks.** Illustrative example of the modeling framework proposed, where SIR spreading occurs on a temporal network. In the example, the network consists of four nodes and four temporal layers, and the spreading dynamics takes place over four discrete temporal stages. For simplicity, in the illustration we set the SIR model parameters $\lambda = \mu = 1$ so that the dynamics is deterministic. (a) The initial condition is such that only node i is infected, while all others are in the susceptible state. At the end of the dynamics, all nodes are either infected or recovered. (b) Nodes i and j are initially infected, and they are recovered in the final configuration. Nodes n and m remain in the susceptible state.

is a random variable, obeying some probability distribution. We stress that $O(\mathcal{X})$ strongly depends on the choice of the parameters λ and μ, the topology of the network layers and their time ordering. However, we do not report the explicit dependence on the factors for shortness of notation.

As in many of the papers on influence maximization (9), we use the average value of the outbreak size as the metric of influence for the seed set $\mathcal{X}$. Specifically, we numerically estimate the influence of the seed set $\mathcal{X}$ over a finite number K of numerical simulations as

$$\langle O(\mathcal{X}) \rangle = \frac{1}{K} \sum_{k=1}^{K} O_k(\mathcal{X}), \tag{5.2}$$

where $O_k(\mathcal{X})$ is the relative outbreak size of Equation 5.1 obtained in kth instance of the model. We use $K = 2,000$ in all our numerical results, unless otherwise specified.

In Figure 5.2, we show typical phase diagrams obtained for seed sets of size one. In the diagrams, a data point of the outbreak size for a given pair of parameter values μ and λ is obtained as follows. We consider N different initial conditions, each corresponding to one of the nodes selected as the initial spreader with all other nodes initially in the susceptible state, *i.e.*, $\mathcal{X} = \{i\}$ for all $i = 1, \ldots, N$. For each initial condition, we run $K = 500$ simulations and estimate the influence of the node according to Equation 5.2. We finally take the average value of the influence over all initial conditions as representative quantity for the system outbreak size. The system is characterized by a phase transition from a non-endemic to an endemic phase as the parameters of the spreading model are varied. Specifically, the endemic phase is obtained for sufficiently large values of the spreading probability λ. The critical λ value, namely λ_c, where the transition occurs is a function of the recovery probability μ, *i.e.*, $\lambda_c = \lambda_c(\mu)$. We note that λ_c increases as μ increases. We stress that the system size is finite here, so we are not facing a genuine phase transition. Nonetheless, the change in the value of average outbreak size lets us easily notice the presence of a regime where the outbreak is confined to a small part of the network and a regime where spreading involves a large portion of the system.

We estimate the critical value of the spreading probability $\lambda_c(\mu)$ for a given value of the recovery probability μ as the λ value that maximizes the ratio between the standard deviation and average value of the outbreak size, both computed over $K = 500$ numerical simulations of the spreading process initiated by a single randomly chosen seed (we use the same procedure as described above, but for simplicity, only nodes with at least one connection in the first layer of the network are considered as possible seeds). We note that looking at the peak of the ratio of standard deviation over average value is not the only possible way of defining and identifying the critical point of the transition. One, for example, may look at the position of the peak of the standard deviation only. In general, different definitions may

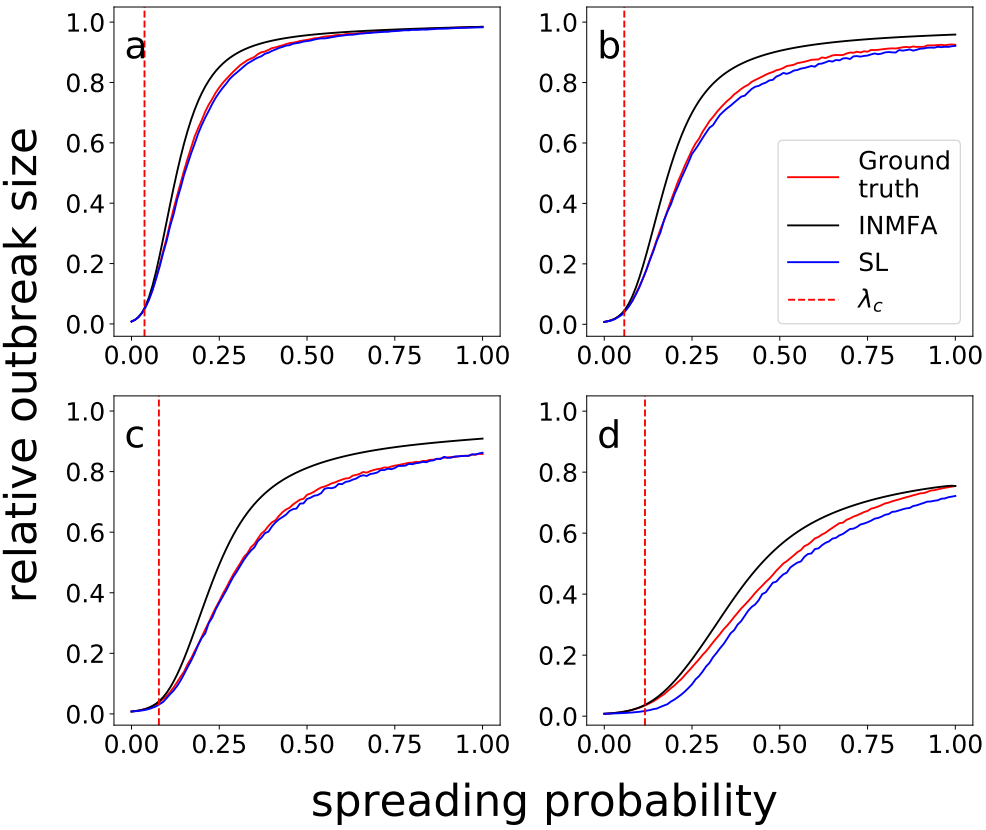

Figure 5.2: **Epidemic transition in real-world temporal networks.** (a) Average value of the relative outbreak size $\langle O(\mathcal{X}) \rangle$ as a function of the spreading probability λ. The seed set corresponds to one randomly chosen node. Results are obtained on the "High school, 2011" network, and by setting $\mu = 0$. Results from numerical simulations on the real network topology (red curve) are compared against those predicted by INFMA (black curve). The dashed red line indicates the position of our best estimate of the critical value of the spreading probability, *i.e.*, λ_c. We further display results of numerical simulations obtained on the same network topology but with the order of the temporal network layers randomized (SL, blue curve). (b) Same as in (a), but for $\mu = 0.25$. (c) Same as in (a), but for $\mu = 0.5$. (d) Same as in (a), but for $\mu = 1$.

lead to slightly different estimates of λ_c. We stress, however, that the λ_c values we obtain seem to identify quite accurately the transition point (see Figure 5.2) and that the exact value of the transition point is not very crucial for the type of analysis we are performing here. In Table 5.2, we report the λ_c values obtained for the various networks considered.

We note that systems display sensitivity to the temporal organization of the underlying networks, and the sensitivity is more apparent for large μ values than for small μ values. Phase diagrams, and resulting values of the spreading probability where transitions occur,

Network	$\lambda_c(\mu = 0)$	$\lambda_c(\mu = 0.25)$	$\lambda_c(\mu = 0.5)$	$\lambda_c(\mu = 1)$
Email, dept. 1	0.016	0.043	0.069	0.130
Email, dept. 2	0.010	0.027	0.049	0.096
Email, dept. 3	0.016	0.038	0.066	0.123
Email, dept. 4	0.010	0.029	0.047	0.099
High school, 2011	0.037	0.057	0.078	0.116
High school, 2012	0.025	0.077	0.136	0.205
High school, 2013	0.023	0.042	0.064	0.119
Hospital ward	0.017	0.048	0.087	0.207
Hypertext, 2009	0.023	0.041	0.060	0.097
Primary school	0.013	0.019	0.029	0.043
Workplace	0.042	0.123	0.241	0.308
Workplace-2	0.023	0.063	0.119	0.248

Table 5.2: **Critical thresholds of real-world temporal networks.** We report our numerical estimates of the critical spreading probability $\lambda_c(\mu)$ for the temporal networks of Table 5.1. Different columns correspond to different values of the recovery probability μ. Errors associated to the estimates are all equal to or smaller than 10^{-3} and they are not reported in the table for sake of compactness.

may dramatically vary by simply randomizing the order of the layers but without changing the actual network topology of the layers (see Figures 5.2 and 5.3 and Appendix C). This is a quite remarkable difference with respect to the SIS modeling framework by Valdano *et al.* where the actual order of the temporal layers is not as important for the outcome of the spreading dynamics (124).

5.2.3 Individual-node mean-field approximation

We can provide a relatively simple description of the spreading process using the individual-node mean-field approximation (INMFA) (32). The approximation consists of describing the stochastic state variable $\sigma_i^{(t)}$ for node i at time t with the three deterministic variables $S_i^{(t)}, I_i^{(t)}$, and $R_i^{(t)}$. Specifically, we have that $S_i^{(t)} = \text{Prob.}[\sigma_i^{(t)} = S]$, *i.e.*, the probability to find node i in state S at stage t over an infinite number of realizations of the process. Similarly, we have that $I_i^{(t)} = \text{Prob.}[\sigma_i^{(t)} = I]$ and $R_i^{(t)} = \text{Prob.}[\sigma_i^{(t)} = R]$. The three deterministic variables are related by the constraint $S_i^{(t)} + I_i^{(t)} + R_i^{(t)} = 1$. Further, the approximation

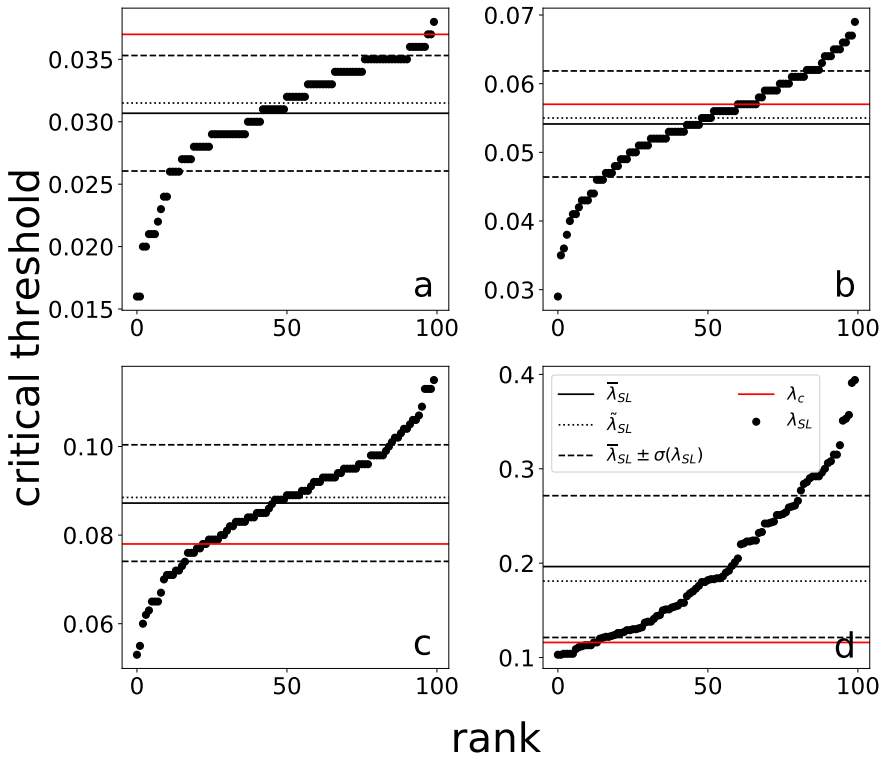

Figure 5.3: **Sensitivity of the spreading outcome to network dynamics.** (a) Best estimates of the critical spreading probability λ_{SL} for randomized versions of the "High school, 2011" temporal network. SIR recovery probability is $\mu = 0$. The randomization consists in reordering the temporal layers only, while the topology of the individual layers is kept invariant. Each black circle corresponds to a specific realization of the randomization process. In the visualization, we simply sort the various realizations depending on their λ_{SL} value. We display horizontal lines identifying the average $\bar{\lambda}_{SL}$ (full black line), the region corresponding to one standard deviation away from the mean ($\bar{\lambda}_{SL} \pm \sigma(\lambda_{SL})$, dashed black lines), the median value $\tilde{\lambda}_{SL}$ (dotted black line), and the actual critical value λ_c measured on the non-randomized version of the network (red full line, Table 5.2). (b) Same as in (a), but for $\mu = 0.25$. (c) Same as in (a), but for $\mu = 0.5$. (d) Same as in (a), but for $\mu = 1$.

consists in neglecting dynamical correlations among two or more state variables, so that joint probabilities can be replaced by products among marginal probabilities. For example, the probability at stage t of the dynamics to find nodes i and j, respectively, in the infected and recovered states is simply written as Prob.$[\sigma_i^{(t)} = I, \sigma_j^{(t)} = R] = I_i^{(t)} R_j^{(t)}$.

Under INMFA, we can describe SIR dynamics on the temporal network with the following set of coupled equations:

$$I_i^{(t)} = (1-\mu)I_i^{(t-1)} + \left(1 - I_i^{(t-1)} - R_i^{(t-1)}\right)\left[1 - \prod_j \left(1 - \lambda A_{ji} I_j^{(t-1)}\right)\right] \tag{5.3}$$

$$R_i^{(t)} = R_i^{(t-1)} + \mu I_i^{(t-1)} \ . \tag{5.4}$$

The initial conditions are suitably chosen depending on the problem at hand. In our case, we set $I_i^{(1)} = 1$ for every $i \in \mathcal{X}$ and $S_i^{(1)} = 1$ for $i \notin \mathcal{X}$, and then use Equations 5.3 and 5.4 to obtain solutions for $t > 1$. Equation 5.3 tells that the probability that node i is in the infected state at time t is the sum of two terms: (i) The probability that the node was already in the infected state at the previous stage of the dynamics but did not recover and (ii) the probability that the node was in susceptible state and received the infection by at least one of its infected neighbors at the previous time step. Equation 5.4 instead tells us that the probability that node i is in the recovered state at time t is the sum of the probability that the node was already recovered or just recovered at the previous stage of the dynamics. INMFA neglects dynamical correlation between nodes. Variables are treated as dynamically independent when instead they are not. In particular, there is a non-zero probability that spreading may occur simultaneously in opposite directions along the same edge, thus causing a systematic overestimation of the true probability of infection. Starting from the imposed initial conditions, one iterates over Equations 5.3 and 5.4 to obtain the marginal probabilities of all nodes in the network at a given stage of the dynamics. The relative size of the outbreak at time t is obtained by simply taking the sum

$$O_{\text{INMFA}}^{(t)} = \frac{1}{N}\sum_{i=1}^{N}\left[I_i^{(t)} + R_i^{(t)}\right]. \tag{5.5}$$

In Figure 5.2, we compare results from the INMFA with ground-truth values obtained from numerical simulations of the spreading process. Due to the independence among variables assumed by INMFA, the approximation overestimates the true outbreak size, and under INMFA the phase transition is expected to happen earlier than in the true dynamical system.

Nonetheless, we note that INMFA provides a relatively good prediction of the true system outcome, especially in the subcritical regime and around criticality.

5.2.4 Influence maximization

As influence maximization is a NP-hard problem (9), its solutions can only be approximated. On static networks, the best available strategy is the greedy algorithm (9), as explained in Section 2.4.1. We have

$$x_k = \arg \max_{v \notin \mathcal{X}_{k-1}} \langle O(\mathcal{X}_{k-1} \cup v) \rangle. \tag{5.6}$$

Essentially, the best seed set $\mathcal{X}$ is built sequentially by adding one node at a time. The node x_k selected at stage k is the one providing the largest marginal increment of influence to the existing seed set. We stress that, at each stage k of the algorithm, one needs to numerically estimate $\langle O(\mathcal{X}_{k-1} \cup v) \rangle$ for all $v \notin \mathcal{X}_{k-1}$ in order to select x_k appropriately, and simulations must be run independently for each potential seed set $\mathcal{X}_{k-1} \cup v$. We remark that the method requires as inputs full topological and dynamical information about the system, including the actual values of the parameters of the spreading model. In the following, we denote solutions of the influence maximization problem obtained via the standard greedy optimization method with the acronym GR.

For the SIR model in static networks, it is known that influence is a growing and submodular function (9), thus greedy solutions are guaranteed to be within a margin $1 - 1/e \simeq 0.63$ from the true optimum (138). On temporal networks, the two above conditions are valid only for the special case $\mu = 0$. However, for $\mu > 0$, influence may decrease as the system size increases, so that the function may also violate the submodularity property (130). As a consequence, the greedy algorithm does not guarantee a known optimality gap.

The complexity of the algorithm described in Equation 5.6 grows cubically with the network size, as estimates of the influence of all seed sets are obtained via numerical simulations. Complexity reduction is possible by approximating $\langle O(\mathcal{X}) \rangle$ in some way, so that the

elementary choice of Equation 5.6 is replaced by

$$x_k = \arg \max_{v \notin \mathcal{X}_{k-1}} F(\mathcal{X}_{k-1} \cup v). \tag{5.7}$$

Here we indicated with $F(\mathcal{X} \cup v)$ a generic function that estimates the incremental importance of node $v \notin \mathcal{X}$ for the influence of the set $\mathcal{X}$, assuming that influence is not directly measured. Typical choices of F leverage parallel and/or partial computation to decrease computational complexity. For the ICM on static networks for example, the equivalence of the spreading model with static bond percolation suggests how to decrease algorithmic complexity without sacrificing performance (18, 139). In (18), $F(\mathcal{X} \cup v)$ is defined as the average size of the clusters that contain node v but no nodes already in $\mathcal{X}$, a quantity that is equivalent to the targeted ground truth $\langle O(\mathcal{X} \cup v) \rangle$ and that can be computed in parallel for all nodes. Variations of the methods in (18, 139) are not easily implementable for the general SIR model, and the temporal nature of the network creates additional challenges. We implement, however, an approximate version of greedy optimization that uses the IN-MFA prediction of Equation 5.5 for the definition of the function F. Many other methods aiming at reducing algorithmic complexity use network centrality metrics for the definition of F such that, during the course of algorithm, the score is static (*e.g.*, degree centrality) or can be quickly recomputed with partial computation (*e.g.*, adaptive degree centrality). On the basis of previous analyses conducted on static networks (24), we focus our attention on adaptive degree centrality only.

In the methods above, we made the strong hypothesis that optimization is performed by knowing in advance that the network is evolving, and how it exactly evolves. Further, the optimization is performed by being aware of the true spreading dynamics, including the actual values of the model parameters and the existence of a specific temporal horizon in the spreading process.

Having full knowledge of all the ingredients of the problem is, however, a strong assumption. In realistic scenarios, it is much more likely to attempt to solve the problem with

limited and/or noisy information. For example, we may have at our disposal only a flat and aggregated version of the true network, where temporal information is absent. In this scenario, we would apply the greedy algorithm to a static network, disregarding completely the existence of network evolution and the time horizon for the spreading. We would further be able to identify the critical regime of spreading for the static network only. In essence, we would still use the same approach as Equation 5.7 where the function F represents an approximation of the ground-truth $\langle O \rangle$ of Equation 5.6. However, the approximation would not be made with the goal of reducing computational complexity. Rather, it would be enforced by the incompleteness of the information at our disposal in the solution of the true problem.

There are many potential ways in which topological information may arrive to us incomplete or noisy. We consider several possibilities listed in Table 5.3. The simplest setting is what we call SL, where layers are randomly reordered, but all other information required for the solution of the influence maximization problem is preserved. SL is the same setting already considered in the study of the sensitivity of the outbreak size to the temporal ordering of the network layers (Figures 5.2 and 5.3). We remark that, in the SL setting, we still rely on the same exact scheme as described for greedy optimization. Thus, to perform the selection step of Equation 5.7, we run multiple numerical simulations of SIR dynamics by assuming that we know the true values of the spreading and recovery probabilities of the spreading model, but also that we believe that the true network dynamics is given by the specific SL setting at our hand.

We then consider cases where part of the temporal information is not present in our input data. For example, we consider the setting FL where only the first temporal layer is used in the solution of the problem. The setting RL is analogous to FL with the only difference that one randomly chosen layer is selected to play the role of the static network. In these cases, we perform standard greedy optimization under the hypothesis of having a static network topology, and we assume that the critical value of the spreading probability is given by the

Approximation	Time horizon	Time order	Temporal layers
GR	yes	yes	all
INMFA	yes	yes	all
RND	no	no	none
SL	yes	no	all
FL	no	no	one
RL	no	no	one
ST	no	no	aggregate
AD-F	no	no	one
AD-A	no	no	aggregate

Table 5.3: **Identification of influential spreaders in temporal networks.** We list here the various approximations used in the solution of the influence maximization problem. From left to right, the columns of the table report the acronym of the approximation, awareness by the approximation about the existence of a temporal horizon in the spreading, awareness by the approximation about the temporal evolution of the network topology, number/type of temporal layers used in the approximation. The various approximations are described in the text.

one of the static network.

We then consider a scenario where temporal information is flattened. The solution to the influence maximization problem relies on an aggregated version of the network and no temporal horizon is provided in the estimate of the outbreak size. We name this setting ST. We perform standard greedy optimization under the hypothesis of having a static network topology, and we assume that the critical value of the spreading probability is the one valid for the aggregated static network.

Also, we consider further approximations where the optimization problem is solved using adaptive degree (AD) centrality computed using full or partial information of the system topology. Essentially, the function $F(\mathcal{X}_{k-1} \cup v)$ appearing in Equation 5.7 equals the number of connections of node v with nodes that do not belong to the set $\mathcal{X}_{k-1}$. Specifically, we consider the approximations AD-F and AD-A where, respectively, the first layer or the aggregation of all layers are used for the computation of the adaptive degree centrality. The method in this case relies on topological information only, and there is no need of feeding the algorithm with information about the spreading model and its parameter values.

Finally, we consider the setting RND, where the seed set is built by randomly selecting nodes. This is the only viable option in case nothing is known about the network, and should provide the worst performance possible.

5.3 Results

We apply the different approximations of Table 5.3 in the identification of the influential spreaders in the 12 temporal networks constructed from the datasets of Table 5.1. We consider four distinct values of the recovery probability $\mu = 0$, $\mu = 0.25$, $\mu = 0.5$, and $\mu = 1$. For each network and μ value, we consider the critical value $\lambda_c(\mu)$ of the spreading probability as shown in Table 5.2. We finally consider separately the cases $\lambda = 0.5\lambda_c(\mu)$, $\lambda = \lambda_c(\mu)$, and $\lambda = 2\lambda_c(\mu)$ as representative for the subcritical, critical, and supercritical dynamical regimes, respectively. In summary, for each of the 12 temporal networks we consider 12 distinct combinations of μ and λ values, so that each approximation for the solution of the influence maximization problem is tested in 144 different experimental settings.

We stress that the true values of the spreading probability are used only for predictions under the GR, INMFA, and SL settings. The other methods assume different critical values for λ, and predictions for the various regimes are made using such a value as a reference. Predictions of all approximations are tested on the ground-truth dynamics. That is, given a set of predicted seeds, we run numerical simulations of the spreading process using true parameter values on the true temporal network.

A typical outcome of the systematic analysis we perform is displayed in Figure 5.4. There, we plot the average value of the relative outbreak size as a function of the relative size of the seed set. We display results only for a selection of the approximations listed in Table 5.3, and only for the critical regime of spreading. Results for the other methods, and for other dynamical regimes, are reported in Appendix C. The best solution is generally obtained by GR, *i.e.*, the straight implementation of the greedy optimization strategy of Equation 5.6. This is not surprising as the method relies on complete topological and dynamical in-

formation. Although $\langle O \rangle$ is not submodular, the shape of the curve indicates an effective submodular behavior resulting from the maximization process. Results obtained under IN-MFA are generally very close to (sometimes even slightly better than) those of GR. As one may expect, the worst performance is obtained by random selection, *i.e.*, RND. Using FL provides a quite good performance despite other temporal information being neglected. SL, where one is aware of network evolution, but does not know exactly how the temporal layers are ordered, displays poor performance. AD-F provides a fair approximation in terms of performance even though it has no information on spreading dynamics.

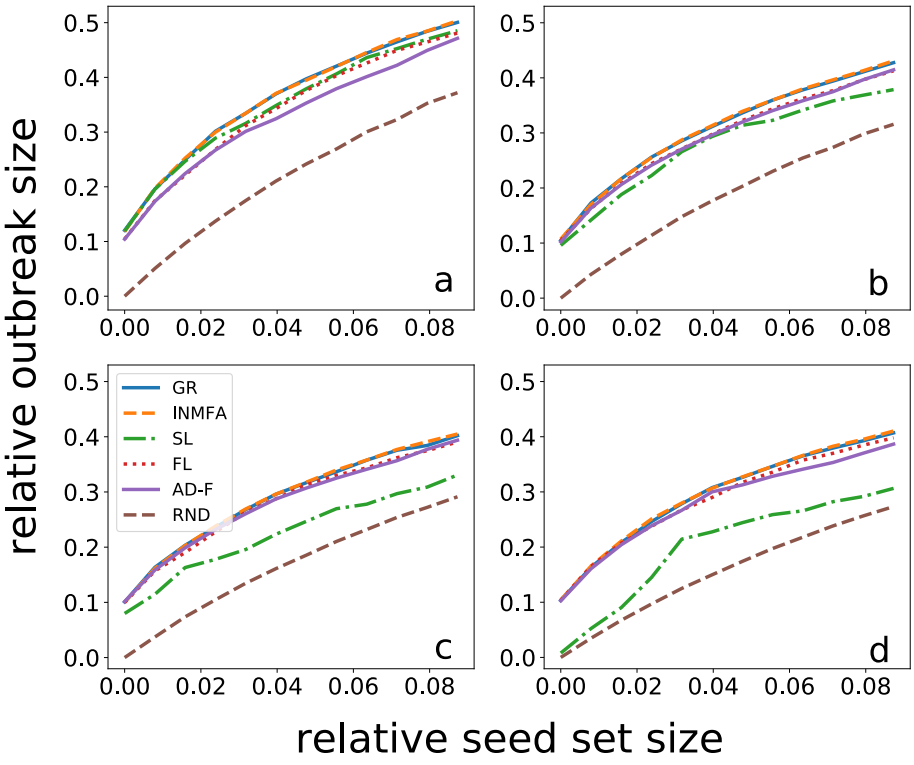

Figure 5.4: **Identification of influential spreaders in temporal networks.** (a) Average value of the relative size of the outbreak, *i.e.*, $\langle O(\mathcal{X}) \rangle$, as a function of the relative size of the seed set, *i.e.*, $|\mathcal{X}|/N$. The seed set is selected according to some of the approximations described in the text and listed in Table 5.3. The network analyzed is "High school, 2011." Spreading dynamics is critical, with recovery probability $\mu = 0$ and $\lambda = \lambda_c(\mu) = 0.037$. (b) Same as in panel a, but for $\mu = 0.25$ and $\lambda = \lambda_c(\mu) = 0.057$. (c) Same as in panel a, but for $\mu = 0.5$ and $\lambda = \lambda_c(\mu) = 0.078$. (d) Same as in panel a, but for $\mu = 1$ and $\lambda = \lambda_c(\mu) = 0.116$.

We use GR as the baseline for assessing the quality of the solutions obtained under the

other approximations. Given a network with N nodes, we fix the targeted seed set size to $K = 0.1N$. For the generic approximation a, we evaluate the area under the curve of the spreading impact of the seed set $\mathcal{X}^{(a)}$ that the approximation identifies, *i.e.*,

$$q_a = \sum_{k=1}^{K} \langle O(\mathcal{X}_k^{(a)}) \rangle, \tag{5.8}$$

where $\mathcal{X}_k^{(a)}$ is the seed set found by the approximation a in the kth step of the optimization algorithm of Equation 5.7. The definition can be clearly adapted to compute q_{GR}, thus leading to a metric of performance for the straight implementation of the greedy algorithm of Equation 5.6. We note that the metric q_a gives importance to the global impact of the seed set $\mathcal{X}_K^{(a)}$ but also to the order in which nodes are placed in the set K during the optimization steps of Equation 5.7. We then normalize the performance q_a of the generic approximation a with GR by simply taking the ratio

$$g_a = \frac{q_a}{q_{GR}}. \tag{5.9}$$

In Figure 5.5, we report summary results of our systematic analysis. Performance of the various approximations highly depend on both the parameters μ and λ. Many of the approximations reach nearly optimal performances for small μ and λ values. As μ grows, having perfect knowledge of the initial topology of the network, such as in the FL or AD-F approximations, becomes essential to reach good performance. Approximations that do not rely on such a knowledge lose 10–20% in performance compared to the performance displayed by the same approximations at low μ values.

The analysis so far is based on evaluating the performances of ex-post approximations. However, in more realistic settings, the observer might only be aware of past snapshots, and has to make decisions based on the available information. As an illustrative example, in Figure 5.6, we show the performance results obtained by comparing ex-ante predictions of the influential spreaders under approximations similar to those we considered earlier.

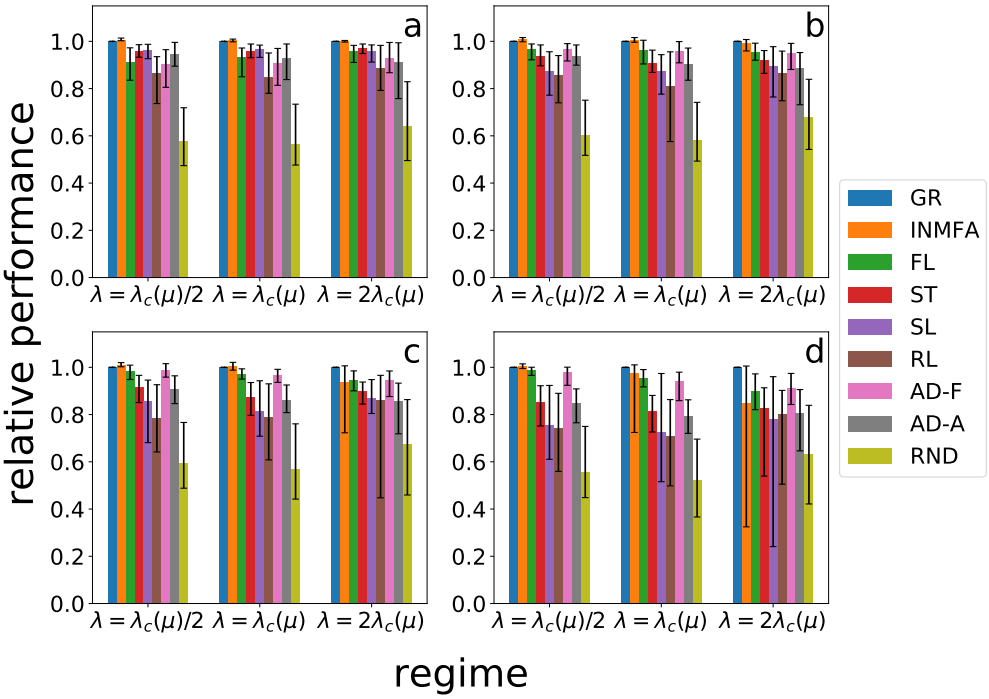

Figure 5.5: **Relative performances of methods for identifying influential spreaders.** (a) Performance, as defined in Equation 5.9, of the various approximations listed in Table 5.3. Performance values are relative to those obtained for GR. The height of the colored bars indicate average values of the relative performance over the set of the twelve temporal networks studied, see Table 5.1. Error bars identify minimum and maximum values of the performance measured over the entire corpus of real networks. We study different dynamical regimes by selecting different spreading probability values while keeping the recovery probability fixed at $\mu = 0$. (b) Same as in (a), but for $\mu = 0.25$. (c) Same as in (a), but for $\mu = 0.5$. (d) Same as in (a), but for $\mu = 1$.

Specifically, we name as $\text{FL}^{(t)}$ the approximation relying on layer t as the only information available about the network, and with $\text{AD-F}^{(t)}$ the approximation relying on adaptive degree centrality computed on the tth layer of the network. We use layers $t = 1, 2$, and 3 to make predictions about the spreaders in the temporal network. The true dynamics starts from layer $t = 3$ in the simulations. We measure the performance of our predictions relative to the best achievable one, here named as $\text{GR}^{(t=3)}$. The results of Figure 5.6 show that the lack of information about the initial layer of the dynamics leads to a significant drop in performance.

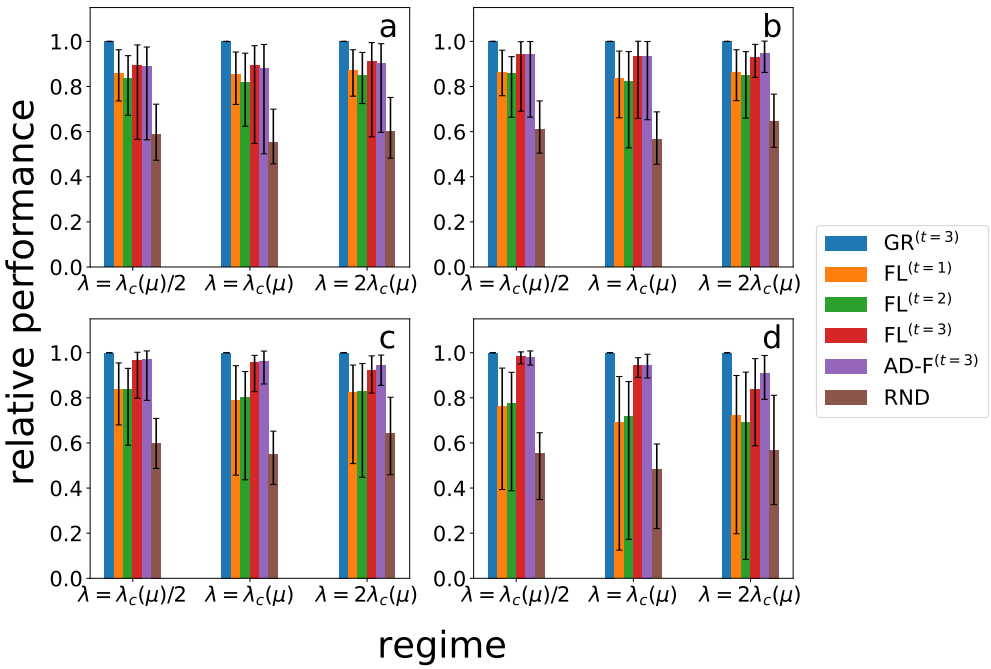

Figure 5.6: **Relative performances of methods for identifying influential spreaders.** (a) Same as in Figure 5.5(a) with the difference that the ground-truth dynamics is started at time $t = 3$ instead of time $t = 1$. Predictions using the GR$^{(t)}$, FL$^{(t)}$ and AD-F$^{(t)}$ approximations are based on perfect knowledge of the network topology/dynamics, but under the assumption that spreading starts at time t. (b) Same as in (a), but for $\mu = 0.25$. (c) Same as in (a), but for $\mu = 0.5$. (d) Same as in (a), but for $\mu = 1$.

5.4 Conclusion

Irreversible spreading models, such as the SIR model, display outcomes that strongly depend on the initial conditions. Such a sensitivity is already apparent in static networks, but it gets amplified when the network exhibits temporal changes on a timescale comparable with the one of the spreading dynamics. While seeking solutions to the problem of influence maximization, sensitivity of the spreading outcome to initial conditions is further extremized. Indeed, our systematic analysis shows that good solutions to the influence maximization problem require accurate knowledge of the network dynamics, especially regarding the order in which network edges appear in the system. Also, one of our most important numerical findings is that having knowledge of only the first snapshot of a temporal network is still sufficient for identifying influential spreaders effectively. The topological characteristics of

the nodes selected as spreaders depend on the dynamical regime. If the recovery probability μ is large, then nodes that are central in the first few layers are good spreaders. If μ is small instead, then nodes that are central on average over all layers are good spreaders. For example, we see that AD-F outperforms AD-A in all settings except for the subcritical and critical regimes when $\mu = 0$.

Our main analysis is based on the evaluation of the performance of different ex-post approximations. In practical settings, however, it may be more realistic to expect the observer to be aware of past snapshots of a temporal network, and use this information to make predictions about top influencers for a future spreading process taking place on temporal network layers with unknown topology. When we compare the performances obtained by comparing ex-ante predictions of influential spreaders under approximations similar to those we considered before, we see that the lack of information about the initial layer of the dynamics leads to a significant drop in performance. Even a small delay between the last known layer and the start of the process may significantly affect the results. This fact indicates that further research is needed to design effective methods for the prediction of influential spreaders in temporal networks.

6 Effective submodularity of influence maximization on temporal networks

6.1 Introduction

In Chapter 5, we studied the influence maximization problem under the susceptible-infected-recovered (SIR) model on temporal networks. We performed a systematic analysis on 12 real-world temporal networks and analyzed the performances of different approximation methods that have different levels of knowledge on the network topology and dynamics. We found that complete knowledge of a network, but in an aggregate way, is not helpful in solving the problem effectively. On the other hand, knowledge of the initial stages of the network helps to find good solutions to the influence maximization problem. The influence function of the SIR model on temporal networks is not submodular, except for the trivial case when the model is equivalent to the SI model (*i.e.*, $\mu = 0$). However, the solutions provided by the greedy algorithm proved to be good upper bounds for the performances of other methods to identify influential spreaders. This fact suggests that even though there is no theoretical proof of the performance of the greedy algorithm, in practice the optimization strategy is effective in approximating solutions to the influence maximization problem. This brings the questions of how often the condition for submodularity is violated and how far the solutions of the greedy algorithm are from the ground-truth optimum.

We answer the above questions in this section. In particular, we provide evidence for an effective submodular behavior of the influence function under greedy optimization. Results of our analysis show that the condition for submodularity is violated frequently for randomly selected seeds. However, when seeds are selected with the greedy algorithm, the frequency drops almost to zero, especially for real-world networks. Also, we show that the solutions of the greedy algorithm have a performance very close to the optimal solution found with brute-force search.

6.2 Methods

6.2.1 Temporal networks

We model temporal networks the same way as in Section 5. A temporal network is represented as a collection of T ordered network layers, namely, $A^{(0)}, A^{(1)}, \ldots, A^{(t)}, \ldots A^{(T-1)}$, each representing the topology of the system at a specific time. All layers of a temporal network are composed of the same N nodes, with labels uniquely identifying the nodes across the layers.

There are $R(N,T) = 2^{T\binom{N}{2}}$ total possible temporal networks with N nodes and T layers. There are, in fact, $2^{\binom{N}{2}}$ different labeled networks with N nodes. Those networks can be permuted in $(2^{\binom{N}{2}})^T$ ways, where permutations also include repetitions.

We consider exhaustive enumerations of all possible temporal networks in only one experiment. Clearly, we choose very small values of the parameters N and T to make the enumeration computationally feasible. In all other experiments, the space of potential temporal networks is sampled by either constructing synthetic models or leveraging real data.

6.2.2 Synthetic temporal network model

We generate random temporal networks with correlated layers. Specifically, the layer $A^{(t+1)}$ of the temporal network is obtained by copying all the edges in the layer $A^{(t)}$, and then shuffling with probability r the end points of each individual edge with those of another random edge. For instance, if the edge (i,j) undergoes shuffling, we first select at random another edge (v,w). We then verify that the edges (i,w) and (v,j) are not yet present in the network. If they are present, we select another edge (v,w) and repeat the operation. Otherwise, we shuffle their end points in the sense that the edges (i,j) and (v,w) are removed from the network, and they are replaced by the edges (i,w) and (v,j). The above procedure of shuffling the end points of edges keeps the degree sequence of the network layers unchanged. For $r = 0$, we have $A^{(t+1)} = A^{(t)}$ for all t, *i.e.*, layers are perfectly correlated and the

temporal network is essentially a static network. For $r = 1$, all edges are surely shuffled, so no correlation exists between $A^{(t+1)}$ and $A^{(t)}$ except for the fact that they have the same degree sequence. In our experiments, we start by generating the first layer $A^{(0)}$ according to the Erdős-Rényi model (140) with average degree k.

Also, we consider a simple uncorrelated model where all layers are generated according to the Erdős-Rényi model with average degree k. Networks of this type are statistically equivalent to those created according to the model above with reshuffling probability $r = 1$, with the only difference that whereas the average degree of the network is invariant across network layers, the degree sequence is not preserved across network layers. In all cases where the value of the parameter r is not specified, we will take advantage of this simple model to generate uncorrelated random temporal networks.

We note that the hypothesis of having an average degree that is invariant across layers is not necessarily a characteristic of real-world temporal networks. To study the problem of influence maximization in realistic settings, we generate temporal networks directly from real data without the need to make any assumption.

6.2.3 Real-world temporal networks

We use 12 empirical datasets containing time-stamped interactions between pairs of nodes. The datasets used are the same ones used in Section 5, and the networks are constructed from these datasets following the exact same procedure. The networks used are listed in Table 5.1.

6.2.4 Spreading dynamics

We consider the discrete version of the susceptible-infected-recovered (SIR) model for spreading dynamics. We use the same model as in Section 5. However, to properly compare the importance of the initial conditions on the long-term behavior of the model, we generate individual instances of the SIR model in a slightly different manner. Under this procedure,

the difference in impact of different initial conditions is measured on identical, deterministic dynamical systems. We then average the difference over multiple individual instances of the SIR model.

Before starting any dynamics on the network, we generate the rth instance of the SIR model with parameters λ and μ by determining the propensity of individual edges to spread the infection and the propensity of individual nodes to recover at particular instants of time. Specifically, for each edge (i, j) appearing at time t, we set the spreading propensity of the edge $\rho^{(r)}_{(i,j)}(t) = 1$ with probability λ; otherwise, we set $\rho^{(r)}_{(i,j)}(t) = 0$. Also, we set the recovery propensity of node i at time t as $\rho^{(r)}_i(t) = 1$ with probability μ, and $\rho^{(r)}_i(t) = 0$ otherwise.

Once propensities of edges and nodes are set for all temporal layers of the network, SIR dynamics can be run in a deterministic fashion starting from the initial condition $\vec{\sigma}(t = 0) = [\sigma_1(t = 0), \ldots, \sigma_N(t = 0)]$, where $\sigma_i(0) = S, I$, or R. Please note that the initial condition does not depend on the actual realization of the SIR model. The rules that determine the dynamics for $t > 0$ are as follows. Indicate with $\sigma^{(r)}_i(t)$ the dynamical state of node i at time t in the rth realization of the SIR model. We have that

$$\sigma^{(r)}_i(t+1) = S \text{ if } \begin{cases} \sigma^{(r)}_i(t) = S \\ \bigwedge \\ \nexists j \,|\, \sigma^{(r)}_j(t) = I \wedge \rho^{(r)}_{(i,j)}(t) = 1 \end{cases}, \tag{6.1}$$

meaning that node i remains in the state S if it does not get in active contact with any infected neighbor. Also, we have that

$$\sigma^{(r)}_i(t+1) = I \text{ if } \sigma^{(r)}_i(t) = I \wedge \rho^{(r)}_i(t) = 0 \tag{6.2}$$

and

$$\sigma_i^{(r)}(t+1) = I \text{ if } \begin{cases} \sigma_i^{(r)}(t) = S \\ \bigwedge \\ \exists j \,|\, \sigma_j^{(r)}(t) = I \wedge \rho_{i,j}^{(r)}(t) = 1 \end{cases} . \tag{6.3}$$

Equation 6.2 describes the case of a node already infected that does not recover. Equation 6.3 accounts instead for the change of the dynamical state of the node i getting infected because of an active contact with at least one infected neighbor. Finally, we have that

$$\sigma_i^{(r)}(t+1) = R \text{ if } \begin{cases} \sigma_i^{(r)}(t) = I \wedge \rho_i^{(r)}(t) = 1 \\ \bigvee \\ \sigma_i^{(r)}(t) = R \end{cases} , \tag{6.4}$$

i.e., node i recovers if infected and prone to recovery at time t, or it does not change its state if it already recovered. After all the above operations are executed for all nodes i, time increases as $t \to t+1$.

Assuming that network evolution and spreading dynamics happen in discrete time and precisely at the same time scale simplify the numerical and analytical analysis of the dynamical system. We expect results obtained under these simplifications to be valid also for temporal networks evolving in continuous time as long as the duration of individual edges is sufficiently homogeneous and provided that the SIR model is reformulated in continuous time (32). We recognize, however, that our framework should likely fail to properly describe SIR dynamics happening on temporal networks characterized by heterogeneous activity of the edges.

