Containing a spread through sequential learning: to exploit or to explore?

Anonymous authors
Paper under double-blind review

Abstract

The spread of an undesirable contact process, such as an infectious disease (e.g. COVID-19), is contained through testing and isolation of infected nodes. The temporal and spatial evolution of the process (along with containment through isolation) render such detection as fundamentally different from active search detection strategies. In this work, through an active learning approach, we design testing and isolation strategies to contain the spread and minimize the cumulative infections under a given test budget. We prove that the objective can be optimized, with performance guarantees, by greedily selecting the nodes to test. We further design reward-based methodologies that effectively minimize an upper bound on the cumulative infections and are computationally more tractable in large networks. These policies, however, need knowledge about the nodes' infection probabilities which are dynamically changing and have to be learned by sequential testing. We develop a message-passing framework for this purpose and, building on that, show novel tradeoffs between exploitation of knowledge through reward-based heuristics and exploration of the unknown through a carefully designed probabilistic testing. The tradeoffs are fundamentally distinct from the classical counterparts under active search or multi-armed bandit problems (MABs). We provably show the necessity of exploration in a stylized network and show through simulations that exploration can outperform exploitation in various synthetic and real-data networks depending on the parameters of the network and the spread.

1 Introduction

We consider learning and decision making in networked systems for processes that evolve both temporally and spatially. An important example in this class of processes is Covid-19 infection. It evolves in time (e.g. through different stages of the disease for an infected individual) and over a contact network and its spread can be contained by testing and isolation. Public health systems need to judiciously decide who should be tested and isolated in presence of limitations on the number of individuals who can be tested and isolated on a given day.

Most existing works on this topic have investigated the spread of COVID-19 through dynamic systems such SIR models and their variants [1, 2, 3, 4, 5, 6]. These models are made more complex to fit the real data in [7, 8, 9, 10, 11, 12]. Estimation of the model parameters by

learning-based methods are considered and verified by real data in [13, 14, 15, 16, 17, 18]. Other attributes such as lockdown policy [19], multi-wave prediction [20], herd immunity threshold [21] are also considered by data-driven experiments. These works mostly focus on the estimation of model parameters thorough real data, and aim to make a more accurate prediction of the spread. None of them, however, consider testing and isolation policies. Our work complements these investigations by designing sequential testing and isolation policies in order to minimize the cumulative infections. For this purpose, we have assumed full statistical knowledge of the spread model and the underlying contact network and we are not concerned with prediction and estimation of model parameters.

Designing optimal testing and control policies in dynamic networked systems often involves computational challenges. These challenges have been alleviated in control literature by capturing the spread through differential equations [22, 23, 24, 25, 26]. The differential equations rely on classical mean-field approximations, considering neighbors of each node as "socially averaged hypothetical neighbors". Refinements of the mean-field approximations such as pair approximation [27], degree-based approximation [28], meta-population approximation [29] etc, all resort to some form of averaging of neighborhoods or more generally groups of nodes. The averaging does not capture the heterogeneity of a real-world complex social network and in effect disregards the contact network topology. But, in practice the contact network topologies are often fully or partially known and therefore testing and control strategies must exploit topological information to control the spread. For example, the most known and practical testing and control policy, the contact tracing (and its variants) [30, 31, 32, 33, 34], relies on the knowledge of the network topology, and therefore does not lend itself to mean-field analysis. Our proposed framework considers the SIR evolution of the disease for each node but also spread through a given network.

The following challenges arise in the design of intelligent testing strategies if one seeks to exploit the spatio-temporal evolution of the disease process and comply with limited testing budget. Observing the state of a node at time t will provide information about the state of (i) the node in time $t+1$ and (ii) the neighbors of the node at time $t, t+1, \ldots$. This is due to the inherent correlation that exists between states of neighboring nodes because an infectious disease spreads through contact. Thus, testing has a dual role. It has to both detect/isolate infected nodes and learn the spread in various localities of the network. The spread can often be silent: an undetected node (that may not be particularly likely to be infected based on previous observations) can infect its neighbors. Thus, testing nodes that do not necessarily appear to be infected may lead to timely discovery of even larger clusters of infected nodes waiting to explode. In other words, there is *an intrinsic tradeoff between exploitation of knowledge vs. exploration of the unknown*. Exploration vs. exploitation tradeoffs were originally studied in classical multi-armed bandit (MAB) problems where there is the notion of a single optimal arm that can be found by repeating a set of fixed actions [35, 36, 37]. MAB testing strategies have also been designed for exploring partially observable networks [38]. Our problem differs from what is mainly studied in the MAB literature because (i) the number of arms (potential infected nodes) is time-variant and actions cannot be repeated; (ii) the exploration vs. exploitation tradeoff in our context arises due to lack of knowledge about the time-evolving set of infected nodes, rather than lack of knowledge about the network or the process model and its parameters.

Note that contact tracing policies are in a sense exploitation policies: upon finding positive nodes, they exploit that knowledge and trace the contacts. While relatively practical, they have two main shortcomings: (i) They are not able to prioritize nodes based on their likelihood of being infected (beyond the coarse notion of contact or lack thereof). (ii) they do not incorporate any type of exploration. In contrast, our work provides a probabilistic framework to not only allow for exploitation in a fundamental manner but also to incorporate exploration in order to minimize the number of infections.

Finally, our problem is also related to active search in graphs where the goal is to test/search for a set of (fixed) target nodes under a set of given (static) similarity values between pairs of nodes [39, 40, 41, 42]. But the target nodes in these works are assumed fixed, whereas the target is dynamic in our setting because the infection spreads over time and space (i.e, over the contact network). Thus, a node may need to be tested multiple times. The importance of exploitation/exploration is also known, implicitly and/or explicitly, in various reinforcement learning literature [43, 44, 45].

To the best of our knowledge, there are only two papers that are related to the particular problem of testing and isolation as we consider in this work [46, 47]. Through a theoretical approach, [46] models the problem as a partially observable Markov decision process (POMDP). An optimal policy can, in principle, be formulated through POMDP, but such strategies are intractable in their general form (and heuristics are often far from optimal) [48, 49]. [46] devises tractable approximate algorithms with a big caveat: In the design, analysis, and evaluation of the proposed algorithms, it is assumed that at each time the process can spread only on a single random edge of the network. This is a very special case that is hard to justify in practice and it is not clear how one could go beyond this assumption. On the other hand, [47] proposes a heuristic by implementing classical learning methods such as Linear support vector machine (SVM) and Polynomial SVM to rank nodes based on a notion of risk score (constructed by real-data) while reserving a portion of the test budget for random testing which can be understood as exploration. No spread model or contact network is assumed. Different from [47], our approach is model-based and we observe novel exploration-exploitation tradeoffs that are not due to lack of knowledge about the model or network, but rather the time-evolving set of unknown infected nodes. We can also utilize knowledge about both the model and the contact network to devise a probabilistic framework for decision making.

Our Contributions In this work, we study a spread process such as Covid-19 and design sequential testing and isolation policies to contain the spread. Our contributions are as follows.

- Formulating the spread process through a compartmental model and a given contact network, we show that the problem of minimizing the total cumulative infections under a given test budget reduces to minimizing a supermodular function expressed in terms of nodes' probabilities of infection and it thus admits a *near-optimal greedy* policy. We further design reward-based algorithms that minimize an upper bound on the cumulative infections and are computationally more tractable in large networks.

- The greedy policy and its reward-based derivatives are applicable if nodes' probabilities of infection were known. However, since the set of infected nodes are unknown, these probabilities are unknown and can only be learned through *sequential testing*. We provide a message-passing framework for sequential estimation of nodes' *posterior* probabilities of infection given the history of test observations.

- We argue that testing has a dual role: (i) discovering and isolating the infected nodes to contain the spread, and (ii) providing more accurate estimates for nodes' infection probabilities which are used for decision making. In this sense, *exploitation* policies in which decision making only targets (i) can be suboptimal. We prove in a stylized network that when the belief about the probabilities is wrong, exploitation can be arbitrarily bad, while a policy that combines exploitation with random testing can contain the spread. This points to novel *exploitation-exploration tradeoffs* that stem from the lack of knowledge about the location of infected nodes, rather than the network or spread process.

- Following these findings, we propose *exploration* policies that test each node probabilistically according to its reward. The core idea is to balance exploitation of knowledge (about the nodes' infection probabilities and the resulting rewards) and exploration of the unknown (to get more accurate estimates of the infection probabilities). Through simulations, we compare the performance of exploration and exploitation policies in several synthetic and real-data networks. In particular, we investigate the role of three parameters on when exploration outperforms exploitation: (i) the *unregulated delay*, i.e., the time period when the disease spreads without intervention; (ii) the *global clustering coefficient* of the network, and (iii) the average *shortest path length* of the network. We show that when the above parameters increase, exploration becomes more beneficial as it provides better estimates of the nodes' probabilities of infection.

2 Modeling

To describe a spread process, we use a *discrete time compartmental model* [50]. Over decades, compartmental models have been key in the study of epidemics and opinion dynamics, albeit often disregarding the network topology. In this work, we capture the spread on a given contact network. For clarity of presentation, we focus on a model for the spread of COVID-19. The ideas can naturally be generalized to other applications. The main notations in the full paper are given in Table 1.

We model the progression of Covid-19 per individual, in time, through four stages or states: *Susceptible (S)*, *Latent (L)*, *Infectious (I)*, and *Recovered (R)*. Per contact, an infectious individual infects a susceptible individual with transmission probability β. An infected individual is initially in the latent state L, subsequently he becomes infectious (state I), finally he recovers (state R). Fig. 1 (left) depicts the evolution. The durations in the latent and infectious states are geometrically distributed, with means $1/\lambda, 1/\gamma$ respectively. We represent the state of node i at time t by random variable $\sigma_i(t)$ and its support set $\mathcal{X} = \{S, L, I, R\}$.

Notations	Definitions
β	transmission probability
$1/\gamma$	mean duration in the latent state
$1/\lambda$	mean duration in the infectious state
$\sigma_i(t)$	state of node i at time t
$\mathcal{G}(t)$	contact network at time t
$\mathcal{V}(t)$	set of nodes at time t
$\mathcal{E}(t)$	set of edges at time t
$N(t)$	cardinality of $\mathcal{V}(t)$
N	$N = N(0)$
$\partial_i(t)$	neighbors of node i at time t
$\partial_i^+(t)$	$\{i\} \cup \partial_i(t)$
$Y_i(t)$	testing result of node i at time t
$\mathcal{O}(t)$	set of nodes tested at time t
$\underline{Y}(t)$	$\{Y_i(t)\}_{\{i \in \mathcal{O}(t)\}}$
$B(t)$	testing budget at time t
π	a testing and isolation policy
$C^\pi(t)$	cumulative infections at time t
$\mathcal{K}^\pi(t)$	set of nodes tested at time t (under policy π)
$K^\pi(t)$	$K^\pi(t) = \lvert\mathcal{K}^\pi(t)\rvert$
T	time horizon
$\underline{v}_i(t)$	true probability vector of node i
$\underline{u}_i(t)$	prior probability vector of node i
$\underline{w}_i(t)$	posterior probability vector of node i
$\underline{e}_i(t)$	updated posterior probability vector of node i
$r_i(t)$	rewards of selecting node i at time t
$\hat{r}_i(t)$	estimated rewards of node i at time t
$\Psi_i(t)$	$\Psi_i(t) = \mathcal{O}(t) \cap \partial_i^+(t-1)$
$\Phi_i(t)$	$\Phi_i(t) = \{j \lvert j \in \partial_k^+(t-1), k \in \Psi_i(t)\} \backslash \{i\}$
$\theta_i(t)$	$\theta_i(t) = \sigma_i(t)\lvert_{\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}}$
$\zeta_i(t)$	$\zeta_i(t) = \sigma_i(t)\lvert_{\{\underline{Y}(\tau)\}_{\tau=1}^{t}}$

Table 1: Summary of main notations

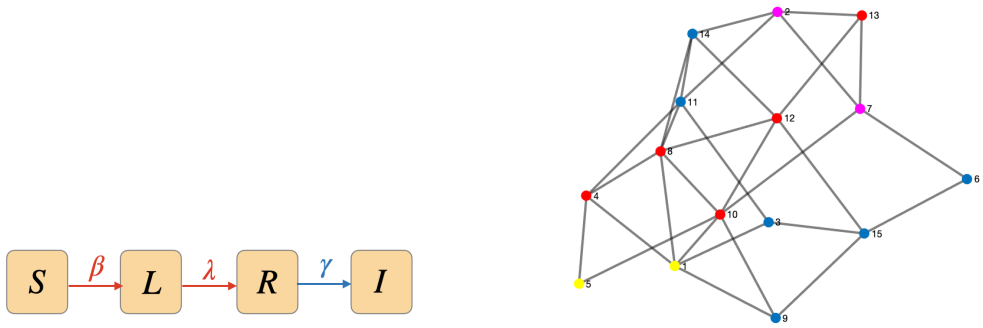

Figure 1: Left: Time evolution of the process per individual nodes. Right: A contact network with nodes in states susceptible (blue), latency (pink), infectious (red), recovered (yellow).

Let $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$ denote the contact network at time t, where $\mathcal{V}(t)$ is the set of nodes/individuals, of cardinality $N(t)$, and $\mathcal{E}(t)$ is the set of edges between the nodes, describing interactions/contacts on day t. Let $\mathcal{V} = \mathcal{V}(0)$, $\mathcal{E} = \mathcal{E}(0)$, $\mathcal{G} = \mathcal{G}(0)$, and $N = N(0)$. The network is time-dependent not only because interactions change on a daily basis, but also because nodes may be tested and isolated. If a node is isolated on day t, we assume that it remains in isolation until he recovers. Therefore, it is removed from the contact network for all subsequent times $t, t+1, \ldots$. Fig. 1 (right) depicts a contact network at a given time t.

Denote the set of neighbors of node i, in day t, by $\partial_i(t)$. The state of each node at time $t+1$ depends on the state of its neighbors $\partial_i(t)$, as well as its own state in day t, as given by the following conditional probability:

$$\Pr\left(\sigma_i(t+1) | \{\sigma_j(t)\}_{j \in \partial_i^+(t)}\right)$$

where $\partial_i^+(t) = \partial_i(t) \cup \{i\}$.

Node i is tested positive on day t if it is in the infectious state (I)[1]. Let $Y_i(t)$ denote the test result:

$$Y_i(t) = \begin{cases} 1 & \sigma_i(t) = I \\ 0 & \sigma_i(t) \in \{S, L, R\}. \end{cases} \tag{1}$$

We do not assume any type of error in testing and $Y_i(t)$ is hence a deterministic function of $\sigma_i(t)$. Let $\mathcal{O}(t)$ be the set of nodes that have been tested (observed) in day t and denote the *network observations* at time t by $\underline{Y}(t) = \{Y_i(t)\}_{i \in \mathcal{O}(t)}$.

Our goal in this paper is to design testing and isolation strategies in order to contain the spread and minimize the cumulative infections. Naturally, testing resources (and hence observations) are often limited and such constraints make decision making challenging. Let $B(t)$ be the maximum expected number of tests that could be performed in day t, called the *testing budget*. $B(t)$ can evolve based on the system necessities, e.g., in contact tracing that is widely deployed for COVID-19, the number of tests is chosen based on the history

[1]We assume that a node in the latent state L is infected, but not infectious. We further assume that latent nodes test negative.

of observations[2]. Also, governments often upgrade testing infrastructure as the number of cases increase. Our framework captures both fixed and time-dependent budget $B(t)$, but we focus on time-dependent $B(t)$ for simulations.

Define the cumulative infections on day t, denoted by $C^\pi(t)$, as the number of nodes who have been infected before and including day t, where π is the testing and isolation policy. Let $K^\pi(t)$ denote the number of tests π performs on day t. Given a large time horizon T, our objective is:

$$\begin{aligned} \min_{\pi} \quad & \mathbb{E}[C^\pi(T)] \\ s.t. \quad & K^\pi(t) \in \mathbb{Z},\ \mathbb{E}[K^\pi(t)] \leq B(t),\ 0 \leq t \leq T-1. \end{aligned} \tag{2}$$

3 Supermodularity

It is complex to solve (2) globally, especially if one seeks to find solutions that are optimal looking into the future. We thus simplify the optimization (2) for policies that are myopic in time as follows. First, note that $C^\pi(T)$ can be re-written as follows through a telescopic sum:

$$C^\pi(T) = \sum_{t=0}^{T-1} C^\pi(t+1) - C^\pi(t). \tag{3}$$

In order to minimize $C^\pi(T)$, we restrict attention to myopic policies that at each time minimize $\mathbb{E}\left[C^\pi(t+1) - C^\pi(t)\right]$. We then show how $\mathbb{E}\left[C^\pi(t+1) - C^\pi(t)\right]$ can be expressed in terms of a supermodular function.

Recall that $\sigma_i(t)$, the state of node i on day t, is a random variable and unknown. For each node i, define a probability vector $\underline{v}_i(t)$ of size $|\mathcal{X}|$, where each coordinate is the probability of the node being in a particular state at the end of time t. The coordinates of $\underline{v}_i(t)$ follow the order (I, L, R, S) and we have

$$\underline{v}_i(t) = \left[v_x^{(i)}(t)\right]_{x\in\mathcal{X}}, \qquad v_x^{(i)}(t) = \Pr\left(\sigma_i(t) = x\right). \tag{4}$$

For example, $v_S^{(i)}(t)$ represents the probability of node i being in state S in time t. We now define $F_i(\mathcal{D}; t)$ to be the *conditional* probability of node i being infected by nodes in $\mathcal{D}$ (for the first time) at day t, as a function of the nodes' states $\{\sigma_i(t)\}_{i\in\mathcal{V}(t)}$. We have

$$F_i(\mathcal{D}; t) = 1_{\{\sigma_i(t)=S\}} \cdot \prod_{j\in\partial_i(t)\setminus\mathcal{D}} \left(1 - \beta 1_{\{\sigma_j(t)=I\}}\right) \cdot \left(1 - \prod_{j\in\mathcal{D}\cap\partial_i(t)} (1 - \beta 1_{\{\sigma_j(t)=I\}})\right). \tag{5}$$

Equation (5), captures the impact that the nodes in $\mathcal{D}$ have on infecting node i at day t. To simplify the analysis, we assume that the state of the nodes are independent, and we find the expectation (with respect to $\{\sigma_i(t)\}_{i\in\mathcal{V}(t)}$) of (5) as follows:

$$\mathbb{E}_{\{\sigma_i(t)\}_{i\in\mathcal{V}(t)}}\left[F_i(\mathcal{D}; t)\right] = v_S^{(i)}(t) \cdot \Big\{ \prod_{j\in\partial_i(t)\setminus\mathcal{D}} (1 - \beta v_I^{(j)}(t)) \Big\} \cdot \Big\{1 - \prod_{j\in\mathcal{D}\cap\partial_i(t)} (1 - \beta v_I^{(j)}(t)) \Big\}. \tag{6}$$

[2]In practical implementations, scheduling constraints do play a role but we disregard that in this work.

It is worth noting that (5) is a probability conditioned on $\{\sigma_i(t)\}_{i\in\mathcal{V}(t)}$, while (6) is an unconditional probability. Define

$$S\Big(\mathcal{D};t\Big) = \sum_{i\in\mathcal{V}(t)} \mathbb{E}\left[F_i(\mathcal{D};t)\right]. \tag{7}$$

Here, $S\Big(\mathcal{D};t\Big)$ represents the (expected) number of newly infectious nodes incurred by nodes in $\mathcal{D}$ at day t. Let $\mathcal{K}^\pi(t)$ be the set of nodes that are tested at time t, and $|\mathcal{K}^\pi(t)| = K^\pi(t)$. We show the following result in Appendix A.

Lemma 1. $\mathbb{E}\Big[C^\pi(t+1) - C^\pi(t)\Big] = S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big)$.

Using (3) along with Lemma 1, we seek to solve the following optimization sequentially in time for $0 \le t \le T-1$:

$$\min_{\mathbb{E}[K^\pi(t)]\le B(t)} \quad S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big). \tag{8}$$

We now prove some desired properties for the set function $S(\mathcal{K}^\pi(t);t)$ (see Appendix B).

Theorem 1. *$S\Big(\mathcal{K}^\pi(t);t\Big)$ defined in (7) is a supermodular[3] and increasing monotone function on $\mathcal{K}^\pi(t)$.*

On day t, and given the network, the probability vectors of all nodes, and $\mathcal{K}_1^\pi(t) \subset \mathcal{K}_2^\pi(t)$, for any node $i \notin \mathcal{K}_2^\pi(t)$, node i will incur larger increment of newly infectious nodes under $\mathcal{K}_2^\pi(t)$ than that under $\mathcal{K}_1^\pi(t)$. This is because node i may have common neighbors with nodes in $\mathcal{K}_2^\pi(t)$. So, supermodularity holds in Theorem 1.

