

# From A Timeline Contact Graph to Close Contact Tracing and Infection Diffusion Intervention

Anonymous Author(s)

## ABSTRACT

In this work, we aim to achieve two goals: (1) how to accurately and efficiently model the virus diffusion according to fine-grained users' movement records, and (2) how to help the government evaluate the effectiveness of different checkpoint deployment strategies to mitigate virus diffusion. For concreteness, we use the public transport system as a scenario for illustration. We associate the virus with two parameters, *incubation period* and *transmissibility*, which simulate how fast the virus could be spread and hence determine how dangerous it is. To accurately model virus diffusion, we propose a diffusion model based on a new data structure namely *timeline contact graph* (TCG), which essentially is a weighted directed graph  $G = (V, E)$ . Here,  $v \in V$  is a trip record of a passenger, and  $e \in E$  from nodes  $v$  to  $v'$  indicates the virus infection from  $v$  to  $v'$ , and the weight of  $e$  is the infection probability. TCG is able to record all close contacts when there are boarding/alighting passengers. In addition, we formulate two problems, Epidemic Mitigating in Public Area problem (EMA) and Epidemic Maximized Spread in Public Area problem (ESA). EMA is to find an ideal checkpoint deployment strategy; ESA is to find a set of 'super-spreaders' to simulate an extreme case that the virus is spread widely, which then help to test the robustness of checkpoint deployment strategies. Finally, we conduct experiments using real-world station datasets and millions of public transport trip records to verify the usefulness and scalability of our approach.

## PVLDB Reference Format:

Anonymous Author(s). From A Timeline Contact Graph to Close Contact Tracing and Infection Diffusion Intervention. PVLDB, 14(1): XXX-XXX, 2022.  
doi:XX.XX/XXX.XX

## PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at <https://github.com/rmitbggroup/VirusPrediction>.

## 1 INTRODUCTION

Plagues and epidemics have ravaged humanity since the birth of civilization, often changing the course of history. In the last two years, the catastrophic COVID-19 forced researchers to devote their effort to studying the infection diffusion, which considerably assists governments in intervening in infection diffusion. For example, a hot topic is the epidemic trend prediction [4, 10, 12, 15, 25]. Based on the statistical mobility of each census and initial infectious proportion, it aims to predict the future infectious proportion of the whole

population in a long trend. These studies are mainly employed to evaluate the effect of mobility restriction strategies, such as lockdown, closing recreational venues, or banning public gatherings. Although these policies have effectively curbed the epidemic, people demand everyday life as we are approaching the post-epidemic stage. Therefore, decision-makers chase precise interventions rather than a harsh city-wide mobility restriction strategy.

In this work, as a complement methodology of the long-term trend prediction, we propose an individual-level diffusion simulator to evaluate the infection status of each person based on short-term city-wide precise individual movement data. For instance, given ten infected persons who have symptoms for a few days, who are in danger of being infected due to direct/indirect contact? The most significant difference is that our simulator captures all close contacts between each person rather than roughly predicting infectious proportion based on statistical data. To illustrate how our work helps against the epidemic, we propose two essential downstream tasks based on WHO's guide [32].

**Task 1. Tracing Close Contacts in Short-term.** Tracing close contacts is to find all potentially infected people [32]. Given a set of infected people, it includes people who directly contact them and others who are potentially infected through people who have a risk of infection. The range of infection is expanding with a cascade diffusion along with exponential growth of the number of potential infectious. Therefore, we only focus on tracing all close contacts in the short-term (e.g., within one month); a long-term tracing is meaningless because everyone will be infected in a long-term cascade diffusion. This differentiates our work from a long list of existing work [3–6, 10, 15, 16, 35, 38] on virus infection within a much longer term (e.g., a year or even longer) at population level.

**Task 2. Deploying Checkpoints in Short-term.** Deploying checkpoints to detect and quarantine infected persons as a practical measure has been suggested by WHO [32], and has been used extensively worldwide. Hence, our second task is to find top- $k$  locations for building checkpoints to detect infected persons and thus, to minimize the risk of an outbreak within a short-term (e.g., one month). We evaluate the effectiveness of deployment strategies from two perspectives (defense vs. offense). From the defensive perspective, we formulate the problem of Epidemic Mitigating in Public Area, EMA, to find an ideal checkpoint deployment strategy such that infection diffusion can be well mitigated. From the offensive perspective, we introduce the Epidemic Maximized Spread in Public Area problem, ESA, to find  $k$  highly infectious persons and utilize them to spread the virus as an attack to the given checkpoint deployment and further evaluate the robustness of the given deployment.

There are two fundamental challenges to achieve a precise close contact tracing based on city-wide individual movements.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing [info@vldb.org](mailto:info@vldb.org). Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.  
doi:XX.XX/XXX.XX

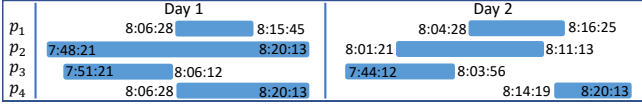


Figure 1: Timetable of check-in records

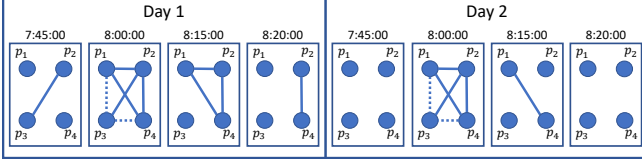


Figure 2: Snapshot updates every 15 minutes

**Challenge 1 - High Memory Consumption.** Most current work is the snapshot-based solution [27, 44, 47]. When the movement data is at a city-wide individual level, the close contact relationship is changing every second, which requires the snapshot to be created with high frequency and hence leads to an unaffordable memory cost. On the other hand, a low frequency cannot capture a correct close contact, as shown in the following example.

**EXAMPLE 1.** As shown in Figure 1, each person  $p_1$  has one check-in and one check-out at the same location. If we try to capture the close contacts between persons accurately, we will need 14 snapshots, where each snapshot is created whenever there is a check-in/out. A counterexample is shown in Figure 2 with inaccurate close contacts, where we create a snapshot every 15 minutes to form a dynamic graph. In each snapshot, we connect two persons if they stay in the same period. Consequently, the snapshot incorrectly captures the close contacts of  $p_3$ ,  $p_1$ , and  $p_3$ ,  $p_4$  (dot-lines) on both two days, as  $p_3$  is checking out before  $p_1$  and  $p_4$  are checking in.

**Challenge 2 - Infection Probability Measurement.** There are two essential factors determining the measurement: (1) An exposure-duration-based spread model. It has been shown that the infection probability highly depends on the exposure duration [30–32]. (2) Infection characteristic. The different epidemics have unique infection diffusion based on their characteristics, such as incubation and infectious probability. The existing snapshot-based model takes graph structure as input and hence is unadjustable. It requires researchers to update the graph and input it again. We will further discuss their limitations in Section 2.

#### Our Contributions:

**Graph Structure.** In order to trace close contacts that are frequently changing ever, we propose a novel graph structure, namely Timeline Contact Graph (TCG). It is able to transform the enormous discrete graph structures (e.g., snapshots) into a single graph, to avoid creating enormous snapshots (Challenge 1), and hence overcome the dilemma of losing the accuracy of changing edges among vertexes or suffering the explosive amount of data.

**Diffusion Model.** Based on TCG, we further introduce an individual infection diffusion model to measure the infection probability based on virus characteristics (Challenge 2). In this model, a virus is associated with two parameters, the *incubation period* and the

*transmissibility* [7, 21]. The former refers to the duration between when a person is infected and when she can infect others; the latter determines the likelihood of a healthy person getting infected by an infected person according to the exposure duration. It is worth noting that picking the right parameter values is not the focus of this paper, and our solution can work with any choice of them.

**Diffusion Intervention Simulator.** We propose a Simulator based on the TCG and the infection diffusion model. It finds the ideal check-point deployment strategy for the EMA problem, and evaluates the effectiveness of strategies through the ‘super-spreaders’ for the ESA problem. We further introduce the concept of *Contact Paths Tracing* (CPT), which allows us to efficiently find these high-risk movements that may infect a large number of persons.

**Simulator & Experimental.** We carefully design experiments based on large-scale real-world movement data to evaluate all baselines in a real scenario. The experimental result shows that (1) our TCG is at most 20 times more accurate than snapshot-based solutions; (2) our solution outperforms the best baselines about three times by average; (3) Our solution is scalable to handle four weeks of movement data with tens of millions of trips.

## 2 RELATED WORK

In this section, we introduce the most related work falling into two categories: *virus immunization* and *influence maximization*. We will mainly introduce virus immunization work since it is closer to our work, which aims to study the prevention and diffusion of the virus.

### 2.1 Virus Immunization

Our work is fundamentally different from most Virus Immunization (VI) studies from the perspectives of the virus model, the objective, and the methodology, respectively, as illustrated below.

**Objective:** Most studies [4, 10, 12, 15, 25?] focus on the question of “will the pandemic happen in the long-term at *census level*?” The difference is three-fold: time period, analytical granularity, and the goal. For instance, given a graph  $G$ , the *goal* of the work [12] is to find a threshold of the first eigenvalue  $\lambda^*$ , such that the virus will diminish when the time approaches infinity (*time period*) if  $\lambda$  of  $G$  is smaller than  $\lambda^*$ , otherwise it will become a pandemic. Each node in the graph represents a city (*analytical granularity*).

In contrast, our *goal* is to accurately track how the virus spreads through each person (*analytical granularity*) based on the close contacts within one month (*time period*). In other words, we answer the question of “Who will be infected?”

**Virus Model:** The three most common states in most epidemic models are Susceptible ( $S$ ), Infected ( $I$ ), and Recovered ( $R$ ) [1]. People in the  $S$  state are healthy but susceptible to being infected; in the  $I$  state are infected; in the  $R$  state are recovered but may be infected again. With different combinations, various models, such as SIS, SIRS, SEIR, and SEIRS, are defined in order to better accommodate the biological properties of real diseases [28, 33]. For instance, the SEIR model is a variation of the SIR model and it includes a stage of exposed ( $E$ ) individuals, referring to persons who have been infected but cannot transmit the disease yet. Since most VI work studies the long-term trend of the virus spreading, they consider the  $R$  state and re-infection phase. Conversely, our work targets a

period of a few weeks, where a person is unlikely to be healed and infected repeatedly. Therefore, we adopt the SI model.

**Methodology:** Existing VI studies fall into two categories, based on either the betweenness centrality [38] or the eigenvalues [5, 35]. In a nutshell, the core idea behind these two classes of solutions is to remove the node/edge that has the highest centrality or can maximally decrease the first eigenvalue of the graph. However, those existing solutions suffer from two common drawbacks. First, it has been shown that the time-varying connectivity pattern of networks will affect the epidemic process in numerous ways [28, 33], while most of the above studies utilize a fixed graph. For a few studies based on a varying graph or temporal graph, a common approach for VI problems is to utilize the snapshots corresponding to different time intervals [27, 44, 47], which causes the accuracy and efficiency issues, to be detailed in Section 4.1. To name a few, in the work [35], given  $N$  snapshots, it first evaluates the drop of the eigenvalue of removing each node for each snapshot and then, at each step, it removes the node that causes the largest drop of the average eigenvalue among all snapshot-based solutions. Second, to the best of our knowledge, none of the existing works fully utilizes spatial-temporal movement records to model and prevent the virus diffusion. As mentioned in Section 1, the spatial and temporal properties of contacts based on the movement records are critical to model the infection diffusion, as the two essential components of virus infection are *close contacts* that are spatial-location dependent, and *exposure duration* that is temporal dependent [32].

## 2.2 Influence Maximization

The Influence Maximization (IM) problem aims to select a  $k$ -sized seed set in an online social network to maximize the expected number of influenced people from the information diffusion of the seed set, which is similar to the ESA problem. There are some existing studies that take the evolving graph into account. The Time-Constrained IM problem [22] introduces a Latency-Aware Independent Cascade model to compute the influence between two persons, where the influence equals the product of influence probability and the delay probability. In the Continuously-activated and Time-restricted Independent Cascade (IC) model [18], each active node can activate its neighbors repeatedly until a given deadline. Given that a social network is constantly evolving and important users with the most influence also change over time, the authors in [17] try to identify a seed set that can influence the largest number of distinct users over a predefined window of time. To the best of our knowledge, no existing work uses spatial-temporal movement to model the infection diffusion, whereas in our work, the infection probability is based on physical contacts and exposure duration among people. Therefore, the IC model is insufficient as a predictive model for tracing virus spread.