In our experiments, we restrict our attention to initial conditions where all nodes are in the susceptible state except for the nodes in the seed set $\mathcal{X}$ which are in the infected state, *i.e.*, $\sigma_i(t=0) = I$ if $i \in \mathcal{X}$ and $\sigma_i(t=0) = S$ if $i \notin \mathcal{X}$. Starting from such initial configurations, the dynamics in a network with T layers is simulated until the stage T. The outbreak size $O^{(r)}(\mathcal{X})$ at the end of the rth realization of the process is calculated as the

total number of infected and recovered nodes at T, *i.e.*,

$$O^{(r)}(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N}\left[\mathbb{1}_{\sigma_i^{(r)}(T),I} + \mathbb{1}_{\sigma_i^{(r)}(T),R}\right] ,$$

where $\mathbb{1}_{x,y}$ is the identity operator, *i.e.*, $\mathbb{1}_{x,y} = 1$ if $x = y$ and $\mathbb{1}_{x,y} = 0$ otherwise.

We indicate the marginal gain in influence of adding node v to the seed set $\mathcal{X}$ as

$$O_{\mathcal{X}}^{(r)}(v) = O^{(r)}(\mathcal{X} \cup \{v\}) - O^{(r)}(\mathcal{X}) . \tag{6.5}$$

We estimate the influence of the set $\mathcal{X}$ by taking the average value of the outbreak size over R independent realizations of the SIR model, *i.e.*,

$$\langle O(\mathcal{X}) \rangle = \frac{1}{R}\sum_{r=1}^{R} O^{(r)}(\mathcal{X}) . \tag{6.6}$$

6.2.5 Influence maximization

Influence maximization is defined as in Chapter 5.2.4. Exact solutions of the problem are obtainable via brute-force search over all possible $\binom{N}{K}$ ways of choosing K seed nodes out of N total nodes in the network. For each of these candidate sets, the influence function of Equation 6.6 should be evaluated. Clearly, the brute-force search can only be applied on relatively small networks and small seed set sizes. In most of the practical settings, the solution of the problem of Equation 5.6 can be only approximated.

6.2.6 Greedy optimization

Approximate solutions to the problem can be obtained using a greedy optimization algorithm, defined as in Chapter 5.2.4. On static networks, the influence function of Equation 6.6 is a submodular function with non-negative marginal gains. These two properties guarantee that the solution provided by the greedy algorithm is at max $1 - 1/e$ times away from the ground-truth optimal solution (138). On temporal networks, such an optimality bound is

guaranteed only for $\mu = 0$. For $\mu > 0$, the influence function is not necessarily a submodular function with non-negative marginal gains, hence there is no guarantee on the optimality gap for the solutions obtained via the greedy algorithm (130).

6.2.7 Submodularity

In their work, Kempe *et al.* showed that the influence function of Equation 6.6 on static networks has non-negative marginal gains and is a submodular function (9).

The influence function has non-negative marginal gains if

$$O_{\mathcal{A}}(v) \geq 0 \ , \tag{6.7}$$

for any set $\mathcal{A}$ and for any node v. Influence is a submodular function if it satisfies the submodularity or "diminishing returns" condition, *i.e.*,

$$O_{\mathcal{A}}(v) \geq O_{\mathcal{B}}(v) \ , \tag{6.8}$$

for all nodes $v \notin \mathcal{B}$ and for all sets of nodes $\mathcal{A} \subseteq \mathcal{B}$. This means that the marginal gain obtained by adding node v to the set $\mathcal{A}$, a subset of $\mathcal{B}$, must be greater than or equal to the marginal gain obtained by adding node v to the set $\mathcal{B}$.

As we already mentioned, properties of Equations 6.7 and 6.8 hold for the SIR model on static networks. They hold for temporal networks too as long as the recovery probability $\mu = 0$. However, they are generally not valid on temporal networks for $\mu > 0$. Violations of the condition of Equation 6.8 may happen in three main ways, as illustrated in Figure 6.1a. Four scenarios are realized depending on whether the marginal gains $O_{\mathcal{A}}(v)$ and $O_{\mathcal{B}}(v)$ are non-negative or negative. For $O_{\mathcal{A}}(v) \geq 0$ and $O_{\mathcal{B}}(v) < 0$ in (ii), the diminishing returns condition holds; for $O_{\mathcal{A}}(v) < 0$ and $O_{\mathcal{B}}(v) \geq 0$ in (iii), the diminishing returns condition is violated. In the other two scenarios illustrated in panels (i) and (iv), it depends on the actual values of the marginal gain.

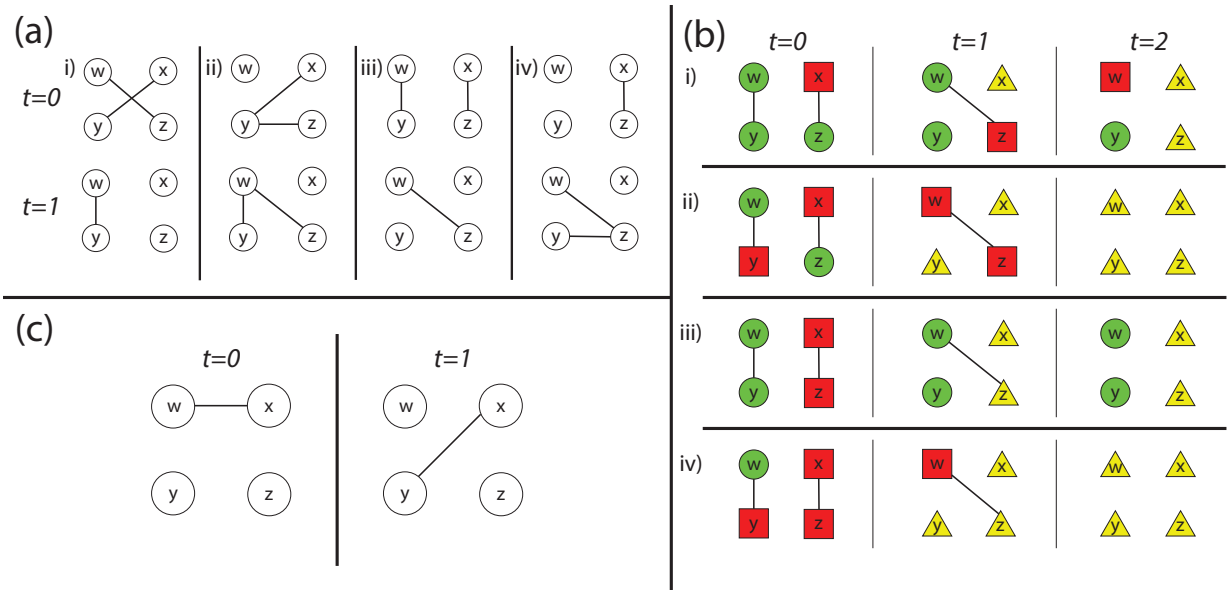

Figure 6.1: **Violation of the necessary conditions for the submodularity of the influence function on temporal networks.** For simplicity, we consider the deterministic case of the SIR model where the probabilities of infection and recovery are $\lambda = \mu = 1$. (a) We display four possible scenarios for the marginal gain of adding node v to sets $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{A} = \{x\}, \mathcal{B} = \{x, y\}, v = z$. The cases are separated on the basis of the marginal gains being negative or not. The marginal gains in the four scenarios are: (i) $O_{\mathcal{A}}(v) = 1, O_{\mathcal{B}}(v) = 2$, (ii) $O_{\mathcal{A}}(v) = 1, O_{\mathcal{B}}(v) = -1$, (iii) $O_{\mathcal{A}}(v) = -1, O_{\mathcal{B}}(v) = 0$, (iv) $O_{\mathcal{A}}(v) = -2, O_{\mathcal{B}}(v) = -1$. The inequality 6.8 is violated in (i), (iii), and (iv). (b) A counter-example for the submodularity of the influence function. Let $\mathcal{A} = \{x\}, \mathcal{B} = \{x, y\}, v = z$. Green circles denote nodes in the susceptible state, red squares denote nodes in the infected state, and yellow triangles denote nodes in the recovered state. (c) A counter-example for the γ-weakly submodularity of the influence function. Let $\mathcal{A} = \{w\}, \mathcal{B} = \{x, y, z\}$.

The inspection of Figure 6.1a reveals that violations of the conditions necessary for the submodularity property are caused by recovered nodes blocking the paths of future infections. In static networks, a recovered node would already have exploited any possible path to infect its neighbors. The difference in temporal networks is that the neighbors of a node change in time. This means that the infection and recovery time of a node are fundamental factors that determine which paths are effectively used by the infection to propagate. In particular, an early infection of a node may be detrimental for the long-term fate of the spreading process just because once recovered the node may block paths that would have been otherwise available if the node was still in the susceptible state.

For example, in Figure 6.1b we set $\mathcal{A} = \{x\}, \mathcal{B} = \{x, y\}$ and $v = z$. We also set $\lambda = \mu = 1$ for simplicity. Each row shows the initial conditions at $t = 0$ for the seed sets $\mathcal{A}, \mathcal{A} \cup v, \mathcal{B}$, and $\mathcal{B} \cup v$, respectively. Infections and recoveries occur at $t = 0$ and $t = 1$, and the final configuration is seen at $t = 2$, where no more infections or recoveries happen. From Figure 6.1b, we can see that $O(\mathcal{A}) = 3$ in (i), $O(\mathcal{B}) = 4$ in (ii), $O(\mathcal{A} \cup v) = 2$ in (iii), and $O(\mathcal{B} \cup v) = 4$ in (iv). Here, the condition of Equation 6.8 is violated. The main reason of the violation is the premature infection of node z. The marginal gain of adding node z to set $\mathcal{A}$ is $O_{\mathcal{A}}(z) = -1$. z recovers at $t = 1$, thus is not able to infect w. The infection could have occurred if the node was susceptible at the beginning of dynamics. So, by adding node z to the set $\mathcal{A}$, the path to infecting node w is blocked by the premature infection and recovery of node z, thus decreasing the total outbreak size at the end of the dynamics.

A toy example of how multiple, misplaced seeds may have dramatic consequences on the size of the outbreak is provided in Figure 6.2. This is a rather specific and unrealistic example, where a single edge is present at each layer. We expect, however, that the very same issue, although in more complicated forms, is at the basis of violations of the necessary conditions for the submodularity of the influence function in real temporal networks.

6.2.8 γ-weakly submodularity

After showing that the influence function is not submodular, we can test weaker definitions of submodularity that would give us looser optimality gaps. In their work, Santiago and Yoshida defined γ-weakly submodularity for non-submodular functions (141). In their definition, a function is γ-weakly submodular when

$$\sum_{v \in \mathcal{B}} O_{\mathcal{A}}(v) \geq \min\{\gamma O_{\mathcal{A}}(\mathcal{B}), \frac{1}{\gamma} O_{\mathcal{A}}(\mathcal{B})\} \tag{6.9}$$

for any disjoint sets of nodes $\mathcal{A}$ and $\mathcal{B}$. In the above inequality, $0 < \gamma \leq 1$. When the inequality holds, it is possible to find a solution with a so-called randomized greedy algorithm resulting in an optimality gap equal to $1 - \gamma e^{-1/\gamma}$.

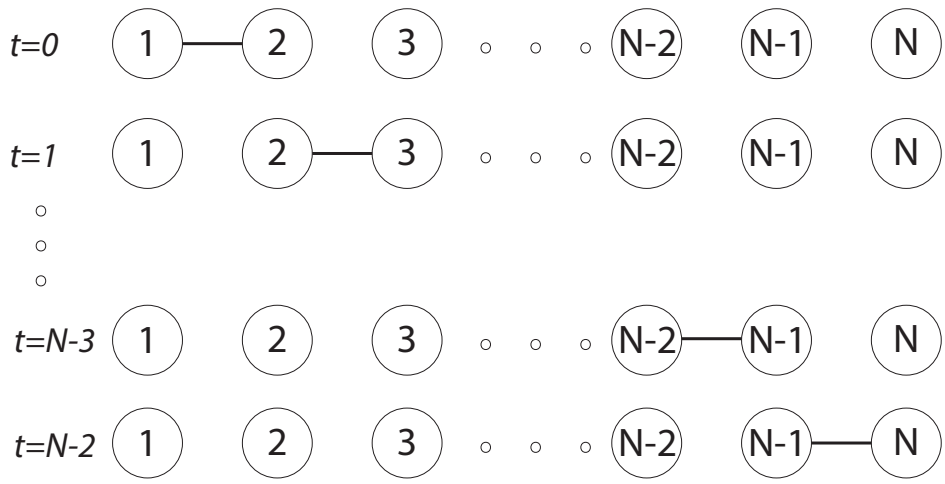

Figure 6.2: **Blocking paths of future infections with additional seeds.** We display a toy network where increasing the number of seeds have catastrophic effects on the outbreak size of the spreading process. For simplicity we consider the deterministic case of the SIR model where the probabilities of infection and recovery are $\lambda = \mu = 1$. Setting node 1 as the only seed of the process leads to maximum spread in the network, *i.e.*, $O(\{1\}) = 1$. However, adding another node $i > 1$, except for node N, to the seed set generates a reduction in the influence function, *i.e.*, $O(\{1, i\}) = i/N$.

Unfortunately, the influence function of the SIR model on temporal networks is not γ-weakly submodular. An example of the violation of the condition of Equation 6.9 is shown in Figure 6.1c. We select $\mathcal{A} = \{w\}, \mathcal{B} = \{x, y, z\}$ and $\lambda = \mu = 1$. The marginal gains are $O_{\mathcal{A}}(\mathcal{B}) = 1, O_{\mathcal{A}}(x) = -1, O_{\mathcal{A}}(y) = 0$, and $O_{\mathcal{A}}(z) = 1$. The left hand side of the inequality 6.9 reads $\sum_{v \in \mathcal{B}} O_{\mathcal{A}}(v) = 0$. This means that for the inequality to hold we need $\gamma = 0$ or $\gamma = \infty$ because $O_{\mathcal{A}}(\mathcal{B}) = 1$. However, the definition of γ-weakly submodularity requires $0 < \gamma \leq 1$, meaning that the inequality does not hold.

6.3 Results

There are temporal networks where the inequality 6.8 is violated, and the influence function is not submodular. In these situations, the greedy algorithm does not provide any guarantee on the optimality of its solutions. However, it is still possible that the algorithm provides solutions close enough to the ground-truth optimum. To investigate this property, we perform a systematic analysis on synthetic and real-world networks.

6.3.1 Synthetic temporal networks

We calculate the frequency of violations of the inequality 6.8 as

$$g(\mathcal{A}, \mathcal{B}, v) = \frac{1}{R} \sum_{r=1}^{R} H(O_{\mathcal{B}}^{(r)}(v) - O_{\mathcal{A}}^{(r)}(v)) \ , \tag{6.10}$$

where $\mathcal{A} \subseteq \mathcal{B}$ and $v \notin \mathcal{B}$, and $H(x)$ is the Heaviside step function, *i.e.*, $H(x) = 1$ if $x > 0$ and $H(x) = 0$ otherwise. Please note that $O_{\mathcal{A}}^{(r)}(v)$ and $O_{\mathcal{B}}^{(r)}(v)$ are both computed on the same rth instance of the SIR model.

In Figure 6.3a, we consider all the $R(N = 4, T = 3) = 262,144$ possible temporal networks that can be formed with $N = 4$ nodes and $T = 3$ layers. Indicating the four nodes of the network as v, w, x, and y, we set $\mathcal{A} = \{y\}$ and $\mathcal{B} = \{x, y\}$. Please note that since we are considering all possible networks, the choice of the sets is irrelevant for our purposes. For a given network, we compute Equation 6.10 over $R = 100$ SIR model instances for each combination of μ and λ values. We then take the average value of $g(\mathcal{A}, \mathcal{B}, v)$ over all possible networks to generate the heatmap of Figure 6.3a. For $\lambda = 0.00$, the influence function is submodular, there is no spreading, and $O_{\mathcal{A}}(v) = O_{\mathcal{B}}(v) = 1$. For $\mu = 0.00$, the spreading model becomes equivalent to the SI model, which displays a submodular influence function in temporal networks. For other λ and μ values, violations of the inequality 6.8 are observed, maximum frequency of violations is registered for $\mu = 1.00$ and $\lambda = 0.90$.

We repeat a similar analysis on random temporal networks with $N = 100, T = 10, k = 5$, and $r = 1$. Results are reported in Figure 6.3b. For each run of the SIR model, we select three nodes at random, namely x, y, and v. We compose the sets $\mathcal{A} = \{y\}$ and $\mathcal{B} = \{x, y\}$, and compute Equation 6.10. Results in the figure are obtained by averaging $g(\mathcal{A}, \mathcal{B}, v)$ over 40 runs of the SIR model, and over 50 realizations of the random temporal network. The pattern revealed from the figure is similar to one obtained from the exhaustive analysis of Figure 6.3a.

Finally, in Figures 6.3c and 6.3d, we report results for two real-world temporal networks.

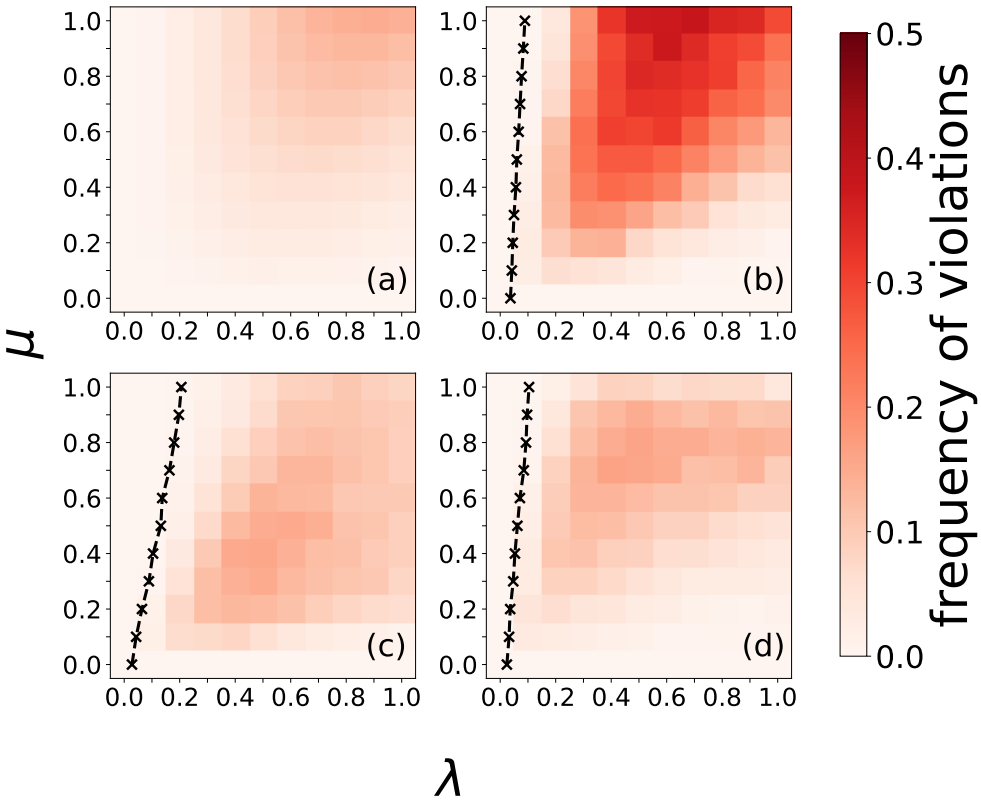

Figure 6.3: **Violations of the submodularity condition on temporal networks.** (a) We display the frequency of violations of the diminishing returns inequality, *i.e.*, g as defined in Equation 6.10, on random synthetic networks of size $N = 4$ as a function of the SIR parameters λ and μ. In the computation of Equation 6.10, the sets $\mathcal{A}$ and $\mathcal{B}$, and node v are selected randomly with $|\mathcal{A}| = 1, |\mathcal{B}| = 2$, and $\mathcal{A} \subset \mathcal{B}$. The simulations are run on a network with $N = 4$ and $T = 3$, and all possible configurations with this specific parameters have been used for the experiments. (b) Same as in panel (a), but only on a random temporal networks with $N = 100, T = 10$, $k = 5$, and $r = 1$. Results are averaged over 50 networks. The dashed black line shows the critical threshold values $\lambda_c(\mu)$ averaged over 10 networks. (c) Same as in panel (a), but for the real-world temporal network "High school, 2012." The dashed black line shows the critical threshold values $\lambda_c(\mu)$. (d) Same as in panel (a), but for the real-world temporal network "Hypertext, 2009."

In this specific case, we still select nodes at random, namely x, y, and v to compose the sets $\mathcal{A} = \{y\}$ and $\mathcal{B} = \{x, y\}$. We compute Equation 6.10 using $R = 2,000$ SIR simulations. For each realization, we randomly select the nodes x, y, and v. The pattern observed is similar to those of the previous two cases, although we observe less violations than in the case of random temporal networks. Also, we observe that the probability to observe a violation of the submodularity inequality is much higher in supercritical regime than in the subcritical

regime. The dynamical regime of the SIR process on the network is supercritical if $\lambda > \lambda_c(\mu)$, and subcritical for $\lambda < \lambda_c(\mu)$. Here, $\lambda_c(\mu)$ is the critical value of the spreading probability for a given value of the recovery probability μ.

We systematically study violations of the submodularity condition in random temporal networks. In Figure 6.4, we display the frequency of violations for different N, T, k, and r values. In each of our experiments, we first generate a network with a given set of parameters. Then, we select three nodes, namely x, y, and v, at random. We form the sets $\mathcal{A} = \{y\}$ and $\mathcal{B} = \{x, y\}$, and use these sets together with node v in Equation 6.10 to tell whether the inequality is violated or not. Please note that we consider $\mu = \lambda = 1.00$ in this set of experiments, so only one realization of the SIR model is possible. We consider $10,000$ networks, and record the average number of Equation 6.10 over such an ensemble. From Figure 6.4, we see that as N increases, the frequency of violation increases. For increasing T, we observe a phase transition from almost no violations to a non-null plateau value. For increasing k, there is a value where the frequency of violations of the diminishing returns property reaches a maximum; instead, when the network is either too sparse or too dense, the frequency drops to zero. As r increases, we see an increase in the violation frequency. Note that for $r = 0$, all the layers of the network are the same, meaning that the network is static and the influence function is submodular.

In Figure 6.4, we reshuffle edges with probability $r = 0.2$. However, a similar phenomenology can be obtained by creating layers independently as shown in Figure 6.5. Given the similarity of the results, from now on, we focus our attention on the simpler model where random temporal networks are composed of layers generated independently.

Also, we measure the frequency of violations of the inequality 6.7 as

$$\tilde{g}(\mathcal{A}, v) = 1 - \frac{1}{R} \sum_{r=1}^{R} H(O_{\mathcal{A}}^{(r)}(v)) \, . \tag{6.11}$$

The above equation quantifies how often the addition of the node v to the set $\mathcal{A}$ generates a marginal loss in the influence function. Results of our analysis are reported in Figure

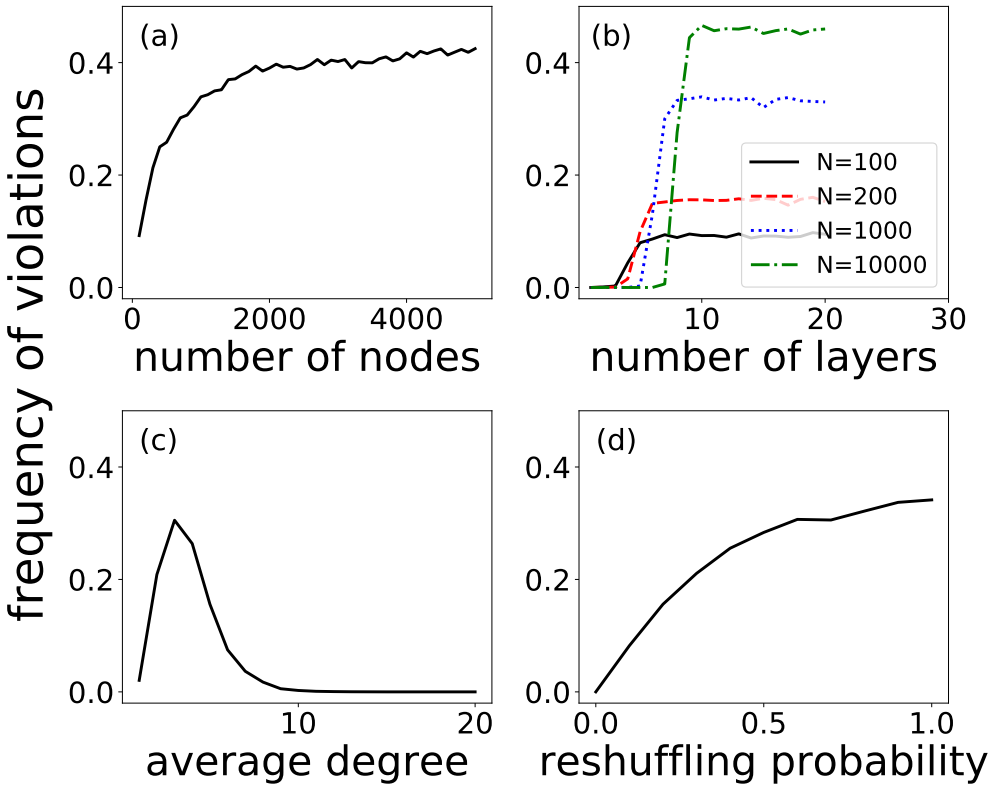

Figure 6.4: **Violations of the submodularity condition in synthetic temporal networks.** In all panels, unless stated otherwise, we consider $10,000$ networks composed of $N = 200$ nodes, average degree $k = 5$, total number of temporal layers $T = 10$, and probability of edge shuffle between consecutive layers $r = 0.2$. SIR parameters are $\lambda = \mu = 1.00$. (a) We display g in Equation 6.10 as a function of N. (b) We display g as a function of T for different values of N. (c) We display g as a function of k. (d) We display g as a function of r.

6.6. We set $\lambda = \mu = 1.00$, and consider $R = 1$ SIR simulations; in each simulation, we select at random two nodes, namely x and v. We set $\mathcal{A} = \{x\}$. We estimate $\tilde{g}$ by taking the average over $10,000$ different networks; for each network we sample $\mathcal{A}$ and v once. We observe that the frequency for observing a marginal loss by adding a node follows a very similar behavior as the frequency of violations of the submodularity inequality, see Figure 6.4. We take advantage of this similarity and focus our attention on marginal loss cases from now on, rather than measuring violations of the submodularity inequality. This allows us to alleviate some computational burden without altering the conclusions of the numerical analysis. We can in fact safely assume that the pattern of violations of the inequality 6.7

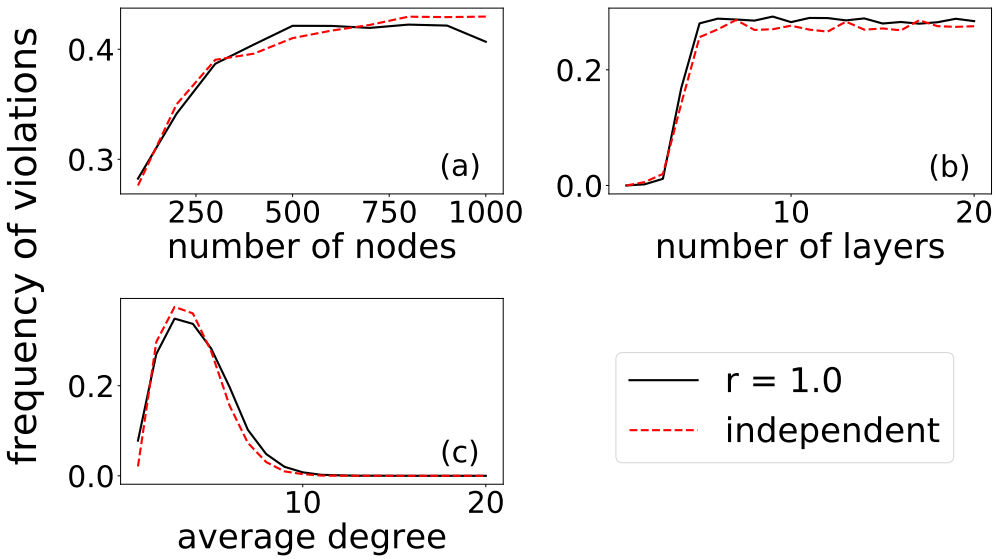

Figure 6.5: **Violations of the submodularity condition in synthetic temporal networks.** Frequency of violations of the inequality 6.8 on synthetic network models with $N = 100, T = 10, k = 5$ unless stated otherwise, and SIR parameters $\lambda = \mu = 1.00$. (a) g in Equation 6.10 as a function of N. Results obtained for $r = 1.0$ (black curve) are compared to the results obtained when layers in the temporal network are created independently (red curve). (b) We display g as a function of T. (c) We display g as a function of k.

are similar to the pattern of violations of the condition of Equation 6.8.

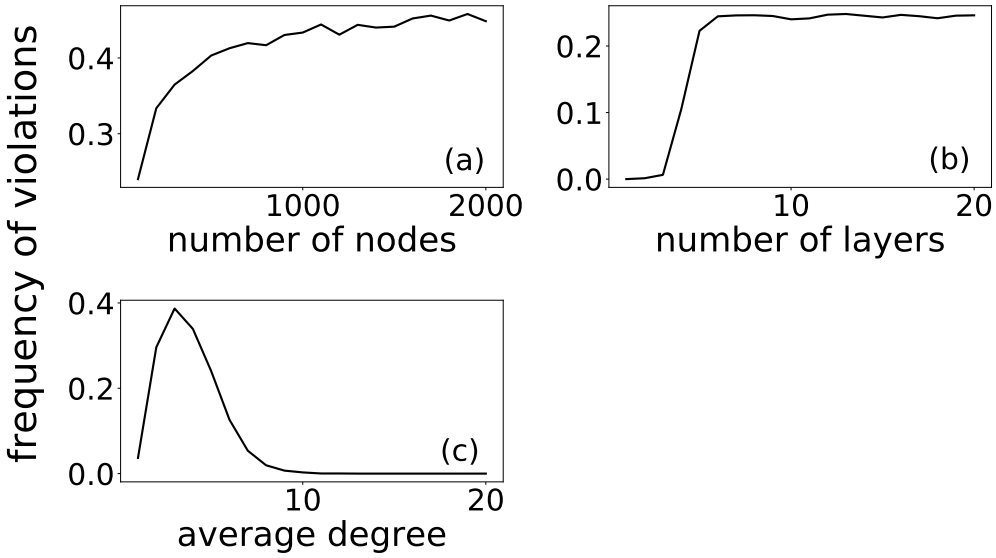

Figure 6.6: **Violations of the condition for marginal gain in synthetic temporal networks.** Frequency of marginal loss cases with random seeds on random temporal networks for $N = 100, T = 10, k = 5$ unless stated otherwise, and SIR parameters $\lambda = \mu = 1.00$. (a) $\tilde{g}$ in Equation 6.11 as a function of N. (b) We display $\tilde{g}$ as a function of T. (c) We display $\tilde{g}$ as a function of k.

All the results presented until now used $\lambda = \mu = 1.00$. In Figure 6.7, we show results valid for $\lambda < 1.00$ and $\mu \leq 1.00$. We see that the frequencies of inequality violations decrease as the recovery probability μ decreases, indicating that with a low recovery probability, it is less likely to observe violations of the inequality 6.7.

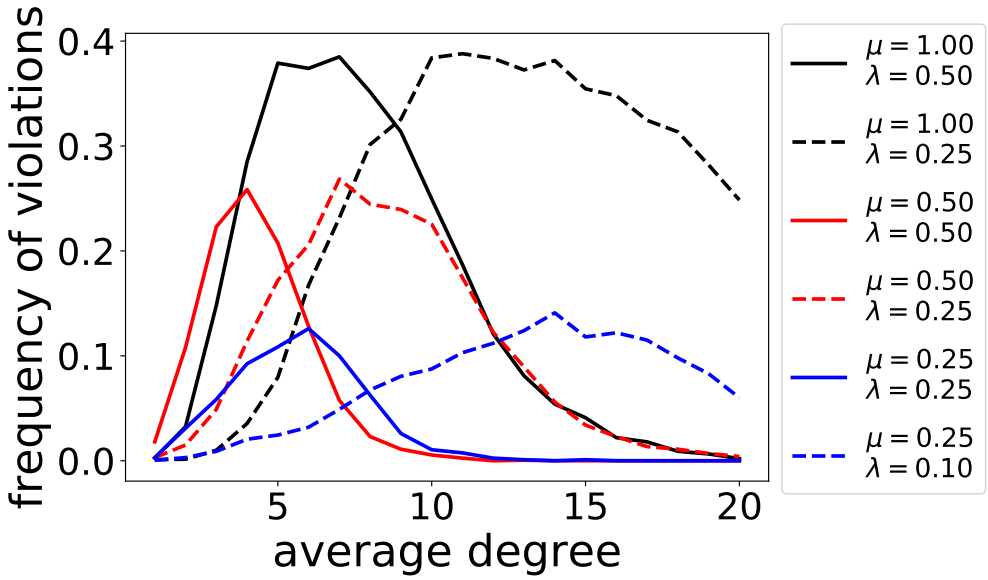

Figure 6.7: **Violations of the condition for marginal gain in synthetic temporal networks.**We measure the frequency of violations of the condition of marginal gain for the influence function using $\tilde{g}$ as defined in Equation 6.11. We analyze random temporal networks composed of $N = 100$ nodes and $T = 10$ layers. We consider different values of the average degree k, and of the SIR parameters λ and μ. In the estimation of Equation 6.11 we select $\mathcal{A}$ and v randomly, and we average over $R = 40$ instances of the SIR model. We further take the average over 50 temporal networks.

All results we obtained so far indicate that violations of the inequalities 6.7 and 6.8 may occur frequently if tested for sets of randomly chosen nodes. It is, however, natural to ask whether such an observation is valid also when nodes are selected according to the greedy optimization protocol of Equation 5.6. In our tests, we measure the marginal gain obtained by adding the kth node at the kth stage of the greedy algorithm. We perform the tests on $2,000$ random temporal networks for various values of the parameters N, T, and k. SIR simulations are performed for $\lambda = \mu = 1.00$. Results are obtained by averaging Equation 6.11 over the various network realizations, and are reported in Figure 6.8. The frequency of

violations of the inequality 6.7 under greedy selection decreases significantly compared to the case where nodes are randomly selected. For the initial stages of the greedy algorithm, the frequency is almost zero; we only start seeing non-null cases of marginal loss in late stages of the algorithm. We do not show the results, but we observe that, for $\lambda < 1.00$ and $\mu < 1.00$, the frequency of marginal loss becomes almost zero when selecting up to 10% of all nodes as initial spreaders.

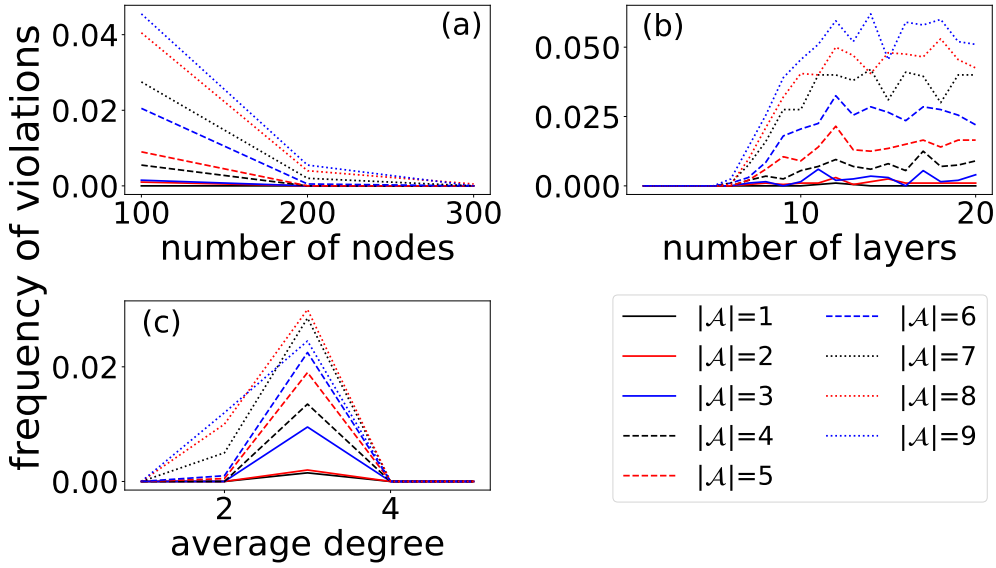

Figure 6.8: **Violations of the condition for marginal gain in synthetic temporal networks under greedy selection.** We measure the frequency of marginal losses using $\tilde{g}$ as defined in Equation 6.11 on random temporal networks for different sizes of $\mathcal{A}$. In all panels, unless stated otherwise, we consider networks composed of $N = 100$ nodes, $T = 10$ layers, and average degree $k = 2.5$. (a) We display $\tilde{g}$ as a function of N. (b) We display $\tilde{g}$ as a function of T. (c) We display $\tilde{g}$ as a function of k.

6.3.2 Real-world temporal networks

We analyze the frequency of violations of condition of Equation 6.7 on real-world temporal networks. We conduct our analysis on the networks shown in Table 5.1. Results of our analysis are reported in Figure 6.9. We observe the same phenomenology as for the case of random temporal networks. If initial spreaders are selected randomly, then the number of configurations for which inequality 6.7 does not hold is not negligible. On the other hand,

if spreaders are chosen according to the greedy strategy, then cases where the inequality 6.7 is violated are almost non-existent. This finding suggests that, even though the right conditions to apply greedy optimization are not satisfied, in practice the greedy algorithm might still work as intended, finding solutions close to the ground-truth optimum.

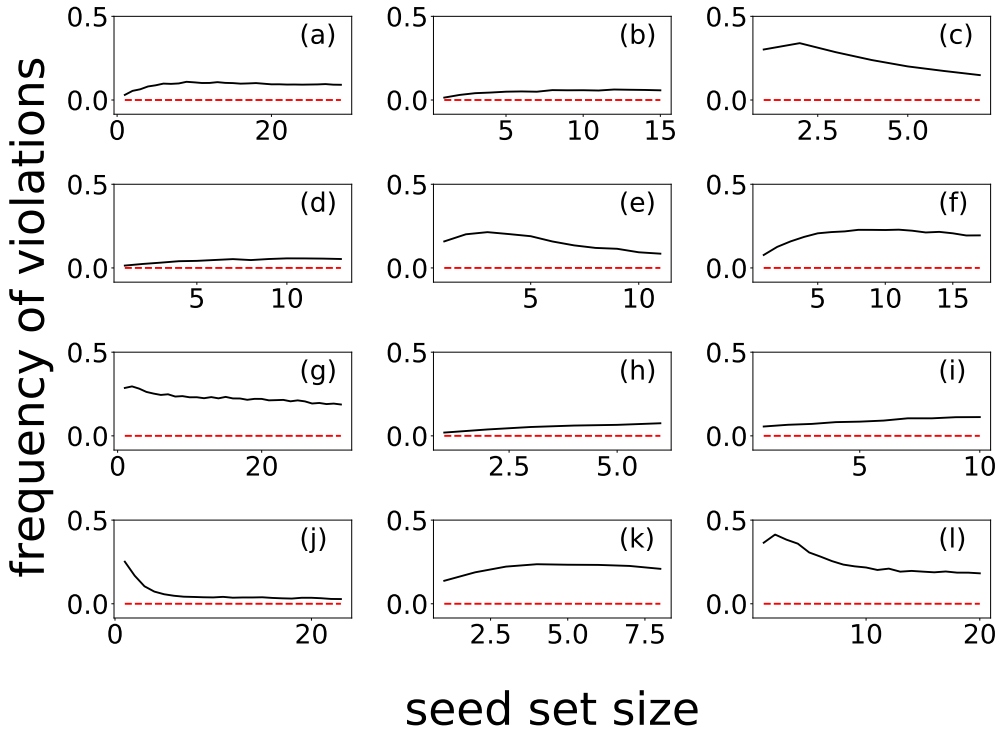

Figure 6.9: **Violations of the condition for marginal gain in real-world temporal networks.** We measure the frequency of marginal losses using $\tilde{g}$ as defined in Equation 6.11 on the real-world temporal networks listed in Table 5.1. Results are displayed as full black curves. Labels of the various panels reflect those appearing in the table. The SIR parameters are $\lambda = \mu = 1.00$, so that $R = 1$ in Equation 6.11. We select $\mathcal{A}$ and v randomly $10,000$ times, and display the average value of the violation of marginal gains. The red dashed curves represent the frequency values when seeds are selected according to greedy optimization. Each panel corresponds to a real-world temporal network: (a) Email, dept. 1; (b) Email, dept. 2; (c) Email, dept. 3; (d) Email, dept. 4; (e) High school, 2011; (f) High school, 2012; (g) High school, 2013; (h) Hospital ward; (i) Hypertext, 2009; (j) Primary school; (k) Workplace; (l) Workplace-2.

6.3.3 Greedy maximization against brute-force optimization

After showing that the greedy algorithm finds solutions characterized by negligible violations of the condition 6.7, we also want to see how close greedy solutions are to the ground-truth

optimal solution. To this end we apply brute-force optimization to find seed sets of small size in both random and real-world temporal networks. We then compare these solutions to those obtained using greedy optimization and random selection. Results from all our tests are summarized in Tables 6.1 and 6.2. Results for a few sample cases are reported in Figure 6.10. In general, we observe that the greedy algorithm is almost optimal. In particular, the performance of the greedy algorithm is far above the $1-1/e$ bound that would be valid if the influence was indeed a submodular function with non-negative marginal gains. At the same time, we see that random selection performs poorly, generating outbreak sizes well below the optimality bound.

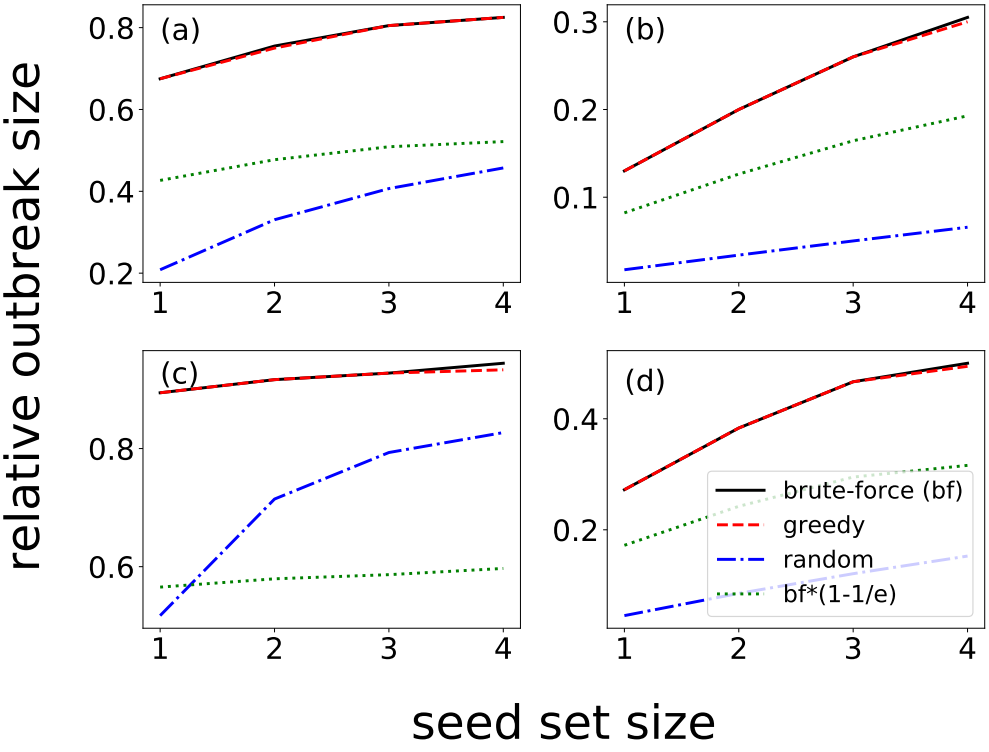

Figure 6.10: **Optimal selection of influential spreaders in temporal networks.** (a) We display the influence function of Equation 6.6 as a function of the size of the set of influential spreaders $\mathcal{X}$. Different methods for the identification of the influential spreaders are used, either brute-force search (black), greedy optimization (red) or random selection (blue). We also display the $1-1/e$ bound from the brute-force solution (green). Results are valid for a random temporal network composed of $N = 200$ nodes, $T = 10$ layers, and average degree $k = 1.5$. SIR parameters are $\lambda = \mu = 1.00$. (b) Same as in (a) but for $\lambda = 0.25$ and $\mu = 0.50$. (c) Same as in (a) but for the real-world temporal network "High school, 2012." (d) Same as in (c) but for $\lambda = 0.10$ and $\mu = 0.25$.

