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

Du Junye (junye.du@epfl.ch), Zou Mengjie (mengjie.zou@epfl.ch)

Q1: Unmitigated Epidemics

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

Under the condition of taking no actions, we ran the epidemic simulation for one episode (with random seed 42), the plots are as followed in Figure 1.

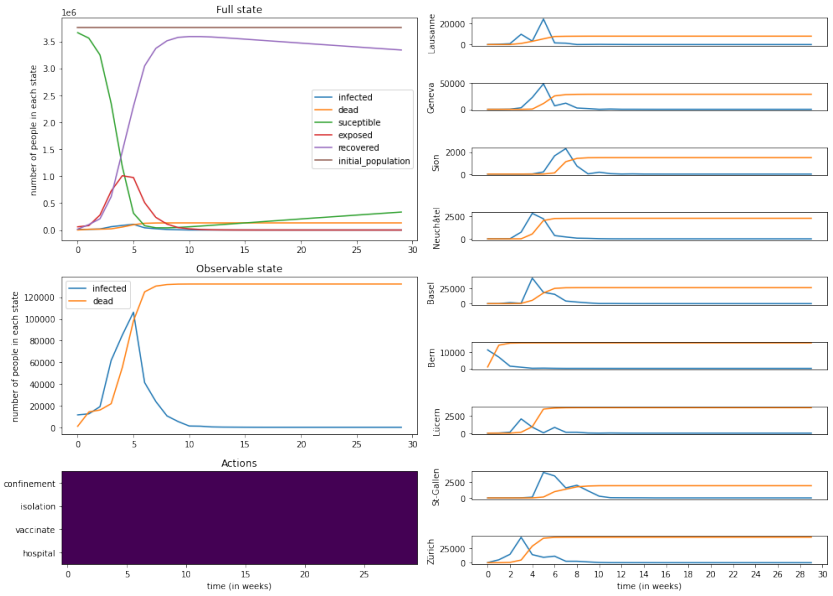


Figure 1: Simluation for unmitigated epidemics. Top left: the plot of variables $s^{[w]}_{\text{total}}, e^{[w]}_{\text{total}}, i^{[w]}_{\text{total}}, r^{[w]}_{\text{total}}, d^{[w]}_{\text{total}}$ over time; Middle left: the plot of variables $i^{[w]}_{\text{total}}, d^{[w]}_{\text{total}}$ over time; Right: the set of plots of variables $i^{[w]}_{\text{city}}, d^{[w]}_{\text{city}}$ over time.

Discuss the evolution of the variables over time.

1. From the *Full state* plot, we could conclude that the number of *susceptible* population and *recovered* population represent opposite trends, while $s^{[w]}_{\text{total}}$ encountered sharp decreased in the begining of the the episode and showed a slow upward trend after around 7 weeks. As about *exposed* population, $e^{[w]}_{\text{total}}$ continued growth since the beginning and achieved peak around week 4 followed by a gradual decline till week 10 and converged to 0.

2. Similar to the trend of $i^{[w]}_{\text{total}}$, the number of *infected* population also peaked around week 5, from different random seed initialization, we found that the shape of $i^{[w]}_{\text{total}}$ is proportional to the total$^{[w]}$ to some extent. For *dead* population, since $d^{[w]}_{\text{total}}$ is a cumulative measure of dead people, it showed monotone increase since beginning and remained stable around 130 thousand after week 9. It is also worthnoting that the explode growth of $i^{[w]}_{\text{total}}$ is usually a little bit prior than the wellspring growth of $d^{[w]}_{\text{total}}$ in our experiments with different seeds, which is consistent with our common sense.

3. From the *Per-City* plot and the map of Switzerland, it is not difficult to derive that the peak *dead* population is proportional to the population size of the corresponding city, in which case, Zurich, the most populous city in Swiss, has the largest *infected* and *dead* population. Besides, infection outbreaks in neighboring cities are usually close in time due to factors of geographic relationship, for instance, neighbouring to the epidemic origin - Bern, successive outbreaks occurred in Zurich and Lucern in a few weeks.

Q2: Professor Russo's Policy

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

Under the Pr. Russo's Policy, when the *infected* population at the end of week w exceeds the threshold 20 thousand, the action *Confine* will be implemented to the whole county for 4 weeks, the simulation results is represented as below in Figure 2.

