Quarantine in Motion: A Graph Learning and Multi-Agent Reinforcement Learning Framework to Reduce Disease Transmission Without Lockdown

Sofia Hurtado
Dept. of Electrical Computer Engineering
The University of Texas at Austin
Austin, TX, 78712, USA
slhurtad@utexas.edu

Radu Marculescu
Dept. of Electrical Computer Engineering
The University of Texas at Austin
Austin, TX, 78712, USA
radum@utexas.edu

***Abstract*—Exposure notification applications are designed to help trace disease spreading by alerting exposed individuals to get tested. However, false alarms can cause users to become hesitant to respond, making the applications ineffective. To address the shortcomings of slow manual contact tracing, costly lockdowns, and unreliable exposure notification applications, better disease mitigation strategies are needed. In this paper, we propose a new disease mitigation *paradigm* where people can reduce infection spreading while maintaining *some* mobility (i.e., Quarantine in Motion). Our approach utilizes Graph Neural Networks (GNNs) to predict disease hotspots such as restaurants, shops and parks, and Multi-Agent Reinforcement Learning (MARL) to collaboratively manage human mobility to reduce disease transmission. As proof of concept, we simulate an infection using real-world mobility data from New York City (over 200,000 devices) and Austin (over 36,000 devices) and train 10,000 agents from each city to manage disease dynamics. Through simulation, we show that a trained population suppresses their reproduction rate below 1, thereby mitigating the outbreak.**

***Index Terms*—Graph Neural Networks, Multi-Agent Reinforcement Learning, Epidemics, COVID-19**

I. Introduction

While large populations, bustling commerce, and inter-regional travel are hallmarks of a modern society's success, these factors also create a favorable environment for spreading infectious diseases [1]. Considering the vulnerabilities of big cities to disease outbreaks, we ask whether we can train agents to optimize their visits at various points of interest (POI), (e.g., restaurants, gyms, parks, etc.) to lessen disease spreading.

The intersection of epidemics, model forecasting, and disease mitigation has been successful at testing non-pharmaceutical interventions at the macro-level (regions, countries, cities) [2]. However, with access to Foursquare mobility data [3] and machine learning techniques, we can now relay the knowledge of disease forecasting back to the individual—in other words, we can provide *actionable* risk analysis.

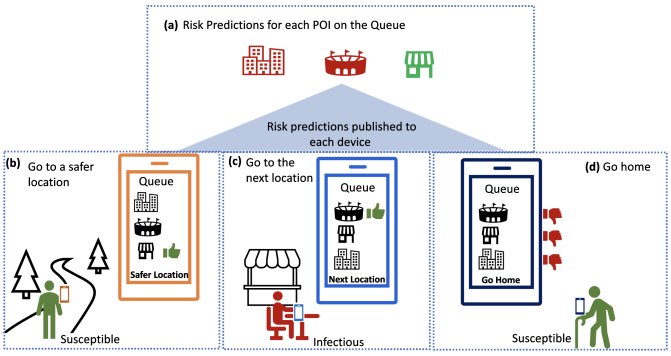

Fig. 1. Quarantine in Motion, i.e., application that collaboratively optimizes human mobility to mitigate disease spreading. (a) When given the user's destination queue, each device has access to the predicted transmission risk for each POI. For example, (b) *Susceptible* person is suggested to avoid the disease hotpots on its destination queue, and instead go to a safer location. (c) *Infectious* person is directed to go to their next intended location. (d) Older *Susceptible* individual is suggested to stay at home instead, likely because their immunity is too low to risk exposure.

In fact, we envision a new disease-aware mobility application where users self-report their disease status (i.e, *Susceptible*, *Exposed*, *Infectious*, *Recovered*) and input their planned visits for the day. The application utilizes reinforcement learning to suggest visits with respect to the user's willingness to cooperate and immunity to the virus (e.g., vaccination status, health factors, mask compliance, etc.). Each agent (i.e., smart device) takes into account the predicted infectious densities at each POI and suggests the user to go to a safer location, visit the next location on their queue, or return home (Fig. 1).

To this end, our contributions are as follows:

1) We propose a GNN node regression problem that performs highly granular (i.e., hourly) risk predictions at various POIs using real-world mobility data.
2) We present a novel MARL disease mitigation framework that can handle thousands of agents.

ASONAM '23, November 6-9, 2023, Kusadasi, Turkey

ACM ISBN 10.1145/3625007.3627727…$15.00
http://dx.doi.org/10.1145/3625007.3627727

3) Our experimental findings demonstrate that the agents successfully mitigate disease spreading across scales in two major cities, i.e., Austin and New York City (NYC).

Taken together, our contributions can help move exposure notification applications from *reactive* to *preemptive* risk management tools. The remainder of this paper is organized as follows: Section II discusses prior work, Section III describes our approach, Section IV presents our experimental results. Finally, Section V summarizes our contributions.

II. Prior Work

In this section we present prior work in disease mitigation and MARL.

A. Disease Mitigation

The majority of work in epidemics such as differential equations, compartmental models, and network approaches has been done at the macro-scale (i.e., counties, countries, continents) for estimating disease spread [4] and underlying social dynamics [5]. Because of these readily available macro-scale epidemic approaches, in response to COVID-19, governments enacted regional lock-downs and travel bans to reduce population mixing. Though successful in reducing new cases, the cost of maintaining long term lock-downs led to pandemic fatigue [6] and thus proved to be an unsustainable strategy.