λ	μ	$\lvert\mathcal{X}\rvert = 2$	$\lvert\mathcal{X}\rvert = 3$	$\lvert\mathcal{X}\rvert = 4$	$\lvert\mathcal{X}\rvert = 5$
1.00	1.00	99.0%	98.8%	98.4%	98.2%
0.50	1.00	99.6%	99.3%	99.1%	98.7%
0.25	1.00	100.0%	99.3%	98.9%	98.5%
0.50	0.50	99.7%	99.6%	99.5%	99.4%
0.25	0.50	99.6%	99.6%	98.9%	98.8%
0.10	0.50	98.8%	98.3%	98.6%	98.5%
0.25	0.25	98.9%	99.3%	99.1%	99.1%
0.10	0.25	99.6%	99.1%	99.3%	99.3%
0.05	0.25	99.2%	99.0%	98.6%	98.7%

Table 6.1: **Greedy selection of optimal spreaders in real temporal networks.** We report the average outbreak size of the seeds found by greedy algorithm on various networks relative to the optimal solution found with brute-force optimization. The results are averaged over all real-world temporal networks listed in Table 5.1. We excluded "Email, dept. 1" and "High school, 2013" due to their size.

λ	μ	$\lvert\mathcal{X}\rvert = 2$	$\lvert\mathcal{X}\rvert = 3$	$\lvert\mathcal{X}\rvert = 4$
1.00	1.00	99.2%	98.4%	98.1%
0.50	1.00	97.9%	97.6%	97.1%
0.25	1.00	100.0%	100.0%	100.0%
0.50	0.50	99.6%	99.4%	98.7%
0.25	0.50	99.7%	99.3%	98.9%
0.10	0.50	100.0%	100.0%	100.0%
0.25	0.25	99.4%	99.4%	99.4%
0.10	0.25	99.7%	99.8%	99.8%
0.05	0.25	100.0%	100.0%	100.0%

Table 6.2: **Greedy selection of optimal spreaders in synthetic temporal networks.** We report the average outbreak size of the seeds found by greedy algorithm on random temporal networks relative to the optimal solution found with brute-force optimization. The results are averaged over random temporal networks created with all combinations of the parameters $N = 200, T \in \{5, 10\}$, and $k \in \{1.3, 1.4, 1.5, 1.6, 1.7, 2.0, 2.5\}$.

6.4 Conclusion

In spreading processes occurring on temporal networks, the influence function is not a submodular function, and yet greedy optimization provides performances far better than those of other heuristic methods (26). Here, we investigated the reasons behind such effectiveness

of the greedy strategy. We measured violations of the necessary conditions for the submodularity property. We observed that the premature infection and recovery of some nodes may have detrimental effects on the outbreak size, which in turn bring violations of the necessary conditions for the submodularity property. In our systematic analysis, we showed that violations occur frequently if seeds are selected at random. When seeds are selected according to the greedy optimization protocol, we observe that the frequency of violations is much lower in random temporal networks and non-existent in real-world temporal networks. This finding suggests that even though the function to be optimized does not satisfy the strict definition of a submodular function, in practice it behaves as an effectively submodular function under greedy optimization. As a matter of fact, solutions found by the greedy algorithm are as good as expected for optimization problems involving truly, mathematically speaking, submodular functions. To actually test this hypothesis, we compared greedy solutions to ground-truth solutions in small networks and for small sizes of the seed sets. We observed that in all considered cases, the performance of the greedy algorithm is very close to the optimal solution. On average, the greedy algorithm has a performance of, at worst, 97% in random temporal networks and, at worst, 98% in real-world temporal networks. This is much higher than the optimality guarantee of 63% for the greedy algorithm in static networks and also much better than the performance that can be achieved with randomly selected nodes. The fact that greedy optimization generates quasi-optimal solutions to the influence maximization problem on temporal networks is associated with its ability of avoiding the selection of seeds whose premature recovery would block future infection paths. We believe that this ability is not related to the specific protocol of optimization, rather to the fact that the optimization procedure relies on direct measurements of the influence function. We expect that any other optimization algorithm that uses dynamical information should be able to achieve similar performance by learning about the blocking effect of some nodes via measurements of the influence function. Also, the great effectiveness displayed by the greedy optimization strategy might indicate the effective existence of a tight lower bound on

the performance of the greedy algorithm in solving the influence maximization problem on temporal networks. We leave to future research the challenging task to formulate a theory for such a performance guarantee.

7 Conclusion

In this thesis, we study the influence maximization problem in complex networks. We first consider the traditional setting for the influence maximization problem, where the analysis is done on static networks. We also propose machine learning methods to improve the success in identifying influential spreaders. Furthermore, we analyze the influence maximization problem under non-traditional, but more realistic settings, such as in the presence of missing or imperfect information, and in temporal networks where interactions in networks are temporary. Finally, we analyze the characteristics of the influence maximization problem in temporal networks, where the problem is harder to solve than in static networks. The ideas and methods used in the thesis are presented in Chapters 1 and 2, while the results of our analysis are presented in Chapters 3, 4, 5, and 6.

In Chapter 3, we systematically compare the performances of methods for identifying influential spreaders in static networks. We analyze the performance of heuristic methods that rely on network structure in identifying influential spreaders, and compare the results to greedy optimization, which is the state-of-the-art in influence maximization. We show that simple heuristics that are much faster than greedy optimization can achieve results that are comparable to the maximally achievable solution. This is especially true when spreading regime is either subcritical or critical. We further show that by combining individual methods and using machine learning, we can train models to learn from a subset of networks, and find influential spreaders in unobserved networks. This substantially increases the performance of heuristic methods without increasing the complexity of the algorithms. We observe all the aforementioned results on a large corpus of small/medium sized real-world networks, and also confirm them on a small set of large sized real-world networks, showing that the results are consistent across networks with different sizes.

We analyze the affects of noisy information in Chapter 4. The methods that are used to collect empirical data, which in turn are used to create networks, are prone to error. In

order to account for the possible existence of errors in data, we study the effectiveness of algorithms for identifying influential spreaders in the presence of noise. We consider two types of sources for noise. The first source is structural noise where either true edges are not recorded in the empirical data, or false edges have been recorded. The second source for errors is dynamical noise, which is the difference between the true spreading probability and our knowledge of it. Implementing the two types of noise and analyzing their effects both by themselves and in combination, we observe that in certain cases the existence of one type of noise can be compensated via artificially implementing the other type of noise. This simple solution does not require any methods such as link prediction to be used, and increases the efficiency of algorithms when identifying influential spreaders in static networks in the presence of structural and dynamical noise.

In Chapter 5, we study the influence maximization problem on temporal networks. In previous chapters, our analysis focused on static networks. In the real-world, many systems have time-varying characteristics. In many real-world networks, nodes can appear and disappear, and maybe even more importantly, interactions between nodes last only for a certain time frame. The complex nature of these interactions create dynamics that are fundamentally different than the dynamics in static networks. This suggests that the results we have observed for the influence maximization problem in static networks are not necessarily true for temporal networks. Thus, we analyze the influence maximization problem in temporal networks using greedy optimization and approximation methods. Even though the influence function is not submodular for the spreading model considered, we observe that greedy optimization consistently provides good solutions. Analyzing the results for approximation methods gives us clues about the importance of the availability of information in different scenarios. We observe that having knowledge of the initial phases of a temporal network is important in identifying influential spreaders, especially when infected nodes recover fast. Furthermore, we observe that aggregating the network to form a static network is detrimental to the performance of algorithms, as is not knowing the ordering of temporal interactions

between nodes.

Finally, in Chapter 6, we analyze the characteristics of the influence function in temporal networks in the context of influence maximization. In the previous chapter, we have shown that the influence function is not submodular, thus there is no guarantee on the performance of greedy optimization. However, greedy optimization still provided good solutions. In order to dive further into the characteristics of the influence function, we analyze how frequently the submodularity and marginal gain conditions are violated. We consider selecting the nodes randomly or using greedy optimization, and test the violation of submodularity and marginal gain conditions using the selected nodes on synthetic and real-world temporal networks. In the case of randomly selected nodes, we observe that violations are common in both synthetic and real-world temporal networks, even though the frequency is relatively lower for the latter. However, when we use the nodes that are selected with greedy optimization, we observe that violations only appear rarely in synthetic networks, and no violations occur in real-world networks. This suggest that even though the influence function is not submodular, it becomes effectively submodular when the nodes are selected with greedy optimization. To further corroborate the effectiveness of greedy optimization, we compare its solutions to the optimal solutions found with brute-force search. The results confirm that greedy optimization is an effective methods to identify influential spreaders in temporal networks even without the same theoretical guarantee of performance it has on static networks.

The work presented in this thesis is not without limitations. Our spreading model choices mean our investigations are on simple contagion. However, spreading dynamics, such as opinion dynamics, can also follow complex contagion, which have different dynamics than simple contagion. In order to see if the results in this thesis on simple contagion extrapolate to complex contagion, one needs to repeat similar analysis. Another possible research area is early adopters. In general, the influence maximization problem is approached assuming that all nodes are inactive/susceptible. However, that is not always the case in real-life scenarios. For example, one might want to increase the reach of a product via marketing even when

the product and its adopters already exist in the market. For such cases, the affects of early adopters on the solutions of influence maximization problem should be investigated. Furthermore, our analysis on temporal networks rely on a simplification of temporal dynamics of networks, which is a potential limitation on understanding the true dynamics of spreading on temporal networks, in our case specifically in the context of the influence maximization problem. Further research can be done for this problem by modeling the temporal dynamics continuously. One final limitation in the thesis, which is also true for most of the literature on influence maximization, is the assumption that the information on the true and complete network structure is available to the decision maker. This is a naive assumption at best, both because methods to collect empirical network data are prone to error, and more importantly data collection is an expensive endeavor. More research is necessary to design methods for identifying influential spreaders on static and temporal networks when the network structure is only partially known or completely unknown by using tools from network science, such as the friendship paradox.

A Appendix: Systematic comparison between methods for the detection of influential spreaders in complex networks

More in depth analysis is available in the supplementary material of Ref. (24).

A.1 Real-world networks

Network	Type	N	E	p_c	Ref.	url
Political books	information	105	441	0.100	(117)	url
College football	social	115	613	0.134	(142)	url
S208	technological	122	189	0.466	(118)	url
High school, 2011	social	126	1709	0.038	(28)	url
Bay Dry	biological	128	2106	0.030	(101, 143)	url
Bay Wet	biological	128	2075	0.031	(101)	url
Radoslaw Email	social	167	3250	0.020	(101, 144)	url
High school, 2012	social	180	2220	0.044	(28)	url
Little Rock Lake	biological	183	2434	0.030	(101, 145)	url
Jazz	social	198	2742	0.031	(146)	url
S420	technological	252	399	0.451	(118)	url
C. Elegans, neural	biological	297	2148	0.045	(2)	url
Network Science	social	379	914	0.398	(147)	url
Dublin	social	410	2765	0.078	(101, 133)	url
US Air Trasportation	transportation	500	2980	0.026	(113)	url
S838	technological	512	819	0.349	(118)	url
Yeast, transcription	biological	662	1062	0.246	(148)	url
Caltech	social	762	16651	0.016	(149–151)	url
Reed	social	962	18812	0.015	(149–151)	url
Mouse retina	biological	1076	90811	0.004	(152, 153)	url
URV email	social	1133	5451	0.056	(1)	url
Political blogs	information	1222	16714	0.015	(117)	url
Air traffic	transportation	1226	2408	0.163	(101)	url
Haverford	social	1446	59589	0.009	(149–151)	url
Simmons	social	1510	32984	0.016	(149–151)	url
Swarthmore	social	1657	61049	0.009	(149–151)	url

Table A.1: **Real-world static networks.** Information of the networks analyzed in Chapter 3. From left to right we report the name of the network, the type of the network, number of nodes in the giant component, number of edges in the giant component, percolation threshold of the network, references to studies where the network is presented and analyzed, and url where the network can be found.

Network	Type	N	E	p_c	Ref.	url
Petster, hamster	social	1788	12476	0.025	(101)	url
UC Irvine	social	1893	13835	0.023	(116)	url
Yeast, protein	biological	2224	6609	0.071	(119)	url
Amherst	social	2235	90954	0.008	(149–151)	url
Bowdoin	social	2250	84386	0.009	(149–151)	url
Hamilton	social	2312	96393	0.008	(149–151)	url
Adolescent health	social	2539	10455	0.117	(153, 154)	url
Trinity	social	2613	111996	0.008	(149–151)	url
USFCA	social	2672	65244	0.011	(149–151)	url
Japanese	information	2698	7995	0.030	(118)	url
Williams	social	2788	112985	0.008	(149–151)	url
Open flights	transportation	2905	15645	0.020	(101, 115)	url
Oberlin	social	2920	89912	0.010	(149–151)	url
Smith	social	2970	97133	0.010	(149–151)	url
Wellesley	social	2970	94899	0.010	(149–151)	url
Vassar	social	3068	119161	0.009	(149–151)	url
Middlebury	social	3069	124607	0.008	(149–151)	url
Pepperdine	social	3440	152003	0.007	(149–151)	url
Colgate	social	3482	155043	0.008	(149–151)	url
Santa	social	3578	151747	0.007	(149–151)	url
Wesleyan	social	3591	138034	0.009	(149–151)	url
Mich	social	3745	81901	0.011	(149–151)	url
Bitcoin Alpha	social	3775	14120	0.027	(155–157)	url
Bucknell	social	3824	158863	0.008	(149–151)	url
Brandeis	social	3887	137561	0.008	(149–151)	url
Howard	social	4047	204850	0.006	(149–151)	url
Rice	social	4083	184826	0.007	(149–151)	url
GR-QC, 1993-2003	social	4158	13422	0.091	(103, 157)	url
Tennis	social	4338	81865	0.007	(114)	None
Rochester	social	4561	161403	0.009	(149–151)	url
US Power grid	technological	4941	6594	0.437	(2)	url
Lehigh	social	5073	198346	0.008	(149–151)	url
Johns Hopkins	social	5157	186572	0.007	(149–151)	url
HT09	social	5352	18481	0.025	(133)	url
Wake	social	5366	279186	0.006	(149–151)	url
Hep-Th, 1995-1999	social	5835	13815	0.108	(158)	url
Bitcoin OTC	social	5875	21489	0.023	(155–157)	url
Reactome	biological	5973	145778	0.011	(101, 159)	url
Jung	technological	6120	50290	0.009	(101, 160)	url
Gnutella, Aug. 8, 2002	technological	6299	20776	0.046	(102, 103, 157)	url

Table A.2: **Real-world static networks.** Continuation of Table A.1

Network	Type	N	E	p_c	Ref.	url
American	social	6370	217654	0.008	(149–151)	url
MIT	social	6402	251230	0.006	(149–151)	url
JDK	technological	6434	53658	0.009	(101)	url
William	social	6472	266378	0.007	(149–151)	url
AS Oregon	technological	6474	12572	0.036	(157, 161)	url
UChicago	social	6561	208088	0.008	(149–151)	url
Princeton	social	6575	293307	0.007	(149–151)	url
Carnegie	social	6621	249959	0.007	(149–151)	url
Tufts	social	6672	249722	0.008	(149–151)	url
UC	social	6810	155320	0.010	(149–151)	url
Wikipedia elections	social	7066	100736	0.008	(157, 162, 163)	url
English	information	7377	44205	0.011	(118)	url
Gnutella, Aug. 9, 2002	technological	8104	26008	0.045	(102, 103, 157)	url
French	information	8308	23832	0.022	(118)	url
Hep-Th, 1993-2003	social	8638	24806	0.072	(103, 157)	url
Gnutella, Aug. 6, 2002	technological	8717	31525	0.065	(102, 103, 157)	url
Gnutella, Aug. 5, 2002	technological	8842	31837	0.056	(102, 103, 157)	url
PGP	social	10680	24316	0.064	(164)	url
Gnutella, Aug. 4, 2002	technological	10876	39994	0.076	(102, 103, 157)	url
Hep-Ph, 1993-2003	social	11204	117619	0.005	(103, 157)	url
Spanish 1	information	11558	43050	0.012	(118)	url
DBLP, citations	information	12495	49563	0.032	(101, 165)	url
Spanish 2	information	12643	55019	0.012	(101)	url
Cond-Mat, 1995-1999	social	13861	44619	0.064	(157, 158)	url
Astrophysics	social	14845	119652	0.018	(158)	url
AstroPhys, 1993-2003	social	17903	196972	0.013	(103, 157)	url
Marvel	social	19365	96616	0.019	(153, 166)	url
Cond-Mat, 1993-2003	social	21363	91286	0.037	(103, 157)	url
Gnutella, Aug. 25, 2002	technological	22663	54693	0.115	(102, 103, 157)	url
Internet	technological	22963	48436	0.019	None	url
Thesaurus	information	23132	297094	0.011	(101, 167)	url
Cora	information	23166	89157	0.045	(101, 168)	url
AS Caida	technological	26475	53381	0.021	(157, 161)	url
Gnutella, Aug. 24, 2002	technological	26498	65359	0.106	(102, 103, 157)	url

Table A.3: **Real-world static networks.** Continuation of Table A.2

A.2 Results of analysis

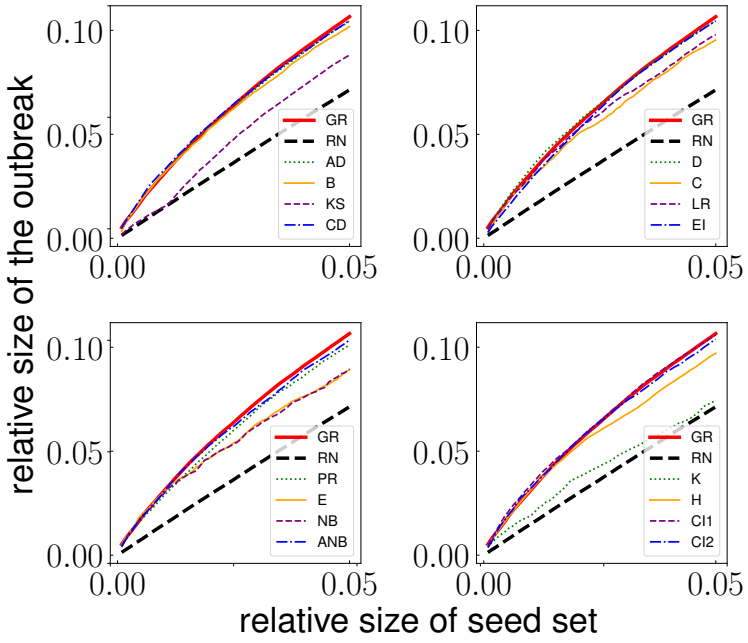

Figure A.1: **Outbreak size as a function of seed set size.** Relative size of the outbreak as a function of the relative size of the seed set for an email communication network (1). The outbreak sizes are calculated with ICM dynamics for $\lambda = 0.5\lambda_c$.

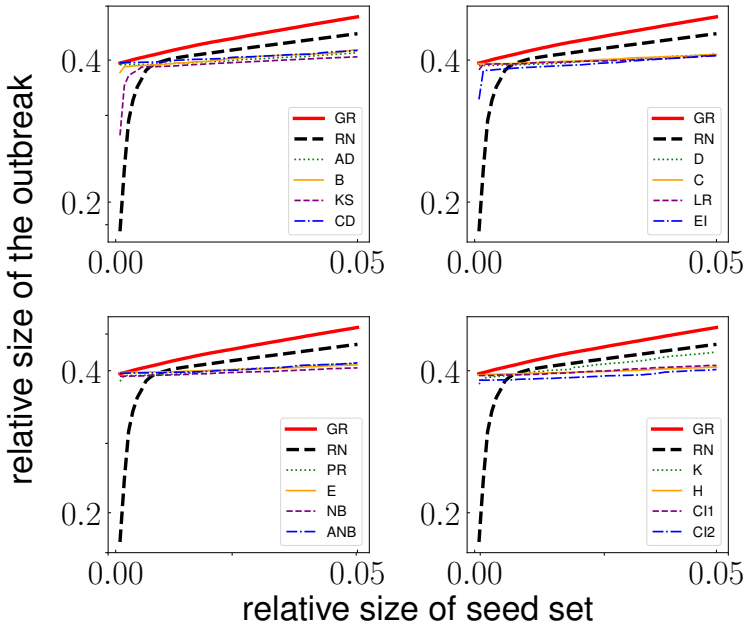

Figure A.2: **Outbreak size as a function of seed set size.** Relative size of the outbreak as a function of the relative size of the seed set for an email communication network (1). The outbreak sizes are calculated with ICM dynamics for $\lambda = 2\lambda_c$.

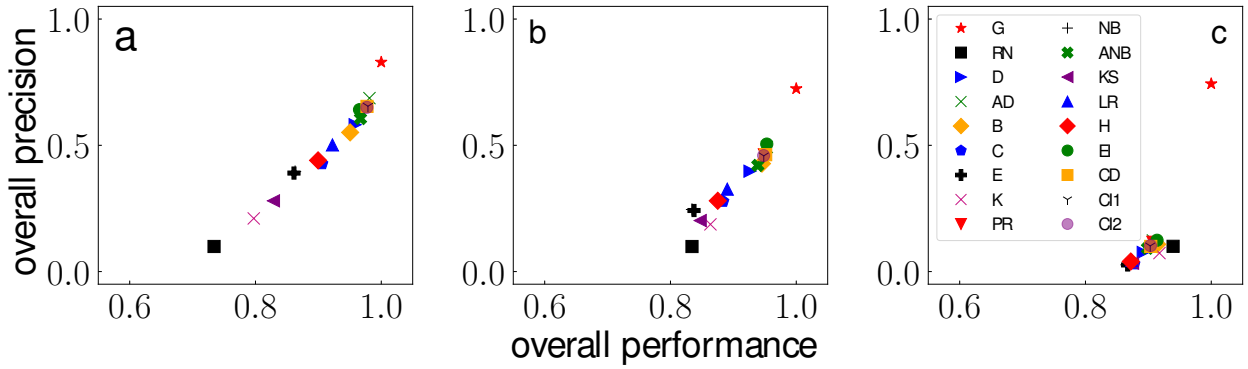

Figure A.3: **Overall performance and overall precision of methods for the identification of influential spreaders in real networks.** Results are based on the systematic analysis of the corpus of 100 real-world networks. We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = 2\lambda_c$, (c) $\lambda = 2\lambda_c$. Each point in a panel corresponds to a single method. Every method is used to identify top TN nodes as spreaders with $T = 0.10$. Methods are characterized by the metrics of performance defined. Both metrics relate the performance of a method m to the performance of greedy algorithm. Overall performance $\langle g_m \rangle$ shows the outbreak size of a method m relative to the outbreak size from greedy algorithm. Overall precision $\langle p_m \rangle$ quantifies the overlap between the seed sets identified by a method m and greedy algorithm.

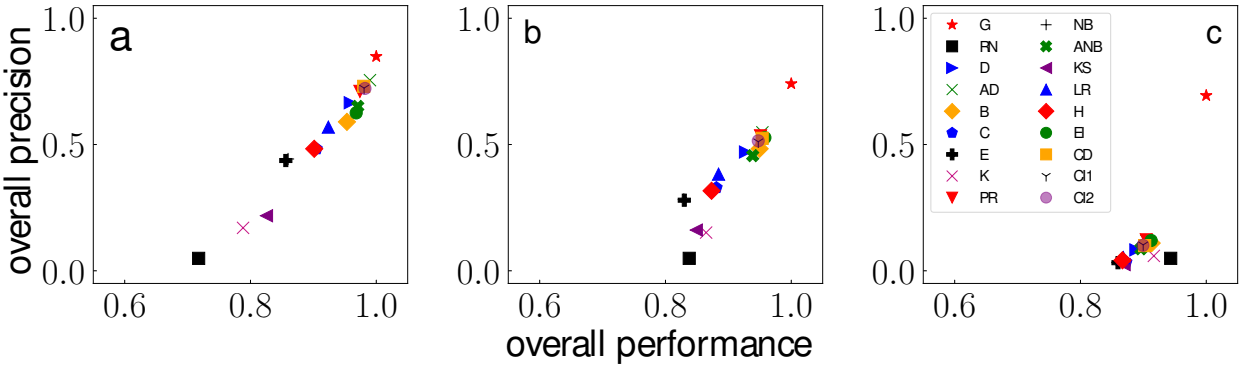

Figure A.4: **Overall performance and overall precision of methods for the identification of influential spreaders in real networks**. We use $V_m^{(T)}$ as the main measure of performance. Results are based on the systematic analysis of the corpus of 100 real-world networks. We consider the analysis for three distinct regimes of spreading: (a) $\lambda = 0.5\lambda_c$, (b) $\lambda = 2\lambda_c$, (c) $\lambda = 2\lambda_c$. Each point in a panel corresponds to a single method. Every method is used to identify top TN nodes as spreaders with $T = 0.05$. Methods are characterized by the metrics of performance defined. Both metrics relate the performance of a method m to the performance of greedy algorithm. Overall performance $\langle g_m \rangle$ shows the outbreak size of a method m relative to the outbreak size from greedy algorithm. Overall precision $\langle p_m \rangle$ quantifies the overlap between the seed sets identified by a method m and greedy algorithm.

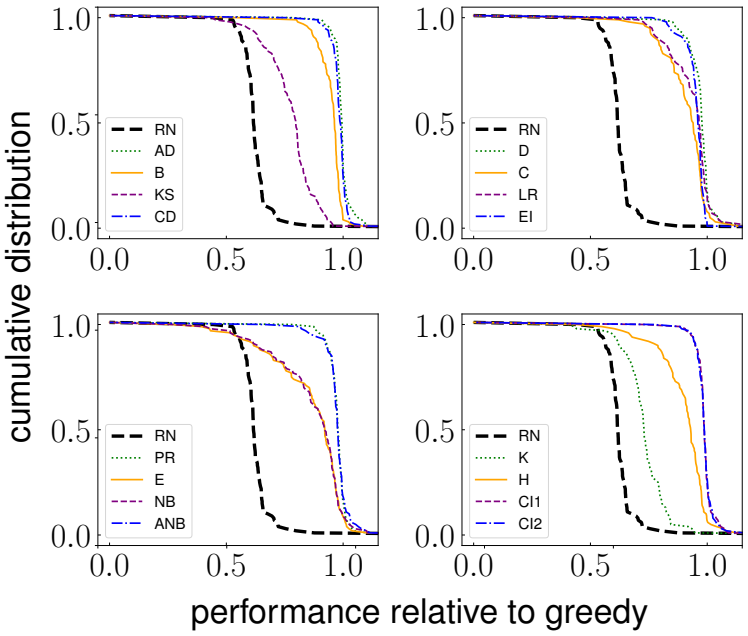

Figure A.5: **Cumulative distribution of the relative performance.** Cumulative distribution of the relative performance $g_m^{(T)}$ for $T = 0.05$. The metric of relative performance is defined in Equation 3.3. The distribution considers all 100 networks in the corpus. The outbreak size is calculated for ICM dynamics at $\lambda = 0.5\lambda_c$.

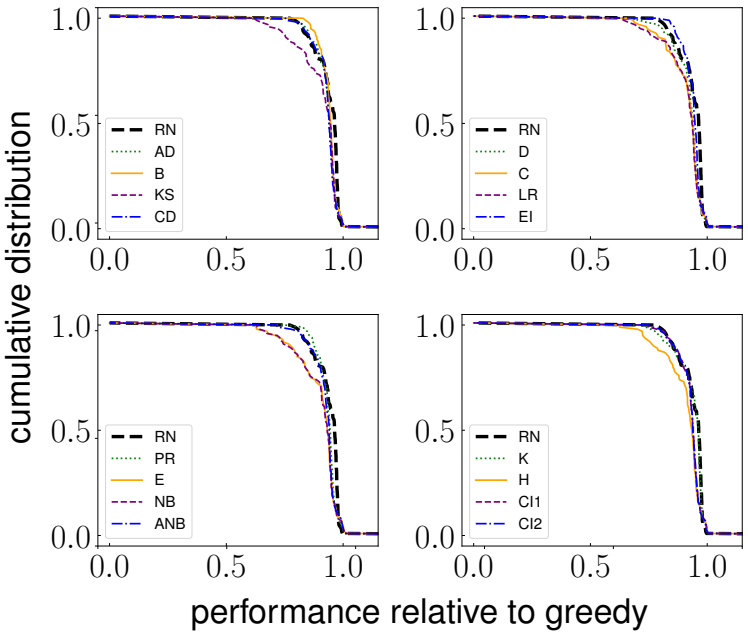

Figure A.6: **Cumulative distribution of the relative performance.** Cumulative distribution of the relative performance $g_m^{(T)}$ for $T = 0.05$. The metric of relative performance is defined in Equation 3.3. The distribution considers all 100 networks in the corpus. The outbreak size is calculated for ICM dynamics at $\lambda = 2\lambda_c$.

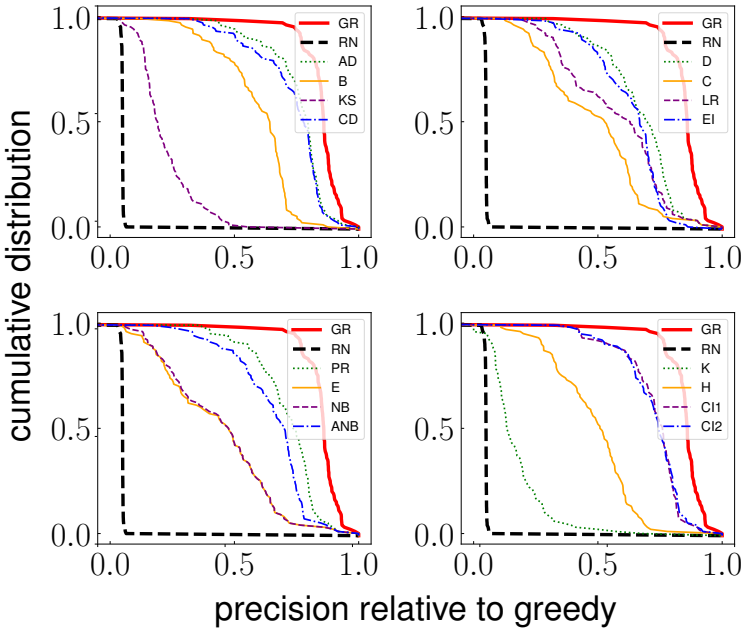

Figure A.7: **Cumulative distribution of precision.** Cumulative distribution of the precision metric $p_m^{(T)}$ for $T = 0.05$ as defined in Equation 3.4. The distribution covers all 100 networks in the corpus. Results for greedy algorithm are obtained for ICM dynamics at $\lambda = 0.5\lambda_c$.

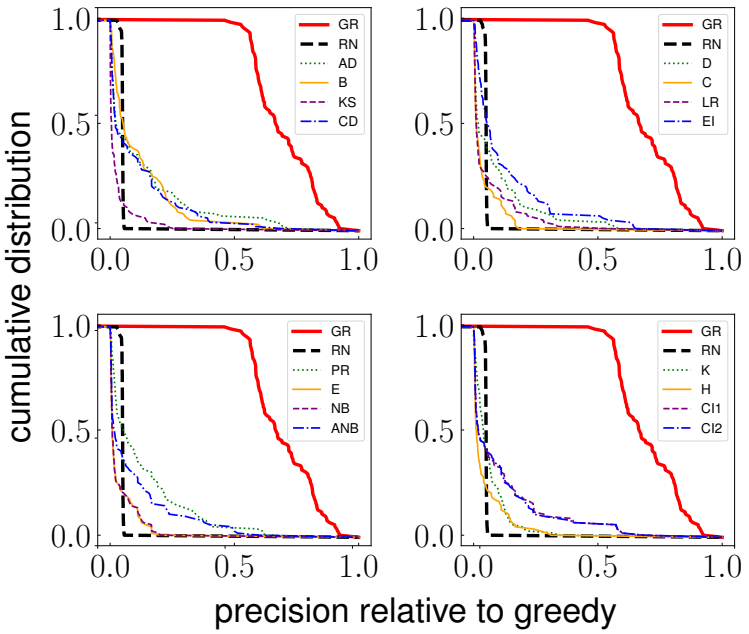

Figure A.8: **Cumulative distribution of precision.** Cumulative distribution of the precision metric $p_m^{(T)}$ for $T = 0.05$ as defined in Equation 3.4. The distribution covers all 100 networks in the corpus. Results for greedy algorithm are obtained for ICM dynamics at $\lambda = 2\lambda_c$.

B Appendix: Influence maximization in noisy networks

B.1 Results of analysis

The results obtained for the analysis of real-world network listed in Table 4.1 are shown in Figures B.1-B.27. Two sets of figures are reported for each network, one for the setting $|\mathcal{X}_{err}| = 100$ and the other for $|\mathcal{X}_{err}| = 10$.

