End-to-End Risk-aware Reinforcement Learning to Detect Asymptomatic Cases in Healthcare Facilities

Yongjian Zhong
Department of Computer Science
University of Iowa
Iowa City, USA
yongjian-zhong@uiowa.edu

Weiyu Huang
Department of Computer Science
University of Iowa
Iowa City, USA
weiyu-huang@uiowa.edu

Bijaya Adhikari
Department of Computer Science
University of Iowa
Iowa City, USA
bijaya-adhikari@uiowa.edu

***Abstract*—This paper studies the problem of detecting asymptomatic cases in epidemic outbreaks within healthcare facilities. Asymptomatic cases pose a significant obstacle in our fight against epidemic outbreaks as they drive latent infection spread, are challenging to surveil, and are hard to intervene against. Detecting asymptomatic cases is challenging for numerous reasons, including lack of data except for large-scale serological surveys, poor generalization from symptomatic cases, and bias towards symptomatic cases in existing epidemiological datasets. Prior works fail in accurately detecting asymptomatic cases as they ignore individual risk factors, ignore the infection transmission pathways, or fail to integrate the two in a principled manner. Here, we formulate the asymptomatic case detection problem over a temporal network as a Prize Collecting Steiner tree with learnable latent prizes, where the latent prizes correspond to individual risks and the edge weights represent the cost of infection transmission. We translate the problem into an equivalent bi-level reinforcement learning problem and propose a deep Q-learning algorithm to tackle the problem.**

To demonstrate the efficacy of our proposed approach, we conduct extensive experiments on real-world networks derived from healthcare facilities. Our experiments over simulated healthcare-associated outbreaks of *Clostridioides difficile infection* (CDI) and COVID-19 reveal that the proposed approach has significant advantages over all the state-of-the-art baselines. Our approach outperforms the closest competitor by up to 29.62 % in detecting asymptomatic cases, which leads to more accurate predictions of symptomatic cases. In our experiments, we also demonstrate that accurately detecting asymptomatic cases leads to more accurate prediction of symptomatic cases. Finally, our case study in real CDI outbreak reveals that the asymptomatic cases detected by our approach are indeed high risk cases.

***Index Terms*—Prize Collecting Streiner Tree, Reinforcement Learning, Epidemic Prediction.**

I. Introduction

A significant challenge in combatting epidemic and pandemic outbreaks is our inability to identify asymptomatic spreaders accurately. It is now known that asymptomatic transmissions were a driving force behind the COVID-19 pandemic [1], [2], [3]. In fact, nearly 32.40% of COVID-19 infections are believed to have been mild or completely asymptomatic [4]. Asymptomatic infections also pose a significant obstacle in our fight against global influenza endemic, regional endemics of infectious diseases (e.g. malaria, trachoma, syphilis, and measles), and healthcare-associated infections (HAIs) such as *methicilin-resistenat Staphylococcus aureus* (MRSA) and *Clostridiodes difficile* (C. diff).

The focus of this paper is on detecting asymptomatic cases in epidemic outbreaks occurring within healthcare facilities. These include HAI outbreaks and nosocomial spread of infectious diseases such as COVID-19 and influenza. From an application viewpoint, it is a significant problem to solve for two primary reasons.

1) As evidenced by the COVID-19 pandemic and HAI outbreaks, healthcare facilities and long-term care facilities are at an increased risk of infection as they consist of vulnerable populations [5], [6].
2) Healthcare providers (nurses, doctors, and caregivers) getting infected in an epidemic outbreak is a significant obstacle in combatting epidemic outbreaks [7]. Hence, the detection of asymptomatic cases in healthcare facilities can help us better secure our limited healthcare capacity.

Despite its importance in infectious disease control, the problem of accurately identifying asymptomatic cases within healthcare facilities still remains open.

Identifying asymptomatic infections is challenging for a number of reasons [8], [9]. These include: (C1) **Data scarcity:** as asymptomatic infections are not easily observed, 'ground truth' data is essentially non-existent, which poses a significant obstacle in leveraging off-the-shelf machine learning algorithms. (C2) **Poor generalization from symptomatic cases:** The risk factors for symptomatic infections could be very different from those of asymptomatic infections; hence, only using identified risk factors for symptomatic infections is not a valid approach for identifying asymptomatic infections. For example, senior citizens were highly susceptible to symptomatic COVID-19 infections; however, they were less likely to have asymptomatic infections. (C3) **Bias towards symptomatic cases:** As mentioned in [10], severe cases get more attention when it comes to testing when capacity is limited. This introduces a bias towards positive symptomatic cases in the already limited data.

This paper overcomes the challenges mentioned above by proposing the first-ever end-to-end machine learning algorithm to detect asymptomatic cases, which harnesses both the cascade structure and risk information without making any as-

sumptions (e.g., [9] trains an ML model on symptomatic cases and takes 'false positives' as candidates for asymptomatic cases). Notably, our approach is entirely data-driven, setting it apart from existing approaches [9], [11], [12] heavily reliant on domain-specific knowledge.

Moreover, we take advantage of the individual-level features, confirmed cases, and contact patterns required to detect asymptomatic cases while overcoming the abovementioned challenges accurately available in healthcare facilities. Individual features such as demographic information and medical history are necessary to estimate personalized risks of asymptomatic infections [13]. On the other hand, contact patterns constructed from healthcare provider mobility (such as terminal logins and check-ins) and patient admissions, transfers, and discharge records are necessary to infer transmission pathways [14]. Here, we propose an end-to-end system to principally integrate both of these data sources to identify asymptomatic cases accurately.

Background and related work: There are a significant number of prior works that study a very closely related problem of inferring missing infections on infection/information cascades. Rozhenstein et al. proposed a directed Steiner-tree-based approach on a time-expanded contact network, where the observed infections corresponded to the terminals and inferred Steiner points were interpreted to be the missing infections [11]. Similar Steiner-tree-based approaches have been used in static [15] and probabilistic [16], [17] settings. Some prior works vie to infer missing infections [18], [19] via identification of the source of the infections [20]. A significant drawback of leveraging these approaches for asymptomatic infection detection is that they do not consider the risk factors of symptomatic and asymptomatic infections. A separate line of work models latent infection states as a function of contact network neighborhood [14], pathogen exposure [21], and node features [13]. However, these approaches assume the underlying contact network is static and/or has limited influence exertion assumptions (such as disease does not spread via a chain of infections). Jang et al. recently proposed a risk-aware Steiner-tree-based approach to incorporate the effects of individual risks (modeled as a feed-forward network over node features) and disease spread (modeled as Steiner-tree). A key limitation of their approach is that the two steps involved in estimating individual risks and inferring asymptomatic cases as nodes in the Steiner tree are independent; thus, it does not overcome the challenges (C2) and (C3) listed above.

