Mini-project 1: Deep Q-learning for Epidemic Mitigation

Yiyang Feng, Jiayi Sun

1 Introduction

Question 1.a) study the behavior of the model when epidemics are unmitigated

These plots are shown in Figure 1. As time goes by, the number of infected people initially increases, then fluctuates slightly before decreasing to 0. The number of deaths initially increases and then remains unchanged. The number of exposed individuals initially fluctuates and then decreases to 0. The number of susceptible individuals initially decreases and then slowly increases. The number of recovered individuals initially increases and then slowly decreases.

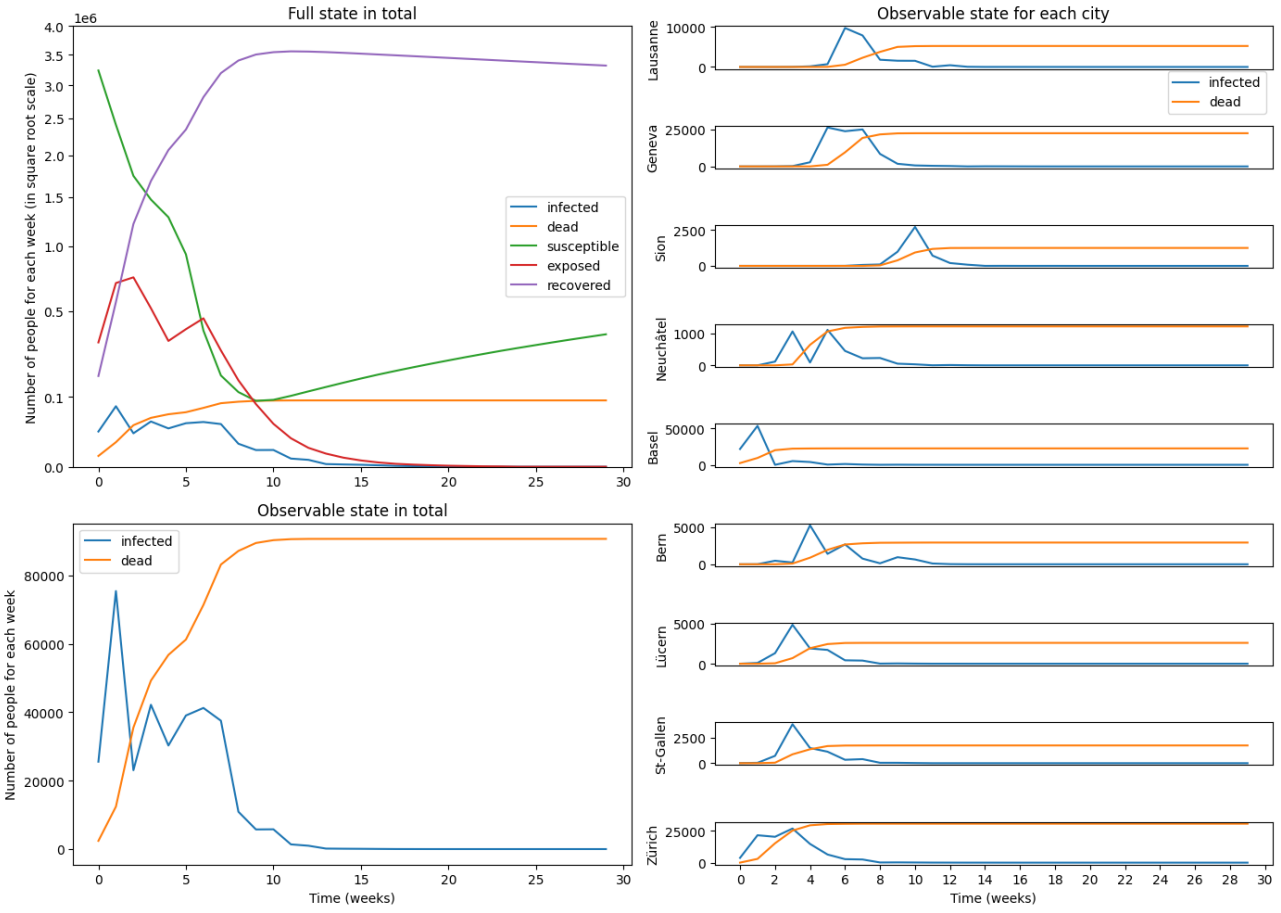

Figure 1: A plot of full states and observational states in total, and observable states for each city.

2 Professor Russo's Policy

Question 2.a) Implement Pr. Russo's Policy

These plots are shown in Figure 2. The number of deaths still increases initially and remains unchanged, but it is lower than in the unmitigated scenario. At the beginning of the confinement, the overall and some city-specific numbers of infected and exposed individuals decrease, while the number of susceptible individuals slightly decreases and the number of recovered individuals slightly increases. After the confinement ends, the overall and some city-specific numbers of infected and exposed individuals increase, the number of susceptible individuals rapidly decreases, and the number of recovered individuals rapidly increases.

