Supplementary Information for

EpiMob: Interactive Visual Analytics of Citywide Human Mobility Restrictions for Epidemic Control

Chuang Yang, Zhiwen Zhang, Zipei Fan, Renhe Jiang, Quanjun Chen, Xuan Song, and Ryosuke Shibasaki

Chuang Yang.
E-mail: chuang.yang@csis.u-tokyo.ac.jp

This PDF file includes:

Supporting Information Text

*Note: the underlined red fonts indicate references to the paper body.

Additional Materials for Visualization View

A. Telecommuting View. To implement telecommuting, it is necessary to extract users' home and workplace addresses. Here, we supplemented the details of our home&work places extraction method.

a) Extract the stay points for each individual. We applied the stay point extraction algorithm (1) to the raw trajectory dataset to identify the stops at which the object remained for at least one hour within a distance of 500 m from a given point.

b) Find the Top-1 stay grid during working hours and after hours, respectively. After extracting each user's stay points, we map the coordinates of these stay points' to the corresponding grid and count the total length of stay for each grid during working hours (11: 00–17: 00) and after hours (00: 00–06: 00) separately. Note that the working hours do not include the period 09: 00–11: 00 because of the high possibility of commuting. We then select the grid with the longest stay time during working hours and after hours as a candidate workplace and home address, respectively.

c) Filter out the substandard candidate home and workplaces. Some candidate places may be substandard. For example, during the working hours in one month, the time Jim stays on each grid is $(G_1, 20h), (G_2, 19h), (G_3, 18h)$, and it is difficult to distinguish the workplace. Therefore, we set a threshold for the stay rate, which is the length of stay in a particular grid divided by the total number of working hours (e.g., for a trajectory of 30 days, the total number of working hours is 30 $\times$ 6 = 180h) to extract the qualifying home and workplaces. In this work, we set it as 75% and obtain 38,452 qualified trajectories from our Greater Tokyo Area dataset. Although there are many other ways to detect work locations, this is not the focus of this study.

B. Spatial Propagation Feature View. Here we describe the construction details of spatial transmission network. To construct the spatial transmission network of policy p, for each infected user, its user id, infection time slot t, infection grid g, and the set of potential infection sources $U^I_{g,t}$ were recorded initially during the simulation. Unlike agent-based models, the trajectory-based model implements SEIR at the grid level. The newly infected does not have a specific source of infection but only a set of potential sources ($U^I_{g,t}$). Assuming that for each newly infected user n at grid g at time slot t, the probability of being infected by sources in $U^I_{g,t}$ is equal. Then, for each s in sources $U^I_{g,t}$, the expectation of propagating from s to n is $1/|U^I_{g,t}|$. Let the infection location of s and n be s_g and n_g, respectively. After that, one spatial propagation path $p_{s\to n}$ can be denoted as $(s_g, n_g, 1/|U^I_{g,t}|)$. From the perspective of propagation, s_g is called "initial infection location", and n_g is called "secondary propagation location". By accumulating all propagation expectations from s to n, the corresponding flow, denoted as $flow_{s\to n}$, is obtained, with which the the spatial transmission network can be constructed.

Additional Materials for Trajectory-based SEIR model

Due to the space limitations of the paper body, we supplement more details about the epidemic model here.

