Reconstructing Network Outbreaks under Group Surveillance

Ritwick Mishra
Department of Computer Science & Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA
mbc7bu@virginia.edu

Abhijin Adiga
Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA
abhijin@virginia.edu

Anil Vullikanti
Department of Computer Science & Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA
vsakumar@virginia.edu

ABSTRACT

A key public health problem during an outbreak is to reconstruct the disease cascade from a partial set of confirmed infections. This has been studied extensively under the Maximum Likelihood Estimation (MLE) formulation, which reduces the problem to finding some type of Steiner subgraph on a network. Group surveillance like wastewater or aerosol monitoring is a form of mass/pooled testing where samples from multiple individuals are pooled together and tested once for all. While a single negative test clears multiple individuals, a positive test does not reveal the infected individuals in the test pool. We introduce the PoolCascadeMLE problem in the setting of a network propagation process, where the goal is to find a MLE cascade subgraph which is consistent with the pooled test outcomes. Previous work on reconstruction assumes that the test results are of individuals, i.e., pools of size one, and requires a consistent cascade to connect the positive testing nodes. In PoolCascadeMLE, a consistent cascade must choose at least one node in each positive pool, adding another combinatorial layer. We show that, under the Independent Cascade (IC) model, PoolCascadeMLE is NP-hard, and present an approximation algorithm based on a reduction to the Group Steiner Tree problem. We also consider a one-hop version of this problem, in which the disease can spread for one time step after being seeded. We show that even this restricted version is NP-hard, and develop a method using linear programming relaxation and rounding. We evaluate the performance of our methods on real and synthetic contact networks, in terms of missing infection recovery and prevalence estimation. We find that our approach outperforms meaningful baselines which correspond to pools of size one and use state-of-the-art methods.

KEYWORDS

Cascade Reconstruction; Combinatorial Optimization; Agent-Based Simulation; Social Network Analysis

ACM Reference Format:
Ritwick Mishra, Abhijin Adiga, and Anil Vullikanti. 2026. Reconstructing Network Outbreaks under Group Surveillance. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/VRPM5709

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/VRPM5709

1 INTRODUCTION

Inferring the characteristics of an outbreak from partial observations is an important problem in many domains such as computational epidemiology, biological invasions, and social contagions [18, 23, 26, 30]. This is a very challenging problem due to multiple reasons. First, the number of tests that is done is typically much smaller than the population. Second, disease transmission is very complex, and there might be many possible outbreak scenarios consistent with observed test results. There has been a lot of work on formalizing how to reconstruct an outbreak from available information. A common and intuitive approach has been to find a maximum likelihood estimation (MLE) solution. However, solving for the MLE is quite challenging, and several works have approximated it with a minimum cost Steiner tree, which is well understood [18, 27, 29, 34]. While this is reasonable in many settings, in general, the MLE solution can be quite different from a Steiner tree, and Mishra et al. [23] show that the MLE solution can, instead, be found directly.

Group surveillance is increasingly emerging as a practical way of monitoring diseases in populations. A classical idea in this regard is pool testing [13, 24]. where the objective is to test a set of samples (or "pool") simultaneously. If the pool test result is negative (and the test is assumed to be perfect), all the individuals in the pool can be cleared of the infection. On the other hand, if a pool test result is positive, one can only infer that at least one of the individuals in the pool was positive. The main motivation is that the availability of tests is often quite limited, especially in the early stages of a pandemic, as in the case of COVID-19. In recent times, various new environmental surveillance methods like wastewater [3, 14, 19, 20, 33] and bioaerosol [6, 32] monitoring have emerged as important group surveillance methods.

In this work, we consider the problem of reconstructing a cascade in a group surveillance setting. Despite a lot of interest in group surveillance, this problem hasn't been considered. Prior work on pool testing for infectious diseases [13, 24] has focused on the problem of allocating tests to maximize the number of entities who get cleared (by negative tests). Wastewater monitoring studies have addressed problems such as determination of hospitalization rates, forecasting, risk assessment, and optimal placement of sensors [7, 32]. Air sampling studies consider the problem of timely detection of bioaerosols in hostpital settings and livestock operations [4, 6]. Note that the prior work on cascade reconstruction can be viewed as a special case of this problem with pool size 1.

Our contributions are as follows:

1. We introduce the PoolCascadeMLE problem under the Independent Cascade (IC) model of disease transmission on a network as

constructing an MLE solution from given group test results (Section 3). We show that PoolCascadeMLE is NP-hard to approximate within a factor of $O(\log^{2-\epsilon} k)$, where k denotes the number of node groups in the problem instance. In contrast, when the group size is 1, an $O(\log k)$ approximation is possible [23]. Here, we present a $O(k^{\epsilon})$-approximation algorithm called ApproxCascade. We also consider the more general setting where tests are unreliable, and show that a similar approximation is possible.

2. We consider a special case of the MLE estimation problem, OneHopCascadeMLE, which corresponds to for one step of disease spread. This is specially motivated by group surveillance in wastewater and animal farms. We show that this problem is also NP-hard, and give an $O(\log k)$ approximation algorithm.

3. We evaluate the performance of ApproxCascade on two tasks: (a) recovering the infected nodes, (b) estimating prevalence, i.e., outbreak size. We use synthetic data generated on multiple synthetic and real contact networks, including a contact network built from the Electronic Health Record (EHR) for the UVA Hospital ICU and a large realistic contact network representing the population of a small city. We find ApproxCascade compares favorably to baselines which reduce pool size to 1 and use the state-of-the-art method in [23]. We also evaluate RoundCascade against a random baseline and find that it too outperforms in missing infection recovery.

4 . We examine the conditions in which the PoolCascadeMLE solution does not recover the ground truth to show the limitations of the MLE approach. We also show that even a little bit of noise in the test results can change the MLE solution of NoisyPoolCascadeMLE quite significantly, compared to that of PoolCascadeMLE.

Due to space constraints, some proofs and experimental details are omitted; they are available in the full version on arXiv [22].

2 RELATED WORK

In the pool testing literature, the general goal is to identify the subset of infections in a relatively very large population by simultaneously testing multiple individuals. The idea was first proposed during World War II in the context of detecting syphilis infected population [11]. Subsequently, it has been applied in various domains such as industrial testing, experiment design, coding theory [12]. Multiple variants of the problem exist accounting for aspects such as adaptive or non-adaptive setting, noiseless or noisy tests, and exact or partial recovery [2]. There has been renewed interest in this topic in the context of COVID-19 [13, 24, 31]. In recent years, there has been work that accounts for heterogeneous interactions of individuals in the population in the design of group testing algorithms. References [5, 28] assume the knowledge of the network of interactions for better pool design. To the best of our knowledge, existing works focus on designing the pools with the objective of minimizing the number of tests to be performed in both non-adaptive and adaptive settings.