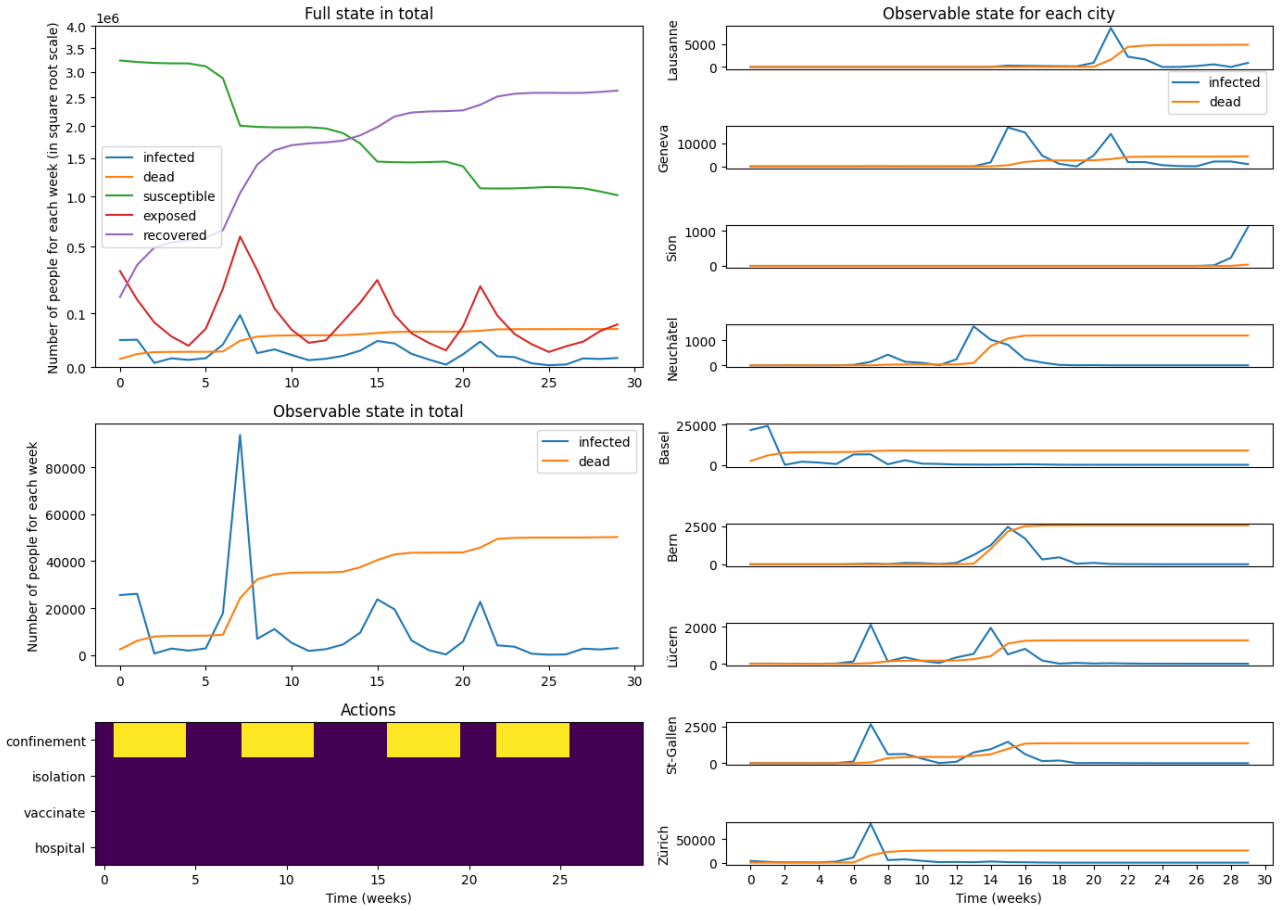

Figure 2: A plot of full states and observational states in total, observable states for each city, and the action sequence for the agent.

Question 2.b) Evaluate Pr. Russo's Policy

For each episode, we plot a distribution of **number of total confined days**, the **cumulative reward**, and **number of total deaths** in histograms. These plots are shown in Figure 3.

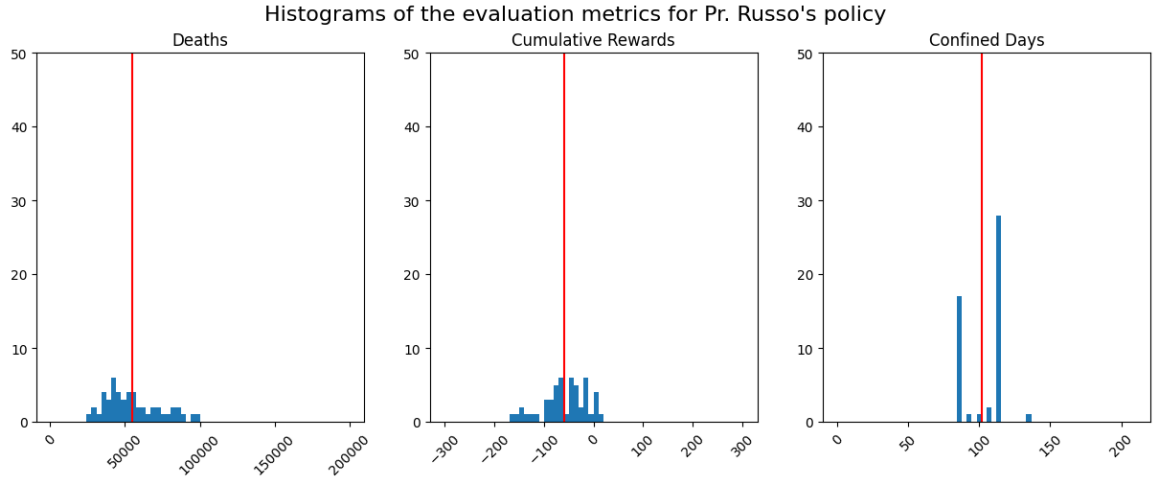

Figure 3: Distribution of the number of total confined days, the cumulative reward, and total deaths in histograms for Professor Russo's policy.

3 A Deep Q-learning approach

Question 3.a) implementing Deep Q-Learning

We use all configurations from the suggestion. The rewards are shown in Figure 4. We save our trained weights to the folder `models/agent_3a_{t}.pt`, where t is the training time (0, 1, or 2). We use the best model `models/agent_3a_{0}.pt` to plot the same episode in question 2.a) in Figure 5.

Figure 4: Training and evaluation traces for DQN without decreasing exploration.

We can see the agent learns a meaningful policy. Figure 4 shows that the average evaluation reward increases and is above zero finally, better than Russo's policy. Figure 5 indicates that the agent tends to do a longer confinement period to control the pandemic. When the number of infected increases suddenly, the agent will do the confinement. When the number drops and remains for about over 6 weeks, the agent will lift lockdown restrictions.

Question 3.b) decreasing exploration

Figure 6 shows the traces. We can see DQN with decreasing exploration gives better results because the training and evaluation rewards are higher than DQN without decreasing exploration. As the nature of neural networks, the update step should be large at the beginning but small at the end. Therefore, a decreasing exploration is better. Decreasing exploration over time in DQN strikes a balance between exploration and exploitation. It allows the agent to explore the environment effectively in the early stages of learning while gradually shifting towards exploiting its learned knowledge for better decision-making as training progresses.

Question 3.c) evaluate the best performing policy against Pr. Russo's policy

We run the best performing policy π^*_{DQN} with decreasing exploration through the evaluation code from question 2.b), generate the same histogram plots, and compare the results in Figure 7. The cumulative reward is higher and the number of death is lower. Therefore, the policy is better than Russo's one.