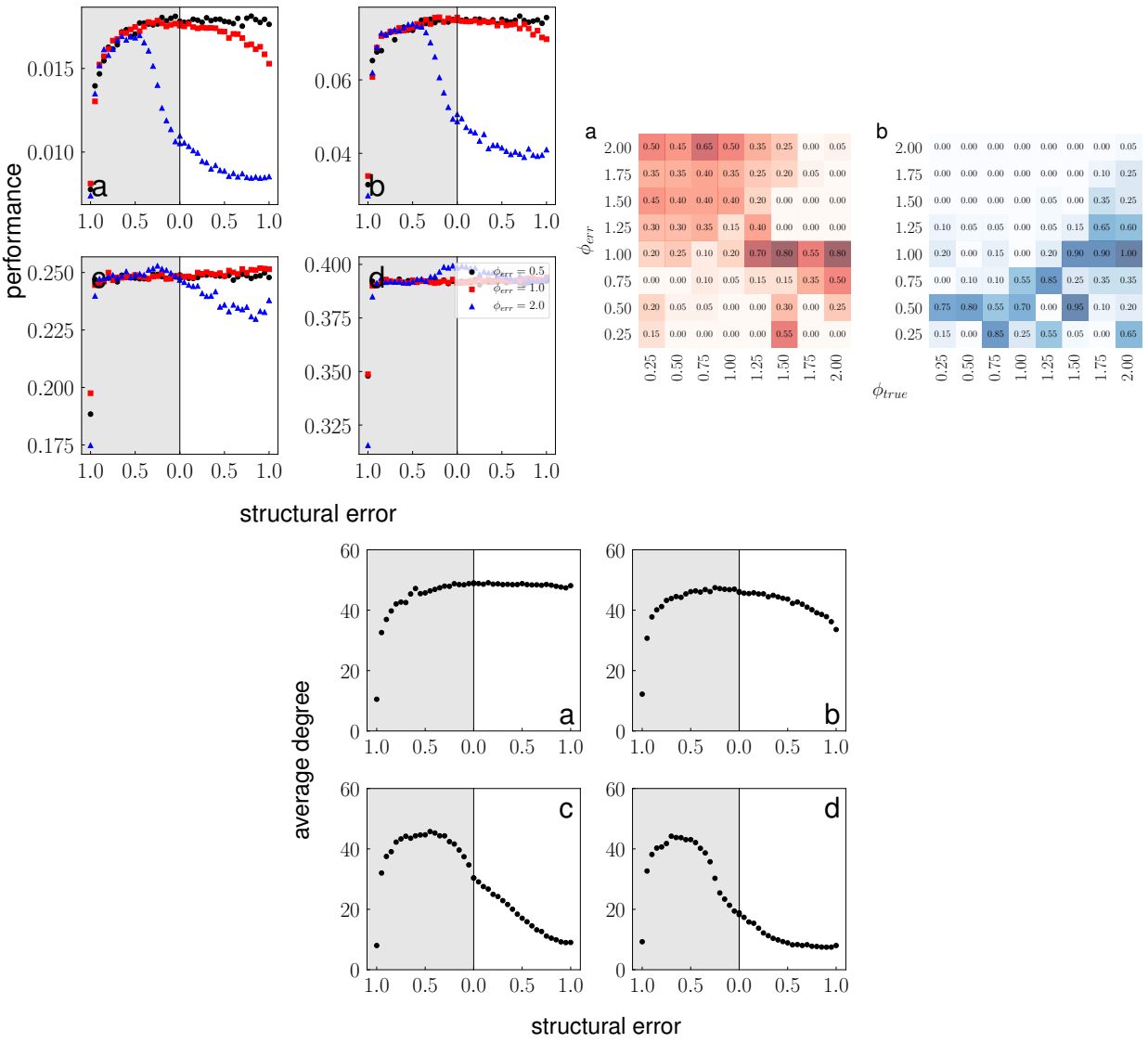

Figure B.1: **URV email.** $|\mathcal{X}_{err}| = 10$

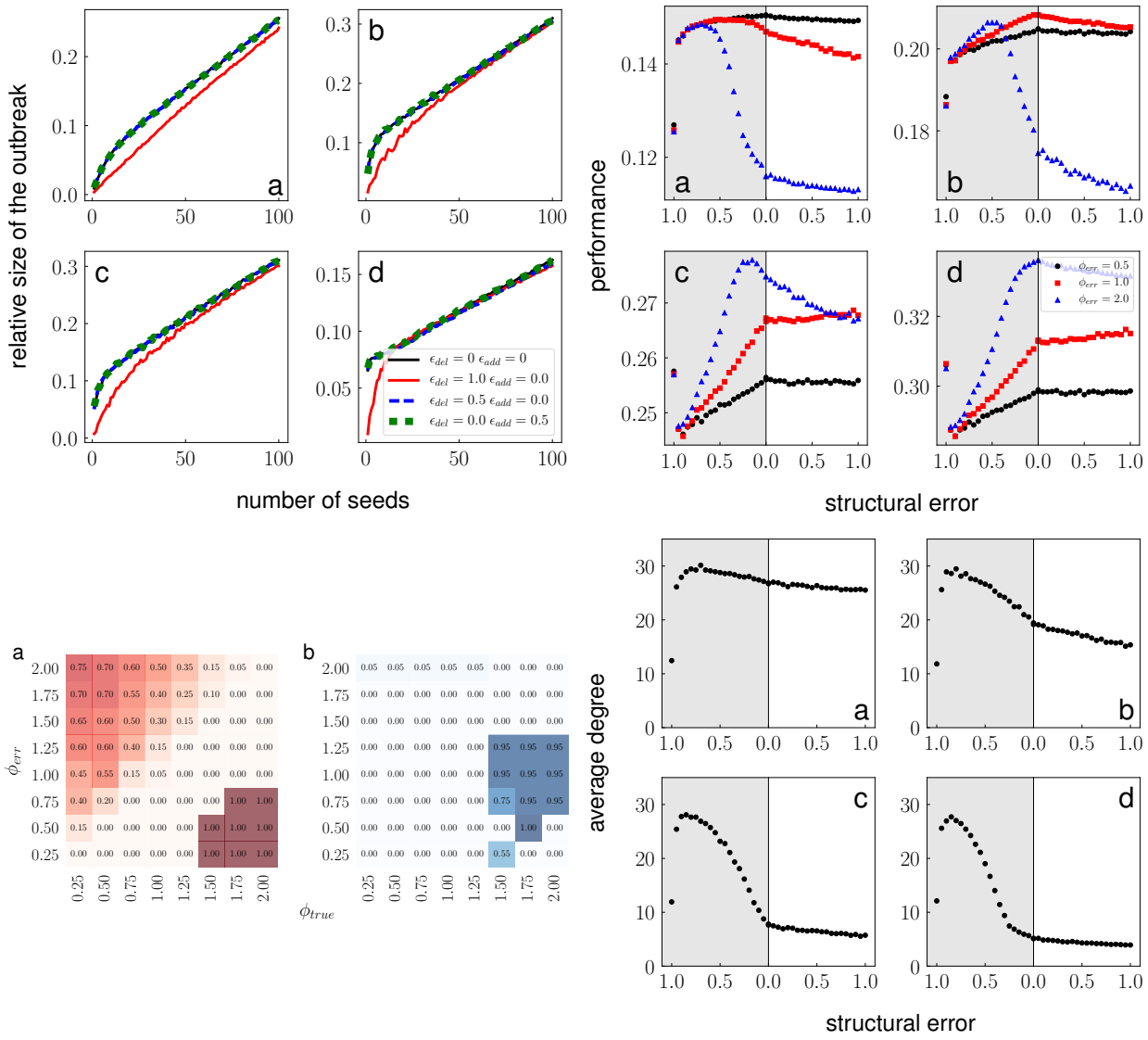

Figure B.2: **US Air Transportation.** $|\mathcal{X}_{err}| = 100$

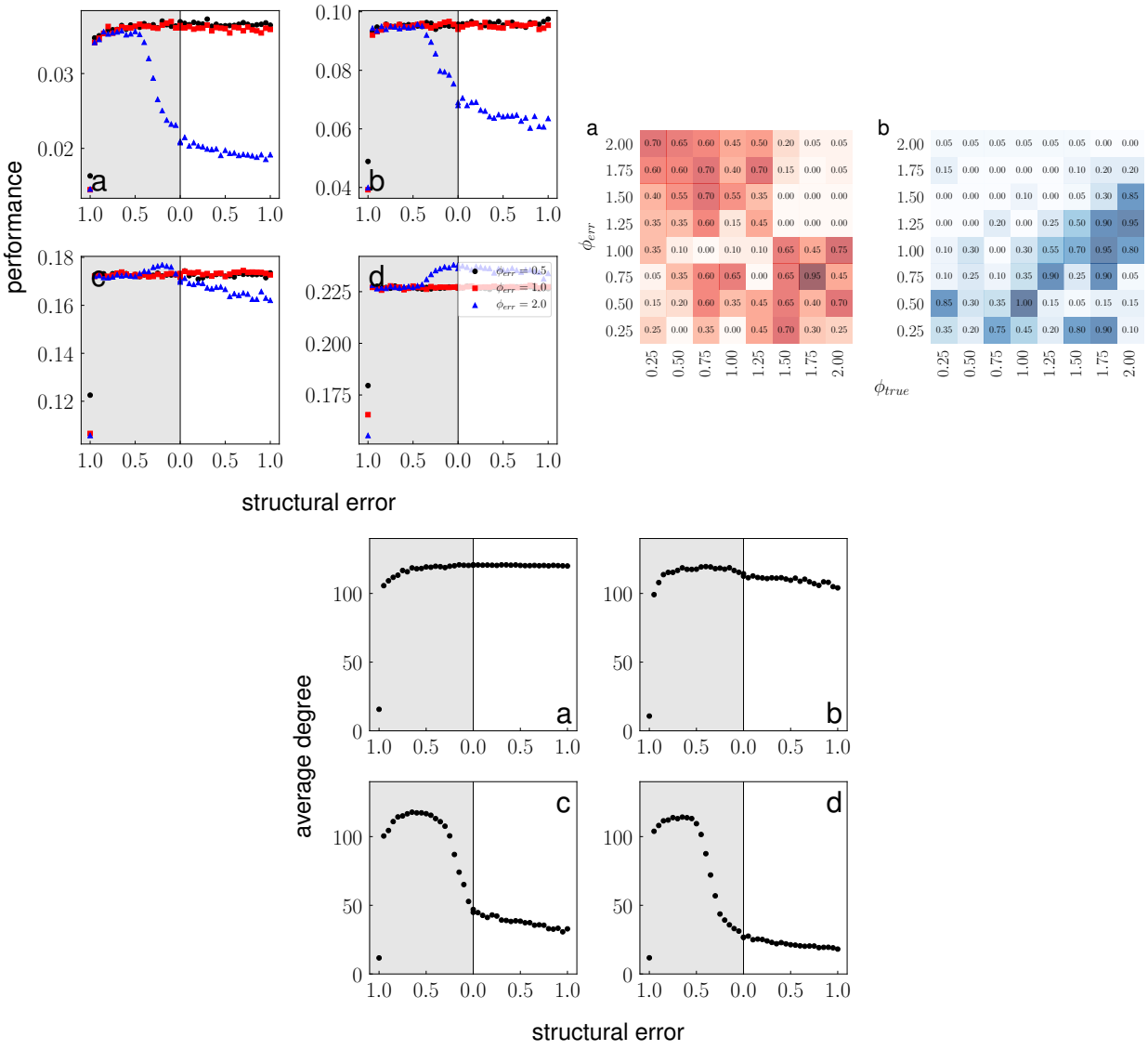

Figure B.3: **US Air Transportation.** $|\mathcal{X}_{err}| = 10$

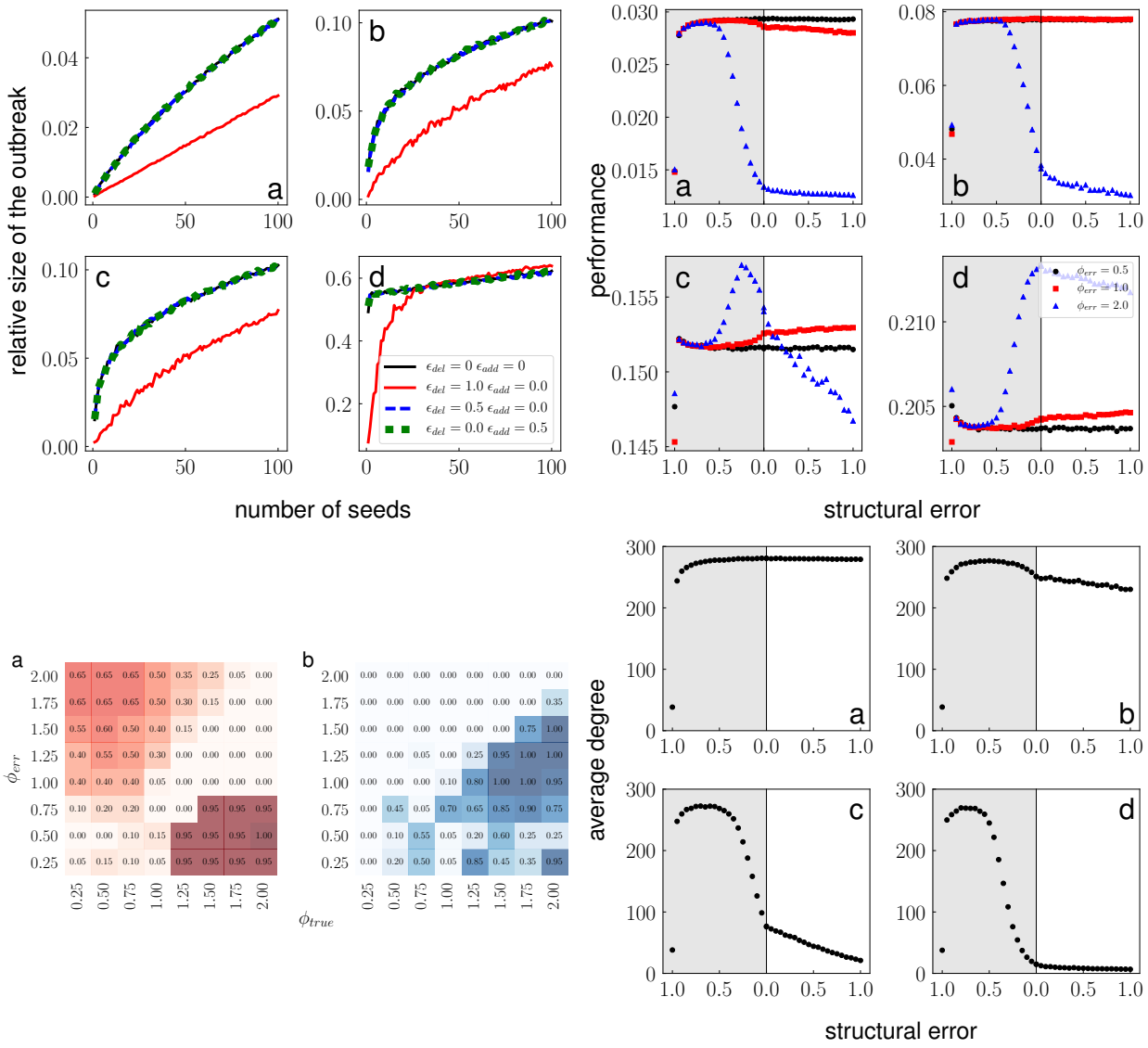

Figure B.4: **Tennis.** $|\mathcal{X}_{err}| = 100$

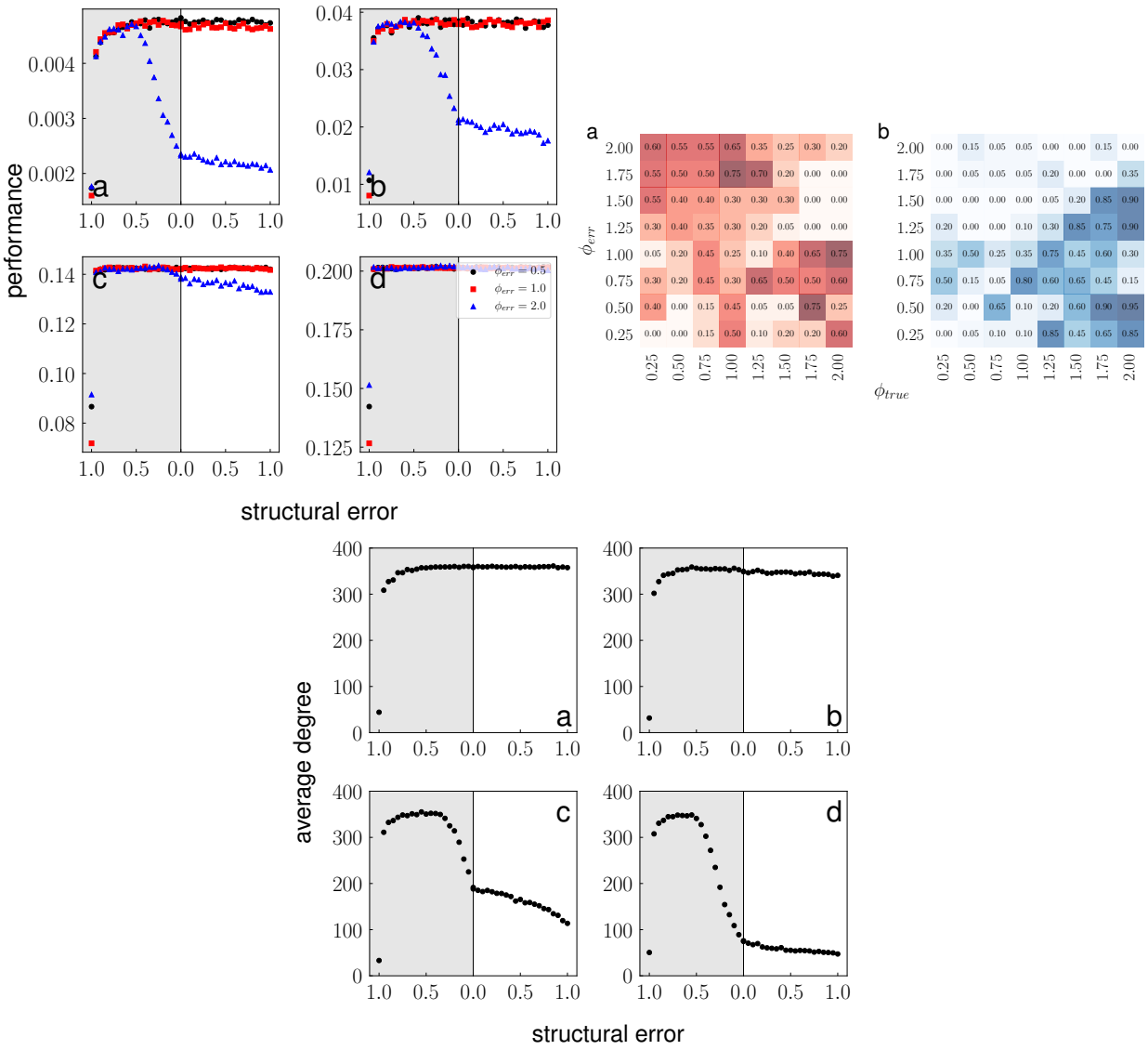

Figure B.5: **Tennis.** $|\mathcal{X}_{err}| = 10$

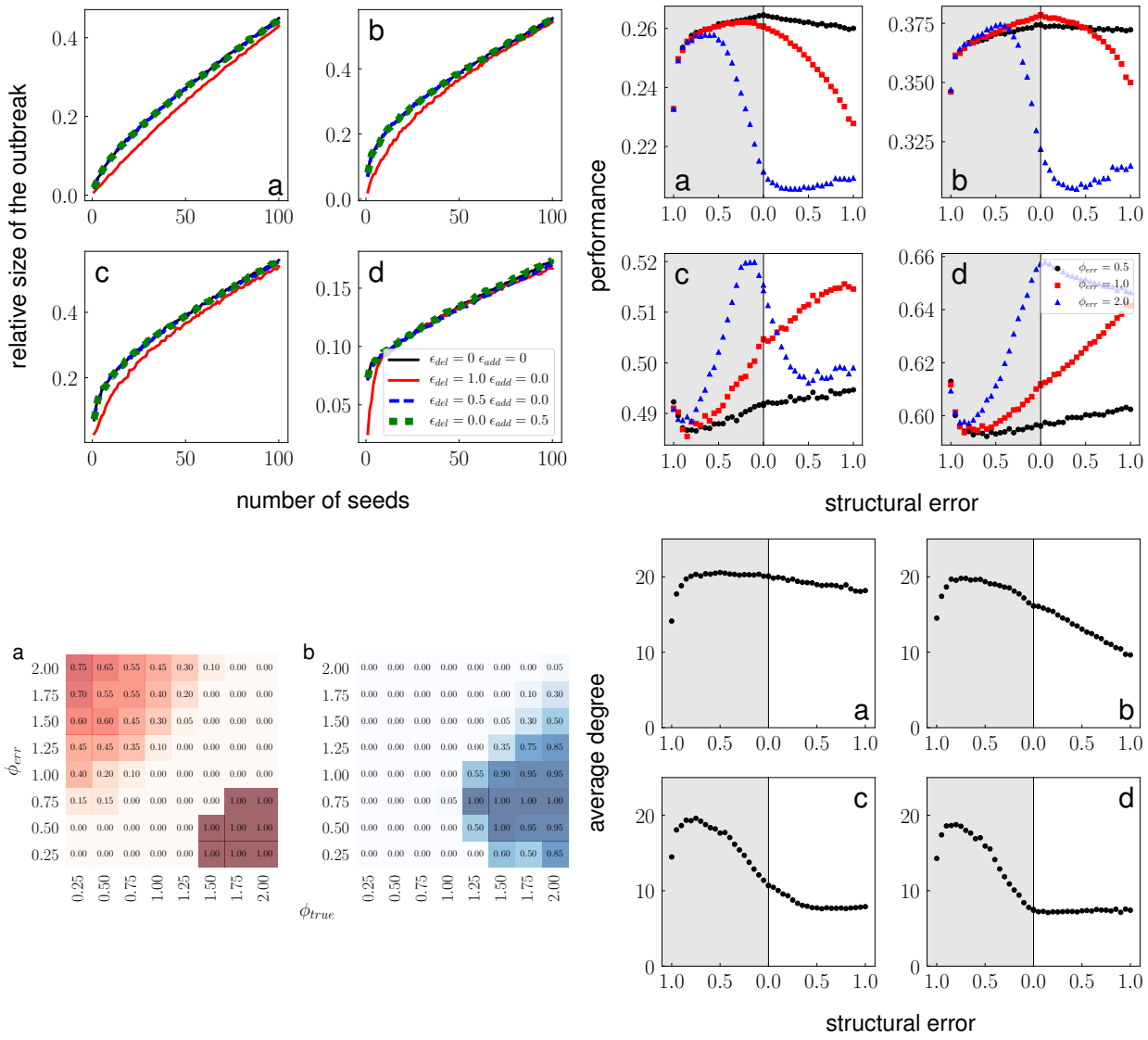

Figure B.6: **C. elegans, neural.** $|\mathcal{X}_{err}| = 100$

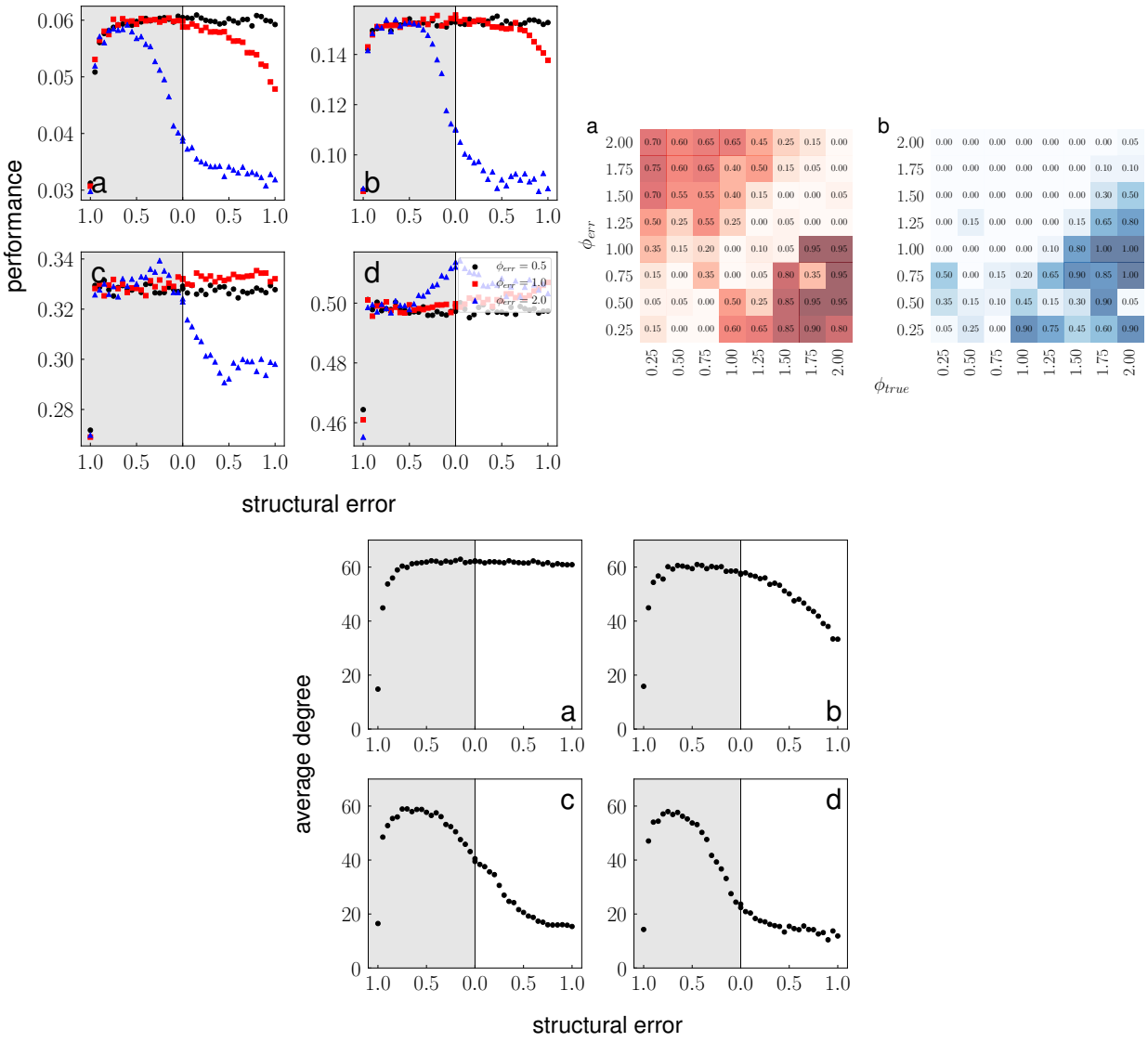

Figure B.7: **C. elegans, neural.** $|\mathcal{X}_{err}| = 10$

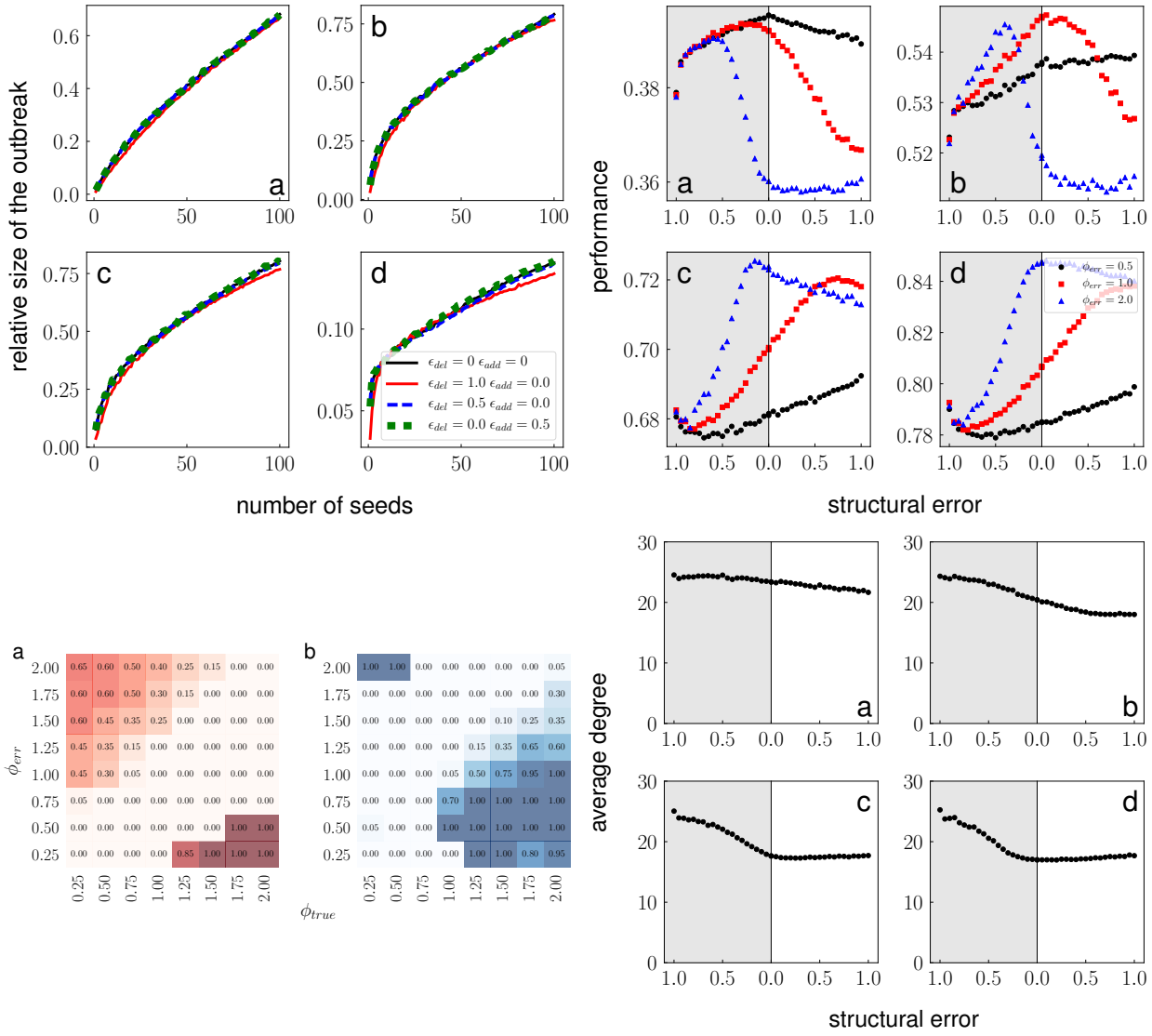

Figure B.8: **High school, 2012.** $|\mathcal{X}_{err}| = 100$

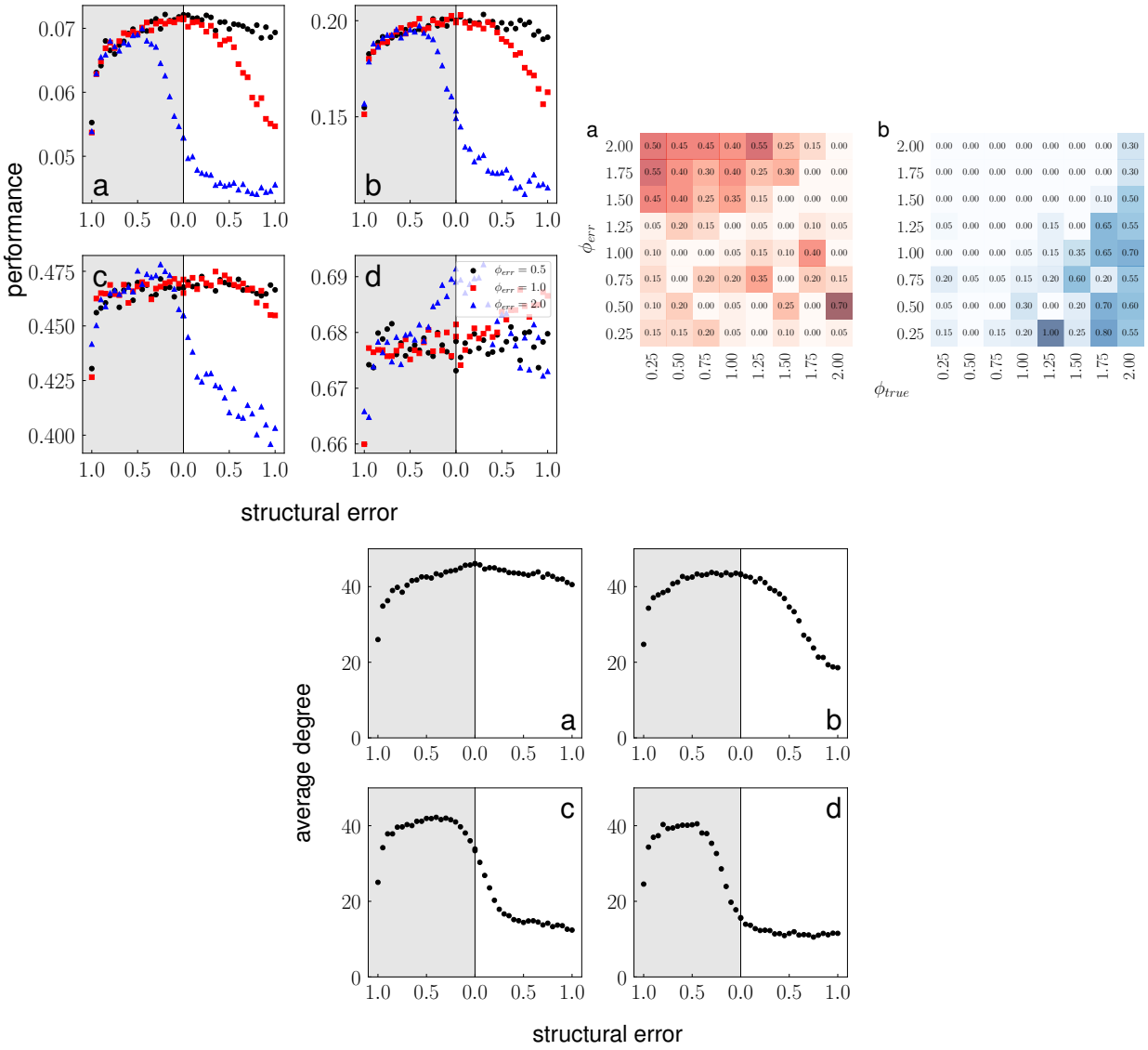

Figure B.9: **High school, 2012.** $|\mathcal{X}_{err}| = 10$

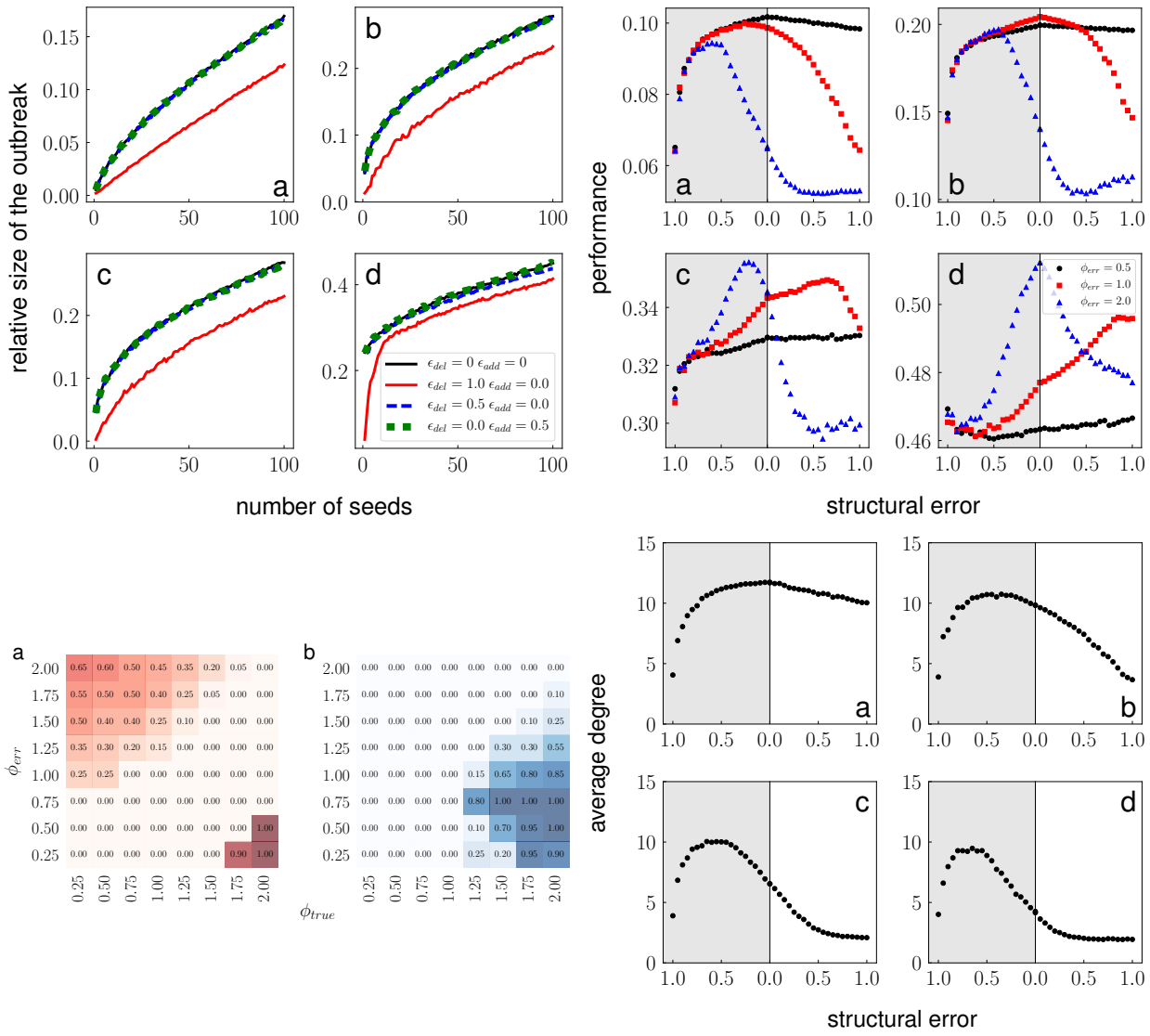

Figure B.10: **Air traffic.** $|\mathcal{X}_{err}| = 100$

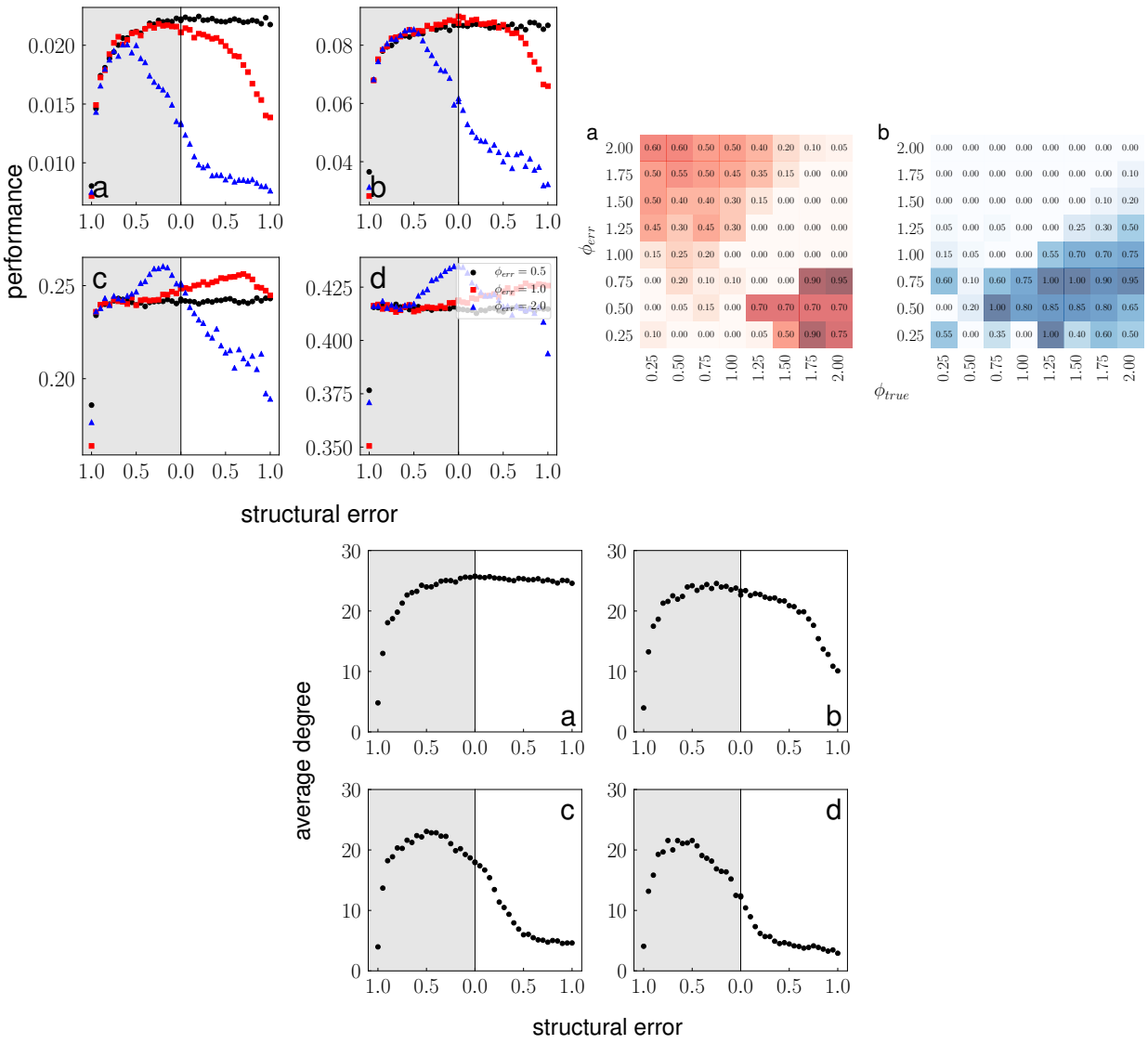

Figure B.11: **Air traffic.** $|\mathcal{X}_{err}| = 10$

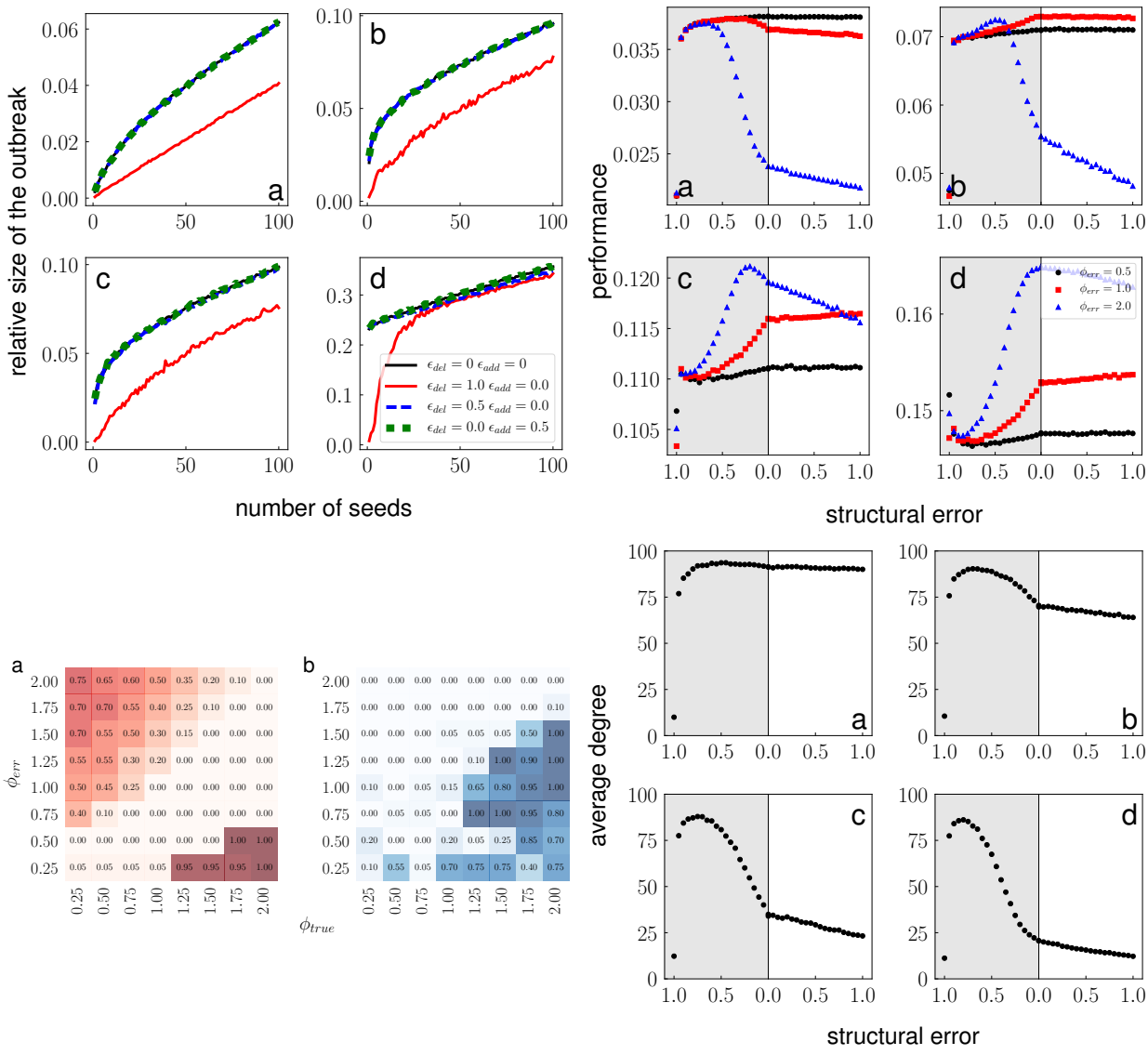

Figure B.12: **Open flights.** $|\mathcal{X}_{err}| = 100$

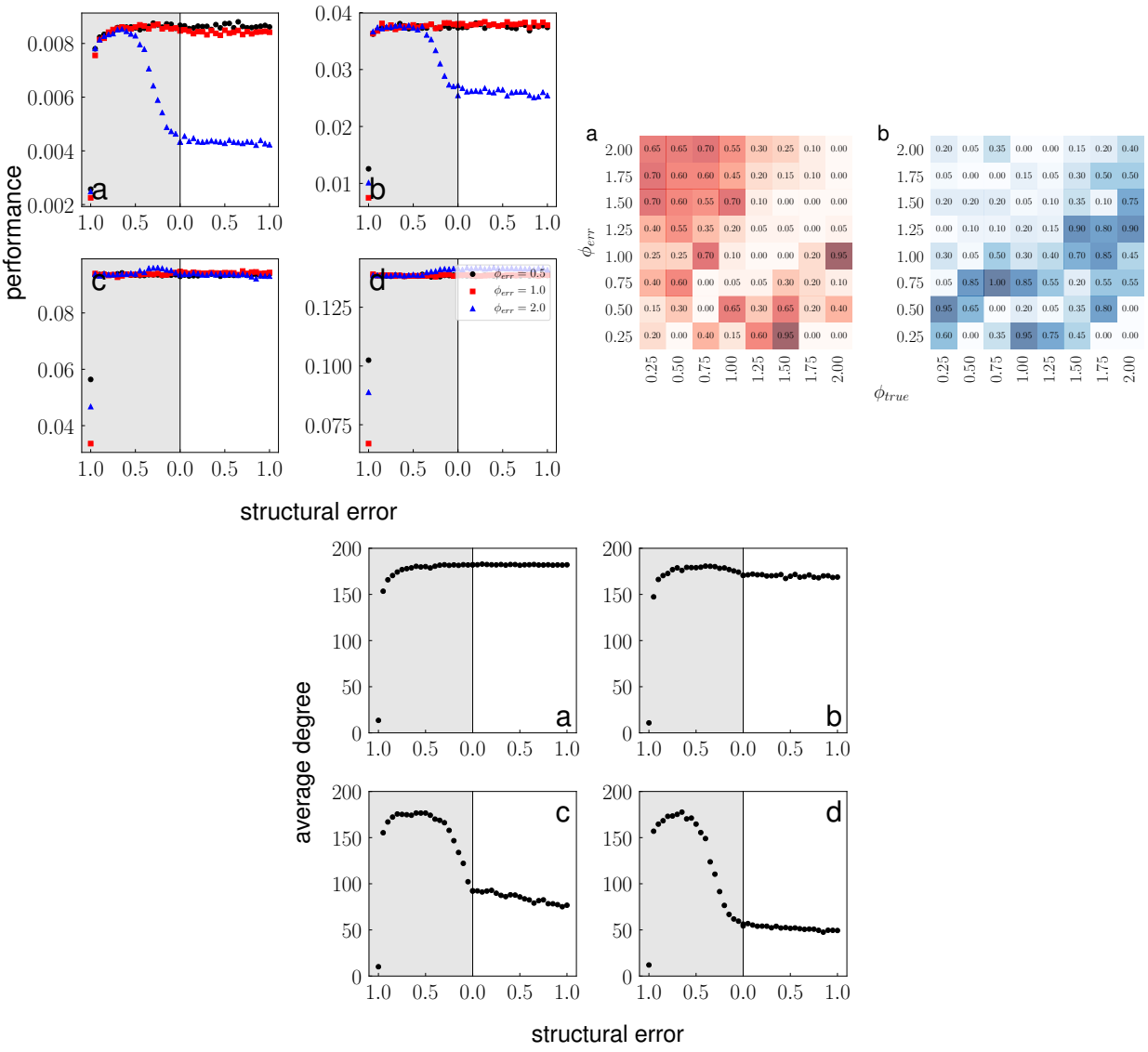

Figure B.13: **Open flights.** $|\mathcal{X}_{err}| = 10$

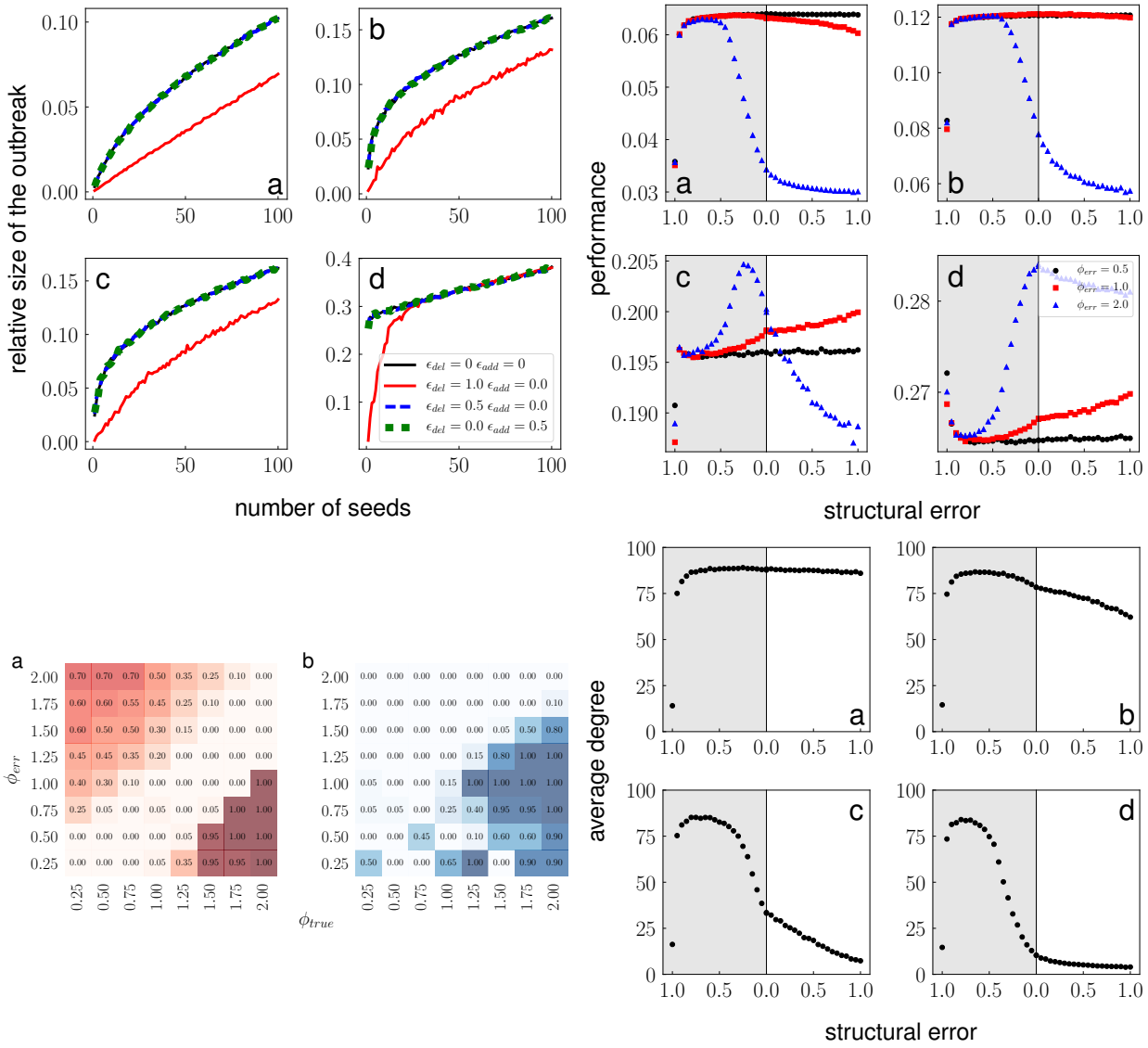

Figure B.14: **UC Irvine.** $|\mathcal{X}_{err}| = 100$

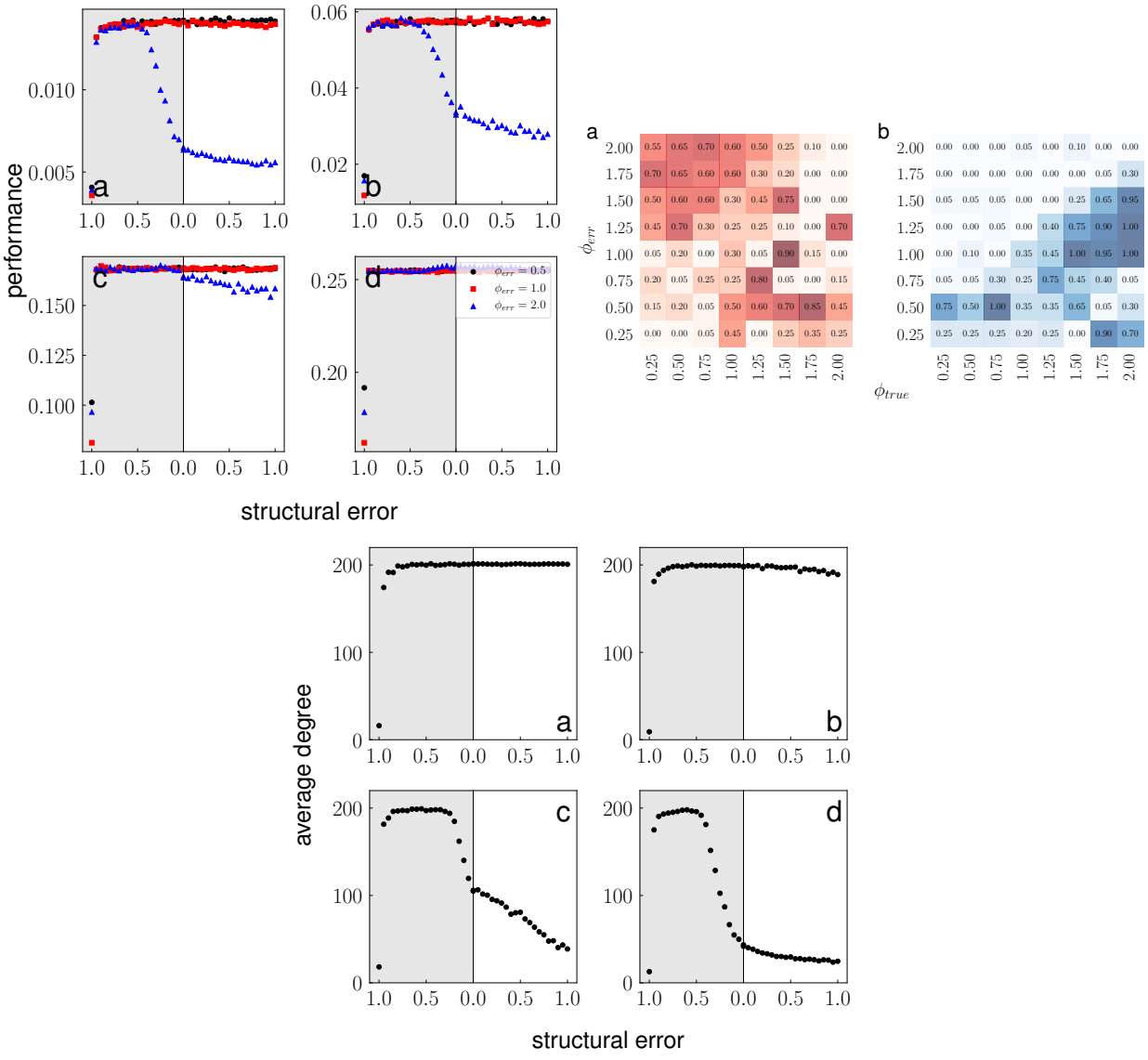

Figure B.15: **UC Irvine.** $|\mathcal{X}_{err}| = 10$

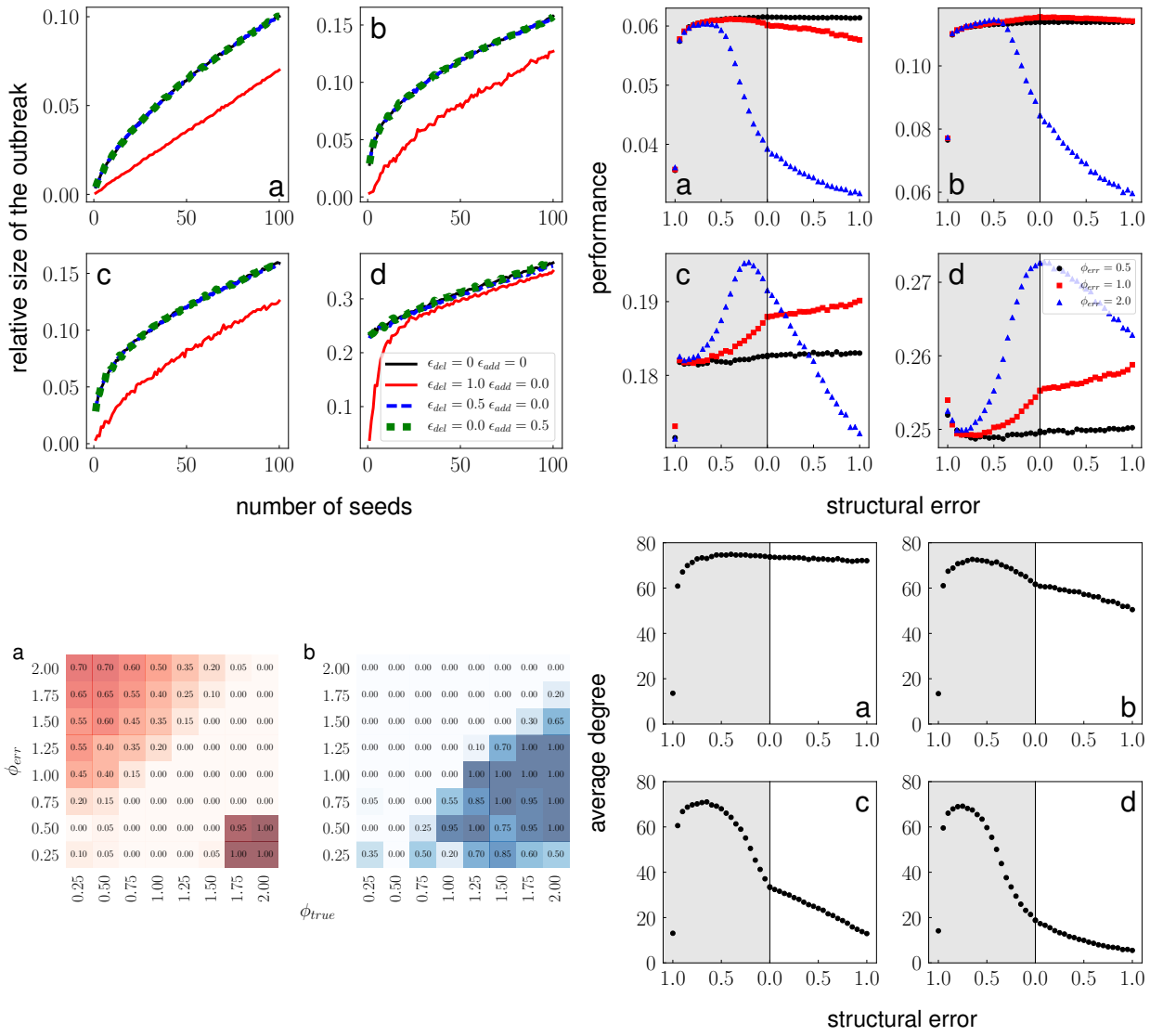

Figure B.16: **Petster, hamster.** $|\mathcal{X}_{err}| = 100$

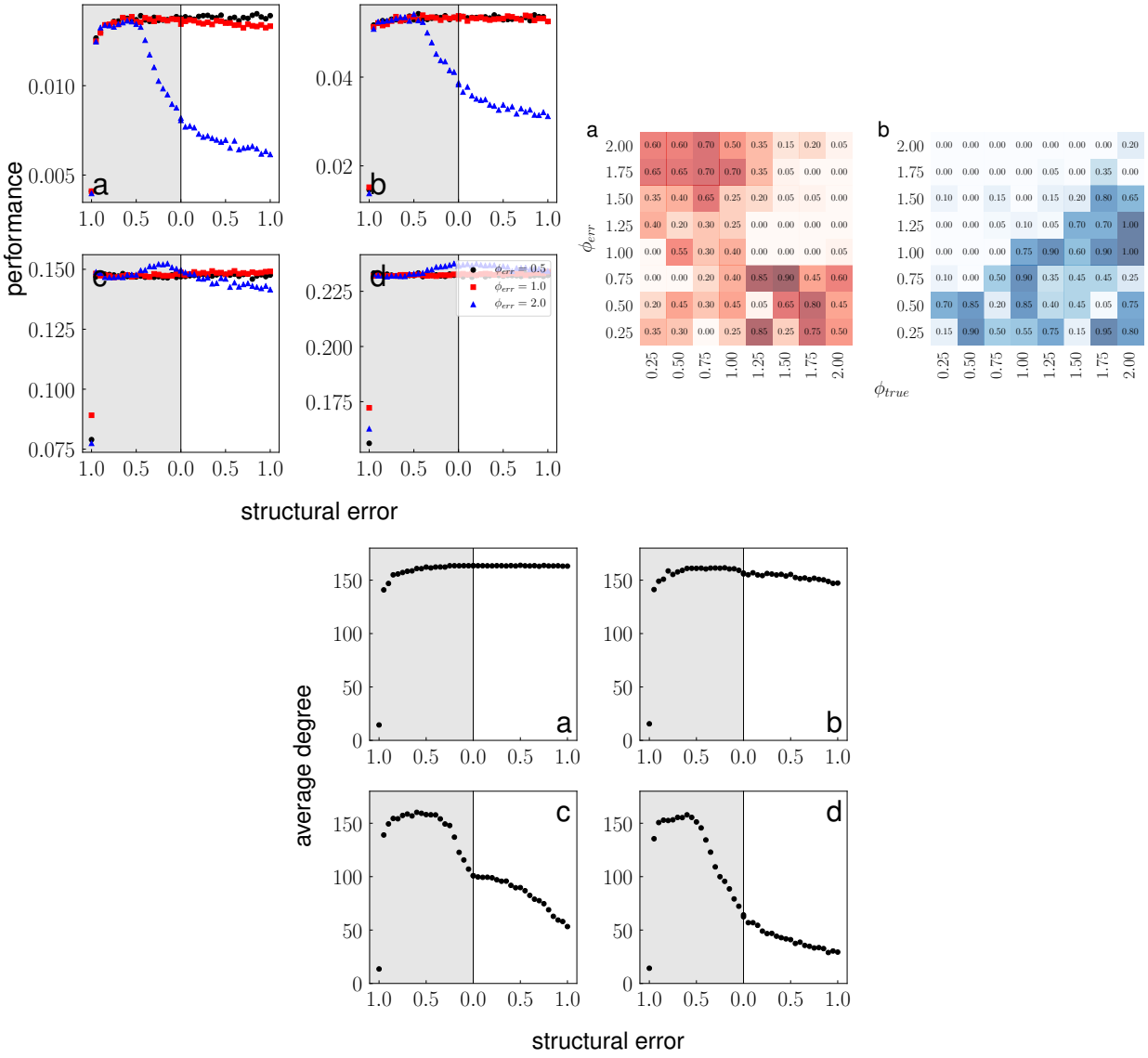

Figure B.17: **Petster, hamster.** $|\mathcal{X}_{err}| = 10$

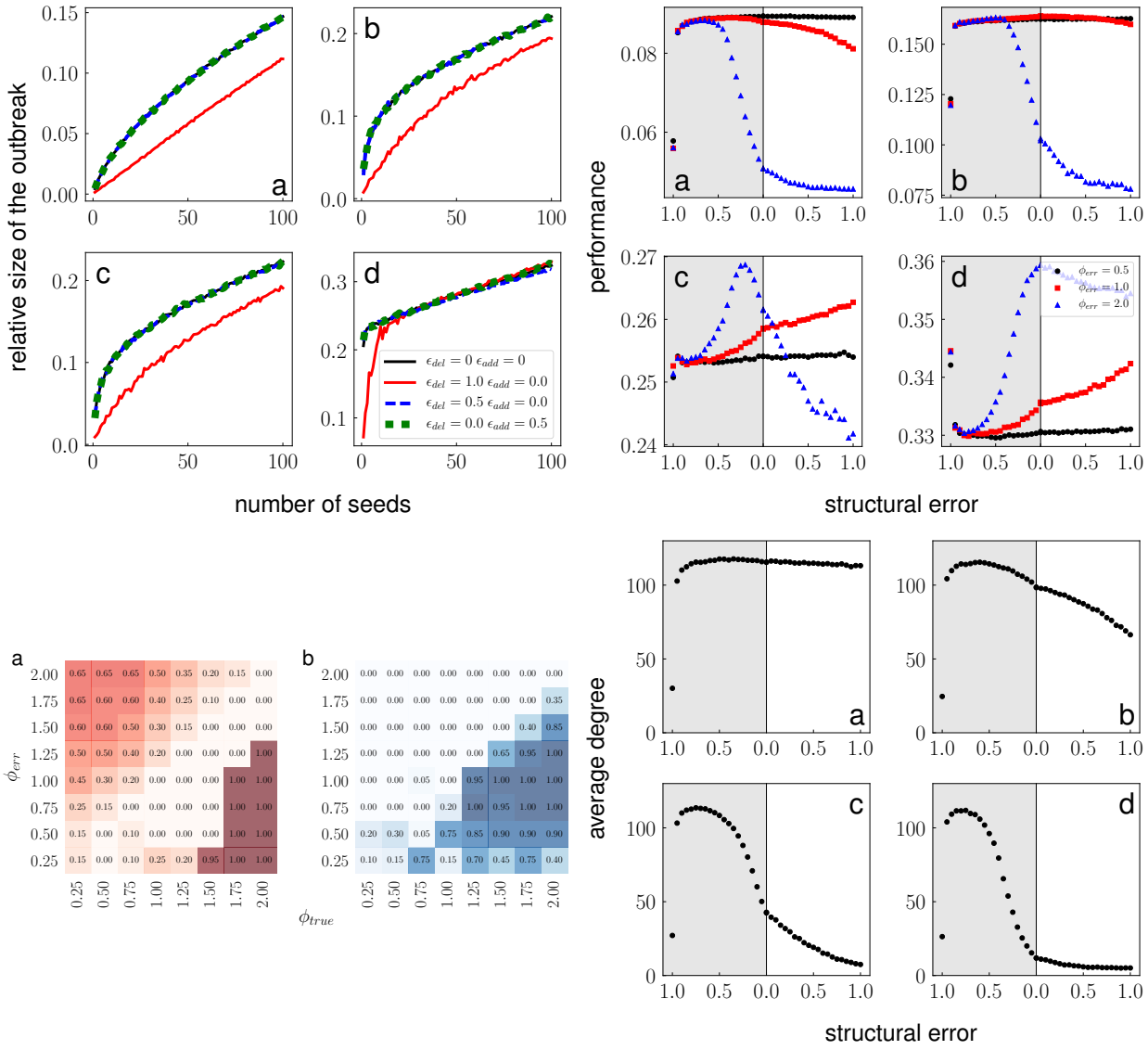

Figure B.18: **Political blogs.** $|\mathcal{X}_{err}| = 100$

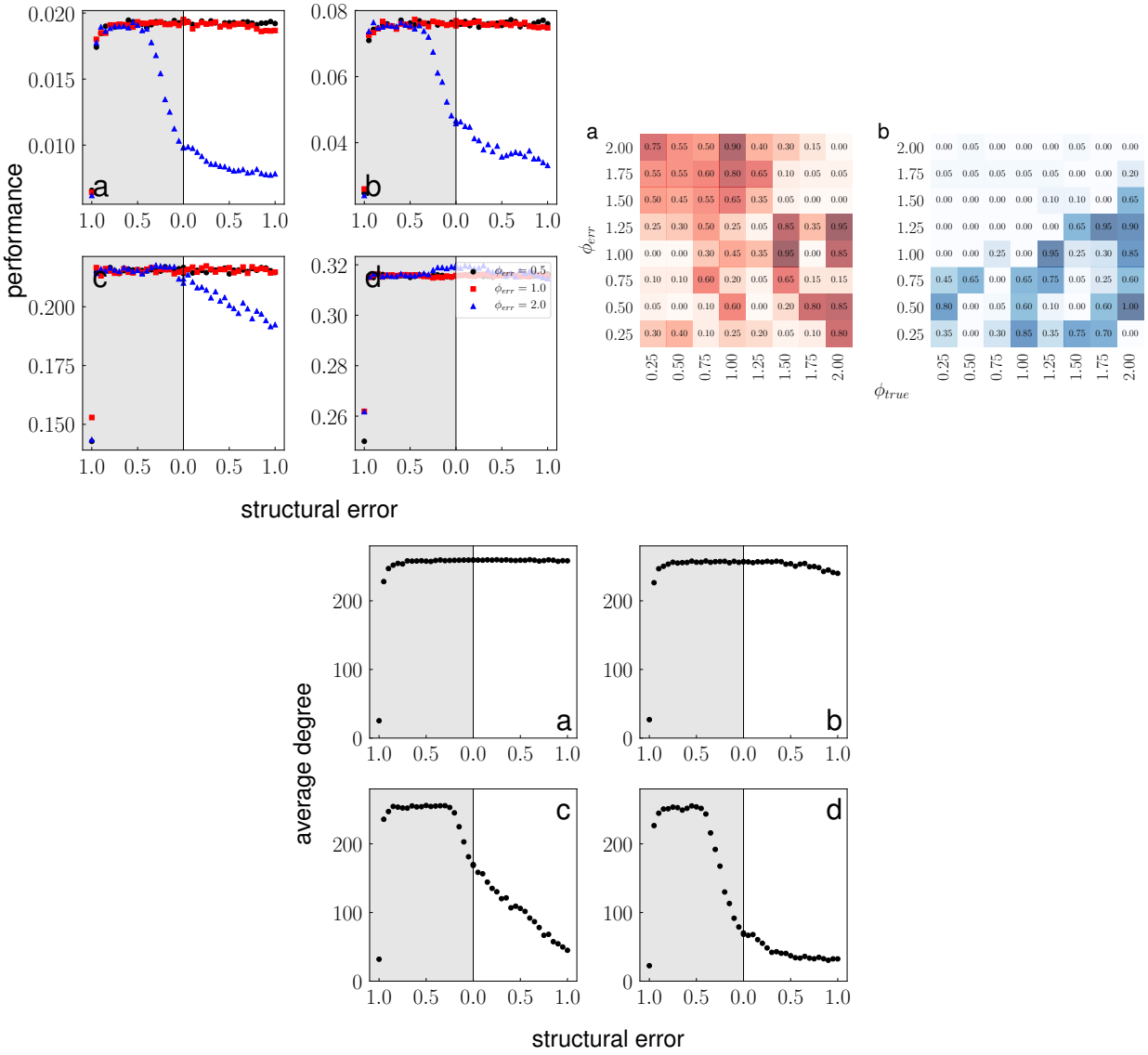

Figure B.19: **Political blogs.** $|\mathcal{X}_{err}| = 10$