Our work is motivated by the emergence of various environmental surveillance methods like wastewater and bioaerosol monitoring in the context of infectious diseases in humans and animals. The objective is to estimate important characteristics of the disease from observations for downstream tasks of risk assessment, situational awareness, and forecasting [9]. In this setting, pools are already provided along with test results (see for example the national-scale monitoring of H5N1 [15, 21]).

In the setting of network propagation processes, reconstructing a cascade given partial information about infected nodes (pool size 1) is well-studied. Rozenshtein et al. [27] use a directed Steiner tree approach in a temporal network setting, where a subset of infected nodes along with the time of infection is provided. Jang et al. [18] account for node attributes to detect asymptotic cases. Mishra et al. [23] formulate an MLE problem to reconstruct cascade given a subset of observed infected nodes and diffusion model. Unlike previous works, they consider true MLE cost that includes failed infection attempts, and use a node-weighted Steiner tree approach for cascade reconstruction. Our work uses this MLE formulation, but generalizes this work to the pool-testing scenario. Qiu et al. [25] consider the problem of reconstructing diffusion history from a single snapshot of the cascade, i.e., the knowledge of all infected nodes. Their barycenter formulation does not assume the knowledge of the diffusion parameters.

3 PRELIMINARIES

Disease model. We consider the Independent Cascade (IC) model of disease spread which is the simplest form of the discrete-time SIR model, on an undirected network $G = (V, E)$. Each node is in one of the following states: susceptible (S), infectious (I) or removed (R). Let $V_0 \subset V$ denote the initial set of nodes in state I that seed the disease at time $t = 0$. At any time t, each node in state I infects its susceptible neighbor with probability according to the weight on the respective edge. Then, v transitions to state R at time t , thus getting only one chance to infect. We consider two variations of this IC process:

(a) `single-seed`: The disease begins from a single known seed node, i.e. $V_0 = \{s\}$, and can spread over multiple time steps, until there are no more new infections.

(b) `one-hop`: The disease spread is limited to one time step after seeding. Here V_0 is not known.

One realization of this random process is called a *cascade*.

Pool testing. A pool test corresponds to a subset $g_i \subset V$. This indicates that the subset g_i are tested simultaneously; the test result is positive if any node in g_i is infected. A set $\Gamma = \{g_1, g_2, \ldots, g_k\}$, with each $g_i \subset V$ denotes a group level surveillance strategy. We use Γ_1 to denote the set of pools which test positive (i.e., at least one node amont them is infected); $\Gamma_0 = \Gamma - \Gamma_1$ is the set of pools which test negative.

Criteria for a consistent cascade. Given a set of observations (Γ_0, Γ_1), we require for a cascade A to be consistent that it meet both of these criteria: (a) that it does not include any node from Γ_0, i.e, $\bigcup_{g \in \Gamma_0} g \cap V(A) = \emptyset$, and (b) that it includes at least one node from each pool in Γ_1, i.e, $\forall g \in \Gamma_1, V(A) \cap g \neq \phi$.

Example. Figure 1 (left) shows a cascade (red edges) on a graph with 10 nodes. There are two pool tests $\Gamma = \{g_1 = \{5, 6, 7\}, g_2 = \{3, 8, 9\}\}$. Suppose nodes 7 and 3 are infected. Then, both the pools g_1 and g_2 will test positive, and $\Gamma_1 = \{g_1, g_2\}$. T_r (red edges) denotes a disease cascade with root r. The subgraph in purple edges in Figure 1(right) denotes a consistent cascade, since it ensures that both positive groups have a node connected to r. The inferred

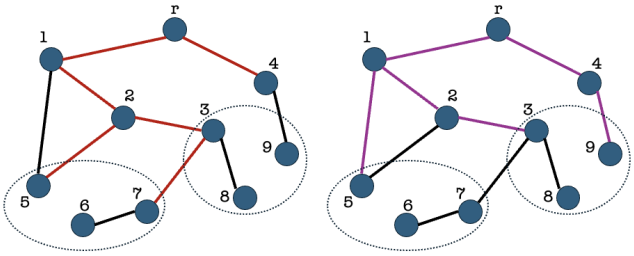

Figure 1: On the left, a cascade (in red) with root r has resulted in two positive-testing pools (in dashed ovals). On the right, a reconstructed cascade T_r is shown (in purple) which is *consistent*, i.e., it contains at least one node from each positive pool. Here, $\delta_{T_r} = \{(3,7),(3,8)\}, \lambda_{T_r} = \{(2,5)\}$. The probability $P(T_r) = p_{r1}p_{15}p_{12}p_{23}p_{r4}p_{49}(1-p_{37})(1-p_{38})$.

cascade in the right is not the same as the one on the left, but shares some nodes.

3.1 Problem formulation: MLE construction for `single-seed` instance

We first define the probability of a cascade, and transform that to a cost, in order to formalize the MLE problem. Let the set of all nodes in the negative pools be $S_0 = \bigcup_{g\in\Gamma_0} g$. Let $N_e(u)$ be the set of edges incident at node u. Given a cascade T_r in $G = (V, E)$, we define: (1) $E(T_r)$: the set of edges in T_r; (2) $E(T_r, S_0)$: the set of all edges with one endpoint in T_r and the other in S_0; (3) δ_{T_r}: the set of edges with one endpoint in T_r and the other in $V \setminus (V(T_r) \cup S_0)$; (4) λ_{T_r}: the set of edges with both endpoints in T_r, but not belonging to T_r. Let $d_{T_r}(u,v)$ denote the distance in T_r between the nodes u and v. Under the IC disease model , the probability of the cascade T_r is:

$$\overline{P}(T_r) = \prod_{e\in E(T_r)} p_e \prod_{e\in E(T_r,S_0)} (1-p_e) \prod_{e\in\delta_{T_r}} (1-p_e) \prod_{\substack{e=(u,v)\in\lambda_{T_r},\\ d_{T_r}(r,u)\neq d_{T_r}(r,v)}} (1-p_e)$$

Example. In the example in Figure 1 (right), the set $E(T_r)$ for the cascade T_r consists of the purple edges.

An MLE cascade is one which maximizes the above probability. Let $c_e = -\log p_e$ and $d_e = -\log(1-p_e)$. As in prior formulations of MLE [23], we formalize the cost of a cascade as

$$\text{Cost}(T) = \sum_{e\in E(T)} c_e + \sum_{e\in E(T,S_0)} d_e + \sum_{e\in\delta_T} d_e + \sum_{e\in\lambda_T} d_e \quad (1)$$

As in [23], we disregard the fact that two neighbors in the cascade equidistant from the source (in the cascade) do not contribute d_e costs.

The PoolCascadeMLE problem. Given an undirected graph $G = (V, E)$, a seed r, a set of pool-tested node groups $\Gamma = \Gamma_0 \cup \Gamma_1$, find a subgraph T_r rooted at r, which is consistent with Γ and minimizes $\text{Cost}(T_r)$.

We also consider a version where the tests are imperfect, i.e., the observations are noisy.