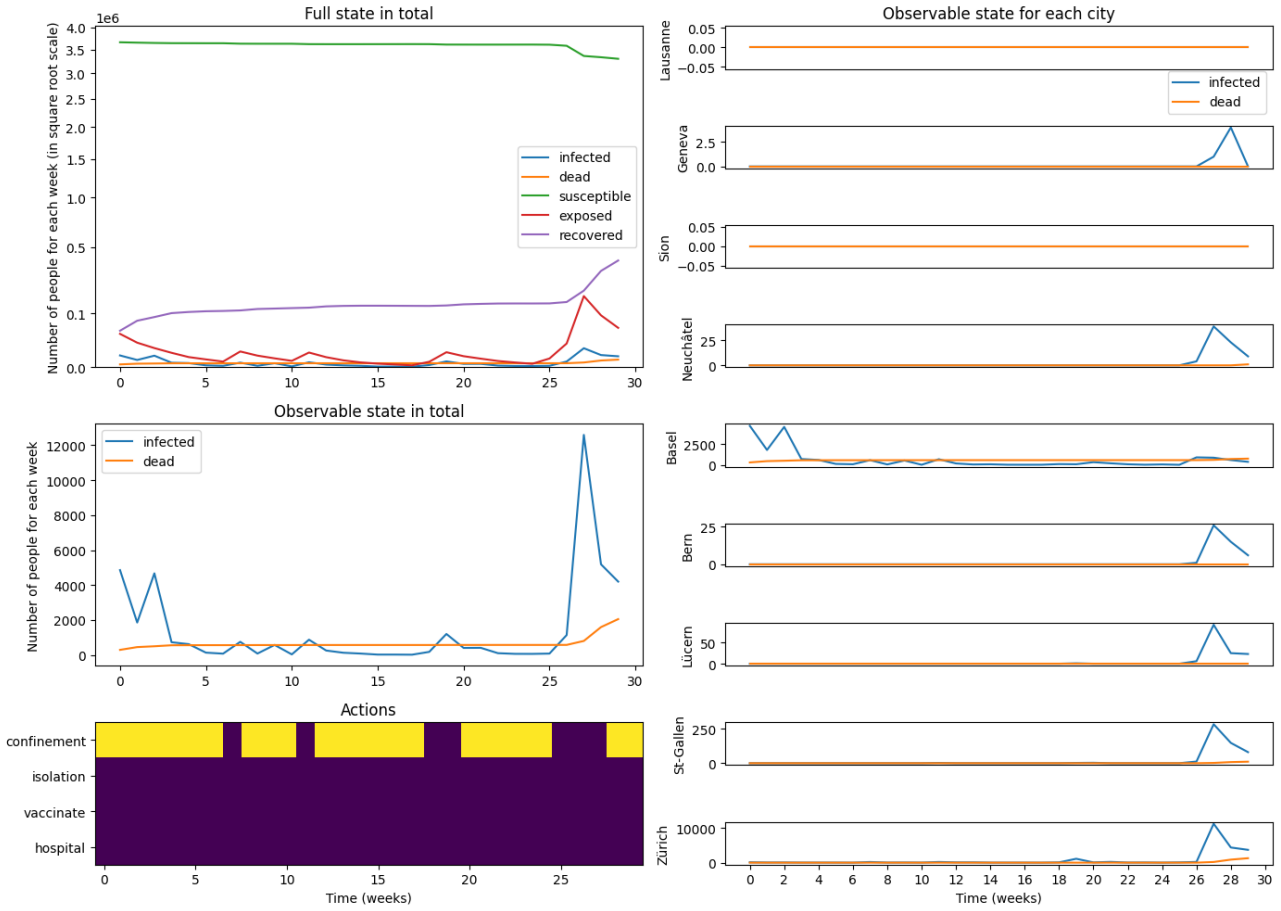

Figure 5: Training and evaluation traces for DQN without decreasing exploration.

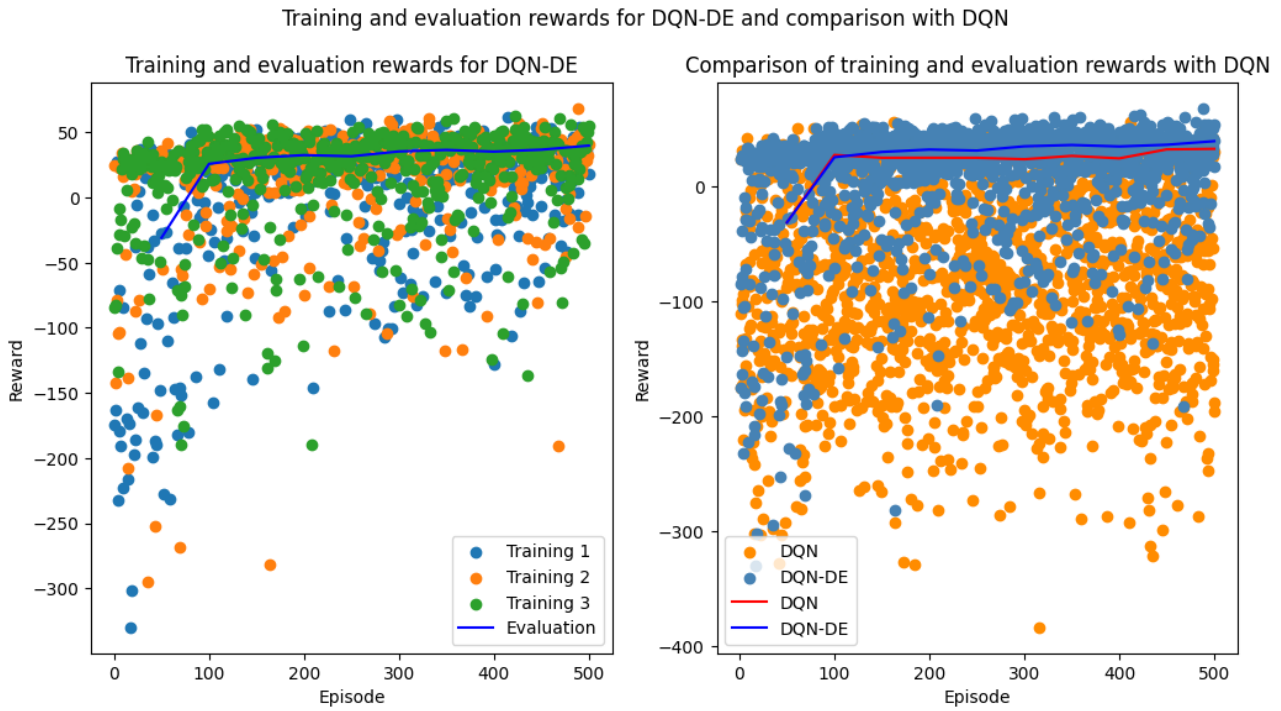

Figure 6: Training and evaluation rewards for DQN-DE (with decreasing epsilon) and comparison with DQN.