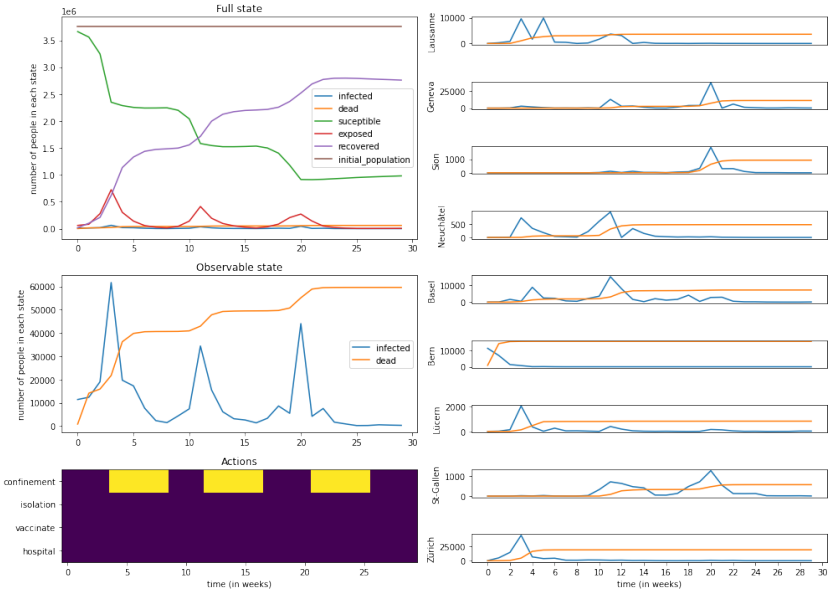


Figure 2: Simulation for Russo's Policy. Top left: the plot of variables $s^{[w]}_{\text{total}}, e^{[w]}_{\text{total}}, i^{[w]}_{\text{total}}, r^{[w]}_{\text{total}}, d^{[w]}_{\text{total}}$ over time; Middle left: the plot of variables $i^{[w]}_{\text{total}}, d^{[w]}_{\text{total}}$ over time; Bottom left: the plot of the actions taken by Russo's policy; Right: the set of plots of variables $i^{[w]}_{\text{city}}, d^{[w]}_{\text{city}}$ over time.

Discuss how the epidemic simulation responds to Pr. Russo's Policy

1. From the *Full State* plot, we could easily derive that $s^{[w]}_{\text{total}}, r^{[w]}_{\text{total}}$ generally remained stable during the *Confinement* week, the reason is largely due to that the *Confinement* action dramatically reduced the potential of people being exposed to virus. Besides, comparing to the unmitigated epidemics, the *exposed* population is reduced and the following outbreaks gradually diminished in intensity.

2. Pr. Russo's Policy greatly decreased both the $i^{[w]}_{\text{total}}$, and the $d^{[w]}_{\text{total}}$ to nearly half of the unmitigated case. Besides, unlike rising straight to the peak, the *dead* population under Russo's Policy showed phased growth, where it kept stable during the *Confinement* weeks. Also, each time the *infected* population reached the top, the *Confinement* action will enforce it to drop rapidly to the bottom till the unblocking week.

3. From the *Per-City* Plot, the implementation of Russo's policy slowed down the spread of the epidemic from city to city, for instance, before week 4, the epidemic had already spread from Bern to cities like Lucern, and the *Confinement* action enforced the infection rate in above cities to drop dramatically and effectively prevented the spread of the virus to rest cities like Geneva.

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

In this section, we ran 50 simulation episodes where actions are chosen from Russo's Policy π_{Russo}, for each episode, we recoreded the following variables: $N_{\text{confinement}}$, $R_{\text{cummulative}}$, N_{deaths} and plotted a histogram as followed in Figure 3.

Q3: A Deep Q-learning approach with a Binary Action Space

Question 3.a) DQN with Fixed Exploration

In section 3, we improved Pr. Russo's policy using deep Q-learning approach, where the input of the neural network is a 2*9*7 matrix indicating the death and infected population of each city and the output is a binary