The NoisyPoolCascadeMLE problem. Given an undirected graph $G = (V, E)$, a set of noisy pool-tested node groups $\Gamma = \Gamma_0 \cup \Gamma_1$, find outcomes $\Gamma_0' \cup \Gamma_1'$, and a subgraph T, which is consistent with $\Gamma_0' \cup \Gamma_1'$, and minimizes $\text{Cost}(T|\Gamma_0' \cup \Gamma_1') + \log(1/q(\Gamma_0, \Gamma_1, \Gamma_0', \Gamma_1'))$. Here $q(\Gamma_0, \Gamma_1, \Gamma_0', \Gamma_1')$ denotes the probability that the actual outcomes are (Γ_0', Γ_1') when the observed outcomes are (Γ_0, Γ_1).

3.2 Problem formulation: MLE construction for the `one-hop` instance

For simplicity, we assume the disease starts on nodes in V_0 and spreads to nodes in $V_1 = V \setminus V_0$, and we are given observations from V_1. For a cascade consisting of a subset A of edges, let $V_0(A)$ denote the set of seeds in V_0. Let σ denote the set of edges between V_0 and V_1. We can write the probability of cascade A in terms of the probabilities of the constituent seeding and transmission events as follows.

$$P(A) = \prod_{v\in V_0(A)} p_v^0 \prod_{v\in V\setminus V_0(A)} (1-p_v^0) \prod_{e\in E(A)} p_e \prod_{e\in\sigma_A\setminus E(A)} (1-p_e)$$

Equivalently, we can write the objective in terms of costs. In addition to c_e, d_e, we define $a_v = -\log p_v^0$ as the cost of seeding v, and $b_v = -\log(1-p_v^0)$ as the cost of not seeding v. The cost of a cascade under the one-hop model is,

$$\text{Cost}^1(A) = \sum_{v\in V_0(A)} a_v + \sum_{v\in V\setminus V_0(A)} b_v + \sum_{e\in E(A)} c_e + \sum_{e\in\sigma_A\setminus E(A)} d_e \quad (2)$$

The One-HopCascadeMLE problem. Given a set of observations $\Gamma = (\Gamma_0, \Gamma_1)$, find a cascade A which is consistent and minimizes $\text{Cost}^1(A)$.

4 HARDNESS RESULTS

We show that both PoolCascadeMLE and One-HopCascadeMLE are computationally hard, even to approximate. In contrast, the MLE for the problem without pool testing (equivalently, all pools of size 1), which was considered in [23], can be approximated within an $O(\log n)$ factor.

Theorem 1. *PoolCascadeMLE is hard to approximate within a $O(\log^{2-\epsilon} k)$ factor, for any $\epsilon > 0$ unless P=NP. Here $|\Gamma_1| = k$.*

The proof is deferred to the full version [22]. It is a reduction from the Group Steiner Tree problem.

Theorem 2. *One-HopCascadeMLE is NP-hard to approximate within a $O(\log k)$ factor, unless P=NP.*

The proof is deferred to the full version [22]. It is a reduction from the Minimum Set Cover problem.

5 OUR APPROACH

Due to the significant hardness of the PoolCascadeMLE and One-HopCascadeMLE problems (Section 4), we focus on approximation algorithms here. We make the following natural assumption about the disease regime.

Assumption 1. *The edge transmission probability $p_e \leq 1/2$ for all edges, i.e., $c(e) \geq d(e)$, for all $e \in E$.*

5.1 PoolCascadeMLE problem

Lemma 1. *Under Assumption 1, a PoolCascadeMLE solution $T^{\star}_{r^*}$ is a tree.*

If the solution has a cycle, then we can reduce its cost by removing an edge in the cycle and still have a feasible solution. Hence, the solution is a tree.

Our approach is based on a reduction to the Group Steiner Tree (GST) problem on a graph with node and edge weights. We are given an undirected graph G and a set of pools Γ. We construct graph G' from G by adding node and edge weights as defined in Algorithm 1, ApproxCascade. We remove the set of nodes belonging to the negative pools, S_0. Now our goal is to find a minimum weighted tree in G' that connects to at least one node in each of the positive pools, Γ_1, which is the Group Steiner Tree problem [8, 17]. Lemma 2 gives the bounds on the weight of such a tree in terms of our Cost (Expression (1)). Let $N_e(u)$ be the set of all incident edges on node u.

Lemma 2. *Let T be any Group Steiner Tree in G' with respect to the groups Γ_1. Let the weight of T in G' be denoted as $w(T)$. Then, T is consistent with Γ and*

$$Cost(T) \le w(T) \le 2\ Cost(T)$$

Proof. For any group Steiner tree T on G', we have

$$\begin{aligned}
w(T) &= \sum_{u\in V(T)} w(u) + \sum_{e\in E(T)} w(e) \\
&= \sum_{u\in V(T)} \sum_{e\in N_e(u)} d_e + \sum_{e\in E(T)} (c_e - d_e) \\
&= \sum_{u\in V(T)} \Big[\sum_{e\in N_e(u)\cap E(T)} d_e + \sum_{e\in N_e(u)\cap E(T,S_0)} d_e \\
&\quad + \sum_{e\in N_e(u)\cap \delta_T} d_e + \sum_{e\in N_e(u)\cap \lambda_T} d_e \Big] + \sum_{e\in E(T)} (c_e - d_e) \\
&= 2\sum_{e\in E(T)} d_e + \sum_{e\in E(T,S_0)} d_e + \sum_{e\in \delta_T} d_e + 2\sum_{e\in \lambda_T} d_e \\
&\quad + \sum_{e\in E(T)} (c_e - d_e) \\
&= \mathrm{Cost}(T) + \sum_{e\in E(T)} d_e + \sum_{e\in \lambda_T} d_e \\
&\le 2\,\mathrm{Cost}(T) - \sum_{e\in E(T,S_0)} d_e - \sum_{e\in \delta_T} d_e \\
&\Rightarrow w(T) \le 2\,\mathrm{Cost}(T)\,.
\end{aligned}$$

□

Lemma 3. *(Charikar et al.[8]) There is an approximation preserving reduction from the Group Steiner Tree problem on a node- and edge-weighted graph to the Directed Steiner Tree problem on a purely edge-weighted graph.*

We describe the reduction in Lemma 3 in the full version [22] (Algorithm 3). We convert the original graph into a purely edge-weighted digraph by employing 'in' and 'out' copies of nodes in the usual manner. Then, we add a dummy node for each group, connecting each node in the group to this dummy node with zero-weighted edges. Setting the dummy nodes as terminals, we solve the Directed Steiner Tree problem [8]. Charikar et al. [8] provide the best-known approximation algorithm for the Directed Steiner Tree problem which has an approximation ratio of $O(k^{\epsilon})$ and runs in time $O(kn^{1/\epsilon})$ for a fixed $\epsilon > 0$, where k is the number of terminals and n is the size of the network. This leads to an approximation ratio of $O(k^{\epsilon})$ for our algorithm, ApproxCascade.