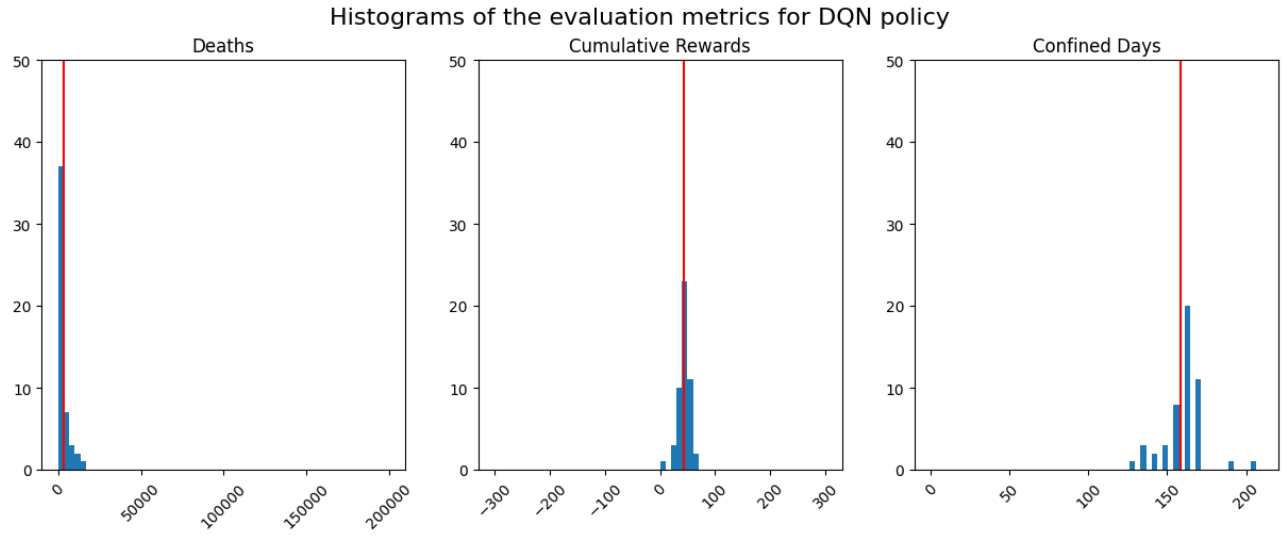

Figure 7: Distribution of the number of total confined days, the cumulative reward, and total deaths in histograms for the best DQN policy (with decreasing exploration).

4 Dealing with a more complex action Space

4.1 Toggle-action-space multi-action agent

Question 4.1.a) (Theory) Action space design

- Impact on **network architecture**: The introduction of a toggle-action space streamlines the network architecture by decreasing the number of output nodes from eight to five. This reduction subsequently lessens the number of parameters w, b within the neural network. Rather than requiring separate output nodes for each action, the network is now designed to output a singular value that signifies the Q-value of the selected action at any given time.
- Impact on **training**: The toggle-action space simplifies the network's output structure and the definition of Q-values, making training and inference more straightforward. During training, the simplified network reduces the computational complexity. In the inference phase, the network can easily select the action with the highest Q-value based on the single output value. Also, by focusing on a single action, the network's learning can achieve smooth transition in action decision and accelerate the convergence of the learning process, as the network directly learns to maximize the Q-value of the chosen action, rather than distributing its learning across all actions.

Question 4.1.b) Toggle-action-space multi-action policy training

We incorporate all configurations as suggested. The corresponding rewards are depicted in Figure 8. Our trained weights are stored in the directory `models/agent_41b_{t}.pt`, where 't' represents the training time (0, 1, or 2). The optimal model, `models/agent_41b_{1}.pt`, is utilized to plot the same episode as in question 2.a) and is displayed in Figure 9.

The agent learns a meaningful policy. As depicted in Figure 8, the average evaluation reward shows an increasing trend and eventually surpasses zero, although it does not exceed the reward of the binary action policy. Figure 9 suggests that the agent consistently implements confinement measures for the whole episode. However, the policy does not seem to actively adapt to changes in the observable state.

Question 4.1.c) Toggle-action-space multi-action policy evaluation

We evaluate the optimal policy π^*_{DQN} with toggle-action-space using the code from question 2.b), generate analogous histogram plots, and contrast the outcomes in Figure 10. Despite a lower cumulative reward, the policy results in marginally fewer deaths and infected compared to the binary action policy, thus demonstrating superior epidemic mitigation.

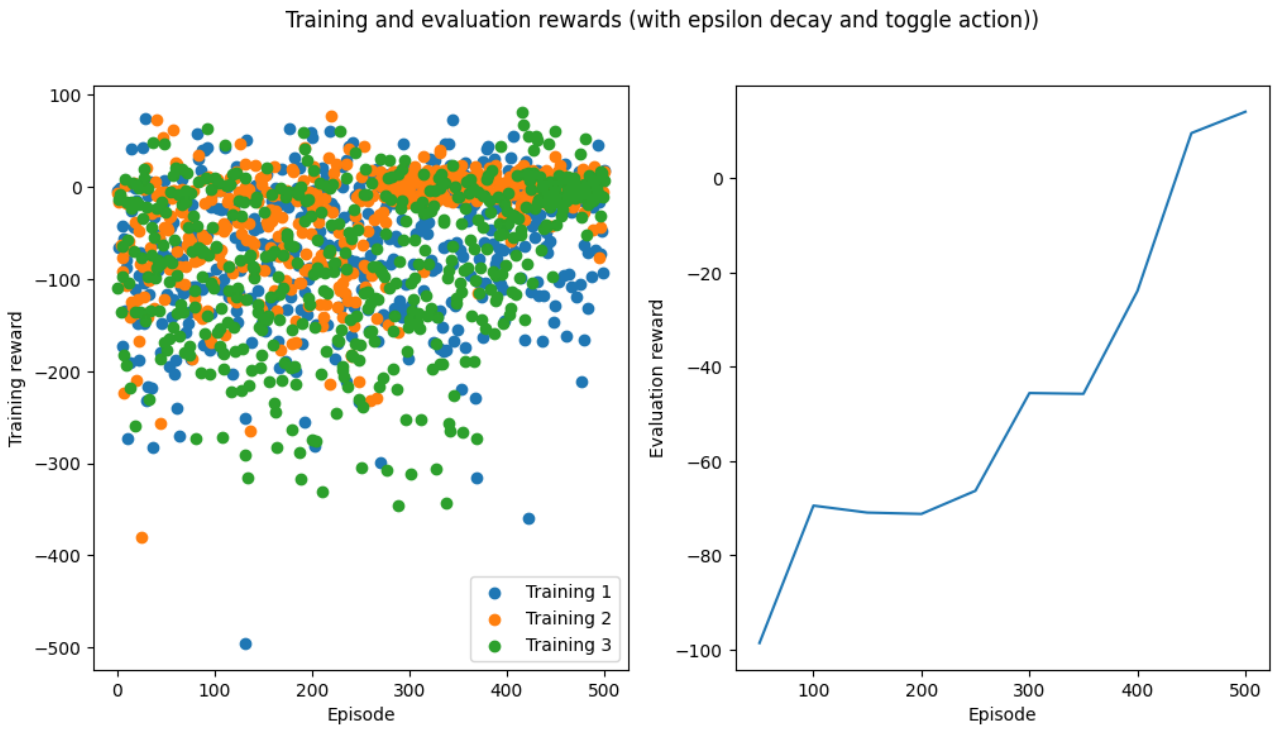

Figure 8: Training and evaluation traces for DQN with toggle-action-space.