Here, we first pose an asymptomatic case detection problem as a *prize-collecting Steiner-tree problem* with *latent prizes on each node* corresponding to the likelihood of being asymptomatic and *weights on each edge* corresponding to the probability of infection spread. We present an end-to-end reinforcement learning framework to solve the problem. Our approach learns the risk of infections from observed infections and then reconstructs the most likely cascade connecting the observed infections with identified asymptomatic infections. As we learn to estimate the risk of infections and to reconstruct the latent cascade together, we overcome all three challenges mentioned above. Specifically, to overcome the challenge (C1), we use observed symptomatic infections (like prior approaches), and for challenges (C2 and C3), we regularize risk estimation and cascade reconstruction with each other in an end-to-end manner. Our contributions are:

- We translate *prize-collecting Steiner-tree problem* for asymptomatic case detection as a bi-level reinforcement learning problem, where the upper level has reinforcement learning objective to solve prize collecting Steiner-tree problem, and the lower level corresponds to learning the node prizes.
- We propose an iterative deep Q-learning algorithm to tackle the bi-level reinforcement learning problem.
- We conduct extensive experiments on networks extracted from real hospitals to demonstrate the efficacy of our algorithm. Our experiments demonstrate that we outperform all state-of-the-art baselines. Our case study reveals that reconstructed cascades are meaningful and identify nodes with high risks of being asymptomatic carriers.

The rest of the paper is organized in the standard way. We present Problem Formulation in Section II followed by our proposed approach in Section III. We then present our empirical findings in Section IV, Finally, we present discussions and conclusions in Section VI.

II. Problem Formulation

We are given a who-visits-whom-when interaction log from a healthcare facility which is represented as a temporal contact network $\mathcal{G} = \{G_1, G_2, \ldots, G_{T-1}, G_T\}$. In each network $G_i = (V_i, E_i, W_i, F_i)$, the interactions between individuals $v_a \in V_i, v_b \in V_i$ at time i is represented by an edge $(v_a, v_b) \in E_i$. Each edge (v_a, v_b) in E_i has a corresponding weight representing the probability of infection flow in either direction. In our experiments (and broadly in practice), this probability is computed as a function of disease parameters (such as infectivity) and edge characteristics (such as duration of contact). Here, we abstract away from these details and assume the probability of infection flow along each edge is part of the input. Each node $v \in V_i$ also has an associated feature vector $F_i[v]$, which represents the node's risk factors such as demographic and medical history. Note that as some demographic information (age, location of residence e.t.c.) and medical history (prescription, comorbidity e.t.c.) change over time, the features are best modeled varying over time.

Now, imagine a disease spreading over $\mathcal{G}$ possibly from multiple sources at different times. At each discrete time i, a set of nodes $\mathcal{I}_i \subset V$ get infected. A majority of these infections occur via contacts in $\mathcal{G}$, some possibly occur due to external factors. However, only a fraction of nodes in $\mathcal{I}_i$ are symptomatic and are observed, i.e., only $\mathcal{S}_i \subset \mathcal{I}_i$ is revealed to us at each time i. The remaining infected nodes $\mathcal{A}_i = \mathcal{I}_i \backslash \mathcal{S}_i$ are deemed to be hidden asymptomatic infections. Note that case severity, including asymptomaticity, depends to some extent on individual features F_i. Hence, we assume a latent probabilistic function of F_i determines the likelihood of a node being symptomatic or asymptomatic. *At a high level,*

asymptomatic case detection can be stated as identifying nodes in $\mathcal{A}_i$ for all i, given $\mathcal{G}$ and $\cup_i \mathcal{S}_i$.

Several prior works have established the equivalence between the temporal graph $\mathcal{G}$ and its time-expanded static version $G_s(V_s, E_s, W_s, F_s)$ when it comes to disease and/or information spread [9], [22], [23]. G_s is a directed static-graph with the nodes in $\mathcal{G}$ organized in layers. There are T time stamps in $\mathcal{G}$, hence there are a total of $T+1$ layers in G_s. Each layer l in G_s has a copy $v_l \in V_s$ of each node $v \in V$ in $\mathcal{G}$. Hence, $|V_s| = (T+1)|V|$. Undirected edge (u, v) in E_l translates to two edges between layers l and $l+1$, i.e., (u, v) in E_l maps to two directed edges (u_l, v_{l+1}) and (v_l, u_{l+1}). Finally, each consecutive copy of a node is connected via a directed edge, i.e., $(v_l, v_{l+1}) \in E_s, \forall v, l$. Similarly, the edge weights W_i translate to W_s and the node features F_i translate to F_s. Note that the transformation preserves the infection time of the nodes in $\mathcal{I}_i$ and the observation time of nodes in $\mathcal{S}_I$. Let $I_s \in V_s$ be the set of nodes in G_s corresponding to the observed infections $\cup_{i+1}^{T} \mathcal{I}_i$ in $\mathcal{G}$. A thing to consider here is that there could be more than one cascade leading to the infections in $\mathcal{G}$ hence, the nodes in I_s. To allow for the detection of more than one cascade, we follow the approach in prior works [9], [11] and add a dummy node r and connect directed edge from r to all nodes $v_0 \in V_s$ with an edge-weight of γ. Now the asymptomatic case detection problem (with potentially more than one origin) we are interested in can be posed as the following rooted prize-collecting Steiner-tree problem where the prizes are defined by a function of the features.

Problem 1: (ROOTED DIRECTED PRIZE-COLLECTING STEINER TREES WITH LATENT PRIZES) Given a time-expanded network $G_s(V_s, E_s, W_s, F_s)$ and observed infections $\mathcal{I} \subset V_s$. Find a tree $T* \subset G_s$ rooted at r and spanning I such that:

$$T* = \arg\min_{T} \sum_{(v_a, v_b) \in T} W_s(v_a, v_b) + \alpha \sum_{v_c \in V_s \setminus T} f(F_{v_c}) \quad (1)$$

where $f(F_{v_c})$ is the probability that node $v_c \in G_s$ is asymptomatic.

Problem 1 is very challenging to solve. The special case of the problem with $\alpha = 0$ is the standard Directed Steiner-Tree problem, which is NP-complete and known to be difficult to approximate in the worst case [24].

III. OUR APPROACH

Our main idea here is to translate Problem 1 into a bi-level reinforcement learning problem where (i) the upper-level objective corresponds to the Directed Prize Collecting Steiner Tree objective and (ii) the lower-level objective corresponds to the estimation of $f(\cdot)$. In recent years, an increasing number of studies have demonstrated that reinforcement learning can be used to solve combinatorial optimization problems including classical problems such as traveling salesman problems [25], [26], max cut [27], [28], graph summarization [29], integer programming [30], just to name a few. Since Problem 1 is a combinatorial optimization problem with a latent component, reinforcement learning is an ideal fit. Next, we describe the reinforcement learning formulation.

A. Reinforcement Learning Formulation

In general, a reinforcement learning problem asks an agent to learn to take an *action* in an *environment* to move between *states* as per a *transition function* with a goal of reaching a *terminal state* while maximizing the sum of *reward*. In our setup, these are defined as follows:

States: Note that in our problem, we are given an input time-expanded network G_s and a set of observed infections $\mathcal{I}$ and are tasked with finding a tree T^* with a root r which spans $\mathcal{I}$ and maximizes the objective in Problem 1. We choose to grow T^* from the dummy node r. Hence, we initialize our solution tree T with a single node $\{r\}$ as our *start state*, and we design our actions to grow the tree until we cover all the nodes in $\mathcal{I}$, at which point we reach the *terminal state*. Note that we always ensure T is connected. Our universe of states consists of all connected sub-trees of G_s partially covering $\mathcal{I}$.