Algorithm 1 ApproxCascade

Input: $G = (V, E)$, seed r, edge probabilities p_e, a set of pool-tested node groups $\Gamma = \Gamma_0 \cup \Gamma_1$.
Output: A tree T_r rooted at r and consistent with Γ.

```
for each edge e do
    Compute the cost of inclusion c_e = −log p_e and cost of exclusion d_e = −log(1 − p_e)
end for
Construct a node and edge-weighted graph G' such that,
for each node u ∈ G do
    w(u) ← Σ_{e∈N_e(u)} d_e
end for
for each edge e ∈ G do
    w(e) ← c_e − d_e
end for
Remove nodes ⋃_{g∈Γ_0} g from G'.
T_r = GroupSteinerTree(G', r, Γ_1)
return T_r
```

Theorem 3. *Let $\widehat{T}_r$, rooted at r, be the tree returned by the Algorithm 1. Let $T^*_{r^*}$ be an optimal solution to PoolCascadeMLE, rooted at r^*. Then,*

$$Cost(\widehat{T}_r) \le O(k^{\epsilon})Cost(T^*_{r^*}),$$

where k is the number of positive pools, i.e., $|\Gamma_1|$ and a fixed $\epsilon > 0$.

The proof is in the full version [22].

Time complexity. Algorithm 1 runs in $O(n^2 + kn^{1/\epsilon})$time. At a fixed $\epsilon = 0.5$, the time complexity is $O(kn^2)$.

5.2 One-HopCascadeMLE problem

Let x_i, y_{ij} and z_{ij} denote indicator variables for node $i \in V_0$ being seeded, disease transmission on edge (i, j) and no disease spread on edge (i, j), respectively. We first describe an integer program, which doesn't exactly solve One-HopCascadeMLE, but is a 2-approximation.

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in V_0} (a_i x_i + b_i(1 - x_i)) + \sum_{(i,j)\in E} (c_{ij} y_{ij} + d_{ij}(z_{ij})) \\
\text{subject to} \quad & \sum_{j\in g} \left[\sum_{i:(i,j)\in E} y_{ij} \right] \ge 1, \quad g \in \Gamma_1 \\
& x_i - y_{ij} \ge 0, \quad (i,j) \in E \\
& z_{ij} \ge x_i, \quad (i,j) \in E \\
& x_i \in \{0,1\}, \quad i \in V_0 \\
& y_{ij} \in \{0,1\}, \quad (i,j) \in E
\end{aligned}$$

Algorithm 2 RoundCascade

Input: A One-HopCascadeMLE instance
Output: A cascade A consisting of set $\{X_i : i \in V_0\}$ and edges $\{Y_{ij} : (i,j) \in E\}$

Solve the LP to get $\{x_i^*\}_{i\in V}, \{y_{ij}^*\}_{(i,j)\in E}$.
for each $i \in V$ **do**
  Independently pick a number $\tau_i \in [0,1]$ uniformly at random.
  Set $X_i = 1$ if $\alpha x_i^* > \tau_i$, otherwise set $x_i = 0$.
end for
for each $(i,j) \in E$ **do**
  if $\alpha y_{ij}^* > \tau_i$ **then**
    Set $Y_{ij} = 1, X_i = 1, Z_{ij} = 0$.
  else
    Set $Y_{ij} = 0$ and $Z_{ij} = X_i$
  end if
end for

The first set of constraints corresponds to the requirement that in each group $g \in \Gamma_1$, at least one node should be covered by a live-edge. The second set of constraints ensures that, a live-edge must begin at a seeded node. The third set of constraints incorporates the cost of non-infection.

Lemma 4. *Let x, y, z denote the optimal integral solutions to the above program. Then, $\sum_{i\in V_0}(a_i x_i + b_i(1-x_i)) + \sum_{(i,j)\in E}(c_{ij}y_{ij} + d_{ij}(z_{ij})) \le 2OPT$, where OPT denotes the cost of the optimal solution to the instance of One-HopCascadeMLE.*

The proof is deferred to the full version [22].

LP relaxation. We relax the integrality constraints by replacing them with $x_i, y_{ij} \ge 0, \forall i \in V, (i,j) \in E$. Let $\{x_i^*\}_{i\in V}, \{y_{ij}^*\}_{(i,j)\in E}$ be the optimal LP solution.

Lemma 5. *Let X, Y, Z denote the solution output by RoundCascade. Then, $E[\sum_i (a_i - b_i)X_i + \sum_{ij} c_{ij}Y_{ij} + d_{ij}Z_{ij}] \le \alpha\, OPT^*$ where OPT^* is the optimal objective value of the LP, and $\alpha = 1 + \ln|\Gamma_1|$.*

This follows directly from the randomized rounding. Finally, we show that the solution is feasible.

Lemma 6. *Let A denote the solution computed by RoundCascade. Then, $Pr[A \text{ is infeasible}] \le ke^{-\alpha} \le 1/n^c$ for $k = \Omega(\log n)$.*

Proof. We show that for any $g \in \Gamma_1$, $\Pr[g \text{ is not connected by } A] \le e^{-\alpha}$. We have,

$$\Pr[g \text{ is not connected by } A] = \Pr[\bigcap_{j\in g}(j \text{ not in A})]$$

$$= \prod_{j\in g} \Pr[(j \text{ is not seeded}) \cap (j \text{ is not covered by a live-edge})]$$

$$= \prod_{j\in g} \Pr[j \text{ is not seeded}] \Pr[j \text{ is not covered by a live-edge})]$$

$$= \prod_{j\in g}\left[(1-\alpha x_j^*)\prod_{i:(i,j)\in E}(1-\alpha y_{ij}^*)\right] \le \prod_{j\in g}\left[e^{-\alpha x_j^*}\prod_{i:(i,j)\in E} e^{-\alpha y_{ij}^*}\right]$$

$$= e^{-\alpha\sum_{j\in g}\left(x_j^* + \sum_{i:(i,j)\in E} y_{ij}^*\right)} \le e^{-\alpha}$$

Above, we have used the independence of events, $\{j \text{ not in } A\}_{j\in g}$, and of the events, ($j$ is not seeded) and (j is not covered by a live-edge). In the last line, we have used the first set of constraints in the LP. Therefore, $Pr[A \text{ is infeasible}] = Pr[\bigcup_{g\in\Gamma_1} \text{ g not connected by A}] \le \prod_{j\in X} \Pr[\text{ j is not in A }] \le ke^{-\alpha}$ by union bound. □

Theorem 4. *RoundCascade is a randomized $(2+2\ln k)$-approximation algorithm for One-HopCascadeMLE, where $|\Gamma_1| = k$.*

6 EXPERIMENTAL RESULTS

6.1 Dataset and Methods

We experimentally evaluate the performance of ApproxCascade and RoundCascade using the networks listed in Table 1 and described below.