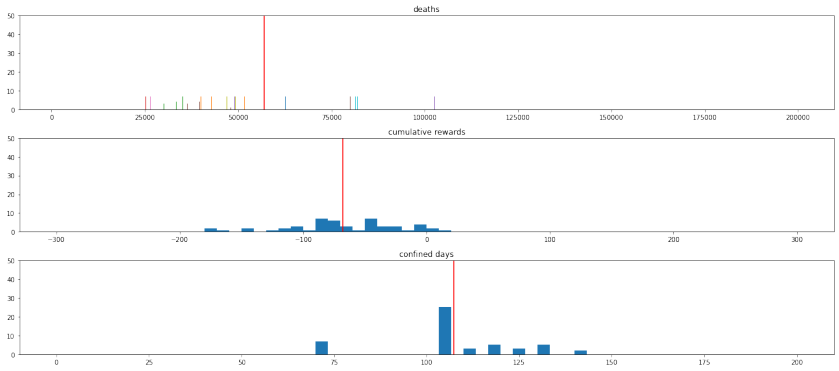


Figure 3: Histograms for Question 2b. Corresponding variables from top to bottom: N_{deaths}, $N_{\text{cumulative}}$ and $R_{\text{confinement}}$.

variable indicating whether to choose the action *Confinement* or not. In our training environment of 3.a, we used the fixed ϵ action policy and set the learning rate as $5 \cdot 10^{-3}$, the reward trace plot of it is shown in Figure 5.

Interpretation of the policy

Similar to the questions above, we interpretate the binary deep Q-learning approach policy through the plots shown in Figure 4

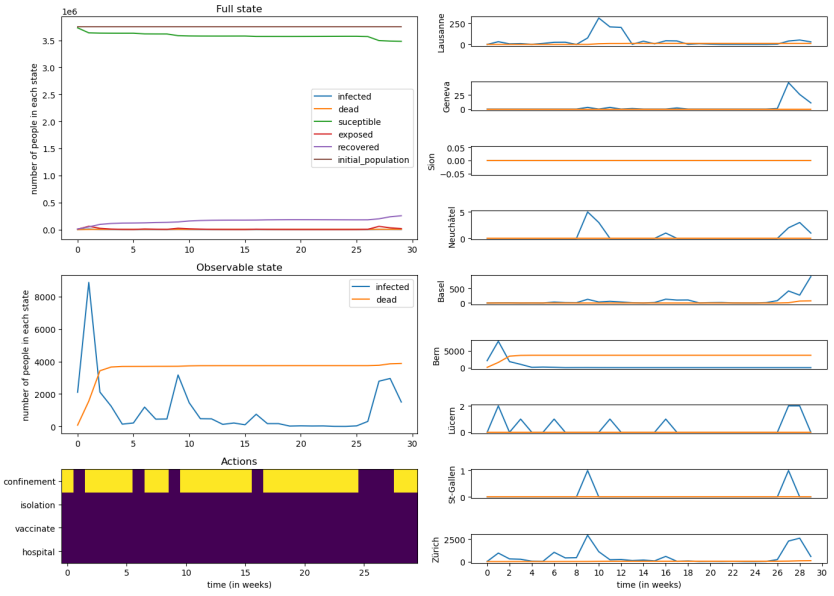


Figure 4: Policy interpretation of the Deep Q-learning approach with Binary Action Space

As shown in Figure 4, the Deep Q-learning approach with Binary Action Space is quite effective and successfully controlled the epidemic death toll at around 12,000, which is only around one fifth of Russo's Policy. Similar Russo's, the action *Confinement* will be implemented when the infected population appears a tendency to explode and performs effectively to decrease the infected population to a controllable state. The different point is that the Deep Q-learning approach implemented the *Confinement* action much more frequently and the infected rate could be stable at a very low level in the late period of the episode.

Question 3.b) DQN with Decreasing Exploration

In this section, we will reimplement DQN agent with decreasing ϵ depending on the time t:

$$\epsilon(t) = \max\left(\frac{\epsilon_0(T_{\max} - t)}{T_{\max}}, \epsilon_{\min}\right) \tag{1}$$

where in our training environment $T_{\max} = 500$, $\epsilon_0 = 0.7$ and $\epsilon_{\min} = 0.2$. In the following part we plot the training and evaluation reward trace after three experiments and compare the results.