The optimization (8) is NP-hard [51]. However, using the supermodularity of $S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big)$, we propose Algorithm 1 based on [52, Algorithm A] to greedily optimize (8). As proved in [52], on every day t, Algorithm 1 attains a solution such that $S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big) \le (1+\epsilon)\cdot \text{OPT}$, where $\epsilon = \frac{\epsilon'}{4(1-\epsilon')}$, and $\epsilon' = \max_{a\in\mathcal{V}(t)} \frac{S(\mathcal{V}(t);t)-S(\mathcal{V}(t)\backslash\{a\};t)-S(\{a\};t)}{S(\mathcal{V}(t);t)-S(\mathcal{V}(t)\backslash\{a\};t)}$, which is the steepness of the set function $S(\cdot;t)$ as described in [52].

In Algorithm 1, on every day t, in every step, we choose the node who provides the minimum increment on $S(\cdot;t)$ based on the results in the previous step, and then remove the node from the current node set. Algorithm 1 is stopped when $K^\pi(t)$ nodes have been chosen. The complexity of this algorithm is discussed in Appendix C.

4 Exploitation and Exploration

In Section 3, we proposed a near-optimal greedy algorithm to sequentially (in time) select the nodes to test. However, Algorithm 1 has two shortcomings. (i) The computation is costly when N and/or T are large (see Appendix C). (ii) The objective function $S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t)\Big)$ is dependent on $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$ which is unknown, even though the network and the process

[3]Let $\mathcal{X}$ be a finite set. A function $f: 2^{\mathcal{X}} \to \mathbb{R}$ is supermodular if for any $\mathcal{A}\subset\mathcal{B}\subset\mathcal{X}$, and $x\in\mathcal{S}\backslash\mathcal{B}$, $f(\mathcal{A}\cup\{x\}) - f(\mathcal{A}) \le f(\mathcal{B}\cup\{x\}) - f(\mathcal{B})$.

Algorithm 1 Greedy Algorithm

Step 0: On day t, input $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$, set $\mathcal{A}_0 = \mathcal{V}(t)$.
repeat
Step i: Let $\mathcal{A}_i = \mathcal{A}_{i-1}\backslash\{a_i\}$, where

$$a_i = \arg\min_{a\in\mathcal{A}_{i-1}} S\Big(\{a\}\cup\{a_1,\cdots,a_{i-1}\};t\Big).$$

until $i = N(t) - K^\pi(t)$, and return $\mathcal{K}^\pi(t) = \mathcal{V}(t)\backslash\mathcal{A}_i$.

are stochastically fully given (see Section 2). This is because the set of infected nodes are unknown and time-evolving.

To overcome the first shortcoming, we propose a simpler reward maximization policy by minimizing an upper bound on the objective function in (8). To overcome the second shortcoming, we estimate $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$ using the history of test observations $\{\underline{Y}(\tau)\}_{\tau=0}^t$ (as presented in Section 5). we refer to the estimates as $\{\underline{u}_i(t)\}_{i\in\mathcal{V}(t)}$. Both the greedy policy and its reward-based variant that we will propose in this section thus need to perform decision making based on the estimates $\{\underline{u}_i(t)\}_{i\in\mathcal{V}(t)}$ and we refer to them as "exploitation" policies.

It now becomes clear that testing has two roles: to find the infected in order to isolate them and contain the spread, and to provide better estimates of $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$. This leads to interesting tradeoffs between exploitation and exploration as we will discuss next. Under exploitation policies, we test nodes deterministically based on a function of $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$, (which is called "reward", and will be defined later); while under exploration policies, nodes are tested according to a probabilistic framework (based on rewards of all nodes).

To simplify the decision making into reward maximization, we first derive an upper bound on $S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big)$. Define

$$r_i(t) = S\Big(\{i\};t\Big). \tag{9}$$

Using the supermodularity of the function $S(\cdot)$, we prove the following lemma in Appendix D.

Lemma 2. $S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big) \le S\Big(\mathcal{V}(t);t\Big) - \sum_{i\in\mathcal{K}^\pi(t)} r_i(t)$.

Remark 1. *Recall that $S(\cdot;t)$ is a supermodular function, then the amount of newly infectious nodes incurred by the set $\mathcal{K}^\pi(t)$, $S(\mathcal{K}^\pi(t);t)$, is larger than the sum of the amount of newly infectious nodes by every individual node in $\mathcal{K}^\pi(t)$, i.e., $\sum_{i\in\mathcal{K}^\pi(t)} r_i(t)$. Thus, $S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big)$ is upper bounded by $S\Big(\mathcal{V}(t);t\Big) - \sum_{i\in\mathcal{K}^\pi(t)} r_i(t)$.*

We propose to minimize the upper bound in Lemma 2 instead of $S\Big(\mathcal{V}(t)\backslash\mathcal{K}^\pi(t);t\Big)$. Since $\mathcal{V}(t)$ is known and $S\Big(\mathcal{V}(t);t\Big)$ is hence a constant, the problem reduces to solving:

$$\max_{\mathbb{E}[K^\pi(t)]\le B(t)} \quad \sum_{i\in\mathcal{K}^\pi(t)} r_i(t). \tag{10}$$

Given probabilities $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$, the solution to (10) is to pick the nodes associated with the $B(t)$ largest values $r_i(t)$. We thus refer to $r_i(t)$ as the *reward* of selecting node i.

Let $\{\underline{u}_i(t)\}_{i\in\mathcal{V}(t)}$ be an estimate for $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$ found by estimating the conditional probability of the state of node i given the history of observations $\{\underline{Y}(\tau)\}_{\tau=0}^{t-1}$. Our proposed reward-based Exploitation (RbEx) policy follows the same idea of selecting the nodes with the highest rewards, but uses $\{\underline{u}_i(t)\}_{i\in\mathcal{V}(t)}$, instead of $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$, to estimate the rewards $\hat{r}_i(t)$ following (9). This is given in Algorithm 2.

Algorithm 2 Reward-based Exploitation (RbEx) Policy

Input $\{\underline{w}_i(0)\}_{i\in\mathcal{V}(t)}$, $\{\underline{u}_i(0)\}_{i\in\mathcal{V}(0)}$, $\underline{Y}(0)$, and $t=0$.
Repeat for $t=1,2,\cdots,T-1$.
Step 1: Calculate $\{\hat{r}_i(t)\}_{i\in\mathcal{V}(t)}$ based on $\{\underline{u}_i(t)\}_{i\in\mathcal{V}(t)}$ and (9).
Step 2: Re-arrange the sequence $\{\hat{r}_i(t)\}_{i\in\mathcal{V}(t)}$ in descending order, and test the first $K(t)$ nodes. Get the new observations $\underline{Y}(t)$.
Step 3: Based on $\underline{Y}(t)$, update $\{\underline{u}_i(t+1)\}_{i\in\mathcal{V}(t+1)}$ by Algorithm 4 (Step 0 $\sim$ Step 2) in Section 5.

The shortcoming of Algorithm 2 is that it targets maximizing the estimated sum rewards, even though the estimates may be inaccurate. In this case, testing is heavily biased towards the history of testing and it does not provide opportunities for getting better estimates of the rewards. For example, consider a network with several clusters. If one positive node is known by Algorithm 2, then it may get stuck in that cluster and fail to locate more positives in other clusters.

In Section 5.2, we will prove, in a line network, that the exploitation policy described in Algorithm 2 can be improved by a constant factor (in terms of the resulting cumulative infections) if a simple form of exploration is incorporated.

We next propose an exploration policy. Our proposed policy is probabilistic in the sense that the nodes are randomly tested with probabilities that are proportional to their corresponding estimated rewards. This approach has similarities and differences to Thompson sampling and more generally posterior sampling. The similarity lies in the probabilistic nature of testing using posterior probabilities. The difference is that in our setting decision making depends on the distributions of decision variables, but not samples of the decision variables.

More specifically, at time t, node i is tested with probability $\min\{1, \frac{B(t)\hat{r}_i(t)}{\sum_{j\in\mathcal{V}(t)}\hat{r}_j(t)}\}$, which depends on the budget $B(t)$. Note that each node is tested with probability at most 1; so if $\frac{B(t)\hat{r}_i(t)}{\sum_{j\in\mathcal{V}(t)}\hat{r}_j(t)} > 1$ for some node i, then we would not fully utilize the budget. The unused budget is thus

$$c(t) = \sum_{i\in\mathcal{V}(t)} \Big(\frac{B(t)\hat{r}_i(t)}{\sum_{j\in\mathcal{V}(t)}\hat{r}_j(t)} - 1\Big)^+ \tag{11}$$

and can be used for further testing[4]. Algorithm 3 outlines our proposed Reward-based Exploration (RbEr) policy.

Algorithm 3 Reward-based Exploration (RbEr) Policy

Input $\{\underline{w}_i(0)\}_{i\in\mathcal{V}(t)}$, $\{\underline{u}_i(0)\}_{i\in\mathcal{V}(0)}$, $\underline{Y}(0)$, and $t=0$.
Repeat for $t=1,2,\cdots,T-1$
Step 1: Calculate $\{\hat{r}_i(t)\}_{i\in\mathcal{V}(t)}$ based on $\{\underline{u}_i(t)\}_{i\in\mathcal{V}(t)}$ and (9).
Step 2: Test node i with probability $\min\{1, \frac{B(t)\hat{r}_i(t)}{\sum_{j\in\mathcal{V}(t)}\hat{r}_j(t)}\}$. If $c(t)$, defined in (11), is positive, randomly select $c(t)$ further nodes to test (see Footnote 4). Get the new observations $\underline{Y}(t)$.
Step 3: Based on $\underline{Y}(t)$, update $\{\underline{u}_i(t+1)\}_{i\in\mathcal{V}(t+1)}$ by Algorithm 4 (Step 0 $\sim$ Step 2) in Section 5.

5 Message-Passing Framework

As discussed in Section 4, the probabilities $\{\underline{v}_i(t)\}_i$ are unknown. In this section, we develop message passing framework to sequentially estimate $\{\underline{v}_i(t)\}_i$ based on the network observations and the dynamics of the spread process. We refer to these estimates as $\{\underline{u}_i(t)\}_i$.

When node i is tested on day t, an observation $Y_i(t)$ is provided about its state. Knowing the state of node i provides two types of information: (i) it provides information about the state of the neighboring nodes in future time slots $t+1, t+2, \ldots$ (because of the evolution of the spread in time and on the network), and (ii) it also provides information about the past of the spread, meaning that we can infer about the state of the (unobserved) nodes at previous time slots. For example, if node i is tested positive in time t, we would know that (i) its neighbors are more likely to be infected in time $t+1$ and (ii) some of its neighbors must have been infected in a previous time for node i to be infected now. This forms the basis for the backward-forward message passing framework.

Given the spread model of Section 2, we first describe the forward propagation of belief. Suppose that at time t, the probability vector $\underline{v}_i(t)$ is given for all i. The probability vector $\underline{v}_i(t+1)$ can be computed as follows (see Appendix E):

$$\underline{v}_i(t+1) = \underline{v}_i(t) \times \mathtt{P}_i\Big(\{\underline{v}_j(t)\}_{j\in\partial_i^+(t)}\Big) \tag{12}$$

where $\mathtt{P}_i\Big(\{\underline{v}_j(t)\}_{j\in\partial_i^+(t)}\Big)$ is a local transition probability matrix given in Appendix E.

Recall that $\underline{Y}(t)$ denotes the collection of network observations on day t. The history of observations is then denoted by $\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}$. Based on these observations, we wish to find an estimate of the probability vector $\underline{v}_i(t)$ for each $i\in\mathcal{V}(t)$. We denote this estimate by $\underline{u}_i(t) = (u_x^{(i)}(t), x\in\mathcal{X})$ and refer to it, in this section, as the *prior probability* of node i in time t. We further define the *posterior probability* $\underline{w}_i(t) = (w_x^{(i)}(t), x\in\mathcal{X})$ of node i in time

[4]Note that $c(t)$ is not always an integer. Instead of $c(t)$, we use $\mathtt{Int}\big(c(t)\big)$ with probability $|\mathtt{Int}\big(c(t)\big)-c(t)|$ where $\mathtt{Int}(\cdot)\in\{\lfloor\cdot\rfloor,\lceil\cdot\rceil\}$.

t (after obtaining new observations $\underline{Y}(t)$). In particular,

$$u_x^{(i)}(t) = \Pr\left(\sigma_i(t) = x | \{\underline{Y}(\tau)\}_{\tau=1}^{t-1}\right)$$
$$w_x^{(i)}(t) = \Pr\left(\sigma_i(t) = x | \{\underline{Y}(\tau)\}_{\tau=1}^{t}\right).$$

Here, the prior probability is defined *at the beginning of* every day, and the posterior probability is defined *at the end of* every day. Conditioning all probabilities in (12) on $\{\underline{Y}(\tau)\}_{\tau=1}^{t}$, we obtain the following *forward-update rule* (see Appendix F)

$$\underline{u}_i(t+1) = \underline{w}_i(t) \times \mathtt{P}_i\Big(\{\underline{w}_j(t)\}_{j\in\partial_i^+(t)}\Big). \tag{13}$$

Remark 2. *Following (13), we need to utilize the observations $\underline{Y}(t)$ and the underlying dependency among nodes' states to update the posterior probabilities $\{\underline{w}_i(t)\}_i$, and consequently update $\{u_i(t+1)\}_i$ based on the forward-update rule (13). This is however non-trivial. A Naive approach would be to locally incorporate node i's observation $Y_i(t)$ into $w_i(t)$ and obtain $\underline{u}_i(t+1)$ using* (13)*. This approach, however, does not fully exploit the observations and it disregards the dependency among nodes' states, as caused by the nature of the spread (An example is provided in Appendix H).*

Backward Propagation of Belief To capture the dependency of nodes' states and thus best utilize the observations, we proceed as follows. First, denote

$$\underline{e}_i(t-1) = (e_x^{(i)}(t-1), x \in \mathcal{X})$$
$$e_x^{(i)}(t-1) = \Pr\left(\sigma_i(t-1) = x | \{\underline{Y}(\tau)\}_{\tau=1}^{t}\right).$$

Vector $\underline{e}_i(t-1)$ is the posterior probability of node i at time $t-1$, after obtaining the history of observations up to and including time t. By computing $\underline{e}_i(t-1)$, we are effectively correcting our belief on the state of the nodes in the previous time slot by *inference* based on the observations acquired at time t. This constitutes the *backward step* of our framework and we will expand on it shortly. The backward step can be repeated to correct our belief also in times $t-2$, $t-3$, etc. For clarity of presentation and tractability of our analysis and experiments, we truncate the backward step at time $t-1$ and present assumptions under which this truncation is theoretically justifiable. Considering larger truncation windows is straightforward but out of the scope of this paper.

Once our belief about nodes' states is updated in prior time slots (e.g., $\underline{e}_i(t-1)$ is obtained), it is propagated forward in time for *prediction* and to provide a more accurate estimate of the nodes' posterior and prior probabilities. More specifically, consider (12) written for time t (rather than $t+1$) and condition all probabilities on $\{\underline{Y}(\tau)\}_{\tau=1}^{t}$. We obtain the following update rule (see Appendix F):

$$\underline{w}_i(t) = \underline{e}_i(t-1) \times \tilde{\mathtt{P}}_i\Big(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big) \tag{14}$$

where $\tilde{\mathtt{P}}_i\Big(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big)$ is given in Appendix F. *Note that the local transition matrix in (14) is not the same as (13). This is because "future" observations were available in*

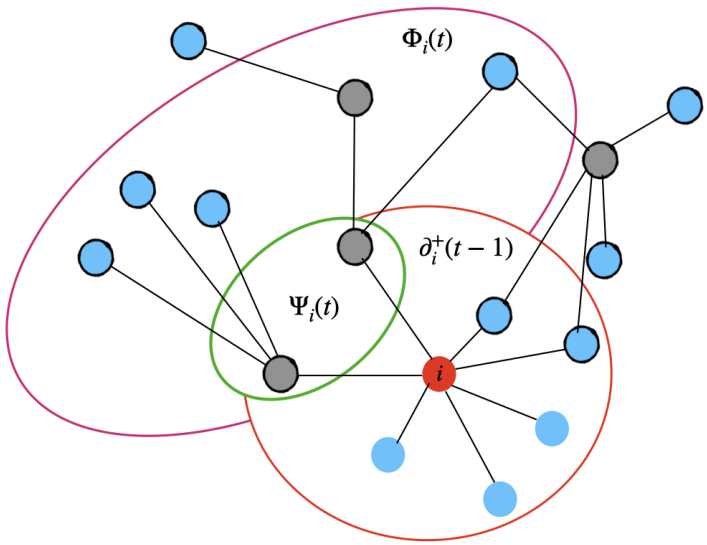

Figure 2: An example of $\Psi_i(t)$, $\Phi_i(t)$ and $\Theta_i(t)$. Node i is marked in red, and its neighborhood $\partial_i^+(t-1)$ is shown by the red contour. Suppose that the gray nodes are tested on day $t-1$, then $\Psi_i(t)$ is the set of nodes within the green contour, and $\Phi_i(t)$ consists of the nodes in the purple contour. Finally, nodes in $\Theta_i(t)$ are marked with bold black border

.

$\tilde{\mathtt{P}}_i\Big(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big)$. The probability vectors $\{\underline{e}_i(t-1)\}_i$ provide better estimates for $\{\underline{w}_i(t)\}_i$ through (14) and the prior probabilities $\{\underline{u}_i(t+1)\}_i$ are then computed using (13) to be used for decision making in time $t+1$. The block diagram in Fig. 3 depicts the high-level idea of our framework. It is worth noting that $\mathtt{P}_i\Big(\{\underline{w}_j(t)\}_{j\in\partial_i^+(t)}\Big)$ in (13) and $\tilde{\mathtt{P}}_i\Big(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big)$ in (14) both depend on the observations, $\{\underline{Y}(\tau)\}_{\tau=1}^t$.