(1) `Barabasi-Albert (BA) networks`: Since many real-world networks have been observed to be in the class of scale-free networks, BA networks are a useful model. We generate random BA networks with the number of edges for each new node, $m = 3$.
(2) `Erdős-Rényi graphs`: We generate a $G(n = 1000, q = 0.02)$ graph for evaluating ApproxCascade.
(3) `hospital-icu`: This is a contact network between patients and healthcare workers built from Electronic Heath Records (EHR) for the UVA Hospital ICU for the time period between Jan 1, 2018 to Jan 8, 2018.
(4) `small-city`: This is a subgraph of the Virginia (VA) digital twin-based network made available at [1] and used previously by [10, 16], created on a synthetic population based on census data and community surveys. Each edge e has a contact duration d_e, which is used to generate the edge-diffusion probability $p_e = 1 - e^{-\beta d_e}$; β is a parameter to control transmissibility.

Baselines. Since this problem has not been considered before, we design the following baselines.

(1) ApproxCascade-Random and RoundCascade-Random: Instead of choosing the infected nodes in a positive pool cost-effectively, randomly choose one node to be the only infection in that pool, reducing to a pool size of 1.
(2) ApproxCascade-All: Here, we consider every node in a positive pool to be infected, also reducing to pool sizes of 1.

Both these baselines reduce the pool size to 1 which can now be solved by the state-of-the-art method in [23]. For speed and scalability, we deploy Charikar et. al.'s node-weighted Steiner tree solver as required by that method [8]. Our goal is to understand the conditions where it is important to systematically select the likely infections within a pool.

Method. We randomly select a fixed proportion of nodes in a given network to be pooled according to a parameter, *pool ratio* set to $\{0.5, 0.9\}$. We randomly bunch these selected nodes into equal-sized pools whose size is determined by a parameter, *pool size* which can be $\{3, 5, 7, 9\}$.

ApproxCascade: We generate IC simulations starting at a random seed and determine the subset of positive pools which are pools having at least one infected node.

RoundCascade: We first construct a time-expanded version of the network, $G' = ((V_0', V_1'), E')$, where each node in $u \in V$ corresponds to two timestamped copies in $u_0 \in V_0'$, $u_1 \in V_1'$, and each undirected

Table 1: Networks and their properties. Shortest path length for the small-city network is omitted as its relevance depends on the diffusion rate β.

Graph Name	Nodes	Edges	Clustering coefficient	Avg. shortest path length	Note
BA $m = 3$	1000	2991	0.036	3.477	Synthetic network
$G(n, q)$ random graph	1000	9974	0.0197	2.6398	Synthetic network
`hospital-icu`	879	3575	0.59	4.31	Real-world network
`small-city`	10001	52575	0.349	–	Realistic synthetic weighted network.

edge (u, v) corresponds to two copies $(u_0, v_1), (v_0, u_1) \in E'$. On G', which is an oriented bipartite network, we assume the pooled observations come only from the V_1' part. To generate infections, we randomly choose seeds from V_0' with homogenous probability $p^0 \in \{0.01, 0.05, 0.10\}$, starting from which we simulate for one time-step.

For the unweighted networks, we use homogeneous diffusion probabilities $p = \{0.01, 0.05, 0.10, 0.20\}$ and for the weighted network, we use the parameter $\beta = \{2, 3, 5, 7\} \times 10^{-6}$. We give the network, the disease parameters, the positive and negative pools (and in case of ApproxCascade, the seed) as input to our algorithms as well as the baselines.

Performance measures. To evaluate the performance of our method, we have two tasks:

1. *Missing infection recovery*: We compare the node set of the reconstructed subgraph with the ground truth infection set. As metrics, we choose F1-score to quantify the success in missing infection recovery, as used in previous works [18, 23, 27]. These are defined as follows. Let the node-set of the reconstructed cascade V_T and the ground truth infected node-set V_G. Define true positives as $TP = |V_T \cap V_G|$, true negatives as $TN = |V \setminus (V_T \cup V_G)|$, false positives as $FP = |V_T \setminus V_G|$, and false negatives as $FN = |V_G \setminus V_T|$. We compute F1-score = $2TP/(2TP + FP + FN)$.

2. *Prevalence estimation*: We compare the sizes of the reconstructed cascade and the ground truth cascade using the relative error $e_{rel} = (|V_G| - |V_T|)/|V_G|$.

The presented results are averaged over 50 replicate runs.

6.2 Results

ApproxCascade

Effect of network structure and diffusion probability. In Figure 2, we report the F1-scores for ApproxCascade and the baselines for different networks. Across networks, we observe that ApproxCascade vastly outperforms the baselines under the low diffusion probability conditions. The performance gap is highest on the large weighted network `small-city`. However, in high probability regimes, the performance suffers especially against ApproxCascade-All. This could be explained by the fact that when the underlying cascade is small, only a small subset of nodes in a positive pool are actually infected. In this scenario, choosing infections carefully with ApproxCascade yields dividends. On the other hand, when the cascade is large and almost all nodes in a pool are infected, the PoolCascadeMLE solution under-selects, trying to keep costs low. The exact threshold beyond which ApproxCascade-All is superior is connected to the diffusion probability threshold beyond which large cascades are common. For example, on $G(n = 1000, q = 0.02)$, this transition is in $(0.05, 0.10)$ whereas on BA $(n = 1000, m = 3)$, it is between $(0.10, 0.20)$.

Pool size and pool ratio. In Figure 2, we plot the F1-scores of all the methods versus the pool size. On most networks, ApproxCascade is consistently better than the baselines for all pool sizes. This shows that there is a range of pool sizes for which ApproxCascade yields good solutions. Secondly, increasing the pool size leads to degradation in the performance. This is to be expected as the pools get bigger, the number of possible subsets of infected nodes grows exponentially. This makes it likelier to obtain reconstructed node-sets that have less overlap with the ground truth. There is little impact of the pool ratio on the relative performance of our method against the baselines. The performance improves with higher pool ratio, which is due to the fact that there are fewer unobserved infections outside the pooled set.

Estimating prevalence. How well does ApproxCascade perform as a method for prevalence estimation? Figure 3 shows the prevalence e_{rel} scores obtained by the methods with respect to the ground truth cascade size. We find that ApproxCascade outperforms the baseline in nearly all regimes, remaining within $[-0.5, 0.5]$ in a wide range of cascade sizes. ApproxCascade tends to underestimate the cascade size which is typical for a MLE cascade method. Like in missing infection recovery, the gap is wider for low diffusion probabilities and narrower for the $G(n, q)$ network. Also, ApproxCascade-All massively overestimates in this task (plots moved to the full version [22]).

RoundCascade

We report the F1-scores obtained by RoundCascade and the random baseline in Figure 4 on the missing infections recovery task. RoundCascade outperforms the baseline across disease regimes and networks. The performance gap is consistent across varying seeding probabilities and edge-diffusion probabilities.