1. Details of Implementation. Our epidemic simulation process consists of two phases "*contagion + movement*" that emerge iteratively. *Contagion Stage*. In this stage, we update the user state at each grid. We denote the list of user IDs at grid g at time t by $U_{g,t}$, and the susceptible, exposed, infected, and recovered subsets as $U^S_{g,t}$, $U^E_{g,t}$, $U^I_{g,t}$, and $U^R_{g,t}$, respectively. By solving the differential Equation 8, we obtain the simulated incremental number $\Delta S_{g,t}$, $\Delta E_{g,t}$, $\Delta I_{g,t}$, and $\Delta R_{g,t}$ at time t, and update the lists of users $U^S_{g,t}$, $U^E_{g,t}$, $U^I_{g,t}$, and $U^R_{g,t}$ by sampling newly exposed, infected, and recovered users in corresponding increments. Note that the increments cannot be ignored, although they are always very small (the spatial and temporal resolution is high and the number of increments in a small area during a short time period is always very small). However, to simulate the infection process, the infection is discrete, with a minimal change of 1. Thus, to obtain the discrete values of the increments $\Delta\hat{S}_{g,t}$, $\Delta\hat{E}_{g,t}$, $\Delta\hat{I}_{g,t}$, and $\Delta\hat{R}_{g,t}$, we quantify the number of increments by implementing a sampling strategy to use randomness to represent the small values. For example, if our simulated incremental number of susceptible users is -0.001 at location g at time t, then we sample one new exposed user from the susceptible list $U^S_{g,t}$ with a probability of 0.001.

Movement Stage. In this stage, we update the user list of each grid. As time proceeds, the users' movements will update the user lists $U^S_{g,t}$, $U^E_{g,t}$, $U^I_{g,t}$, and $U^R_{g,t}$. For example, for one infected user u moves from g_0 to g_1 between time t and $t+1$, we update the user lists by 1) remove u from $U^I_{g_0,t}$, and 2) add u to $U^I_{g_1,t+1}$. Combining these two updates to $U^S_{g,t}$, $U^E_{g,t}$, $U^I_{g,t}$, and $U^R_{g,t}$, we obtain the initial state for the next time step simulation $U^S_{l,t+1}$, $U^E_{g,t+1}$, $U^I_{l,t+1}$, and $U^R_{l,t+1}$. By iterating the process over the target time range, we completed the simulation process in the target area.

2. Data for Parameter Calibration. The data used for fitting β_{global} in the Case Study Section comes from the real reported patient data of the Greater Tokyo Area (2). To simulate the effect of different policies, we need to eliminate the interference of real-world prevention and control policies on the simulation results. Thus, we merely selected the data between the date of the first case until the eve of the emergency declaration (2020/01/16-2020/04/05) to perform the fitting, and the results is $\beta_{global} = 0.302$.

3. Additional Experimental Results for Model Evaluation.

A. Model Rationality. The classical SEIR model could not capture fine-grained spatial characteristics of the infection. For example, it would consider the Greater Tokyo Area in its entirety and generate one infection curve. We applied the classical SEIR model to each prefecture of the Greater Tokyo Area to verify this drawback. As a control group, we ran the same simulation with our trajectory-based model to demonstrate the advantages of our model in terms of capturing spatial characteristics.
Experiment setting: 1) For both models, we fix the initial number of infections I_0 to 10 in the Greater Tokyo Area and set the simulation length as one month. Other essential epidemic parameters were set as mentioned at the start of this section. 2) For the classical SEIR model, the number of initial infections in each prefecture is proportional to its population size, which is obtained from the census data. 3) For the trajectory-based SEIR model, we first extracted the trajectories of 38,452 users for a single month (July 2012) from our dataset. All these trajectories show the exact home and work positions. However, compared to census data, the spatial distribution of these users' home locations is biased. Thus, we resampled 30,000 trajectories from 38,452 to make them conform to the distribution of census data.
Results Visualization: To show the difference in the spatial characteristics between our model and the traditional ones, we used a pie chart to present the spatial distribution information of infected individuals (Fig. S1). Treating the Greater Tokyo Area in its entirety, we divided the infected persons living in this area into four groups according to their place of residence (i.e., Tokyo, Chiba, Saitama, and Kanagawa). Each sector in the pie chart represents a prefecture. The size of each sector shows the number of infections as a proportion of the total number of infections in the Greater Tokyo Area.
Results Analysis: Fig. S1(a) shows the spatial distribution information of real infections obtained from the website of the Government of Japan for the period ending with 2020/10/24. Note that we did not choose this date intentionally. The distribution of the residential addresses of infected people in the Greater Tokyo Area has remained stable to date. Fig. S1(b) and Fig. S1(c) presents the simulation result of our model and that of the classical method, respectively. As we can observe from the results, our model performs very well compared to the classical model.