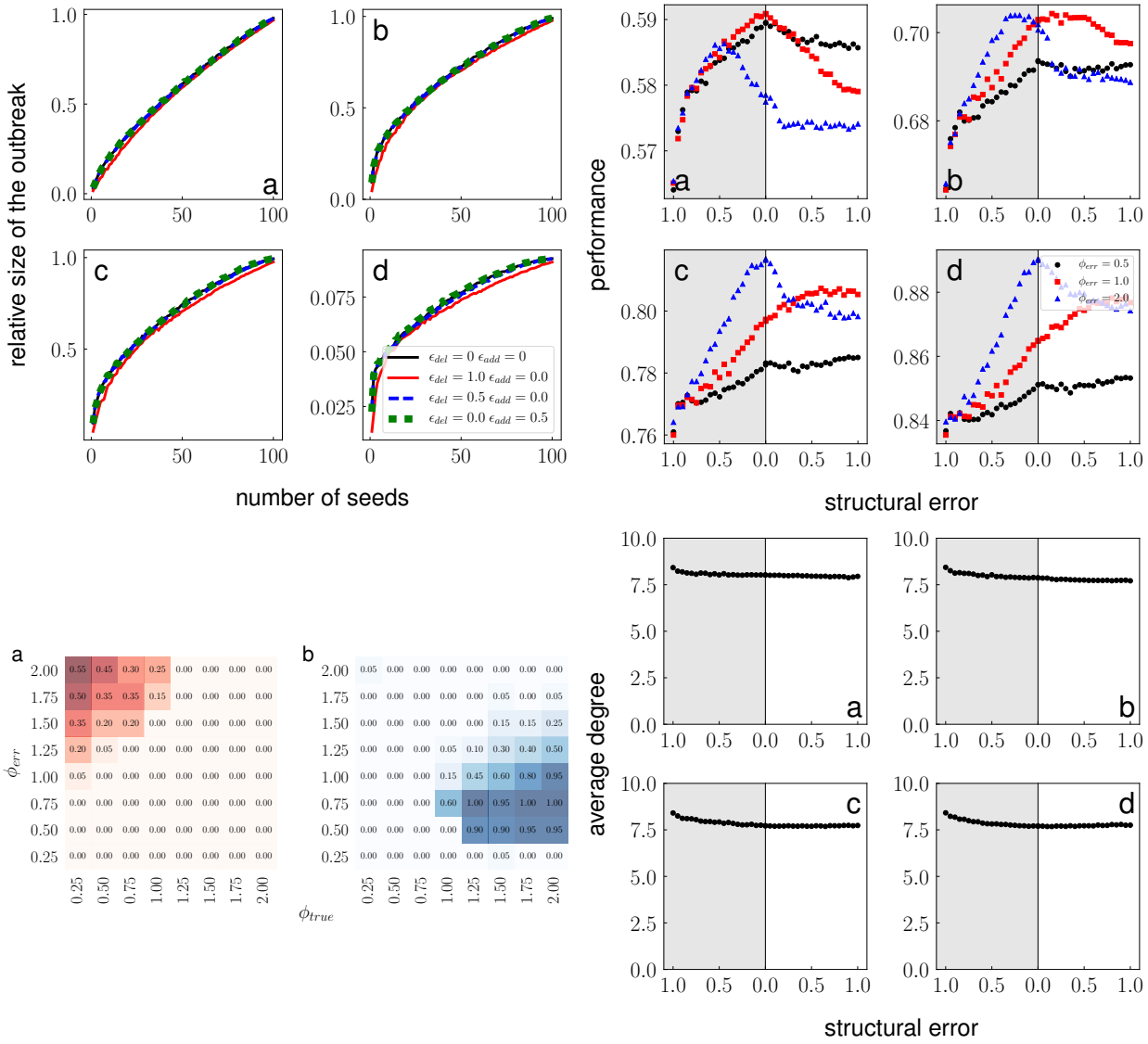

Figure B.20: **Political books.** $|\mathcal{X}_{err}| = 100$

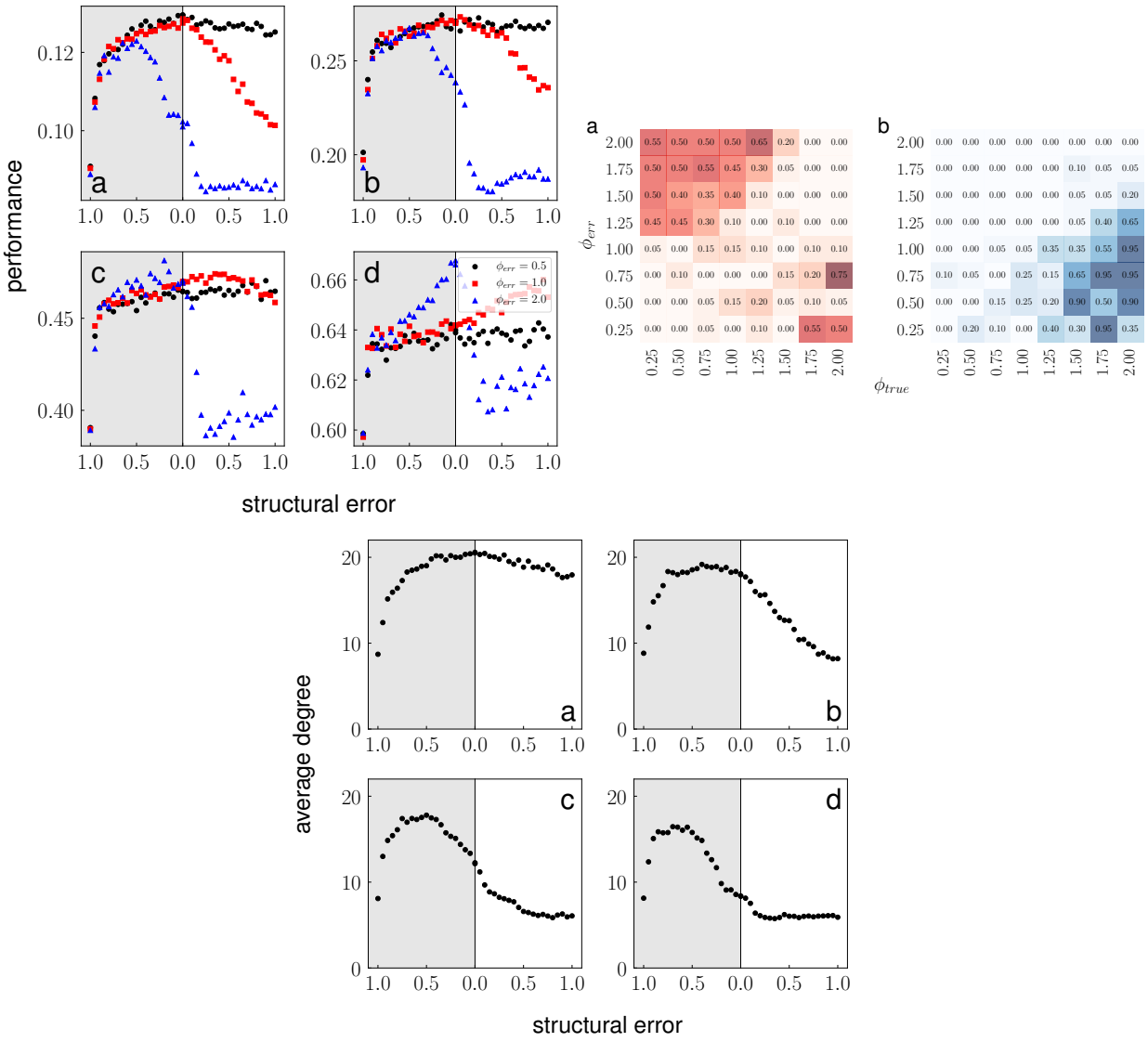

Figure B.21: **Political books.** $|\mathcal{X}_{err}| = 10$

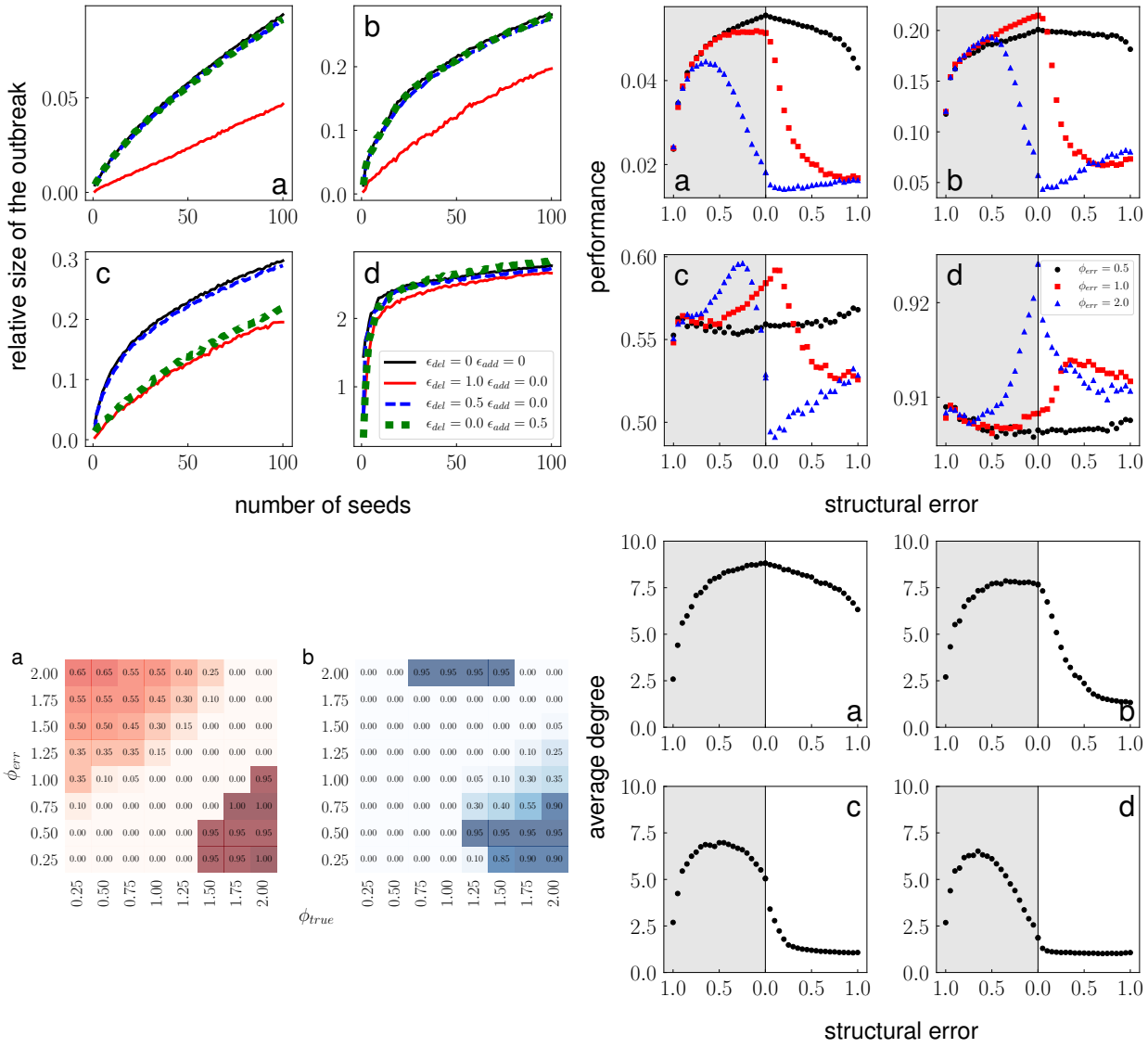

Figure B.22: **US Power grid.** $|\mathcal{X}_{err}| = 100$

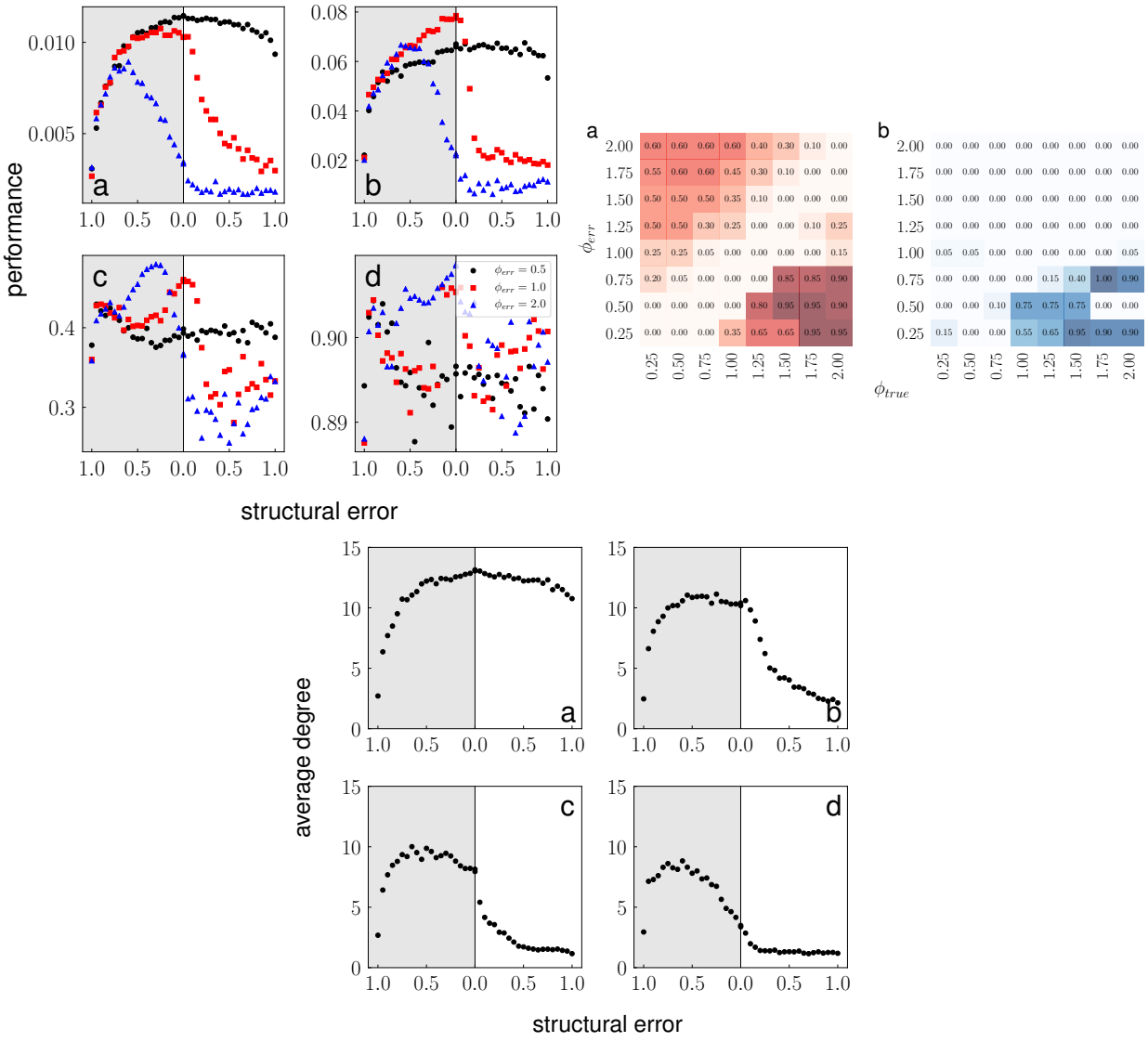

Figure B.23: **US Power grid.** $|\mathcal{X}_{err}| = 10$

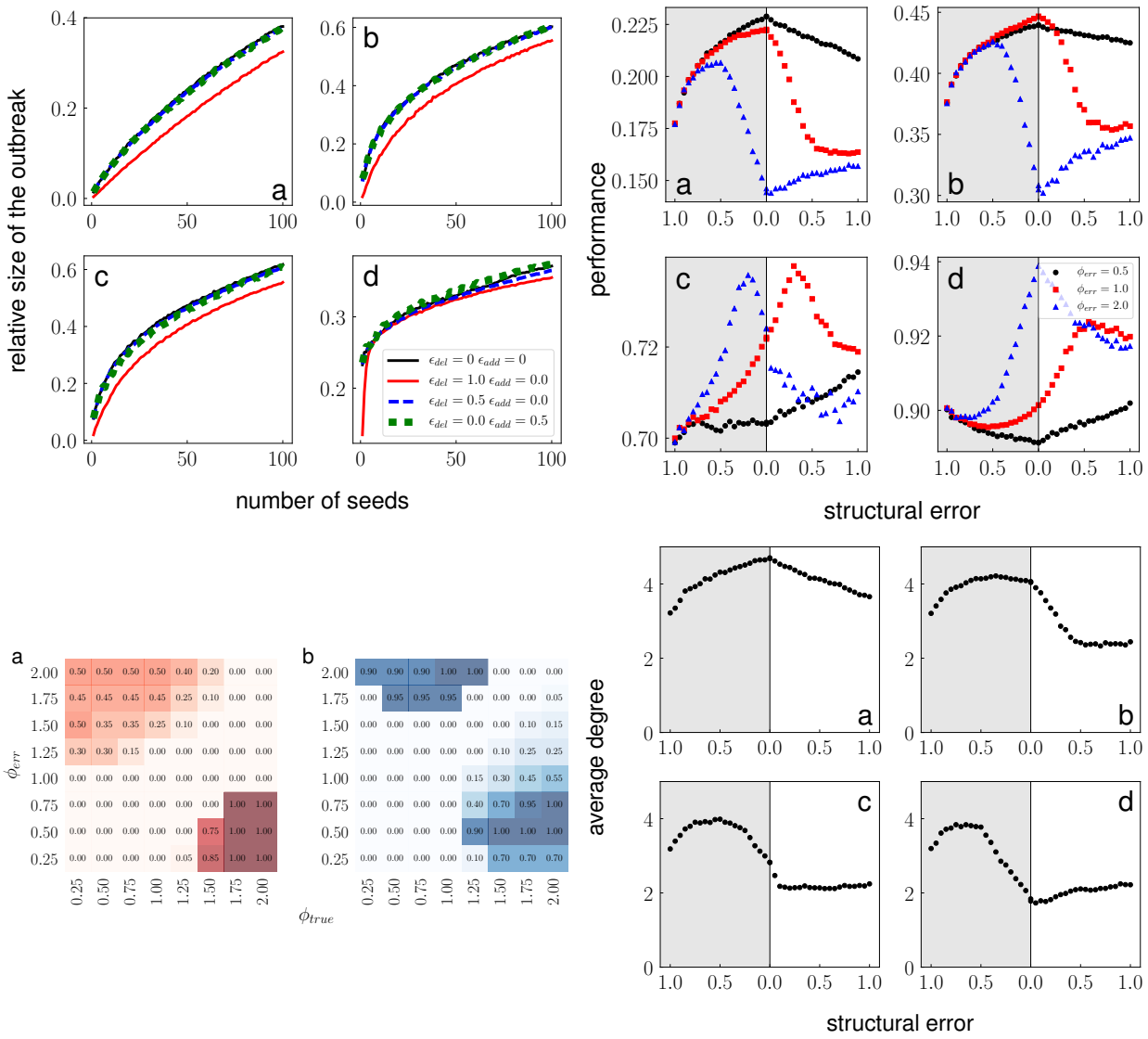

Figure B.24: **S 838.** $|\mathcal{X}_{err}| = 100$

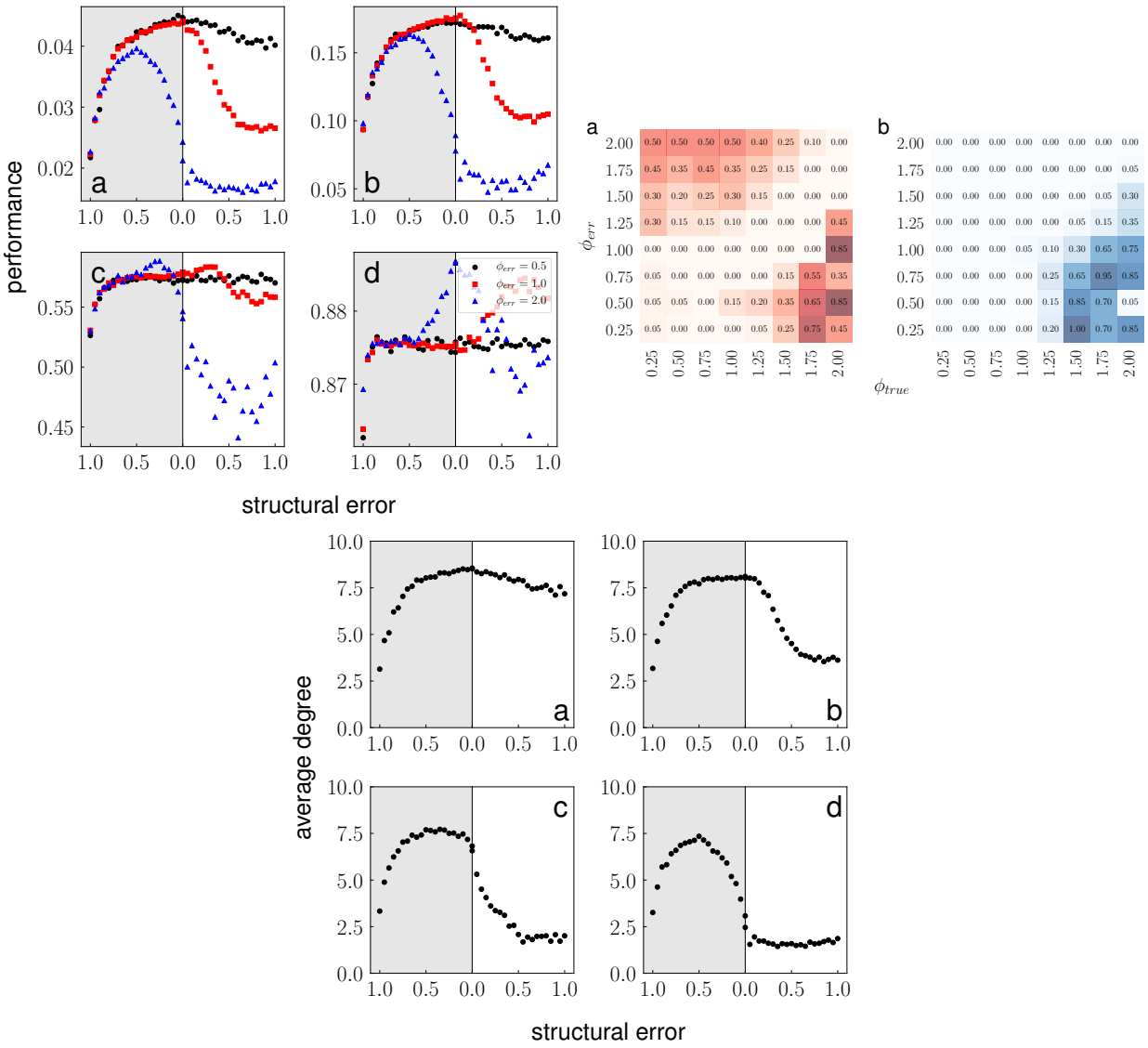

Figure B.25: **S 838.** $|\mathcal{X}_{err}| = 10$

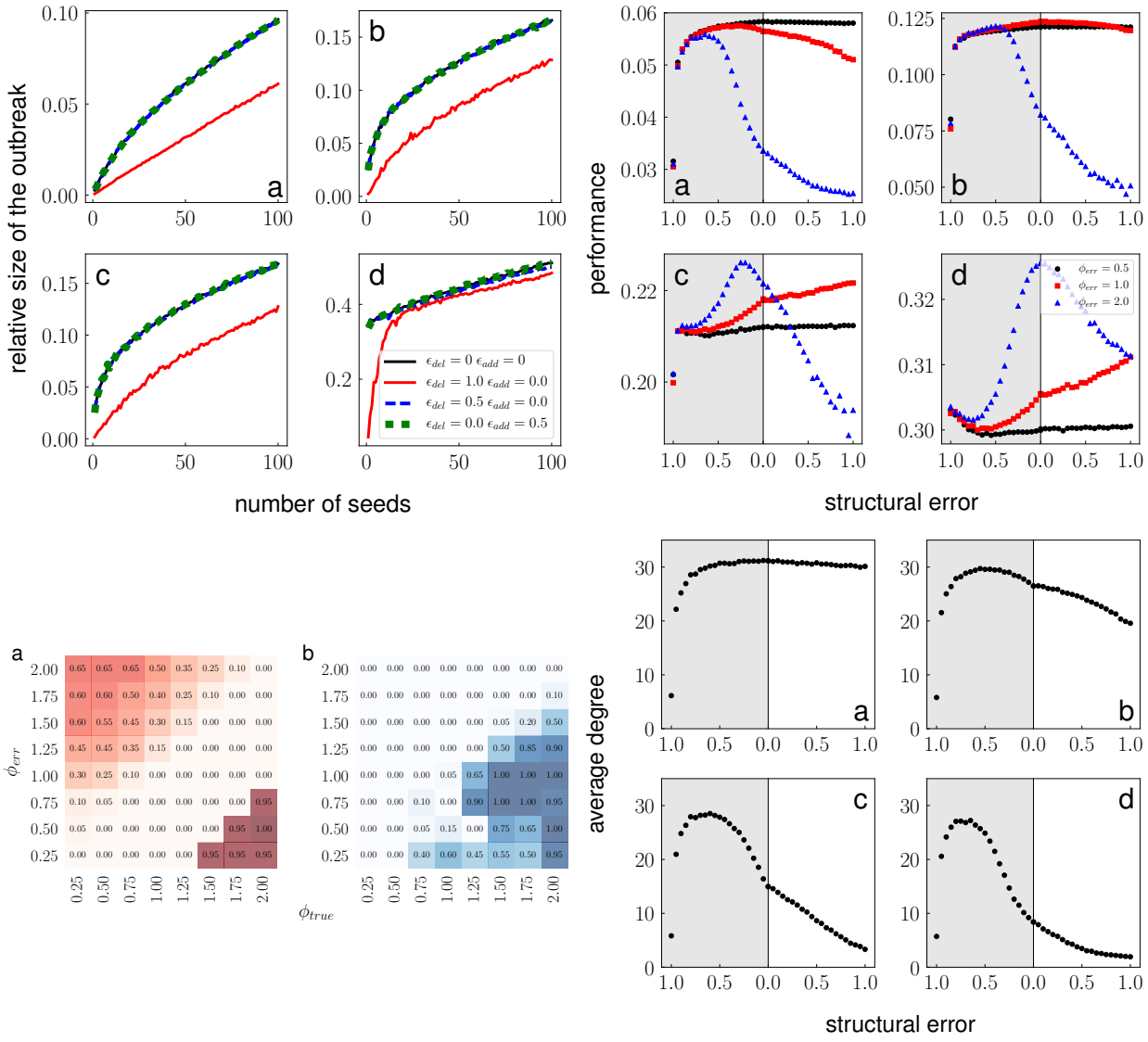

Figure B.26: **Yeast, protein.** $|\mathcal{X}_{err}| = 100$

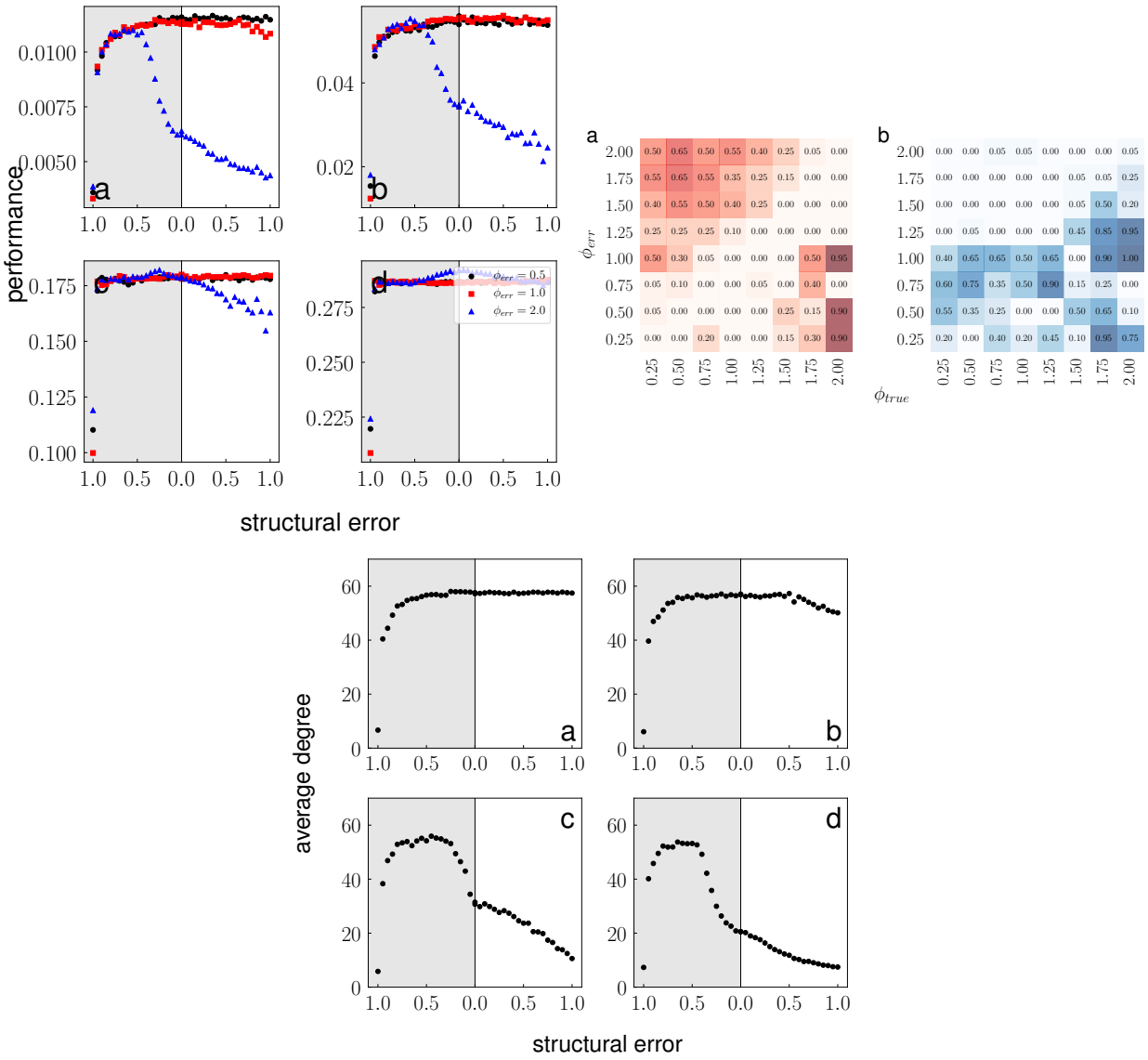

Figure B.27: **Yeast, protein.** $|\mathcal{X}_{err}| = 10$

C Appendix: Influence maximization on temporal networks

C.1 Network characteristics

The datasets used represent temporal networks with interactions between two entities of the network, either directed or undirected. Each interaction is represented with a time stamp indicating the time when it occurred. We divide these datasets into slices of equal length W in time. After dividing, we only consider those slices/layers that have more than 10% of all nodes in the dataset active for the given layer, *i.e.*, have at least one interaction in the layer. To give an example, the "Hypertext, 2009" dataset has been divided into slices of length 14,400 seconds. The division gives 15 slices in total. 4 of these 15 slices have less than 10% of all nodes active. These 4 slices are removed, and we end up with a temporal network with 11 layers. All the information for the other networks regarding the number of total, removed, and remaining slices can be found in Table C.1. Once the sparse layers are removed, we end up with temporal networks that have layers with densities shown Figure C.1.