Figure 5: Training & evaluation trace with fixed ϵ

Figure 6: Training & evaluation trace with decreasing ϵ

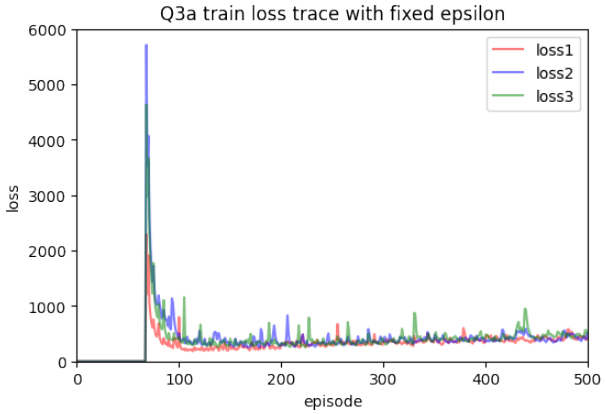


Figure 7: Training loss trace with fixed ϵ

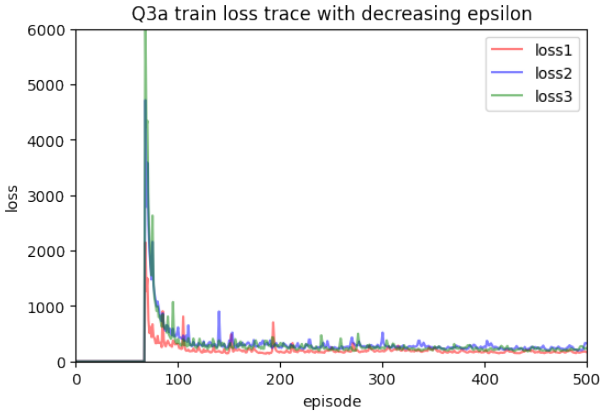


Figure 8: Training loss trace with decreasing ϵ

Discussion on comparing fixed ϵ and decreasing ϵ: From our experiments, the maximum reward under decreasing ϵ policy is slightly higher but it doesn't have that much difference since the action space is binary and the exploitation won't play a big difference in terms of the maximum reward. However, comparing two traces, we could find that the reward of the training set on decreasing ϵ is gradually converging near to the reward on evaluation set since the policy is progressively favoring taking the action with highest Q-value, which is closer to actual action. As such, although in this section the two policies achieve similar rewards, the decreasing ϵ policy has stronger generalization when it comes to higher dimensional action space.

Question 3.c Evaluating best policy against Pr. Russo's policy

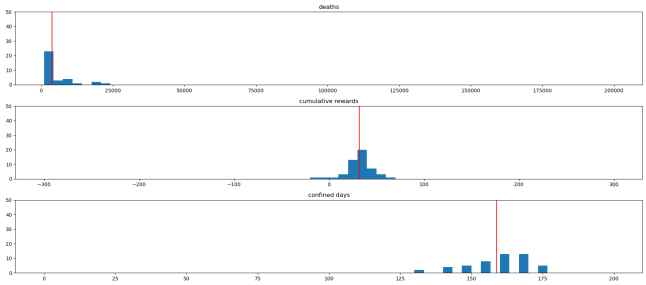


Figure 9: Histograms for Question 3c. Corresponding variables from top to bottom: N_{deaths}, $N_{\text{cumulative}}$ and $R_{\text{Confinement}}$. The red and the blue vertical lines represent mean and median respectively.

From the histogram in Figure 9 under the policy of Deep Q-learning approach, we could observe that the mean *dead* population is greatly reduced from 60000 to 4000 in comparison to the Pr. Russo's policy. Apart from this, the expected cumulative reward also turn positive at around 35 while under the polict of Russo's is only -70. The improvement in terms of deaths and reward is obviously effectively, however, since the Deep Q-learning approach with binary action space frequently implement the *Confinement* action, the mean number

of *Confinement* days also significantlty increase by 50% to around 150 days, that is 21 weeks out of 30 weeks in one episode.

Q4: Policies Towards More Complex Action Spaces

4.1: Toggle-Action-Space Multi-Action Agent

In this section, we extended the simple *Confinement* action space to more complex space containing *Confinement*, *Isolation*, *Hospital* and *Vaccination*. To achieve this, we implemented *Toggle* and *Factorized Q-value* approaches and compared their performance.

Question 4.1.a) (Theory) Action space design

The reasons for preferring toggle action space over computing Q values for individual actions can be classified into the following main categories:

1. Curse of Dimensionality: Toggle-action space has fewer output dimensions compared to direct action space. Instead of outputting 16 action values, *Toggle* policy could only output action changes. This reduction in dimensionality theoretically enhances convergence and stability during the training process.

2. Smooth policies: At each desicion time, the agent can only toggle on or off one type of action. These avoids drastic changes in adjacent actions based on the empirical assumption that the dynamics of the environment is also smooth (i.e., no sudden changes).