7 LIMITATIONS OF THE MLE APPROACH

Observation 5. *There exist instances where PoolCascadeMLE solution does not recover any part of the ground truth cascade.*

In Figure 5, the testing pool, shown by the dotted circle around the infected leaf nodes, leads to a positive test. As a PoolCascadeMLE solution needs to connect the root r to at least one node in the pool,

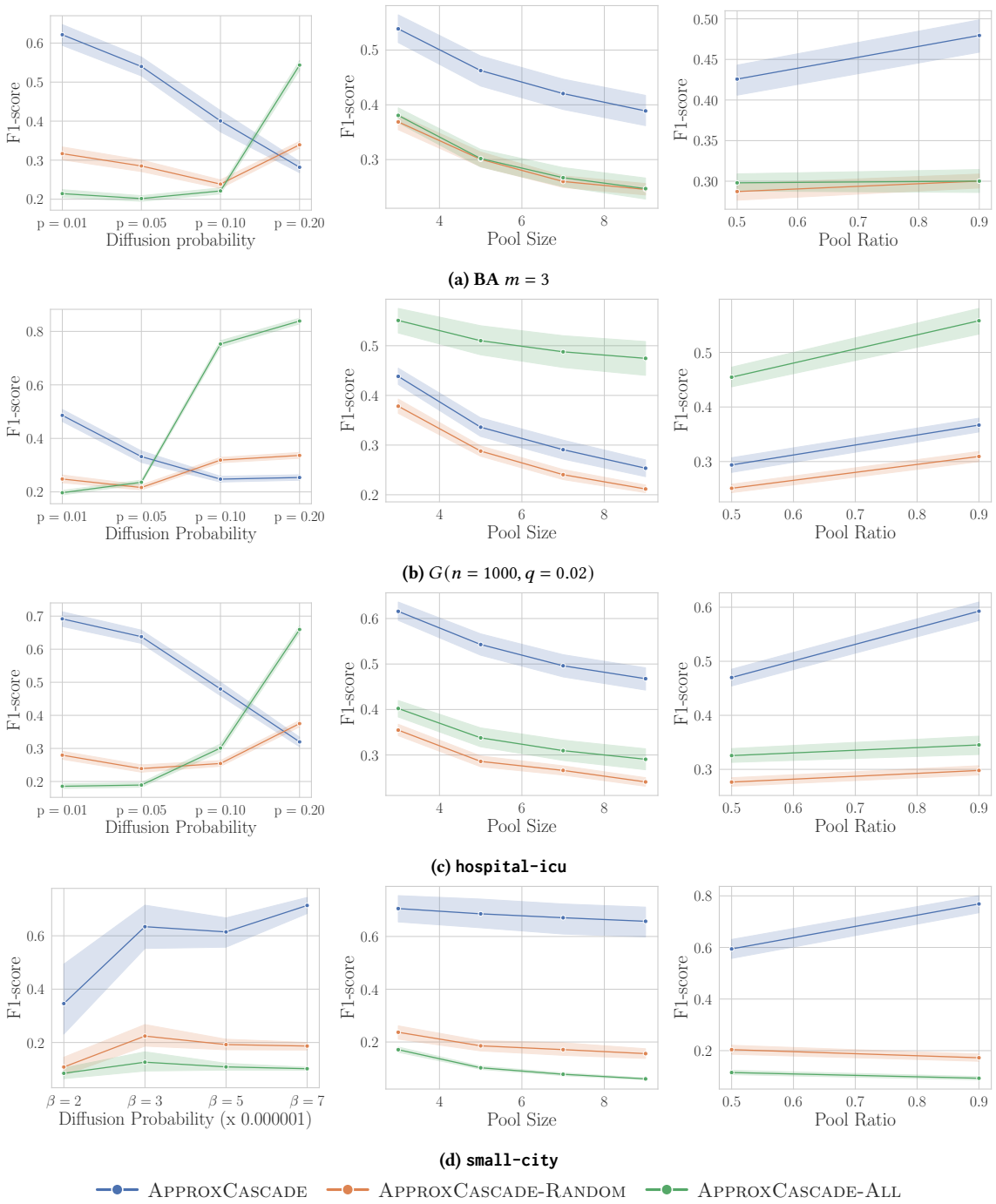

Figure 2: Performance of ApproxCascade against baselines.

while minimizing the Cost, the green path is picked over the higher cost ground truth tree shown in red. This arises due to the infected leaf nodes being pooled together whereas if they were individually tested, the MLE cascade would be closer to the ground truth.

Impact of noisy testing. We demonstrate an example showing that the MLE solution can be very different compared to the ground truth in a noisy test condition. The example network $G(V, E)$ consists of three sets of nodes– $V = \{u\} \cup V' \cup V''$, where u is a special *central* node, $V' = v_1, \ldots, v_k$ such that V' induces the path $v_1 v_2 \cdots v_k$, and $V'' = \{w_{ij} \mid i, j = 1, \cdots, k\}$ such that $u w_{i1} \cdots w_{ik} v_i$ is an induced path for all $i = 1, \cdots, k$. Let each $P_i = \{v_i, w_{ik}\}$ be a

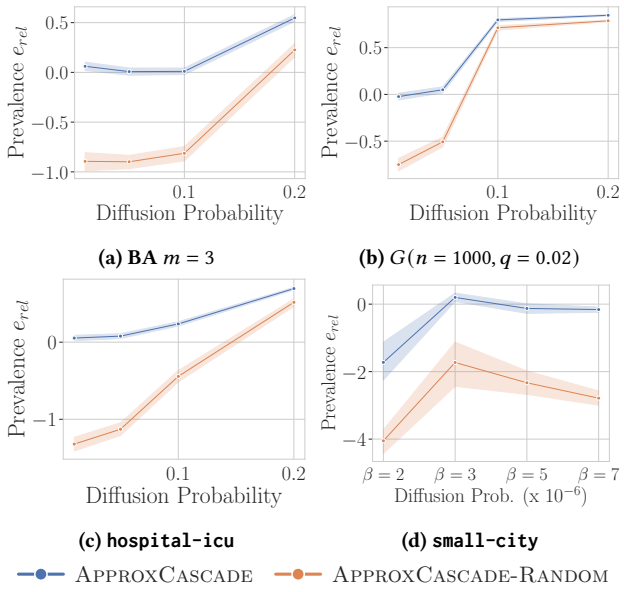

Figure 3: Performance comparison in terms of relative error in prevalence estimation, e_{rel}.