C.2 Finding the critical threshold

In order to analyze the problem of influence maximization in different dynamical regimes, we need to estimate the critical value λ_c of the spreading probability λ of each temporal network. We remark that $\lambda_c = \lambda_c(\mu)$, *i.e.*, λ_c is a function of the recovery probability μ. To estimate λ_c, we start the spreading process from each active node in the first layer of the temporal network. The process is repeated 500 times for each node and averaged over all realizations and all nodes. The numerical simulations are done for a range of λ values, the λ value that maximizes the ratio of standard deviation and mean of outbreak sizes is found using brute-force search, and the value found gives the critical threshold for a temporal network and a given μ value.

Dataset	W	total	removed	T
Email, dept. 1	2880000	25	7	18
Email, dept. 2	2880000	25	7	18
Email, dept. 3	2880000	25	7	18
Email, dept. 4	2880000	25	7	18
High school, 2011	14400	19	8	11
High school, 2012	14400	51	30	21
High school, 2013	14400	26	12	14
Hospital ward	14400	25	5	20
Hypertext, 2009	14400	15	4	11
Primary school	7200	17	6	11
Workplace	28800	35	15	20
Workplace-2	28800	35	15	20

Table C.1: **List of the empirical datasets used to construct temporal networks.** From left to right, we report: the name of the dataset, the length W of the temporal window used to slice the data (time is reported in seconds), the total number of slices obtained by dividing the dataset, the number of removed slices that had less than 10% of all nodes active in it, and the number T of network layers resulting after slicing and cleaning data.

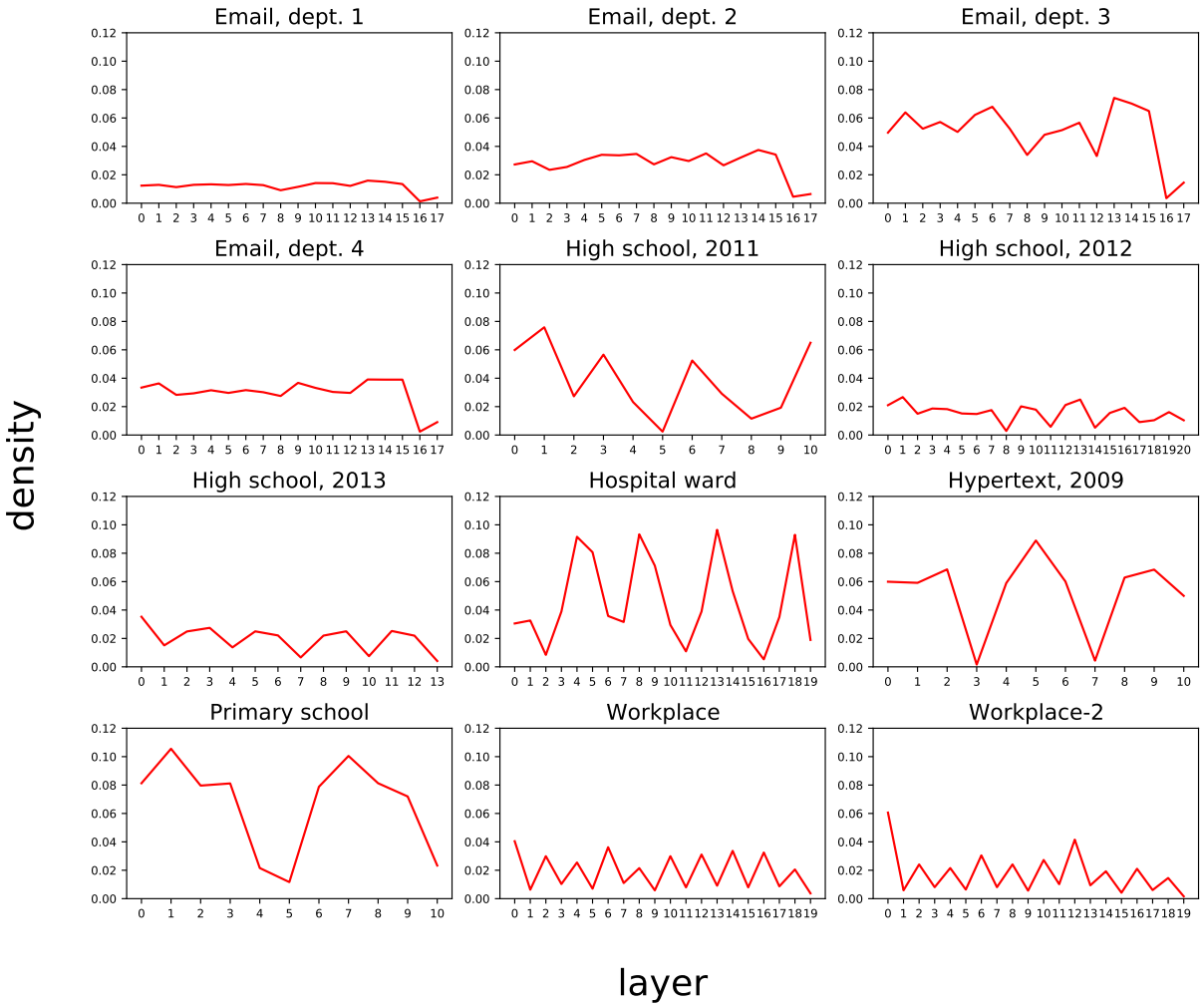

Figure C.1: **Density values of each layer in the temporal networks.** Each panel represents a single network, showing the evolution of density between layers. The density is defined as $2E_t/(N(N-1))$, where E_t is the number of edges in layer t, and N is the total number of nodes in the network, including all those that had at least one interaction in the whole temporal network.

C.3 Effect of shuffling layer order on critical threshold

In our analysis, we observe that the temporal ordering of the layers has a significant effect on the dynamics. The value of the critical threshold changes with the ordering, and also the influential spreaders identified differ depending on the layer ordering. In Figures C.2-C.12 we show the effect of shuffling the order of the layers on the critical threshold value of the temporal network. In most cases, we observe that the true value of the critical threshold is lower than the mean and median of the critical thresholds calculated from multiple realizations of the temporal network with randomly ordered layers. This pattern suggests that the interaction dynamics in a temporal network are not happening randomly, and there is a correlation between the interaction patterns of layers closer to each other in the ordering.

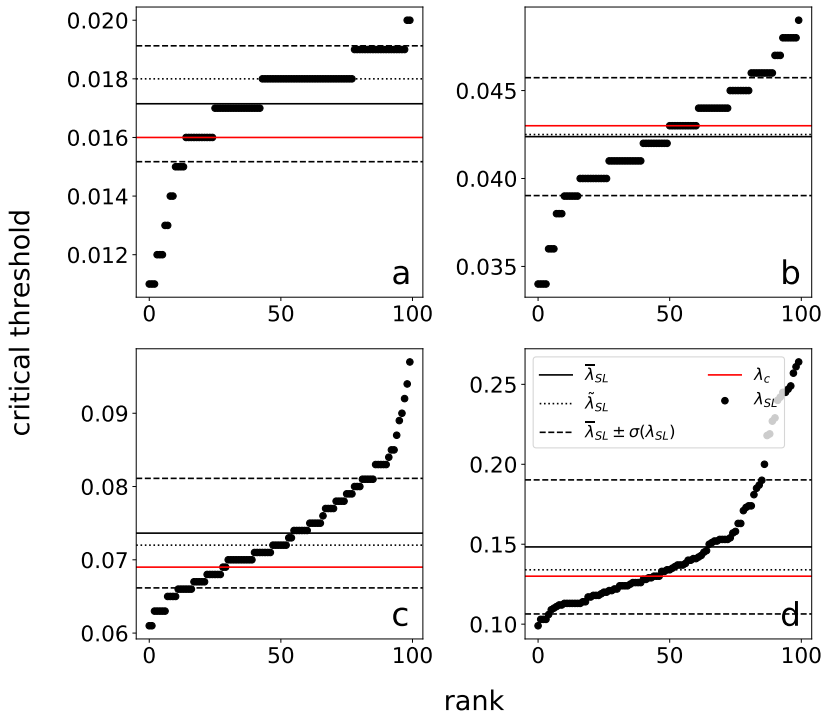

Figure C.2: **Sensitivity of the critical threshold.** (a) Best estimates of the critical spreading probability λ_{SL} for randomized versions of the "Email, dept. 1" temporal network. SIR recovery probability is $\mu = 0$. We display horizontal lines identifying the average $\bar{\lambda}_{SL}$ (full black line), the region corresponding to one standard deviation away from the mean $[\bar{\lambda}_{SL} \pm \sigma(\lambda_{SL})$, dashed black lines], the median value $\tilde{\lambda}_{SL}$ (dotted black line), and the actual critical value λ_c measured on the non-randomized version of the network (red full line). (b) Same as in panel a, but for $\mu = 0.25$. (c) Same as in panel a, but for $\mu = 0.5$. (d) Same as in panel a, but for $\mu = 1$.

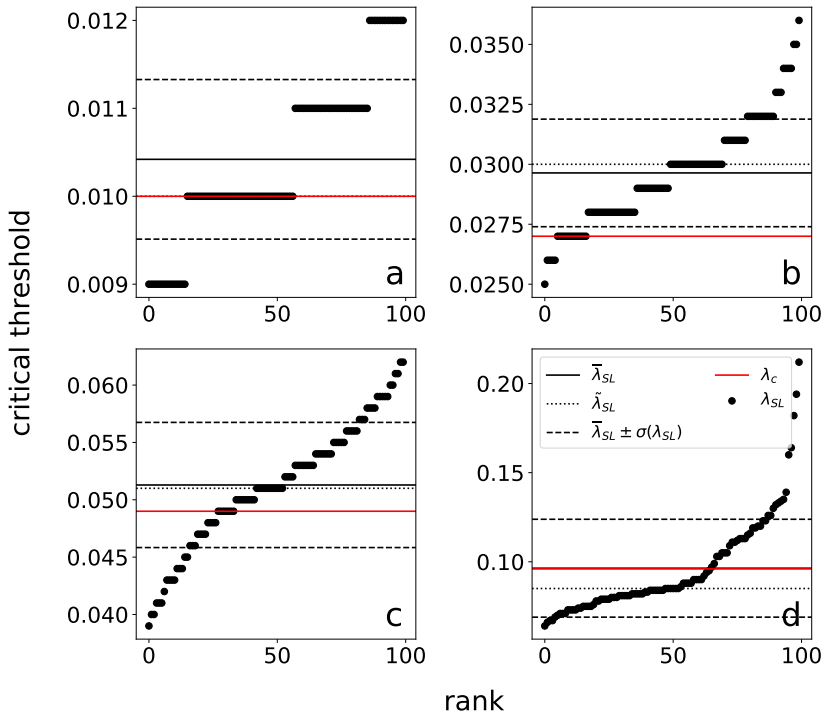

Figure C.3: Same as Fig. C.2, but for "Email, dept. 2" network.

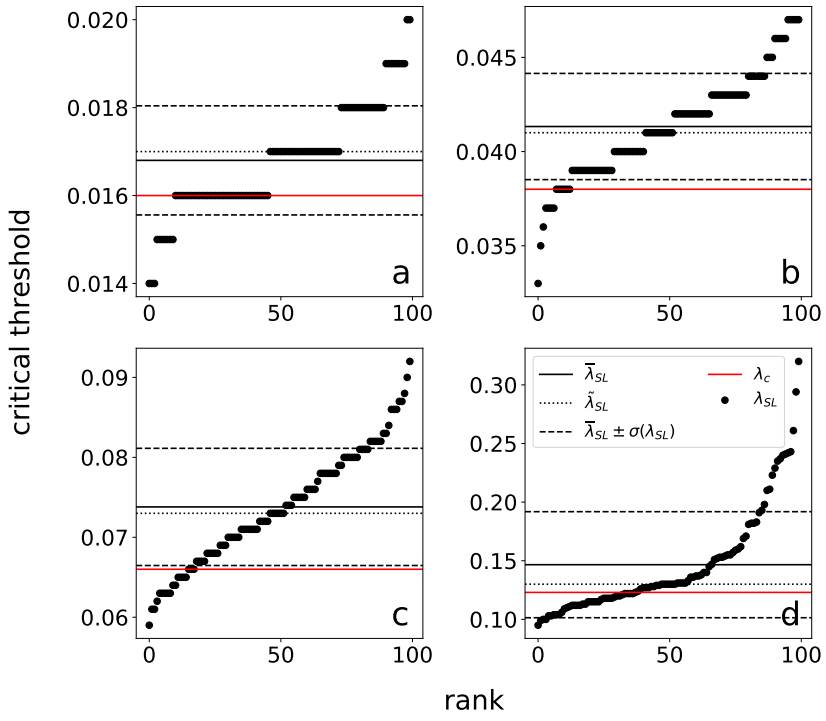

Figure C.4: Same as Fig. C.2, but for "Email, dept. 3" network.

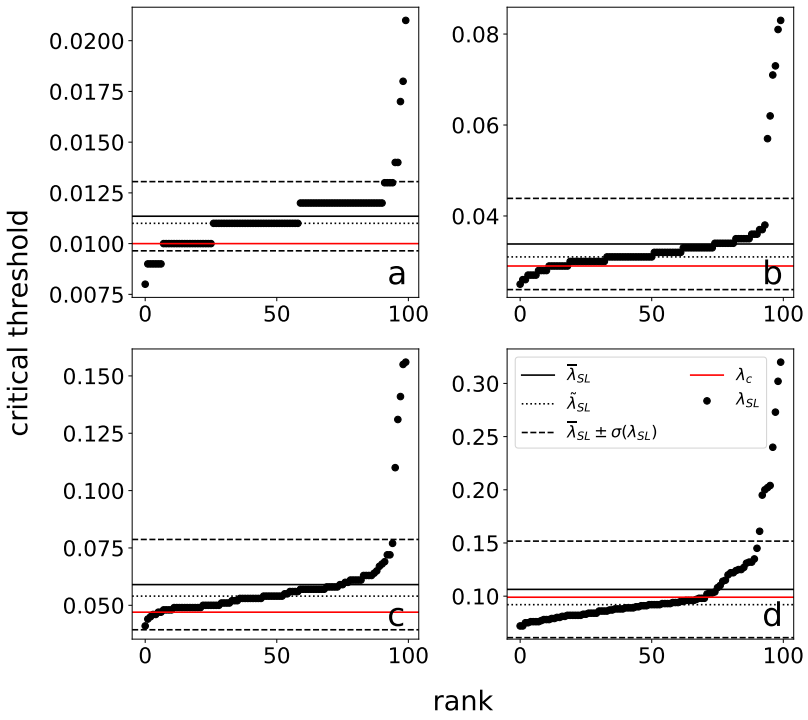

Figure C.5: Same as Fig. C.2, but for "Email, dept. 4" network.

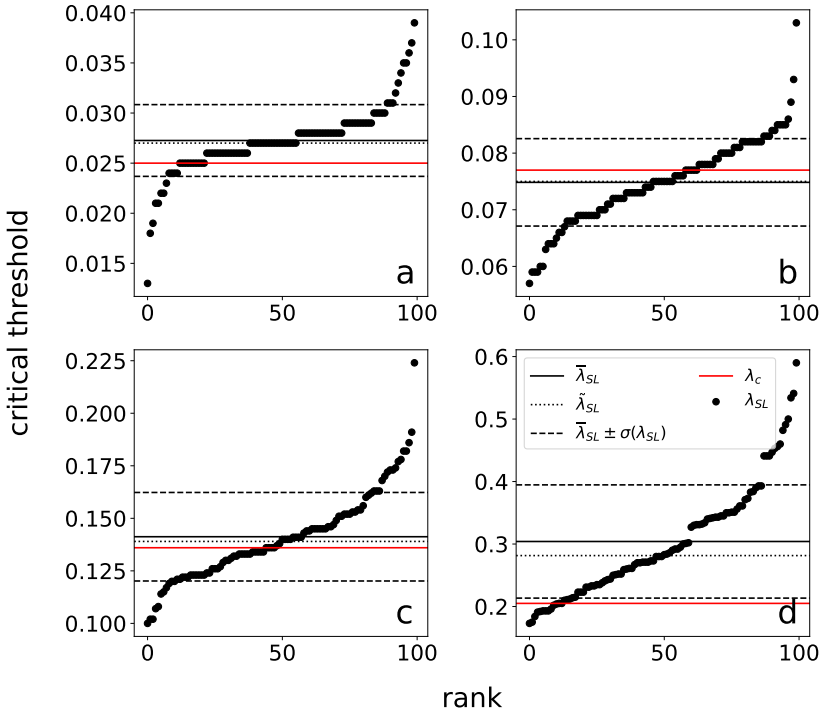

Figure C.6: Same as Fig. C.2, but for "High school, 2012" network.

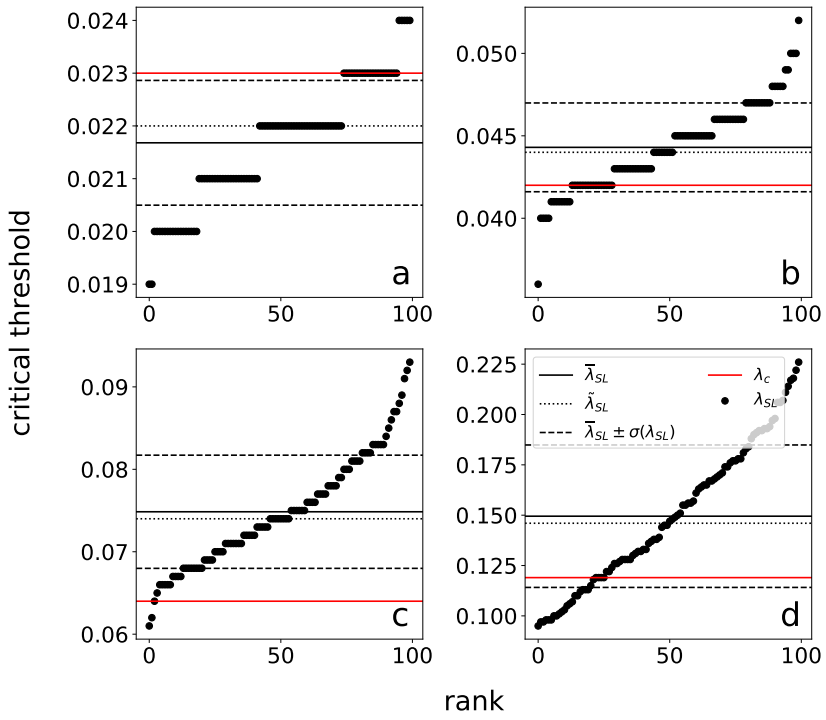

Figure C.7: Same as Fig. C.2, but for "High school, 2013" network.

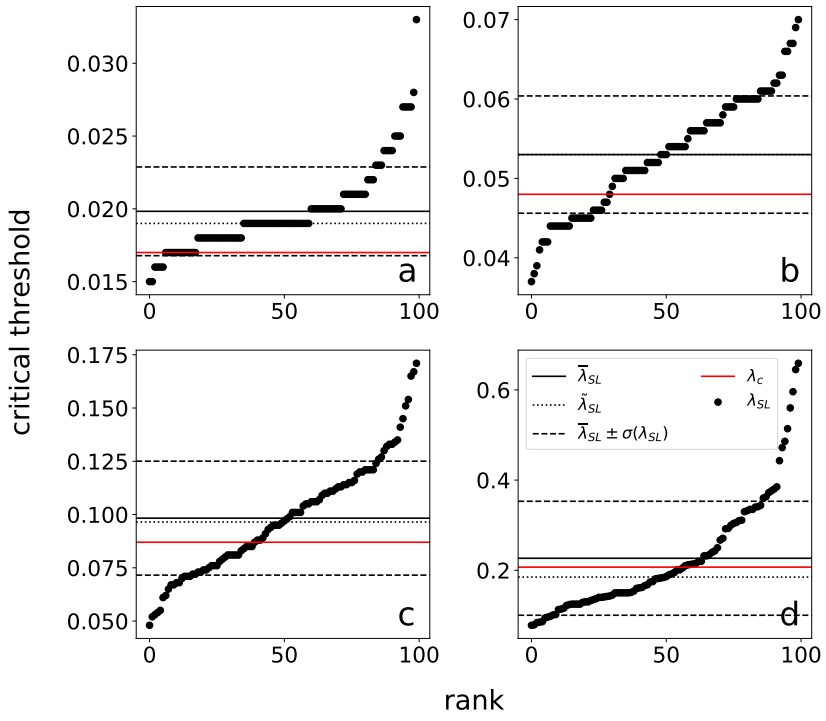

Figure C.8: Same as Fig. C.2, but for "Hospital ward" network.

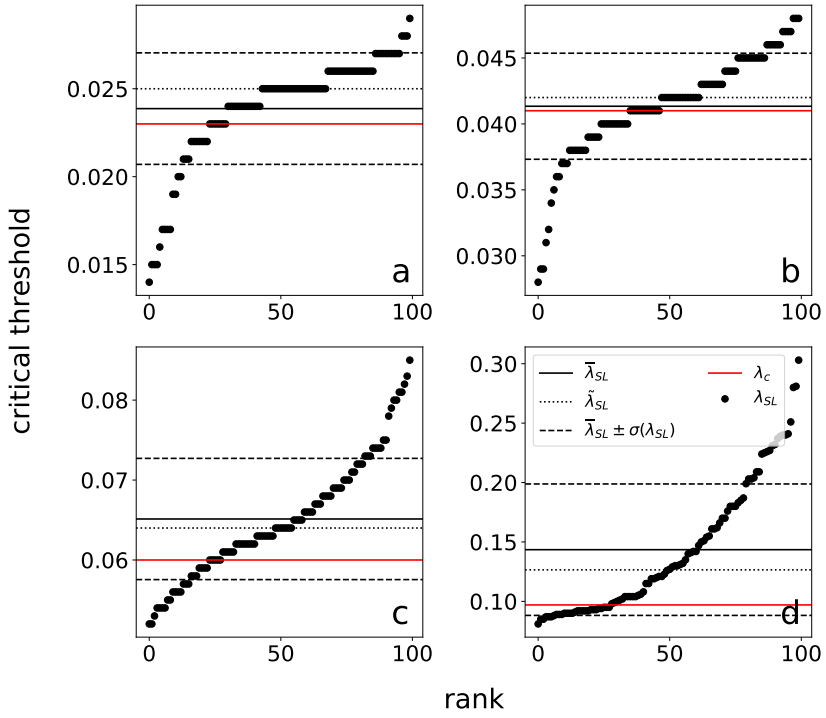

Figure C.9: Same as Fig. C.2, but for "Hypertext, 2009" network.

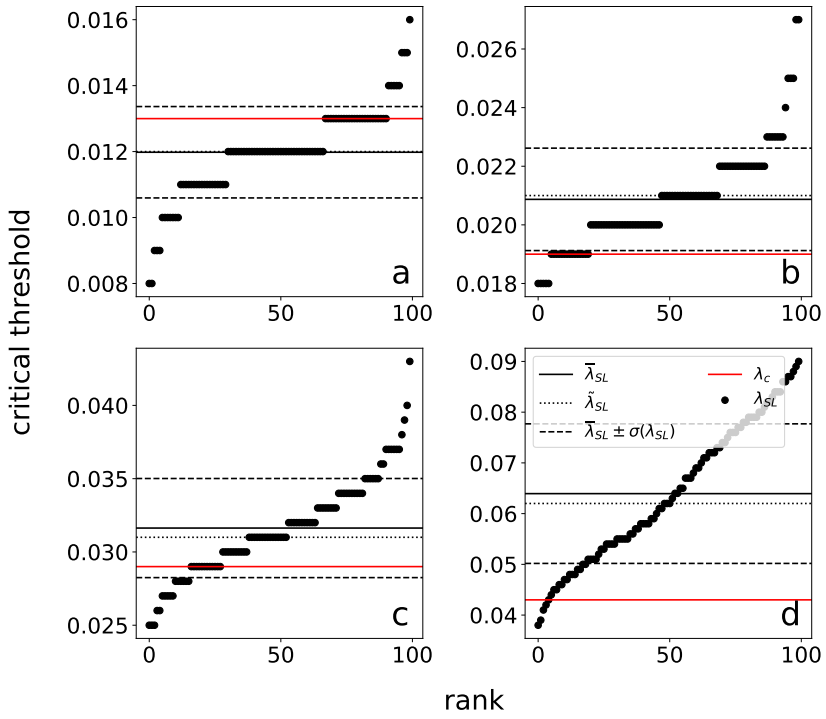

Figure C.10: Same as Fig. C.2, but for "Primary school" network.

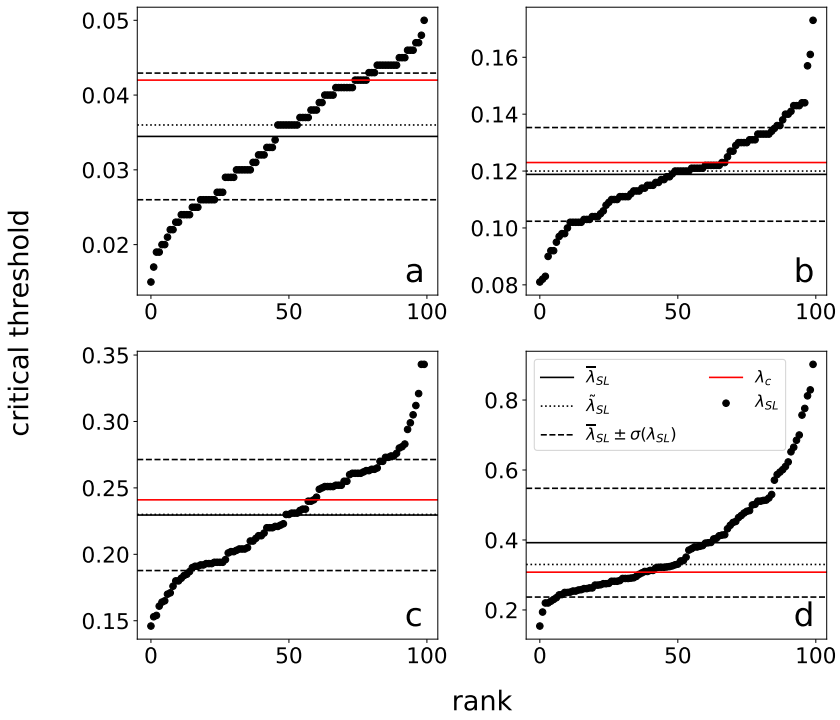

Figure C.11: Same as Fig. C.2, but for "Workplace" network.

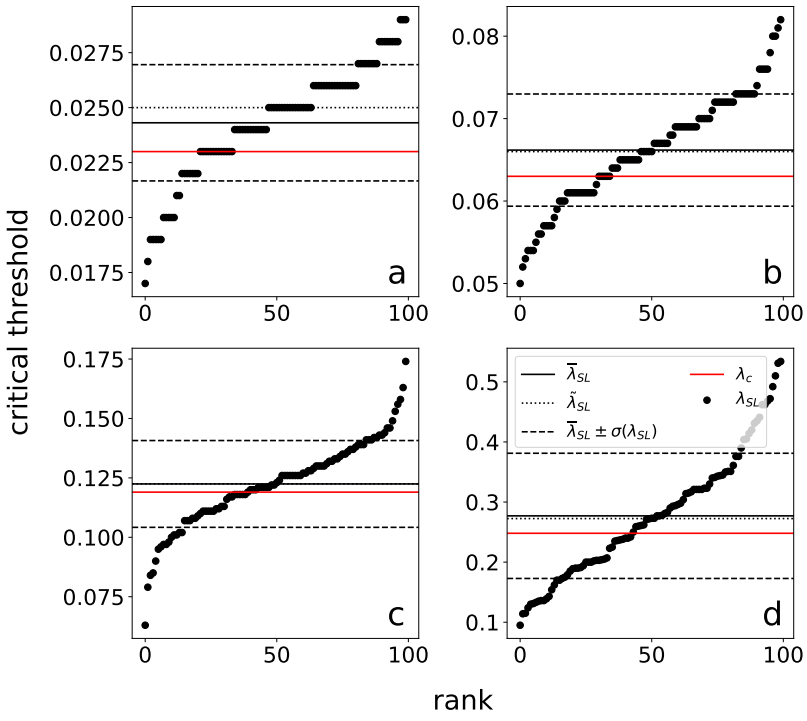

Figure C.12: Same as Fig. C.2, but for "Workplace-2" network.

C.4 Accounting for time horizon

We consider three methods that use a static network as the only topological input for the identification of the influential spreaders. The first uses the first layer (FL), the second uses a randomly selected layer (RL), and the last uses the aggregated version of the temporal network (ST) to find influential spreaders. When the SIR model is employed on a static network, traditionally the model stops when there are no more infected nodes remaining. In our SIR model on temporal networks, the spreading happens until the end of the time horizon (*i.e.*, number of layers in a temporal network). In the three methods previously mentioned, when $\mu > 0$, there is no information on time horizon (when $\mu = 0$, the time horizon is known because if we let the SIR model run until there are no more infected nodes, the spreading would end when all nodes are infected, which is not informative for selecting influential spreaders). To observe the effect of knowing the time horizon on these three methods, we implement FL-T, RL-T, and ST-T. All these methods run for T (number of layers) steps to find influential spreaders, rather than waiting for no nodes to be left in the infected state. The results can be observed in Figure C.13. From the figure, it can be observed that adding the information of time horizon to the methods has no significant effect on their performance.

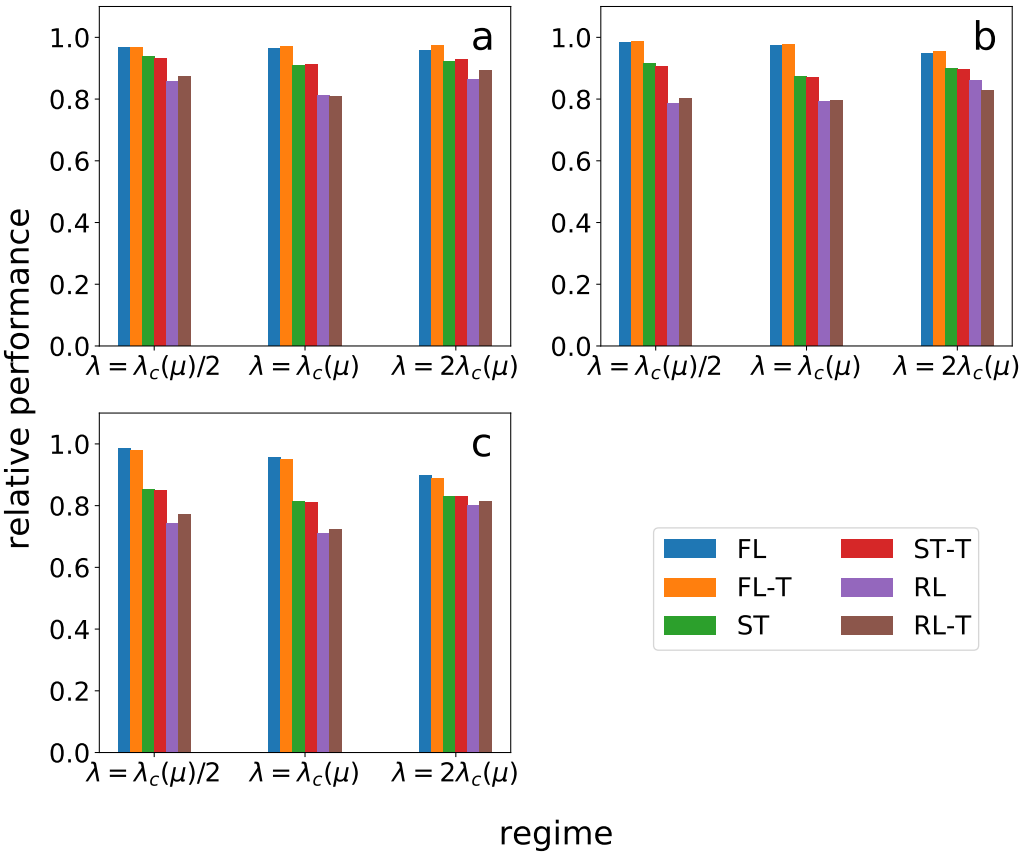

Figure C.13: **Relative performances of methods for identifying influential spreaders.** (a) Similar to Figure 5.5, but for different methods. The figure serves to analyze the effect of the information of time horizon on the identification of influential spreaders. Different dynamical regimes are studied with recovery probability $\mu = 0.25$ fixed. (b) Same as in panel a, but for $\mu = 0.5$. (c) Same as in panel a, but for $\mu = 1$.

C.5 Results for tests of performance

Here we present the average value of the outbreak size as a function of the relative size of the seed set, for all networks, under different values of the spreading and recovery probabilities. The figures are similar to the Figure 5.4, but here all the methods analyzed are included in the plots.

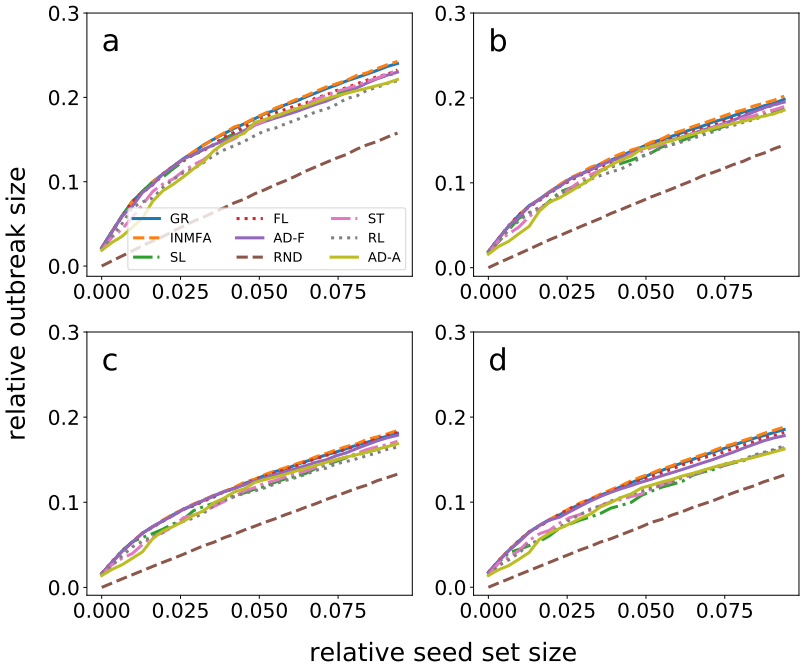

Figure C.14: **Identification of influential spreaders in temporal networks.** (a) Average value of the relative size of the outbreak, *i.e.*, $\langle O(\mathcal{X})\rangle$, as a function of the relative size of the seed set, *i.e.*, $|\mathcal{X}|/N$. The network analyzed is "Email, dept. 1". Spreading dynamics is subcritical, *i.e.*, $\lambda = 0.5\lambda_c(\mu)$, and the recovery probability is $\mu = 0$. (b) Same as in panel a, but for $\mu = 0.25$. (c) Same as in panel a, but for $\mu = 0.5$. (d) Same as in panel a, but for $\mu = 1$.

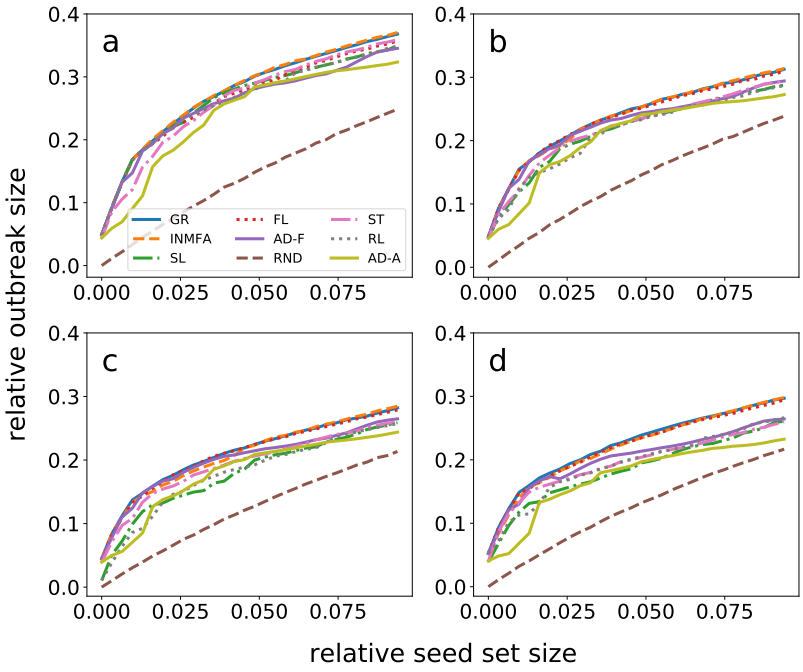

Figure C.15: Same as Fig. C.14, but for "Email, dept. 1" network in critical regime, *i.e.*, $\lambda = \lambda_c(\mu)$.

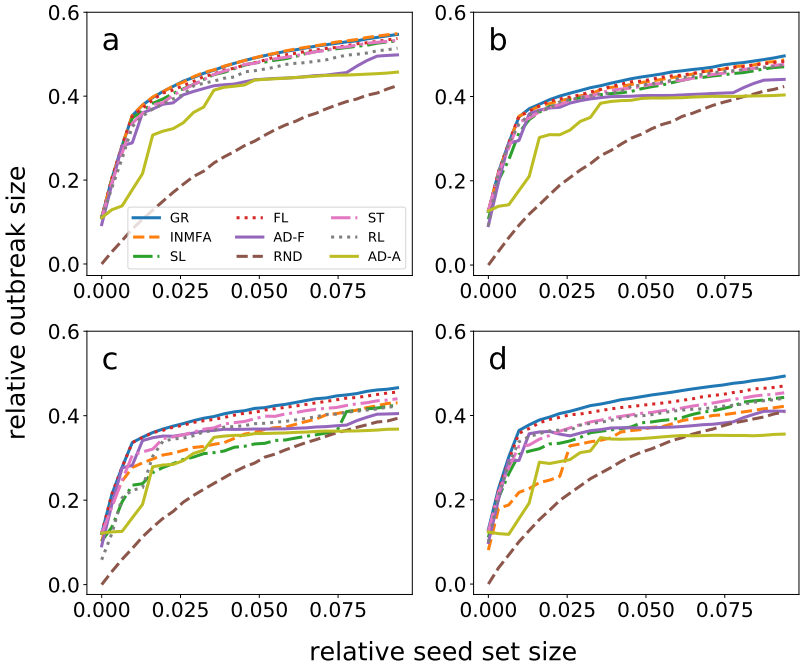

Figure C.16: Same as Fig. C.14, but for "Email, dept. 1" network in supercritical regime, *i.e.*, $\lambda = 2\lambda_c(\mu)$.