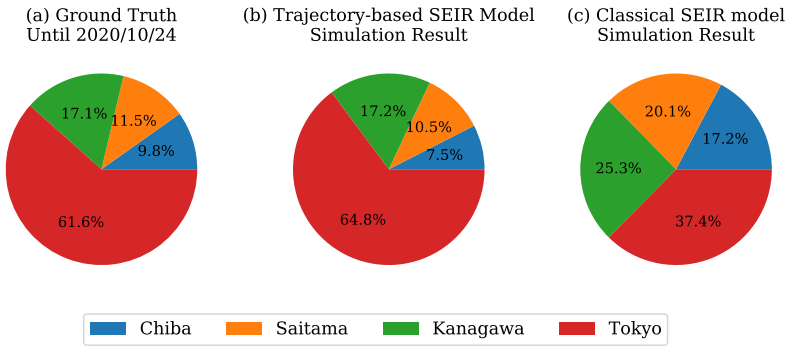

Fig. S1. Distribution of residential address of infected people in the Greater Tokyo Area. The size of each sector on the pie chart shows the proportion of infections in each prefecture to the total number of infections in the Greater Tokyo Area. (a) Distribution of the home addresses of persons who were actually reported to be infected until 2020/10/24. (b)&(c) are the simulated distribution of the home addresses of infected persons generated by trajectory-based SEIR simulation and the classical SEIR model, separately.

B. Model Sensitivity. As shown in Fig. S2, using the cumulative number of infected persons as measurement, fixing other parameters, and varying a particular parameter, we determined the sensitivity of our model. The experimental dataset consists of the one-month (July 2012) trajectories of 30,000 users, who were randomly selected from our dataset. Fig. S2(a) displays the sensitivity of β_{global}. As it becomes larger, the number of infections increases dramatically. We can conclude that compared to other parameters, β_{global} has the most significant impact on the results of the model, which is consistent with the judgments of our infectious disease experts; Fig. S2(b) displays the effect of σ on the cumulative number of infections. The results show that the shorter the incubation period, the more massive the infection scale. The result indicates that if a disease has a short incubation period, one person can become infectious faster, then spread it to others; Fig. S2(c) illustrates the impact of the recovery/death rate γ. It is clear that as γ increases, the cumulative number of infections declines rapidly. We can infer from this that the faster infectivity is eliminated, the lesser the risk of the spread expanding; Fig. S2(d) shows the impact of the number of initially infected persons (I_0) on the spread of the disease. The greater the number of initially infected persons, the larger the cumulative infected scale of the persons will be. Our experts considered the experimental results in Fig. S2 to be consistent with their professional judgment.

C. Computational Performance. The test environment is Supermicro Server with two Intel Xeon E5-2690 v4 Processors and 128GB memory, running Ubuntu 18.04.4. Here we show the relationship between the number of trajectories and the single simulation time (Fig .S3). The simulation period is one month, covering 8640 time intervals. We can see that the average single simulation time is 11.5 min, which is an acceptable balance between granularity and speed.

D. Data Dependency. Our model is driven by trajectory data. If the trajectory data is very biased and can not reflect the movement pattern of the city, then the results may not guarantee. To explore this, we repeated all the simulation experiments that appeared in the Case Study Section with 5000 (Fig. S5) and 50,000 (Fig. S6) people, respectively. We can find that the larger