At the meso-scale, early in the COVID-19 pandemic, hospitals deployed a cohort model that sectioned off health care providers and patients to reduce population mixing [7]. Schools then followed suit by organizing student-teacher cohorts to reduce disease spreading [8]. If one cohort experiences an outbreak, the others can continue functioning without going into a full lockdown. In this paper, we propose pushing this cohort paradigm to highly dynamic systems (e.g., population in a city) by training RL agents to self-organize into mobility cohorts where we can incentivize *Infectious* people to frequent different locations from the *Susceptible* people.

As a means to manage economic and social costs, researchers apply RL to optimize disease mitigation mandates at the government level [9], [10]. In their work, the agent (i.e., government) manages a city while under a disease threat. Alternatively, Libin et. al. deploy single-agent RL to manage school shut downs as a means to reduce infection spreading at disease hubs like classrooms [11]. Though helpful in advising decision making at the macro-scale, we are rather interested in informing *distributed decisions* at the micro-scale (i.e., individual level) to ultimately mitigate disease spreading at the meso-scale. We envision an anti-fragile society whose individuals can continue their daily lives while collaboratively avoiding infection hot-spots. To this end, we investigate using MARL to mitigate spreading.

B. Multi-Agent Reinforcement Learning

MARL is a field within reinforcement learning that focuses on studying the interaction and coordination of multiple agents in complex environments. Unlike single-agent reinforcement learning, where a single agent learns to maximize its own rewards, MARL involves multiple agents learning and interacting with each other to achieve collective goals [12].

One of the key challenges in MARL is the dynamic nature of the environment. As agents learn and adapt, the environment can change as a result of the actions taken by other agents, leading to non-stationarity [13]. This creates a complex learning problem as agents must continuously adjust their strategies based on the actions and policies of other agents. In addition, as agents affect their environment and thereby affect other agent's learning policies, scalability remains challenging due to the compounding dependencies. While recent efforts have focused on tackling scalability [14]–[16], to the best of our knowledge, our work stands as the first *implementation* of MARL at a scale of thousands of agents.

We build on prior work in exposure notification applications, risk assessment, graph learning in epidemics, and disease mitigation by 1) implementing node regression to predict risk at various POIs and 2) proposing a new MARL framework that manages population mixing during an infectious outbreak.

III. Approach

We present a high level overview of our Approach in Fig. 2. We work with the Foursquare mobility dataset [3] that logs real visits at POIs on an hourly basis by compiling location tracking data from third party smart device applications. Because we do not have access to the health status of the anonymous individuals within the dataset, we fill in this gap by simulating a viral outbreak. We then train a GNN to predict infectious densities at various POIs through two metropolitan areas, namely Austin and NYC. To test our mitigation strategy, we load 10,000 agents with mobility decisions made by real people during the COVID-19 pandemic (May-August of 2020). The agents then learn to suggest visits for each user. Finally, we evaluate our mitigation strategy by comparing the final reproduction number of the mitigated population against the original simulated infection.

In this section we define our approach for network construction, graph learning set up, and RL problem formulation.

A. Network Construction

We construct the *network* as a composition of spatial and mobility graphs, $G = (G_s, G_f)$ where G_s is the *spatial* network and G_f is the *mobility* (i.e., foot traffic) network. We define the *spatial* network $G_s = (V, E_s)$ where V is a set of nodes that represent each POI, E_s is the set of edges that connect two POIs according to their physical proximity. To form the *spatial* edges E_s, we calculate the Haversine Distance [17] between each POI's latitude and longitude coordinates. Then for each POI, we connect their nearest neighbors. We define the *mobility* (foot traffic) network $G_f = (V, E_f)$ where V is the same set of POIs, and E_f connects two POIs when an individual visits both locations. We note that by utilizing these two types of edges, we can capture both the *spatial* and *mobility* relationships between POIs (Fig. 3).