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

In the observation preprocessor, we added 4 additional observations, which are 4 previous actions retrieved from *ModelDynamics*. Meanwhile, the action preprocessor is also modified so that the agent's output can toggle on or off a previous action. We tested different learning rates for the agent. During our experiment, we found that a learning rate of 10^{-3} can better stablizing the evaluated reward curve and reach a quicker convergence.

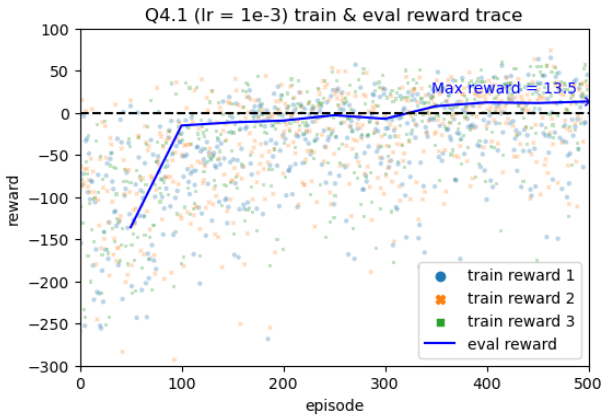


Figure 10: Traced training rewards and evaluation rewards for agent in Q4.1 with the learning rate of 10^{-3}.

Figure 11: Traced training rewards and evaluation rewards for agent in Q4.1 with learning rate of 10^{-5}.

Interpretation of policy: Figure 12 illustrates how the agent behaves in one episode. It is noteworthy that the agent tries to implement *Confinement* action when the *infected* population expresses rising trend and persists in applying *Confinement* action until the *infected* population returns to relatively low stage. Thanks to the more complex action space, the agent now is able to implement *Hospital* action to prevent the *death* number from increasing too fast. The *Isolation* and *Vaccination* are relatively less used by the agent, from our experiments, these two actions are more likely to appear when the *infection* and *death* population are at a stable situation.

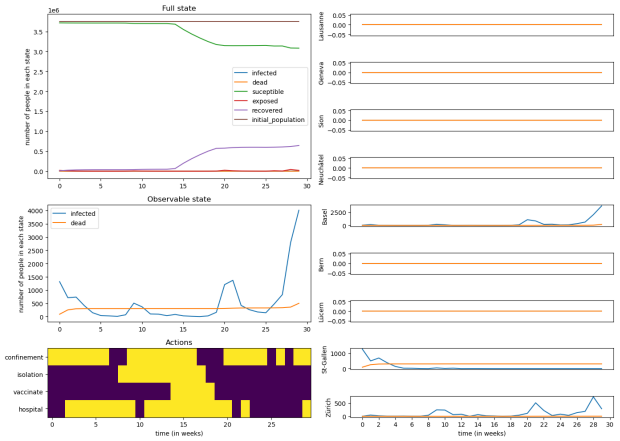


Figure 12: Simulation of one episode using agent in Q4.1

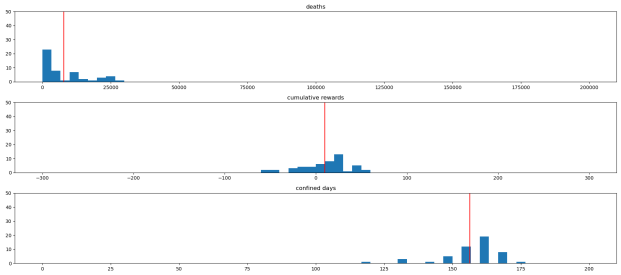


Figure 13: Histograms for Question 4.1.c. Corresponding variables from top to bottom: $N_{\text{confinement}}$, N_{deaths} and $R_{\text{cumulative}}$. The red vertical lines represent mean value.

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

As shown on Figure 13, the distribution of the statistics is sparser while in Figure 9, the distribution of which is more concentrated. This suggests that $\pi^{\star}_{toggle}$ has a weaker generalization ability compared to the Binary policy. Comparing the *death* number and the total *confined* days, we can assume that $\pi^{\star}_{toggle}$ has learned to decrease the number of *confined* days in order to gain more reward. However, this results in more deaths numbers as shown in the histogram.

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

1. **Discrete Binary action**: The actions implemented by the agent are dicrete vaiables, for instance, one agent could only choose whether to take *Hospital* or not but couldn't decide the exact number.
2. **Independence**: Different actions of *Toggle* policy at a given decision time are independent from each other. If actions are mutually affected, the *Toggle* policy wouldn't make sense.
3. **Correlation between actions at adjacent decision time**: The agent has an understanding of the actions in the previous step, and current actions are made based on not only the observables but also the previous actions.