We next discuss how $\underline{e}_j(t-1)$ can be computed, starting with some notations. Denote by

$$\zeta_i(t) = \sigma_i(t)|_{\{\underline{Y}(\tau)\}_{\tau=1}^t}, \quad \theta_j(t) = \sigma_i(t)|_{\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}}, \tag{15}$$

the state of the nodes in the posterior probability spaces conditioned on the observations $\{\underline{Y}(\tau)\}_{\tau=1}^t$ and $\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}$, respectively. We further define $\Psi_i(t)$ to be the set of those neighbors of node i at time $t-1$, including node i, who are observed/tested at time t. This set consists of all nodes whose posterior probabilities will be updated at time $t-1$ (given a new observation $Y_i(t)$). The set of all neighbors (except node i) of the nodes in $\Psi_i(t)$ then defines $\Phi_i(t)$. The set $\Phi_i(t)$ consists of all nodes whose posterior probabilities at time t is updated by the observation $Y_i(t)$. More precisely, we have

$$\begin{aligned}
\Psi_i(t) &= \mathcal{O}(t) \cap \partial_i^+(t-1),\\
\Phi_i(t) &= \{j | j \in \partial_k^+(t-1), k \in \Psi_i(t)\} \backslash \{i\},\\
\Theta_i(t) &= \{j | j \in \partial_k^+(t-1), k \in \mathcal{O}(t)\} \backslash \{i\}
\end{aligned}$$

where $\mathcal{O}(t)$ is the set of observed nodes at time t (see Figure 2). In Appendix G, we show

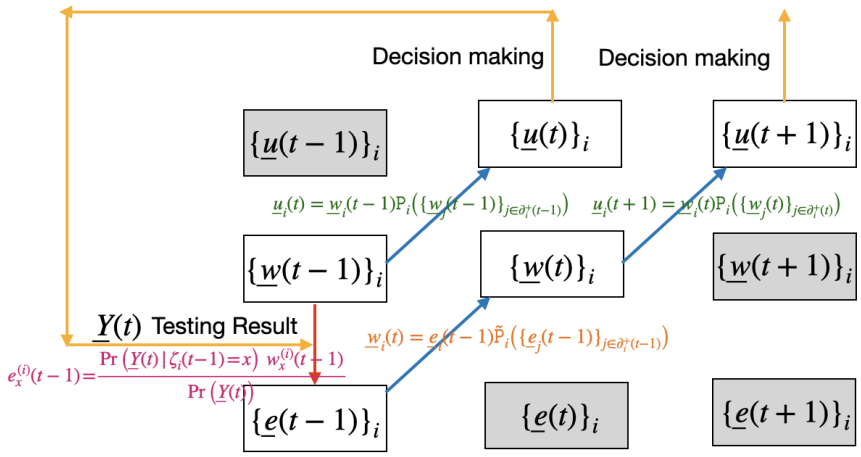

Figure 3: The process of Algorithm 4. One example of a complete process is given in unshaded blocks. Recall that $\underline{u}_i(\tau)$, $\underline{w}_i(\tau)$, and $\underline{e}_i(\tau)$, where $\tau \in \{t-1, t, t+1\}$, are the prior probabilities, the posterior probabilities, and the updated posterior probabilities, respectively.

$$e_x^{(i)}(t-1) = \frac{\Pr\left(\underline{Y}(t)|\zeta_i(t-1)=x\right)\ w_x^{(i)}(t-1)}{\Pr\left(\underline{Y}(t)\right)}. \tag{16}$$

It suffices to find $\Pr\left(\underline{Y}(t)|\zeta_i(t-1) = x\right)$. The denominator $\Pr\left(\underline{Y}(t)\right)$ is then found by normalization of the enumerator in (16). Let $\{x_j\}_{j\in\mathcal{O}(t)}$ be a realization of $\{\theta_j(t)\}_{j\in\mathcal{O}(t)}$ and $\{y_l\}_{l\in\Theta_i(t)}$ be a realization of $\{\zeta_l(t-1)\}_{l\in\Theta_i(t)}$. We prove the following in Appendix G under a simplifying truncation assumption (see Assumption 1 in Appendix G) where the backward step is truncated in time $t-1$:

$$\begin{aligned}
&\Pr\left(\underline{Y}(t)|\zeta_i(t-1) = x\right) = \Pr\left(\{Y_j(t)\}_{j\in\Psi_i(t)}|\zeta_i(t-1) = x\right) \\
&= \sum_{\{x_j\}_{j\in\Psi_i(t)}} \prod_{j\in\Psi_i(t)} \Pr\left(Y_j(t)|\theta_j(t)\right) \\
&\times \sum_{\{y_l\}_{l\in\Phi_i(t)}} \prod_{j\in\Psi_i(t)} \Pr\left(x_j|\{y_l\}_{l\in\partial_j^+(t-1)\setminus\{i\}}, x\right) \times \prod_{l\in\{\Phi_i(t)\}} w_{y_l}^{(i)}(t-1).
\end{aligned} \tag{17}$$

We finally present our *Backward-Forward Algorithm* to sequentially compute estimates $\{\underline{u}_i(t)\}_i$ in Algorithm 4. The process of Algorithm 4 is given in Fig 3, and we also give a simple example to show the process of Algorithm 4 in Appendix H.

5.1 Necessity of Backward Updating

Now we provide an example which illustrates the necessity of backward updating.

Example 1. *Consider a line network with the node set $\mathcal{V} = \{1, 2, \ldots, N\}$ and the edge set $\mathcal{E} = \{(i, i+1), 1 \le i \le N-1\}$ (see Figure 4). On the initial day, we assume that each node is infected independently with probability $1/N$. Let $\beta = 1$, $\lambda = 0$, $\gamma = 0^5$, and $B(t) = 1$. We further assume that there is no isolation when a positive node is tested.*

Algorithm 4 Backward-Forward Algorithm

Input $\underline{Y}(0)$, $\{\underline{e}_i(0)\}_{i\mathcal{V}(0)}$, $\{\underline{w}_i(0)\}_{i\mathcal{V}(0)}$, $\{\underline{u}_i(0)\}_{i\mathcal{V}(0)}$.
Repeat for $t = 1, 2, \cdots, T-1$
Step 0: Based on $\underline{Y}(t)$, get $\mathcal{V}(t)$ from $\mathcal{V}(t-1)$.
Step 1: Backward step. Update $\underline{e}_i(t-1)$ by (16), (17), and then compute $\underline{w}_i(t)$ by (14).
Step 2: Forward step. Compute $\underline{u}_i(t+1)$ by (13).

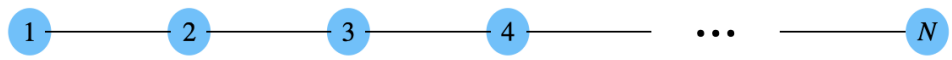

Figure 4: The line network in Example 1.

Based on Example 1, we show that the naive approach of Remark 2 (i.e., forward-only updating) will cause the estimated probabilities to never converge to the true probabilities of infection. Nonetheless, if we use the Backward-Forward Algorithm 4, the estimated probabilities converge to the true probabilities after a certain number of steps. Formally, we prove the following result in Appendix J.

Theorem 2. *For any testing policy that sequentially computes $\{\underline{u}_i(t)\}_i$ based on (13) (see Remark 2), with probability (approximately) $\frac{1}{e}$, we have $\sum_{i=1}^{N} ||\underline{v}_i(t) - \underline{u}_i(t)|| \overset{t\to\infty}{\to} \Theta(N)$, for large N. On the other hand, there exists a testing policy that sequentially updates $\{\underline{u}_i(t)\}_i$ based on Algorithm 4 and attains $\sum_{i=1}^{N} ||\underline{v}_i(t) - \underline{u}_i(t)|| = 0,\ t \geq 2N$.*

Roadmap of proof: Consider a simple case where every node is susceptible. Since each node is infected with probability $1/N$, then the case occurs with probability $\simeq 1/e$.

Under the case above, consider any testing policy based on the algorithm in Remark 2 . If a node is tested on day t, then the policy "clears" the tested node. Since the updating rule of the algorithm can not go back to the information on day $t-1$, then it can not "clear" any neighbors of the tested node and its probability of infection updates to a non-zero value in the next day. Furthermore, we show that almost all nodes have an significantly large probability of infection when time horizon is sufficiently large, hence $\sum_{i=1}^{N} ||\underline{v}_i(t) - \underline{u}_i(t)|| \overset{t\to\infty}{\to} \Theta(N)$.

On the other hand, we can propose a specific testing policy. Note that there is no infection, if Algorithm 4 is used to update probabilities, then it can reveal the states of all nodes under the specific testing policy after at most $2N$ days. So we have $\sum_{i=1}^{N} ||\underline{v}_i(t) - \underline{u}_i(t)|| = 0,\ t \geq 2N$.

In Theorem 2, we illustrate the necessity of backward updating when testing is limited. In essence, we want to "clear" the graph and confirm that there are no infections. If the number of tests is limited, we have mathematically shown that no algorithm can correctly estimate the nodes' infection probabilities if it does not use the backward (inference) step. On the contrary, there is an algorithm that uses the backward step along with the forward step and the estimates that it provides for the nodes' infection probabilities converge to the true

[5]Here, $\lambda = 0$ implies there is no latent state, and $\gamma = 0$ implies that nodes never recover.

probabilities of the nodes after some finite steps. Even though the considered graph is simple but the phenomena it captures is general.

As discussed Theorem 2, the backward updating is necessary. However, to trade off the impact of backward updating and the reduction of computation complexity, we propose an α-linking backward updating algorithm in Appendix K, where Algorithm 4 is applied on a random subgraph with fewer edges.

5.2 Necessity of Exploration

Note that in reality we have no information for $\{\underline{v}_i(t)\}_i$, and only have the estimates $\{\underline{u}_i(t)\}_i$. One may wonder if exploitation based on wrong initial estimated probability vectors, i.e., $\{\underline{u}_i(0)\}_i$, misleads decision making by providing poorer and poorer estimates of the probabilities of infection. If so, exploration may be necessary.

Example 2. *Consider $\mathcal{V} = \{1, 2, \ldots, N\}$ and edges $\mathcal{E} = \{(i, i+1), 1 \leq i \leq N-1\}$ (see Figure 4). Let $N \gg 10$, $\beta = 1$, $\lambda = 0$, $\gamma = 0$, and $B(t) = 10$. Suppose that on the initial day, node 1 is infected and all other nodes are susceptible. Consider a wrong initial estimate: $w_I^{(i)}(0) = u_I^{(i)}(0) = 0$ if $i \leq \frac{9N}{10}$, and $w_I^{(i)}(0) = u_I^{(i)}(0) = \frac{10\epsilon}{N}$ otherwise, where $\epsilon > 0$. With this initial belief, we have $\sum_{i=1}^{N} ||\underline{w}_i(0) - \underline{v}_i(0)|| = O(1+\epsilon)$.*

Different from Example 1, here we consider the isolation of nodes that are tested positive. In Example 2, suppose that a specific exploration policy is applied: 1 (out of 10) tests is done randomly, and the other 9 tests are done following exploitation. Now, in Appendix L, we show that under the RbEx policy, the cumulative infection is at least aN for a constant a, while under the exploration policy defined above, the cumulative infection is at most bN with very high probability, and the ratio a/b can be any constant for a large enough N. More formally, we have the following theorem. Define p_0 as a large probability.

Theorem 3. *With probability $p_0 \geq \frac{99}{100}$, the cumulative infections of the policy with exploration is, by a constant factor, smaller than what the RbEx policy achieves.*

Roadmap of proof: Under the RbEx policy, we test nodes based on their predicted probabilities. Since the nodes that are located towards the end of the line (right side in Fig. 4) have non-zero probabilities, they are tested first while the disease spreads on the other end of the network (left side in Fig. 4). Mathematically, suppose that for the first time, an infectious node is tested at day $t = aN$, then there are at least $\min\{aN, N\}$ infectious nodes before the spread can be contained.

Under the specific exploration policy described above, consider the event that, for the first time, an infectious node is explored on day $t = b'N$ ($b' < a$). We argue that with probability p_0, the exploration policy catches at least two new infections at each step after $t = b'N$. After $2t$, the algorithm catches all the infections, and we have at most $2b'N$ infections. Let $b = 2b'$. This is an improvement by a factor of at least $\frac{a}{b}$ in comparison to the RbEx strategy. Factor $\frac{a}{b}$ depends on the values of N and p_0.

In Theorem 3, we show the necessity of exploration when our initial belief is slightly wrong, i.e., it is slightly biased toward the other end of the network (In general, this could be due to a wrong belief, prior test results, etc). We have formally proved that when the testing capacity

is limited, exploration can significantly improve the cumulative infections, i.e., contain the spread. This motivates the design of exploration policies. Even though the setting is simple, the phenomena it captures is much more general.

6 Simulations

6.1 Overview

In this section, we use simulations to study the performance of the proposed exploitation and exploration policies for various synthetic and real-data networks. Towards this end, we define some metrics that quantify how different metrics perform and key network parameters and attributes that determine the values of these metrics and thereby how exploitation and exploration compare. We also identify benchmark policies which represent the extreme ends of the tradeoff between exploration and exploitation to compare with the policies we propose and assess the performance enhancements brought about by judicious combinations of exploration and exploitation. Through our experiments, we aim to answer two main questions for various synthetic and real-data networks: (i) Can exploration policies do better that exploitation policies and if so, when would that be the case? (ii) What parameters would affect the performance of exploration and exploitation policies? These are important questions to shed light on the role of exploration. These questions are particularly raised by Theorem 3 in which we prove that exploration can significantly outperform exploitation in some (stylized) networks. We design the experiments in order to shed light on the above questions and to understand the extent of the necessity of exploration in different network models and scenarios.

Network parameters We consider the following parameters: (i) The unregulated delay ℓ which is the time from the initial start of the spread to the first time testing and intervention starts; (ii) The *(global) clustering coefficient* [53, Chapter 3], denoted by γ_c, which is defined as a measure of the degree to which nodes in a graph tend to cluster together; (iii) The *path-length*, denoted by L_p, which measures the average shortest distance between every possible pair of nodes. We consider attributes such as the initialization of the process, and the lack of knowledge about $\{\underline{v}_i(t)\}_i$.

Performance metrics We consider the expected number of infected nodes in a time horizon $[0, T]$ as the performance measure for various policies. Let $C^0(T)$ be the number of infected nodes if there is no testing and isolation, $C^{RbEx}(T), C^{RbEr}(T)$ be the corresponding numbers respectively for the *RbEx* policy (Algorithm 2) and the *RbEr* policy (Algorithm 3). We consider a ratio between the expectations of these:

$$\texttt{Ratio} = \frac{\mathbb{E}[C^{RbEx}(T)] - \mathbb{E}[C^{RbEr}(T)]}{\mathbb{E}[C^0(T)]}. \tag{18}$$

We define the *estimation error* $\texttt{Err}_\pi(t)$ towards capturing the impact of the lack of knowledge about $\{\underline{v}_i(t)\}_i$.

$$\texttt{Err}_\pi(t) = \frac{1}{N(t)} \sum_{j \in \mathcal{G}(t)} ||\underline{v}(t) - \underline{u}(t)||_2^2. \tag{19}$$

We consider the difference between the estimation errors of *RbEx* and *RbEr* policies: $\Delta_{\texttt{Err}} = \texttt{Err}_{RbEx}(T) - \texttt{Err}_{RbEr}(T)$.

Benchmark policies We will compare the proposed policies with 3 benchmark policies. (i) Contact Tracing: we tested every day the nodes who have infectious neighbors, denoted by *candidate nodes*. Only some candidate nodes are selected randomly due to testing resources being limited. Note that only exploitation is utilized under this benchmark. (ii) Random Testing: Every day, we randomly select nodes to test. Typical testing policies that could come out of SIR optimal control formulations for our problem would naturally reduce to random testing as they treat all nodes to be statistically identical and ignore the impact of network topology. One can interpret that random testing implements exploration to its full extent. (iii) Logistics Regression: We use ideas presented in [47], where simple classifiers were proposed based on the features of real data. In our setting, we choose the classifier to be based on logistic regression, and we define the feature of node i as $X_i(t) = [1, n_i(t) + \epsilon]^T$. Here, $n_i(t)$ is the number of quarantined neighbors node i has contacted before and including day t, and $\epsilon \neq 0$ is a superparameter aiming to avoid the case when $n_i(t) = 0$. In simulations, we set $\epsilon = 0.1$. Let the observation $Y_i(t)$ be the testing result of node i. In particular, if node i is not tested on day t, then we do *not* collect the data $(X_i(t), Y_i(t))$. Thus, the probability of node i being infectious is defined as the *Sigmoid* function

$$\frac{1}{1 + \exp(-X_i(t) \cdot w^T)},$$

where w is the parameter which should be learned.

Simulation Setting We consider a process as described in Section 2 with n_0 randomly located initial infected nodes. The process evolved without any testing/intervention for ℓ days and we refer to ℓ as the *unregulated delay*. After that, one of the (initial) infectious nodes, denoted by node i_0, is (randomly) provided to the policies. Subsequently, the initial estimated probability vector is set to $\underline{u}_{i_0}(\ell) = (1, 0, 0, 0)$, and $\underline{u}_i(\ell) = (0, 0, 0, 1)$ when $i \neq i_0$. We consider the budget to be equal to the expected number of infected nodes at time t, i.e., $B(t) = \sum_{j=1}^{N(t)} v_I^{(j)}(t)$.

We choose model parameters considering the particular application of COVID-19 spread. In particular, 1) the mean latency period is $1/\lambda = 1$ or 2 days [54]; 2) the mean duration in the infectious state (I) is $1/\gamma = 7 \sim 14$ days [54, 55, 56]; 3) we choose the transmission rate β in a specific network such that after a long time horizon, if no testing and isolation policies were applied, then around $60 \sim 90$ percent individuals are infected. We did not consider the case where 100 percent individuals are infected because given the recovery rate (and the topology), the spread may not reach every node.

We consider both synthetic networks such as Watts-Strogatz (WS) networks [57], as well as real-data networks. Further results for the Stochastic Block Model (SBM) [58] and some variants of it are presented in Appendix M.2.

Watts-Strogatz networks We consider a network $\text{WS}(N, d, \delta)$ with N nodes, degree d, and rewiring probability δ. The transmission probability of the spread is set to $\beta = 0.4$ and the initial seed is $n_0 = 3$.

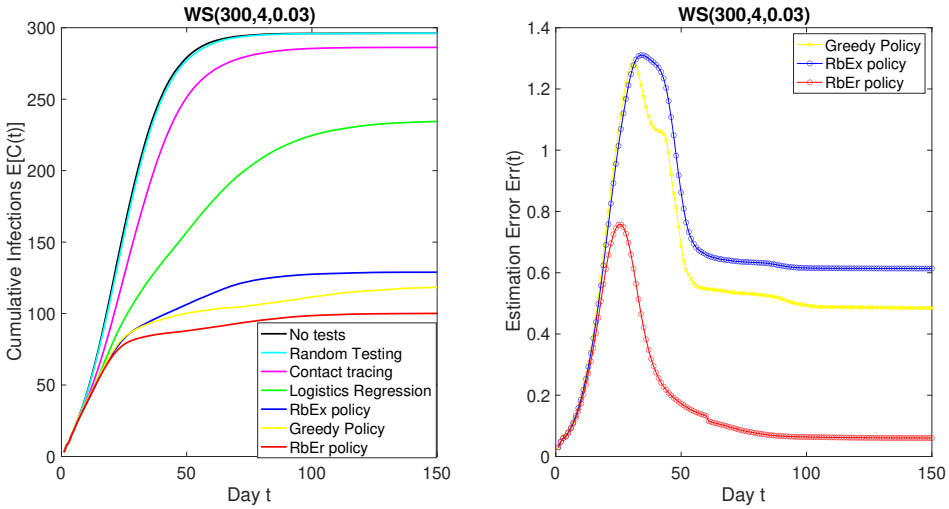

Figure 5: Performances and estimation errors of different policies in WS(300, 4, 0.03).

The Real-data Network We consider the contact network of university students in the Copenhagen Networks Study [59]. The network is built based on the proximity between participating students recorded by smartphones, at 5 minute resolution. According to the definition of close contact by [56], we only used proximity events between individuals that lasted more than 15 minutes to construct the daily contact network. The contact network has 672 individuals spanning 28 days. To guarantee a long time-horizon, we replicate the contact network 4 times so that the time-horizon is 112 days. We set $\beta = 0.05$ and $n_0 = 5$ to have a realistic simulation of the Covid-19 spread..

6.2 Simulation Results

6.2.1 Synthetic networks

Comparison of performance of different policies Figure 5 shows the performance of different policies in a WS network with $N = 300$, $d = 4$, and $\delta = .03$. We show that all the proposed policies outperform the contact tracing policy, the Greedy policy outperforms the RbEx policy, and the RbEr policy outperforms both the greedy and the RbEx policy. In addition, we see on the right figure that the RbEr policy provides a more accurate estimation for $\{\underline{v}_i(t)\}_i$.

Impact of network parameters on the tradeoff between exploration and exploitation We next vary the rewiring probability δ from 0 to 0.03 to investigate the role of path length L_p and clustering coefficient γ_c (see Table 2). The results show that as L_p or γ_c decreases, the benefits of exploration compared to exploitation decrease as well. This confirms one's intuition that exploration is particularly helpful in clustered networks with larger path lengths where undetected infection can spread without any intervention as exploitation would be heavily focus on local infections that were previously found. This is also confirmed by the better estimation error that exploration provides in such scenarios, as shown in Table 2.

In order to investigate the role of γ_c and L_p individually, not through the common factor δ, we consider different WS networks with degrees $d = 4, 6$. We adjust the rewiring proba-

WS, δ	0	.0075	.015	.0225	.03
γ_c	.5	.489	.473	.467	.456
L_p	62.876	21.264	14.253	12.171	10.81
`Ratio`	0.191	0.182	0.174	0.126	0.097
$\Delta_{\texttt{Err}}$	1.153	1.423	0.991	0.779	0.554

Table 2: Role of clustering coefficient and path length

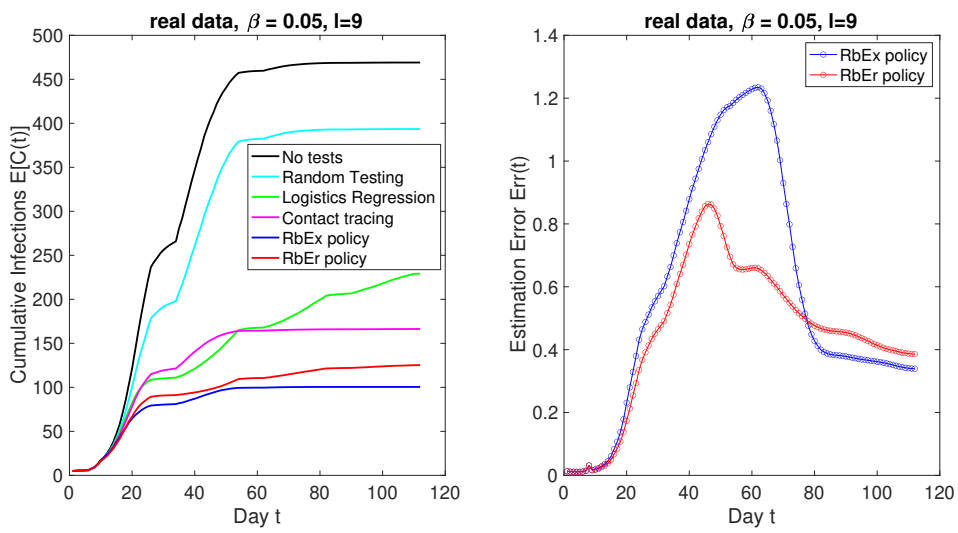

Figure 6: Performances and estimation errors of different policies in the real data network.

bility δ, such that one of (γ_c, L_p) is almost constant, and the other is varying. We show in Appendix M.1 that the trend is similar to what we observed by varying δ in Table 2.