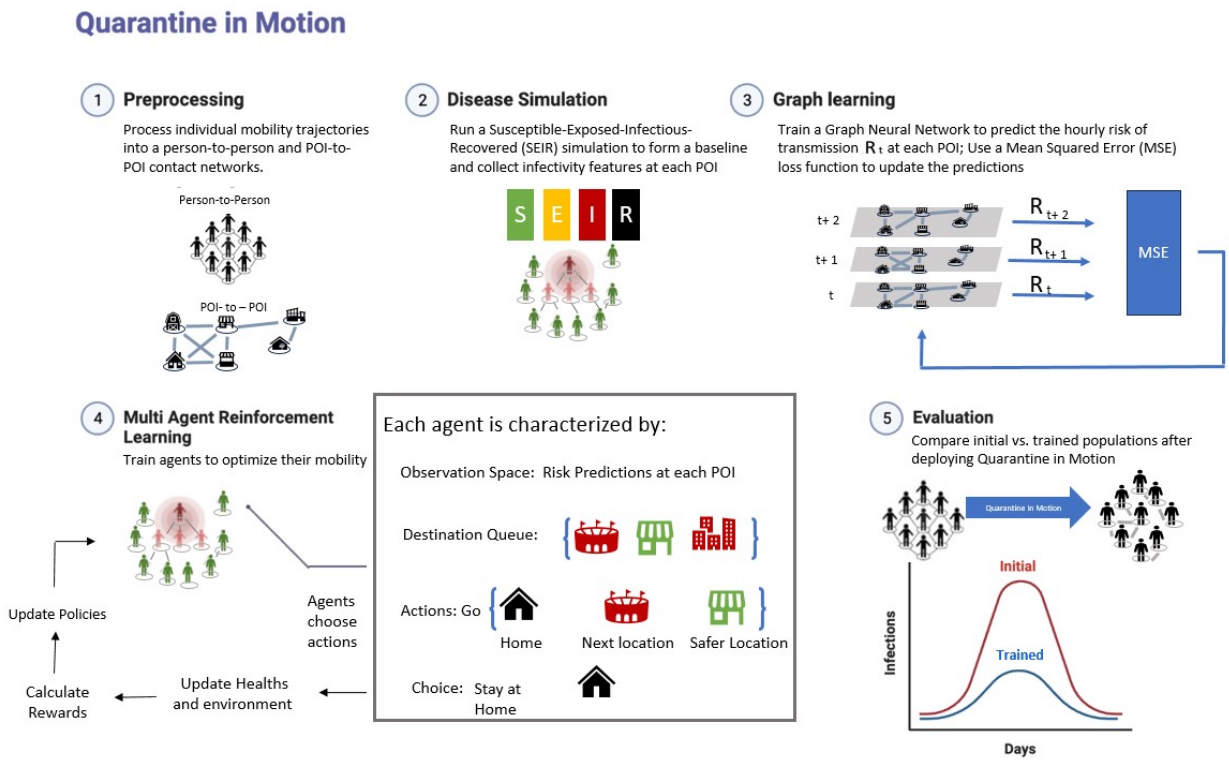

Fig. 2. A high level overview of the Quarantine in Motion: 1) the Foursquare mobility data gets processed into a POI-to-POI network to capture the spatial infectious spreading and a Person-to-Person network to keep track of individual infections. 2) We establish a baseline by simulating a disease on the untrained *initial* population. We collect features for each POI on the network on an hourly basis using three months of Foursquare data. 3) We then train a GNN to predict the risk of *transmission* for the following month and feed the predictions into the agents. 4) Each smart-device agent then learns to suggest which location to choose next on their destination queue, or alternatively go home. When all agents suggest their user's next action, the environment updates and records the latest health status of all users. The rewards are then calculated and relayed back to each agent to update their suggestion policies. 5) To evaluate our approach, we compare the new infections from the risk-informed (mitigated) population against the baseline (initial) to see if our approach reduces infection spreading.

B. Epidemic model

We apply the SEIR model [18] to individuals where an agent moves from the initial *Susceptible* state to the *Exposed* state when coming into contact with *Infectious* individuals. We then transition a *Susceptible* person to incubating when they visit a POI where the *Infectious* population density exceeds their immunity δ threshold. After an agent is incubating, they transition to being *Infectious* after the incubation period (5 days), and to *Recovered* state after an illness period (7 days). Note that the immunity threshold, incubation period, and illness period, are all tunable parameters that could be fit to simulate a different infectious disease.

We seed the outbreak by choosing 10% of the Foursquare population that have the most data points (hence are the most active) and initialize their health as *Infectious*. We define a POI's hourly *risk metric* as the ratio between the number of *Susceptible* people that catch the virus (and change to *Incubating*) after exposure to *Infectious* people at a POI.

C. Graph Learning Set Up

We utilize node regression to predict the hourly risk at each POI. We deploy neural network for each node in the graph that inputs the collected features, performs convolution across the neighborhoods, and then outputs the predicted risk value (Fig. 4). We utilize the deep graph learning library (DGL) [19] to implement the SAGE convolutional layers. The SAGE algorithm utilizes message passing along edges to aggregate (in our case, average) feature weights [20].

We add a sigmoid layer to predict the exposure risk Y_i per each node i (POI) between 0 and 1, where 1 means 100% of *Susceptible* people will transition to *Incubating* in the next hour following a visit to node i. We define the input features *per node per hour* $X_t = [I_t, S_t, \delta_t, \rho_t, \eta_t]$ as the number of *Infectious* people I_t, number of *Susceptible* people S_t , number of people that transition from *Susceptible* to *Incubating* δ_t, the infectious density ρ_t, and the percent of total population η_t that the POI is responsible for infecting. Of note, these features are collected in the COVID-19 simulation using the SEIR model.

D. MARL Problem Formulation

We define the MARL problem as follows. The *environment* consists of the POIs within a city. Each *agent* is loaded with destination queues pulled from real people's data within the Foursquare visits dataset. At each time step (i.e., hour), the agents can choose from three actions, namely *'go to the next location on the queue'*, *'stay at home'*, or *'choose a safer*