Action and Transition: At any given point, our learner (or an agent) is in one of the states $T \subset G_s$. An action a takes us from T to $\hat{T}$. As both T and $\hat{T}$ are required to be connected and $\hat{T}$ cannot have cycles, we can only grow T by adding an edge (u, v) with only one end-point in T. Therefore, our universe of actions at a state T consists of adding a single edge among all possible edges with exactly one end-point in T. We employ a deterministic transition function where the addition of an edge (u, v) to a tree T results in a tree $T \cup \{(u, v)\}$ with probability 1.

Reward: Here we derive the reward from the objective function of Problem 1. Our derivation assumes that the function $f(\cdot)$ does not change as the nodes are added to the partial solution (and we also ensure this during training). Recall that our objective is the following:

$$\arg\min_{T} \sum_{(v_a, v_b) \in T} W_s(v_a, v_b) + \alpha \sum_{v_c \in V_s \setminus T} f(F_{v_c}) \quad (2)$$

which is equivalent to

$$\arg\max_{T} -\alpha\Big(\sum_{v_c \in V_s} f(F_{v_c}) - \sum_{v_c \in T} f(F_{v_c})\Big) - \sum_{(v_a, v_b) \in T} W_s(v_a, v_b) \quad (3)$$

Note that for a fixed $f(\cdot)$ (as we ensure within each iteration), $\alpha \sum_{v_c \in V_s} f(F_{v_c})$ is a constant and does not have an effect on the optimization result. Therefore, we have

$$\arg\max_{T} \left[\alpha \sum_{v_c \in T} f(F_{v_c}) - \sum_{(v_a, v_b) \in T} W_s(v_a, v_b)\right] \quad (4)$$

Let $O(T)$ be the value of the objective function in 4 for an arbitrary tree T in the following discussion. Now, imagine we are adding an edge (v_a, v_b) to T'. Recall that our definition of

the action only allows for edges with exactly one endpoint to be added to T'. Let us assume v_a is already in the tree. Now, we define our *reward* $r(T', (v_a, v_b))$ to be the increase in the value of O. Therefore,

$$\begin{aligned} r(T', (v_a, v_b)) &= O(T' \cup \{(v_a, v_b)\}) - O(T') \\ &= \alpha f(F_{v_b}) - W_s(v_a, v_b) \end{aligned} \tag{5}$$

Remark 1: By construction, for any arbitrary $T \subset G_s$ obtained by taking a sequence of actions $a_1, a_2, \cdots, a_n$, the cumulative reward, $\sum_{a_i} r(T_i, a_i)$ is equal to the objective in Equation 4 and therefore the objective in Equation 1.

Finally, we can pose our entire optimization problem as follows:

Problem 2: (BI-LEVEL REINFORCEMENT LEARNING) Given a time-expanded network G_s and observed infections $\mathcal{I} \subset V_s$. Find a sequence of actions A^* (i.e., sequence of edge additions) to obtain to $T^* \subset G_s$ spanning I from the initial state of $\{r\}$ such that:

$$A^* = \arg\max_{A} \left[\sum_{(v_a, v_b) \in A} \alpha f^*(F_{v_b}) - W_s(v_a, v_b) \right]$$
$$\text{such that } f^*(F_{v_c}) = \max_{f} ||f(F_{v_c}) - \mathbf{1}(v_c \in \mathcal{I})|| \tag{6}$$

Algorithm 1 Q-learning for the Reinforce Algorithm

for episode $e = 1, 2, \ldots, L$ **do**
  Initialize the state to T_1 = $((root, None))$
  train GAE model for τ_1 iterations
  train FFN model for τ_2 iterations
  $\mu^0 = GCN(G, features)$
  Update prizes $f()$ on each node
  for step $t = 1, 2, \ldots, T'$ **do**
    $N_t :=$ neighbors of T_t
    With probability ϵ, update

$$(u, v_t) = \left(v_t \in \bar{T}_t \cap N_t, u = argmin_{u \in T_t} w(u, v_t)\right)$$

    Otherwise

$$(u, v_t) = argmax_{v \in \bar{T}_t, u \in T_t} \hat{Q}(T_t, (u, v_t); W)$$

    $T_{t+1} := (T_t, (u, v_t))$
  end for
  if $e \geq n$ and $e\%n == 0$ **then**
    for $T_1, a_1, r_2, ..., T_{T'-1}, a_{T'-1}, r'_T$ **do**
      $W \leftarrow W + (OBJ_t - OBJ_b)\nabla log \pi_W(T_t, a_t)$
    end for
  end if
  return W
end for

B. Overall Approach

Next, we describe the overall training framework for our deep Q-learning approach. Our key idea here is to learn a low-dimensional representation of each node in a latent space shared by both the upper and lower-level problems in Problem 2 and optimize both levels until convergence in an iterative fashion. Towards this end, we first employ a graph auto-encoder to learn meaningful representations of the nodes. We then use the learned embeddings to tackle the lower-level problem using a feed-forward network. We also use the node representations to encode the *state* in our reinforcement learning framework. We then combine the weights given by the lower-level problem and the state representations to search for the policy. Our overall approach is presented in Figure 1. Next, we describe each component of our approach.

Graph Auto-Encoder: First, we learn the node representations of $G_s(V_s, E_s, W_s, F_s)$ by employing a graph auto-encoder, with a graph convolutional network (GCN) encoder and two decoders, each for features and the graph structure. We use two-layered standard GCN [31], with ReLU as the activation function, to learn the representation matrix $\mathbf{X}$. Note that each row $\mathbf{X}[v]$ of $\mathbf{X}$ corresponds to the representation for node v in G_s. We then reconstruct both the features and the graph structure from $\mathbf{X}$. We use the inner product decoder, defined as $\tilde{\mathbf{A}}_s = \sigma(\mathbf{X}\mathbf{X}^\top)$ to reconstruct the adjacency matrix $\mathbf{A}$ of G_s [32]. We use a feed-forward network to produce the feature matrix F_s.

Prize Estimation: Note that Problem 2 asks us to estimate $f^*(F_v)$ for each node v. To do so, we take the node representations $\mathbf{X}$ learned above and define a graph convolutional layer, $\mu^0 = GCN(G_s, \mathbf{X})$. We pass μ^0 through a feed-forward network. The output of the feed-forward network is taken to be the estimated $f^*(F_v)$ prize. We train the prize estimation to predict observed infections $\mathcal{I}$ accurately. We train the module by minimizing the binary cross entropy loss.

Deep Q-learning: As noted earlier, a *state* in our reinforcement learning framework represents a tree T partially covering the observed infections $\mathcal{I}$ and the *actions* available at any given state T is to add an edge (u, v) with exactly one end-point (let's sat u) in T. Next, we define deep Q-state representation for the *state-action* pair $(T, (u, v))$.