## 3 PRELIMINARIES

In this work, we have two tasks: (1) how to trace close contacts according to fine-grained user movement, and (2) how to help the government evaluate the effectiveness of different checkpoint deployment strategies to mitigate infection diffusion. For concreteness, we study this work based on the real dataset of Singapore

public transport system. It is worth noting that our work is not limited to the public transport dataset but is applicable to any dataset recording the individual movements containing the individual ID, location and check-in/out times, such as general individual check-in data or close contact data. The infection diffusion simulation task that utilizes the cause crowd movement trend instead of fine individual movement is orthogonal to our work.

In the following, we first present the important notations. We then present how to measure the infectious probability between passengers (for task 1); and formulate the infection number measurement (for task 2). Table 1 lists notations that will be used frequently in the paper.

**Trip.** Each trip  $tr_i$  is a tuple  $\{b, s_b, s_e, t_b, t_e\}$ , where  $b$  denotes the riding bus,  $s_b$  and  $s_e$  denote the boarding station and alighting station respectively, and  $t_b$  and  $t_e$  are the boarding time and alighting time respectively.

**Passenger.** Each passenger  $p_n$  has a set of trips  $p_n.Tr = \{tr_i, \dots, tr_j\}$ . Let  $p_n^i$  denote the trip  $tr_i$  of  $p_n$ . A passenger is in either a healthy state ( $p_n = 0$ ) or an infected state ( $p_n = 1$ ).

**Infection probability.** The infection probability refers to the risk of a healthy passenger getting infected by an infected passenger during the close contact. Existing studies [7, 21] have shown that a longer exposure duration leads to a higher infection probability. We follow this to define  $Pr(p_n^i, p_m^j) = f(R(p_n^i, p_m^j))$  as the infection probability between two passengers after one contact (i.e., contact between  $i$ -th trip of  $p_n$  and  $j$ -th trip of  $p_m$ ). Here,  $R()$  is the exposure duration that two passengers stay at the same bus  $b$  location.  $f()$  measures the infection probability based on the exposure duration. In this work, we build a continuous model and a discrete model. We will discuss the parameters' details in Section 7.1.

Subsequently, the probability of a passenger being infected after a trip is  $Pr(p_n^i) = 1 - \prod_{tr_j \in (\mathcal{T} \setminus p_n.Tr)} (1 - Pr(p_n^i, p_m^j))$ . Intuitively, it equals one minus the probability of avoiding being infected from all close contacts after the trip  $p_n^i$ . Consequently, the probability that a passenger  $p_n$  will be infected (by considering all trips that  $p_n$  has made) is defined as  $Pr(p_n) = 1 - \prod_{tr_i \in p_n.Tr} (1 - Pr(p_n^i))$ .

**EXAMPLE 2.** We use the trip data shown in Figure 1 to present how to measure the infected probability. For concreteness, we quantify the infection probability based on the  $EC_{50}$  model [7, 21, 37], where  $f(R) = 1/(1 + (EC_{50}/R)^\tau)$ .  $EC_{50}$  and  $\tau$  are two parameters defined in the  $EC_{50}$  model. We set  $EC_{50} = 800$  seconds and  $\tau = 5$ , which means that a healthy person will have a 50% chance to be infected if she has a close contact with an infected person for 800 seconds. Under this setting,  $Pr(p_n^i, p_m^j) = 50\%$  when  $R(p_n^i, p_m^j) = 800$  seconds. According to the trip data, where  $p_1.Tr = \{tr_1, tr_2\}$ ,  $p_2.Tr = \{tr_1, tr_2\}$  and  $p_3.Tr = \{tr_1, tr_2\}$ , we have  $Pr(p_2^1, p_1^1) = 1/(1 + (800/557)^5) = 14\%$ , and  $Pr(p_2^1, p_3^1) = 1\%$ . Subsequently,  $Pr(p_2^1) = 1 - (1 - 14\%)(1 - 1\%) = 14.86\%$ . Similarly, we have  $Pr(p_2^2) = 64.79\%$ . Consequently, after day 2, the probability of  $p_2$  getting infected is  $Pr(p_2) = 70.02\%$ .

It is worth noting that our work is orthogonal to the choice of  $f()$  as long as it is based on the exposure duration.  $EC_{50}$  model [7, 21, 37] is one option that is widely used in measuring the concentration required for the drug, antibody, or toxicant. Our solution sd applied to other measurements. For instance, instead of the probabilistic

measurement like the  $EC_{50}$  model, governments also adopt a deterministic rule to classify the possible infection. One example is the close contact rule made by WHO for COVID-19, in which the threshold for COVID-19 is 15 minutes [32]. It means that a healthy person will be counted as a highly possible infected (i.e.,  $f = 100\%$ ) if  $R() \geq 15$  minutes. We will present more details of selecting infection probability models and show the experimental result of applying different models in the experimental study (Section 7).

**Incubation Period.** When a passenger is infected, she/he is not able to infect other healthy passengers immediately because of the incubation period. We define the incubation period as  $\gamma$  hours. For instance, if  $p_n$  is infected at the trip  $tr_i$ , then  $p_n$  will be infectious and detectable from  $tr_i.tb + \gamma$  onward, i.e.,  $\gamma$  hours after  $tr_i$ .

**Expected Number of Infections** Let  $I(p_n^i)$  denote the set of people who will be infected by  $p_n^i$ . Accordingly,  $I(p_n) = \cup_{tr_i \in p_n.Tr} I(p_n^i)$  denotes the set of people who will be infected by  $p_n \in P$ . Then, we have  $\cup_{p_i \in P} I(p_i)$ , the set of people who will be infected by  $P$ . Consequently, we have  $I_{\mathcal{P}, \mathcal{T}, \gamma}(P) = |\cup_{p_i \in P} I(p_i)| = \mathbb{E}[\sum_{p_n \in \mathcal{P}} Pr(p_n)]$ , which is the expected number of infected people caused by  $P$ .

## 4 DIFFUSION INTERVENTION SIMULATOR

This section introduces the diffusion intervention simulator consisting of a novel graph structure, TCG, and the diffusion model. As highlighted in Section 1, two fundamental challenges are high memory consumption caused by generating snapshots, and infection probability measurement. Therefore, we propose TCG and the diffusion model to solve these challenges. In what follows, we first introduce the drawbacks of employing existing works in dealing with the dynamic infectious graph; we then introduce a novel graph structure, namely Timeline Contact Graph (TCG); we finally explain how the model simulates the infection diffusion to find all potentially infected people based on a set of initial infected people.

### 4.1 Pain Points

As mentioned in Section 2, most existing studies are based on snapshots, where each node in a snapshot represents a passenger, and an edge between two passengers denotes the infection probability between them. A new snapshot is created to capture changes once the graph changes, including adding/deleting nodes/edges. However, the snapshot-based solutions suffer from either the efficiency issue or the accuracy issue, as shown in Example 1.

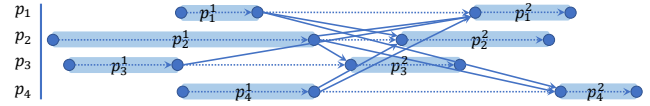
**Accuracy Issue.** The snapshot-based solutions may overlook the *fine-grained temporal order* of close contacts, as the order information is lost in one snapshot.

**EXAMPLE 3.** Figure 1 presents how the dynamic graph with snapshots for every 15 minutes captures the contacts between persons. Since  $p_1$ ,  $p_2$  and  $p_3$  are at the same location from 8:00:00 to 8:14:59 on day 1 (i.e., they are connected to each other in the third snapshot of Figure 1), they can infect each other. However, it is incorrect because the moment  $p_1$  checks-in this location,  $p_3$  checks-out the location. Consequently,  $p_1$  cannot infect  $p_3$  on day 1, which contradicts the facts captured by the first snapshot. The correct contact relationship is shown in Figure 3.

**Efficiency Issue.** As shown in the above example, a low-frequent snapshot (e.g., generating a snapshot every 30 minutes) cannot

**Table 1: Important notations**

Notation	Description
$\mathcal{P}, \mathcal{T}$	Passenger database; Trip database
$\gamma$	Incubation threshold
$p_n^i$	The $i$ -th trip of passenger $p_n$ , $p_n^i \in p_n.Tr$
$R()$	The duration of exposure of one contact
$f()$	Infection probability measurement based on $R()$
$I(P)$	Infection range, the set of people infected by $p$
$W(S)$	Coverage width, the set of check-ins covered by $S$



**Figure 3: Timeline Contact Graph according to Figure 1**

capture a correct close contact, while a high-frequent snapshot (e.g., 10 seconds) will be unaffordable to any existing solution.

For example, deploying checkpoints, the downstream task that we will introduce later, is studied in a recent work [5]. It prevents infection diffusion by removing nodes based on the first eigenvalue. However, the experimental result shows that removing 50 nodes from a graph with 13,947 nodes and 61,168 edges will take more than 4,000 seconds. Suppose we utilize the snapshot-based solution, each snapshot will contain all passengers as nodes, and edges connect all pairs of passengers who have close contact in that interval. Considering the number of passengers on the same vehicle at the same time, every trip of one passenger will have hundreds of close contacts or even more. With similar performance to the work [5], based on our dataset, each snapshot with a one-hour interval will cost at least one hour to find high-infection people in each snapshot. As a result, a four-week virus prevention task may cost several weeks. It is time-consuming for virus prevention as the pandemic situation is rapidly changing.

### 4.2 Timeline Contact Graph (TCG)

In order to address the above issues, we propose TCG that is able to capture all the close contacts between trips. Essentially, in this graph, a vertex is not a passenger, but a trip.

**DEFINITION 1.** *Timeline Contact Graph: A TCG is an ordered pair  $G = (V, E)$ , where  $V$  refers to a set of vertices, with each representing a trip  $tr_i \in \mathcal{T}$ , and  $E = \{\{v_i, v_j\} | v_i, v_j \in V \text{ and } v_i \neq v_j\}$  refers to a set of edges. Each edge  $\{v_i, v_j\}$  refers to the close contact from  $v_i$  to  $v_j$ , and is assigned with a weight  $w_{(i,j)}$  indicating the infection probability from  $v_i$  to  $v_j$ . i.e.,  $w_{(i,j)} = Pr(p_n^i, p_m^j)$ .*

There are two different types of edges as shown in Figure 3, the self-infection edge (dotted-lines) and the infection edge (solid-lines) between different passengers. The former is to connect a trip from boarding to alighting, or connect a trip made by a passenger to her next trip based on the temporal order of the trip time. We assume self-infection is 100%. The latter is to connect a trip  $tr_j$  made by passenger  $p_m$  to another trip  $tr_i$  made by a different passenger  $p_n$ , if they have a close contact, i.e.,  $Pr(p_n^i, p_m^j) > 0$ .

Noting that, we adopt the SI model that does not consider the recovery and reinfection cases, in which an infected person cannot be recovered and then be infected again. It is because we only focus on tracing the infection diffusion through the precise individual-level close contact for a short period (i.e., 7-28 days) instead of simulating the virus trend at the population-level for a long time. In such a short time, one person is unlikely to go through the full SIRS cycle (i.e., susceptible, infected, recovered, susceptible).

### 4.3 Infection Propagation Model

When an infected passenger  $p_n$  is making a trip  $tr_i$ ,  $p_n$  will infect another passenger  $p_m$  with a probability  $Pr(p_n^i, p_m^j)$ . We describe the following routine to simulate the infection propagation process.

1. We initialize an empty priority queue of trip records, with all the trips sorted in an ascending order of  $t_b$ .
2. Given a set  $P$  of passengers who are initially infected, we assume the remaining passengers (i.e.,  $\mathcal{P} - P$ ) are healthy. We push the first trips of all the infected passengers (i.e.,  $\cup_{p_n \in P} p_n^1$ ) into the queue.
3. The queue pops a record  $tr_i$  iteratively. While checking in a POI via  $tr_i$ ,  $p_n$  infects other passengers in the same POI with a probability  $Pr(p_n^i, p_m^j)$ .
4. We push all the trips that have not been infected but will be infected by  $tr_i$  into the queue.
5. Repeat steps 3 and 4 until the queue is empty.