Finally, the impact of ℓ is shown in Table 3. Fix $\delta = .03$. As ℓ increases, so does `Ratio`, implying that exploration becomes more essential. This is because exploitation may get stuck in the neighborhoods of some infected nodes.

WS, ℓ	3	5	7	9	11
`Ratio`	0.097	0.128	0.177	0.207	0.297
$\Delta_{\texttt{Err}}$	0.553	0.814	1.092	1.197	1.449

Table 3: Role of the unregulated delay ℓ

6.2.2 A real-data network

In Figure 6, we show that our proposed policies can contain the spread and they outperform the contact tracing policy. In this figure, we have $\ell = 9$ and exploitation is shown to provide a better estimation error and a better performance in terms of the cumulative infections. As ℓ increases, however, as we show in Table 4 that the benefit of exploitation decreases and exploration outperforms exploitation for larger ℓ. Our results are thus consistent with synthetic networks.

ℓ	5	8	11
`Ratio`	-0.0559	-0.0255	0.009
$\Delta_{\texttt{Err}}$	-0.061	-0.030	0.035

Table 4: Role of the unregulated delay ℓ

7 Conclusions and Future Work

In this paper, we studied the problem of containing a spread process (e.g. an infectious disease such as COVID-19) through sequential testing and isolation. We modeled the spread process by a compartmental model that evolves in time and stochastically spreads over a given contact network. Given a daily test budget, we aimed to minimize the cumulative infections. Under mild conditions, we proved that the problem can be cast as minimizing a supermodular function expressed in terms of nodes' probabilities of infection and proposed a near-optimal greedy testing policy. We subsequently designed a computationally tractable reward-based policy that preferentially tests nodes that have higher rewards, where the reward of a node is defined as the expected number of new infections it induces in the next time slot. We showed that this policy effectively minimizes an upper bound on the cumulative infections.

These policies, however, need knowledge about nodes' infection probabilities which are unknown and evolving. Thus, they have to be actively learned by testing. We discussed how testing has a dual role in this problem: (i) identifying the infected nodes and isolating them in order to contain the spread, and (ii) providing better estimates for the nodes' infection probabilities. We proved that this dual role of testing makes decision making more challenging. In particular, we showed that reward based policies that make decisions based on nodes' estimated infection probabilities can be arbitrarily sub-optimal while incorporating simple forms of exploration can boost their performance by a constant factor. Motivated by this finding, we devised exploration policies that probabilistically test nodes according to their rewards and numerically showed that when (i) the unregulated delay, (ii) the global clustering coefficient, or (iii) the average shortest path length increase, exploration becomes more beneficial as it provides better estimates of the nodes' probabilities of infection.

Given the history of observations, computing nodes' estimated probabilities of infection is itself a core challenge in our problem. We developed a message-passing framework to estimate these probabilities utilizing the observations in form of the test results. This framework passes messages back and forth in time to iteratively predict the probabilities in future and correct the errors in the estimates in prior time instants. This framework can also be of independent interest.

We showed novel tradeoffs between exploration and exploitation, different from the ones commonly observed in multi-armed bandit settings: (i) in our setting, the number of arms is time-variant and actions cannot be repeated; (ii) the tradeoffs in our setting are not due to lack of knowledge about the network or the process model, but rather due to lack of knowledge about the time-evolving unknown set of infected nodes.

We have made two assumptions for the purpose of analytical tractability which do not hold in practice: (1) given the entire history of testing results the states of nodes on the truncation day are independent (Assumption 1), (2) the infections from different nodes are independent (Mean-field approximations). These assumptions were used to derive the message passing framework and to prove that the objective function is super-modular which in turn led to a myopic testing strategy which is also optimal. Next, our framework can be extended to incorporate delay and/or error in test results in a relatively straightforward manner (an outline of the extension incorporating a delay is given in Appendix I), but generalizing the performance guarantees for the proposed policies in these cases forms another direction of future research. This includes establishing fundamental lower bounds using genie-aided myopic policies.

Acknowledgments

This work was supported by NSF CAREER Award 2047482 and NSF Award 1909186.

References

[1] B. Shulgin, L. Stone and Z. Agur . Pulse vaccination strategy in the SIR epidemic model. *Bulletin of Mathematical Biology*, 60:1123 – 1148, 1998.

[2] P. Tapaswi and J. Chattopadhyay. Global stability results of a "susceptible-infective-immune-susceptible" (SIRS) epidemic model. *Ecological Modelling*, 87(223 - 226), 1996.

[3] L.Stone, B.Shulgin and Z.Agur. Theoretical examination of the pulse vaccination policy in the SIR epidemic model. *Mathematical and Computer Modelling*, 31:207 – 215, 2000.

[4] Y. Takeuchi, W. Ma and E. Beretta. Global asymptotic properties of a delay SIR epidemic model with finite incubation times. *Nonlinear Analysis: Theory, Methods & Applications*, 42:931 – 947, 2000.

[5] J. Aron. Acquired immunity dependent upon exposure in an SIRS epidemic model. *Mathematical Biosciences*, 88:37 – 47, 1988.

[6] L. Allen. Some discrete-time SI, SIR, and SIS epidemic models. *Mathematical Biosciences*, 124:83 – 105, 1994.

[7] A. M. Ramos, M. R. Ferrandez, M. Vela-Perez and et al. A simple but complex enough θ-SIR type model to be used with COVID-19 real data. Application to the case of Italy. *Physica D*, 421-132839, 2021.

[8] A. G. M. Neves and G. Guerrero. Predicting the evolution of the COVID-19 epidemic with the A-SIR model: Lombardy, Italy and Sao Paulo state, Brazil. *Physica D*, 413-132693, 2020.

[9] A. Simha, R. Prasad and S. Narayana. A simple Stochastic SIR model for COVID-19 Infection Dynamics for Karnataka after interventions – Learning from European Trends. arXiv: 2003.11920, 2020.

[10] B. Ndiaye, L. Tendeng and D. Seck. Analysis of the COVID-19 pandemic by SIR model and machine learning technics for forecasting. arXiv: 2004.01574, 2020.

[11] J. Zhu, P. Ge, C. Jiang and et al. Deep-learning artificial intelligence analysis of clinical variables predicts mortality in COVID-19 patients. *Journal of the American College of Emergency Physicians Open*, 1(6):1364–1373, 2020.

[12] C. Mahanty, R. Kumar, B. K. Mishra, and et al. Prediction of COVID-19 active cases using exponential and non-linear growth models. *Expert Systems*, 39(3), 2020.

[13] E. B. Postnikov. Estimation of COVID-19 dynamics "on a back-of-envelope": Does the simplest SIR model provide quantitative parameters and predictions? *Chaos, Solitons and Fractals*, volume = 135-109841, 2020.

[14] B. Ndiaye, L. Tendeng and D. Seck. Comparative prediction of confirmed cases with COVID-19 pandemic by machine learning, deterministic and stochastic SIR models. arXiv: 2004.13489, 2020.

[15] I. Rahimi, A. H. Gandomi, P. G. Asteris and et al. Analysis and Prediction of COVID-19 Using SIR, SEIQR, and Machine Learning Models: Australia, Italy, and UK Cases. *Information*, 12(109), 2021.

[16] G. Hu and J. Geng. Heterogeneity learning for SIRS model: an application to the COVID-19. *Statistics and Its Interface*, 14:73 – 81, 2021.

[17] R. Vega, L. Flores and R. Greiner. SIMLR: Machine Learning inside the SIR Model for COVID-19 Forecasting. *Forecasting*, 4(1):72 – 94, 2022.

[18] H. Bastani, K. Drakopoulos, V. Gupta and et al. Efficient and targeted COVID-19 border testing via reinforcement learning. *Nature*, 599:108 – 113, 2021.

[19] S. A. Alanazi, M. M. Kamruzzaman, M. Alruwaili and et al. Measuring and Preventing COVID-19 Using the SIR Model and Machine Learning in Smart Health Care. *Journal of Healthcare Engineering*, 2020-8857346, 2020.

[20] G. Perakis, D. Singhvi, O. S. Lami, and et al. COVID-19: A multiwave SIR-based model for learning waves. *Production and Operations Management*, (13681), 2022.

[21] S. Chowdhury, S. Roychowdhury and I. Chaudhuri. Universality and herd immunity threshold : Revisiting the SIR model for COVID-19. *International Journal of Modern Physics C*, 3(6), 2021.

[22] W. Choi and E. Shim. Optimal strategies for social distancing and testing to control COVID-19. *Journal of Theoretical Biology*, 512(110568), 2021.

[23] D. Acemoglu, A. Fallah, A. Giometto and et al. Optimal adaptive testing for epidemic control: combining molecular and serology tests. arXiv:2101.00773, 2021.

[24] L. Abraham, G. Becigneul and B. Scholkopf. Crackovid: Optimizing Group Testing. arXiv:2005.06413, 2020.

[25] C. Tsay, F. Lejarza, M. Stadtherr and et al. Modeling, state estimation, and optimal control for the US COVID-19 outbreak. *Scientific reports*, 10(10711), 2020.

[26] F. Piguillem and L. Shi. Optimal COVID-19 quarantine and testing policies. *Nature Communications*, 12(356), 2021.

[27] M. Tanaka K. Kuga and J. Tanimoto. Pair approximation model for the vaccination game: predicting the dynamic process of epidemic spread and individual actions against contagion. *Proceedings of the Royal Society A*, 477(2246):20200769, 2021.

[28] K. Kuga K. Kabir and J. Tanimoto. The impact of information spreading on epidemic vaccination game dynamics in a heterogeneous complex network-a theoretical approach. *Chaos, Solitons & Fractals*, 132:109548, 2020.

[29] K. Kabir and J. Tanimoto. Evolutionary vaccination game approach in metapopulation migration model with information spreading on different graphs. *Chaos, Solitons & Fractals*, 120:41–55, 2019.

[30] L. Willem, S. Abrams, P. J. K. Libin and et al. The impact of contact tracing and household bubbles on deconfinement strategies for COVID-19. *Nature Communications*, 12(1524), 2021.

[31] J. Kim, X. Chen, H. Nikpey and et al. Tracing and testing multiple generations of contacts to COVID-19 cases: cost-benefit tradeoffs. medRxiv: 20205047, 2020.

[32] A. Aleta, D. Martin-Corral, A. Piontti and et al. Modelling the impact of testing, contact tracing and household quarantine on second waves of COVID-19. *Nature Human Behaviour*, 4:964–971, 2020.

[33] J. Hellewell, S. Abbott, A. Gimma and et al. Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. *The Lancet Global Health*, 8(4), 2020.

[34] A. Kucharski, P. Klepac, A. Conlan and et al. Effectiveness of isolation, testing, contact tracing, and physical distancing on reducing transmission of SARS-CoV-2 in different settings: a mathematical modelling study. *The Lancet Infectious Diseases*, 20(10), 2020.

[35] P. Auer, N. Cesa-Bianchi and P. Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47:235–256, 2002.

[36] S. Agrawal and N. Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23, pages 1–26, 2012.

[37] S. Agrawal and N. Goyal. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[38] K. Madhama and T. Murata. A multi-armed bandit approch for exploring partially observed networks. *Applied Network Science*, 4(26):1–18, 2019.

[39] M. Bilgic, L. Mihalkova and L. Getoor. Active learning for networked data. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 79–86, 2010.

[40] X. Wang and R. Garnett and J. Schneider. Active search on graphs. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 731–738, 2013.

[41] Y. Ma and T. K. Huang and J. Schneider. Active search and bandits on graphs using sigma-optimality. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 542 – 551, 2015.

[42] R. Garnett, Y. Krishnamurthy, D. Wang and et al. Bayesian optimal active search on graphs. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 843–850, 2011.

[43] D. Zhao, J. Liu, R. Wu and et al. Data-Efficient Reinforcement Learning Using Active Exploration Method. In *International Conference on Neural Information Processing*, pages 265–276, 2018.

[44] Y. Burda, H. Edwards, A. Storkey and et al. Exploration by random network distillation. arXiv:1810.12894, 2018.

[45] M. Bellemare, S.Srinivasan, G. Ostrovski and et al. Unifying count-based exploration and intrinsic motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1479–1487, 2016.

[46] R. Singh, F. Liu and N. B. Shroff. A Partially Observable MDP Approach for Sequential Testing for Infectious Diseases such as COVID-19. arXiv:2007.13023, 2020.

[47] H. Grushka-Cohen, R. Cohen, B. Shapira and et al. A framework for optimizing COVID-19 testing policy using a Multi Armed Bandit approach. arXiv:2007.14805, 2020.

[48] L. Kaelbling and M. Littman and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[49] G. Monahan. State of the Art—A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms. *Management Science*, 28(1):1–16, 1982.

[50] G. Walter and M. Contreras. *Compartmental Modeling with Networks*. Birkhauser, Boston, MA, 1999.

[51] D. Topkis. *Supermodularity and Complementarity*. Princeton University Press, 1998.

[52] V. Ilev. An approximation guarantee of the greedy descent algorithm for minimizing a supermodular set function. *Discrete Applied Mathematics*, 114:131–146, 2001.

[53] D. L. Hansen, B. Shneiderman, M. A. Smith and et al. *Analyzing Social Media Networks with NodeXL (Second Edition)*. Morgan Kaufmann, 2020.

[54] S. Ma, J. Zhang, M. Zeng and et al. Epidemiological parameters of coronavirus disease 2019: a pooled analysis of publicly reported individual data of 1155 cases from seven countries. Medrxiv: https://doi.org/10.1101/2020.03.21.20040329, 2020.

[55] A. Byrne, D. McEvoy, A. Collins and et al. Inferred duration of infectious period of SARS-CoV-2: rapid scoping review and analysis of available evidence for asymptomatic and symptomatic COVID-19 cases. *BMJ Open*, 10, 2020.

[56] CDC. The U.S. Centers for Disease Control and Prevention (CDC). https://www.cdc.gov/coronavirus/2019-ncov/php/contact579tracing/contact-tracing-plan/contact-tracing.html, Sept 2020.

[57] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(4):440–442, 1998.

[58] C. Lee and D. J. Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(122), 2019.

[59] P. Sapiezynski, A. Stopczynski, D. D. Lassen and et al. Interaction data from the Copenhagen Networks Study. *Scientific Data*, 6(1):1–10, 2019.

[60] R. D. Shachter. Bayes-Ball: The Rational Pastime (for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams). arXiv: 1301.7412, 2013.

[61] M. Jordan. An Introduction to Probabilistic Graphical Models. https://people.eecs.berkeley.edu/ jordan/prelims, 2003.

A Proof of Lemma 1

Note that a node is counted in $C^\pi(t)$ once it has been infected. Then, on day $t+1$, $C^\pi(t+1)$ increases (comparing to $C^\pi(t)$) only because some susceptible nodes are infected by infectious nodes and are in the latent state for the first time.

After testing, positive nodes in $\mathcal{K}^\pi(t)$ would not infect others because they are quarantined, and negative nodes would not infect others due to the model assumptions. Hence

$$C^\pi(t+1) = C^\pi(t) + \sum_{i \in \mathcal{V}(t)} F_i(\mathcal{V}(t) \backslash \mathcal{K}^\pi(t); t).$$

Taking the expectation on both sides, we obtain the desired result.

B Proof of Theorem 1

To show $S\Big(\mathcal{K}^\pi(t); t\Big)$ defined in (7) is a supermodular function. It suffices to show that for any $\mathcal{A} \subset \mathcal{B} \subset \mathcal{V}(t)$, and for $x \in \mathcal{V}(t) \backslash \mathcal{B}$, we have

$$S\Big(\mathcal{A} \cup \{x\}; t\Big) - S\Big(\mathcal{A}; t\Big) \leq S\Big(\mathcal{B} \cup \{x\}; t\Big) - S\Big(\mathcal{B}; t\Big). \tag{20}$$

Then, it suffices to show for any $i \in \mathcal{V}(t)$,

$$\mathbb{E}[F_i\Big(\mathcal{A} \cup \{x\}; t\Big)] - \mathbb{E}[F_i\Big(\mathcal{A}; t\Big)] \leq \mathbb{E}[F_i\Big(\mathcal{B} \cup \{x\}; t\Big)] - \mathbb{E}[F_i\Big(\mathcal{B}; t\Big)]. \tag{21}$$

Now, we consider three cases.

Case 1. If $\mathcal{A} \cap \partial_i(t) = \mathcal{B} \cap \partial_i(t)$, then from (6), the LHS and RHS in (21) are exactly the same. Hence, (21) holds.

Case 2. If $\mathcal{A} \cap \partial_i(t) \subset \mathcal{B} \cap \partial_i(t)$, and $x \notin \partial_i$, then from (6), $f_i(\mathcal{A} \cup \{x\}) = f_i(\mathcal{A})$ and $f_i(\mathcal{B} \cup \{x\}) = f_i(\mathcal{B})$. Hence (21) holds.

Case 3. If $\mathcal{A} \cap \partial_i(t) \subset \mathcal{B} \cap \partial_i(t)$, and $x \in \partial_i(t)$, let $\mathcal{Y} = \Big(\mathcal{B} \cap \partial_i(t)\Big) \backslash \Big(\mathcal{A} \cap \partial_i(t)\Big)$. Here $x \notin \mathcal{Y}$. From (6), we can compute

$$\begin{aligned}
&\mathbb{E}[F_i(\mathcal{A} \cup \{x\}; t)] - \mathbb{E}[F_i(\mathcal{A}; t)] \\
&= v_S^{(i)}(t) \prod_{j \in \partial_i(t) \backslash (\mathcal{A} \cup \{x\})} \Big(1 - \beta v_I^{(j)}(t)\Big) \\
&\times \Big(1 - \prod_{j \in \partial_i(t) \cap (\mathcal{A} \cup \{x\})} (1 - \beta v_I^{(j)}(t)) - (1 - \beta v_I^{(x)}(t))\Big(1 - \prod_{j \in \partial_i(t) \cap \mathcal{A}} (1 - \beta v_I^{(j)}(t))\Big)\Big),
\end{aligned}$$

which implies

$$\begin{aligned}
&\mathbb{E}[F_i(\mathcal{A} \cup \{x\}; t)] - \mathbb{E}[F_i(\mathcal{A}; t)] \\
&= v_S^{(i)}(t) \prod_{j \in \partial_i(t) \backslash (\mathcal{A} \cup \{x\})} \Big(1 - \beta v_I^{(j)}(t)\Big) \\
&\times \Big(\beta v_I^{(x)}(t) - \prod_{j \in \partial_i(t) \cap (\mathcal{A} \cup \{x\})} (1 - \beta v_I^{(j)}(t)) + \prod_{j \in \partial_i(t) \cap (\mathcal{A} \cup \{x\})} (1 - \beta v_I^{(j)}(t))\Big)\Big) \\
&= v_S^{(i)}(t) \prod_{j \in \partial_i(t) \backslash (\mathcal{A} \cup \{x\})} \Big(1 - \beta v_I^{(j)}(t)\Big) \beta v_I^{(x)}(t).
\end{aligned}$$

Similarly, note that $\Big(\mathcal{B} \cap \partial_i(t)\Big) = \Big(\mathcal{A} \cap \partial_i(t)\Big) \cup \mathcal{Y}$. We have

$$\begin{aligned}
&\mathbb{E}[F_i(\mathcal{B} \cup \{x\}; t)] - \mathbb{E}[F_i(\mathcal{B}; t)] \\
&= v_S^{(i)}(t) \prod_{j \in \partial_i(t) \backslash (\mathcal{A} \cup (\{x\} \cup \mathcal{Y}))} \Big(1 - \beta v_I^{(j)}(t)\Big) \beta v_I^{(x)}(t).
\end{aligned}$$

Thus,

$$\frac{\mathbb{E}[F_i(\mathcal{A} \cup \{x\}; t)] - \mathbb{E}[F_i(\mathcal{A}; t)]}{\mathbb{E}[F_i(\mathcal{B} \cup \{x\}; t)] - \mathbb{E}[F_i(\mathcal{B}; t)]} = \prod_{y \in \mathcal{Y}} \Big(1 - \beta v_I^{(y)}(t)\Big) \leq 1,$$

which implies $S\Big(\mathcal{TP}^\pi(t)\Big)$ is supmodular.