The previous modules, which are independently pre-trained, provide excellent starting representations μ^0 of each node in the network. We leverage these first to encode the current *state* T. To do so, we perform a max-pool operator [33] on nodes in T and separately another max-pool operator on nodes not in T. We take the concatenation of the outputs of the two max-pool operators to be the representation μ^T of the current state T. Specifically, we perform the following operation.

$$\mu^T = \left[\sigma(\text{max-pool}_{v \in T}\left\{\mu_v^0\right\}) || \sigma(\text{max-pool}_{v \notin T}\left\{\mu_v^0\right\})\right] \tag{7}$$

where σ denotes the ReLU function. Note that $||$ is the concatenation operator. Next, we use μ^T to represent the *Q-state* $(T, (u, v))$ as follows:

$$\hat{Q}(T, (u, v); W) = W_0^\top \sigma([W_1 \mu^T || W_2 \mu^0[v]]) \tag{8}$$

Here, W_0, W_1, and W_2 are learnable weight matrices. Note that our representation of $\hat{Q}(T, (u, v); W)$ combines the

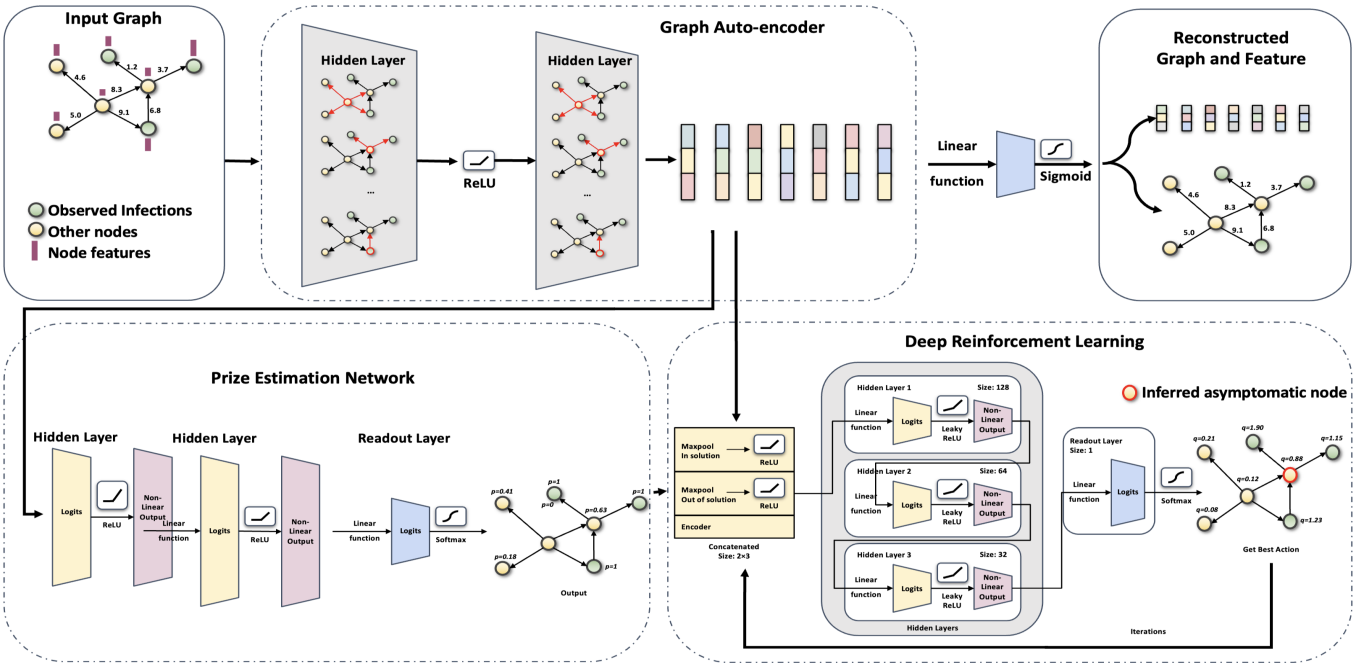

Fig. 1: Overall training framework. The graph auto-encoder learns the low-dimensional representation of the nodes. The learned representation is first used to estimate the node prizes. The prizes and the representations are used in the deep reinforcement learning component to infer asymptomatic nodes.

information that the current state is T and an edge (u, v) being added to T.

The policy search strategy we employ iterates over all edges in the set E' where $E' = \{(u, v)|u \in T \oplus v \in T\}$, where $\oplus$ is the XOR operator. Specifically, our policy π corresponds to adding the edge $(u, v)^* = \arg\max_{(u,v)\in E'} \hat{Q}(T, (u, v); W)$ to T. Each edge addition results in a reward given by Equation 5. Note that we will use the node prizes $f^*(F_v)$ estimated by the prize estimation module to evaluate the policy and the reward.

C. Training

The learnable parameters in the model defined include the GCN encoder and feature decoder in the graph auto-encoder (the adjacency matrix decoder employs inner-product and does not have extra parameters), the feed-forward network used for the prize estimation, and collection of weights used to represent $\hat{Q}(T, (u, v); W)$. Our goal, as stated earlier, is to train the entire model in an end-to-end manner.

Pre-training: We begin by pre-training the graph auto-encoder. Once pre-trained, we use the embeddings generated to pre-train the feed-forward network. Once converged, we train the auto-encoder and the feed-forward network together. This completes our pre-training step.

Q-learning: We use a modified version of the popular REINFORCE algorithm [34] as shown in Algorithm 1 to train and update the parameters of all three modules. For each episode e, we first train the graph auto-encoder for τ_1 number of iterations. We then train the feed-forward network for τ_2 number of iterations. Using the embeddings learned and the prizes estimated we start searching for the optimal tree T representing the outbreak. Note that we keep the estimated prized fixed within each episode e.

Now, we first initialize our solution to be a singleton root $\{r\}$. We then obtain node representations μ^0 for all the nodes using graph auto-encoder. We then grow our tree either by doing a random exploration with some small probability ϵ (i.e., add a random edge with exactly one node in the current solution) or by taking the optimal action by selecting the edge (u, v) with the highest $\hat{Q}$ value as mentioned in section 3.2 with probability $1-\epsilon$. In any case, we grow our tree one node at a time. Note that the newly selected node v should not be part of the current tree T, and to avoid unconnected nodes, v should be a neighbor of a node $u \in T$. We repeat this process until we reach the terminal state where all the nodes in $\mathcal{I}_s$ are covered in the solution tree T. Ma et al. [35] discovered that a large margin easily reduces the total objective of a reinforcement learning problem in the first few improvement steps. Hence, assigning a large reward to those steps is not beneficial for the overall learning process. To avoid this, they proposed to use the average improvement after the first iteration as a reward for each action executed. We leverage this idea for our training as well. We first calculate the objective presented in Problem 2 in the first iteration as OBJ_b and use it as a base objective value. Then, for each subsequent iteration, we measure the average improvement over the base objective as a reward, i.e., for n^{th} action taken in iteration i, the reward is computed as $|\mathrm{OBJ}_b - \mathrm{OBJ}_i|/n$, where OBJ_i is the latest objective value. We use the REINFORCE algorithm [34] to