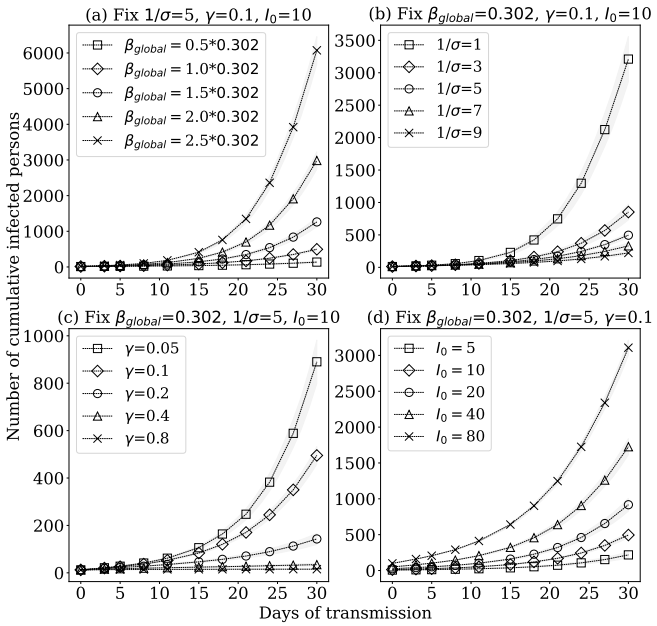

Fig. S2. Sensitivity charts of epidemic parameters, including the Sensitivity of (a) the number of adequate contacts per unit time–β_{global}, (b) the incubation rate–σ, (c) recovery/death rate–γ, (d) the initial number of infected persons–I_0. The horizontal- and vertical-axes indicate the number of days included in the transmission simulation, and the cumulative number of infections, respectively.

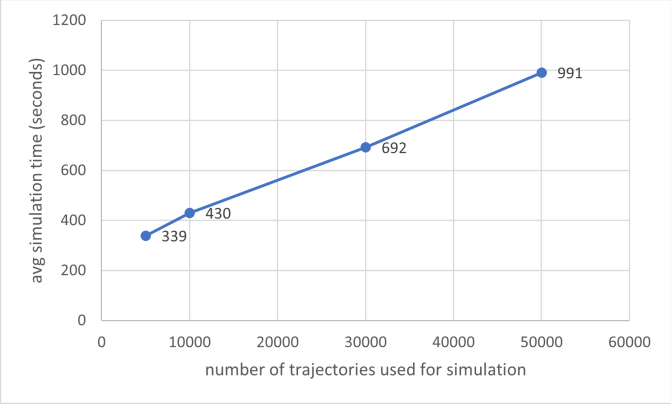

Fig. S3. The line chart of average single simulation time versus number of trajectories

.

amount of data reduces the uncertainty of the results, compared to that of 30,000 people (Fig. S4). Similarly, the reduced amount of data increases the uncertainty. Nevertheless, reducing or increasing the amount of data does not affect the policy conclusions in the Case Study Section. The minimum amount of data depends on the quality of the trajectory, and currently, for our data set, 5000 people is acceptable. Hence, to determine the appropriate data volume, the user needs to implement a series of simulations with the same parameters to observe the stability of the result.

E. Comments from Experts. Our model reproduced the distribution of people infected by the prefecture and discovered several potential infection hotspots. For this, EA commented: *"The result of the no policy simulation was impressive. Not just for COVID-19, it will be very useful for future epidemic condition"* EC shared the view of EA, saying that *"it is beneficial for allocating our limited medical resources and helping to develop local emergency plans"*. All the experts considered the trajectory-based fine-grained epidemic simulation to be promising.

Additional Materials for System Usability

We additional invited seven public health experts, mainly from disease control departments and universities, to evaluate the system's usability. The measuring method is System Usability Scale (SUS). It is a 10-item questionnaire with five response options for respondents to choose from; ranging from strongly agree to disagree strongly, corresponding to a score from 5 to 1, where Q1, Q3, Q5, Q7, Q9 are positive questioning tones, and Q2, Q4, Q6, Q8, Q10 are negative. After the questionnaire