To show $S\Big(\mathcal{K}^\pi(t); t\Big)$ is an increasing monotone function on $\mathcal{K}^\pi(t)$, it suffices to show $\mathbb{E}[F_i\Big(\mathcal{K}^\pi(t); t\Big)]$ is an increasing monotone function on $\mathcal{K}^\pi(t)$ for any i.

For $\mathcal{A} \subset \mathcal{B}$, we have $\partial_i(t)\backslash\mathcal{B} \subset \partial_i(t)\backslash\mathcal{A}$, and $\mathcal{A} \cap \partial_i(t) \subset \mathcal{B} \cap \partial_i(t)$. Then

$$\prod_{j\in\partial_i(t)\backslash\mathcal{B}} \Big(1-\beta v_I^{(j)}(t)\Big) \geq \prod_{j\in\partial_i(t)\backslash\mathcal{A}} \Big(1-\beta v_I^{(j)}(t)\Big)$$
$$\prod_{j\in\mathcal{B}\cap\partial_i(t)} (1-\beta v_I^{(j)}(t)) \leq \prod_{j\in\mathcal{A}\cap\partial_i(t)} (1-\beta v_I^{(j)}(t)),$$

and thus, from (6), we have $\mathbb{E}[F_i\Big(\mathcal{A};t\Big)] \leq \mathbb{E}[F_i\Big(\mathcal{B};t\Big)]$.

C Complexity of Algorithm 1

First of all, we consider the complexity of (6). Suppose $\{\underline{v}_i(t)\}_{i\in\mathcal{V}(t)}$ is given for every day t. For any $\mathcal{K}^\pi(t)$, the complexity of computing (6) is

$$1+|\partial_i(t)\backslash\mathcal{K}^\pi(t)|-1+1+|\partial_i(t)\backslash\mathcal{K}^\pi(t)|+|\partial_i(t)\cap\mathcal{K}^\pi(t)|-1+|\partial_i(t)\cap\mathcal{K}^\pi(t)| = 2|\partial_i(t)|.$$

Then, for any $\mathcal{K}^\pi(t)$, the complexity of computing $S\Big(\mathcal{K}^\pi(t);t\Big)$ is

$$2\sum_{j\in\mathcal{V}(t)} |\partial_j(t)|.$$

From Algorithm 1, in step i, the complexity is

$$\Big(N(t)-i+1\Big)\times 2\sum_{j\in\mathcal{V}(t)} |\partial_j(t)|.$$

And in total we have $\Big(N(t)-K^\pi(t)\Big)$ steps, therefore, on day t the complexity of Algorithm 1 is

$$\sum_{i=0}^{N(t)-K^\pi(t)} 2\Big(N(t)-i+1\Big)\sum_{j\in\mathcal{V}(t)} |\partial_j(t)|.$$

Recall that the time horizon is T, then the total complexity of Algorithm 1 is

$$\sum_{t=0}^{T-1}\sum_{i=0}^{N(t)-K^\pi(t)} 2\Big(N(t)-i+1\Big)\sum_{j\in\mathcal{V}(t)} |\partial_j(t)|.$$

Note that

$$\sum_{i=1}^{N(t)-K^\pi(t)} 2\Big(N(t)-i+1\Big) \leq O\Big(N^2(t)\Big)$$
$$\sum_{i\in\mathcal{V}(t)} |\partial_i(t)| \leq O\Big(N^2(t)\Big).$$

Then, the total complexity is bounded by

$$O\Big(\sum_{t=0}^{T-1} N^4(t)\Big).$$

D Proof of Lemma 2

As defined in [52] (an equivalent definition of footnote 3), consider a finite set I, $f : 2^I \to \mathbb{R}$ is a supermodular function if for all $X, Y \subset I$,

$$f(X \cup Y) + f(X \cap Y) \geq f(X) + f(Y). \tag{22}$$

Following the supermodularity of function $S(\cdot)$ as shown in Theorem 1, set $X = \mathcal{V}(t) \backslash \mathcal{K}^\pi(t)$ and $Y = \mathcal{K}^\pi(t)$ in (22), we have

$$S\Big(\mathcal{V}(t) \backslash \mathcal{K}^\pi(t); t\Big) \leq S\Big(\mathcal{V}(t); t\Big) - S\Big(\mathcal{K}^\pi(t); t\Big). \tag{23}$$

Again, set $X = \mathcal{K}^\pi(t) \backslash \{i\}$ and $Y = \{i\}$ in (22), and use (22) repeatedly to obtain:

$$S\Big(\mathcal{K}^\pi(t); t\Big) \geq \sum_{i \in \mathcal{K}^\pi(t)} S\Big(\{i\}; t\Big) = \sum_{i \in \mathcal{K}^\pi(t)} r_i(t). \tag{24}$$

Substituting (24) in (23), we obtain

$$S\Big(\mathcal{V}(t) \backslash \mathcal{K}^\pi(t); t\Big) \leq S\Big(\mathcal{V}(t); t\Big) - \sum_{i \in \mathcal{K}^\pi(t)} r_i(t).$$

E Local Transition Equations

In this section, we will describe the local transition matrix $\mathtt{P}_i\Big(\{\underline{v}_j(t)\}_{j \in \partial_i^+(t)}\Big)$ used in (12). The state of each node evolves as follows: (i) if node i is susceptible on day t, then it might be infected by its neighbors in $\partial_i(t)$; (ii) an infectious node remains in the latent state with probability $1 - \lambda$, and changes state to the infectious state (I) with probability λ; (iv) if node i is in state I, it will recover after a geometric distribution with parameter γ. Let $\xi_i(t) = 1 - \prod_{m \in \partial_i(t)} \Big(1 - v_I^{(m)}(t)\beta\Big)$. In particular, define $\xi_i(t) = 0$ if $\partial_i(t) = \varnothing$. Then, the probabilities of nodes being in different states evolve in time as follows:

$$v_I^{(i)}(t+1) = v_I^{(i)}(t)(1-\gamma) + v_L^{(i)}(t)\lambda \tag{25}$$

$$v_L^{(i)}(t+1) = v_L^{(i)}(t)(1-\lambda) + v_S^{(i)}(t)\xi_i(t) \tag{26}$$

$$v_R^{(i)}(t+1) = v_R^{(i)}(t) + v_I^{(i)}(t)\gamma \tag{27}$$

$$v_S^{(i)}(t+1) = v_S^{(i)}(t)\Big(1 - \xi_i(t)\Big). \tag{28}$$

Note that row vector $\underline{v}_i(t)$ is defined in (4). Collecting (25) - (28), we define the local transition probability matrix as given below:

$$\mathtt{P}_i\Big(\{\underline{v}_j(t)\}_{j \in \partial_i^+(t)}\Big) = \begin{bmatrix} (1-\gamma) & 0 & \gamma & 0 \\ \lambda & 1-\lambda & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \xi_i(t) & 0 & 1-\xi_i(t) \end{bmatrix}. \tag{29}$$

and we obtain (12).

F Proofs of (13) and (14)

First of all, we give the following definition.

Definition 1. *Let X be a random variable and $\mathcal{B}$ be an event. Define $X|_{\mathcal{B}}$ as the random variable X given $\mathcal{B}$; i.e.,*

$$\Pr\Big(X|_{\mathcal{B}} = x\Big) = \Pr\Big(X = x|\mathcal{B}\Big). \tag{30}$$

For brevity, let us define

$$\theta_i(t) = \sigma_i(t)|_{\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}}, \quad \zeta_i(t) = \sigma_i(t)|_{\{\underline{Y}(\tau)\}_{\tau=1}^{t}}.$$

We thus have

$$u_x^{(i)}(t) = \Pr\Big(\theta_i(t) = x\Big), \quad w_x^{(i)}(t) = \Pr\Big(\zeta_i(t) = x\Big).$$

Recall that

$$\underline{v}_i(t) = \Big[v_x^{(i)}(t)\Big]_{x\in\mathcal{X}}, \; v_x^{(i)}(t) = \Pr\Big(\sigma_i(t) = x\Big).$$

Then, (12) can be re-written as

$$\Pr\Big(\sigma_i(t+1) = x_i'\Big) = \Pr\Big(\sigma_i(t) = x_i\Big)\mathtt{P}_i\Big(\{\sigma_j(t)\}_{j\in\partial_i^+(t)} = \{x_j\}_{j\in\partial_i^+(t)}\Big), \tag{31}$$

where $x_i', \{x_j\}_{j\in\partial_i^+(t)} \in \mathcal{X}$. Conditioning both sides of (31) on $\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}$, state variables $\sigma_i(t)$ and $\sigma_i(t-1)$ in (31) can be replaced by $\theta_i(t)$ and $\zeta_i(t-1)$, respectively, to obtain

$$\underline{u}_i(t) = \underline{w}_i(t-1) \times \mathtt{P}_i\Big(\{\underline{w}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big), \tag{32}$$

which gives (13). In addition, define

$$\phi_i(t) = \sigma_i(t)|_{\{\underline{Y}(\tau)\}_{\tau=1}^{t+1}},$$

and

$$\begin{aligned} &\underline{e}_i(t-1) = (e_x^{(i)}(t-1), x \in \mathcal{X}), \\ &e_x^{(i)}(t-1) = \Pr\Big(\phi_i(t-1) = x\Big). \end{aligned} \tag{33}$$

This notation implies

$$\phi_i(t-1) = \theta_i(t-1)|_{\underline{Y}(t)}. \tag{34}$$

Similarly, conditioning both sides of (32) on $\underline{Y}(t)$, we find

$$\underline{w}_i(t) = \underline{e}_i(t-1) \times \tilde{\mathtt{P}}_i\Big(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big), \tag{35}$$

which gives (14). $\tilde{P}_i(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)})$ is obtained in the following subsection.

F.1 Computing the transition probability matrix $\tilde{P}_i(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)})$

Note that $\tilde{P}_i(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)})$ is not the same as $P_i(\{\underline{w}_j(t)\}_{j\in\partial_i^+(t)})$. *This is because "future" observations were available in* $\tilde{P}_i(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)})$. To get $\tilde{\mathtt{P}}_i\Big(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big)$, we split the nodes $\mathcal{V}(t)$ into two classes of nodes: (i) nodes who do not get new observations and (ii) nodes who get new observations. $\tilde{P}_i(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)})$ is obtained by the following rules. For the first class of nodes, the local transition matrix in (35), i.e., $\tilde{P}_i(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)})$, is the same as that in (32). However, for the second class of nodes, the local transition matrices are changed accordingly because of the new observations. Let $[A]_{\{i,:\}}$ be the ith row of matrix A, and q_i be a 1×4 vector with the ith element being one and the rest zero. For brevity, denote the local transition matrices in (32) and (35) by $\mathtt{P}_i(t-1)$ and $\tilde{\mathtt{P}}_i(t-1)$, respectively. We have the following three cases:

(i) If node i is not observed, then node i does not have new observation and we have

$$\tilde{P}_i(t-1) = P_i(t-1). \tag{36}$$

(ii) If $Y_i(t) = 0$, then node i is not infectious in day t with probability 1. The local transition matrix is changed to

$$[\tilde{\mathtt{P}}_i(t-1)]_{\{j,:\}} = \begin{cases} q_3 & j=1 \\ q_2 & j=2 \\ [\mathtt{P}_i(t-1)]_{\{j,:\}} & \text{otherwise} \end{cases}. \tag{37}$$

(iii) If $Y_i(t) = 1$, then node i is infectious in day t with probability 1. The local transition matrix is changed to

$$[\tilde{\mathtt{P}}_i(t-1)]_{\{j,:\}} = \begin{cases} q_1 & j=1 \\ q_1 & j=2 \\ [\mathtt{P}_i(t-1)]_{\{j,:\}} & \text{otherwise} \end{cases}. \tag{38}$$

G Proofs of (16) and (17)

Using new observations, we aim to move backward in time and update our belief (posterior probability) in previous time slots. Define a *truncation number* g and suppose that $\{\underline{Y}(t)\}$ affects the posterior probabilities from day t to day $t-g$. We call day $t-g$ the *truncation day* associated with day t. To get accurate posterior probabilities in every day, we need to set $g=t$ on every day t and track back to the initial time. However, the influence weakens as time elapses backwards, and for computation tractability, we continue under the following assumption where $g=1$. Recall that $\zeta_i(t) = \sigma_i(t)|_{\{\underline{Y}(\tau)\}_{\tau=1}^t}$.

Assumption 1. *On the truncation day* $(t-g)$, $\{\zeta_i(t-g)\}_i$ *are independent over* i. *In the following, the truncation number is assumed to be* $g=1$.

Remark 3. *In Assumption 1, we assume that the nodes' states* $\zeta(t-g)$ *(in the posterior probability space on day* $t-g$*) are independent. This assumption is only used at time* t *of our*

probability update in a moving window kind of way. It provides us with a truncation time for each backward step. In particular, under Assumption 1, once we get the observations $\underline{Y}(t)$, we do the backward step and truncate at time $t-g$. For example, in the trivial case of $g=t$, the assumption holds. This assumption does not impose independence on the state of the nodes, but only in the posterior space at a specific time. In a sense, in the process of propagating information back to time $t-g$, we are assuming that there is no further correlation between time $t-g-1$ and time $t-g$ worthwhile to exploit given observations at time t. Naturally, as g gets larger and larger, our framework and calculations become more precise but this comes at a huge computational cost. The idea behind truncating the backward step lies in the observation that the impact of the testing results at time t in inferring about the nodes' probabilities at time $t-g$ vanishes as g gets large. For simplicity of derivations and to have manageable complexity, we set $g=1$. The idea and the derivations can be generalized in a straightforward manner to larger g.

Note that the posterior probabilities on day $t-1$, $\underline{w}_i(t-1)$, $i \in \mathcal{V}(t-1)$, are assumed known (and are conditioned on the history of observations $\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}$). The probability vector $\underline{e}_i(t-1)$ is the new posterior probability at time $t-1$ which is updated (from $\underline{w}_i(t-1)$) based on new observations $\underline{Y}(t)$. In other words, we infer about the previous state of the nodes given new observations at present time.

To obtain $\{\underline{w}_i(t)\}_{i\in\mathcal{V}(t)}$, it suffices to obtain $\underline{e}_i(t-1)$ and the corresponding local transition matrix $\tilde{\mathsf{P}}_i\Big(\{\underline{e}_j(t-1)\}_{j\in\partial_i^+(t-1)}\Big)$, see (14). Note that the posterior probabilities $\underline{w}_i(t-1)$, $i \in \mathcal{V}(t-1)$, which are calculated based on $\underline{Y}(t-1)$, are known. The vector $\underline{e}_i(t-1)$ is the new posterior probability which is updated based on $\underline{Y}(t)$ and $\underline{w}_i(t-1)$.

Equation (16), which we aim to prove, simply follows from Definition 1, (33)-(34), and Bayes rule:

$$\begin{aligned} e_x^{(i)}(t-1) &= \Pr\Big(\zeta_i(t-1)=x|\underline{Y}(t)\Big) \\ &= \frac{\Pr\Big(\underline{Y}(t)|\zeta_i(t-1)=x\Big)w_x^{(i)}(t-1)}{\Pr\Big(\underline{Y}(t)\Big)}. \end{aligned} \tag{39}$$

To find $\Pr\Big(\underline{Y}(t)|\zeta_i(t-1)=x\Big)$, and establish (17), we now proceed as follows. We introduce $\{\theta_j(t)\}_{j\in\mathcal{O}(t)}$ into (39). In particular, we have

$$\Pr\Big(\underline{Y}(t)|\zeta_i(t-1)=x\Big) = \sum_{\theta_j(t),\ j\in\mathcal{O}(t)} \Pr\Big(\{\theta_j(t)\}_{j\in\mathcal{O}(t)}, \underline{Y}(t)|\zeta_i(t-1)=x\Big)$$

By the chain rule of conditional probability,

$$\begin{aligned} &\Pr\Big(\underline{Y}(t)|\zeta_i(t-1)=x\Big) \\ &= \sum_{\theta_j(t),\ j\in\mathcal{O}(t)} \Pr\Big(\underline{Y}(t)|\{\theta_j(t)\}_{j\in\mathcal{O}(t)}, \zeta_i(t-1)=x\Big) \times \Pr\Big(\{\theta_j(t)\}_{j\in\mathcal{O}(t)}|\zeta_i(t-1)=x\Big). \end{aligned}$$

From (15), $\{\zeta_j(t)\}_{j\in\mathcal{V}(t)}$ and $\{\theta_j(t)\}_{j\in\mathcal{V}(t)}$ are variables defined by $\{\sigma_j(t)\}_{j\in\mathcal{V}(t)}$ in posterior spaces of $\{\underline{Y}(\tau)\}_{\tau=1}^{t}$ and $\{\underline{Y}(\tau)\}_{\tau=1}^{t-1}$, respectively. Since $\underline{Y}(t)$ is a deterministic function of

$\{\sigma_j(t)\}_{j\in\mathcal{O}(t)}$, and hence $\{\theta_j(t)\}_{j\in\mathcal{O}(t)}$, then $\underline{Y}(t)$ is independent of $\zeta_i(t-1)$ given $\{\theta_j(t)\}_{j\in\mathcal{O}(t)}$. In addition, the testing result $Y_j(t)$ (on day t) of node j only depends on its state, i.e., given $\theta_j(t)$, the testing results are determined. Therefore, we have

$$\begin{aligned}&\Pr\Big(\underline{Y}(t)|\{\theta_j(t)\}_{j\in\mathcal{O}(t)},\zeta_i(t-1)=x\Big)\\=&\Pr\Big(\underline{Y}(t)|\{\theta_j(t)\}_{j\in\mathcal{O}(t)}\Big)=\prod_{j\in\mathcal{O}(t)}\Pr\Big(Y_j(t)|\theta_j(t)\Big).\end{aligned}$$

The product above is an indicator which takes values on $\{0,1\}$. We can thus re-write it as follows:

$$\Pr\Big(\underline{Y}(t)|\{\theta_j(t)\}_{j\in\mathcal{O}(t)},\zeta_i(t-1)=x\Big)\triangleq\delta(\{Y_j(t),\theta_j(t)\}_{j\in\mathcal{O}(t)}).$$

where

$$\delta(\{Y_j(t),\theta_j(t)\}_{j\in\mathcal{O}(t)})=1$$

if the pairs $\{Y_j(t),\theta_j(t)\}_{j\in\mathcal{O}(t)}$ are consistent, and

$$\delta(\{Y_j(t),\theta_j(t)\}_{j\in\mathcal{O}(t)})=0$$

otherwise.