The situation this method wouldn't make sense when the actions are no longer independent or discrete. For instance, when taking the hours of *Confinement* or the percentage of *Vaccination* into consideration, the action space is no longer discrete and the method won't apply.

4.2 Factorized Q-values, multi-action agent

To represent all combinations of actions, we used a four-digit binary number to represent the total 16 combinations of actions. Each digit corresponds to one specified action, and thus in the neural network we need to calculate the Q-values for the two states (on or off) of each action. The input dimension for the neural network is again $2 * 9 * 7$, and the output dimension is extended to $4 * 2$, which is the number of actions multiplied by the two on-off states. The Q-value for the combined action is calculated with the following formula:

$$Q(\mathbf{a}^{[w]}, s) = Q(a^{[w]}_{conf} \cup a^{[w]}_{\text{isol}} \cup a^{[w]}_{\text{hosp}} \cup a^{[w]}_{\text{vacc}}, s) = \sum_{\mathfrak{d} \in \text{decisions}} Q(a_{\mathfrak{d}}, s). \tag{2}$$

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

From Figure 14 we can see that the agent gains higher rewards as the training episodes increases and the maximum average reward over three different training processes is 20.7. Comparing π_{toggle} with π_{factor} in Figure 15, we can see that at the learning rate of 10^{-3}, both agents converged quickly and gradually achieved higher rewards during the training process. We can say the agent has learned from the training process.

Interpretation of policy:

Figure 16 records how the agent behaves in one episode. We can see that the main strategy of the agent is to combine *Confinement* with *Add hospital beds*, the other two types of actions are rarely used. In comparison to how $\pi^{\star}_{toggle}$ behaves in Figure 12, $\pi^{\star}_{toggle}$ seems to learn to apply *Add hospital beds* slightly before *Confinement* to control the number of death during large outbreaks.

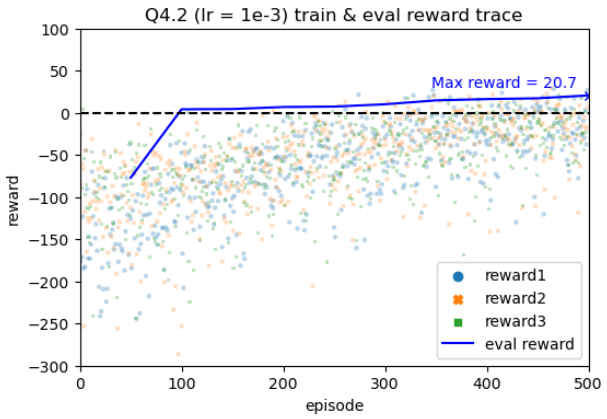


Figure 14: Traced training rewards and evaluation rewards for agent in Q4.2 with the learning rate of 10^{-3}. Scatter points are the training rewards.

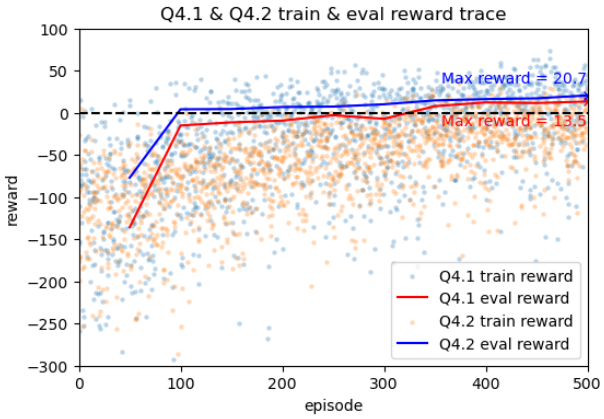


Figure 15: Traced training rewards and evaluation rewards for agent in Q4.1 and Q4.2 with the learning rate of 10^{-3}.