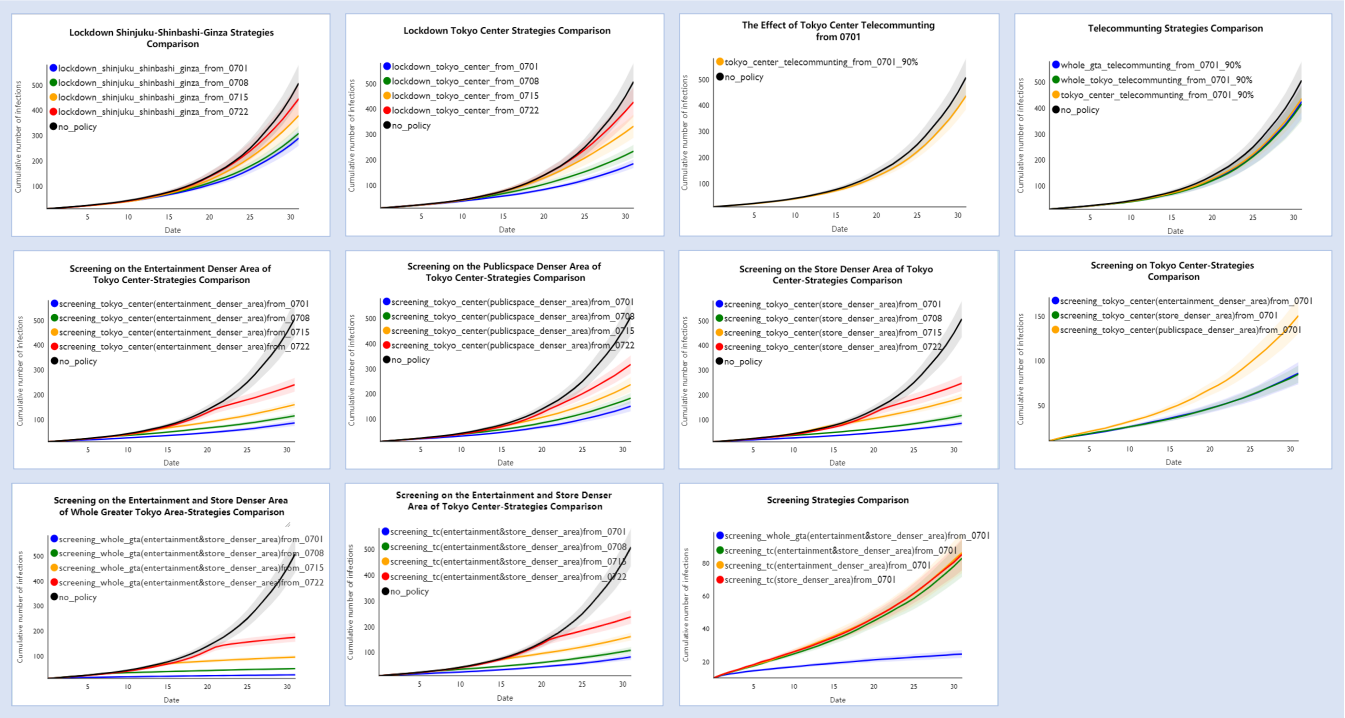

Fig. S4. The simulation and comparison results under a series of policies, which the number of trajectories is 30000 (Extracted from the Fig. 9 of the paper body)

.

results were collected, the score was calculated in the manner described in (3). The result shows that the overall average SUS score of EpiMob is 77.14 out of 100, illustrating a positive response from the participants. In addition, the distribution of scores of each question is plotted by boxplot, as shown in Fig. S7. It can be found that respondents diverge more on Q4 and Q10 (learnability) and score lower on Q2 (redundancy), which suggests possible optimization directions for our system.

References

1. L Pappalardo, F Simini, G Barlacchi, R Pellungrini, scikit-mobility: a python library for the analysis, generation and risk assessment of mobility data (2019).
2. NHK, Japan covid-19 prefectures daily data (https://www3.nhk.or.jp/n-data/opendata/coronavirus/nhk_news_covid19_prefectures_daily_data.csv) (2020) Accessed: 2021-10-22.
3. J Sauro, Measuring usability with the system usability scale (sus) (https://measuringu.com/sus/) (2011) Accessed: 2022-02-26.