Next, define

$$\Theta_i(t)=\{j|j\in\partial_k^+(t-1),k\in\mathcal{O}(t)\}\backslash\{i\}$$

to represent the neighbors (in day $t-1$) of nodes in $\mathcal{O}(t)$ excluding node i. Then,

$$\Pr\left(\{\theta_j(t)\}_{j\in\mathcal{O}(t)}|\zeta_i(t-1)=x\right)=\sum_{\zeta_l(t-1),\ l\in\Theta_i(t)}\Pr\left(\{\theta_j(t)\}_{j\in\mathcal{O}(t)},\{\zeta_l(t-1)\}_{l\in\Theta_i(t)}|\zeta_i(t-1)=x\right).\tag{40}$$

By the chain rule of conditional probability,

$$\begin{aligned}&\Pr\Big(\{\theta_j(t)\}_{j\in\mathcal{O}(t)}|\zeta_i(t-1)=x\Big)\\=&\sum_{\zeta_l(t-1),\ l\in\Theta_i(t)}\Pr\Big(\{\theta_j(t)\}_{j\in\mathcal{O}(t)}|\{\zeta_l(t-1)\}_{l\in\Theta_i(t)},\zeta_i(t-1)=x\Big)\times\Pr\Big(\{\zeta_l(t-1)\}_{l\in\Theta_i(t)}|\zeta_i(t-1)=x\Big).\end{aligned}$$

Given $\{\zeta_l(t-1)\}_{l\in\Theta_i(t)}\cup\{\zeta_i(t-1)\}$, $\{\theta_j(t)\}_{j\in\mathcal{O}(t)}$ are independent. We thus have

$$\begin{aligned}&\Pr\Big(\{\theta_j(t)\}_{j\in\mathcal{O}(t)}|\{\zeta_l(t-1)\}_{l\in\Theta_i(t)},\zeta_i(t-1)=x\Big)\\=&\prod_{j\in\mathcal{O}(t)}\Pr\Big(\theta_j(t)|\{\zeta_l(t-1)\}_{l\in\Theta_i(t)},\zeta_i(t-1)=x\Big)\\=&\prod_{j\in\mathcal{O}(t)}\Pr\Big(\theta_j(t)|\{\zeta_l(t-1)\}_{l\in\partial_j^+(t-1)\backslash\{i\}},\zeta_i(t-1)=x\Big).\end{aligned}$$

Based on Assumption 1,

$$\Pr\Big(\{\zeta_l(t-1)\}_{l\in\Theta_i}|\zeta_i(t-1)=x\Big)=\prod_{l\in\{\Theta_i(t)\}}\Pr\Big(\zeta_l(t-1)\Big).$$

Therefore,

$$\begin{aligned}
&\Pr\Big(\underline{Y}(t)|\zeta_i(t-1)=x\Big)\\
=&\sum_{\theta_j(t),\ j\in\mathcal{O}(t)}\delta(\{Y_j(t),\theta_j(t)\}_{j\in\mathcal{O}(t)})\\
\times&\sum_{\zeta_l(t-1)}\prod_{j\in\mathcal{O}(t)}\Pr\Big(\theta_j(t)|\{\zeta_l(t-1)\}_{l\in\partial_j^+(t-1)}\backslash\{i\},\zeta_i(t-1)=x\Big)\\
&\times\prod_{l\in\{\Theta_i(t)\}}\Pr\Big(\zeta_l(t-1)\Big).
\end{aligned}\tag{41}$$

Denote $\{x_j\}_{j\in\mathcal{O}(t)}$ as a realization of $\{\theta_j(t)\}_{j\in\mathcal{O}(t)}$ and $\{y_l\}_{l\in\Theta_i(t)}$ as a realization of $\{\zeta_l(t-1)\}_{l\in\Theta_i(t)}$. Then,

$$\begin{aligned}
&\Pr\Big(\underline{Y}(t)|\zeta_i(t-1)=x\Big)\\
=&\sum_{\{x_j\}_{j\in\mathcal{O}(t)}}\delta(\{Y_j(t),x_j\}_{j\in\mathcal{O}(t)})\\
&\times\sum_{\{y_l\}_{l\in\Theta_i(t)}}\prod_{j\in\mathcal{O}(t)}\Pr\Big(x_j|\{y_l\}_{l\in\partial_j^+(t-1)\backslash\{i\}},\zeta_i(t-1)=x\Big)\\
&\times\prod_{l\in\{\Theta_i(t)\}}\Pr\Big(\zeta_l(t-1)=y_l\Big).
\end{aligned}$$

Denote

$$\begin{aligned}
&\rho\Big(\{x_j\}_{j\in\mathcal{O}(t)},x\Big)\\
=&\sum_{\{y_l\}_{l\in\Theta_i(t)}}\prod_{j\in\mathcal{O}(t)}\Pr\Big(x_j|\{y_l\}_{l\in\partial_j^+(t-1)\backslash\{i\}},\zeta_i(t-1)=x\Big)\times\prod_{l\in\Theta_i(t)}\Pr\Big(\zeta_l(t-1)=y_l\Big).
\end{aligned}\tag{42}$$

Then,

$$\Pr\Big(\underline{Y}(t)|\zeta_i(t-1)=x\Big)=\sum_{x_j\in\mathcal{X},j\in\mathcal{O}(t)}\delta(\{Y_j(t),x_j\}_{j\in\mathcal{O}(t)})\rho\Big(\{x_j\}_{j\in\mathcal{O}(t)},x\Big).\tag{43}$$

Based on Assumption 1, we can further simplify (43). Consider node i, $\underline{Y}(t)$ can be split into $\underline{Y}_{i,1}(t)$ and $\underline{Y}_{i,2}(t)$, where $\underline{Y}_{i,1}(t)$ is the observations of the set $\mathcal{O}(t)\cap\partial_i^+(t-1)$, and $\underline{Y}_{i,2}(t)$ is the observations of the rest of the nodes. Note that $\underline{Y}_{i,1}(t)\cup\underline{Y}_{i,2}(t)=\underline{Y}(t)$ and $\underline{Y}_{i,1}(t)\cap\underline{Y}_{i,2}(t)=\varnothing$.

Lemma 3. *Conditioned on $\underline{Y}_{i,1}(t)$, $\zeta_i(t-1)$ is independent of $\underline{Y}_{i,2}(t)$.*

Proof. To show Lemma 3, we use the structured belief network as defined in [60]. $\zeta_j(t)$ is the random variable associated with node j. Note that $Y_j(t)$ is the test result of $\zeta_j(t)$ on day t. Now, we consider $j\in\Big(\mathcal{O}(t)\backslash(\mathcal{O}(t)\cap\partial_i^+(t-1))\Big)$. By [60, Theorems 1] and Bayes ball algorithm defined in [61, Section 2], we investigate the following two cases.

(i) For any $j\in\Big(\mathcal{O}(t)\backslash(\mathcal{O}(t)\cap\partial_i^+(t-1))\Big)$ with $Y_j(t)=1$, the corresponding state $\zeta_j(t)$ is determined (which is I). Then, probabilities conditioning on $Y_j(t)$ is equivalent

to (equal to) probabilities conditioning on $\zeta_j(t)$. By Bayes ball algorithm [60, 61], the information (the ball) is blocked at $\zeta_j(t)$ when the information (the ball) reaches $\zeta_j(t)$, which implies the information (the ball) can not reach $\zeta_i(t-1)$.

(ii) For any $j \in \left(\mathcal{O}(t)\backslash(\mathcal{O}(t) \cap \partial_i^+(t-1))\right)$ with $Y_j(t) = 0$, $\zeta_j(t)$ is not determined. By Bayes ball algorithm [60, 61], when the information (the ball) reaches $\zeta_j(t)$, it can traverse $Y_j(t)$ when blocking $Y_j(t)$ (conditioning on $Y_j(t)$). However, by Assumption 1, $\zeta_i(t-1)$ and $\zeta_j(t-1)$ are independent, so any path between $\zeta_i(t-1)$ and $\zeta_j(t-1)$ is blocked, including the path $\zeta_j(t-1) \leftrightarrow \zeta_j(t) \leftrightarrow Y_j(t) \leftrightarrow \zeta_j(t) \leftrightarrow \zeta_i(t-1)$. Thus, the information (the ball) can not reach $\zeta_i(t-1)$.

A simple example is given in Figure 7: Let $Y_1(t) = 0$ and $Y_2(t) = 1$. Given $Y_1(t)$ and $Y_2(t)$, $Y_3(t)$ is independent of $\zeta_1(t-1)$. □

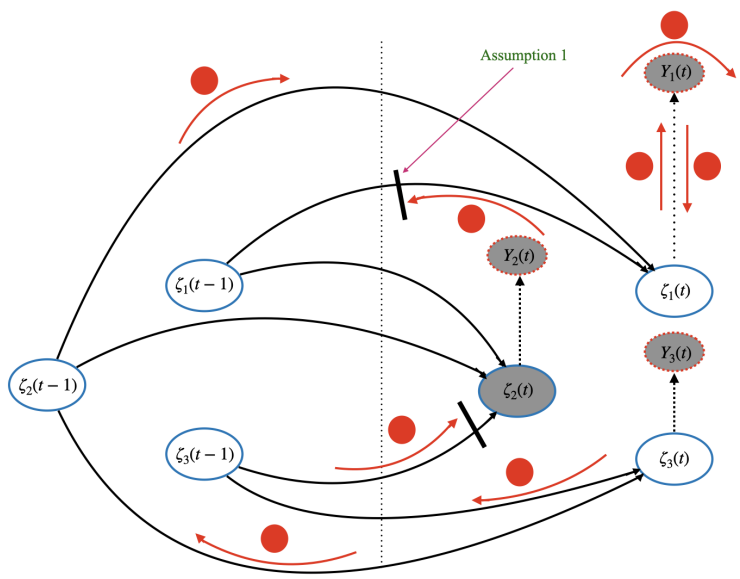

Figure 7: Bayes ball algorithm in the network of 3 nodes. The terms on which we have conditioning are shaded gray and are equivalently blocked.

From Lemma 3,

$$e_x^{(i)}(t-1) = \Pr\left(\zeta_i(t-1) = x|\underline{Y}(t)\right) = \Pr\left(\zeta_i(t-1) = x|\underline{Y}_{i,1}(t)\right). \tag{44}$$

We simplify (43) based on Lemma 3 or (44). From (44), denote the observations of nodes in $\partial_i^+(t-1)$ as $\underline{Y}_{\partial_i^+}(t)$, $\underline{Y}_{\partial_i^+}(t)$ is independent of $\zeta_i(t-1)$. Denote $\Psi_i(t) = \mathcal{O}(t) \cap \partial_i^+(t-1)$. Then, We can replace $\mathcal{O}(t)$ by $\Psi_i(t)$ in (16). Subsequently, denote $\Phi_i(t) = \{j|j \in \partial_k^+(t-1), k \in \Psi_i(t)\}\backslash\{i\}$, and we can replace $\Theta_i(t)$ by $\Phi_i(t)$ in (40). Thus, from (42) and (43), we respectively have

$$\begin{aligned}&\rho\Big(\{x_j\}_{j\in\Psi_i(t)}, x\Big)\\ &= \sum_{\{y_l\}_{l\in\Phi_i(t)}} \prod_{j\in\Psi_i(t)} \Pr\left(x_j|\{y_l\}_{l\in\partial_j^+(t-1)\backslash\{i\}}, \zeta_i(t-1) = x\right) \times \prod_{l\in\Phi_i(t)} \Pr\left(\zeta_l(t-1) = y_l\right)\end{aligned} \tag{45}$$

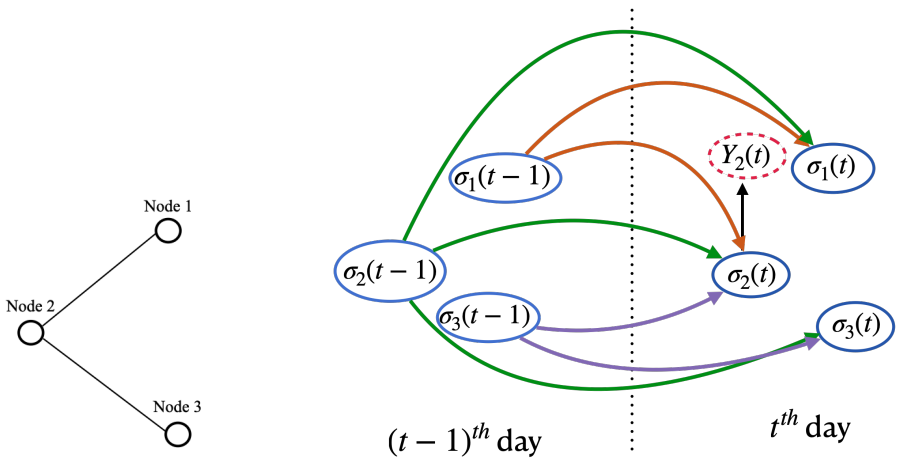

Figure 8: The original graph (left). The graphical model of states and observations (right)

and

$$\Pr\Big(\{Y_j(t)\}_{j\in\Psi_i(t)}|\zeta_i(t-1)=x\Big)=\sum_{x_j\in\mathcal{X},j\in\Psi_i(t)}\delta(\{Y_j(t),x_j\}_{j\in\Psi_i(t)})\rho\Big(\{x_j\}_{j\in\Psi_i(t)},x\Big) \tag{46}$$

which give the desired result (17).

H A Simple Example for Algorithm 4

In this section, we give a simple example to illustrate the ideas and steps of Algorithm 4. Besides, we compare our proposed algorithm (Algorithm 4) with the Naive approach discussed in Remark 2. Consider a simple network with three nodes. Node 1 has an edge with node 2, node 2 has an edge with node 3 (see Fig 8). Nodes 1 and 3 are symmetric and statistically identical, and node 2 has higher degree.

Consider the following situation: on the initial day (day 0), assume that nodes 1 and 3 are susceptible, and node 2 is infectious. On day 1, let node 2 be tested. Recall that we define the posterior probability vectors at the end of every day, and the prior probability vectors at the beginning of every day. Nodes' states change in the beginning of every day and testing is also done in the beginning of every day. Let the initial belief, i.e., the posterior probability $\underline{w}_i(0)$ and the prior probability $\underline{u}_i(0)$ on day 0, of the nodes be

$$\underline{w}_i(0)=[1/3,0,0,2/3],\quad i=1,2,3$$
$$\underline{u}_i(0)=[1/3,0,0,2/3],\quad i=1,2,3.$$

Day 0: No tests on day 0, the prior probabilities are updated by the forward step (**Step 2** in Algorithm 4).

Day 1: By **Step 2** in Algorithm 4, we have

$$\underline{u}_1(1)=[0.3144,0.0373,0.0633,0.5849]$$
$$\underline{u}_2(1)=[0.3559,0.0676,0.0633,0.5131]$$
$$\underline{u}_3(1)=[0.3144,0.0373,0.0633,0.5849].$$

After testing node 2, we know that node 2 is positive. We use the test result to infer about the state of the nodes in prior times. In particular, we update the posterior probability on day 0 ($\underline{w}_i(0)$). Denoting the updated posterior probability as $\underline{e}_i(0)$, by Step 1 in Algorithm 4, we find

$$\begin{aligned}\underline{e}_1(0) &= [0.3144, 0.0373, 0.0633, 0.5849]\\ \underline{e}_2(0) &= [0.9615, 0.0385, 0.0, 0.0]\\ \underline{e}_3(0) &= [0.3144, 0.0373, 0.0633, 0.5849].\end{aligned}$$

We can now say that at the end of day 0, node 2 was infectious with probability 0.9615 and it was in the latent state with probability 0.0385. Moreover, we see that the (posterior) infection probabilities of nodes 1 and 3 on day 0 have increased since they may have infected node 2 on day 0, i.e., $0.3517 = e_I^{(i)}(0) + e_L^{(i)}(0) > 1/3 = w_I^{(i)}(0) + w_L^{(i)}(0)$ with $i = 1, 3$. Next, we obtain the posterior probability on day 1. Recall that $\underline{w}_i(t)$ describes the posterior probability vector of node i at the end of day t. By Step 1 in Algorithm 4,

$$\begin{aligned}\underline{w}_1(1) &= [0.4008, 0.0, 0.1268, 0.4724]\\ \underline{w}_2(1) &= [0.90, 0.0, 0.10, 0.0]\\ \underline{w}_3(1) &= [0.4008, 0.0, 0.1268, 0.4724].\end{aligned}$$

One may wonder why the posterior probability is $[0.9, 0, 0.1, 0]$ rather than $[1, 0, 0, 0]$. This is because testing is done in the beginning of time t and the posterior probabilities are defined at the end of time slots t. The infected node may have recovered by the end of time $t = 1$ and this is reflected in the posterior probabilities computed.

Day 2: We can get the prior probability vectors on day 2 by our forward update (making predictions):

$$\begin{aligned}\underline{u}_1(2) &= [0.2883, 0.0, 0.1268, 0.5849]\\ \underline{u}_2(2) &= [0.8135, 0.0, 0.1865, 0.0]\\ \underline{u}_3(2) &= [0.2883, 0.0, 0.1268, 0.5849].\end{aligned}$$

On the other hand, if we apply the naive updating rule defined in Remark 2, on day 2, we find

$$\begin{aligned}\underline{u}'_1(2) &= [0.4074, 0.0896, 0.0948, 0.4082]\\ \underline{u}'_2(2) &= [0.81, 0.0, 0.19, 0.0]\\ \underline{u}'_3(2) &= [0.4074, 0.0896, 0.0948, 0.4082].\end{aligned}$$

Recall that we use Assumption 1 in the proposed algorithm (Algorithm 4), and the Naive approach in Remark 2 does not have the backward step, so both approaches do not capture the correlations among nodes. By Monte Carlo simulations, the correlations among nodes are captured, and the nodes' probability vectors are approximated on day 2 as follows:

$$\begin{aligned}\underline{v}_1(2) &= [0.3235, 0.0976, 0.0196, 0.5593]\\ \underline{v}_2(2) &= [0.7244, 0, 0.2756, 0]\\ \underline{v}_3(2) &= [0.3158, 0.1072, 0.019, 0.558]\end{aligned}$$

which yields the following comparison for the incurred estimation errors:

$$\begin{aligned}0.4342 &= \sum_{i=1}^{3} ||\underline{u}_i(2) - \underline{v}_i(2)|| \\ &< \sum_{i=1}^{3} ||\underline{u}_i'(2) - \underline{v}_i(2)|| = 0.5018.\end{aligned}$$

The left hand side shows the estimation error under our proposed backward-forward update and the right hand side shows the estimation error under the naive approach.

I Delay of Testing Results

One can extend the framework to a more realistic case where testing results are not able to be obtained on the same day, but will be obtained after a delay a. In other words, if nodes are tested on day $t-a$, the test results are provided on day t. The extended framework is summarized as follows.

On day t, before getting the test results of day $t-a$, the algorithm knows the following information: (i) the network topology from day $t-a-1$ to day t, i.e., $\mathcal{G}(t-a-1), \cdots, \mathcal{G}(t)$ (it is affected by the past actions); (ii) the posterior probability of nodes on day $t-a-1$, $\left\{\underline{w}_i(t-a-1)\right\}_{i\in\mathcal{G}(t-a-1)}$; and (iii) and the prior probability vectors of nodes from day $t-a$ to day t, i.e., $\left\{\underline{u}_i(t-a)\right\}_{i\in\mathcal{G}(t-a)}, \cdots, \left\{\underline{u}_i(t)\right\}_{i\in\mathcal{G}(t)}$.

After getting the test results on day $t-a$, we can obtain the updated posterior probability vectors on day $t-a-1$, and the posterior probability vectors on day $t-a$, i.e., $\left\{\underline{e}_i(t-a-1)\right\}_{i\in\mathcal{G}(t-a-1)}$, and $\left\{\underline{w}_i(t-a)\right\}_{i\in\mathcal{G}(t-a)}$, by **Step 1** in Algorithm 4.

Based on $\{\underline{w}(t-a)\}_{i\in\mathcal{G}(t-a)}$, by **Step 2** in Algorithm 4, we update the prior probability from day $t-a+1$ to day t, and obtain the prior probability on day $t+1$, i.e., $\{\underline{u}_i(t+1)\}_{i\in\mathcal{G}(t+1)}$.

Repeating the process, we can compute the estimated probability vectors of nodes and apply the exploration and exploitation policies.

J Proof of Theorem 2

Step 1: Preliminaries.

We divide the distributions of initial infectious nodes into two complementary events:

$$\begin{aligned}\mathcal{I}_1 =& \{\text{No node is infectious}\} \\ \mathcal{I}_2 =& \mathcal{I}_1^c.\end{aligned}$$

Let N be sufficiently large,

$$\begin{aligned}&\Pr\{\mathcal{I}_1\} = (1-1/N)^N \approx 1/e \\ &\Pr\{\mathcal{I}_2\} \approx 1-1/e.\end{aligned}$$

In event $\mathcal{I}_1$, since there is no infection on the initial day, then no node is infectious in the future, i.e., the true probability of nodes $v_I^{(i)}(t) = 0$ for all $i \in \mathcal{V}(t)$ and $t \geq 1$.

Note that in Example 1, each node can be in one of two states, S and I. The transmission probability $\beta = 1$. So, on day t, the probability of node i being in state I includes the infection of node i on day $t-1$, and the infection from its neighbors. Then, based on (13), we have

$$\begin{aligned} u_I^{(i)}(t) &= w_I^{(i)}(t-1) + \{1 - w_I^{(i)}(t-1)\}\Big\{1 - \Big(1 - w_I^{(i-1)}(t-1)\Big)\Big(1 - w_I^{(i+1)}(t-1)\Big)\Big\} \\ &= 1 - \{1 - w_I^{(i-1)}(t-1)\}\{1 - w_I^{(i)}(t-1)\}\{1 - w_I^{(i+1)}(t-1)\}. \end{aligned} \tag{47}$$

For convention, we assume that nodes 0 and $N+1$ are two virtual nodes with no probability of infection, i.e., $u_I^{(0)}(t) = u_I^{(N+1)}(t) = 0$ for all t, and no tests are applied to these two nodes all the time.