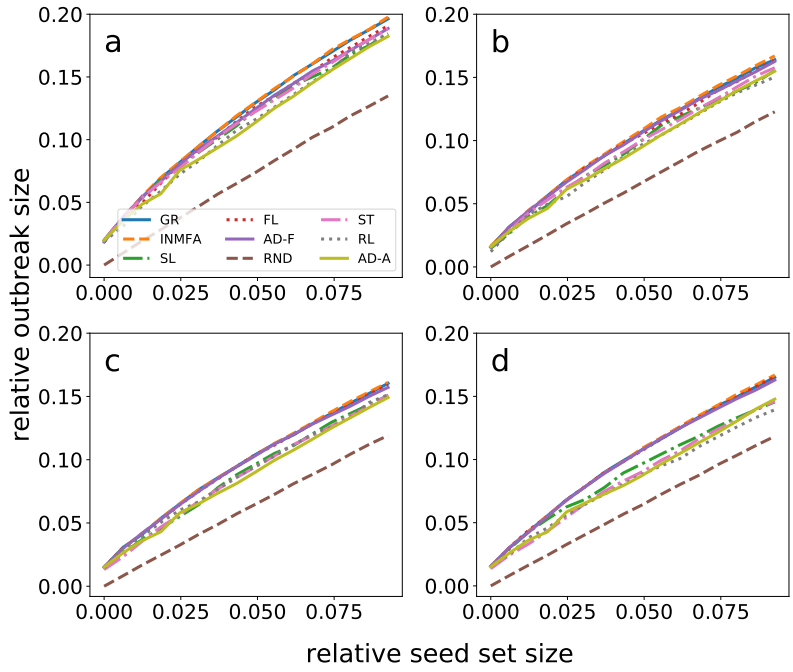

Figure C.17: Same as Fig. C.14, but for "Email, dept. 2" network in subcritical regime.

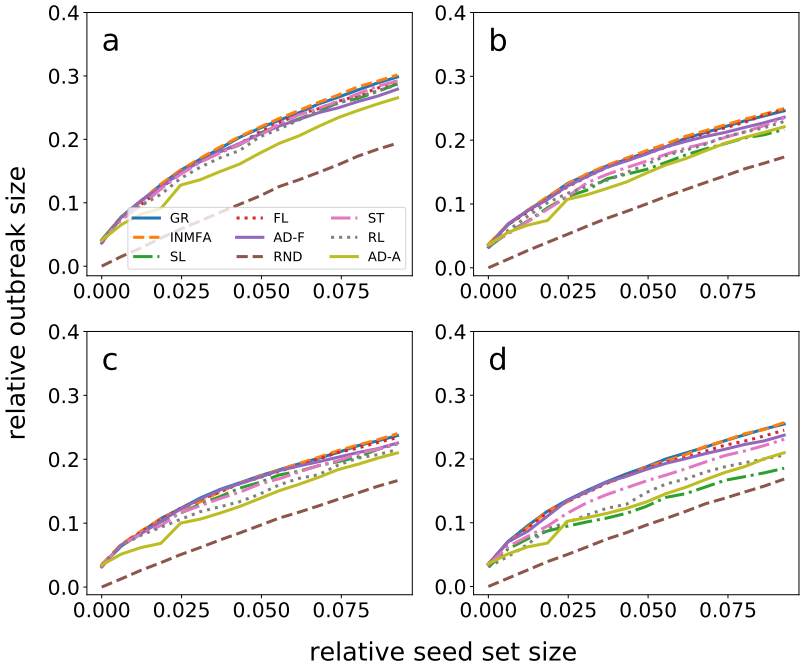

Figure C.18: Same as Fig. C.14, but for "Email, dept. 2" network in critical regime.

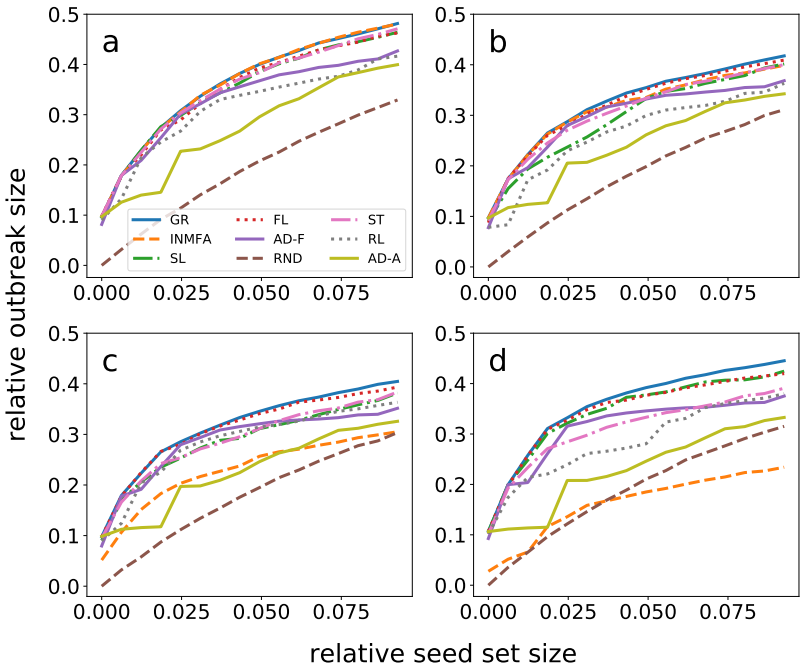

Figure C.19: Same as Fig. C.14, but for "Email, dept. 2" network in supercritical regime.

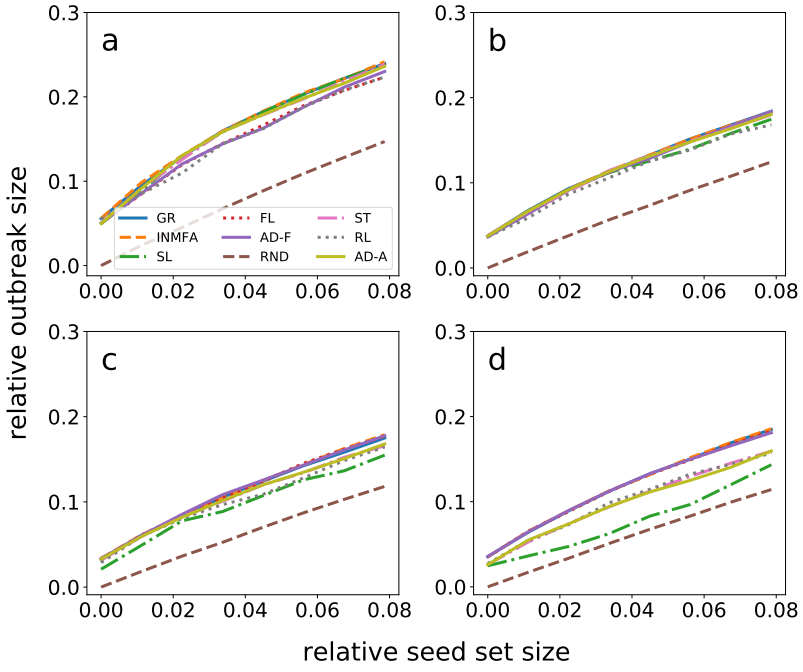

Figure C.20: Same as Fig. C.14, but for "Email, dept. 3" network in subcritical regime.

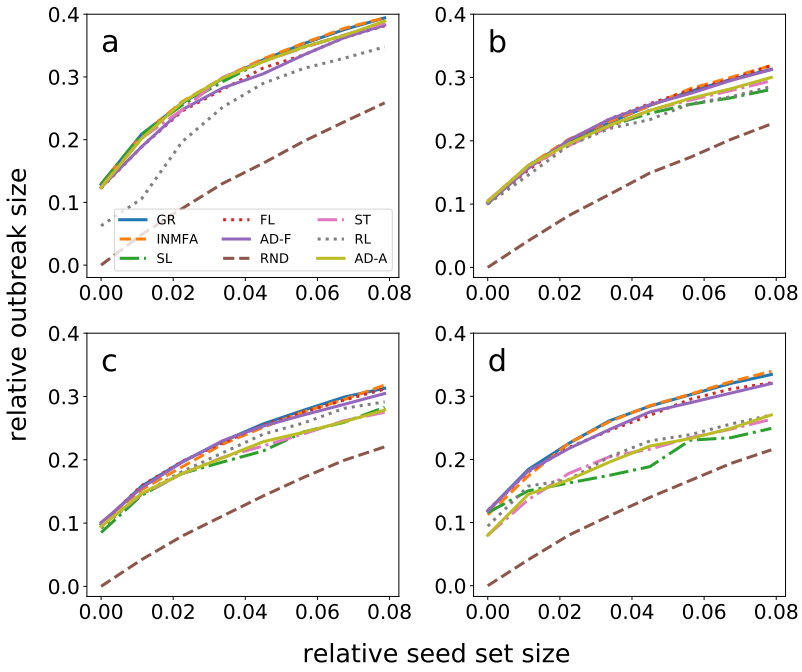

Figure C.21: Same as Fig. C.14, but for "Email, dept. 3" network in critical regime.

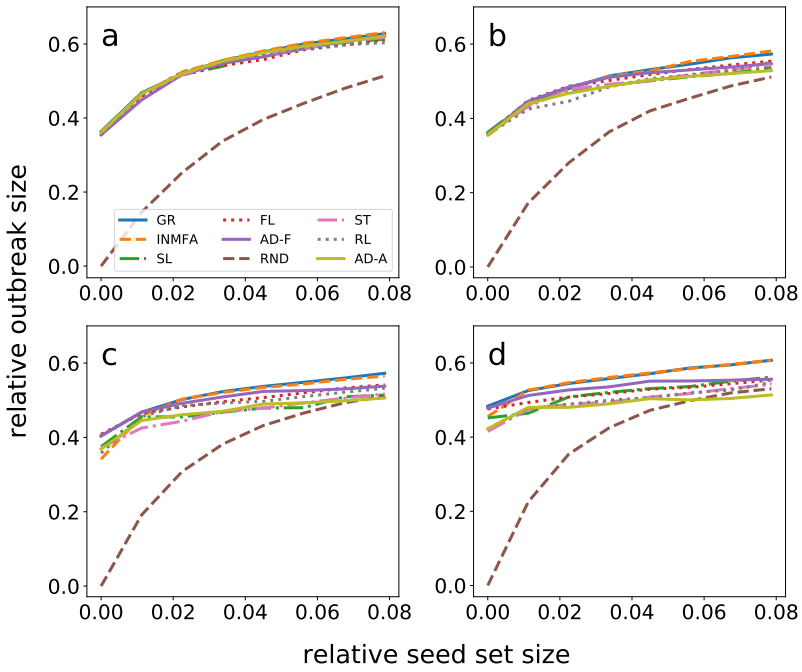

Figure C.22: Same as Fig. C.14, but for "Email, dept. 3" network in supercritical regime.

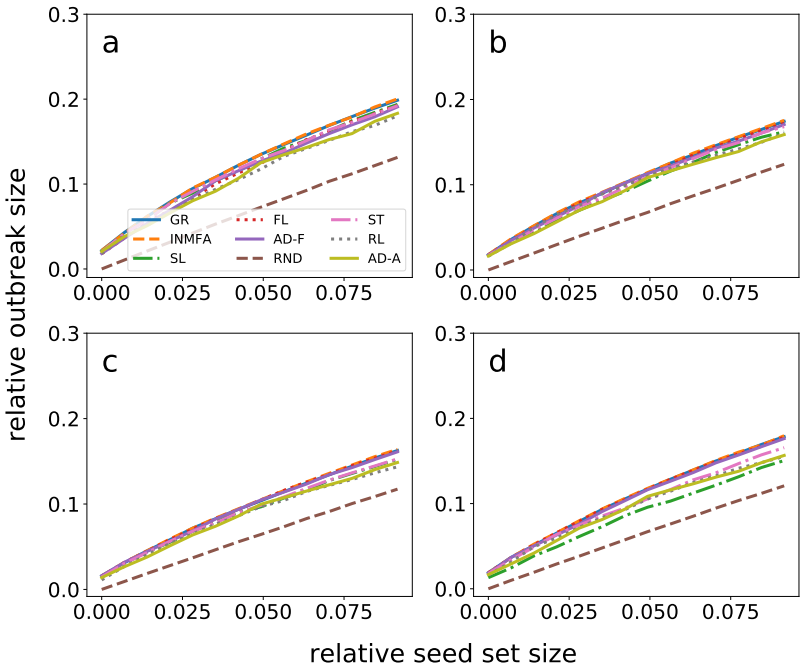

Figure C.23: Same as Fig. C.14, but for "Email, dept. 4" network in subcritical regime.

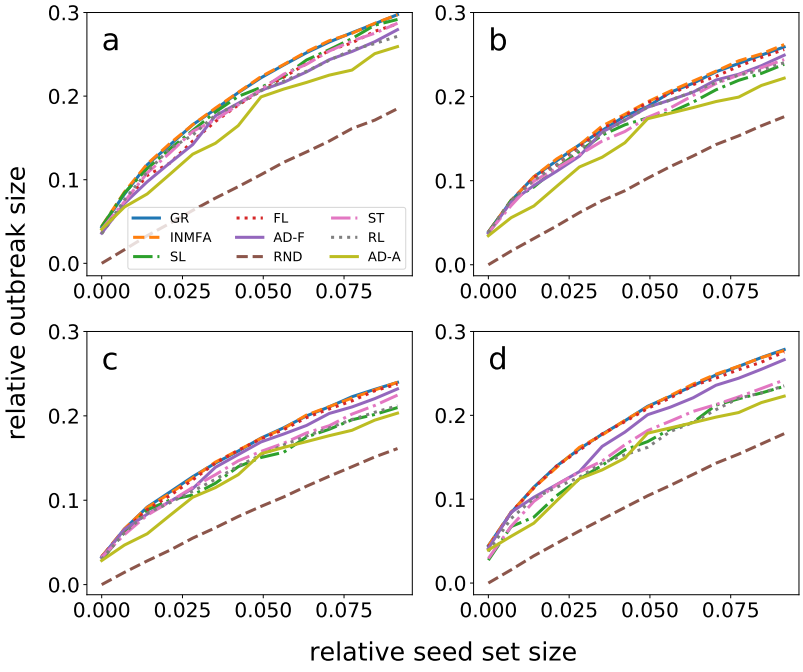

Figure C.24: Same as Fig. C.14, but for "Email, dept. 4" network in critical regime.

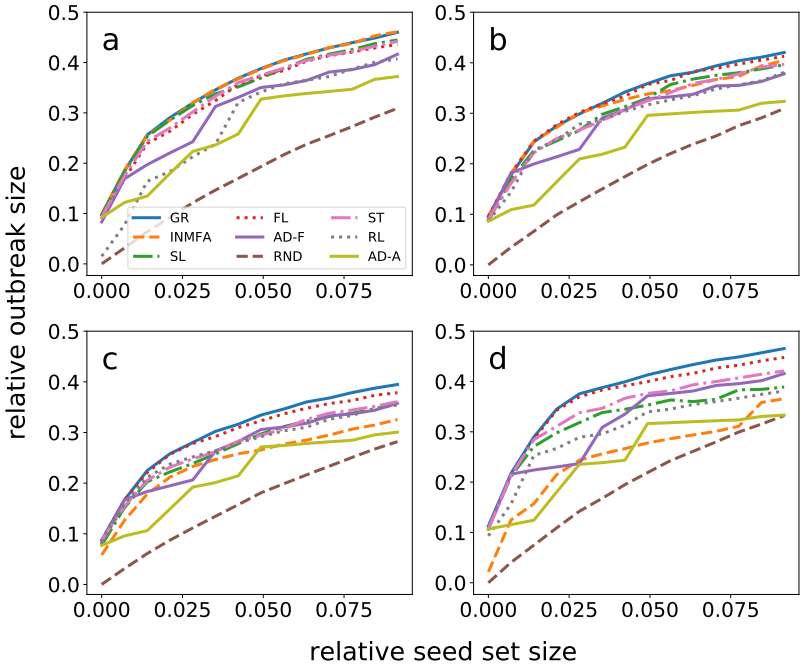

Figure C.25: Same as Fig. C.14, but for "Email, dept. 4" network in supercritical regime.

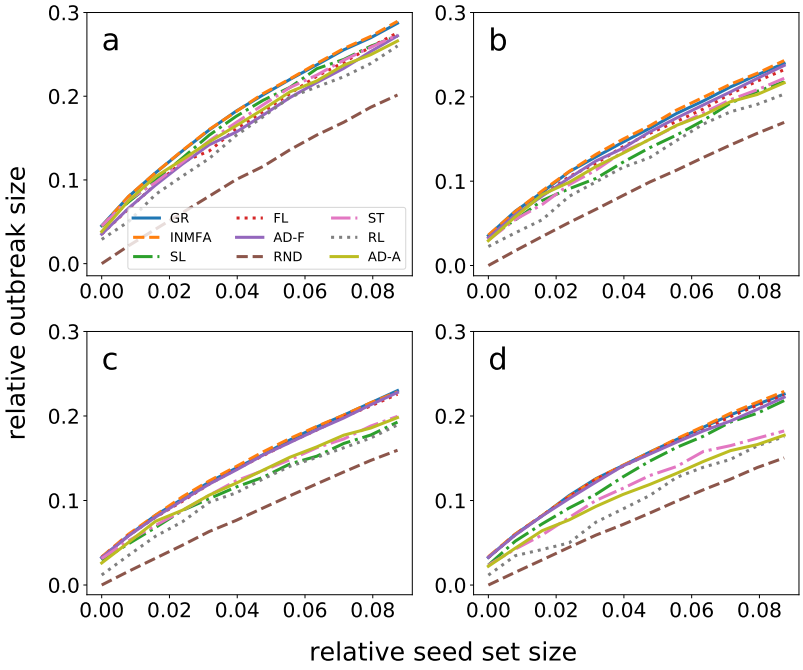

Figure C.26: Same as Fig. C.14, but for "High school, 2011" network in subcritical regime.

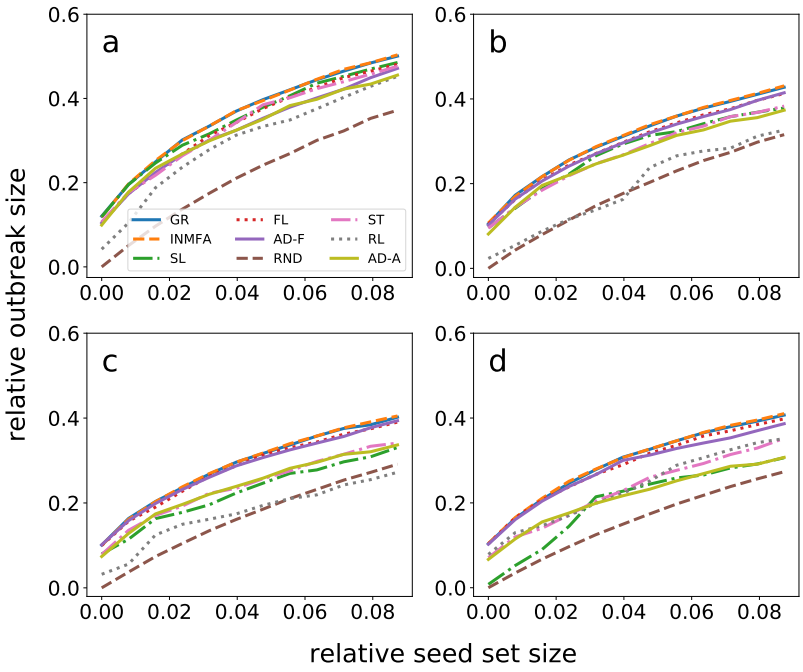

Figure C.27: Same as Fig. C.14, but for "High school, 2011" network in critical regime.

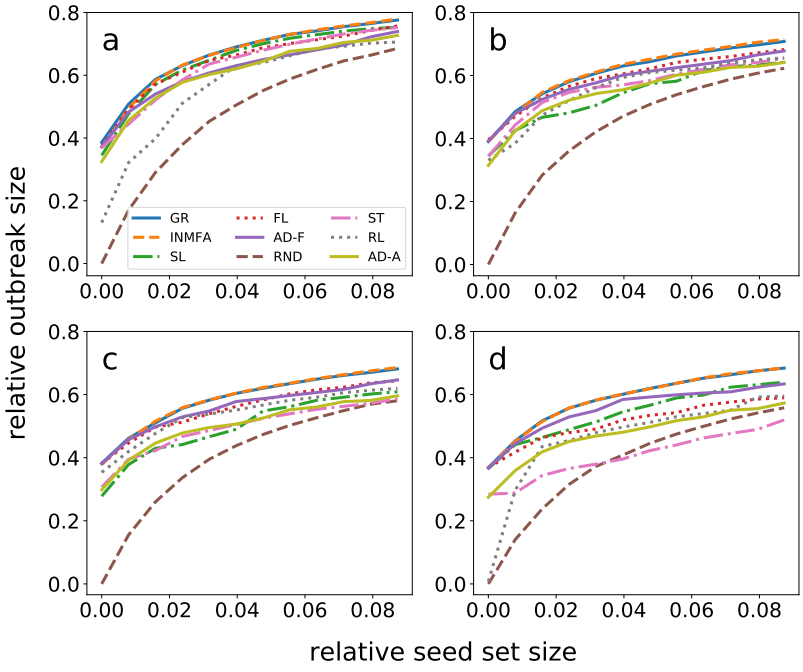

Figure C.28: Same as Fig. C.14, but for "High school, 2011" network in supercritical regime.

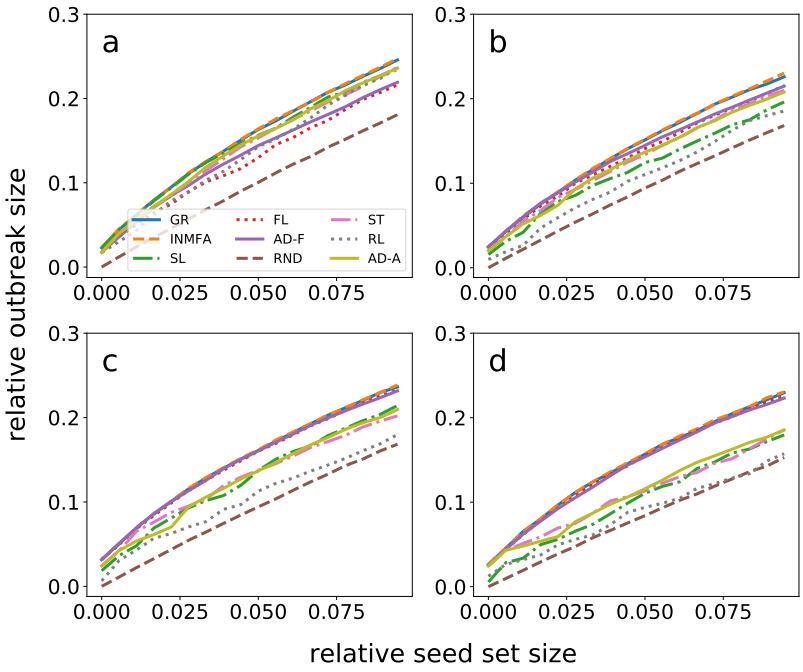

Figure C.29: Same as Fig. C.14, but for "High school, 2012" network in subcritical regime.

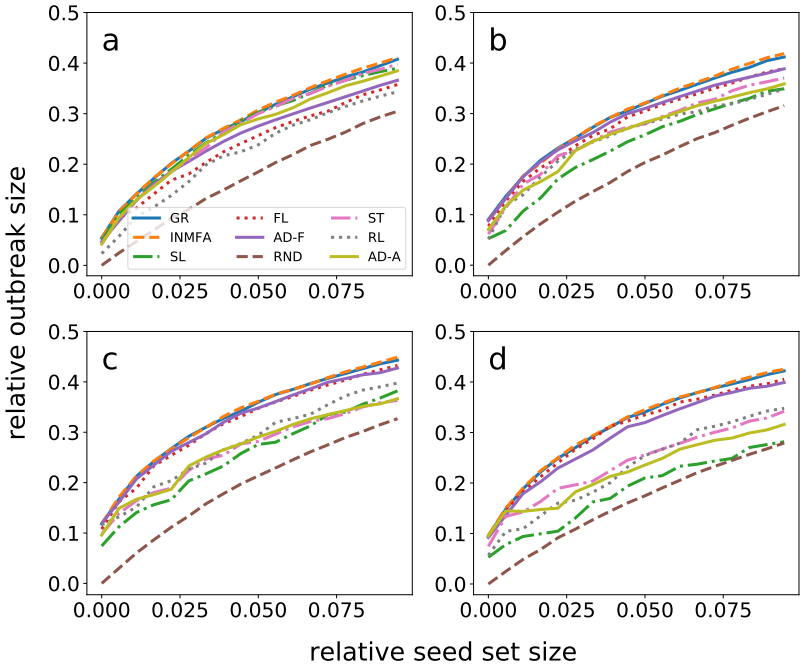

Figure C.30: Same as Fig. C.14, but for "High school, 2012" network in critical regime.

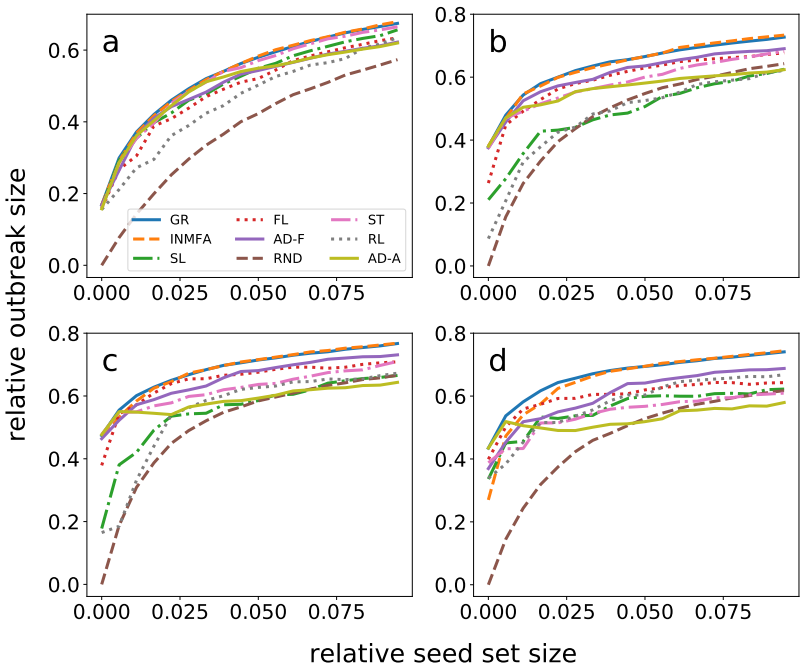

Figure C.31: Same as Fig. C.14, but for "High school, 2012" network in supercritical regime.

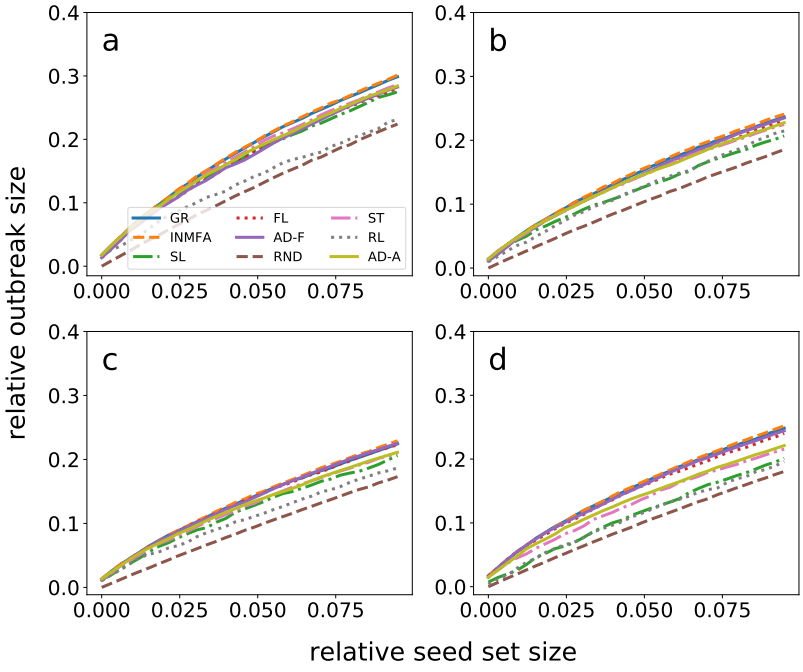

Figure C.32: Same as Fig. C.14, but for "High school, 2013" network in subcritical regime.

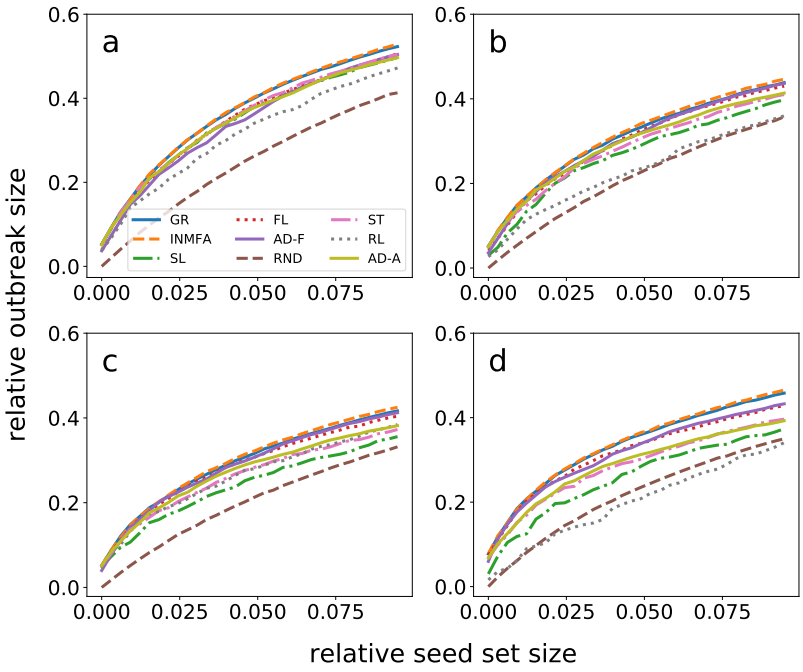

Figure C.33: Same as Fig. C.14, but for "High school, 2013" network in critical regime.

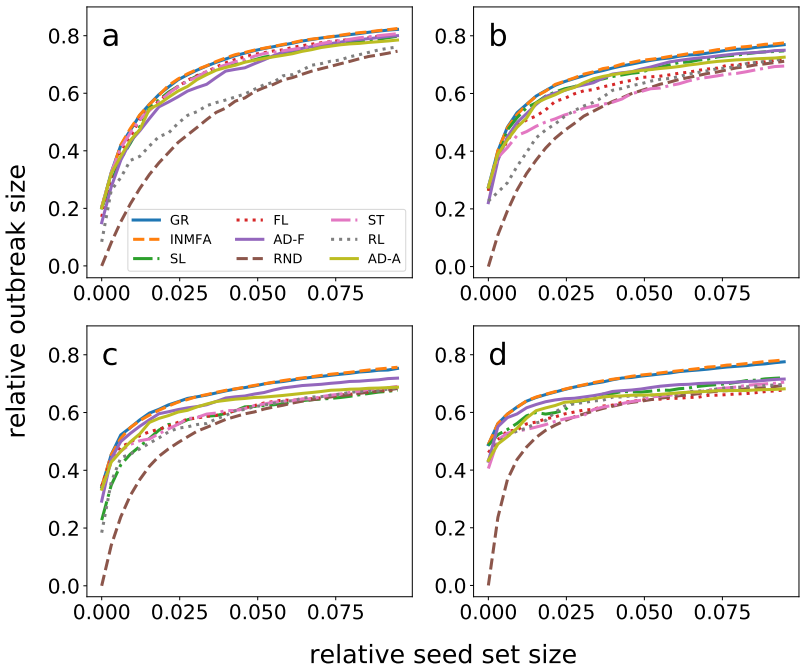

Figure C.34: Same as Fig. C.14, but for "High school, 2013" network in supercritical regime.

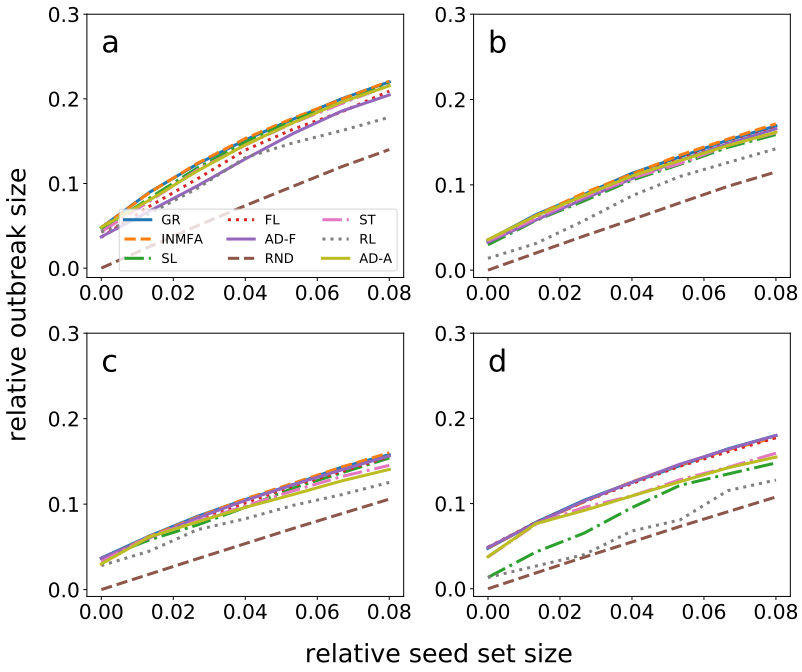

Figure C.35: Same as Fig. C.14, but for "Hospital ward" network in subcritical regime.

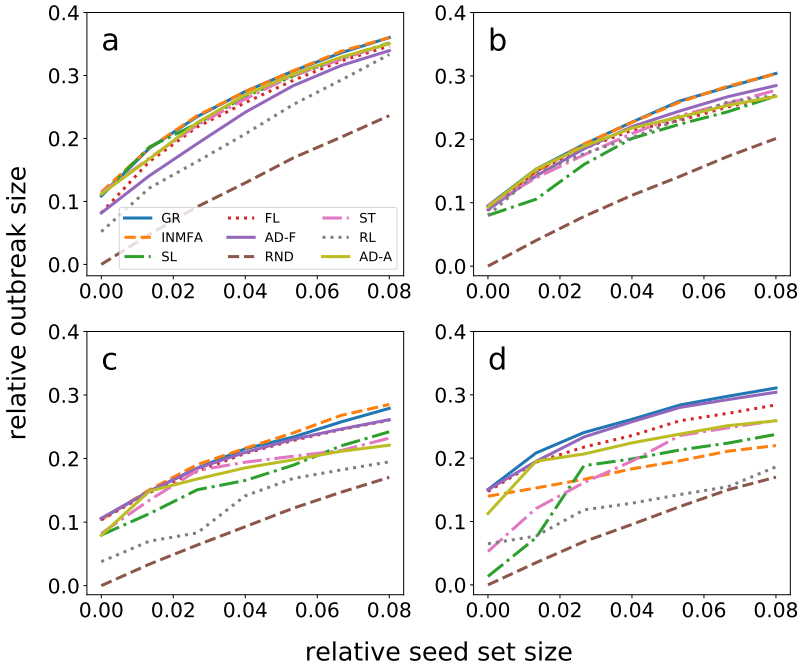

Figure C.36: Same as Fig. C.14, but for "Hospital ward" network in critical regime.

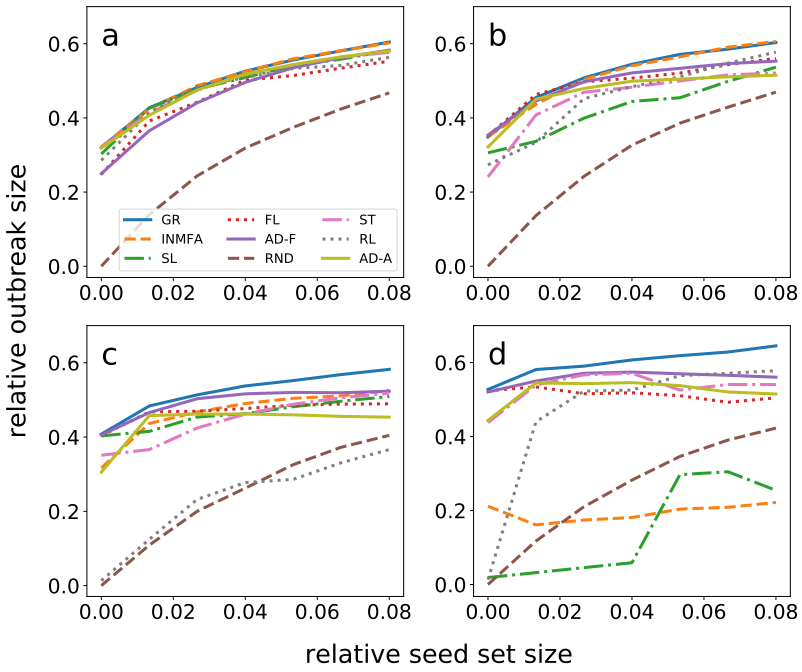

Figure C.37: Same as Fig. C.14, but for "Hospital ward" network in supercritical regime.

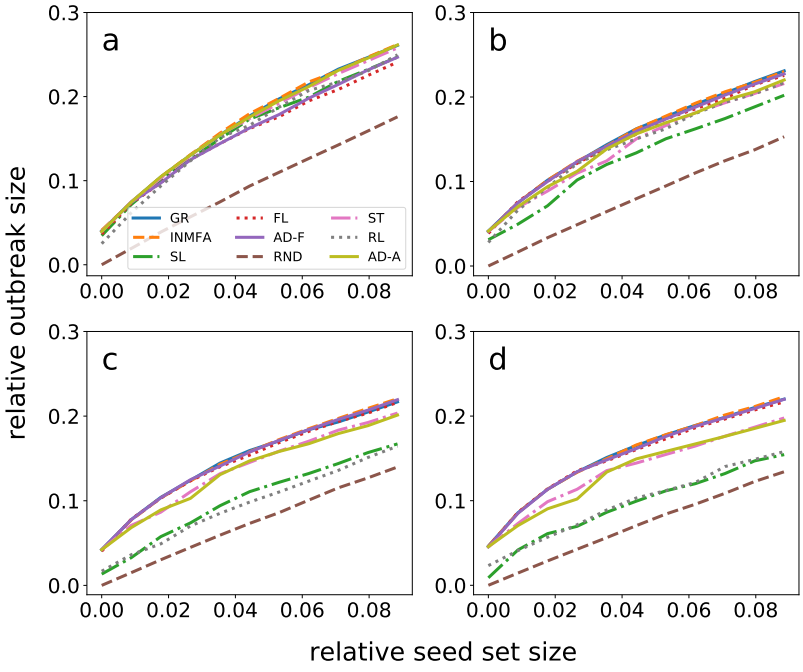

Figure C.38: Same as Fig. C.14, but for "Hypertext, 2009" network in subcritical regime.

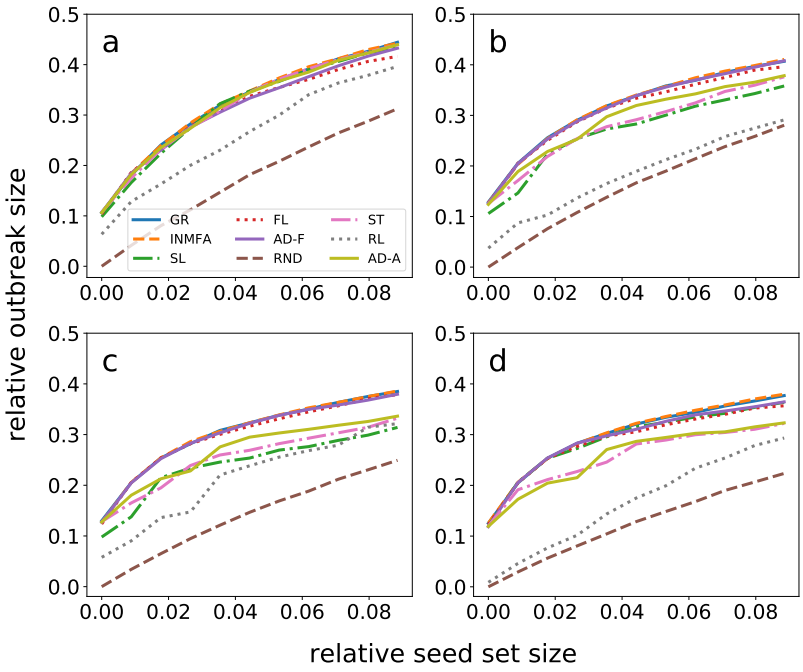

Figure C.39: Same as Fig. C.14, but for "Hypertext, 2009" network in critical regime.

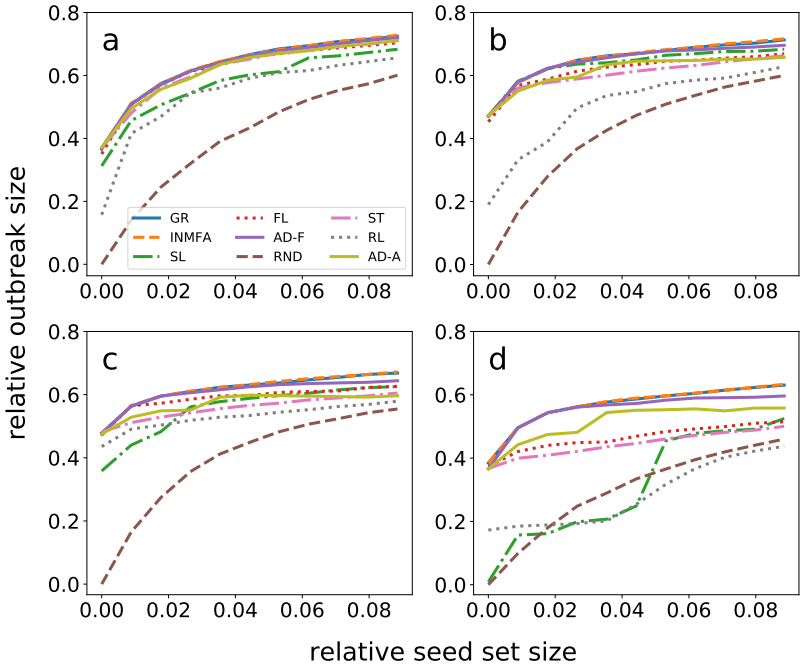

Figure C.40: Same as Fig. C.14, but for "Hypertext, 2009" network in supercritical regime.

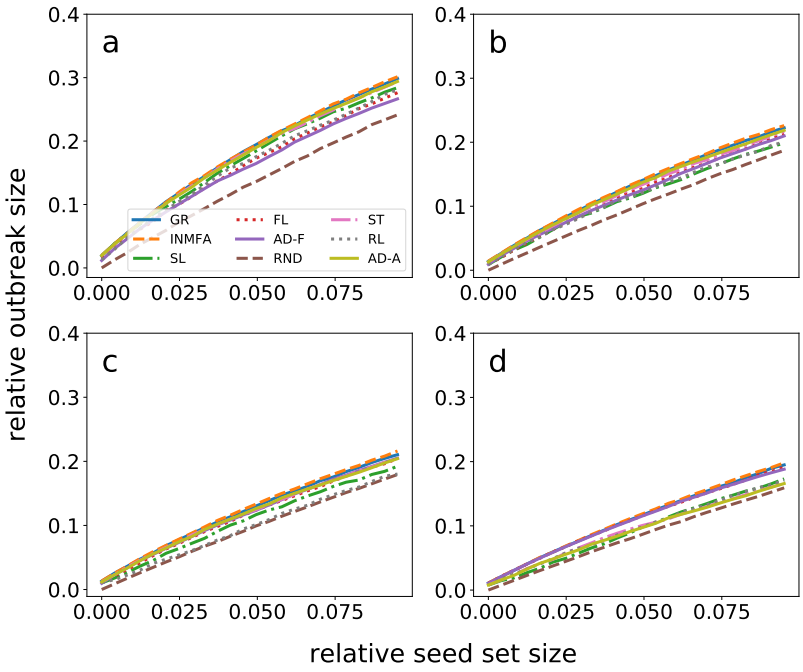

Figure C.41: Same as Fig. C.14, but for "Primary school" network in subcritical regime.