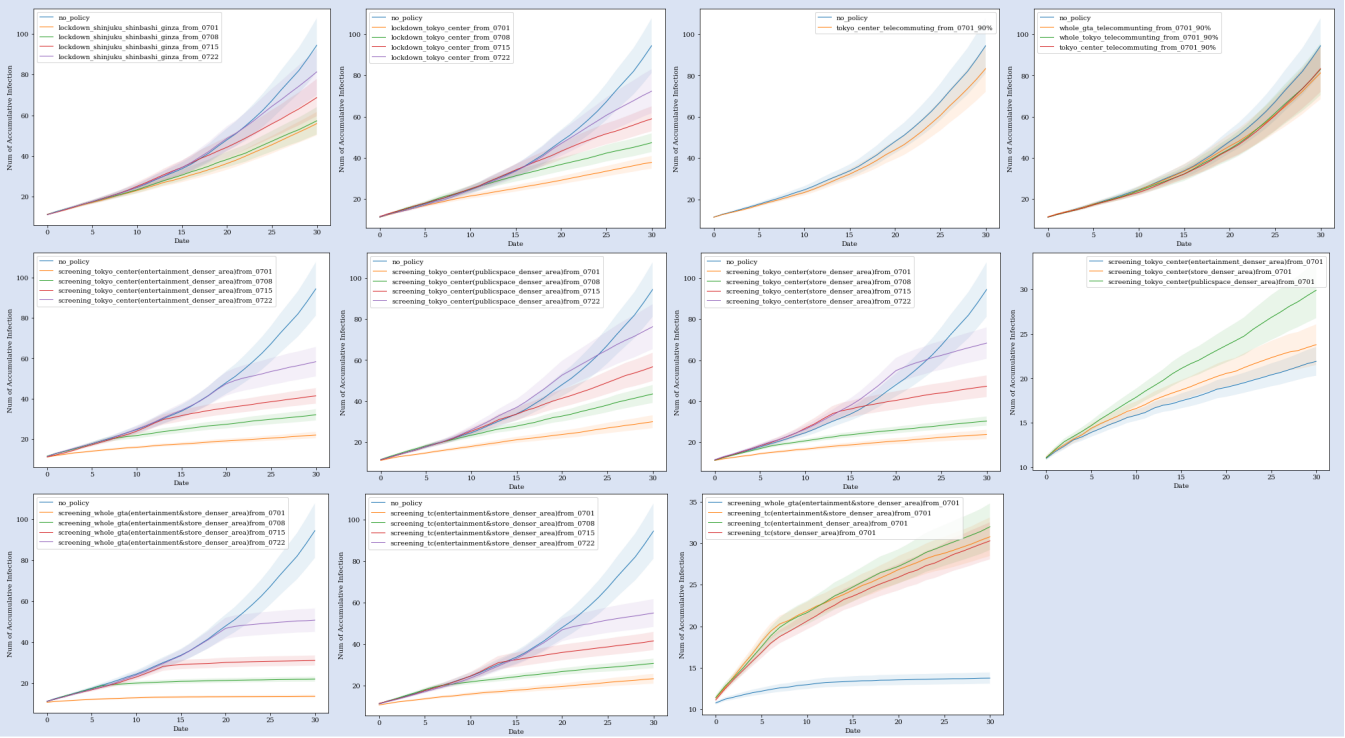

Fig. S5. The simulation and comparison results under a series of policies, which the number of trajectories is 5000.