Since $w_I^{(i)}(t+1), w_I^{(i-1)}(t+1), w_I^{(i+1)}(t+1) \in [0,1]$, then from (47),

$$u_I^{(i)}(t) \geq 1 - 1 \times (1 - w_I^{(i)}(t-1)) \times 1 = w_I^{(i)}(t-1). \tag{48}$$

Thus, by symmetry over $w_I^{(i-1)}(t-1)$, $w_I^{(i)}(t-1)$, and $w_I^{(i+1)}(t-1)$ we get the inequality

$$u_I^{(i)}(t) \geq \max\{w_I^{(i-1)}(t-1), w_I^{(i)}(t-1), w_I^{(i+1)}(t-1)\}. \tag{49}$$

Step 2: Consider the computation of $\{\underline{u}_i(t)\}_i$ based on (13) $\Big($equivalently (47)$\Big)$ under event $\mathcal{I}_1$.

Recall that $B(t) = 1$ for all t. On any day t, if node i_0 is tested, then the result is negative, and $w_I^{(i_0)}(t) = 0$, and

$$w_I^{(i)}(t) = u_I^{(i)}(t) \text{ for all } i \neq i_0. \tag{50}$$

In (49), at most one of $w_I^{(i-1)}(t-1)$, $w_I^{(i)}(t-1)$, and $w_I^{(i+1)}(t-1)$ is updated to 0. We first prove the following facts.

Fact 1. $u_I^{(i)}(t) \geq \frac{1}{N}$ for all t. On any day t, $w_I^{(i)}(t) \geq \frac{1}{N}$ with $i \neq i_0$, where i_0 is the index of node tested on day t.

Proof. We prove **Fact 1** by mathematical induction. On the initial day, by model assumption in Example 1, $u_I^{(i)}(0) = \frac{1}{N}$ for all i. Then, if node i_0 is tested, then as mentioned above, $w_I^{(i_0)}(0) = 0$, and by (50) $w_I^{(i)}(0) = u_I^{(i)}(0) = 1/N$ for all $i \neq i_0$.

Suppose **Fact 1** holds for all $\tau \leq t-1$. Now, we consider $\tau = t$. From (49), we have $u_I^{(i)}(t) \geq \max\{w_I^{(i-1)}(t-1), w_I^{(i)}(t-1), w_I^{(i+1)}(t-1)\} \geq 1/N$. Then, if node i_0 is tested, we have $w_I^{(i_0)}(t) = 0$, and then by (50), $w_I^{(i)}(t) = u_I^{(i)}(t) \geq 1/N$ for all $i \neq i_0$. □

Fact 2. If node i has not been tested up to day t, then $u_I^{(i)}(t)$ tends to 1 as $t \to \infty$.

Proof. Since node i is not tested from the initial day to day t, then

$$w_I^{(i)}(\tau) = u_I^{(i)}(\tau),\ \tau \leq t. \tag{51}$$

Note that at most one of its neighbors is tested on day t. By (47) and **Fact 1**,

$$u_I^{(i)}(t) \geq 1 - (1-1/N)(1 - w_I^{(i)}(t-1)) = 1 - (1-1/N)(1 - u_I^{(i)}(t-1)),,$$

which implies

$$(1-1/N)(1 - u_I^{(i)}(t-1)) \geq 1 - u_I^{(i)}(t),$$

which implies

$$1 - u_I^{(i)}(t) \leq (1-1/N)^t(1 - u_I^{(i)}(0)) = (1-1/N)^{t+1}.$$

Letting $t \to \infty$ completes the proof. □

Fact 3. If node i is not tested on day $t-1$, then

$$u_I^{(i)}(t) \geq w_I^{(i)}(t-1) + \frac{1}{N}(1 - w_I^{(i)}(t-1))w_I^{(i)}(t-1).$$

Proof. By **Fact 3**, if node i is not tested on day $t-1$, then $w_I^{(i)}(t-1) > 0$. From (47), by some algebra,

$$\begin{aligned} u_I^{(i)}(t) =& w_I^{(i)}(t-1) + (1 - w_I^{(i)}(t-1))(w_I^{(i-1)}(t-1) + w_I^{(i+1)}(t-1) - w_I^{(i-1)}(t-1)w_I^{(i+1)}(t-1)) \\ =& (1+\epsilon)w_I^{(i)}(t-1) \end{aligned}$$

where

$$\epsilon = \frac{1 - w_I^{(i)}(t-1)}{w_I^{(i)}(t-1)} \times (w_I^{(i-1)}(t-1) + w_I^{(i+1)}(t-1) - w_I^{(i-1)}(t-1)w_I^{(i+1)}(t-1)).$$

Note that at most one of the neighbors of node i is tested on day $t-1$, then

$$\begin{aligned} &\frac{1 - w_I^{(i)}(t-1)}{w_I^{(i)}(t-1)} \geq 1 - w_I^{(i)}(t-1) \\ &w_I^{(i-1)}(t-1) + w_I^{(i+1)}(t-1) - w_I^{(i-1)}(t-1)w_I^{(i+1)}(t-1) \geq \max\{w_I^{(i-1)}(t-1), w_I^{(i+1)}(t-1)\}. \end{aligned}$$

From **Fact 1**, $\max\{w_I^{(i-1)}(t-1), w_I^{(i+1)}(t-1)\} \geq 1/N$. Thus, $\epsilon \geq (1 - w_I^{(i)}(t-1)) \times 1/N$. Hence, $u_I^{(i)}(t) \geq w_I^{(i)}(t-1) + \frac{1}{N}(1 - w_I^{(i)}(t-1))w_I^{(i)}(t-1)$. □

Since we consider all possible sequential testing policies, then we divide all nodes into two sets

$$\begin{aligned} \mathcal{S}_1(t) =& \{\text{nodes that have not been tested up to day } t\} \\ \mathcal{S}_2(t) =& \mathcal{S}_1^c(t). \end{aligned}$$

In the following proof, let $t \to \infty$. By **Fact 2**, $u_I^{(i)}(t) \to 1$ if $i \in \mathcal{S}_1(t)$. Next, we focus on the set $\mathcal{S}_2(t)$. Denote the index of node which is tested on day $t-1$ as $i_0(t)$. By **Fact 1**, $w_I^{(i)}(t-1) \geq 1/N$ for all $i \neq i_0(t)$. Then, we define

$$\begin{aligned}\mathcal{S}_{21}(t) =& \{i | 1/N \leq w_I^{(i)}(t-1) < 1 - 1/N\} \\ \mathcal{S}_{22}(t) =& \{i | 1 - 1/N \leq w_I^{(i)}(t-1)\}.\end{aligned}$$

Thus, we have $\mathcal{S}_2(t) = \mathcal{S}_{21}(t) \cup \mathcal{S}_{22}(t) \cup \{i_0(t)\}$. Due to the equivalence of norms, without loss of generality, we consider L_1 norm in the rest of the proof.

(i) If $i \in \mathcal{S}_1(t)$, then $u_I^{(i)}(t) \to 1$. Thus $||\underline{u}_i(t) - \underline{v}_i(t)||_1 \to 2$.

(ii) If $i \in \mathcal{S}_{21}(t)$, then $||\underline{u}_i(t) - \underline{v}_i(t)||_1 \geq ||\underline{u}_i(t-1) - \underline{v}_i(t-1)||_1 + \frac{2(N-1)}{N^3}$. In fact, since $i \in \mathcal{S}_{21}(t)$, then $i \neq i_0(t)$, thus by (50) and **Fact 3**,

$$u_I^{(i)}(t) \geq u_I^{(i)}(t-1) + \frac{1}{N}(1 - w_I^{(i)}(t-1))w_I^{(i)}(t-1).$$

Note that N is sufficiently large, so $1/N < 1/2 < 1 - 1/N$. If $x \in [1/N, 1 - 1/N)$, then the fuction $f(x) = x(1-x)$ has the minimum value $\frac{N-1}{N^2}$ when $x = 1/N$. Thus,

$$u_I^{(i)}(t) \geq u_I^{(i)}(t-1) + \frac{N-1}{N^3}. \tag{52}$$

Recall that $v_I^{(i)}(t) = 0$ and $v_S^{(i)}(t) = 1$ for all t, and $u_I^{(i)}(t) + u_S^{(i)}(t) = 1$, then

$$||\underline{u}_i(t) - \underline{v}_i(t)||_1 = |u_I^{(i)}(t) - v_I^{(i)}(t)| + |u_S^{(i)}(t) - v_S^{(i)}(t)| = 2|u_I^{(i)}(t) - v_I^{(i)}(t)|. \tag{53}$$

From (52),

$$\begin{aligned}&||\underline{u}_i(t) - \underline{v}_i(t)||_1 = 2|u_I^{(i)}(t) - v_I^{(i)}(t)| \geq 2|u_I^{(i)}(t-1) + \frac{N-1}{N^3} - v_I^{(i)}(t-1)| \\ &\geq 2|u_I^{(i)}(t-1) - v_I^{(i)}(t-1)| + \frac{2(N-1)}{N^3} = ||\underline{u}_i(t-1) - \underline{v}_i(t-1)||_1 + \frac{2(N-1)}{N^3}.\end{aligned}$$

(iii) If $i \in \mathcal{S}_{22}(t)$, then node i is not tested on day t, thus from (49), $u_I^{(i)}(t) \geq w_I^{(i)}(t-1) = 1 - 1/N$. Thus, by (53), $||\underline{u}_i(t) - \underline{v}_i(t)||_1 \geq \frac{2(N-1)}{N}$.

Since we consider N sufficiently large, then we can prove the following lemma.

Lemma 4. $\lim_{t\to\infty} \mathcal{S}_{21}(t) = \varnothing$.

Proof. We first prove the following Claims.

Claim 1. If (i) $u_I^{(i-1)}(t) \geq 1 - 1/N$ and node $i-1$ is not tested on day t, or (ii) $u_I^{(i+1)}(t) \geq 1-1/N$ and node $i+1$ is not tested on day t, or (iii) $u_I^{(i-1)}(t) \geq 1-1/N$ and $u_I^{(i+1)}(t) \geq 1-1/N$, then $u_I^{(i)}(t+1) \geq 1 - 1/N$.

Proof. By (47) and (50), we can derive $u_I^{(i)}(t+1) \geq 1 - 1/N$ directly. □

Claim 2. No node can stay in $\mathcal{S}_{21}(t)$ for successive $\lceil N^3/(N-1)\rceil$ days.

Proof. if node i stays in $\mathcal{S}_{21}(t)$ for successive $\lceil N^3/(N-1)\rceil$ days, i.e., from day τ to day $\tau+\lceil N^3/(N-1)\rceil$, then by (52), $u_I^{(i)}(\tau+\lceil N^3/(N-1)\rceil)>1$, which contradicts with $u_I^{(i)}(t)\leq 1$ for all t. □

Now, we prove the lemma by contradiction. Based on **Claim 2**, assume there exists at least one j and an increasing sequence $\{t_i\}_{i=0}^{\infty}$ with $\lim_{n\to\infty} t_n=\infty$, such that $j\in\mathcal{S}_{21}(t_i)$ for all $\{t_i\}_{i=0}^{\infty}$.

For some i, node j is in $\mathcal{S}_{22}(t_i-1)$ on day t_i-1, and node j is in $\mathcal{S}_{21}(t_i)$ on day t_i. In other words, $u_I^{(j)}(t_i)<1-1/N\leq u_I^{(j)}(t_i-1)$. From (47) and **Calim 1**, $u_I^{(j)}(t_i)<1-1/N\leq u_I^{(j)}(t_i-1)$ holds only because node j is tested on day t_i-1, and all of its neighbors (i.e., nodes $j-1$, $j+1$) have $u_I^{(j-1)}(t_i-1)<1-1/N$ and $u_I^{(j+1)}(t_i-1)<1-1/N$. However, since $u_I^{(j)}(t_i-1)\geq 1-1/N$ and node j is tested on day t_i-1, then by **Claim 1**, $u_I^{(j-1)}(t_i)\geq 1-1/N$ and $u_I^{(j+1)}(t_i)\geq 1-1/N$. Subsequently, by **Claim 1**, we have $u_I^{(j)}(t_i+1)\geq 1-1/N$. Thus, on day t_i+1, at least one of its neighbors, say $j-1$, has $u_I^{(j-1)}(t_i+1)\geq 1-1/N$. By **Claim 1**, node j never fall into $\mathcal{S}_{21}(t)$ for $t\in\{t_{i+1},t_{i+2},\cdots\}$, which contradicts with the assumption. □

From Lemma 4, when $t\to\infty$, we have $|\mathcal{S}_1(t)|=\Theta(N)$ or $|\mathcal{S}_{22}(t)|=\Theta(N)$. Thus, $\sum_{i=1}^{N}||\underline{u}_i(t)-\underline{v}_i(t)||_1=\Theta(N)$.

Step 3: Consider the computation of $\{\underline{u}_i(t)\}_i$ based on Algorithm 4.

In this step, we consider a specific testing policy: We test node i on day k, where $k\equiv i-1(mod\ M)$ for all $1\leq i\leq M$.

In event $\mathcal{I}_2$, since the transmission probability $\beta=1$, then all nodes are infected at most N days because there is no recovery. Thus, no node with positive testing result is repeatedly tested. So in at most $2N$ days, all nodes are infectious, and the algorithm finds all infected nodes, so $\underline{u}_i(t)=\underline{v}_i(t)$, $t\geq 2N$.

In event $\mathcal{I}_1$, whenever a node is tested, it is negative. Node 1 is tested on day 0, the result is negative. On day 1, node 2 is tested and the result is negative. By backward updating, since $\beta=1$ and no recovery, then nodes 1&3 are inferred to be in state S on day 0. Since node 2 is in state S on day 1. Then, node 1 is inferred in state S on days 0 and 1.

Assume that nodes $1,2,\cdots,k-2$ are inferred to be in state S by day $k-1$. Now, we day k, where $k\leq N$. On day k, node $k-1$ is tested negative, hence by backward updating, nodes $k-2$ and k are inferred to be in state S on day $k-1$. By the testing result of node $k-1$ on day k, nodes $1,2,\cdots,k-1$ are inferred in state S by day k. By induction, after N days, it clears every node, so $\underline{u}_i(t)=\underline{v}_i(t)$, $t\geq N$.

From **Steps 1∼3**, we complete the proof.

K α-linking Backward Updating

K.1 Complexity Reduction

Let $\{x_j\}_{j\in\mathcal{O}(t)}$ be a realization of $\{\theta_j(t)\}_{j\in\mathcal{O}(t)}$ and $\{y_l\}_{l\in\Theta_i(t)}$ be a realization of $\{\zeta_l(t-1)\}_{l\in\Theta_i(t)}$. Let node i have state x in day $t-1$. Consider one node $k\in\partial_j^+(t-1)\backslash\{i\}$ and the probability

$$\Pr\Big(x_j|\{y_l\}_{l\in\partial_j^+(t-1)\backslash\{i\}},x\Big), j\in\Psi_i(t).$$

Since node k is not infectious if $y_k=L$, $y_k=R$ or $y_k=S$, then the probability above remians the same no matter whether $y_k=L$, $y_k=R$ or $y_k=S$.

Thus, we introduce a new state, denoted by E, to be a replacement of $\{L,R,S\}$, and

$$\Pr\Big(y_k=E\Big)=\sum_{x\in\{L,R,S\}}\Pr\Big(y_k=x\Big).$$

Next, denote $\mathcal{X}'=\{I,E\}$. Equation (17) can be re-written as follows:

$$\begin{aligned}&\Pr\Big(\{Y_j(t)\}_{j\in\Psi_i(t)}|\zeta_i(t-1)=x\Big)\\&=\sum_{\{x_j\}_{j\in\Psi_i(t)}}\prod_{j\in\Psi_i(t)}\Pr\Big(Y_j(t)|\theta_j(t)\Big)\\&\times\sum_{\{y_l\}_{l\in\Theta_i(t)}}\prod_{j\in\Psi_i(t)}\mathtt{P}_j\Big(x_j|\{y_l\}_{l\in\partial_j^+(t-1)\backslash\{i\}},x\Big)\\&\times\prod_{z_l\in\mathcal{X}',l\in\Theta_i(t)}\Pr\Big(\zeta_l(t-1)=z_l\Big),\end{aligned}\tag{54}$$

with reduces the computation complexity. Subsequently, $\underline{e}_i(t-1)$ in (16) can be calculated by (54) directly.

K.2 α-linking Backward Updating

In the backward step, the computation complexity is large even in (54). To further reduce the complexity in (54), one way is to update the posterior probability $\underline{e}_i(t)$ in a sparser network. Now, we define *α-linking Backward Updating* as follows:

(i) We generate a subgraph $\mathcal{G}_\alpha(t)$ based on the pre-determined graph $\mathcal{G}(t)$: Suppose that each edge (in $\mathcal{G}(t)$) exists with probability α, $0\le\alpha\le1$. If $\alpha=1$, then $\mathcal{G}_\alpha(t)=\mathcal{G}(t)$; if $\alpha=0$, then $\mathcal{G}_\alpha(t)$ is a graph with no edges.

(ii) Backward updating in $\mathcal{G}_\alpha(t)$: Similar with $\partial_i(t)$, $\Psi_i(t)$, $\Phi_i(t)$ and $\Theta_i(t)$, we define $\partial_{i,\alpha}(t)$, $\Psi_{i,\alpha}(t)$, $\Phi_{i,\alpha}(t)$ and $\Theta_{i,\alpha}(t)$ on graph $\mathcal{G}_\alpha(t)$, respectively. Subsequently, replace $\partial_i(t)$, Ψ_i, $\Phi_i(t)$ and $\Theta_i(t)$ by $\partial_{i,\alpha}(t)$, $\Psi_{i,\alpha}(t)$, $\Phi_{i,\alpha}(t)$ and $\Theta_{i,\alpha}(t)$ in (54), respectively.

L Proof of Theorem 3

Step 1. Preliminaries.

In Example 2, $\beta=1$, $\lambda=0$, and $\gamma=0$, there is no recovery and we assume no latent state. Based on (9), the expression of rewards $\hat{r}_i(t)$ for every node is given as follows. If node i has two neighbors (without quarantine)

$$\hat{r}_i(t)=u_S^{(i-1)}(t)(1-u_I^{(i-2)}(t))u_I^{(i)}(t)+u_S^{(i+1)}(t)(1-u_I^{(i+2)}(t))u_I^{(i)}(t).\tag{55}$$

If node i only has one neighbor, then

$$\hat{r}_i(t) = u_S^{(i+d)}(t)\Big(1 - u_I^{(i+2d)}(t)\Big)u_I^{(i)}(t),\ d \in \{-1, 1\}. \tag{56}$$

For simplicity, we introduce artificial nodes $-1, 0, N+1, N+2$ with $u_I^{(-1)}(t) = u_I^{(0)}(t) = u_I^{(N+1)}(t) = u_I^{(N+2)}(t) = 0$ for all t, and these 4 nodes are never tested.

Step 2. The RbEx policy.

Under the RbEx policy, the algorithm always tests the nodes with maximum rewards. Let an infectious node be found, for the first time, on day aN, where a is a positive real number. Note that until the first infected node is found, in any application of the RbEx policy, $u_I^{(i)}(t)$ is the same for any given i, and hence $\hat{r}_i(t)$ is also the same. So, a is the same for any application of the RbEx policy. Recall that in Example 2, nodes that are tested positive will be isolated. The cumulative infections is at least $\min\{aN, N\}$ in the end.

Step 3. Consider the exploration process of the specific exploration policy.

Recall that from **Step 2**, an infectious node is found, for the first time, by the RbEx policy with budget 10 tests on day aN. Under the specific defined exploration policy, we can choose a specific b' with $b' < a$, such that no infectious node is tested by the RbEx policy with budget 9 tests before and including day $t = b'N$.

We know that on day τ, nodes $1, 2, \cdots, \tau$ are infectious since $\beta = 1$. Note that one test is applied to exploration (randomly choice) on every day, so with probability

$$\prod_{\tau'=1}^{\tau}(1 - \frac{\tau'}{N}), \tag{57}$$

no infectious node is explored from the initial day to day τ. Then, with probability

$$\prod_{\tau'=1}^{\tau-1}(1 - \frac{\tau'}{N}) \cdot \frac{\tau}{N},$$

one infectious node is detected on day τ. Thus, with probability

$$\sum_{\tau=1}^{t}\prod_{\tau'=1}^{\tau-1}(1 - \frac{\tau'}{N}) \cdot \frac{\tau - 1}{N}, \tag{58}$$

one infectious node is tested by exploration process on day τ ($\tau \leq t$), and this node is not the new infectious one on day τ, i.e., has index τ. The probability defined in (58) increases with t when N is fixed, and it can be close to 1 when t close to N. Therefore, We can choose proper parameters b' and N such that the probability defined in (58) is larger than or equal to p_0. In particular, if N is large, we can choose a relatively small b'. In Theorem 3, we set $p_0 \geq 99/100$.