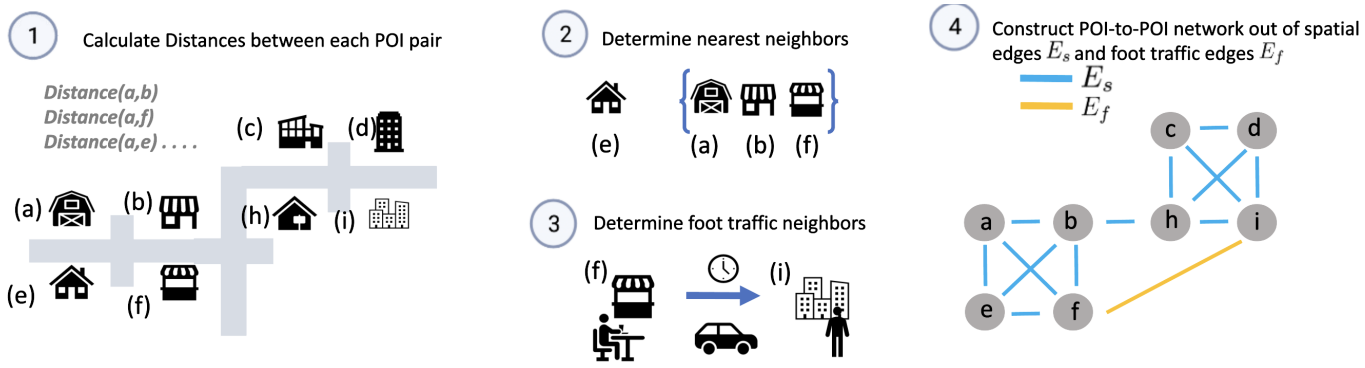

Fig. 3. First, we calculate the distances between each POI pair within the Foursquare Dataset and, second, determine the nearest neighbors to form the spatial edge E_s. In the third step, we count how many people flow between two POIs where a person dwells at the source POI and then travels and dwells at the next POI and consider this the foot-traffic edge E_f. Lastly, we abstract the POIs as nodes and connect them via the spatial edges and foot traffic edges.

location' (see Fig. 1). To account for data sparsity on the temporal axis, we assume that the users behind the smart devices are moving every hour and thus repopulate their destination queues when they run out of locations to visit.

We define the reward functions for each health status as a composition of sub-rewards $R_{exposure}$ (equation 1), $R_{fatigue}$ (equation 2), $R_{footprint}$ (equation 3) and R_{global} (equation 4):

$$R_{exposure} = 1 - \mid \delta - \frac{\#infections_{POI}}{\#people_{POI}} \mid \quad (1)$$

$$R_{fatigue} = 1 - \mid \alpha - \frac{\#deviations_{t:T}}{\#actions_{t:T}} \mid \quad (2)$$

$$R_{footprint} = \frac{1}{\#infectees_{t:T}} \quad (3)$$

$$R_{global} = \frac{1}{global\ infections_t} \quad (4)$$

The $R_{exposure} \in [0, 1]$ is meant to incentivize agents to reduce exposure to infectious people with respect to their user's immunity threshold $\delta \in [0, 1]$. For example, an agent for a *Susceptible* user with a higher immunity δ receives less of a penalty for suggesting POIs with more $\#infections_{POI}$ (number of infections at a POI) than an agent whose user has a lower immunity threshold. The $R_{fatigue} \in [0, 1]$ is meant to incentivize agents to respect their user's cooperation by suggesting to alter their user's intended behavior with respect to their fatigue parameter $\alpha \in [0, 1]$. For example, an agent whose user has a high pandemic fatigue α will be penalized for suggesting to deviate from their user's intended visits by 'staying at home' or 'going to safer location' more times than their threshold α allows. To keep track of the number of deviations, we calculate the cumulative $\#deviations$ and $\#actions$ from the beginning (i.e., t) to the end (i.e., T) of the episode. The $R_{footprint} \in [0, 1]$ is meant to penalize agents whenever their user infects other people by incriminating their number of $\#infectees$ every time their user is *Infectious* and visiting a POI that produces a new infection. Finally, $R_{global} \in [0, 1]$ is meant to motivate agents to suggest altering behavior if the population's $global\ infections_t$ are high at each time step t, even if their user is not getting exposed.

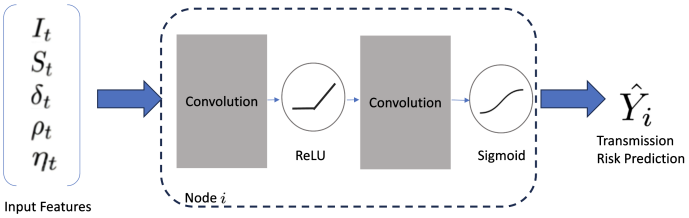

Fig. 4. The GNN inputs hourly node features $[I_t, S_t, \delta_t, \rho_t, \eta_t]$ as the number of *Infectious* people I_t , number of *Susceptible* people S_t , number of people that transition from *Susceptible* to incubating δ_t, the infectious density ρ_t, and the percent of total population η_t that the POI is responsible for infecting. The features go through two convolution layers, with a ReLU activation function in between. The output of these layers is then fed forward into a sigmoid activation function, which predicts the risk of transmission $\hat{Y}_i$ at each node i.

The rewards for each health status are defined in Table I. We incentivize *Susceptible* agents to take into account their risk of exposure at a POI with respect to their own willingness to change behavior for the social good. When *Infectious*, they no longer worry about being exposed, but instead, they keep track of the number of people they directly infect ($R_{footprint}$) to weigh against their respective pandemic fatigue α. The *Incubating* and *Recovered* reward functions are similar, as these agents do not worry about being exposed. Because each sub-reward (e.g., $R_{exposure}$) $\in [0, 1]$, the maximum reward for each agent is 1. As a starting point, we weigh each sub-reward equally and leave optimizing the weights for future work.