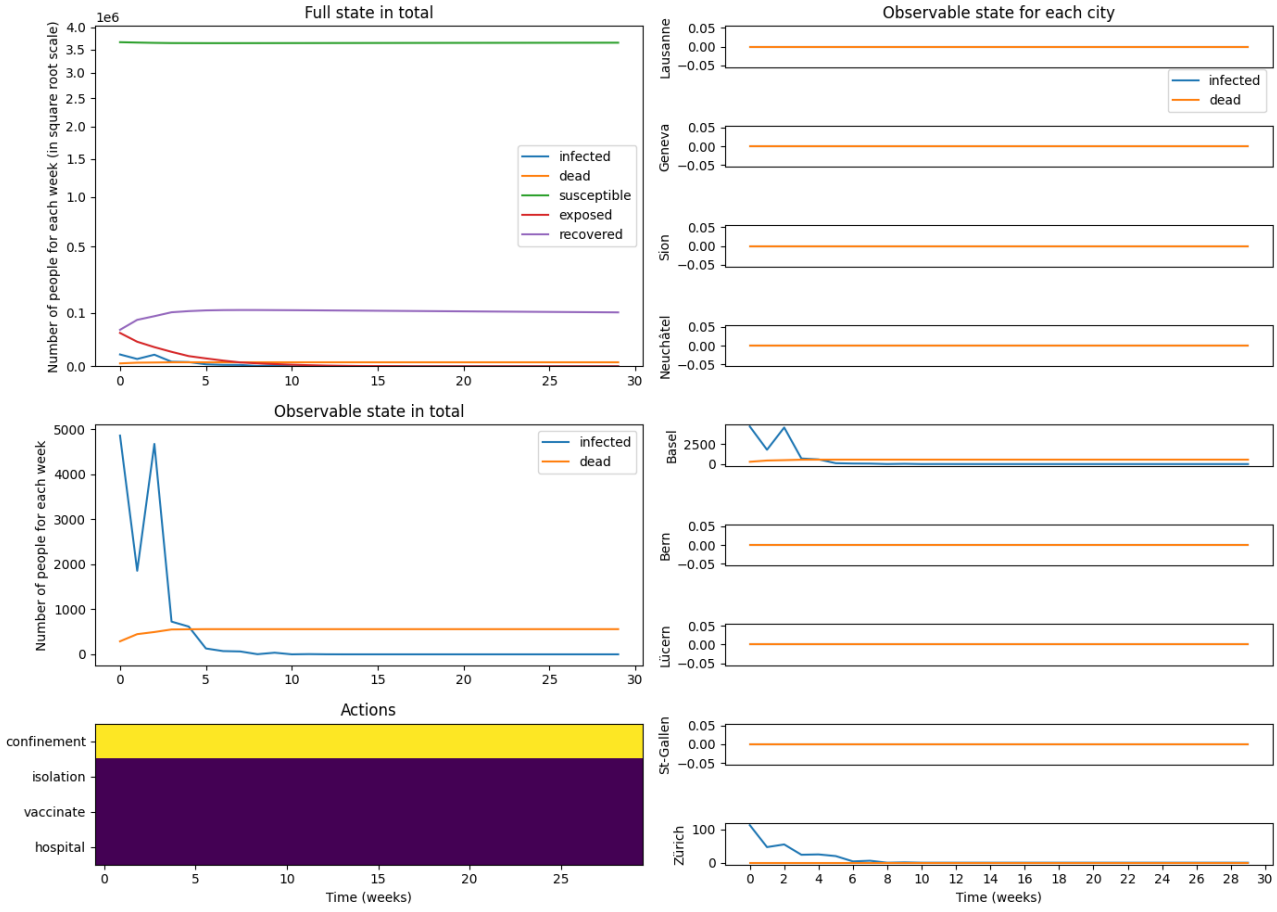

Figure 9: Training and evaluation traces for DQN with toggle-action-space.

Question 4.1.d) (Theory) question about toggled-action-space policy, what assumption does it make?

Employing a toggle-action-space policy presumes a discrete action space amenable to straightforward toggling between actions. It also implies the agent cannot alter states of multiple actions concurrently, necessitating a

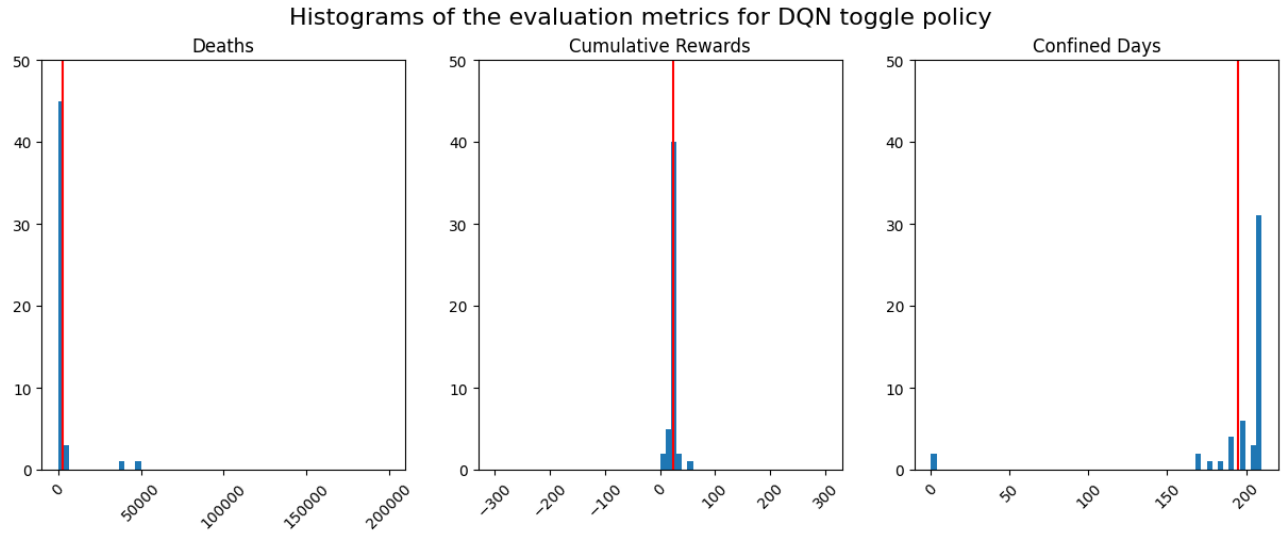

Figure 10: Distribution of the number of total confined days, the cumulative reward, and total deaths in histograms for the best DQN policy (with toggle-action-space).

compromise for the most rewarding switch. Nonetheless, there exist scenarios where such action toggling may be inappropriate.