The TCG has two advantages. *First*, by utilizing the TCG, the enormous discrete graph structures (e.g., snapshots) can be transformed into a single graph that contains all the information, which avoids the dilemma of whether losing the accuracy of changing edges among vertexes or suffering from the explosive amount of data. *Second*, according to the government's rules, the levels of strict intervention may change, and hence the definition of close contact varies. The TCG handles this scenario easily by involving a threshold for pushing infected trips. For example, for a loose intervention rule that contacts for more than 10 minutes will be counted as close contact, we only push trips where the infection probability is larger than  $f(10\text{minutes})$  into the queue. The following example shows the infection propagation process.

**EXAMPLE 4.** This example is based on Figure 2. For initialization, we set the incubation period as 12 hours, and set  $p_1$  to be infected and push  $p_1^1$  into the queue, while both  $p_2$  and  $p_3$  are healthy. First, we pop  $p_1^1$ , and calculate the infection probability from  $tr_1$  with all contacts. Assume  $p_2$  is infected after  $p_2^1$ , due to the incubation,  $p_2$  is infected at 20:01:12. Subsequently, we push  $p_2^2$  into the queue. In addition,  $p_1^2$  will be infected by  $p_1^1$  with 100% as it is a self-infection.

## 5 EPIDEMIC MITEGATING IN PUBLIC AREA

As mentioned in Section 1, our second goal is to evaluate the effectiveness of different checkpoint deployment strategies from two different perspectives (defense v.s. offense). In this section, from the defensive perspective, we study the problem of Epidemic Mitigating in Public Area (EMA), in order to find an ideal checkpoint deployment strategy such that virus diffusion can be well mitigated.

Noting that, it is almost infeasible to accurately estimate the number of infected persons in practice [11, 20, 34, 42, 45]. This implies that the measures of controlling pandemics are usually operated without knowing the *exact* picture of the disease. Therefore, we study a more challenging yet practical problem for policy-makers: how to maximize the coverage of checkpoints? Essentially, the coverage of a POI is evaluated by the "vulnerability", that is, if infected persons arrive at this POI, how many persons would be infected.

In the following, we first introduce the problem definition of EMA and prove that EMA is NP-hard. Next, based on the novel structure TCG, we propose a concept called *Contact Paths Tracing (CPT)* that is generated based on the latest check-in record of a person  $p_n$ . It presents a set of contacts that have the potential to infect the person  $p_n$ . Last, we propose our solution to address the EMA problem based on CPT.

### 5.1 Problem Definition of EMA

The motivation of EMA is that, if we could detect that  $p_1$  is infected before she accesses a POI, we could stop  $p_1$  from accessing a POI and infecting other persons in the same POI, and meanwhile quarantine  $p_1$  from spreading the virus in the following days. To implement the above idea to block the diffusion of the virus, we introduce a checkpoint corresponding to a POI, which allows us to implement certain measures (e.g., new cutting-edge rapid tests are able to detect COVID-19 in a few minutes [9, 13, 40]) and to find all potentially infected persons when they are accessing POIs.

**DEFINITION 2.** *Checkpoint: Given a POIs and a person  $p_n$  who makes a check-in  $tr_i$  at  $s$ , if  $s$  is set as a checkpoint,  $p_n$  is allowed to check-in and check-out if she is healthy and meanwhile passes the screening test implemented at  $s$ ;  $p_n$  will be isolated if she is infected (and accordingly,  $p_n.Tr = \{tr_i, \dots, tr_{|Tr|}\}$  are removed from  $\mathcal{T}$ ).*

Recall that  $I(p_n^i)$ , introduced in Section ??, denotes the set of persons who will be infected by  $p_n^i$  (i.e., have contacts with  $p_n$  during  $tr_j$ ). Let  $W(tr_i) = I(p_n^i)$ . Accordingly, we define a new metric  $W(s_n)$  to denote the coverage of the checkpoint  $s_n$  below, where  $\partial s_n$  refers to a set of check-in records such that  $\forall tr_i \in \partial s_n, tr_i.s = s_n$ . Consequently, we have  $W(s_n) = \cup_{tr_i \in \partial s_n} W(tr_i)$ . We are now ready to formally present the problem definition for EMA.

**DEFINITION 3.** *EMA problem: Given a person dataset  $\mathcal{P}$ , a check-in record dataset  $\mathcal{T}$ , a checkpoint candidates dataset  $\mathcal{S}$ , a budget  $k$ , and an incubation threshold  $\gamma$ , EMA is to find a set of POIs  $S \subseteq \mathcal{S}$  to set up checkpoints so as to maximize the coverage of checkpoints.*

$$S = \underset{|S| \leq k}{\operatorname{argmax}} W_{\mathcal{P}, \mathcal{T}, \gamma}^*(S) = |\cup_{s_i \in S} W(s_i)|. \quad (1)$$

A straightforward solution is to select the top- $k$  POIs that have the largest number of check-in records. However, a POI with a larger number of check-in records is not necessarily a POI having a higher vulnerability. For example, considering a POI with many check-in records that come from a fixed crowd with a constant movement pattern, where another POI with fewer check-in records from a group of highly mobile persons. Clearly, the latter POI is more vulnerable. The experimental results to be reported in Section 7.6 also confirm this observation. As compared to the checkpoint deployment strategy that always picks the top- $k$  POIs

(based on the volume of check-in records), our solution can improve the coverage by at most three times.

Theoretically speaking, an optimal checkpoint deployment strategy should be able to well mitigate the virus from spreading among all situations, which is equivalent to minimizing the maximum number of finally infected persons no matter who the initially infected persons are [24, 29]. However, to the best of our knowledge, there is no existing work that could be applied to address our problem because of two main reasons: (1) They do not consider a varying graph as a result of persons' movements. (2) The scale of the people/check-in records considered by existing work is significantly smaller than that considered in our problem, and hence all existing works cannot be a solution to our problem because of efficiency issue. For instance, existing work [29] proposes a dynamic index data structure (WC) designed for influence analysis when some of edges or nodes are removed from the graph. With the dataset that is one order of magnitude smaller (Flickr (WC)  $2.2 \times 10^6$  nodes,  $2.3 \times 10^7$  edges; EZ-LINK (Our work)  $1.3 \times 10^7$  nodes,  $1 \times 10^8$  edges), WC needs more than 4,000 seconds to build the dynamic index data structure, and needs to update the index after removing edges/nodes in each iteration. In contrast, our solution costs less than 300 seconds in the default settings for finding 50 POIs to build checkpoints.

## 5.2 Hardness of EMA

**THEOREM 1.** *The EMA problem is NP-hard.*

**PROOF.** We prove it by reducing the Minimum k-union Problem (MinKU) [43]. In the MinKU problem, given a collection of set  $S' = \{s'_1, s'_2, \dots, s'_m\}$  where each set  $s'_i$  is a subset of a given ground set  $\mathcal{P}'$ , an integer  $0 \leq k \leq m$  and  $0 \leq \tau \leq |\mathcal{P}'|$ , it aims to find  $S' \subseteq S'$ , such that  $|S'| = k$  and  $|\cup_{s'_i \in S'} s'_i| \leq \tau$ . We map the MinKU problem to the EMA problem with the following process: (1) In EMA problem, let each person has two check-ins only; let  $\gamma = 0$ ; let  $f(R()) = 100\%$  if  $R() > 0$ . (2) We map each check-in  $p$  in EMA to each element  $p' \in \mathcal{P}'$  in MinKU, and map each candidate location  $s \in \mathcal{S}$  in EMA to each set  $s' \in S'$  in MinKU. (3) For any check-in  $p_n^i$ , if  $Pr(p_n^i) > 0$ , we connect  $p_n$  to the location  $p_n^i.s$ , and connect the corresponding  $p'_n$  to  $s'$ . Clearly the mapping can be done in polynomial time. Consequently, EMA is equivalent to deciding, given  $k$  and  $\tau$ , whether MinKU problem can find  $k$  subsets  $S'$  such that  $|\cup_{s'_i \in S'} s'_i| \leq \tau$ . If the answer is Yes, then EMA can find  $|S| - k$  locations from  $S$  that saves  $|\mathcal{P}| - |\cup_{s_i \in S} S_i|/2 \geq |\mathcal{P}| - \tau/2$  persons. This is because, for a checkpoint, people who connects to it will be healthy as they only have two check-ins. Therefore, for a set of locations that are not checkpoints  $S$ , the maximum number of infected persons is  $|\cup_{s_i \in S} S_i|/2$ . Since MinKU is NP-complete, the decision problem of EMA is NP-complete. Hence, the optimization problem of EMA is NP-hard.  $\square$

## 5.3 Algorithm 1. CPT-based Station Selection (CPT-SS)

According to Definition 3, given a set of checkpoints  $S$ , we have  $W(S)$  that captures the set of people who can be covered (i.e., saved from being infected) if checkpoints are placed at the set  $S$ . Accordingly, we define  $W(S|s_i) = |W(S \cup \{s_i\})| - |W(S)|$  as the marginal

---

### Algorithm 1: CPT-based Station Selection (CPT-SS)

---

**Input:** Checkpoint number  $k$ , Timeline Graph  $G$ ,  $\theta$

**Output:** Checkpoint set  $S$

```

1.1 Based on  $G$ , generate  $\theta$  CPT set into  $\mathcal{R}$ 
1.2 while  $|S| \leq k$  do
1.3   Select  $S_i \in \mathcal{S}$  that covers the most CPT sets in  $\mathcal{R}$ 
1.4    $S \leftarrow S \cup \{s_i\}$ 
1.5    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_i\}$ 
1.6   Remove all CPT sets covered by  $s_i$  from  $\mathcal{R}$ 

```

---

gain of adding  $s_i$  into  $S$ . A naive greedy-based solution will iteratively select a POI with the maximum  $W(S|s_i)$ . However, the challenge is how to efficiently evaluate  $W(S|s_i)$ . Hence, we introduce the CPT-based POI Selection (CPT-SS).

**DEFINITION 4.** *CPT: Given a person  $p_n \in \mathcal{P}$ , the Contact Paths Tracing (CPT) of  $p_n$  is the set of check-in records in  $\mathcal{T}$  that can infect  $p_n$ .*

Intuitively, a CPT is generated based on the latest check-in record made by a person  $p_n \in \mathcal{P}$  and it includes all the check-in records  $tr$  that have a certain probability ( $> 0$ ) of infecting the person  $p_n$  based on the given infectious model. We generate an CPT set as following:

#### Generation of one CPT.

1. Randomly select  $p_n \in \mathcal{P}$ ; initialize an empty queue  $Q$  and an empty CPT;
2. Add  $tr_i$  into both  $Q$  and CPT where  $tr_i \in p_n.Tr$  refers to the latest check-in record made by  $p_n$ ;
3. Pop the top record  $tr_i$  out of  $Q$ ;
4. For each incoming edge of  $tr_i$  (from node  $p_m^j$ ), flip a coin with  $Pr(p_n^i, p_m^j)$  probability;
5. If true and  $tr_j \notin CPT$ , add  $tr_j$  into both  $Q$  and CPT.
6. Repeat steps 3 to 5 until  $Q$  is empty.

According to Definition 4, all records in the CPT of a person  $p_n$  can infect  $p_n$ . It has been proven that the probability of CPT overlapping with any set of records  $T$  is equivalent to the probability of  $T$  infecting  $p_n$  [2, 14, 39]. Thus, the number of times a record  $tr_i$  appearing in the CPT set of different persons indicates the potential impact of  $tr_i$  in terms of the capability to spread the virus, e.g., the record appearing in the largest number of CPT contributes the most to the spread of the virus and hence shall be isolated first. The last question is how many CPT we need to accurately evaluate  $W(S|s_i)$ .