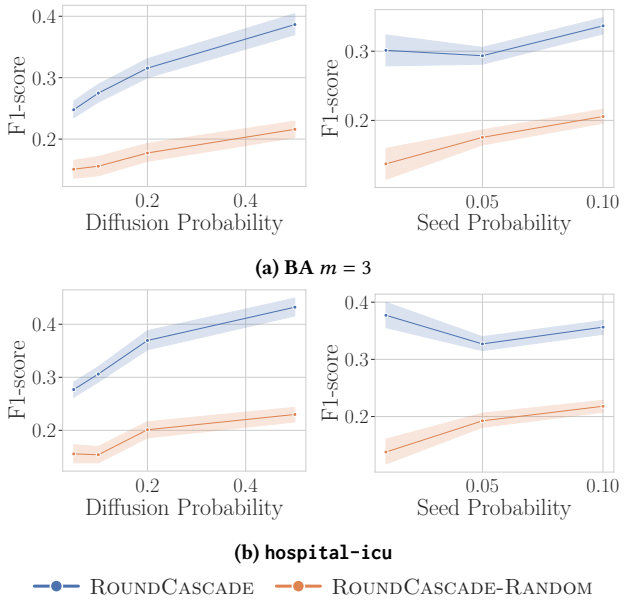

Figure 4: Performance of RoundCascade vs. baseline.

pool. There are exactly k pools. Also, let the false positive probability $q_{fp} = 0$, while the false negative probability $q_{fn} > 0$. Suppose that the cascade graph is precisely the graph $v_1v_2 \cdots v_k$. Then, all pool tests will be positive. However, the probability that at least one of the pools in $\{P_2, P_3, \cdots, P_{k-1}\}$ is reported as negative while P_1 and P_k are reported as positive is $(1 - q_{fn})^2\big(1 - (1 - q_{fn})^{k-2}\big)$. If this "bad" event occurs, then, any MLE solution must contain v_1 and v_k, which cannot be connected via the path $v_1v_2 \cdots v_k$. This in turn implies that it must involve the central node w and at least $2k$ nodes from V''. Thus the MLE solution differs from ground truth by more than 50% of the nodes. Note that the probability of the bad event can be made arbitrarily high by increasing k.

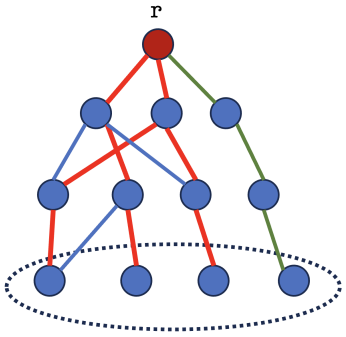

Figure 5: Ground truth cascade shown in red. PoolCascadeMLE solution shown in green. The testing pool is the dotted circle around the leaf nodes.

Observation 6. *There exist instances for which the solution to the NoisyPoolCascadeMLE has $o(n)$ overlap with the ground truth, compared to a solution to PoolCascadeMLE.*

The above construction implies that the MLE solution under noise is significantly different from that without noise. It can be shown that for the above instance, carefully chosen pool tests involving the nodes on the path can ensure that the path is inferred as being infected. This implies that overlapping tests are more powerful than non-overlapping tests for the epidemic reconstruction problem. This is a significant contrast with the observation in [13] that non-overlapping pools are approximately optimal in the welfare maximization problem.

8 CONCLUSION

We introduce the PoolCascadeMLE problem for reconstructing an epidemic outbreak under group surveillance. Despite a lot of work in group surveillance, including for infectious diseases, wastewater monitoring, the problem of reconstructing an outbreak hasn't been studied before. PoolCascadeMLE is NP-hard to even approximate, and we design an approximation algorithm using Group Steiner tree techniques. We consider One-HopCascadeMLE, which is a special case for an IC process that can spread for a maximum of one-hop from the unknown seeds. We find that even this problem is NP-hard to approximate and use randomized rounding on an LP relaxation to achieve a logarithmic approximation bound. Experiments on synthetic and real contact networks, including a hospital contact network show that our methods which systematically connect infections in a pool considerably outperform baselines which do not do so. We also show that noise has a very significant impact on the MLE solution. While our work assumes that the pools are given, designing optimal pools in the context MLE algorithms for better inference of the outbreak is an interesting future direction that can help improve surveillance strategies.

ACKNOWLEDGMENTS

This research is partially supported by NSF grants CCF-1918656 and CNS-2317193, and DTRA award HDTRA1-24-R-0028, Cooperative Agreement number 6NU50CK000555-03-01 from the Centers for Disease Control and Prevention (CDC) and DCLS, Network Models of Food Systems and their Application to Invasive Species Spread, grant no. 2019-67021-29933 from the USDA National Institute of Food and Agriculture, Agricultural AI for Transforming Workforce and Decision Support (AgAID) grant no. 2021-67021-35344 from the USDA National Institute of Food and Agriculture.

REFERENCES

[1] A. Adiga, H. Baek, S. Eubank, P. Porebski, M. Marathe, H. Mortveit, S. Swarup, M. Wilson, and D. Xie. 2022. Synthetic population for USA_VIRGINIA. https://zenodo.org/records/6505964.

[2] Matthew Aldridge, Oliver Johnson, Jonathan Scarlett, et al. 2019. Group testing: an information theory perspective. *Foundations and Trends® in Communications and Information Theory* 15, 3-4 (2019), 196–392.

[3] Fabian Amman, Rudolf Markt, Lukas Endler, Sebastian Hupfauf, Benedikt Agerer, Anna Schedl, Lukas Richter, Melanie Zechmeister, Martin Bicher, Georg Heiler, et al. 2022. Viral variant-resolved wastewater surveillance of SARS-CoV-2 at national scale. *Nature biotechnology* 40, 12 (2022), 1814–1822.

[4] Benjamin D Anderson, John A Lednicky, Montserrat Torremorell, and Gregory C Gray. 2017. The use of bioaerosol sampling for airborne virus surveillance in swine production facilities: a mini review. *Frontiers in Veterinary Science* 4 (2017), 121.

[5] Batuhan Arasli and Sennur Ulukus. 2023. Group testing with a graph infection spread model. *Information* 14, 1 (2023), 48.

[6] Fabjola Bilo, Annalisa Zacco, Paola Cirelli, Filippo Marciano, Giuseppe Tomasoni, Sara Comai, Giorgio Vassena, Luca Perfetti, Antonio Gualtiero Mainardi, Silvia Mastrolembo Ventura, et al. 2025. Indoor air sampling strategies by active bioaerosol samplers: a case study to detect SARS-CoV-2 in hospital settings. *Journal of Aerosol Science* 189 (2025), 106618.

[7] Eusebi Calle, David Martínez, Roser Brugués-i Pujolràs, Miquel Farreras, Joan Saló-Grau, Josep Pueyo-Ros, and Lluís Corominas. 2021. Optimal selection of monitoring sites in cities for SARS-CoV-2 surveillance in sewage networks. *Environment International* 157 (2021), 106768.

[8] Moses Charikar, Chandra Chekuri, To-yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. 1999. Approximation Algorithms for Directed Steiner Problems. *Journal of Algorithms* 33, 1 (1999), 73–91. https://doi.org/10.1006/jagm.1999.1042

[9] Chen Chen, Yunfan Wang, Gursharn Kaur, Aniruddha Adiga, Baltazar Espinoza, Srinivasan Venkatramanan, Andrew Warren, Bryan Lewis, Justin Crow, Rekha Singh, et al. 2024. Wastewater-based epidemiology for COVID-19 surveillance and beyond: A survey. *Epidemics* 49 (2024), 100793.