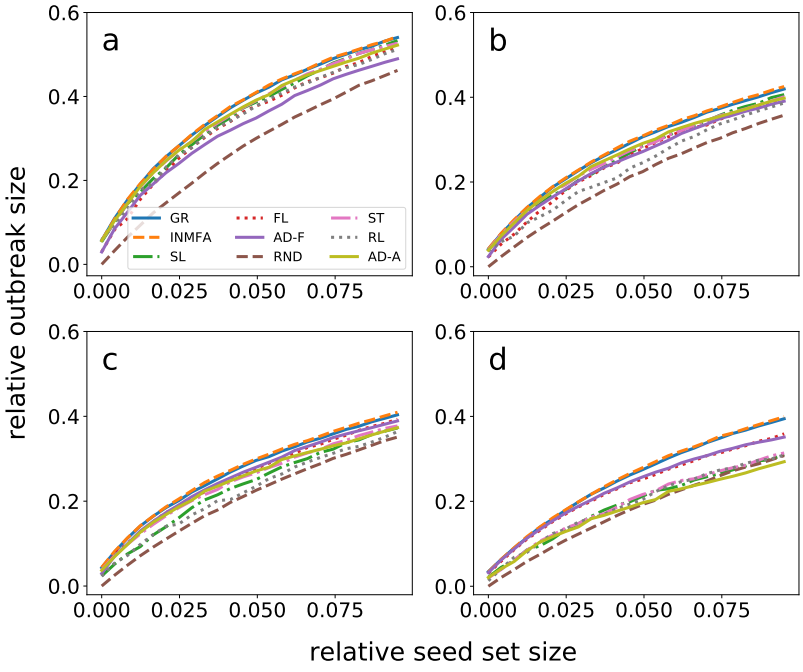

Figure C.42: Same as Fig. C.14, but for "Primary school" network in critical regime.

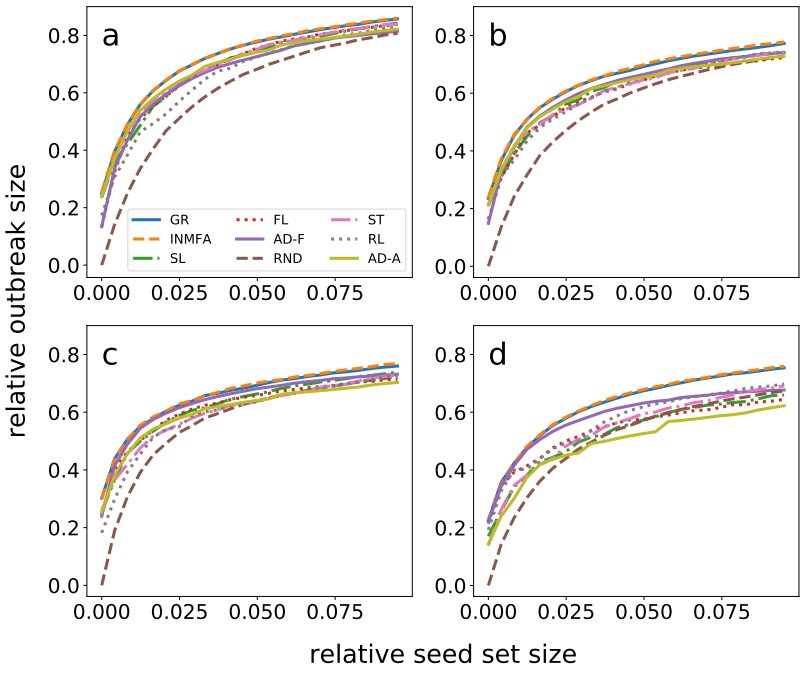

Figure C.43: Same as Fig. C.14, but for "Primary school" network in supercritical regime.

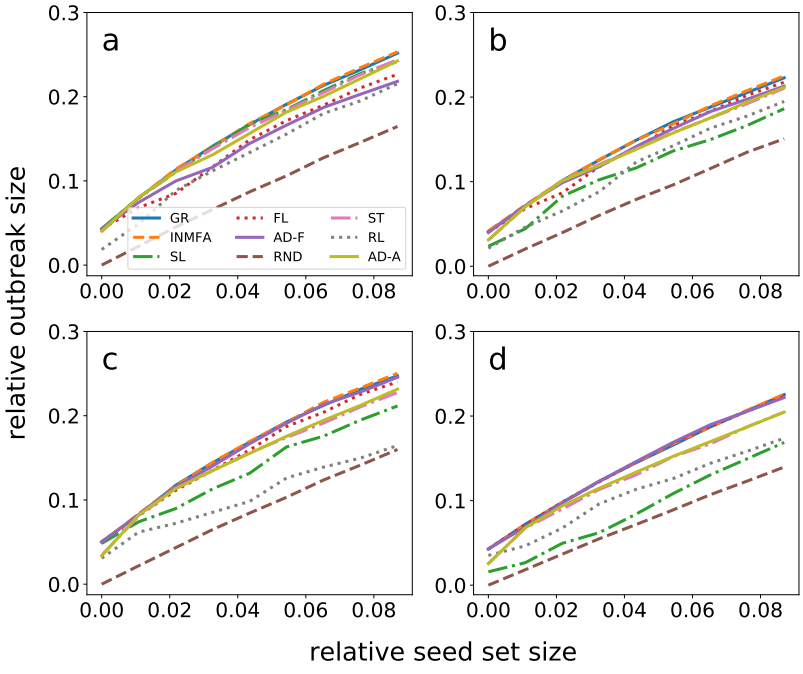

Figure C.44: Same as Fig. C.14, but for "Workplace" network in subcritical regime.

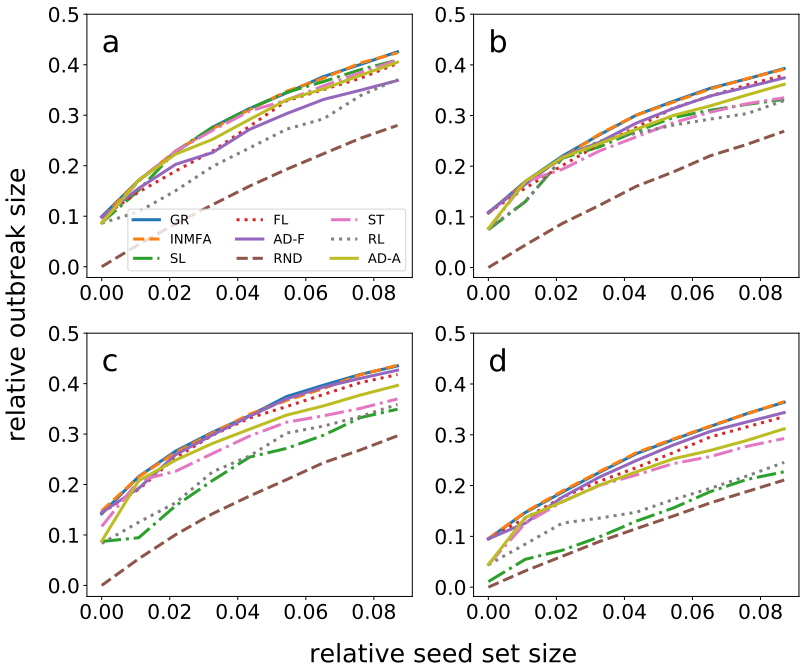

Figure C.45: Same as Fig. C.14, but for "Workplace" network in critical regime.

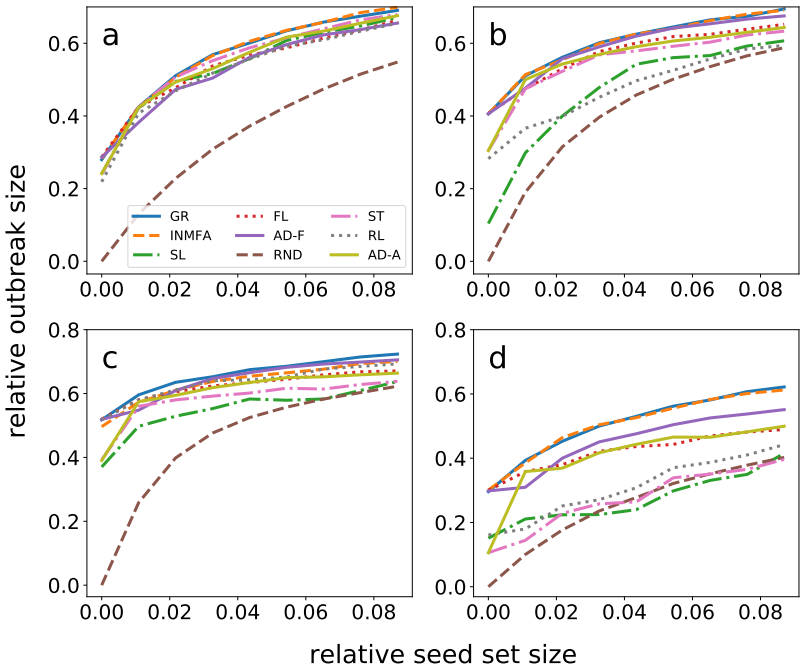

Figure C.46: Same as Fig. C.14, but for "Workplace" network in supercritical regime.

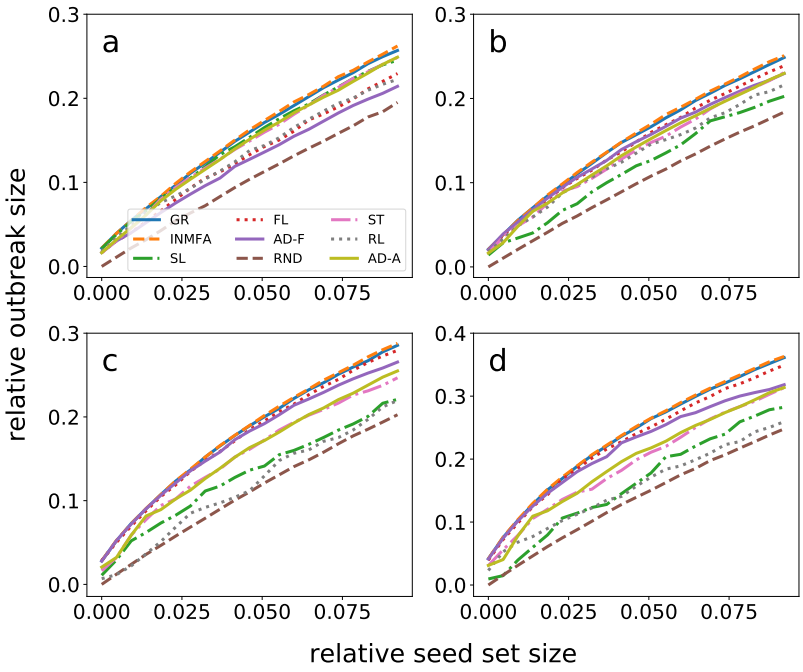

Figure C.47: Same as Fig. C.14, but for "Workplace-2" network in subcritical regime.

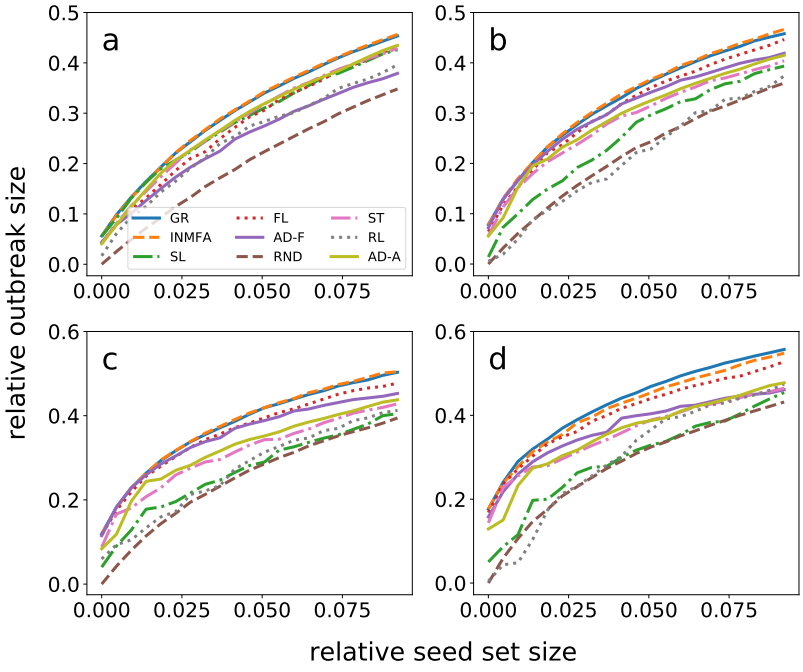

Figure C.48: Same as Fig. C.14, but for "Workplace-2" network in critical regime.

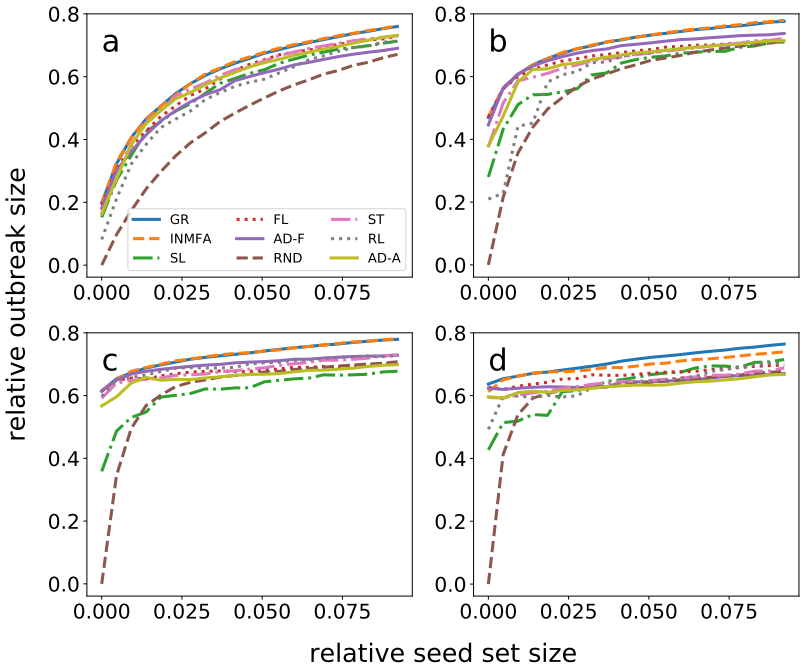

Figure C.49: Same as Fig. C.14, but for "Workplace-2" network in supercritical regime.

References

[1] Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., and Arenas, A. (2003) Self-similar community structure in a network of human interactions. *Physical review E,* **68**(6), 065103.

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

[3] Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Patil, S., Flammini, A., and Menczer, F. (2011) Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th international conference companion on World wide web* pp. 249–252.

[4] Acemoglu, D., Ozdaglar, A., and ParandehGheibi, A. (2010) Spread of (mis) information in social networks. *Games and Economic Behavior,* **70**(2), 194–227.

[5] Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H. E., and Quattrociocchi, W. (2016) The spreading of misinformation online. *Proceedings of the National Academy of Sciences,* **113**(3), 554–559.

[6] Centola, D. (2010) The spread of behavior in an online social network experiment. *science,* **329**(5996), 1194–1197.

[7] Lerman, K. and Ghosh, R. (2010) Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Fourth international AAAI conference on weblogs and social media*.

[8] Notarmuzi, D. and Castellano, C. (2018) Analytical study of quality-biased competition dynamics for memes in social media. *EPL (Europhysics Letters),* **122**(2), 28002.

[9] Kempe, D., Kleinberg, J., and Tardos, É. (2003) Maximizing the spread of influence

through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 137–146.

[10] Lü, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C., and Zhou, T. (2016) Vital nodes identification in complex networks. *Physics Reports,* **650**, 1–63.

[11] Marsden, P. V. (1990) Network data and measurement. *Annual review of sociology,* pp. 435–463.

[12] Dall'Asta, L., Alvarez-Hamelin, I., Barrat, A., Vázquez, A., and Vespignani, A. (2005) Statistical theory of Internet exploration. *Physical Review E,* **71**(3), 036135.

[13] Leskovec, J. and Faloutsos, C. (2006) Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 631–636.

[14] Erkol, Ş. and Yücel, G. (2017) Influence maximization based on partial network structure information: A comparative analysis on seed selection heuristics. *International Journal of Modern Physics C,* **28**(10), 1750122.

[15] Chen, W., Peng, B., Schoenebeck, G., and Tao, B. (2022) Adaptive greedy versus non-adaptive greedy for influence maximization. *Journal of Artificial Intelligence Research,* **74**, 303–351.

[16] Yuan, J. and Tang, S. (2016) No time to observe: Adaptive influence maximization with partial feedback. *arXiv preprint arXiv:1609.00427,*.

[17] Tang, S. and Yuan, J. (2020) Influence maximization with partial feedback. *Operations Research Letters,* **48**(1), 24–28.

[18] Chen, W., Wang, Y., and Yang, S. (2009) Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 199–208.

[19] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007) Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 420–429.

[20] Parmer, T., Rocha, L. M., and Radicchi, F. (2022) Influence maximization in Boolean networks. *Nature Communications,* **13**(1), 3457.

[21] Banerjee, S., Jenamani, M., and Pratihar, D. K. (2020) A survey on influence maximization in a social network. *Knowledge and Information Systems,* **62**, 3417–3455.

[22] Tian, Y. and Lambiotte, R. (2022) Unifying information propagation models on networks and influence maximization. *Physical Review E,* **106**(3), 034316.

[23] Holme, P. and Saramäki, J. (2012) Temporal networks. *Physics reports,* **519**(3), 97–125.

[24] Erkol, Ş., Castellano, C., and Radicchi, F. (2019) Systematic comparison between methods for the detection of influential spreaders in complex networks. *Scientific reports,* **9**(1), 1–11.

[25] Erkol, Ş., Faqeeh, A., and Radicchi, F. (2018) Influence maximization in noisy networks. *EPL (Europhysics Letters),* **123**(5), 58007.

[26] Erkol, Ş., Mazzilli, D., and Radicchi, F. (2020) Influence maximization on temporal networks. *Physical Review E,* **102**(4), 042307.

[27] Erkol, Ş., Mazzilli, D., and Radicchi, F. (2022) Effective submodularity of influence maximization on temporal networks. *Physical Review E,* **106**(3), 034301.

[28] Fournet, J. and Barrat, A. (2014) Contact patterns among high school students. *PloS one,* **9**(9), e107878.

[29] Mastrandrea, R., Fournet, J., and Barrat, A. (2015) Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PloS one,* **10**(9).

[30] Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010) Community structure in time-dependent, multiscale, and multiplex networks. *science,* **328**(5980), 876–878.

[31] Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014) Multilayer networks. *Journal of complex networks,* **2**(3), 203–271.

[32] Pastor-Satorras, R., Castellano, C., Van Mieghem, P., and Vespignani, A. (2015) Epidemic processes in complex networks. *Reviews of modern physics,* **87**(3), 925.

[33] De Domenico, M., Granell, C., Porter, M. A., and Arenas, A. (2016) The physics of spreading processes in multilayer networks. *Nature Physics,* **12**(10), 901–906.

[34] Moreno, Y., Nekovee, M., and Pacheco, A. F. (2004) Dynamics of rumor spreading in complex networks. *Physical review E,* **69**(6), 066130.

[35] Granovetter, M. (1978) Threshold models of collective behavior. *American journal of sociology,* **83**(6), 1420–1443.

[36] Schelling, T. C. (2006) Micromotives and macrobehavior, WW Norton & Company, .

[37] Shakarian, P., Bhatnagar, A., Aleali, A., Shaabani, E., and Guo, R. (2015) The independent cascade and linear threshold models. In *Diffusion in Social Networks* pp. 35–48 Springer.

[38] Domingos, P. and Richardson, M. (2001) Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 57–66.

[39] Grassberger, P. (1983) On the critical behavior of the general epidemic process and dynamical percolation. *Mathematical Biosciences,* **63**(2), 157–172.

[40] Newman, M. and Ziff, R. M. (2000) Efficient Monte Carlo algorithm and high-precision results for percolation. *Physical Review Letters,* **85**(19), 4104.

[41] Radicchi, F. (2015) Predicting percolation thresholds in networks. *Physical Review E,* **91**(1), 010801.

[42] Nguyen, H. T., Thai, M. T., and Dinh, T. N. (2016) Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 international conference on management of data* pp. 695–710.

[43] Tang, Y., Shi, Y., and Xiao, X. (2015) Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data* pp. 1539–1554.

[44] Minutoli, M., Halappanavar, M., Kalyanaraman, A., Sathanur, A., Mcclure, R., and McDermott, J. (2019) Fast and scalable implementations of influence maximization algorithms. In *2019 IEEE International Conference on Cluster Computing (CLUSTER)* IEEE pp. 1–12.

[45] Wang, L., Ma, L., Wang, C., Xie, N.-g., Koh, J. M., and Cheong, K. H. (2021) Identifying influential spreaders in social networks through discrete moth-flame optimization. *IEEE Transactions on Evolutionary Computation,* **25**(6), 1091–1102.

[46] Zareie, A., Sheikhahmadi, A., and Jalili, M. (2020) Identification of influential users in social network using gray wolf optimization algorithm. *Expert Systems with Applications,* **142**, 112971.

[47] Sheikhahmadi, A. and Zareie, A. (2020) Identifying influential spreaders using multiobjective artificial bee colony optimization. *Applied Soft Computing,* **94**, 106436.

[48] Li, Y., Gao, H., Gao, Y., Guo, J., and Wu, W. (2022) A Survey on Influence Maximization: From an ML-Based Combinatorial Optimization. *arXiv preprint arXiv:2211.03074,*.

[49] Ma, L., Shao, Z., Li, X., Lin, Q., Li, J., Leung, V. C., and Nandi, A. K. (2022) Influence maximization in complex networks by using evolutionary deep reinforcement learning. *IEEE Transactions on Emerging Topics in Computational Intelligence,*.

[50] Li, H., Xu, M., Bhowmick, S. S., Sun, C., Jiang, Z., and Cui, J. (2019) Disco: Influence maximization meets network embedding and deep learning. *arXiv preprint arXiv:1906.07378,*.

[51] Lotf, J. J., Azgomi, M. A., and Dishabi, M. R. E. (2022) An improved influence maximization method for social networks based on genetic algorithm. *Physica A: Statistical Mechanics and its Applications,* **586**, 126480.

[52] Fan, C., Zeng, L., Sun, Y., and Liu, Y.-Y. (2020) Finding key players in complex networks through deep reinforcement learning. *Nature machine intelligence,* **2**(6), 317–324.

[53] Schoenebeck, G., Tao, B., and Yu, F.-Y. (2022) Think globally, act locally: On the optimal seeding for nonsubmodular influence maximization. *Information and Computation,* **285**, 104919.

[54] Ren, T., Li, Z., Qi, Y., Zhang, Y., Liu, S., Xu, Y., and Zhou, T. (2020) Identifying vital nodes based on reverse greedy method. *Scientific Reports,* **10**(1), 1–8.

[55] Lin, J., Chen, B.-L., Yang, Z., Liu, J.-G., and Tessone, C. J. (2023) Rank the spreading influence of nodes using dynamic Markov process. *New Journal of Physics,* **25**(2), 023014.

[56] Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. (2010) Identification of influential spreaders in complex networks. *Nature physics,* **6**(11), 888–893.

[57] De Arruda, G. F., Barbieri, A. L., Rodriguez, P. M., Rodrigues, F. A., Moreno, Y., and da Fontoura Costa, L. (2014) Role of centrality for the identification of influential spreaders in complex networks. *Physical Review E,* **90**(3), 032812.

[58] Yang, P.-L., Xu, G.-Q., Yu, Q., and Guo, J.-W. (2020) An adaptive heuristic clustering algorithm for influence maximization in complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science,* **30**(9), 093106.

[59] Wang, M., Li, W., Guo, Y., Peng, X., and Li, Y. (2020) Identifying influential spreaders in complex networks based on improved k-shell method. *Physica A: Statistical Mechanics and its Applications,* **554**, 124229.

[60] Maji, G., Mandal, S., and Sen, S. (2020) A systematic survey on influential spreaders identification in complex networks with a focus on K-shell based techniques. *Expert Systems with Applications,* **161**, 113681.

[61] Basaras, P., Iosifidis, G., Katsaros, D., and Tassiulas, L. (2017) Identifying influential spreaders in complex multilayer networks: A centrality perspective. *IEEE Transactions on Network Science and Engineering,* **6**(1), 31–45.

[62] Katukuri, M. and Jagarapu, M. (2022) CIM: clique-based heuristic for finding influential nodes in multilayer networks. *Applied Intelligence,* pp. 1–12.

[63] Yang, X.-H., Xiong, Z., Ma, F., Chen, X., Ruan, Z., Jiang, P., and Xu, X. (2021) Identifying influential spreaders in complex networks based on network embedding and node local centrality. *Physica A: Statistical Mechanics and its Applications,* **573**, 125971.

[64] Tang, J., Zhang, R., Wang, P., Zhao, Z., Fan, L., and Liu, X. (2020) A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks. *Knowledge-Based Systems,* **187**, 104833.

[65] Patwardhan, S., Radicchi, F., and Fortunato, S. (2022) Influence Maximization: Divide and Conquer. *arXiv preprint arXiv:2210.01203,*.

[66] Yan, X.-L., Cui, Y.-P., and Ni, S.-J. (2020) Identifying influential spreaders in complex networks based on entropy weight method and gravity law. *Chinese physics B,* **29**(4), 048902.

[67] Liu, Y., Zeng, Q., Pan, L., and Tang, M. (2023) Identify Influential Spreaders in Asymmetrically Interacting Multiplex Networks. *IEEE Transactions on Network Science and Engineering,*.

[68] Zhang, X., Xu, L., and Xu, Z. (2022) Influence maximization based on network motifs in mobile social networks. *IEEE Transactions on Network Science and Engineering,* **9**(4), 2353–2363.

[69] Wang, S., Liu, J., and Jin, Y. (2019) Finding influential nodes in multiplex networks using a memetic algorithm. *IEEE transactions on cybernetics,* **51**(2), 900–912.

[70] Rezaei, A. A., Munoz, J., Jalili, M., and Khayyam, H. (2022) Vital Node Identification in Complex Networks Using a Machine Learning-Based Approach. *arXiv preprint arXiv:2202.06229,*.

[71] Maji, G., Namtirtha, A., Dutta, A., and Malta, M. C. (2020) Influential spreaders identification in complex networks with improved k-shell hybrid method. *Expert Systems with Applications,* **144**, 113092.

[72] Li, Z. and Huang, X. (2022) Identifying influential spreaders by gravity model considering multi-characteristics of nodes. *Scientific Reports,* **12**(1), 9879.

[73] Li, H., Shang, Q., and Deng, Y. (2021) A generalized gravity model for influential spreaders identification in complex networks. *Chaos, Solitons & Fractals,* **143**, 110456.

[74] Xie, M., Zhan, X.-X., Liu, C., and Zhang, Z.-K. (2022) Influence Maximization in Hypergraphs. *arXiv preprint arXiv:2206.01394,*.

[75] Li, X., Zhang, X., Zhao, C., Yi, D., and Li, G. (2020) Identifying highly influential nodes in multilayer networks based on global propagation. *Chaos: An Interdisciplinary Journal of Nonlinear Science,* **30**(6), 061107.

[76] Pei, S., Wang, J., Morone, F., and Makse, H. A. (2020) Influencer identification in dynamical complex systems. *Journal of complex networks,* **8**(2), cnz029.

[77] Kim, D. A., Hwong, A. R., Stafford, D., Hughes, D. A., O'Malley, A. J., Fowler, J. H., and Christakis, N. A. (2015) Social network targeting to maximise population behaviour change: a cluster randomised controlled trial. *The Lancet,* **386**(9989), 145–153.

[78] Wang, Y., Wang, S., and Deng, Y. (2019) A modified efficiency centrality to identify influential nodes in weighted networks. *Pramana,* **92**(4), 68.

[79] Oettershagen, L., Mutzel, P., and Kriege, N. M. (2022) Temporal Walk Centrality: Ranking Nodes in Evolving Networks. In *Proceedings of the ACM Web Conference 2022* pp. 1640–1650.

[80] Freeman, L. C. (1977) A set of measures of centrality based on betweenness. *Sociometry,* pp. 35–41.

[81] Sabidussi, G. (1966) The centrality index of a graph. *Psychometrika,* **31**(4), 581–603.

[82] Bonacich, P. (1972) Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology,* **2**(1), 113–120.

[83] Katz, L. (1953) A new status index derived from sociometric analysis. *Psychometrika,* **18**(1), 39–43.

[84] Brin, S. and Page, L. (1998) The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems,* **30**(1-7), 107–117.

[85] Martin, T., Zhang, X., and Newman, M. E. (2014) Localization and centrality in networks. *Physical review E,* **90**(5), 052808.

[86] Rombach, M. P., Porter, M. A., Fowler, J. H., and Mucha, P. J. (2014) Core-periphery structure in networks. *SIAM Journal on Applied mathematics,* **74**(1), 167–190.

[87] Zhang, X., Martin, T., and Newman, M. E. (2015) Identification of core-periphery structure in networks. *Physical Review E,* **91**(3), 032803.

[88] Holme, P. (2005) Core-periphery organization of complex networks. *Physical Review E,* **72**(4), 046111.

[89] Chen, D., Lü, L., Shang, M.-S., Zhang, Y.-C., and Zhou, T. (2012) Identifying influential nodes in complex networks. *Physica a: Statistical mechanics and its applications,* **391**(4), 1777–1787.

[90] Hirsch, J. E. (2005) An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences,* **102**(46), 16569–16572.

[91] Lü, L., Zhou, T., Zhang, Q.-M., and Stanley, H. E. (2016) The H-index of a network node and its relation to degree and coreness. *Nature communications,* **7**(1), 1–7.

[92] Morone, F. and Makse, H. A. (2015) Influence maximization in complex networks through optimal percolation. *Nature,* **524**(7563), 65–68.

[93] Radicchi, F. and Castellano, C. (2017) Fundamental difference between superblockers and superspreaders in networks. *Physical Review E,* **95**(1), 012318.

[94] Zdeborová, L., Zhang, P., and Zhou, H.-J. (2016) Fast and simple decycling and dismantling of networks. *Scientific reports,* **6**(1), 1–6.

[95] Clusella, P., Grassberger, P., Pérez-Reche, F. J., and Politi, A. (2016) Immunization and targeted destruction of networks using explosive percolation. *Physical review letters,* **117**(20), 208301.

[96] Achlioptas, D., D'Souza, R. M., and Spencer, J. (2009) Explosive percolation in random networks. *science,* **323**(5920), 1453–1455.

[97] Goyal, A., Lu, W., and Lakshmanan, L. V. (2011) Celf++ optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web* pp. 47–48.

[98] Tang, Y., Xiao, X., and Shi, Y. (2014) Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* pp. 75–86.

[99] Lin, J.-H., Guo, Q., Dong, W.-Z., Tang, L.-Y., and Liu, J.-G. (2014) Identifying the node spreading influence with largest k-core values. *Physics Letters A,* **378**(45), 3279–3284.

[100] Gómez, V., Kaltenbrunner, A., and López, V. (2008) Statistical analysis of the social network and discussion threads in slashdot. In *Proceedings of the 17th international conference on World Wide Web* ACM pp. 645–654.

[101] Kunegis, J. (2013) KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion* pp. 1343–1350.

[102] Ripeanu, M., Foster, I., and Iamnitchi, A. (2002) Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *arXiv preprint cs/0209028,*.

[103] Leskovec, J., Kleinberg, J., and Faloutsos, C. (2007) Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD),* **1**(1), 2.

[104] Richardson, M., Agrawal, R., and Domingos, P. (2003) Trust management for the semantic web. In *The Semantic Web-ISWC 2003* pp. 351–368 Springer.

[105] McAuley, J. and Leskovec, J. (2012) Learning to Discover Social Circles in Ego Networks. In *Advances in Neural Information Processing Systems* pp. 548–556.

[106] Cho, E., Myers, S. A., and Leskovec, J. (2011) Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* ACM pp. 1082–1090.

[107] Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics,* **6**(1), 29–123.

[108] Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007) The dynamics of viral marketing. *ACM Transactions on the Web (TWEB),* **1**(1), 5.

[109] Yang, J. and Leskovec, J. (2012) Defining and Evaluating Network Communities based on Ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics* ACM p. 3.

[110] Newman, M. E. (2018) Network structure from rich but noisy data. *Nature Physics,* **14**(6), 542–545.

[111] Saito, K., Nakano, R., and Kimura, M. (2008) Prediction of information diffusion probabilities for independent cascade model. In *International conference on knowledge-based and intelligent information and engineering systems* Springer pp. 67–75.

[112] Ou, J., Buskens, V., Van De Rijt, A., and Panja, D. (2022) Influence maximization under limited network information: Seeding high-degree neighbors. *Journal of Physics: Complexity,* **3**(4), 045004.

[113] Colizza, V., Pastor-Satorras, R., and Vespignani, A. (2007) Reaction–diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics,* **3**(4), 276–282.

[114] Radicchi, F. (2011) Who is the best player ever? A complex network analysis of the history of professional tennis. *PloS one,* **6**(2), e17249.

[115] Opsahl, T., Agneessens, F., and Skvoretz, J. (2010) Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks,* **32**(3), 245–251.

[116] Opsahl, T. and Panzarasa, P. (2009) Clustering in weighted networks. *Social networks,* **31**(2), 155–163.

[117] Adamic, L. A. and Glance, N. (2005) The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery* ACM pp. 36–43.

[118] Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., and Alon, U. (2004) Superfamilies of evolved and designed networks. *Science,* **303**(5663), 1538–1542.

[119] Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., et al. (2003) Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic acids research,* **31**(9), 2443–2450.

[120] Faqeeh, A., Melnik, S., Colomer-de Simón, P., and Gleeson, J. P. (2016) Emergence of coexisting percolating clusters in networks. *Physical Review E,* **93**(6), 062308.

[121] Prakash, B. A., Tong, H., Valler, N., Faloutsos, M., and Faloutsos, C. (2010) Virus propagation on time-varying networks: Theory and immunization algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* Springer pp. 99–114.

[122] Karsai, M., Kivelä, M., Pan, R. K., Kaski, K., Kertész, J., Barabási, A.-L., and Saramäki, J. (2011) Small but slow world: How network topology and burstiness slow down spreading. *Physical Review E,* **83**(2), 025102.

[123] Perra, N., Gonçalves, B., Pastor-Satorras, R., and Vespignani, A. (2012) Activity driven modeling of time varying networks. *Scientific reports,* **2**(1), 1–7.

[124] Valdano, E., Ferreri, L., Poletto, C., and Colizza, V. (2015) Analytical computation of the epidemic threshold on temporal networks. *Physical Review X,* **5**(2), 021005.

[125] Osawa, S. and Murata, T. (2015) Selecting seed nodes for influence maximization in dynamic networks. In *Complex Networks VI* pp. 91–98 Springer.

[126] Michalski, R., Kajdanowicz, T., Bródka, P., and Kazienko, P. (2014) Seed selection for spread of influence in social networks: Temporal vs. static approach. *New Generation Computing,* **32**(3), 213–235.

[127] Murata, T. and Koga, H. (2018) Extended methods for influence maximization in dynamic networks. *Computational social networks,* **5**(1), 1–21.

[128] Han, M., Yan, M., Cai, Z., Li, Y., Cai, X., and Yu, J. (2017) Influence maximization by probing partial communities in dynamic online social networks. *Transactions on Emerging Telecommunications Technologies,* **28**(4), e3054.

[129] Zhuang, H., Sun, Y., Tang, J., Zhang, J., and Sun, X. (2013) Influence maximization in dynamic social networks. In *2013 IEEE 13th International Conference on Data Mining* IEEE pp. 1313–1318.

[130] Gayraud, N. T., Pitoura, E., and Tsaparas, P. (2015) Diffusion maximization in evolving social networks. In *Proceedings of the 2015 acm on conference on online social networks* pp. 125–135.

[131] Gemmetto, V., Barrat, A., and Cattuto, C. (2014) Mitigation of infectious disease at school: targeted class closure vs school closure. *BMC infectious diseases,* **14**(1), 695.

[132] Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., et al. (2011) High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one,* **6**(8).

[133] Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.-F., and Van den Broeck, W. (2011) What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of theoretical biology,* **271**(1), 166–180.

[134] Vanhems, P., Barrat, A., Cattuto, C., Pinton, J.-F., Khanafer, N., Régis, C., Kim, B.-a., Comte, B., and Voirin, N. (2013) Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PloS one,* **8**(9).

[135] Paranjape, A., Benson, A. R., and Leskovec, J. (2017) Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* pp. 601–610.

[136] Génois, M., Vestergaard, C. L., Fournet, J., Panisson, A., Bonmarin, I., and Barrat, A. (2015) Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science,* **3**(3), 326–347.

[137] Génois, M. and Barrat, A. (2018) Can co-location be used as a proxy for face-to-face contacts?. *EPJ Data Science,* **7**(1), 11.

[138] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978) An analysis of approxima-

tions for maximizing submodular set functions—I. *Mathematical programming,* **14**(1), 265–294.

[139] Hu, Y., Ji, S., Jin, Y., Feng, L., Stanley, H. E., and Havlin, S. (2018) Local structure can identify and quantify influential global spreaders in large scale social networks. *Proceedings of the National Academy of Sciences,* **115**(29), 7468–7472.

[140] Erdős, P., Rényi, A., et al. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci,* **5**(1), 17–60.

[141] Santiago, R. and Yoshida, Y. (2020) Weakly submodular function maximization using local submodularity ratio. *arXiv preprint arXiv:2004.14650,*.

[142] Girvan, M. and Newman, M. E. (2002) Community structure in social and biological networks. *Proceedings of the National Academy of Sciences,* **99**(12), 7821–7826.

[143] Ulanowicz, R., Bondavalli, C., and Egnotovich, M. (1998) Network analysis of trophic dynamics in south florida ecosystem, FY 97: The florida bay ecosystem. *Annual Report to the United States Geological Service Biological Resources Division Ref. No.[UMCES] CBL,* pp. 98–123.

[144] Michalski, R., Palus, S., and Kazienko, P. (2011) Matching Organizational Structure and Social Network Extracted from Email Communication. In *Lecture Notes in Business Information Processing* Springer Berlin Heidelberg Vol. 87, pp. 197–206.

[145] Martinez, N. D. (1991) Artifacts or attributes? Effects of resolution on the Little Rock Lake food web. *Ecological Monographs,* pp. 367–392.

[146] Gleiser, P. M. and Danon, L. (2003) Community structure in jazz. *Advances in complex systems,* **6**(04), 565–573.

[147] Newman, M. E. (2006) Finding community structure in networks using the eigenvectors of matrices. *Physical review E,* **74**(3), 036104.

[148] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network motifs: simple building blocks of complex networks. *Science,* **298**(5594), 824–827.

[149] Traud, A. L., Mucha, P. J., and Porter, M. A. (Aug, 2012) Social structure of Facebook networks. *Phys. A,* **391**(16), 4165–4180.

[150] Traud, A. L., Kelsic, E. D., Mucha, P. J., and Porter, M. A. (2011) Comparing Community Structure to Characteristics in Online Collegiate Social Networks. *SIAM Rev.,* **53**(3), 526–543.

[151] Rossi, R. A. and Ahmed, N. K. (2015) The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[152] Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., and Denk, W. (2013) Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature,* **500**(7461), 168.

[153] Clauset, A., Tucker, E., and Sainz, M. (2016) The Colorado Index of Complex Networks.

[154] Moody, J. (2001) Peer influence groups: identifying dense clusters in large networks. *Social Networks,* **23**(4), 261–283.

[155] Kumar, S., Spezzano, F., Subrahmanian, V., and Faloutsos, C. (2016) Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on* IEEE pp. 221–230.

[156] Kumar, S., Hooi, B., Makhija, D., Kumar, M., Faloutsos, C., and Subrahmanian, V. (2018) Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the*

Eleventh ACM International Conference on Web Search and Data Mining ACM pp. 333–341.

[157] Leskovec, J. and Krevl, A. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data (June, 2014).

[158] Newman, M. E. (2001) The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences,* **98**(2), 404–409.

[159] Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G., Wu, G., Matthews, L., et al. (2005) Reactome: a knowledgebase of biological pathways. *Nucleic acids research,* **33**(suppl 1), D428–D432.

[160] Šubelj, L. and Bajec, M. (2012) Software systems through complex networks science: Review, analysis and applications. In *Proceedings of the First International Workshop on Software Mining* ACM pp. 9–16.

[161] Leskovec, J., Kleinberg, J., and Faloutsos, C. (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* ACM pp. 177–187.

[162] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010) Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems* ACM pp. 1361–1370.

[163] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010) Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web* ACM pp. 641–650.

[164] Boguñá, M., Pastor-Satorras, R., Díaz-Guilera, A., and Arenas, A. (2004) Models of social networks based on social distance attachment. *Physical Review E,* **70**(5), 056122.

[165] Ley, M. (2002) The DBLP computer science bibliography: Evolution, research issues, perspectives. In *String Processing and Information Retrieval* Springer pp. 1–10.

[166] Alberich, R., Miro-Julia, J., and Rosselló, F. (2002) Marvel Universe looks almost like a real social network. *arXiv preprint cond-mat/0202174,*.

[167] Kiss, G. R., Armstrong, C., Milroy, R., and Piper, J. (1973) An associative thesaurus of English and its computer analysis. *The computer and literary studies,* pp. 153–165.

[168] Šubelj, L. and Bajec, M. (2013) Model of complex networks based on citation dynamics. In *Proceedings of the 22nd international conference on World Wide Web companion* International World Wide Web Conferences Steering Committee pp. 527–530.

Curriculum Vita

Şirag Erkol

Luddy School of Informatics Computing and Engineering

Indiana University

EDUCATION

- Ph.D. in Informatics, Complex Networks and Systems track, Indiana University
  08/2017 - 05/2023
- M.S. in Informatics, Indiana University
  08/2017 - 05/2021
- M.S. in Industrial Engineering, Boğaziçi University
  09/2014 - 07/2016
- B.S. in Industrial Engineering, Boğaziçi University
  09/2010 - 06/2014

EXPERIENCE

- Research & Teaching Assistant, Indiana University
  08/2017 - 05/2023
- Research Scientist Intern, Pacific Northwest National Laboratory
  06/2021 - 08/2021
- Research & Teaching Assistant, Boğaziçi University
  10/2014 - 07/2017

HONORS & AWARDS

- IU Luddy Summer Research Award (2022)

PUBLICATIONS

- Consistency pays off in science

  Ş. Erkol, S. Sikdar, F. Radicchi, and S. Fortunato, *Quantitative Science Studies* (2023)

- Effective submodularity of influence maximization on temporal networks

  Ş. Erkol, D. Mazzilli and F. Radicchi, *Physical Review E*, 106, 034301 (2022)

- Who is the best coach of all time? A network-based assessment of the career performance of professional sports coaches

  Ş. Erkol and F. Radicchi, *Journal of Complex Networks*, 9, cnab012 (2021)

- Influence maximization on temporal networks

  Ş. Erkol, D. Mazzilli and F. Radicchi, *Physical Review E*, 102, 042307 (2020)

- Systematic comparison between methods for the detection of influential spreaders in complex networks

  Ş. Erkol, C. Castellano and F. Radicchi, *Scientific Reports*, 9, 15095 (2019)

- Influence maximization in noisy networks

  Ş. Erkol, A. Faqeeh and F. Radicchi, *EPL*, 123, 58007 (2018)

- Influence Maximization Based on Partial Network Structure Information: A Comparative Analysis on Seed Selection Heuristics

  Ş. Erkol and G. Yücel, *International Journal of Modern Physics C*, 28, 1750122 (2017)

TALKS

- Effective submodularity of influence maximization on temporal networks, NetSci 2022, Online, July 2022

- Influence maximization on temporal networks, Networks2021, Online, July 2021

- Influence maximization on temporal networks, University of Limerick, Online, March 2021

LANGUAGES

- Armenian (native), Turkish (native), English (fluent), French (beginner), Spanish (beginner)