Let the infectious node detected (for the first time) by the exploration process have index j on day t', where $t' \leq t$. As discussed above, node j is not the new infectious node on day t', so we have $j < t'$. In other words, node $j + 1$ must be infecitous on day t' with a positive

test result, i.e., $Y_j(t') = 1$. By Step 1 in Algorithm 4, the updated posterior probability of node j

$$e_I^{(j)}(t'-1) = 1, \quad e_S^{(j)}(t'-1) = 0. \tag{59}$$

Again, by Step 1 in Algorithm 4,

$$w_I^{(j-1)}(t') = w_I^{(j+1)}(t') = 1. \tag{60}$$

Then, by Step 2 in Algorithm 4,

$$u_I^{(j-2)}(t'+1) = u_I^{(j-1)}(t'+1) = u_I^{(j)}(t'+1) = u_I^{(j+1)}(t'+1) = u_I^{(j+2)}(t'+1) = 1. \tag{61}$$

Since j is detected and isolated on day t', then,

$$\hat{r}_j(t'+1) = 0. \tag{62}$$

By (56) and (61),

$$\begin{aligned} \hat{r}_{j-1}(t'+1) &= u_S^{(j-2)}(t'+1)\Big(1 - u_I^{(j-3)}(t'+1)\Big) = 0 \\ \hat{r}_{j+1}(t'+1) &= u_S^{(j+2)}(t'+1)\Big(1 - u_I^{(j+3)}(t'+1)\Big) = 0. \end{aligned} \tag{63}$$

By (55) and (61),

$$\begin{aligned} \hat{r}_{j-2}(t'+1) &= u_S^{(j-3)}(t'+1)\Big(1 - u_I^{(j-4)}(t'+1)\Big) \\ \hat{r}_{j+2}(t'+1) &= u_S^{(j+3)}(t'+1)\Big(1 - u_I^{(j+4)}(t'+1)\Big). \end{aligned} \tag{64}$$

Step 4.The exploitation process of the specific exploration policy.

We first study an extreme case where no tests are applied. In this case, denote the prior probability of node i on day τ as $U_I^{(i)}(\tau)$, which can be calculated by the following recursion:

$$U_I^{(i)}(\tau+1) = U_I^{(i)}(\tau) + U_S^{(i)}(\tau)\Big(1 - (1 - U_I^{(i-1)}(\tau))(1 - U_I^{(i+1)}(\tau))\Big). \tag{65}$$

Based on (65), recall that $U_I^{(i)}(0) = 0$ if $i \leq \frac{9N}{10}$, and $U_I^{(i)}(0) = \frac{10\epsilon}{N}$ if $\frac{9N}{10} < i \leq N$, then $U_I^{(i)}(\tau)$ increases over τ and is a function of ϵ. Then, given b', N and $t = b'N$, we can choose a small enough ϵ, denoted by $\epsilon(b', N)$, such that $U_I^{(i)}(2t) < \frac{1}{2}$ for all i. Since $U_I^{(i)}(\tau)$ increases over τ, then $U_I^{(i)}(\tau) < \frac{1}{2}$, $\tau \leq 2t$.

Now, we introduce the exploitation process. Let $t = b'N < \min\{\frac{9}{40}, a\}N$. There are at most $2t$ infectious nodes on day $2t$, i.e., nodes $1, 2, \cdots, 2t$. Since $t < \min\{\frac{9}{40}, a\}N$, then nodes with index from $9N/10 - 2t$ to N are in state S, which implies nodes with index from $9N/10 - 2t$ to N can never be tested positive before day $2t$. Thus, on any day $\tau \leq 2t$, for $9N/10 - 2t \leq i \leq N$, if node i is tested, and the testing result is negative. Recall that $U_I^{(i)}(\tau)$ in (65) is calculated without any negative testing results. Hence, $u_I^{(i)}(\tau) \leq U_I^{(i)}(\tau)$.

Furthermore, with the condition $t = b'N < \min\{\frac{9}{40}, a\}N$, we can find a small enough $\epsilon(b', N)$, such that under the specific exploration policy,

$$u_I^{(i)}(\tau) < \frac{1}{2}, \quad \tau \leq 2t,\ 9N/10 - 2t \leq i \leq N. \tag{66}$$

In the rest, we divide the nodes in to 3 sets: $\mathcal{Q}_1 = \{i | i \leq 2t\}$, $\mathcal{Q}_2 = \{i | 2t < i < 9N/10 - 2t\}$, and $\mathcal{Q}_3 = \{i | 9N/10 - 2t \leq i \leq N\}$.

Fact 1. For $i \in \mathcal{Q}_1$ and $\tau \leq 2t$, $u_I^{(i)}(\tau) = 1$ or $u_I^{(i)}(\tau) = 0$.

Proof. If no test is applied to $\mathcal{Q}_1$, then $u_I^{(i)}(\tau) = 0$ for all $i \in \mathcal{Q}_1$.

On some day $\tau \leq 2t$, if one node with index $j \in \mathcal{Q}_1$ is tested positive on day $\tau - 1$, then by (61), $u_I^{(j-2)}(\tau) = u_I^{(j-1)}(\tau) = u_I^{(j)}(\tau) = u_I^{(j+1)}(\tau) = u_I^{(j+2)}(\tau) = 1$. In other words, if node j is tested positive on day $\tau - 1$, then node j, its neighbors and neighbors of neighbors have probability of infection equal to 1 on day τ.

On some day τ, if node j is not tested positive on day $\tau - 1$, and neither of its neighbors and neighbors of neighbors are is not tested positive, then $u_I^{(j)}(\tau) = 1$ only when $u_I^{(j)}(\tau - 1) = 1$, or $u_I^{(j-1)}(\tau - 1) = 1$ or $u_I^{(j+1)}(\tau - 1) = 1$ since $\beta = 1$. Otherwise $u_I^{(j)}(\tau) = 0$. □

Fact 2. For $i \in \mathcal{Q}_1$ and $\tau \leq 2t$, $\hat{r}_i(\tau) = 1$ or $\hat{r}_i(\tau) = 0$.

Proof. If $u_I^{(i)}(\tau) = 0$, then $\hat{r}_i(\tau) = 0$ by (55) and (56).

Now, we consider $u_I^{(i)}(\tau) = 1$ in the following cases: (i) If both neighbors of node i are isolated, then $\hat{r}_i(\tau) = 0$. (ii) If one of neighbors of node i (for example, node $i - 1$) is isolated, then by (56), $\hat{r}_i(\tau) = 0$ when $u_I^{(i+1)}(\tau) = 1$, and $\hat{r}_i(\tau) = 1$ when $u_I^{(i+1)}(\tau) = 0$. (iii) If both neighbors are not isolated, then $u_I^{(i-1)}(\tau - 1) = 1$ or $u_I^{(i+1)}(\tau - 1) = 1$, otherwise, $u_I^{(i)}(\tau) = 0$. Since there is no recovery, then $u_I^{(i-1)}(\tau) = 1$ or $u_I^{(i+1)}(\tau) = 1$. By **Fact 1**, $u_I^{(j)}(\tau) = 1$ or $u_I^{(j)}(\tau) = 0$ when $j \in \mathcal{Q}_1$. If $u_I^{(i+1)}(\tau) = u_I^{(i-1)}(\tau) = 1$, then $\hat{r}_i(\tau) = 0$. If $u_I^{(i+1)}(\tau) = 1$, then by (55), $\hat{r}_i(\tau) = 0$ when $u_I^{(i-2)}(\tau) = 1$, $\hat{r}_i(\tau) = 1$ when $u_I^{(i-2)}(\tau) = 0$. If $u_I^{(i-1)}(\tau) = 1$, then by (55), $\hat{r}_i(\tau) = 0$ when $u_I^{(i+2)}(\tau) = 1$, $\hat{r}_i(\tau) = 1$ when $u_I^{(i+2)}(\tau) = 0$. □

From (66), for all $\tau \leq 2t$ and $i \in \mathcal{Q}_3$, we have

$$\hat{r}_i(\tau) \leq 2u_I^{(i)}(\tau) < 1. \tag{67}$$

Note that only nodes in $\mathcal{Q}_1$ and $\mathcal{Q}_3$ may have positive probability of infection. For all $\tau \leq 2t$ and $i \in \mathcal{Q}_2$, since $u_I^{(i)}(\tau) = 0$, then $\hat{r}_i(\tau) = 0$. Therefore, a node with reward equal to 1 has the largest reward.

Recall that on day t', node j is tested positive. From (64), nodes $j - 2$ and $j + 2$ have largest rewards ($= 1$) on day $t' + 1$, which are exploited on day $t' + 1$, and all other nodes in $\mathcal{Q}_1$ have rewards 0. This is because t' is the first day when a positive node is found. Since node j is tested positive and isolated on day t', then all infectious nodes with index less than j can no longer infect other nodes in the line network. Now, we consider the nodes with index larger than j. Recall that $j < t'$, so node $j + 1$ must be infectious on day t', and node $j + 2$ must

be infectious on day $t'+1$ since $\beta = 1$. Thus, node $j+2$ is tested positive and is isolated. Since the network is a line, both nodes $j+1$ and $j+2$ can no longer infect other nodes once node $j+2$ is isolated. Note that nodes in $\mathcal{Q}_3$ have positive rewards. When N is sufficiently large, in the rest of the exploitation process, nodes in $\mathcal{Q}_3$ are tested. Recall that we have one test for exploration, and we can isolate at least 2 infectious nodes with index larger than j.

Repeat the process, we exploit nodes $j+4, j+6, \cdots$ on day $t'+2, t'+3, \cdots$, respectively. Consider the direction from node 1 to node N. On every day, there is at most one new infectious node, but at least two infectious nodes can be isolated. On some day, denote as day $t'+x$, the exploitation process can progress beyond the infections (exceeding by one node) for the first time. In other words, node $j+2x$ is tested negative on day $t'+x$. By Step 2 in Algorithm 4, $e_I^{(j+2x)}(t'+x-1) = 0$. However, since $w_I^{(j+2x-1)}(t'+x) = 1$ becuase node $j+2x-2$ is tested positive on day $t'+x-1$. By Step 1 in Algorithm 4, $u_I^{(j+2x)}(t'+x+1) = 1$, hence by (64), $\hat{r}_{j+2x}(t'+x+1) = 1$, which implies node $j+2x$ has the largest reward and is exploited on day $t'+x+1$, and it will be tested positive. On day $t'+x+1$, all infectious nodes are isolated.

Finally, we can calculate the total number of infections to be

$$j + 2(t' - j) = 2t' - j \leq 2t' \leq 2t = 2b'N.$$

Let $b = 2b'$. This is an improvement by a factor of at least $\frac{a}{b}$ in comparison to the RbEx strategy, where $\frac{a}{b}$ can be as large as desired by increasing the value of N or decreasing p_0.

M Further Results

M.1 The impact of γ_c and L_p individually

In this subsection, we investigate the role of γ_c and L_p individually. We consider different WS networks with degrees $d = 4, 6$, and then adjust the rewiring probability δ, such that one of (γ_c, L_p) is almost constant, and the other is varying. We can see that the trend is similar to what we observed by varying δ in Table 2.

WS, (d, δ)	γ_c	L_p	`Ratio`
$(6, 0.05)$	**0.504**	4.952	.0003
$(4, 0)$	**0.500**	62.876	0.191
$(6, 0.1)$	**0.456**	5.718	-0.027
$(4, 0.03)$	**0.456**	10.810	0.097

Table 5: Clustering coefficients of WS networks

M.2 Simulation Results in SBM and V-SBM

In this section, we investigate SBMs and its variants. The RbEx policy always outperforms the RbEr policy in SBM. However, the RbEx policy outperforms the RbEr policy when ℓ is small, which the RbEr policy is better when ℓ is large.

WS, (d, δ)	γ_c	L_p	Ratio
$(6, .001)$	0.599	**21.188**	0.209
$(4, .0075)$	0.489	**21.264**	0.182
$(6, .005)$	0.592	**14.310**	0.211
$(4, .015)$	0.473	**14.253**	0.174
$(6, .009)$	0.585	**12.081**	0.137
$(4, .0225)$	0.467	**12.171**	0.125

Table 6: Clustering coefficients of WS networks

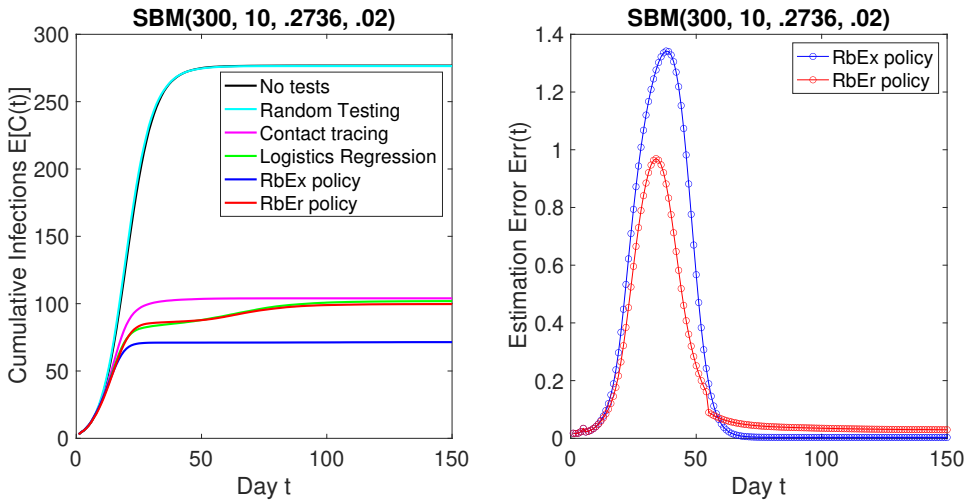

Figure 9: Performances and estimation errors of different policies in SBM(10, $p_{1,s}$, $p_{2,s}$, 5, .04).

Stochastic Block Model The SBM is a generative model for random graphs. The graph is divided into several communities, and subsets of nodes are characterized by being connected with one another with particular edge densities.[6] The intra-connection probability is p_1, and inter-connection probability is p_2. We denote the SBM as SBM(N, M, p_1, p_2). We consider the transmission rate to be $\beta = 0.04$ and the initial seed to be $n_0 = 3$.

Note that the (expected) number of edges, denoted by $|\mathcal{E}|$, is

$$|\mathcal{E}| = \frac{p_1}{2} N(\frac{N}{M} - 1) + \frac{p_2}{2} \frac{N^2}{M}(M - 1). \tag{68}$$

We choose the pair (p_1, p_2) under a fixed $|\mathcal{E}|$ in (68). Here, we choose a fixed expected number of edges, i.e., $|\mathcal{E}|$ to guarantee that the transmission of the disease would not be affected by edges.

Figure 9 shows the performance of different policies in a SBM network with $N = 300$, $M = 10$, and $p_1 = .274$, and $p_2 = .2$. Let $\ell = 5$. We show that the RbEx policy outperforms the RbEr policy. In addition, we see on the right figure that the RbEx policy provides a (slightly) more accurate estimation for $\{\underline{v}_i(t)\}_i$.

[6]Here, we assume that M is an exact divisor of N.

We next vary p_1 to investigate the role of path length L_p and clustering coefficient γ_c (see Table 7). First we can see the RbEx policy outperforms the PbEr policy in SBM. This is also confirmed by the better estimation error that exploitation provides compared to exploration, as shown in Table 7. From Table 7 and Table 8 (in Table 8, we fix $(p_1, p_2) = (.274, .02)$), as L_p, γ_c, or ℓ decreases, `Ratio` decreases as well, which implies the exploitation provides more benefits as L_p, γ_c, or ℓ decreases.

SBM, (p_1, p_2)	γ_c	L_p	`Ratio`	$\Delta_{\texttt{Err}}$
$(0.274, 0.02)$	0.111	2.573	-0.092	-0.026
$(0.214, 0.026)$	0.075	2.518	-0.103	-0.023
$(0.159, 0.032)$	0.056	2.492	-0.113	-0.026
$(0.102, 0.039)$	0.048	2.480	-0.118	-0.023
$(0.045, 0.045)$	0.043	2.455	-0.124	-0.027

Table 7: Clustering coefficients and path lengths of SBM networks

V-SBM Denote a variant of SBM as V-SBM. Different from SBM, we only allow nodes in cluster i to connect to nodes in successive clusters (the neighbor clusters). Denote the V-SBM as V-SBM(N, M, p_1, p_2). Let $\beta = 0.04$ and $n_0 = 3$. For this model, the expected number of edges, denoted by $|\mathcal{E}|$, is

$$|\mathcal{E}| = \frac{p_1}{2} N(\frac{N}{M} - 1) + p_2 \frac{N^2}{M}. \tag{69}$$

We choose the pair (p_1, p_2) under a fixed $|\mathcal{E}|$ in (69).

Figure 10 shows the performance of different policies in a V-SBM network with $N = 300$, $M = 10$, and $p_1 = .418$, and $p_2 = .2$. Let $\ell = 5$. We show that the RbEx policy outperforms the RbEr policy, and the RbEx policy provides a more accurate estimation for $\{\underline{v}_i(t)\}_i$.

Varying p_1, we then investigate the role of path length L_p and clustering coefficient γ_c in Table 9. The trends are similar with Table 7.

SBM, ℓ	5	7	9	11	13
`Ratio`	-0.092	-0.079	-0.042	-0.035	-0.025
$\Delta_{\texttt{Err}}$	-0.026	-0.015	-0.010	-0.009	-0.009

Table 8: Role of the unregulated delay ℓ

The impact of ℓ is shown in Table 10. Fix $(p_1, p_2) = (.418, .02)$. As ℓ increases, so does `Ratio`, implying that exploration becomes more essential. In addition, the exploitation policies perform better when ℓ is small, and the exploration policies perform better when ℓ is relatively large. One interesting observation is that exploration policies perform better when ℓ is large, but the corresponding estimation error is larger than that in exploitation policies. In this specific network topology, it appears that smaller estimation error does not always correspond to better cumulative infections. The exploration policies is sensitive to ℓ in this topology, i.e., we can achieve smaller cumulative infections under exploration policies even if the estimation error is larger.

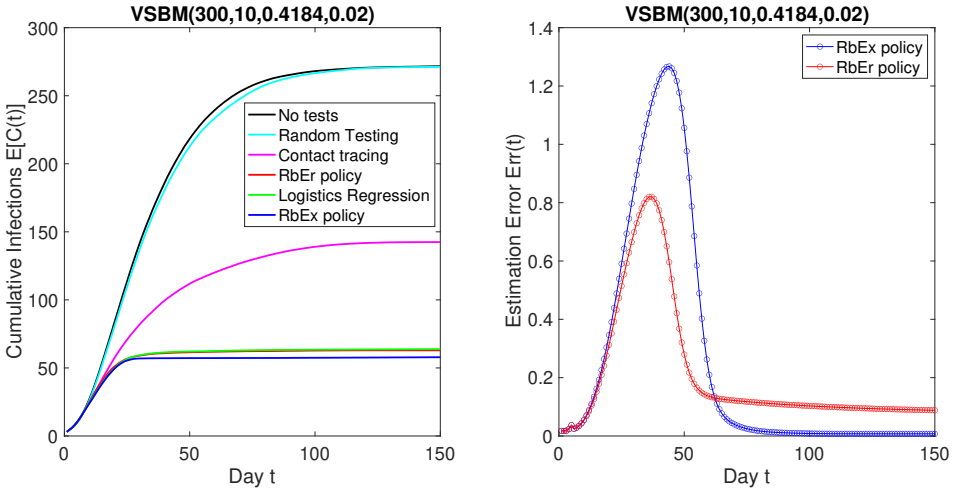

Figure 10: Performances and estimation errors of different policies in VSBM(10, $p_{1,v}$, $p_{2,v}$, 5, .04).

V-SBM, (p_1, p_2)	γ_c	L_p	`Ratio`	$\Delta_{\texttt{Err}}$
$(0.418, 0.020)$	0.3557	4.4264	-0.022	-0.081
$(0.351, 0.052)$	0.2365	3.6584	-0.091	-0.045
$(0.284, 0.085)$	0.1769	3.307	-0.104	-0.055
$(0.217, 0.085)$	0.1385	3.1562	-0.112	-0.041
$(0.150, 0.0150)$	0.1170	3.0563	-0.123	-0.042

Table 9: Clustering coefficients and path lengths of RSBM networks

SBM, ℓ	5	7	9	11	13
`Ratio`	-0.022	-0.016	-0.007	0.011	0.019
$\Delta_{\texttt{Err}}$	-0.081	-0.066	-0.046	-0.033	-0.025

Table 10: Role of the unregulated delay ℓ