TABLE I
REWARD FUNCTIONS FOR EACH HEALTH STATUS

Health Status	Reward
Susceptible	$\frac{1}{3}R_{exposure} + \frac{1}{3}R_{fatigue} + \frac{1}{3}R_{global}$
Incubating	$\frac{1}{2}R_{fatigue} + \frac{1}{2}R_{global}$
Infectious	$\frac{1}{3}R_{fatigue} + \frac{1}{3}R_{footprint} + \frac{1}{3}R_{global}$
Recovered	$\frac{1}{2}R_{fatigue} + \frac{1}{2}R_{global}$

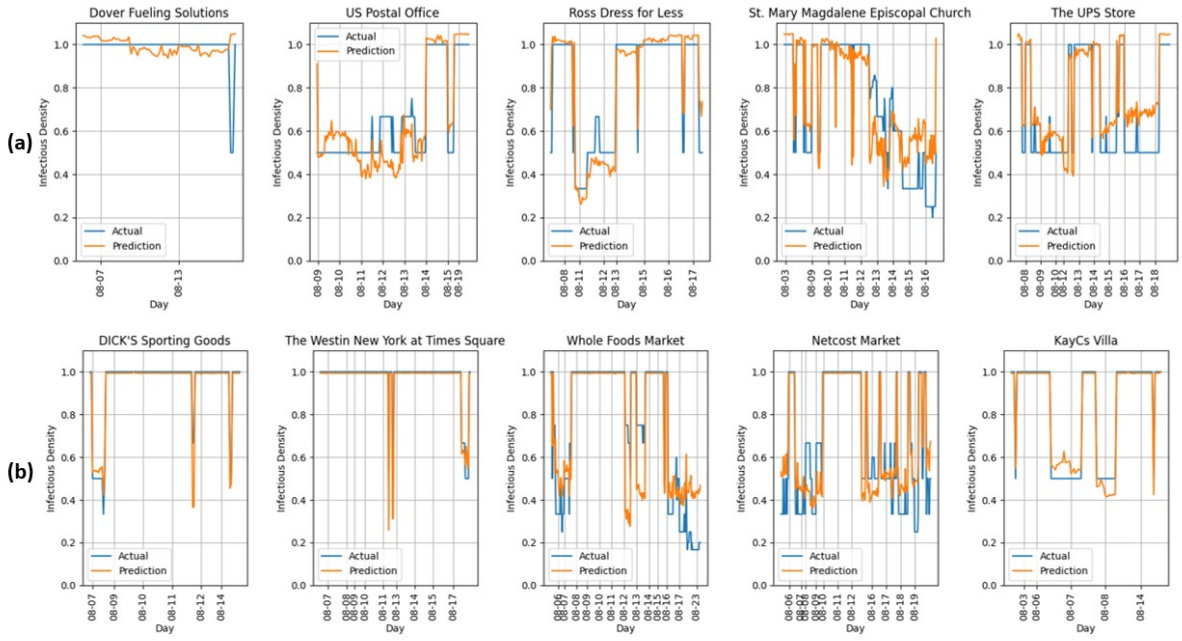

Fig. 5. Predicted vs Actual infectious density at POIs within (a) Austin and (b) NYC, at an hourly basis. Here we concatenate the results from when data exists at the POI (i.e., during business hours).

E. MARL Algorithm

We deploy the policy gradient REINFORCE [21], [22] algorithm with a value approximation baseline on each agent. In each episode, we seed a *Susceptible* population with the same 10% *Infectious* people that are considered the most active within the Foursquare dataset. At every timestep, each agent suggests their user's next action A_{t+1} based on their respective policy $\pi = p(A_{t+1}|S_t)$ given their current state S_t. We terminate the episode when the virus has no one left to infect and consider this time T.

Between each episode, for each agent, we accumulate the rewards R_n at each timestep from the beginning of an episode t to its termination T into an accumulative reward G reduced by a discount parameter γ (equation 5) to approximate the long term returned rewards. We then subtract the state value $\hat{v}(S_t)$, which represents the updated expected return, to use as a baseline (equation 6). We approximate the $\hat{v}(S_t)$ using a two layer neural network that updates the weights w where α denotes the learning rate and Δ denotes the gradient (equation 7). We then update the policy gradient θ (equation 8) by using another two layer neural network that outputs the probability of maximizing the reward for each action.

$$G \leftarrow \sum_{n=t+1}^{T} \gamma^{n-t-1} R_n \tag{5}$$

$$\delta \leftarrow G - \hat{v}(S_t, w) \tag{6}$$

$$w \leftarrow w + \alpha^w \delta \Delta \hat{v}(S_t, w) \tag{7}$$

$$\theta \leftarrow \theta + \alpha^\theta \gamma^t \delta \Delta \ln \pi(A_t|S_t, \theta) \tag{8}$$

Our approach can be summarized as follows: first, preprocess the Foursquare mobility dataset into a spatio-temporal network of POIs, then simulate an infectious outbreak on the population using SEIR to collect node features, then perform node regression to predict hourly risk of *transmission*. Finally, we cast disease mitigation as an MARL problem that allows for analyzing the collective behavior from many individual's decisions and performing epidemic analysis at the meso-scale.

IV. Results

In this section we present our experimental results for the graph learning predictions and disease mitigation.

A. Graph Learning

We utilize Mean Squared Error (MSE) as the evaluation metric for our graph learning model. For each city, we train the GNN to learn the risk of transmission at each POI over three months (May - July 2020) and test the predictions over the next month (August 2020). To get a more granular feel, we visualize the dynamic predictions at POIs shown in Fig. 5. To get a diverse representation of the data, we choose locations with various functions, for example, post office, supermarket, church, transportation center, hotel, department store, etc.