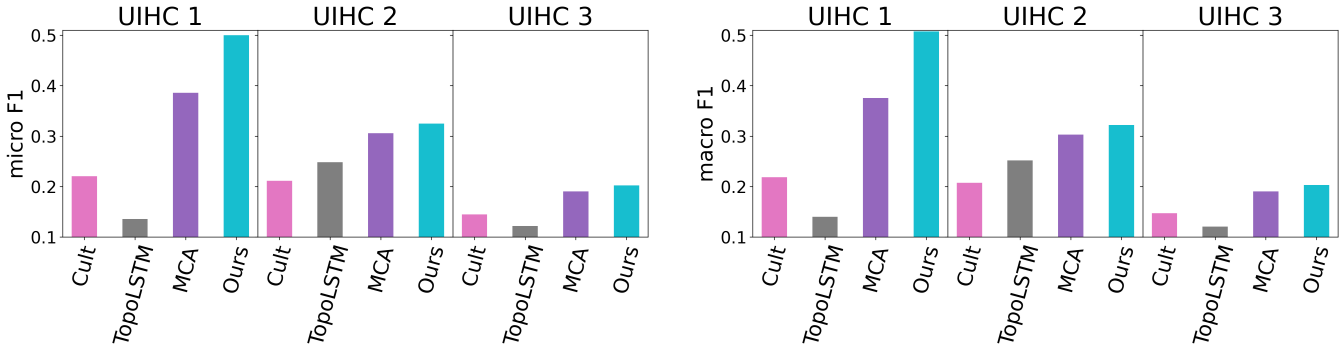

Fig. 2: Performance of our approach and baselines on detecting asymptomatic cases on the simulated outbreak. Our approach outperforms all the baselines in both micro (left) and macro (right) F1 scores.

update the gradient of the policy network, which updates the parameters of graph auto-encoder, feed-forward network, and reinforcement learning models.

Post Processing: After training, we use our algorithm to generate the optimal tree T and prune it by removing all the nodes with no terminals as descendants.

IV. Experiments

We now present an extensive empirical evaluation of our approach to real data with both synthetic and real outbreaks. All of our experiments were conducted on an AMD EPYC 7763 machine with 2TB memory and 8 NVIDIA A30 GPUs each with 24GB memory. We will release our code and data for academic purposes[1].

Data: Our data was extracted from the *University of Iowa Hospitals and Clinics* (UIHC). The data consists of interactions between patients and healthcare workers between 2003 and 2013. It also consists of observed cases of *Clostridiodes Difficile* between 2005 and 2013. *Clostridiodes Difficile* Infection is a well-known Healthcare-associated infection, which spreads in healthcare facilities. From this data, we extracted five sets of graphs with 500 (UIHC1), 2000 (UIHC2), and 5000(UIHC3) nodes respectively. We ensured that the extracted graphs are from the same unit within UIHC and correspond to periods with the highest number of observed infections of *Clostridiodes Difficile*.

Baselines: We use three primary baseline approaches to compare against our approach. Our primary competitor is the minimum-cost arborescence (**MCA**) based approach proposed by Jang et al. in [9]. The proposed approach reduces the prize-collecting Steiner tree problem into a directed Steiner tree problem with prizes as input and solves it heuristically using a minimum-cost arborescence technique. Our second baseline **CuLT** [11] is a state-of-the-art Steiner-tree-based missing infection detection algorithm that assumes the underlying infection spread is based on the popular SI model. A strong drawback of **CuLT** is that it does not take features into account. Finally, our final baseline is **TopoLSTM** [36], which reconstructs cascades using a topological recurrent neural network.

[1]https://github.com/yongjian16/SteinerTree/tree/main

A. Performance on simulated HAI outbreak: asymptomatic cases detection

Our first experiment is designed to evaluate the performance of our approach as compared to the baselines for detecting asymptomatic cases. As we do not have a dataset with ground-truth asymptomatic cases, we resort to simulation to produce outbreaks with known asymptomatic cases using the biased-SIS model proposed in [9]. The model assumes that known risk factors for *Clostridiodes Difficile* such as longer length of stay in inpatient facilities, prescription of high-risk antibiotics and gastric acid suppressors increases, age less than 5 and higher than 60, are correlated with infections. We run the biased SIS model on our graphs and obtain the symptomatic and asymptomatic infections. We feed the graph and symptomatic infections as input to our and the baseline approaches and ask them to find the asymptomatic infections. We measure success by comparing the overlap between the asymptomatic infections produced by the simulation (which are hidden from all the approaches) and the ones inferred by the methods. We repeat this experiment on datasets of all three sizes. Within each size, we repeat the experiment 5 times and report micro and macro-F1 scores. The results are presented in Figure 2.

As seen in the figure, our approach comprehensively outperforms all the baselines in all three datasets on both micro and macro F-1 scores. However, as it follows the non-flexible framework of first inferring node weights (from node features) and then asymptomatic cases using the node weights, it does perform as well as our approach. Our performance gain over **MCA** is up to **29.62%** on micro F-1 scores and **35.18%** on macro F-1 scores. **CuLT** and **TopoLSTM** are not competitive as they do not take individual risks into account when inferring asymptomatic infections. We note that the performance of almost all methods deteriorates as the graphs get larger. This is to be expected as the problem gets more challenging as the graph size increases especially since outbreak size

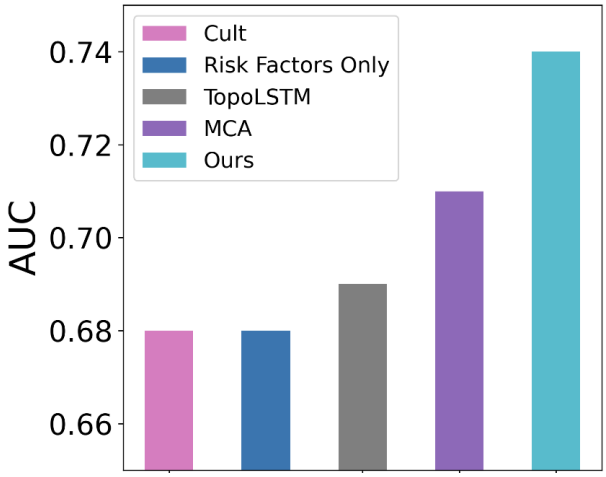

Fig. 3: AUC for observed infection prediction (best viewed in color). Our approach, shown in the rightmost bar, outperforms all the baselines significantly.

remains fairly consistent. Overall, our experiments show that our algorithm can recover asymptomatic cases more accurately than the baselines consistently.

As mentioned earlier, our approach is designed to detect nodes that have both *i)* a high probability of **being exposed** to the pathogen and *ii)* a high likelihood of **remaining asymptomatic**. **CULT** only looks at the first aspect as it ignores individual risks and **TopoLSTM** is a cascade prediction model that does not distinguish between symptomatic and asymptomatic cases. While the **MCA** prioritizes minimizing cost by selecting a spanning tree, it operates in two stages: first assessing individual asymptomatic risks and then optimizing the tree. This can lead to suboptimal outcomes due to potential inaccuracies in the first stage. Our method, in contrast, directly tackles both challenges simultaneously through an end-to-end optimization process. This enables us to avoid the limitations of **MCA** and achieve greater accuracy and efficiency in identifying asymptomatic cases.