Let  $\theta$  be the number of CPT sets that we need,  $\mathcal{F}(S)$  be the fraction of CPT sets over all CPT sets covered by a given set of checkpoints  $S$ . The existing work [39] shows that,  $n\mathcal{F}(S)$  is an accurate estimator of  $W(S)$ , when  $\theta$  is sufficiently large, i.e.,  $\theta \geq n(8+2\epsilon)(l \log n + \log \binom{n}{k} + \log 2)/(OPT \cdot \epsilon^2)$ , where  $n$  is the number of check-ins and  $\epsilon$  is the approximate ratio.  $OPT$  can be estimated by  $n(1 - (1 - w(CPT))^k)$ , where  $w(CPT)$  is the number of edges of CPT. For more details of deriving  $\theta$ , please refer to the work [39].

Algorithm 1 presents the pseudo-code of CPT-SS. To minimize the number of infected persons, we need to find POIs that have the largest coverage, i.e., able to block the most dangerous check-in records that may infect a large number of persons. Since the most dangerous check-in record is expected to appear in the most number



of CPT sets, we generate CPT sets and then find checkpoints that can detect the largest number of dangerous check-in records. We first generate  $\theta$  CPT sets as  $\mathcal{R}$  based on the timeline contact graph  $G$  (Line 1.1). In the selection loop, we select a POI  $s_i$  that covers the most CPT sets in  $\mathcal{R}$  (Lines 1.3-1.5), and remove all CPT sets that are covered by  $s_i$  from  $\mathcal{R}$  (Line 1.6). The loop terminates when  $|\mathcal{S}| = k$ , and  $S$  is returned.

## 6 EPIDEMIC MAXIMIZED SPREAD IN PUBLIC AREA

In the previous section, we study the EMA problem that aims to find  $k$  checkpoints to mitigate the virus diffusion. In this section, from the offensive perspective, we aim to find  $k$  ‘super-spreaders’, a small group of persons that will spread the virus the most thus possibly leading to a pandemic. This is necessary and critical to the government to evaluate the robustness of checkpoint deployment strategies in the extreme case that the virus is spread widely.

### 6.1 Problem Definition of ESA

We first define the Epidemic Maximized Spread in Public Area (ESA) problem. Our goal is to evaluate the effectiveness of a given checkpoint deployment strategy in the worst case. Specifically, given a checkpoint deployment strategy, we aim to find  $k$  persons such that if they are infected at the beginning, they can cause the largest number of persons infected.

With considering the checkpoint, we have the infection probability between two records as below:

$$Pr(p_n^i, p_m^j) = \begin{cases} f(R(\cdot)) & \text{otherwise} \\ 0 & \text{if } tr_i \in S \vee tr_j \in S \end{cases} \quad (2)$$

where  $S$  refers to a set of checkpoints. Intuitively, any infected person is unable to infect others at any check point, since he will be placed on quarantine. Based on Equation 2, we introduce the ESA problem formally as follows.

**DEFINITION 5.** *ESA problem: Given a person dataset  $\mathcal{P}$ , a check-in record dataset  $\mathcal{T}$ , a set of infected persons  $P \in \mathcal{P}$ , a set of checkpoints  $S$ , a budget  $k$ , and an incubation period  $\gamma$ , ESA is to find at most  $k$  persons  $P \in \mathcal{P}$  such that  $P$  can maximize the following equation:*

$$P = \operatorname{argmax}_{|P| \leq k} I_{\mathcal{P}, \mathcal{T}, \gamma, S}(P). \quad (3)$$

### 6.2 Hardness of ESA

**THEOREM 2.** *The ESA problem is NP-hard.*

**PROOF.** We prove it by reducing the Set Cover problem to the ESA problem. In the Set Cover problem, given a collection of sets  $S' = \{s'_1, s'_2, \dots, s'_m\}$  where each set  $s'_i$  is a subset of a given ground set  $\mathcal{P}' = \{p'_1, \dots, p'_n\}$ , it aims to find whether there exist  $k$  of the subsets whose union is equal to  $\mathcal{P}'$ . We reduce the Set Cover problem to ESA with the following process: (1) We map each element  $p'_i \in \mathcal{P}'$  in the Set Cover problem to each person  $p_i \in \mathcal{P}$  in the ESA problem. (2) We map each set  $s'_i \in S'$  in the Set Cover Problem to a set of persons  $\{p_i, \dots, p_j\} = I(p_n) \subseteq \mathcal{P}$  in the ESA problem, where  $p'_m \in s'_n$  maps to the person  $p_m \in I(p_n)$  that will be infected by  $p_n$ . Consequently, the Set Cover problem is equivalent to deciding whether there are the  $k$ -sized set of initially infected persons

---

### Algorithm 2: CPT-based Person Selection

---

**Input:** Person number  $k$ , Timeline Graph  $G$ , A checkpoint set  $S$

**Output:** Person set  $P$

- 2.1 Based on  $G$ , generate  $\theta$  CPT set into  $\mathcal{R}$  according to the updated Step 4
  - 2.2 **while**  $|\mathcal{P}| \leq k$  **do**
  - 2.3     Select  $p_i \in \mathcal{P}$  that covers the most CPT sets in  $\mathcal{R}$
  - 2.4      $P \leftarrow P \cup \{p_i\}$
  - 2.5      $\mathcal{P} \leftarrow \mathcal{P} \setminus \{p_i\}$
  - 2.6     Remove all CPT sets covered by  $p_i$  from  $\mathcal{R}$
- 

$P \in \mathcal{P}$ , such that  $I(P) = |\mathcal{P}|$ . Because the Set Cover problem is NP-complete, the decision problem of EMA is NP-complete. Hence, the optimization problem of EMA is NP-hard.  $\square$

### 6.3 Algorithm 2. CPT-based Person Selection (CPT-PS)

Now, we are ready to present the solution, CPT-based Person Selection (CPT-PS), for the ESA problem. The main objective is to find the set of super-spreaders, with each being able to infect many healthy persons. Similar to the EMA problem, a super-spreader is a person whose records appear in the most number of CPT sets. Based on the timeline contact graph  $G$ , we initialize and generate  $\theta$  number of CPT sets as  $\mathcal{R}$  first. Then, a person  $p_i$  that covers the largest fraction of CPT sets is added into  $P$ ; and all CPT sets that are covered by  $p_i$  are removed from  $\mathcal{R}$ . The loop terminates when  $|P| = k$ , and  $P$  is returned. It is worth noting that there is one difference between ESA and EMA, the process of generating a CPT. According to Definition 2, if an infected person  $p_n$  checks-in at  $s$  via  $tr_i$ ,  $p_n$  will be isolated and  $p_n.Tr = \{tr_i, \dots, tr_{|Tr|}\}$  will be removed from  $\mathcal{T}$ . Moreover, the CPT of  $p_n$  is the set of check-in records in  $\mathcal{T}$  that can infect  $p_n^{|Tr|}$ . Therefore, an infected trip is not able to infect  $p_n$  for the CPT of  $p_n$ . As a result, in the process of generating CPT, we change Step 4 to: for each incoming edge of  $tr_i$ , if  $tr_i.s \notin S$ , flip a coin with  $Pr(p_n^i, p_m^j)$  probability, where  $S$  is the set of checkpoints.

## 7 EXPERIMENT

In this section, we evaluate the effectiveness and the efficiency of our solutions. First, we introduce two infection probability measurements that are common in the epidemic field in Section 7.1. Recall that our work is independent of the choice of infection probability  $f()$  and hence we take  $f()$  as an input to our framework. Second, we introduce the datasets in Section 7.2 and present the detailed experiment setup in Section 7.3. Last but not least, we report the experimental results from Section 7.4 to Section 7.6.

**Evaluation Purposes.** As stated in Section 1, we would like to achieve two goals: 1) Tracing Close Contacts in Short-term, and 2) Deploying Checkpoints in Short-term. Accordingly, we conduct two different sets of experiments. First, we evaluate how accurately and efficiently our timeline contact graph models the virus diffusion as compared to the dynamic graph with different snapshots, and present the results in Section 7.4. Second, we evaluate how well

**Table 2: Statistics of Datasets**

Dataset	$ \mathcal{T} $	$ \mathcal{P} $	$ \mathcal{S} $	Period
D-1	$2 \times 10^6$	$5.2 \times 10^4$	4962	14 days
D-2	$1.3 \times 10^7$	$6.3 \times 10^5$	4962	28 days

**Table 3: Parameter Settings**

$k$	25, <b>50</b> , 75
$\gamma$ (Hours)	24, <b>72</b> , 120
$EC_{50}$ (Seconds)	1800, <b>3600</b> , 7200
Period (Days)	7, <b>14</b> , 21, 28
$P_s$	0.01%, 0.05%, <b>0.1%</b> , 0.2%, 0.5%, 1%

the various algorithms could mitigate the epidemic from a set of infected persons from two perspectives. (1) *Defensive perspective*: a set of infected persons that are randomly chosen (the EMA problem) in Section 7.5; (2) *Offensive perspective*: a set of infected persons who are specifically selected to maximally attack checkpoints (the ESA problem) in Section 7.6.

### 7.1 Infection Probability Measurement

As described in Section 3,  $f()$  denotes the infection probability and  $R()$  is the exposure duration. To demonstrate that our work can take any infection probability  $f()$  as an input, we adopt two different models with unique  $f()$  for illustration. The first model follows studies in the virus domain [7, 21, 37] and sets  $f(R) = 1/(1 + (EC_{50}/R)^\tau)$ , where  $EC_{50}$  is a time threshold that a healthy person will have a 50% probability of being infected if she stays in the same POI closely with another infected person;  $\tau$  is used to adjust the slope of  $f()$ . We name it as  $EC_{50}$ , and set  $\tau = 3$ . The second model is based on a threshold of exposure duration, which we name as  $InfTre$ . Specifically, given a threshold of exposure duration,  $f() = 100\%$  if  $R()$  is not shorter than the threshold, and  $f() = 0\%$  otherwise. For example, for the COVID-19, more than 15 minutes of contact can be considered as highly infectious [32]. Therefore, we set the threshold of  $InfTre$  to 15 minutes. We have  $f() = 100\%$  if  $R() \geq 15$  minutes, and  $f() = 0\%$  otherwise.

We adopt the  $EC_{50}$  model as the default model and present the experimental results of  $InfTre$  model in Section 7.7. It is worth noting that, the overall observations that we make from two models are consistent. Again, we want to highlight that the choice of measurement  $f()$  is orthogonal to this work so long as  $f()$  is related to the exposure duration.

### 7.2 Datasets

As reported in Table 2, we use two different EZ-LINK datasets recording trips of different months (provided by the authors of [41]). We adopt the first dataset in our experimental study as the default dataset. The second dataset is used to evaluate the scalability of our solutions with the results to be reported in Section 7.8. EZ-link is a smart card system used in Singapore for public transport. The number of passengers after excluding those taking less than two trips daily is listed in the table. We want to highlight that the scale of the datasets is close to  $|\mathcal{P}|/|\mathcal{T}|$  at a city-level.

A trip record from the public transport data is not exactly the same as the check-in record defined in Section 3. Our model can be applied to both trip records and check-in records with a slight modification. We define a trip  $tr_i$  as a tuple  $\{b, s_b, s_e, t_b, t_e\}$ .  $b$  denotes the riding bus,  $s_b$  and  $s_e$  denote the boarding station and alighting station respectively, and  $t_b$  and  $t_e$  are the boarding time and alighting time respectively. The deviation from the original check-in records does not affect the virus diffusion, but only the exposure duration calculation  $R$  and the checkpoint. To be more specific, for  $R$  w.r.t. trip records, the critical condition is modified as  $tr_i.b = tr_j.b$ . For normal check-in records, each record is for one POI and there is no  $s_b$  or  $s_e$ ; for trip records,  $b$  refers to the physical bus and  $s_b$  and  $s_e$  refer to two different stations. Accordingly, we rewrite the definition of a checkpoint as below.

Given a checkpoint  $s$ , when an infected person  $p_n$  boards or alights at  $s$  via trip  $tr_i$ ,  $p_n$  will be isolated before boarding or after alighting at  $s$ , i.e.,  $p_n.Tr = \{tr_i, \dots, tr_{|Tr|}\}$  will be removed from  $\mathcal{T}$ .

It is worth noting that, if the infected passenger is identified before boarding, she will be isolated immediately. However, if she is only checked after alighting, although she will be isolated immediately, passengers taking the same bus as her on this trip still run the risk of being infected.

Astute readers may wonder whether the public transport system data can represent the individual-level city-wide movement. We chose EZ-LINK dataset for three seasons.