[10] Jiangzhuo Chen, Stefan Hoops, Henning S Mortveit, Bryan L Lewis, Dustin Machi, Parantapa Bhattacharya, Srinivasan Venkatramanan, Mandy L Wilson, Chris L Barrett, and Madhav V Marathe. 2025. Epihiper—A high performance computational modeling framework to support epidemic science. *PNAS nexus* 4, 1 (2025), pgae557.

[11] Robert Dorfman. 1943. The Detection of Defective Members of Large Populations. *The Annals of Mathematical Statistics* 14, 4 (1943), 436–440. http://www.jstor.org/stable/2235930

[12] Dingzhu Du, Frank K Hwang, and Frank Hwang. 2000. *Combinatorial group testing and its applications*. Vol. 12. World Scientific.

[13] Simon Finster, Michelle González Amador, Edwin Lock, Francisco Marmolejo-Cossío, Evi Micha, and Ariel D. Procaccia. 2023. Welfare-Maximizing Pooled Testing. arXiv:2206.10660 [cs.GT]

[14] Aikaterini Galani, Reza Aalizadeh, Marios Kostakis, Athina Markou, Nikiforos Alygizakis, Theodore Lytras, Panagiotis G Adamopoulos, Jordan Peccia, David C Thompson, Aikaterini Kontou, et al. 2022. SARS-CoV-2 wastewater surveillance data can predict hospitalizations and ICU admissions. *Science of The Total Environment* 804 (2022), 150151.

[15] Margaret A Honein, Sonja J Olsen, Daniel B Jernigan, and Demetre C Daskalakis. 2024. Challenges and opportunities for wastewater monitoring of influenza viruses during the multistate outbreak of highly pathogenic avian influenza A (H5N1) virus in dairy cattle and poultry. , 1309–1312 pages.

[16] Stefan Hoops, Jiangzhuo Chen, Abhijin Adiga, Bryan Lewis, Henning Mortveit, Hannah Baek, Mandy Wilson, Dawen Xie, Samarth Swarup, Srinivasan Venkatramanan, et al. 2021. High Performance Agent-Based Modeling to Study Realistic Contact Tracing Protocols. In *2021 Winter Simulation Conference (WSC)*, Kim Sojung, Ben Feng, Katy Smith, Sara Masoud, and Zeyu Zheng (Eds.). IEEE, 1–12.

[17] Edmund Ihler. 1991. Bounds on the quality of approximate solutions to the group Steiner problem. In *Graph-Theoretic Concepts in Computer Science*, Rolf H. Möhring (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 109–118.

[18] Hankyu Jang, Shreyas Pai, Bijaya Adhikari, and Sriram V. Pemmaraju. 2021. Risk-aware Temporal Cascade Reconstruction to Detect Asymptomatic Cases : For the CDC MInD Healthcare Network. In *2021 IEEE International Conference on Data Mining (ICDM)*. 240–249. https://doi.org/10.1109/ICDM51629.2021.00034

[19] Pruthvi Kilaru, Dustin Hill, Kathryn Anderson, Mary B Collins, Hyatt Green, Brittany L Kmush, and David A Larsen. 2023. Wastewater surveillance for infectious disease: a systematic review. *American journal of epidemiology* 192, 2 (2023), 305–322.

[20] Joshua I Levy, Kristian G Andersen, Rob Knight, and Smruthi Karthikeyan. 2023. Wastewater surveillance for public health. *Science* 379, 6627 (2023), 26–27.

[21] Souci Louis. 2024. Wastewater surveillance for influenza A virus and H5 subtype concurrent with the highly pathogenic avian influenza A (H5N1) virus outbreak in cattle and poultry and associated human cases—United States, May 12–July 13, 2024. *MMWR. Morbidity and Mortality Weekly Report* 73 (2024).

[22] Ritwick Mishra, Abhijin Adiga, and Anil Vullikanti. 2026. Reconstructing Network Outbreaks under Group Surveillance. arXiv:2602.11419 [cs.SI] https://arxiv.org/abs/2602.11419

[23] Ritwick Mishra, Jack Heavey, Gursharn Kaur, Abhijin Adiga, and Anil Vullikanti. 2023. Reconstructing an Epidemic Outbreak Using Steiner Connectivity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 11613–11620.

[24] Leon Mutesa, Pacifique Ndishimye, Yvan Butera, Jacob Souopgui, Annette Uwineza, Robert Rutayisire, Ella Larissa Ndoricimpaye, Emile Musoni, Nadine Rujeni, Thierry Nyatanyi, et al. 2021. A pooled testing strategy for identifying SARS-CoV-2 at low prevalence. *Nature* 589, 7841 (2021), 276–280.

[25] Ruizhong Qiu, Dingsu Wang, Lei Ying, H. Vincent Poor, Yifang Zhang, and Hanghang Tong. 2023. Reconstructing Graph Diffusion History from a Single Snapshot. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. https://doi.org/10.1145/3580305.3599488

[26] A.P. Robinson, T. Walshe, M.A. Burgman, and M. Nunn. 2017. *Invasive Species: Risk Assessment and Management*. Cambridge University Press. https://books.google.com/books?id=LYThDgAAQBAJ

[27] Polina Rozenshtein, Aristides Gionis, B. Aditya Prakash, and Jilles Vreeken. 2016. Reconstructing an Epidemic Over Time. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 1835–1844. https://doi.org/10.1145/2939672.2939865

[28] Daniel K Sewell. 2022. Leveraging network structure to improve pooled testing efficiency. *Journal of the Royal Statistical Society Series C: Applied Statistics* 71, 5 (2022), 1648–1662.

[29] Devavrat Shah and Tauhid Zaman. 2011. Rumors in a Network: Who's the Culprit? *Information Theory, IEEE Transactions on* 57 (09 2011), 5163 – 5181. https://doi.org/10.1109/TIT.2011.2158885

[30] Jeffrey Shaman et al. 2020. An estimation of undetected COVID cases in France. *Nature* 590 (2020), 38–39. Issue 7844.

[31] Angela Felicia Sunjaya and Anthony Paulo Sunjaya. 2020. Pooled testing for expanding COVID-19 mass surveillance. *Disaster Medicine and Public Health Preparedness* 14, 3 (2020), e42–e43.

[32] Sunyu Wang, Ke Xu, and Yulun Zhou. 2024. Cost-effective sensor placement optimization for large-scale urban sewage surveillance. *Sustainable Cities and Society* (2024), 105250.

[33] Marlene K Wolfe, Dorothea Duong, Bridgette Shelden, Elana MG Chan, Vikram Chan-Herur, Stephen Hilton, Abigail Harvey Paulos, Xiang-Ru S Xu, Alessandro Zulli, Bradley J White, et al. 2024. Detection of hemagglutinin H5 influenza A virus sequence in municipal wastewater solids at wastewater treatment plants with increases in influenza A in spring, 2024. *Environmental Science & Technology Letters* (2024).

[34] Kai Zhu and Lei Ying. 2014. Information source detection in the SIR model: A sample-path-based approach. *IEEE/ACM Transactions on Networking* 24, 1 (2014), 408–421.