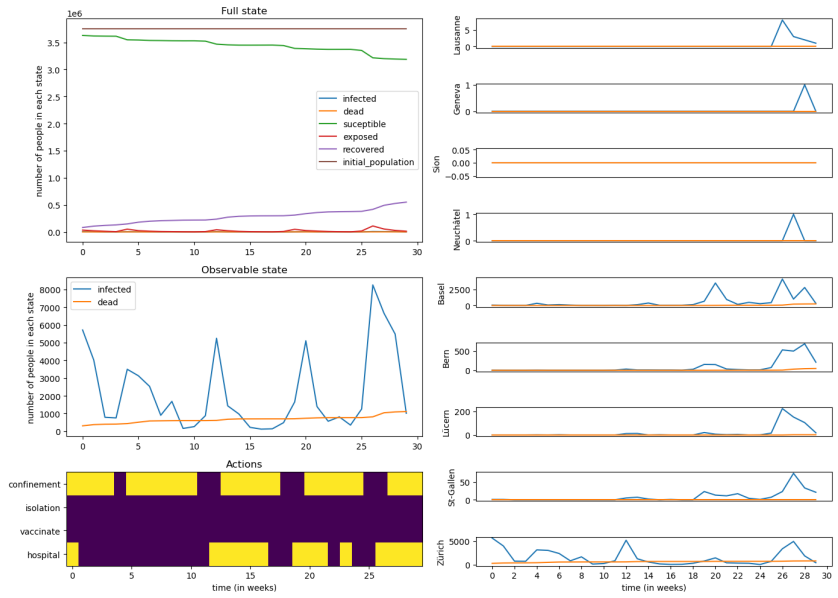


Figure 16: Simulation of one episode using $\pi^{\star}_{factor}$.

Question 4.2.b Multi-action factorized Q-values policy evaluation

The histogram of $\pi^{\star}_{factor}$ in Figure 17 suggests that the policy achieved a higher cumulative reward than $\pi^{\star}_{toggle}$. Comparing to Figure 13, the average death number is much lower and the distribution is more concentrated, which suggests that $\pi^{\star}_{factor}$ generalizes better than $\pi^{\star}_{toggle}$ on different environment seeds.

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

Besides of the assumption on the discrete action space which has been discussed in Question 4.1.d, the factorization of the combined Q-values relies on the assumption that the Q-values for four actions are mutually independent. Apart from it, the Factorized Q-values approach has the memoryless feature and take actions regardless of former situation.

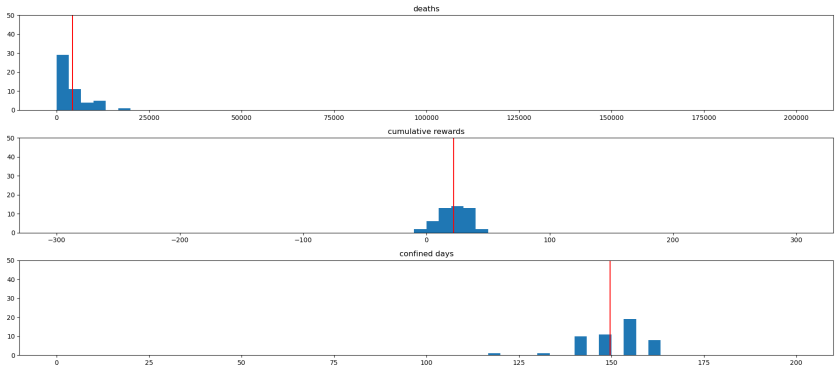


Figure 17: Histogram for $\pi^{\star}_{factor}$. Corresponding variables from top to bottom: N_{deaths}, $R_{\text{cumulative}}$ and $N_{\text{Confinement}}$. The red vertical lines represent mean value.

Q5: Wrapping Up

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

From Figure 18, we can see that the DQN policy outperformed other policies and even multi-action policies in terms of cumulative reward. Additionally, the single-action policies have a steeper training curve and converge faster within the initial 500 episodes. The reason behind this phenomenon would be that the multi-action policies are under-fitted with only 500 episodes of training since the action space is significantly larger in the multi-action scenario than the binary-action one. However, if given sufficient number of episodes in training, it's possible that the multi-action policies could achieve better results than what is shown on the Figure 18.