First, for privacy protection, it is hard to get city-wide movement data with individual identification. There are some larger-scale datasets in which each movement record is anonymous. However, without individual identification, it is impossible to evaluate the infection range through tracing the movement of a particular infected person, as it is not clear when a person was infected who will be the close contact with him in the following days.

Second, to the best of our knowledge, other general check-in datasets with individual identification are in much smaller scales. For example, the Australian government implements a COVID Safe Check-In program [?] to record the check-in and check-out data to help with virus intervention. However, from 1/1/2022 to 5/24/2022, the average number of daily check-in for each person is less than two; the average number of daily check-in for each place is around 40. In contrast, for the EZ-LINK dataset, the average number of daily check-in for each person is around eight; the average number of daily check-in for each place is more than five thousand. Therefore, the scale of the EZ-LINK dataset is close to a city-level, and is more challenging and suitable for our problems.

Last but not least, public transport is the perfect bedding for virus diffusion, with a closed space for a sufficiently long time of close contact. From the perspective of infection diffusion intervention, governments should prioritize it.

### 7.3 Experiment Setup

All key parameters are summarized in Table 3, including the number of checkpoints  $k$ , the incubation period  $\gamma$ , and the threshold of contact time  $EC_{50}$ . In each set of experiments, we vary only one parameter and set the rest parameters to their default values highlighted in bold.



**Incubation period  $\gamma$ .** This is a period between when a passenger is able to infect others and when she is infected. It varies from disease to disease. For instance, the average incubation of Influenza is two days [8], and that of COVID-19 is five days [19, 31]. Hence, we set  $\gamma$  to 24 hours (1 day), 72 hours (3 days), and 120 hours (5 days), to represent short, normal, and long incubation periods to cover most cases.

**Transmissibility  $EC_{50}$ .** As aforementioned in Section 7.1, a smaller  $EC_{50}$  indicates that the healthy passengers are more easily infected. The longer the contact time they have, the higher the infected probability is, and vice versa. In our study, we set  $EC_{50}$  to 1,800 seconds, 3,600 seconds, and 5,400 seconds, which represent high, normal, and low transmissibility.

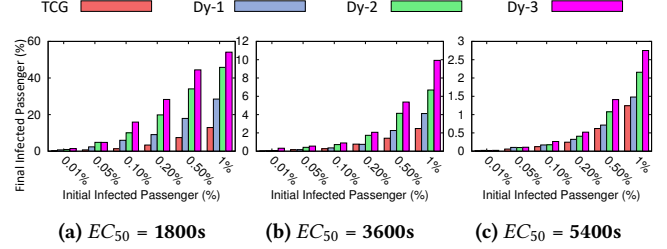
**Percentage of Initially Infected Passengers  $P_s$ .** The number of initially infected passengers is important. The earlier the government is involved in the epidemic, the fewer the initially infected passengers. Accordingly, it will be easier to mitigate the epidemic. A straightforward setup is to assume different numbers of initially infected passengers, e.g., 1,000 or 10,000. However, a fixed number is meaningless without considering the total number of passengers. Therefore, we set  $P_s$  as the percentage of the passengers that are initially infected. Then, the number of initially infected passengers will be  $P_s \times |\mathcal{P}|$ , where  $|\mathcal{P}|$  is the total number of passengers. In each experiment, we set  $P_s$  to 0.01%, 0.05%, 0.1%, 0.2%, 0.5%, and 1%, which represent the cases from a few initially infected persons to a small group of initially infected persons.

**Experiment Environment.** All codes are implemented in Java. They are available online<sup>1</sup> for reproducibility. All experiments are conducted on a machine with a CPU of Intel Core i7-8750U, 32GB RAM and running Windows 10 OS.

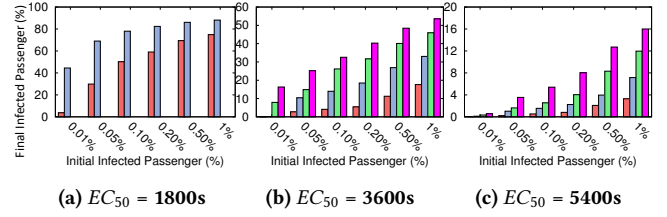
**Performance Metrics.** The total number of infected passengers and the running time are employed as the effectiveness metric and the efficiency metric respectively. For each experiment, we report the average result of 20 runs.

**Compared Methods.** The comparison is conducted from two perspectives, w.r.t. two different goals presented earlier.

- (1) **Diffusion Models:** we compare our TCG with the dynamic graph [35] of different durations, i.e., Dy- $x$ . Here, Dy- $x$  stands for dynamic graph with snapshots and  $x$  refers to the number of hours between two consecutive snapshots (i.e., a smaller  $x$  value indicates that more snapshots will be generated based on a shorter interval). Each snapshot is a graph  $G = (V, E)$  representing the contact information of all passengers, where  $V$  is the set of passengers. For every two passengers  $v_i$  and  $v_j$ , there will be an edge connecting them if within the duration of  $x$ -hours they have a contact (i.e., have check-in records in the same location). The weight of the edge is the infection probability, that is measured as  $f(R)$ , where  $R$  is the shortest time that  $v_i$  or  $v_j$  stays at the location.
- (2) **Checkpoints Deployment Strategies:** (1) For the EMA problem, we compare three deployment strategies: i) Top- $k$ :  $k$  checkpoints that cover the largest number of trips, ii) CPT-SS: the CPT-based checkpoints selection (Section 5.3),



**Figure 4: Graph Competition of varying  $EC_{50}$  (Seconds) when incubation  $\gamma = 72$  hours within 7 days**



**Figure 5: Graph Competition of varying  $EC_{50}$  (Seconds) when incubation  $\gamma = 72$  hours within 14 days**

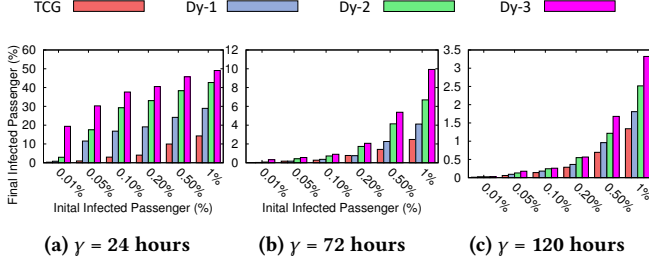
and iii) Eig: the eigenvalue-based solution presented in Section 2.1. Despite the different goals, most of VI studies are based on an eigenvalue solution [3, 4, 10, 12, 15, 23, 25, 26, 36, 46]. Therefore, we build the Eig baseline using the same methodology. The weight of an edge between nodes is the normalized value that refers to the number of trips connecting them. In each iteration, we select a node that maximizes the decrease of the first eigenvalue. (2) For the ESA problem, we compare two different sets of initially infected passengers: i) *Rand*:  $P_s$  randomly-selected passengers, and ii) *Max*:  $P_s$  super-spreaders identified by CPT-PS (Section 6.3).

## 7.4 Evaluation of The Timeline Contact Graph

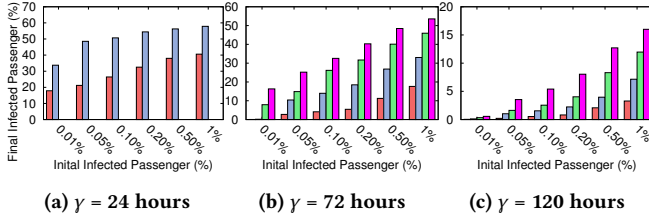
In our first set of experiments, we compare the accuracy and efficiency of our TCG and the dynamic graph in terms of modelling the virus diffusion without checkpoints. We vary two parameters, the transmissibility  $EC_{50}$  and the incubation period  $\gamma$ . As mentioned in Section ??, the lower the  $EC_{50}$  and  $\gamma$  are, the faster the breakout of a disease is. Therefore, a small  $EC_{50}$  and a small  $\gamma$  require accurate modelling. All initially infected passengers are randomly picked. Figures 4 - 7 show the experimental results. We make the following four main observations.

First, TCG can more accurately model the finally infected passengers. For instance, in Figure 4b, when  $P_s = 0.1\%$ , Dy-1, Dy-2 and Dy-3 overestimate the number of infections by 1.6 times, 2.9 times and 3.8 times than TCG, respectively. This is because when the model is based on the snapshots, the temporal orders of trip records might not be accurately captured, e.g., two passengers will be regarded as close contacts when they appear in the same bus within one snapshot, even though the durations they stay in the bus do not overlap. Example 3 illustrates how the snapshot-based solution inaccurately models the virus spreading. In order to correctly

<sup>1</sup><https://github.com/rmitbgroup/VirusPrediction>



**Figure 6: Graph Competition of varying  $\gamma$  when incubation  $EC_{50} = 3600$  seconds within 7 days**



**Figure 7: Graph Competition of varying  $\gamma$  when incubation  $EC_{50} = 3600$  seconds within 14 days**

capture the contacts between all passengers, it requests a new snapshot whenever there is an arriving/leaving passenger. Obviously, it is impractical to track virus diffusion using dynamic graphs for a public system that has millions of passengers.

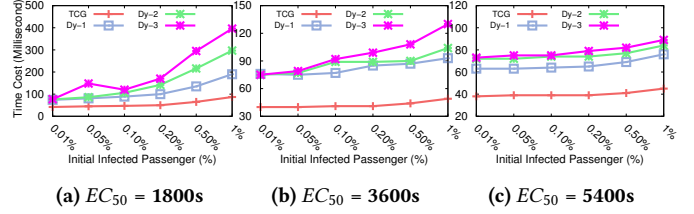
Second, the advantage of TCG becomes more significant when  $EC_{50}$  becomes smaller. For example, when  $EC_{50} = 1,800$ s, in terms of the number of infected persons, TCG outperforms Dy-1, Dy-2, and Dy-3 by 4.4 times, 7.5 times, and 11.9 times respectively, as shown in Figure 4a.

Third, with an increasing prediction period (i.e., from 7 days to 14 days), the overestimation of dynamic graph of different durations becomes even larger and the concern on efficiency becomes more severe. In Figure 5a, Dy-2 and Dy-3 are unable to finish the modelling due to memory overflow. Even Dy-1 that takes one snapshot per hour is not able to achieve a good accuracy and an even higher frequency is required to enhance the accuracy as the number of overestimation (even under Dy-1) is huge.

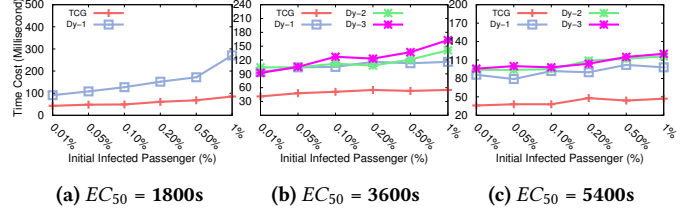
Last, the experimental results reported in Figure 6 and Figure 7 demonstrate a similar trend. When  $EC_{50}$  is small, a short exposure duration may lead to an infection. This in turn will cause an explosive increase in the number of infected passengers, since dynamic graph (based on snapshots) mistakenly considers non-contact passengers as having contact. Therefore, given a small  $EC_{50}$ , it is very easy for dynamic graph to overestimate the number of infected passengers. For the same reason in Figure 5a, we omit the results of Dy-2 and Dy-3 in Figure 7a.

We also evaluate the efficiency of different methods and the results are reported in Figure 8 and Figure 9. It is worth noting that we only report the running time of modelling the virus diffusion but exclude the time required to build checkpoints. We have the following observations.

Consistent with our expectation, TCG requires significantly less time, as compared with dynamic graph. For dynamic graphs, a short



**Figure 8: Running Time of varying  $EC_{50}$  (Second) when incubation  $\gamma = 72$  hours within 7 days**



**Figure 9: Running Time of varying  $EC_{50}$  (Second) when incubation  $\gamma = 72$  hours within 14 days**

interval between snapshots will obviously accelerate the modelling. Apparently, compared with the 7-day prediction, the 14-day prediction has a higher running time for all models. One interesting point is that, with a larger  $EC_{50}$ , the increase of  $P_s$  shows a smaller impact on the running time. It is because a large  $EC_{50}$  can avoid the explosive growth of the final infection number. Therefore, the larger the  $EC_{50}$  is, the smaller the impact of  $P_s$  on running time is. In Figure 9c, results of Dy-2 and Dy-3 are not reported. This is because they both are out of memory as they generate a massive number of edges due to the inaccurate contact recording.