- **Continuous Control Actions**: Toggle-action-space policies may not be suitable for tasks requiring continuous control, such as robotic control tasks that demand smooth movement or precise force application. Discrete action toggling lacks the necessary granularity for such control.
- **Hierarchical Actions**: In scenarios with a hierarchical action space, where actions comprise sub-actions or action sequences, toggle-action-space policies may not adequately represent the hierarchical structure due to their one-action-at-a-time limitation.

4.2 Factorized Q-values, multi-action agent

Question 4.2.a) multi-action factorized Q-values policy training

We use all configurations from the suggestion. The rewards are shown in Figure 11. We save our trained weights to the folder `models/agent_42a_{t}.pt`, where t is the training time (0, 1, or 2). We use the best model `models/agent_42a_{1}.pt` to plot the same episode in question 2.a) in Figure 12.

The agent learns successfully though with some problems. As depicted in Figure 11, the average evaluation reward increases and finally surpasses zero yet the dead hugely increases compared with previous policy. Figure 12 suggests that the agent consistently prioritizes vaccination efforts throughout the entire process and increases the number of hospital beds in the first 8 weeks. However, the agent does not significantly alter the isolation and confinement actions. Overall, the policy appears to be somewhat realistic as it minimizes the extent of isolation and confinement, which could potentially disrupt people's daily lives, while still effectively utilizing healthcare resources.

Question 4.2.b) multi-action factorized Q-values policy evaluation

We run the best performing policy π^*_{DQN} with multi-action factorized Q-values policy through the evaluation code from question 2.b), generate the same histogram plots, and compare the results in Figure 13.

In comparison to the toggled policy, the factorized Q-value policy yields greater rewards but leads to an increase in fatalities. Additionally, it entails no confinement, allowing people to avoid continuous quarantine. Moreover, the factorized scenario exhibits a higher percentage of individuals with established immunity, whereas the majority of people remain susceptible in the toggled policy.

Question 4.2.c) (Theory) Factorized-Q-values, what assumption does it make?

Using factorized Q-values assumes that the action space can be decomposed into independent decisions, where each decision can be evaluated separately. This approach assumes that the decisions do not have dependencies on each other and can be made independently.

Factorizing Q-values may not be suitable for action spaces where the decisions have **strong dependencies or interactions**. For example, consider a game where the actions involve a combination of moves that need

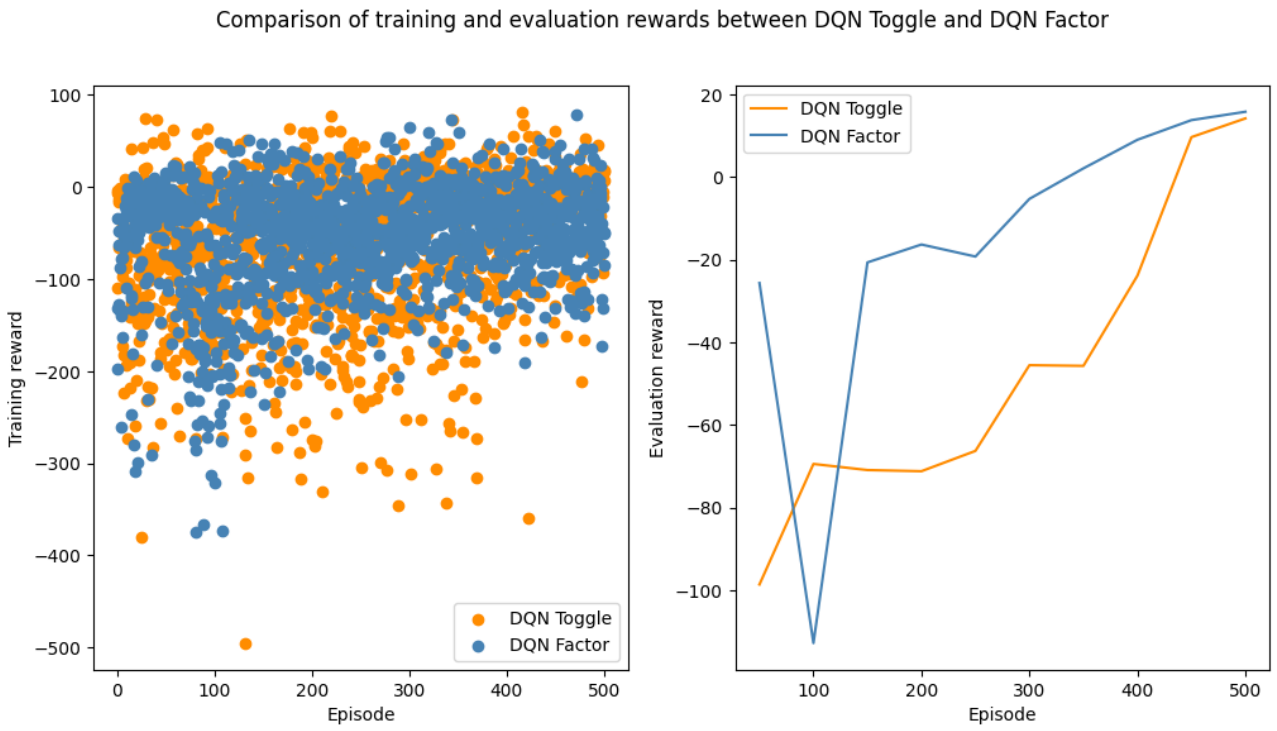

Figure 11: Comparison of training and evaluation traces between DQN Toggle and DQN Factorized.