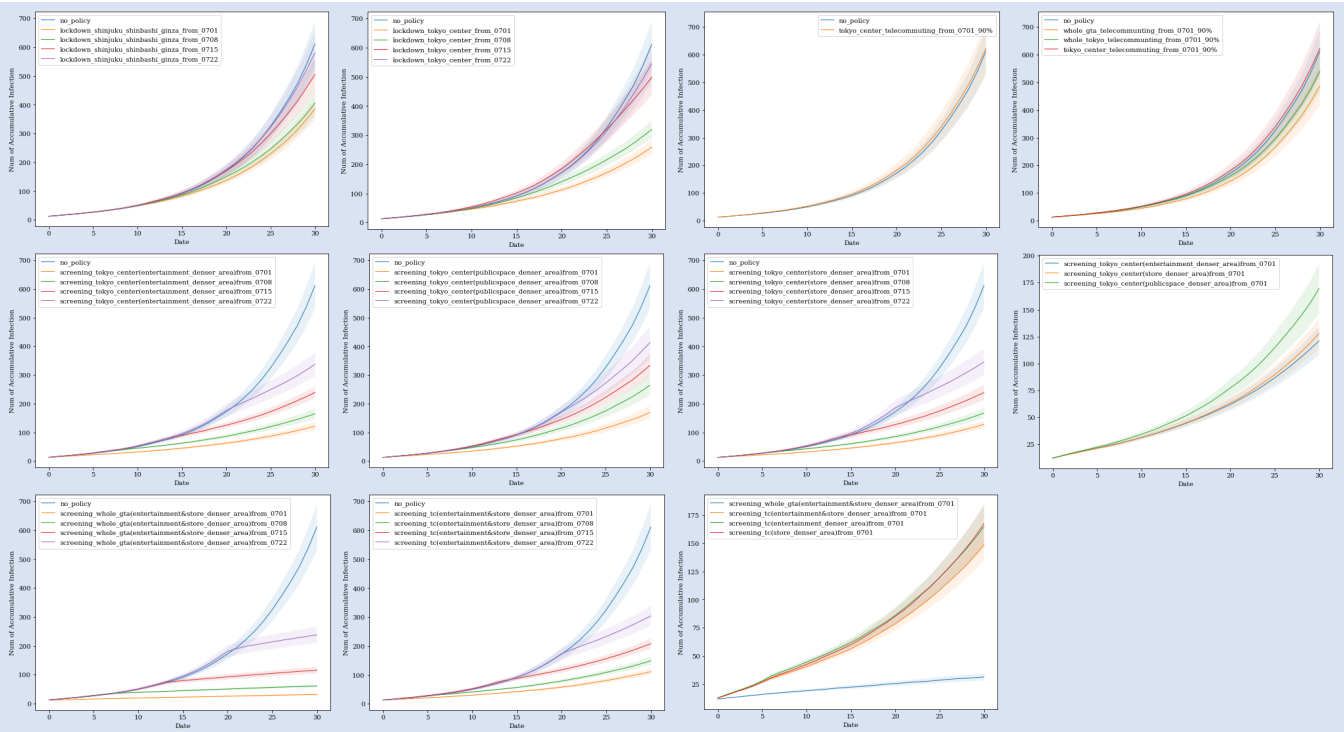

Fig. S6. The simulation and comparison results under a series of policies, which the number of trajectories is 50000.

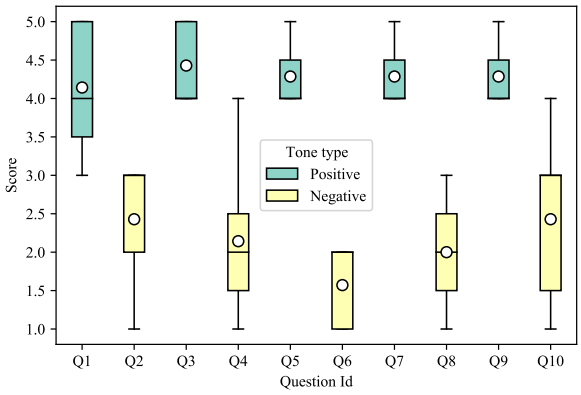

Fig. S7. The visualization of SUS based user ratings for EpiMob, by boxplot.