## 7.5 Evaluation of Checkpoints Deployment Strategies to Address the EMA Problem

### 7.5.1 Effectiveness Study.

In this part, we evaluate different algorithms in terms of *how well they can mitigate the virus diffusion under different settings of  $\gamma$  and  $EC_{50}$* . As aforementioned in Section 3, the incubation period  $\gamma$  and transmissibility  $EC_{50}$  determine how easily a virus could be spread. Specifically, under the default number of checkpoints ( $k = 50$ ), we simulate three different categories corresponding to the virus of low/medium/high danger, which lead to nine scenarios, as summarized in Table 4, together with the corresponding results.

For each scenario, we vary the percentage of initially infected passengers,  $P_s$ , from 0.01% to 1%, to simulate how serious the epidemic is. Here, we adopt the *Rand* setting, i.e.,  $P_s \times |\mathcal{P}|$  randomly selected passengers are considered to be initially infected. For evaluating how the algorithms CPT-SS, Top- $k$ , and Eig mitigate the virus diffusion compared to the case without deploying checkpoints, the experimental results of none-checkpoint have been presented with a green bar and named as "None". Figures 10-12 show the experimental results where we make five main observations.

First, with a larger  $P_s$ , all the algorithms result in a larger number of finally infected passengers. It is clear that an early adoption of the measures to control the virus spread could help mitigate the epidemic. Moreover, when both  $\gamma$  and  $EC_{50}$  are small, the virus is

Table 4: Effectiveness Experiment Setup

$\gamma$	$EC_{50}$	High $\longleftrightarrow$ Low		
		24h	72h	120h
High	1800s	Figure 10a	Figure 10b	Figure 10c
$\updownarrow$	3600s	Figure 11a	Figure 11b	Figure 11c
Low	5400s	Figure 12a	Figure 12b	Figure 12c

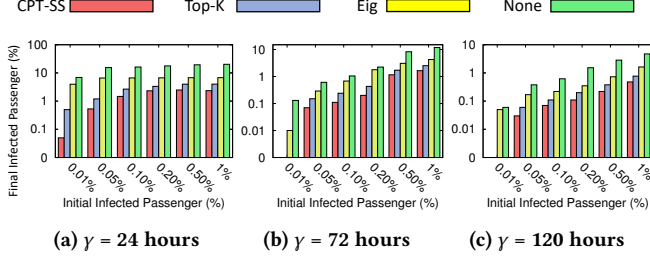


Figure 10: Infection of varying incubation  $\gamma$  when  $EC_{50}=1800s$

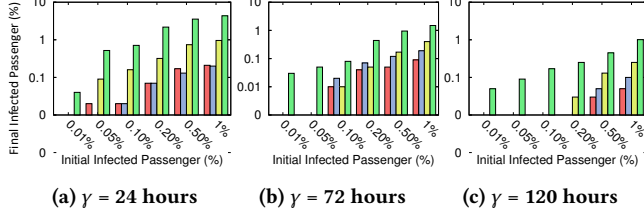


Figure 11: Infection of varying incubation  $\gamma$  when  $EC_{50}=3600s$

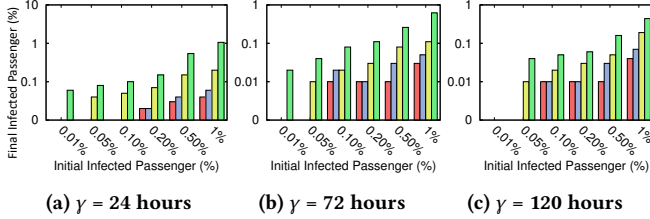


Figure 12: Infection of varying incubation  $\gamma$  when  $EC_{50}=5400s$

highly contagious, and hence a few infected passengers will lead to a huge number of infected passengers.

Second, with a fixed  $EC_{50}$ , the smaller the incubation period  $\gamma$  is, the bigger the number of finally infected passengers will be. This is because when  $\gamma$  is small, a passenger will soon be able to infect others after she is infected.

Third, with a fixed  $\gamma$ , when  $EC_{50}$  becomes larger, the numbers of finally infected passengers under all the algorithms become smaller. This is because a larger  $EC_{50}$  indicates that healthy passengers will be less likely infected.

Fourth, CPT-SS outperforms Top- $k$  and Eig by three times and ten times, respectively. The poor performance of Top- $k$  reflects that establishing checkpoints based on passenger flow is less effective. On the other hand, Eig is unable to find right locations to build checkpoints due to the inaccurate contact recording. It is worth noting that eigenvalue-based solutions severely suffer from the scalability issue. For example, in order to calculate the eigenvalue

Table 5: Efficiency Study of CPT-SS (Second)

Period	$\gamma$	$k$	$EC_{50}$		
			1800	3600	5400
7	24	25	31.02	62.68	88.92
		50	27.88	50.14	67.72
		75	32.06	55.35	61.22
	72	25	42.98	65.34	72.12
		50	30.12	48.89	56.3
		75	33.35	45.17	50.72
	120	25	52.31	62.51	80.59
		50	37.06	45.12	55.1
		75	36.36	44.77	61.95
14	24	25	75.68	137.01	201.4
		50	85.93	143.08	161.55
		75	145.45	178.85	162.15
	72	25	72.17	179.73	191.5
		50	61.26	133.8	143.84
		75	66.49	127.85	139.36
	120	25	102.57	164.63	183.4
		50	77.87	128.38	140.44
		75	73	123.17	141.42

for a dataset with  $1.3 \times 10^7$  trips, it requests an adjacent matrix of size  $(1.3 \times 10^7)^2$ . Neither the storing of this matrix nor the calculation of the eigenvalue of this matrix is practical.

Fifth, the advantage of CPT-SS becomes more significant when the epidemic situation is more serious (i.e.,  $\gamma = 24$  hours or  $EC_{50} = 1,800$  seconds or  $P_s \geq 1\%$ ). This is because when the virus is highly contagious, it is easy to cause an epidemic without planning the deployment of checkpoints carefully. On the other hand, when the virus is relatively less harmful, both algorithms are able to mitigate the epidemic.

In addition, we test how the number of finally infected passengers changes by varying the number of checkpoints  $k$  (i.e., 25, 50, and 75), and report the results in Figure 13. We make the following three observations.

First, with more checkpoints, the number of finally infected passengers becomes smaller. Second, with more checkpoints ( $k = 75$ ), the average advantage of CPT-SS becomes more significant. This is because checkpoints are only able to control the spread of virus to certain degree. When  $k$  is large, since the high volume stations may only located at city, the coverage of them are overlapped, and hence other locations may be missed. Third, when the group of initially infected passengers is bigger (i.e.,  $P_s \geq 0.5\%$ ), the advantage of CPT-SS becomes more significant. The reason is the same as the second one. When there are more initially infected passengers, the pandemic situation becomes more serious. If the deployment strategy could not allocate the limited checkpoints well but “waste” resources (i.e., overlapped coverage), some locations may be missed, and hence causes more infected passengers.

### 7.5.2 Efficiency Study.

The condition of the epidemic changes quickly. The government may need to review and adjust the deployment strategy frequently (e.g., in a daily basis) based on updated epidemic situation. Hence, the efficiency of our mitigation algorithm CPT-SS is critical. We have three main observations.

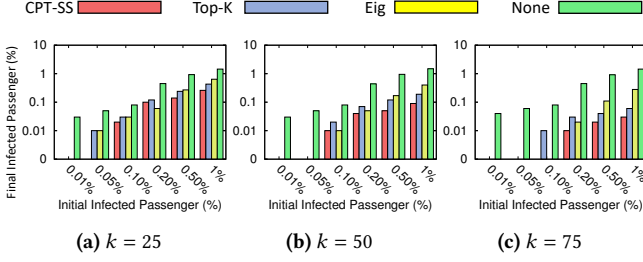


Figure 13: Infection of varying the checkpoint number  $k$

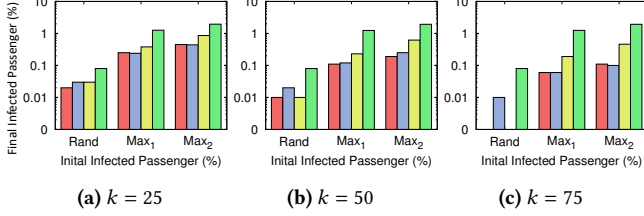


Figure 14: Robustness evaluation of checkpoint deployment strategies

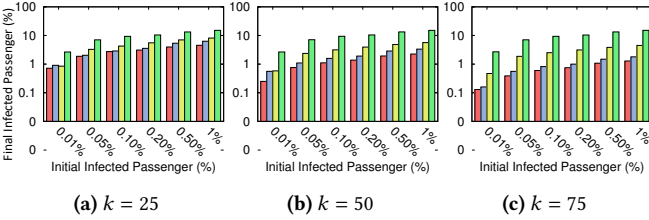


Figure 15: Evaluation of influence models when the threshold of exposure duration is 15 minutes.

First, the increasing  $EC_{50}$  will increase time-cost. According to existing studies [2, 14, 39], a smaller infection probability leads to a smaller average width of CPT (contact paths tracing), which in turn requests a larger number of CPT to evaluate each trip's infection range, hence more time are needed. Second, the number of checkpoints only slightly affects the time-cost of CPT-SS. Intuitively, this is because, when the number of checkpoints increases, we do not need a highly accurate evaluation of the coverage of checkpoints since a larger number will "remedy" the accurate losing. Third, the varying period will hugely affect the time-cost. The reason is that the extending period will contain more trips, increasing the size of the graph, and hence we need more CPT.

## 7.6 Evaluation of the Impact of Super-spreader in ESA

As described in Goal 2 in Section 1, from the offensive perspective, we first find a set of 'super-spreaders' who are a small group of persons that will spread the virus the most, thus possibly leading to a pandemic; we then set this set of 'super-spreaders' as the initially infected passengers and evaluate the robustness of checkpoint deployment strategies.

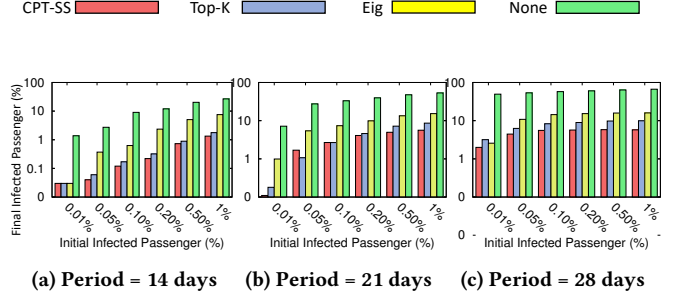


Figure 16: Scalability - Varying period

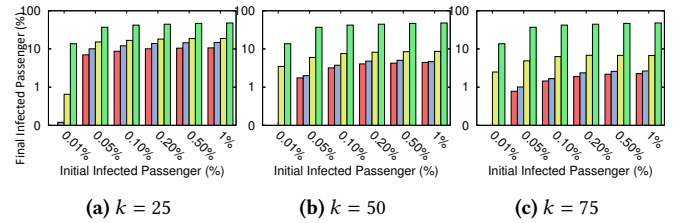


Figure 17: Scalability - Varying  $k$  when  $EC_{50} = 3600s$  and  $\gamma = 72$  hours

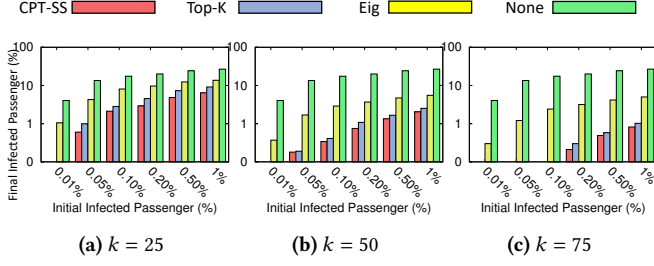
In order to demonstrate the potential impact of 'super-spreaders', we have to generate 'super-spreaders'  $P$ . In our study, we adopt two strategies,  $Max_1$  and  $Max_2$  based on the CPT-PS (i.e., Algorithm 2). The difference is that, in  $Max_1$ , CPT-PS generates  $P$  without knowing the checkpoints  $S$ . In other words, there are few infected passengers riding the public transportation which have the highest passenger flow. Therefore, it tries to simulate a general worst case. In contrast, in  $Max_2$ , CPT-PS generates  $P$  with the knowledge of the detailed deployment of checkpoints  $S$ . Therefore,  $P$  is specifically generated to attack  $S$ . In addition, we also implement  $Rand$ , which assumes that all initially infected passengers are selected from  $\mathcal{P}$  with an equal probability. The experimental results are plotted in Figure 14, and we have three observations.