We see that the GNN can predict the dynamic infectious densities for POIs within Austin (5a) and NYC (5b). In principle, a user who was planning on going clothes shopping at Ross Dress for Less on August 8th can be informed of the predicted infectious density and choose to go a couple of days later, say on August 13th, instead. Alternatively, the user

Fig. 6. Spatio-temporal representation of the maximum MSE at POIs within Austin (a) and New York City (b). The yellow color indicates MSE of 40% whereas the purple indicates very low MSE.

could choose to go to a different clothing store with a less risky infectious density.

In Fig. 6, we then visualize the maximum MSE loss at each POI for the month of August 2020 in both Austin (6a) and NYC (6b). In both cities, we see that the maximum MSE loss stays under 40% for any given POI. A breakdown of the maximum loss at various POIs is shown in Table II.

TABLE II
MAXIMUM MSE AT VARIOUS POIS IN AUSTIN AND NYC.

City	MSE > 0.2	MSE > 0.25
Austin	9%	2%
New York City	7.6%	3.5%

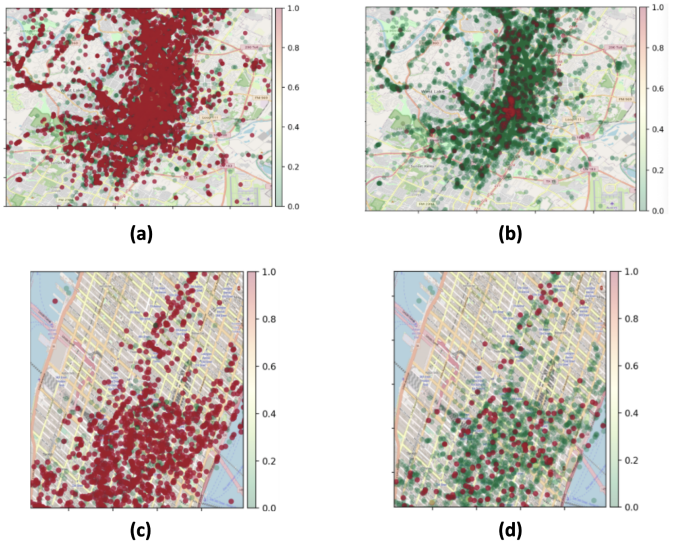

Fig. 7. Population mixing at various POIs for Austin (a,b) and NYC (c,d), where each dot represents a POI. The red dots show when contamination $C \approx 1$ while the green dots show C is closer to 0. For each plot, we run an SEIR simulation of 10,000 agents using: (a) Austin baseline, (b) Austin MARL mitigation, (c) NYC baseline and (d) NYC MARL mitigation. We find that Austin's MARL mitigation is able to reduce the number of POIs with contamination greater than 20% by $\approx 97\%$ where the baseline has 7,998 POIs and the MARL mitigation has 287 POIs. Similarly, NYC's MARL mitigation is able to reduce the number of POIs with contamination greater than 20% by $\approx 87\%$ where the baseline has 10,512 POIs and mitigation has 1,296 POIs.

B. Mitigation

Because we aim to keep the *Susceptible* and *Infectious* people separate from each other, we formulate the contamination metric C (equation 9). The I_{POI} represents the number of *Infectious* or incubating users within a POI, and N_{POI} represents the total people within the POI. We consider an infectious density of 50% percent to mean complete population mixing between infectious and non-infectious people at a POI ($C = 1$). Therefore, we divide the difference by 0.5 to make the contamination metric $C \in [0, 1]$.

$$C = 1 - \frac{\mid \frac{I_{POI}}{N_{POI}} - 0.5 \mid}{0.5} \tag{9}$$

In Fig. 7, we plot the maximum contamination C for Austin (7a, 7b) and NYC (7c, 7d). Each dot represents a POI and the dark red color indicates that C is close to 1, whereas, the green indicates low population mixing. We can see that the baseline for both Austin (7a) and NYC (7c) have more highly contaminated POIs than in the mitigated SEIR runs (7b, 7d). After confirming that the MARL mitigation technique results in a decrease in population mixing, we investigate its social feasibility by examining our approximated user satisfaction.

Because our MARL strategy comes in the context of a mobility suggestion application, we incentivize the agents to take into consideration their user's willingness to socially cooperate. We define *cooperation* as the number of *deviations* taken from a user's recorded destination queue. For the agent, cooperation means suggesting a user to 'go home' or to 'choose a safer location'. Each agent is aware of their user's social fatigue parameter (α), and are trained, in part, by the $R_{fatigue}$ (equation 2) reward function which penalizes the agent any time they suggest to deviate from their queue beyond the user's willingness. For this reason, we approximate a *user's satisfaction* as being the difference between accumulative suggested social cooperation (number of times their agent suggests to deviate) and the user's fatigue parameter α. In fact, we can draw an analogy of the agents ability to satisfy the user to the acceleration and deceleration in a car. Because a user's satisfaction is accumulative in nature, the agent tries to balance the user's actions by suggesting to socially cooperate (deviate from their path), or to defect (continue to their next intended location). Before training, we assign heterogeneous α values to the users on a Gaussian distribution with a mean of 70%. Then after training, we plot the agents over-cooperation (α +) or under-cooperation (α -) for every health group in Fig. 8. For example, if an agent suggests to cooperate 100% of the time, however their user's $\alpha = 0.7$, then the resulting cooperation vs time plot would show $\alpha + 0.3$ over-cooperation.