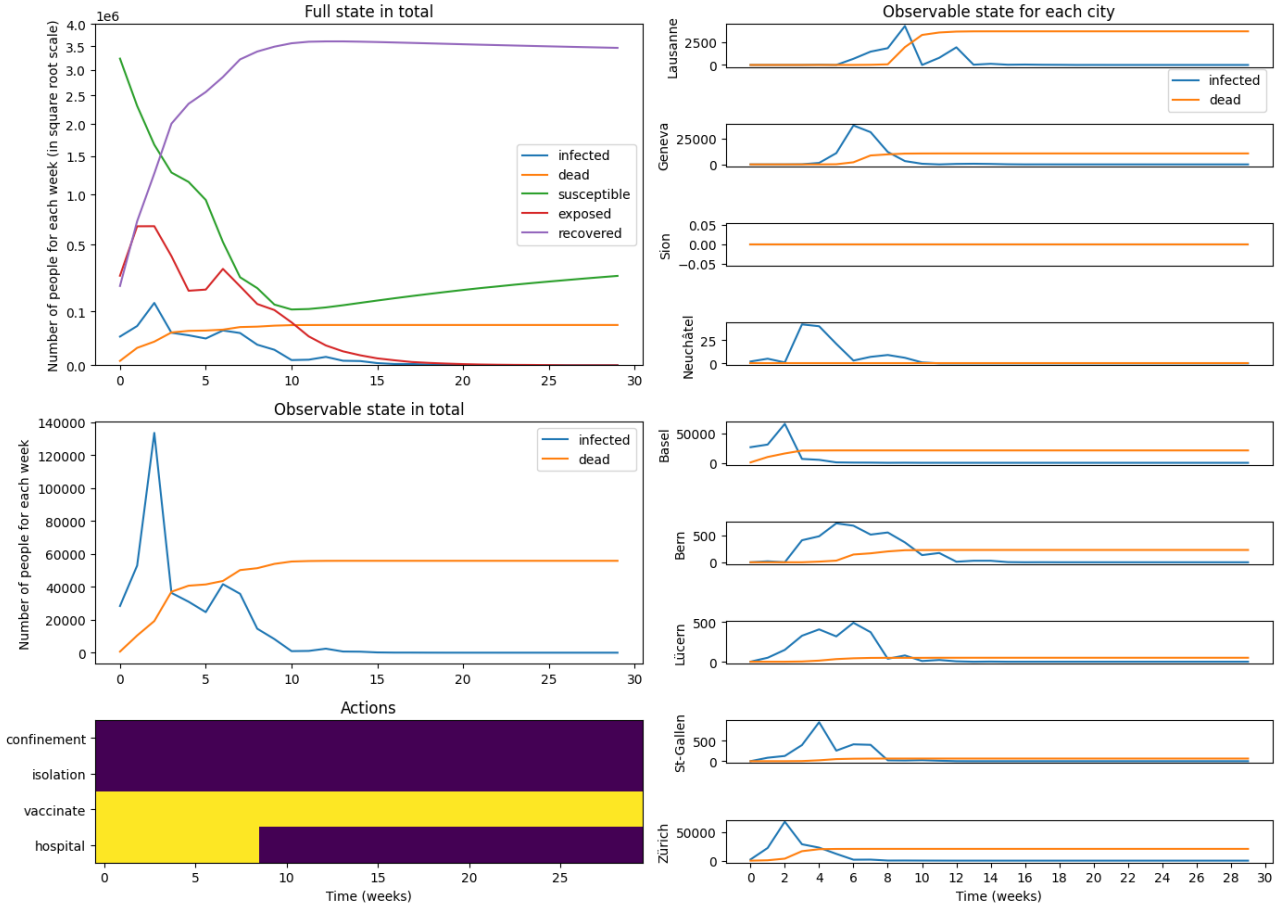

Figure 12: Training and evaluation traces for DQN with multi-action factorized Q-values.

to be executed in a specific sequence or timing. Factorizing Q-values independently for each move without considering their order or timing may result in ineffective strategies.

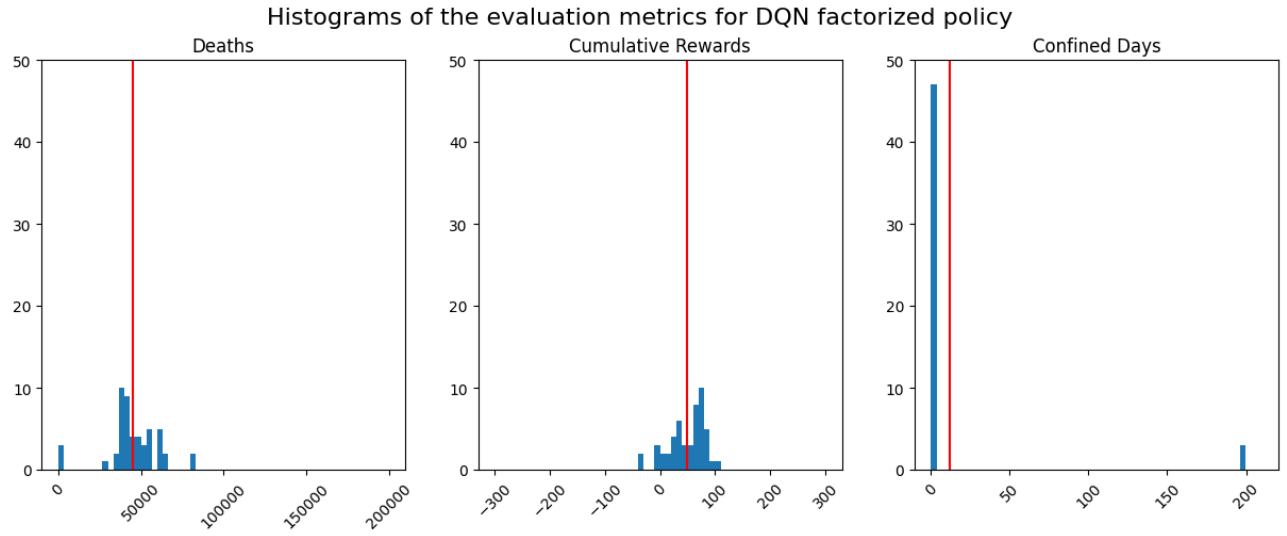

Figure 13: Distribution of the number of total confined days, the cumulative reward, and total deaths in histograms for the best DQN policy (with toggle-action-space).

5 Wrapping Up

Question 5.a) (Result analysis) Comparing the training behaviors

The outcomes are depicted in Figure 14, where DQN-DE denotes the decreasing epsilon DQN policy. During the initial 500 episodes, all neural network methodologies surpass Russo's policy. Notably, DQN-DE emerges as the superior single-action policy, while DQN Factorized slightly leads among multi-action policies. Single-action policies (DQN, DQN-DE) outdo multi-action policies. Despite this, given the complexity of the multi-action policy's action space and the increased parameter count in the neural network, these policies might require additional episodes to converge and potentially surpass single-action policies in performance.

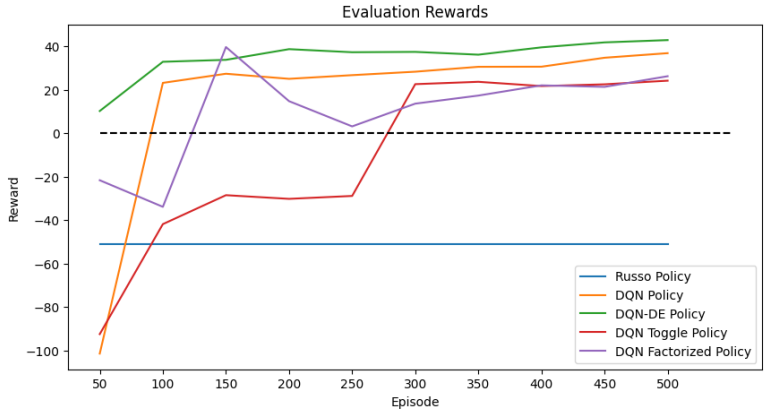

Figure 14: Evaluation traces for different policies.