First, among all experimental results, the number of finally infected passengers of  $Rand$  is smaller than that of  $Max_1$  and  $Max_2$ . It proves that 'super-spreaders' do have a higher risk of causing the pandemic.

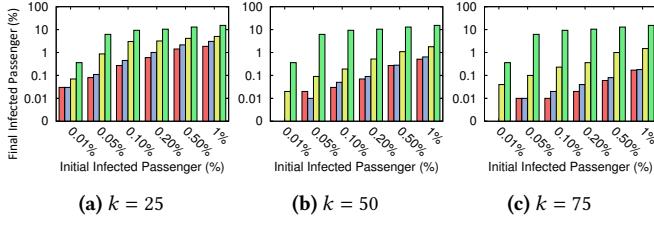
Second, the larger the number of checkpoints is, the more significant the advantage of CPT-SS is. When there are not enough checkpoints, we cannot protect the entire population even when we are able to locate the 'super-spreaders'. As the number of checkpoints increases, it becomes easier and more effective for CPT-SS to deploy checkpoints to block the 'super-spreaders'. It also demonstrates that establishing checkpoints based purely on passenger flow is less effective.

Third, all solutions have a worse result in  $Max_2$ . It is because, in  $Max_2$ , the initially infected passengers are selected specifically to attack checkpoints. Even though, our solution CPT-SS still outperforms others.

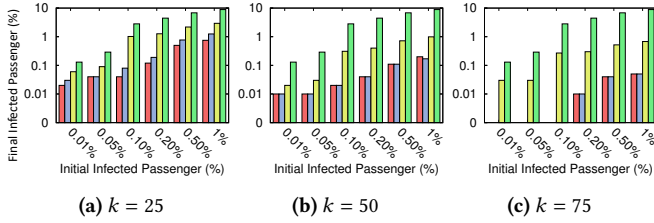




**Figure 18: Scalability - Varying  $k$  when  $EC_{50} = 3600s$  and  $\gamma = 120$  hours**



**Figure 19: Scalability - Varying  $k$  when  $EC_{50} = 5400s$  and  $\gamma = 72$  hours**



**Figure 20: Scalability - Varying  $k$  when  $EC_{50} = 5400s$  and  $\gamma = 120$  hours**

### 7.7 Evaluation of Influence Models ( $EC_{50}$ v.s. $InfTre$ )

Though  $EC_{50}$  is the default model, we also evaluate the other influence model, i.e. the  $InfTre$  model. As aforementioned in Section 7.1, the  $InfTre$  model evaluates the infection probability between two records based on a given threshold of exposure duration. If the close contact time  $R()$  is no shorter than the threshold,  $f() = 100\%$ ; otherwise,  $f() = 0\%$ . As shown in an official report that the threshold for COVID-19 is 15 minutes [32], we set the threshold of exposure duration to be 15 minutes. Figure 15 reports the result.

We observe that, compared with the  $EC_{50}$  model, the numbers of finally infected passengers of all solutions are much higher. This is because the average travel time of bus trips is 826 seconds, which is close to the threshold. Roughly, almost half of contacts will cause a 100% infection ratio. It is worth noting that, even in this situation, CPT-SS still outperforms Top- $k$  and Eig by an average of 50% and three times, respectively.

### 7.8 Scalability Study

In order to evaluate the performance of our solution CPT-SS in a different dataset, we conduct more experiments on a different

EZ-link dataset with a ten times larger size from a different month. This dataset contains more than 13 million trip records. We also extend the prediction period from 14 days to 28 days to evaluate the scalability. Table 6 shows the full experimental results, while Figure Figure 16 to 20 show a partial experimental results.

The result reported in Figure 16 shows the results of varying period. The longer periods is, the higher probability that we may suffer a pandemic. Figure 17 to 20 show the results of varying the number of checkpoints  $k$  in different incubation  $\gamma$  and infectious level  $EC_{50}$ . The results show that our solution consistently outperforms baselines by up to one order of magnitude. More specifically, compared with Top- $k$  and Eig, the average advantages of our solution are around 40% and 350%, respectively. For the effectiveness, our method is able to finish with 10 minutes in all experiments.

We do not conduct the experiment when  $EC_{50} = 1800$  second. In such a long period, a highly infectious virus will almost cause a pandemic, in which more than 90% passengers will be infected for all solutions. It is unrealistic and meaningless for compassion since the difference between each solution is small.

## 8 CONCLUSION

In this work, we study how to trace and mitigate virus diffusion according to fine-grained users' movement records. A critical challenge is how to capture the contact between mobile crowds to allow us to compute the infection probability based on the exposure duration. We first introduce a diffusion model based on TCG, which is able to exactly model the virus diffusion according to users' movements. Based on TCG, we propose *ESA* and *EMA* to assist governments to mitigate the epidemic from the defensive and offensive perspectives. In *EMA*, we aim to find an ideal checkpoint deployment strategy such that virus diffusion can be well mitigated. In *ESA*, we aim to find 'super-spreaders' to spread the virus maximally, in order to test deployment strategies. Finally, we conduct experiments using two real-world station datasets containing millions of public transport trip records to show that our TCG graph is able to evaluate the virus diffusion model more accurately compared with a snapshot-based dynamic graph. Moreover, in terms of finding a checkpoint deployment strategy, our solution outperforms the baselines over randomly infected passengers and super-spreaders under the varying infectious of viruses and different periods.

## REFERENCES

- [1] Roy M Anderson, B Anderson, and Robert M May. 1992. *Infectious diseases of humans: dynamics and control*. Oxford university press.
- [2] Christian Borgs, Michael Brautbar, Jennifer T. Chayes, and Brendan Lucier. 2014. Maximizing Social Influence in Nearly Optimal Time. In *SODA*, Chandra Chekuri (Ed.). SIAM, 946–957.
- [3] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jure Leskovec, and Christos Faloutsos. 2008. Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.* 10, 4 (2008), 1:1–1:26.
- [4] Serina Chang, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, and Jure Leskovec. 2020. Mobility network models of COVID-19 explain inequities and inform reopening. *Nature* (2020), 1–8.
- [5] Chen Chen, Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. 2016. Eigen-Optimization on Large Graphs by Edge Manipulation. *ACM Trans. Knowl. Discov. Data* 10, 4 (2016), 49:1–49:30.
- [6] Yiping Chen, Gerald Paul, Shlomo Havlin, Fredrik Liljeros, and H Eugene Stanley. 2008. Finding a better immunization strategy. *Physical review letters* 101, 5 (2008), 058701.
- [7] Centers for Disease Control and Prevention (CDC). 2006. *Principles of Epidemiology in Public Health Practice*. Technical Report. U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES.

**Table 6: Scalability Study of CPT-SS when Period is 28 days (Second)**

$\gamma$	$k$	$EC_{50}$	P	Random						Max					
				0.01%	0.05%	0.10%	0.20%	0.50%	1%	0.01%	0.05%	0.10%	0.20%	0.50%	1%
72	25	3600	CPT-SS	0.10%	7.02%	8.72%	10.13%	10.52%	10.68%	7.45%	9.66%	10.16%	10.42%	10.43%	10.74%
			TOP-K	0.12%	10.08%	12.09%	13.85%	14.58%	14.86%	10.17%	13.24%	14.08%	14.49%	14.68%	15.07%
			Eig	0.65%	15.27%	16.77%	18.15%	18.68%	18.84%	14.96%	18.20%	18.42%	18.57%	18.54%	18.52%
			None	13.71%	36.99%	42.16%	44.52%	46.74%	47.84%	37.89%	44.86%	45.99%	46.93%	48.08%	48.85%
		5400	CPT-SS	0.03%	0.08%	0.27%	0.60%	1.42%	1.86%	0.88%	1.23%	1.66%	1.93%	2.36%	2.77%
			TOP-K	0.03%	0.11%	0.45%	1.01%	2.16%	3.06%	1.33%	1.88%	2.46%	2.88%	3.55%	4.09%
	50	3600	Eig	0.07%	0.87%	3.02%	3.18%	4.15%	5.04%	2.85%	3.96%	4.68%	5.07%	5.68%	6.16%
			None	0.36%	6.20%	9.33%	10.45%	12.90%	15.24%	7.95%	10.63%	13.29%	14.41%	16.16%	17.48%
			CPT-SS	0.02%	1.75%	3.20%	4.06%	4.25%	4.44%	0.31%	3.57%	4.04%	4.18%	4.28%	4.36%
			TOP-K	0.01%	2.00%	3.75%	4.80%	5.04%	4.65%	2.46%	4.13%	4.81%	4.97%	5.12%	5.38%
		5400	Eig	3.48%	6.01%	7.59%	8.21%	8.47%	8.65%	1.05%	8.06%	8.15%	8.25%	8.46%	8.65%
			None	16.42%	27.80%	31.18%	32.47%	33.30%	33.77%	28.42%	32.10%	32.85%	33.26%	33.64%	33.88%
			CPT-SS	0.00%	0.02%	0.03%	0.07%	0.27%	0.51%	0.04%	0.26%	0.44%	0.48%	0.59%	0.77%
			TOP-K	0.00%	0.01%	0.05%	0.09%	0.28%	0.64%	0.09%	0.17%	0.46%	0.57%	0.82%	1.03%
		3600	Eig	0.02%	0.09%	0.19%	0.52%	1.09%	1.78%	0.22%	1.26%	1.65%	1.82%	2.10%	2.37%
			None	0.36%	6.20%	9.33%	10.45%	12.90%	15.24%	7.95%	10.63%	13.29%	14.41%	16.16%	17.48%
			CPT-SS	0.00%	0.78%	1.45%	1.89%	2.18%	2.25%	0.00%	1.64%	1.92%	2.12%	2.18%	2.28%
			TOP-K	0.00%	1.01%	1.67%	2.37%	2.59%	2.64%	0.88%	1.96%	2.26%	2.47%	2.55%	2.94%
	75	3600	Eig	2.49%	4.89%	6.31%	6.82%	6.80%	6.74%	0.93%	6.73%	6.72%	6.61%	6.40%	6.17%
			None	13.71%	36.99%	42.16%	44.52%	46.74%	47.84%	37.89%	44.86%	45.99%	46.93%	48.08%	48.85%
		5400	CPT-SS	0.00%	0.01%	0.01%	0.02%	0.06%	0.17%	0.00%	0.06%	0.10%	0.13%	0.18%	0.26%
			TOP-K	0.00%	0.01%	0.02%	0.04%	0.08%	0.18%	0.02%	0.06%	0.16%	0.18%	0.26%	0.33%
			Eig	0.04%	0.10%	0.23%	0.36%	1.00%	1.49%	0.02%	1.02%	1.46%	1.64%	1.95%	2.19%
			None	0.36%	6.20%	9.33%	10.45%	12.90%	15.24%	7.95%	10.63%	13.29%	14.41%	16.16%	17.48%
120	25	3600	CPT-SS	0.04%	0.60%	2.12%	2.92%	4.83%	6.41%	2.22%	2.91%	4.43%	5.93%	7.70%	8.50%
			TOP-K	0.07%	0.99%	2.82%	4.53%	7.31%	9.14%	2.93%	4.42%	6.20%	8.51%	10.75%	11.74%
			Eig	1.05%	4.27%	8.07%	9.65%	12.39%	13.67%	6.24%	8.25%	10.86%	12.95%	14.74%	15.74%
			None	5.34%	18.05%	23.81%	26.53%	32.76%	35.73%	17.13%	25.02%	29.01%	33.46%	37.53%	39.74%
		5400	CPT-SS	0.02%	0.04%	0.04%	0.12%	0.50%	0.75%	0.05%	0.27%	0.51%	0.92%	1.51%	1.97%
			TOP-K	0.03%	0.04%	0.08%	0.19%	0.77%	1.25%	0.07%	0.43%	0.79%	1.34%	2.31%	3.02%
	50	3600	Eig	0.06%	0.09%	1.02%	1.27%	2.18%	2.94%	1.03%	2.00%	2.68%	3.16%	4.13%	4.95%
			None	0.13%	0.29%	2.81%	4.46%	6.83%	8.88%	3.31%	6.39%	7.90%	9.62%	11.99%	13.83%
		5400	CPT-SS	0.01%	0.18%	0.34%	0.75%	1.34%	2.04%	0.67%	0.86%	1.18%	1.78%	2.73%	3.25%
			TOP-K	0.00%	0.19%	0.41%	1.08%	1.66%	2.52%	0.56%	0.73%	1.12%	2.29%	3.35%	3.98%
			Eig	0.37%	1.69%	2.88%	3.68%	4.71%	5.52%	2.42%	3.30%	3.94%	4.94%	6.20%	6.70%
			None	4.04%	13.50%	17.47%	20.04%	24.30%	26.80%	13.03%	18.70%	21.22%	24.78%	28.26%	29.81%
		3600	CPT-SS	0.01%	0.01%	0.02%	0.04%	0.11%	0.20%	0.02%	0.04%	0.10%	0.22%	0.40%	0.60%
			TOP-K	0.01%	0.01%	0.02%	0.04%	0.11%	0.17%	0.00%	0.06%	0.11%	0.25%	0.50%	0.70%
		5400	Eig	0.02%	0.03%	0.31%	0.40%	0.72%	0.99%	0.31%	0.78%	0.97%	1.19%	1.58%	1.95%
			None	0.13%	0.29%	2.81%	4.46%	6.83%	8.88%	3.31%	6.39%	7.90%	9.62%	11.99%	13.83%
	75	3600	SIM	0.00%	0.03%	0.07%	0.21%	0.49%	0.82%	0.27%	0.39%	0.51%	0.74%	1.27%	1.52%
			TOP-K	0.00%	0.04%	0.09%	0.30%	0.58%	1.02%	0.36%	0.45%	0.61%	0.89%	1.53%	1.87%
			Eig	0.30%	1.21%	2.41%	3.17%	4.17%	5.01%	2.60%	3.35%	3.79%	4.57%	5.60%	6.01%
			None	3.87%	13.88%	19.01%	21.47%	26.76%	29.35%	16.24%	21.78%	24.19%	28.00%	31.17%	32.75%
		5400	SIM	0.00%	0.00%	0.00%	0.01%	0.04%	0.05%	0.00%	0.01%	0.03%	0.08%	0.12%	0.19%
			TOP-K	0.00%	0.00%	0.00%	0.01%	0.04%	0.05%	0.00%	0.01%	0.03%	0.09%	0.15%	0.22%
			Eig	0.03%	0.03%	0.27%	0.30%	0.52%	0.68%	0.25%	0.59%	0.73%	0.98%	1.27%	1.56%
			None	0.13%	0.29%	2.81%	4.46%	6.83%	8.88%	3.31%	6.39%	7.90%	9.62%	11.99%	13.83%