Fig. 8a shows the population of 1,000 untrained agents from Austin (at the 0th epoch) using a random policy to make their suggestions. Regardless of timestep, the population of agents bounce between over- and under -cooperation showing that the decision-making does not respect the user's willingness to cooperate α. However in contrast, 8b shows the cooperation in the last epoch of training. At the population level, *Susceptible* users are asked to over-cooperate more than the other health

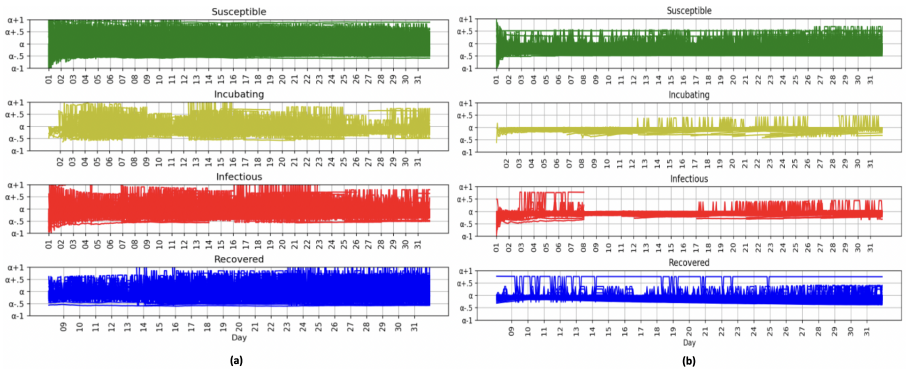

Fig. 8. Cooperation vs Time (days) for a population of 1,000 agents in Austin within the *Susceptible*, *Incubating*, *Infectious* and *Recovered* health categories. (a) displays the cooperation of untrained agents using a random policy to make cooperate vs defect suggestions, while (b) exhibits the agents after training. The trained agents suggest their users to cooperate within the user's willingness to cooperate parameter α.

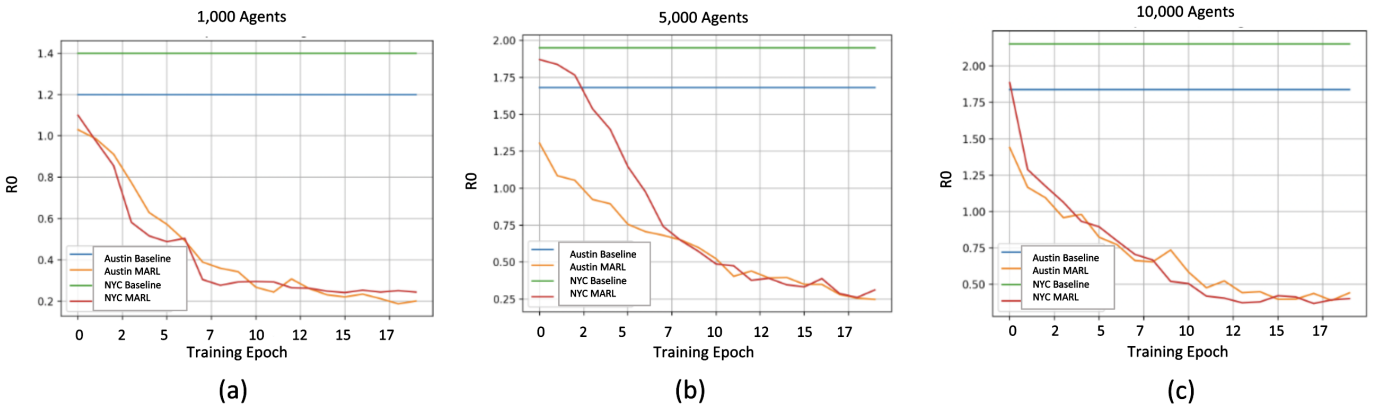

Fig. 9. R0 vs training epoch for Austin and NYC in a population of (a) 1,000 agents, (b) 5,000 agents, and (c) 10,000 agents. The agents are initialized to choose random actions in the 0th epoch and then learn in between each epoch for the remainder of training. One epoch represents an episode that spans one month of Foursquare data (August 2020). As shown in each plot above, the trained agents are able to push R0 below 1, meaning they can direct infectious traffic to effectively decrease the speed case reproduction and dampen spreading.

groups throughout the outbreak. However, *Incubating* and *Infectious* agents respect their user's α in periods of low contagion and then over-cooperate ($\alpha+$) when infections rise.

For each experiment, we use the reproduction number metric R0 [23] to evaluate the efficacy of the mitigation strategy. The reproduction number R0 is a measure used to describe the potential spread of an infectious disease. It represents the average number of people who will contract the disease from an infected person. For example, if the R0 is 2, then on average, each infected person will transmit the disease to two others. If R0 is below 1, it means that the disease is slowing down and the outbreak is likely to be contained. If the R0 is above 1, then the disease spreading is still on the rise. We compare the experimental R0 values for Austin and NYC to their respective baseline R0 values, in which agents follow their users' destination queue without making any deviations.

The agents update their policies and value approximations between each epoch and use their updated policies to suggest which location to visit (or whether to go home) in the next epoch. Over 20 epochs, we see a significant reduction of R0 in each experiment compared to the baseline.

In the case of of 1,000 agents (9a) and 5,000 agents (9b) are able to reduce their R0 to less than $R0 = 0.3$ in both Austin and NYC. This can be interpreted as both cities were able to decrease spreading such that three infections would lead to only one new infection. To test the feasibility at a larger scale, we run our mitigation experiment on 10,000 agents. We are interested to see if population mixing reduction would max out at some population density, however, even in the case of 10,000 agents (shown in 9c) we get a clear reduction of R0 to less than 0.5 for both Austin and NYC. These results suggest that MARL is able to manage a city's mobility to mitigate a disease without a complete lockdown.