Question 5.b) (Result analysis) Comparing policies

Table 1 shows the results and clearly mark the best performing policy with respect to each metric. It can be inferred that the policies π_{DQN} and $\pi_{\text{DQN-DE}}$ show progress in terms of both improving avg$[R_{\text{cumulative}}]$ and reducing avg$[N_{\text{deaths}}]$ compared to π_{russo}, at the cost of more confinement days. Policy π_{toggle} performs best in avg$[N_{\text{vaccination}}]$, avg$[N_{\text{hospital}}]$ and avg$[N_{\text{deaths}}]$, while policy π_{factor} outperforms in avg$[N_{\text{confinement}}]$, avg$[N_{\text{isolation}}]$ and avg$[R_{\text{cumulative}}]$.

Question 5.c) (Interpretability) Q-values

The heatmaps for the single-action DQN policies π_{DQN} and $\pi_{\text{DQN-DE}}$ are depicted in Figure 15 and Figure 16, respectively, while the heatmap for the π_{factor} policy is presented in Figure 17. Bold numbers denote the chosen action in the action space. The agent invariably selects the action with the higher Q-value for each action

Policy	avg[$N_{\text{confinement}}$]	avg[$N_{\text{isolation}}$]	avg[$N_{\text{vaccination}}$]	avg[N_{hospital}]	avg[N_{deaths}]	avg[$R_{\text{cumulative}}$]
π_{russo}	101.9	-	-	-	55155.1	-58.1
π_{DQN}	156.2	-	-	-	4925.3	32.9
$\pi_{\text{DQN-DE}}$	158.8	-	-	-	3210.4	44.1
π_{toggle}	194.9	50.0	**11.6**	**26.7**	**2764.6**	23.3
π_{factor}	**11.8**	**0.0**	210.0	61.2	45128.0	**49.8**

Table 1: Empirical means of variables under different policies (bold numbers are the best).

decision (True/False). Notably, π_{DQN} and $\pi_{\text{DQN-DE}}$ sporadically enforce city confinement, while π_{factor} initially increases hospital bed availability and sustains vaccination throughout the entire period. These policies reflect diverse strategies for epidemic control, with π_{DQN} and $\pi_{\text{DQN-DE}}$ implementing periodic confinement, and π_{factor} prioritizing hospital bed augmentation and vaccination.

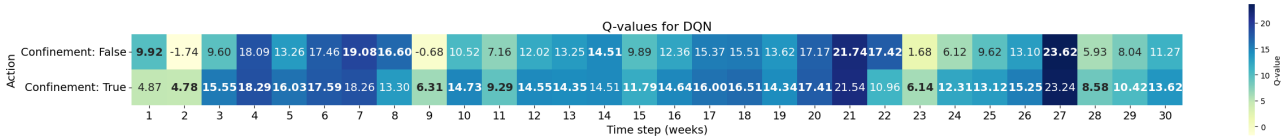

Figure 15: The heatmap of the π_{DQN} policy (without decreasing epsilon).

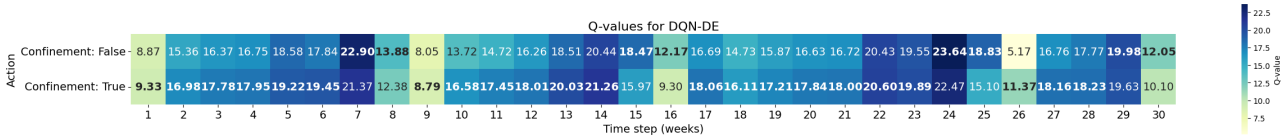

Figure 16: The heatmap of the $\pi_{\text{DQN-DE}}$ policy (with decreasing epsilon).

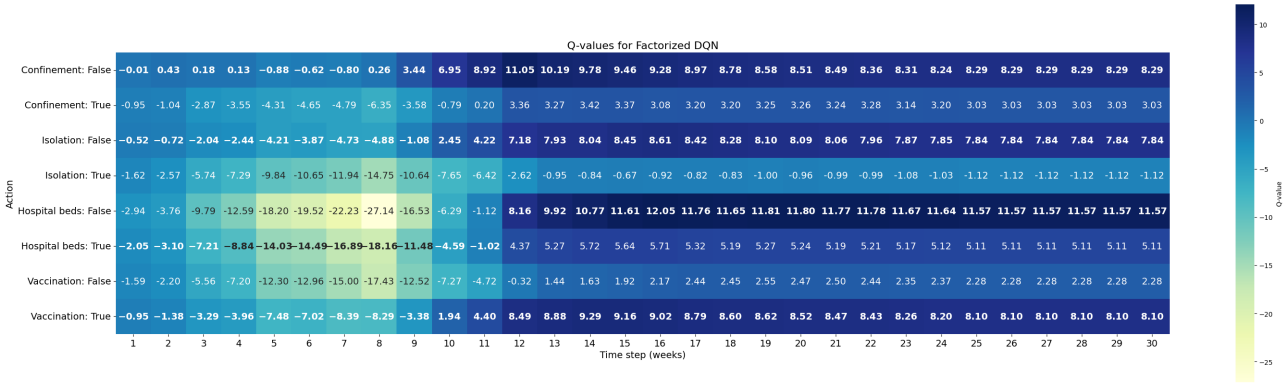

Figure 17: The heatmap of the π_{factor} policy (factorized DQN).

Question 5.d) (Theory), Is cumulative reward an increasing function of the number of actions?

Figure 14 shows that the cumulative reward is not a monotonically increasing function of action count. While more actions may suggest proactive epidemic mitigation to mitigate the number of death, the reward in Equation 2 is not solely death toll-dependent.

As Equation 1 shows, more actions can increase the action cost, reducing the reward. Despite potentially lowering the death toll, the decrease in $\Delta d_{\text{total}}^{[w]}$ may not sufficiently counterbalance the cost of increased actions. Table 1 indicates low death counts for single-action policies, suggesting limited potential for significant $\Delta d_{\text{total}}^{[w]}$ reduction. Thus, the reward may decrease with more actions due to sensitivity to action count.

$$\begin{aligned} \textbf{Action cost} \quad & \mathcal{C}(\mathbf{a}^{[w]}) = \mathcal{A}(\mathbf{a}^{[w]}) + \mathbf{1}_{vac} \cdot V + \mathbf{1}_{hosp} \cdot H + \mathbf{1}_{conf} \cdot C + \mathbf{1}_{isol} \cdot I \\ \textbf{Annoucement costs} \quad & \mathcal{A}(\mathbf{a}^{[w]}) = A \cdot (\mathbf{1}^{+}_{\text{vac}} + \mathbf{1}^{+}_{\text{isol}} + \mathbf{1}^{+}_{\text{conf}}) \end{aligned} \tag{1}$$

$$R^{[w]} = R_{\text{c}} - \mathcal{C}(\mathbf{a}^{[w]}) - D \cdot \Delta d_{\text{total}}^{[w]} \tag{2}$$