B. Performance on real HAI outbreaks: observed infection prediction

Here we demonstrate the application of our approach in predicting future infections. As asymptomatic infections are still infectious, the nodes that interact with the identified asymptomatic cases are at a higher risk of developing infection than others. Here, we start by sampling one month of interactions from UIHC to construct $\mathcal{G}$. We chose the time span with 68 observed cases. Out of the 68 observed cases, we chose 80% of these to be in the 'training set' and were given our approach and the baselines as the observed infections. After each method infers the asymptomatic infections, we compute the so-called asymptomatic pressures [9] (i.e., a normalized metric computing interaction frequency with asymptomatic cases) for all the nodes. Note that the asymptomatic pressure metric is useful only if the inferred asymptomatic cases are accurate. We then trained a multi-layer perceptron with standard risk factors of CDI and asymptomatic pressures to predict observed cases. The multi-layered perceptron was also trained on the same 80% of the observed cases in the training data. We then compute AUC to evaluate the accuracy of the predicted observed infections.

The result is presented in Figure 3. Here, we add an additional baseline **Risk-factors-only**. The baseline trains the MLP without taking any asymptomatic pressure into account. As seen from the figure, TopoLSTM, MCA, and our approach lead to a significant gain over the Risk-factors-only baseline. This demonstrates that identifying asymptomatic cases helps with infection prediction in general. We observe that CuLT is unable to improve upon the Risk-factors-only baseline in this graph. We believe this is due to the fact that CuLT assumes underlying disease process in SI-like, however, healthcare-associated infections such as *Clostridiodes Difficile* do not lend themselves to a simple SI model. As in the previous experiment, we can observe that our approach clearly beats all the baselines.

C. Case Study on CDI outbreak

To demonstrate that the asymptomatic cases predicted by our method are clinically meaningful, we perform case studies on the same one month interaction network collected from UIHC as in Subsection IV-B. Here, we took all the 68 observed infections from the previous setup and leveraged our approach to identify asymptomatic infection. We obtained the solution tree and visualized a pruned version of it in the Figure 4.

Here, we highlight two cases, node A and node B, determined to be asymptomatic by our approach (in red). The patients represented by the nodes turned out to be *high-risk* newborns who stayed in the intensive care unit (ICU) for 10 and 12 days respectively and then transferred to the neonatal intensive care unit (NICU) and stayed there for another 3 and 11 days respectively. Admissions to intensive care unit and small age are both known risk factors of CDI [37], [38], [39]. Moreover, the patients with symptomatic CDI cases (namely node $A2$ and node $B2$) were also in the same unit during that period. Hence, it is very likely that the newborn patients contracted *Clostridiodes Difficile* pathogens while in the NICU and transmitted the infections to observed cases. This experiment highlights the importance of the proposed work as accurate identification of the red nodes as asymptomatic would have enabled healthcare providers to deploy intervention (such as isolation), which would have prevented downstream cases.

D. Performance on simulated COVID-19 outbreaks

In the next experiment, we leverage our approach to detect asymptomatic COVID-19 infections in synthetic outbreaks. Here, we first run the CovaSim model to simulate COVID-19 outbreaks in UIHC graphs [40]. CovaSim is an open-source agent-based model where infected agents can also be asymptomatic. As in our earlier setup, we run the CovaSim model on our graphs and generate symptomatic and asymptomatic infections. The graph and the symptomatic infections are

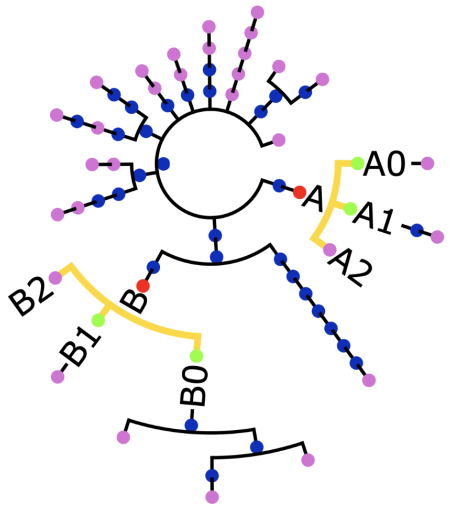

Fig. 4: Case study on UIHC graph. The detected asymptomatic cases (nodes A and B visualized in Red) were later determined to be high-risk nodes and were present in the same unit as symptomatic nodes.

revealed in all the methods. We measure success by contrasting the inferred asymptomatic cases with the simulated 'ground-truth' asymptomatic infections. We conducted this experiment on UIHC1 and UIHC2 datasets. Our results are summarized in the Table I.

TABLE I: Comparison of performance of our approach against state-of-the-art baselines in detecting COVID-19 asymptomatic cases. Our approach significantly outperforms the baselines

Method	UIHC 1		UIHC 2	
	micro-F1	macro-F1	micro-F1	macro-F1
CuLT	0.37	0.43	0.21	0.26
TopoLSTM	0.34	0.38	0.12	0.08
MCA	0.49	0.44	0.38	0.32
Ours	**0.68**	**0.63**	**0.48**	**0.41**

As observed in the table, our approach significantly outperforms all the baselines in both datasets. As in the previous experiment, **MCA**'s performance is closest to ours followed by **CuLT** and **TopoLSTM**. In UIHC 1, the best-performing baseline, **MCA**, was able to achieve a micro-F1 score of 0.49 and a macro-F1 score of 0.44. Our approach's micro-F1 and macro-F1 scores were 0.68 and 0.63 (an impressive approximate 40% improvement). Our approach outperformed the best baseline, **MCA**, by 25% in terms of micro-F1 score and by 28% in terms of macro-F1 score in the UIHC 2 dataset.

Interestingly, all the methods performed better in detecting asymptomatic COVID-19 cases than in detecting asymptomatic CDI cases. This behavior could be attributed to the fact that there were more observed COVID-19 cases than CDI cases, making the task easier.

E. Ablation-study

We conducted additional experiments on the simulated CDI outbreak in the UIHC-1 dataset to demonstrate how the performance changes when we change our end-to-end framework. Here, we ran two modifications of our approach. In the first modified approach, **Mod-1**, we use the GCN module for prize estimation and use the **MCA** approach for solving the Prize-Collecting Steiner-tree problem. Similarly, in the second modified approach, **Mod-2**, we use GCN for prize estimation and standard non-deep approximate Q-learning approach for the Steiner tree. The performance of these modified approaches with our original approach is presented in Table II.

TABLE II: Performance of modified approach.

Method	micro-f1	macro-f1
Ours	0.50	0.51
Mod-1	0.42	0.39
Mod-2	0.38	0.32

We see a consistent performance drop in both scores with the modified approaches. **Mod-1** results in 16% and 23% drop in micro and macro F1 scores respectively. Similarly, **Mod-2** results in 24% and 35% drop in the same metrics. The results show that if we simply replace our deep Q networks with combinatorial algorithms or standard Q-learning algorithms, the results deteriorate.