- [8] Centers for Disease Control and Prevention (CDC). 2020. *Clinical Signs and Symptoms of Influenza*. Technical Report. U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES.
- [9] Edwin Garcia. 2020. *New cutting-edge rapid test detects both COVID-19 and flu*. <https://health.ucdavis.edu/health-news/newsroom/new-cutting-edge-rapid-test-detects-both-covid-19-and-flu/2020/11>
- [10] Salah Ghamizi, Renaud Rwemalika, Maxime Cordy, Lisa Veiber, Tegawendé F. Bissyandé, Mike Papadakis, Jacques Klein, and Yves Le Traon. 2020. Data-driven Simulation and Optimization for Covid-19 Exit Strategies. In *SIGMOD*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 3434–3442.
- [11] Bruno Gonçalves. 2020. *Epidemic Modeling 102: All CoVID-19 models are wrong, but some are useful*. <https://medium.com/data-for-science/epidemic-modeling-102-all-covid-19-models-are-wrong-but-some-are-useful-c81202cc6ee9>
- [12] Sebin Gracy, Philip E. Paré, Henrik Sandberg, and Karl Henrik Johansson. 2021. Analysis and Distributed Control of Periodic Epidemic Processes. *IEEE Trans. Control. Netw. Syst.* 8, 1 (2021), 123–134. <https://doi.org/10.1109/TCNS.2020.3017717>
- [13] Giorgia Guglielmi. 2020. *Fast coronavirus tests: what they can and can't do*. <https://www.nature.com/articles/d41586-020-02661-2>
- [14] Qintian Guo, Sibao Wang, Zhewei Wei, and Ming Chen. 2020. Influence Maximization Revisited: Efficient Reverse Reachable Set Generation with Bound Tightened. In *SIGMOD*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 2167–2181.
- [15] Qianyu Hao, Lin Chen, Fengli Xu, and Yong Li. 2020. Understanding the Urban Pandemic Spreading of COVID-19 with Real World Mobility Data. In *SIGMOD*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 3485–3492.
- [16] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. 2002. Attack vulnerability of complex networks. *Physical review E* 65, 5 (2002), 056109.
- [17] Shixun Huang, Zhifeng Bao, J. Shane Culpepper, and Bang Zhang. 2019. Finding Temporal Influential Users Over Evolving Social Networks. In *ICDE*. IEEE, 398–409.
- [18] Jinha Kim, Wonyeol Lee, and Hwanjo Yu. 2014. CT-IC: Continuously activated and Time-restricted Independent Cascade model for viral marketing. *Knowl. Based Syst.* 62 (2014), 57–68.
- [19] Stephen A Lauer, Kyra H Grantz, Qifang Bi, Forrest K Jones, Qulu Zheng, Hannah R Meredith, Andrew S Azman, Nicholas G Reich, and Justin Lessler. 2020. The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application. *Annals of internal medicine* 172, 9 (2020), 577–582.
- [20] Ruiyun Li, Sen Pei, Bin Chen, Yimeng Song, Tao Zhang, Wan Yang, and Jeffrey Shaman. 2020. Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2). *Science* 368, 6490 (2020), 489–493. <https://doi.org/10.1126/science.abb3221> arXiv:<https://science.sciencemag.org/content/368/6490/489.full.pdf>



- [21] Kun Lin, Daniel Yee-Tak Fong, Biliu Zhu, and Johan Karlberg. 2006. Environmental factors on the SARS epidemic: air temperature, passage of time and multiplicative effect of hospital infection. *Epidemiology & Infection* 134, 2 (2006), 223–230.
- [22] Bo Liu, Gao Cong, Dong Xu, and Yifeng Zeng. 2012. Time Constrained Influence Maximization in Social Networks. In *ICDM*, Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu (Eds.). IEEE Computer Society, 439–448.
- [23] Ji Liu, Philip E. Paré, Angelia Nedic, Choon Yik Tang, Carolyn L. Beck, and Tamer Basar. 2019. Analysis and Control of a Continuous-Time Bi-Virus Model. *IEEE Trans. Autom. Control* 64, 12 (2019), 4891–4906. <https://doi.org/10.1109/TAC.2019.2898515>
- [24] Alvis Logins, Yuchen Li, and Panagiotis Karras. 2020. On the Robustness of Cascade Diffusion under Node Attacks. In *WWW*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 2711–2717.
- [25] Van Sy Mai, Abdella Battou, and Kevin Mills. 2018. Distributed Algorithm for Suppressing Epidemic Spread in Networks. *IEEE Control. Syst. Lett.* 2, 3 (2018), 555–560. <https://doi.org/10.1109/LCSYS.2018.2844118>
- [26] Marco Minutoli, Prathyush Sambaturu, Mahantesh Halappanavar, Antonino Tumeo, Ananth Kalyanaraman, and Anil Vullikanti. 2020. Preempt: scalable epidemic interventions using submodular optimization on multi-GPU systems. In *2020 SC20*. IEEE Computer Society, 765–779.
- [27] Matthieu Nadini, Kaiyuan Sun, Enrico Ubaldi, Michele Starnini, Alessandro Rizzo, and Nicola Perra. 2018. Epidemic spreading in modular time-varying networks. *Scientific reports* 8, 1 (2018), 1–11.
- [28] Cameron Nowzari, Victor M Preciado, and George J Pappas. 2016. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems Magazine* 36, 1 (2016), 26–46.
- [29] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2016. Dynamic Influence Analysis in Evolving Networks. *Proc. VLDB Endow.* 9, 12 (2016), 1077–1088.
- [30] World Health Organization. 2018. *Managing epidemics: key facts about major deadly diseases*. Technical Report.
- [31] World Health Organization. 2020. *Advice on the use of masks in the context of COVID-19*. Technical Report. World Health Organization.
- [32] World Health Organization. 2020. *Contact tracing in the context of COVID-19*. Technical Report. World Health Organization.
- [33] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. 2015. Epidemic processes in complex networks. *Reviews of modern physics* 87, 3 (2015), 925.
- [34] John Allen Paulos. 2020. *We’re Reading the Coronavirus Numbers Wrong*. <https://www.nytimes.com/2020/02/18/opinion/coronavirus-china-numbers.html>
- [35] B. Aditya Prakash, Hanghang Tong, Nicholas Valler, Michalis Faloutsos, and Christos Faloutsos. 2010. Virus Propagation on Time-Varying Networks: Theory and Immunization Algorithms. In *PKDD (Lecture Notes in Computer Science)*, José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag (Eds.), Vol. 6323. Springer, 99–114.
- [36] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. 2014. Optimal Resource Allocation for Network Protection Against Spreading Processes. *IEEE Trans. Control. Netw. Syst.* 1, 1 (2014), 99–108. <https://doi.org/10.1109/TCNS.2014.2310911>
- [37] Christian Ritz, Florent Baty, Jens C Streibig, and Daniel Gerhard. 2015. Dose-response analysis using R. *PLoS one* 10, 12 (2015), e0146021.
- [38] Christian M Schneider, Tamara Mihaljev, Shlomo Havlin, and Hans J Herrmann. 2011. Suppressing epidemics with a limited amount of immunization units. *Physical Review E* 84, 6 (2011), 061911.
- [39] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD*, Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu (Eds.). ACM, 75–86.
- [40] Coronavirus testing finally gathers speed. 2020. *Cormac Sheridan*. <https://www.nature.com/articles/d41587-020-00021-z>
- [41] Xiancai Tian and Baihua Zheng. 2018. Using Smart Card Data to Model Commuters’ Responses Upon Unexpected Train Delays. In *IEEE Big Data*. 831–840.
- [42] Financial Times. 2020. *The mystery of the true coronavirus death rate*. <https://www.ft.com/content/f3796baf-e4f0-4862-8887-d09c7f706553>
- [43] Staal A Vinterbo. 2002. A note on the hardness of the k-ambiguity problem. *Technical Report DSG-T R-2002-006* (2002).
- [44] Bryan Wilder, Sze-Chuan Suen, and Milind Tambe. 2018. Preventing Infectious Disease in Dynamic Populations Under Uncertainty. In *AAAI*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 841–848.
- [45] Sean L Wu, Andrew N Mertens, Yoshika S Crider, Anna Nguyen, Nolan N Pokpongkiat, Stephanie Djajadi, Anmol Seth, Michelle S Hsiang, John M Colford, Art Reingold, et al. 2020. Substantial underestimation of SARS-CoV-2 infection in the United States. *Nature communications* 11, 1 (2020), 1–10.
- [46] Shouhuai Xu, Wenlian Lu, Li Xu, and Zhenxin Zhan. 2014. Adaptive Epidemic Dynamics in Networks: Thresholds and Control. *ACM Trans. Auton. Adapt. Syst.* 8, 4 (2014), 19:1–19:19. <https://doi.org/10.1145/2555613>
- [47] Justin Zhan, Timothy Rafalski, Gennady Stashkevich, and Edward Verenich. 2017. Vaccination allocation in large dynamic networks. *Journal of Big Data* 4, 1 (2017), 2.