TABLE III
CONTACT NETWORK PROPERTIES BEFORE AND AFTER TRAINING

Population	Nodes	Edges	Avg Degree	CC	APL
Initial	1000	34,838	69	0.14	2
Trained	1000	6,336	13	0.04	3

We then analyze the network properties from the original and mitigated contact networks (Table III). We build the contact networks by connecting two nodes (people) when they co-locate at a POI within the same hour. We use 1,000 people

from Austin during August 2020 for the original Foursquare co-location network and build another network after the MARL mitigation training. We find that though the number of nodes stays the same, the edges decrease significantly resulting in a smaller average clustering coefficient (CC) and a higher average path length (APL). By dismantling the small world effect, the trained agents make disease spreading harder.

V. Conclusion

In conclusion, we have presented a smart phone recommendation system that advises people on how to optimize their mobility during an disease outbreak. To this end, we have trained a GNN on Foursquare mobility data from Austin and NYC during May-July 2020 to predict risk of transmission for the following month of August. Finally, we have provided a disease mitigation framework and proposed a location suggestion application that is backed by MARL. We have shown that a trained population of 1,000, 5,000, and 10,000 agents effectively reduce the disease reproduction number (R0) below 1, while maintaining some mobility.

Our work is limited by the lack of ground truth health labels that would otherwise be self-reported by app users, therefore we have to rely on disease spreading simulations. Furthermore, scalability remains a problem when pushing this centralized MARL framework to the hundreds of thousands of agents due to the large computational complexity. However, we intend for this framework to become decentralized when pushed to edge devices; thus we leave this for future work.

VI. Acknowledgements

This work is supported, in part, by NSF grant CCF 2107085.

References

[1] S. Lai and J. Huang, "Why large cities are more vulnerable to the covid-19 pandemic," *Journal of Urban Management*, vol. 1, no. 11, pp. 1–5, 2022.

[2] J. Brauner, S. Mindermann, M. Sharma, D. Johnston, J. Salvatier, and T. Gavenčiak, "Inferring the effectiveness of government interventions against covid-19," *Science*, vol. 371, no. 6531, pp. 1–5, 2020.

[3] *Foursquare visits dataset*, https : / / foursquare . com / products/visits/.

[4] D. Kluger, Y. Aizenbud, A. Jaffe, *et al.*, "Impact of healthcare worker shift scheduling on workforce preservation during the covid-19 pandemic," *Infection Control and Hospital Epidemiology*, 2020.

[5] S. Kaiser, A. Watson, B. Dogan, *et al.*, "Preventing covid-19 transmission in education settings," *Pediatrics*, 2021.

[6] A. Franzen and F. Woehner, "Fatigue during the covid-19 pandemic: Evidence of social distancing adherence from a panel study of young adults in switzerland," *PLOS ONE*, 2021.

[7] D. Wei, Z. Fang, P. Zhang, G. Guo, and Q. Xiaogang, "Mathematical and computational approaches to epidemic modeling: A comprehensive review," *Frontiers of Computer Science*, 2015.

[8] A. Glaubitz and F. Fu, "Oscillatory dynamics in the dilemma of social distancing," *Proceedings of the Royal Society A: Mathematical Physical and Engineering Sciences*, 2020.

[9] V. Kompella, R. Capobianco, S. Jong, *et al.*, "Reinforcement learning for optimization of covid-19 mitigation policies," in *AAAI Fall Symposium on AI for Social Good*, 2020.

[10] S. Bushaj, Y. Xuecheng, A. Beqiri, D. Andrews, and E. Buyuktahtakin, "A simulation-deep reinforcement learning (sirl) approach for epidemic control optimization," *Annals of Operations Research*, 2020.

[11] P. Libin, A. Moonens, T. Verstraeten, *et al.*, "Deep reinforcement learning for large-scale epidemic control," in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track.*, 2020.

[12] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artificial Intelligence Review*, vol. 55, pp. 895–943, 2022.

[13] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. de Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," *CoRR*, vol. abs/1707.09183, 2017.

[14] F. Charbonnier, T. Morstyn, and M. McCulloch, "Scalable multi-agent reinforcement learning for distributed control of residential energy flexibility," *Applied Energy*, 2022.

[15] G. Qu, Y. Lin, A. Wierman, and N. Li, "Scalable multi-agent reinforcement learning for networked systems with average reward," in *Proc. Neural Information Processing Systems*, 2020.

[16] R. Zohar, S. Mannor, and G. Tennenholtz, "Locality matters: A scalable value decomposition approach for cooperative multi-agent reinforcement learning," in *Proc. Association for the Advancement in Artificial Intelligence*, 2022.

[17] C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, 1957.

[18] S. He, Y. Peng, and K. Sun, "Seir modeling of the covid-19 and its dynamics," *Nonlinear Dynamics*, 2020.

[19] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, and X. Song, "Deep graph library: A graph-centric, highly performant package for graph neural networks," *arXiv*, vol. 1909.01315, 2020.

[20] W. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Advances in Neural Information Processing Systems*, Dec. 2017.

[21] R. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, 1992.

[22] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2018.

[23] P. Delamater, E. Street, T. Leslie, T. Yang, and K. Jacobsen, "Complexity of the basic reproduction number (R_0)," *Emerging Infectious Disease*, 2019.