We would like to point out that a key advantage of the proposed method over these baselines, and those discussed earlier, is that our approach allows for end-to-end inference of both the prizes (susceptibility risk) and the cascade (infection transmission), contributing significantly to its performance superiority.

F. The effect of α and γ on the performnace.

There are two hyper-parameters in our proposed approach, namely α and γ. Recall that the parameter α quantified the importance of the individual risk as compared to the transmission risk, and parameter γ is the edge weight connecting the dummy root node with all the nodes in $\mathcal{G}$ (see Section 3). Note that larger values of γ prefer fewer branches out of the root node, hence resulting in a smaller number of trees in $\mathcal{G}$. In our experiments, we tried to nullify the effect of α by normalizing both the edge weights and the predicted prizes (node weights). Here, we show the effect of varying γ and α on a small UIHC 1 graph with 500 nodes. We vary α from 0 to 4 while keeping γ fixed. We repeat this for γ values of 0, 16, and 128. The result is presented in Figure 5.

As observed in the figure, the F1-score peaks at $\alpha = 1$ for γ of 16 and 128. Similarly, it has a clear elbow at $\alpha = 1$ for $\gamma = 0$. Similarly, in the next set of experiments, we varied γ for different values of α and noticed that the F1-score usually increases with γ. This is probably due to the fact that there were very few observed cases in the graph, and most of them could be easily connected with each other; hence the larger value of γ, which prefers fewer branches out of the dummy root, performs the best.

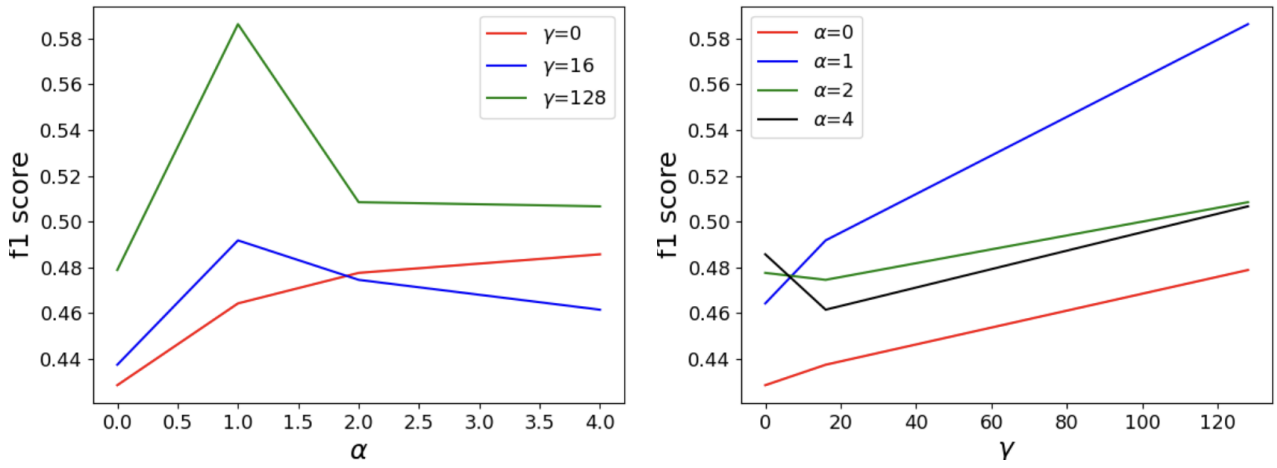

Fig. 5: Performance of our approach with different α and γ

V. Discussion

Here we dive deeper into the aforementioned challenges and justify the superiority of our method. The three challenges in detecting asymptomatic cases that we identified were (C1) **Data Scarcity**, (C2) **Poor generalization from symptomatic cases**, and (C3) **Bias towards symptomatic cases**. The first challenge arises due to difficulty in collecting data associated with asymptomatic infections which are usually only obtainable by conducting extensive blanket serological testing. We address this challenge by leveraging symptomatic case data in a careful manner. Note that the likelihood of a node being asymptomatic hinges on two key factors: 1) the probability of a node being exposed to the virus via symptomatic or other asymptomatic cases and 2) the likelihood of a node remaining asymptomatic upon exposure. Our approach incorporates both of these aspects. Most previous approaches focus on one of these factors exclusively. When only considering the first factor, it gives rise to the second challenge (i.e. poor generalization) as we only observe symptomatic cases. On the other hand, addressing the second factor solely from symptomatic data leads to the third challenge (bias towards symptomatic). In our novel approach, we introduce an end-to-end risk-aware framework that addresses the first factor by constructing infection paths represented as a prize-collecting Steiner tree. We handle the second factor by incorporating risk-aware incentives into the Steiner tree problem. Furthermore, our end-to-end framework allows for the joint optimization of both aspects, leading to enhanced generalization capabilities compared to the traditional two-stage optimization approaches.

VI. Conclusion

In this paper, we formalized the asymptomatic case detection problem as a prize-collecting Steiner-tree problem with latent prizes. We then reduced this problem to a bi-level reinforcement learning problem. Our reduction is exact, meaning solving the reinforcement learning problem will also solve the prize-collecting Steiner tree problem. We then solved the bi-level reinforcement learning problem using a deep Q-learning approach. We conducted extensive experiments on graphs of different sizes extracted from the operations data from the University of Iowa Hospitals and Clinics. Our results on synthetic outbreaks show that the proposed approach is able to identify asymptomatic infection with high accuracy while the baseline approaches are less accurate. We see a similar trend in our second experiment where we leveraged inferred asymptomatic infection for observed infection prediction. Our result improves significantly over the baseline which only takes individual risk factors into account. Other baselines are less successful in doing so. Finally, our case study reveals that the inferred asymptomatic cases are meaningful and are, in fact, high-risk patients.

VII. Acknowledgement

This work was partially supported by the CDC MInD Healthcare group under cooperative agreement U01-CK000594 and its associated COVID-19 supplemental funding and a startup fund from the University of Iowa. The authors acknowledge feedback from other University of Iowa CompEpi group members.

References

[1] Y. Bai, L. Yao, T. Wei, F. Tian, D.-Y. Jin, L. Chen, and M. Wang, "Presumed asymptomatic carrier transmission of covid-19," *Jama*, vol. 323, no. 14, pp. 1406–1407, 2020.

[2] J. B. Aguilar, J. S. Faust, L. M. Westafer, and J. B. Gutierrez, "Investigating the impact of asymptomatic carriers on covid-19 transmission," 2020.

[3] J. Shaman, "An estimation of undetected covid cases in france," 2021.

[4] W. Shang, L. Kang, G. Cao, Y. Wang, P. Gao, J. Liu, and M. Liu, "Percentage of asymptomatic infections among sars-cov-2 omicron variant-positive individuals: a systematic review and meta-analysis," *Vaccines*, vol. 10, no. 7, p. 1049, 2022.

[5] C.-C. Lai, J.-H. Wang, W.-C. Ko, M.-Y. Yen, M.-C. Lu, C.-M. Lee, P.-R. Hsueh *et al.*, "Covid-19 in long-term care facilities: an upcoming threat that cannot be ignored," *Journal of microbiology, immunology, and infection*, vol. 53, no. 3, p. 444, 2020.

[6] C. Suetens, K. Latour, T. Kärki, E. Ricchizzi, P. Kinross, M. L. Moro, B. Jans, S. Hopkins, S. Hansen, O. Lyytikäinen *et al.*, "Prevalence of healthcare-associated infections, estimated incidence and composite antimicrobial resistance index in acute care hospitals and long-term care facilities: results from two european point prevalence surveys, 2016 to 2017," *Eurosurveillance*, vol. 23, no. 46, p. 1800516, 2018.

[7] J.-T. Wei, Z.-D. Liu, Z.-W. Fan, L. Zhao, and W.-C. Cao, "Epidemiology of and risk factors for covid-19 infection among health care workers: a multi-centre comparative study," *International Journal of Environmental Research and Public Health*, vol. 17, no. 19, p. 7149, 2020.

[8] B. Snider, B. Patel, and E. McBean, "Asymptomatic cases, the hidden challenge in predicting covid-19 caseload increases," *Infectious Disease Reports*, vol. 13, no. 2, pp. 340–347, 2021.

[9] H. Jang, S. Pai, B. Adhikari, and S. V. Pemmaraju, "Risk-aware temporal cascade reconstruction to detect asymptomatic cases: For the cdc mind healthcare network," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 240–249.

[10] R. Subramanian, Q. He, and M. Pascual, "Quantifying asymptomatic infection and transmission of covid-19 in new york city using observed cases, serology, and testing capacity," *Proceedings of the National Academy of Sciences*, vol. 118, no. 9, p. e2019716118, 2021.

[11] P. Rozenshtein, A. Gionis, B. A. Prakash, and J. Vreeken, "Reconstructing an epidemic over time," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1835–1844.

[12] H. Jang, P. M. Polgreen, A. M. Segre, D. K. Sewell, and S. V. Pemmaraju, "A data-driven approach to identifying asymptomatic c. diff cases," 2020.

[13] M. Makar, J. Guttag, and J. Wiens, "Learning the probability of activation in the presence of latent spreaders," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[14] Z. Xing, B. Nicholson, M. Jimenez, T. Veldman, L. Hudson, J. Lucas, D. Dunson, A. K. Zaas, C. W. Woods, G. S. Ginsburg *et al.*, "Bayesian modeling of temporal properties of infectious disease in a college student population," *Journal of Applied Statistics*, vol. 41, no. 6, pp. 1358–1382, 2014.

[15] H. Xiao, P. Rozenshtein, N. Tatti, and A. Gionis, "Reconstructing a cascade from temporal observations," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 666–674.

[16] H. Xiao, C. Aslay, and A. Gionis, "Robust cascade reconstruction by steiner tree sampling," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 637–646.

[17] R. Mishra, J. Heavey, G. Kaur, A. Adiga, and A. Vullikanti, "Reconstructing an epidemic outbreak using steiner connectivity," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 10, 2023, pp. 11 613–11 620.

[18] M. Farajtabar, M. G. Rodriguez, M. Zamani, N. Du, H. Zha, and L. Song, "Back to the past: Source identification in diffusion networks from partially observed cascades," in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 232–240.

[19] S. Sundareisan, J. Vreeken, and B. A. Prakash, "Hidden hazards: Finding missing nodes in large graph epidemics," in *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015, pp. 415–423.

[20] H. Jang, A. Fu, J. Cui, M. Kamruzzaman, B. A. Prakash, A. Vullikanti, B. Adhikari, and S. V. Pemmaraju, "Detecting sources of healthcare associated infections," 2023.

[21] K. Fan, C. Li, and K. Heller, "A unifying variational inference framework for hierarchical graph-coupled hmm with an application to influenza infection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[22] B. Adhikari, Y. Zhang, S. E. Amiri, A. Bharadwaj, and B. A. Prakash, "Propagation-based temporal network summarization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 4, pp. 729–742, 2017.

[23] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge university press, 2010.

[24] E. Halperin and R. Krauthgamer, "Polylogarithmic inapproximability," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003, pp. 585–594.

[25] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[26] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[27] T. Barrett, W. Clements, J. Foerster, and A. Lvovsky, "Exploratory combinatorial optimization with reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3243–3250.

[28] Q. Cappart, E. Goutierre, D. Bergman, and L.-M. Rousseau, "Improving optimization bounds using machine learning: Decision diagrams meet deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1443–1451.

[29] S. E. Amiri, B. Adhikari, A. Bharadwaj, and B. A. Prakash, "Netgist: Learning to generate task-based network summaries," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 857–862.

[30] T. Zhang, A. Banitalebi-Dehkordi, and Y. Zhang, "Deep reinforcement learning for exact combinatorial optimization: Learning to branch," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 3105–3111.

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[32] ——, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[33] N. Goyal and D. Steiner, "Graph neural networks for image classification and reinforcement learning using graph representations," *arXiv preprint arXiv:2203.03457*, 2022.

[34] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Reinforcement learning*, pp. 5–32, 1992.

[35] Y. Ma, X. Hao, J. Hao, J. Lu, X. Liu, T. Xialiang, M. Yuan, Z. Li, J. Tang, and Z. Meng, "A hierarchical reinforcement learning based optimization framework for large-scale dynamic pickup and delivery problems," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 609–23 620, 2021.

[36] J. Wang, V. W. Zheng, Z. Liu, and K. C.-C. Chang, "Topological recurrent neural network for diffusion prediction," in *2017 IEEE international conference on data mining (ICDM)*. IEEE, 2017, pp. 475–484.

[37] H. H. Kim, Y. S. Kim, D. S. Han, Y.-H. Kim, W. H. Kim, J. S. Kim, H. Kim, H.-S. Kim, Y.-S. Park, H. J. Song *et al.*, "Clinical differences in clostridium difficile infection based on age: A multicenter study," *Scandinavian journal of infectious diseases*, vol. 46, no. 1, pp. 46–51, 2014.

[38] T. J. Sandora, M. Fung, K. Flaherty, L. Helsing, P. Scanlon, G. Potter-Bynoe, C. A. Gidengil, and G. M. Lee, "Epidemiology and risk factors for clostridium difficile infection in children," *The Pediatric infectious disease journal*, vol. 30, no. 7, pp. 580–584, 2011.

[39] Y. Mu, M. Dudeck, K. Jones, Q. Li, M. Soe, A. Nkwata, and J. Edwards, "Trends in hospital onset clostridioides difficile infection incidence, national healthcare safety network, 2010–2018," *Infection Control & Hospital Epidemiology*, vol. 41, no. S1, pp. s53–s54, 2020.

[40] C. C. Kerr, R. M. Stuart, D. Mistry, R. G. Abeysuriya, K. Rosenfeld, G. R. Hart, R. C. Núñez, J. A. Cohen, P. Selvaraj, B. Hagedorn *et al.*, "Covasim: an agent-based model of covid-19 dynamics and interventions," *PLOS Computational Biology*, vol. 17, no. 7, p. e1009149, 2021.