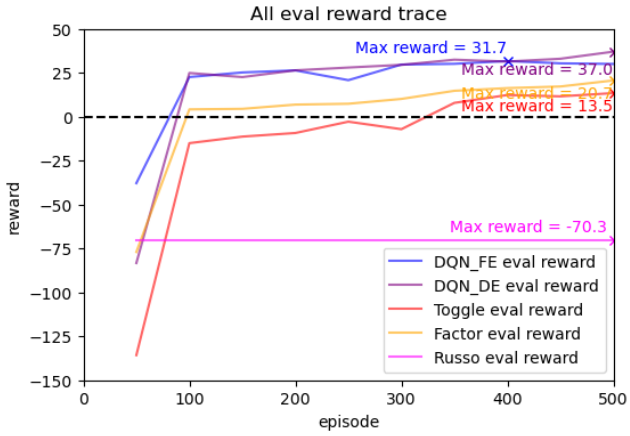


Figure 18: Combined plot of all policies. DQN_FE represents the DQN agent with fixed epsilon. DQN_DE represents the agent with decreasing epsilon. Toggle corresponds to the curve for policy π_{toggle}, and Factor corresponds to the curve for policy π_{factor}.

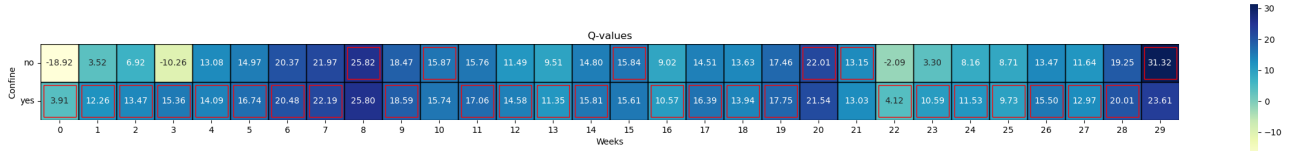
Question 5.b (Result analysis) Comparing policies

policy	avg$[N_{confinement}]$	avg$[N_{isolation}]$	avg$[N_{hospital}]$	avg$[N_{vaccination}]$	avg$[N_{deaths}]$	avg$[R_{cumulative}]$
π_{russo}	**98.98**	–	–	–	58922.90	-70.28
π_{DQN}	158.90	–	–	–	**3812.86**	**32.06**
π_{toggle}	156.38	17.22	88.06	12.60	7882.12	10.00
π_{factor}	149.66	**0.7**	**72.8**	**0.28**	4340.66	21.89

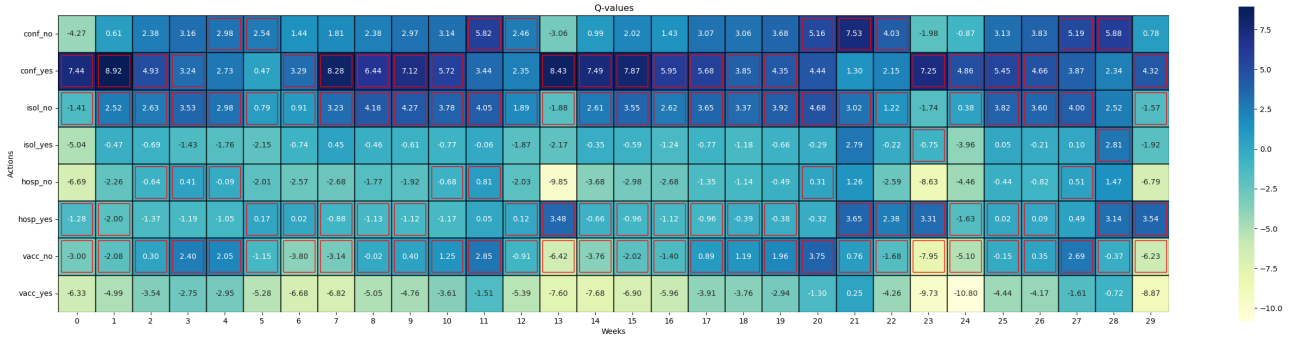
Table 1: Empirical means of statistics computed over the 50 episodes for every policy.

Discussion: As shown in Table 1, π_{DQN} has the best average cumulative reward even though it only acts on *Confinement*. π_{russo} has the least average number of *confined* days, but the average death number is the highest, and the average cumulative reward is the lowest. Among the agents powered by neural networks, single-action agents outperformed multi-action agents in terms of average cumulative reward and average number of deaths.

Among the multi-action agents, π_{factor} achieved better results than π_{factor} in all criterias, and the average cumulative reward is twice as much as that of π_{factor}.



(a) The Q-values of $\pi^{\star}_{DQN}$ in one episode. The red rectangles mark the action the agent takes.



(b) The Q-values of $\pi^{\star}_{factor}$ in one episode. The red rectangles mark the action the agent takes.

Figure 19: Q-values heatmap for $\pi^{\star}_{DQN}$ and $\pi^{\star}_{factor}$.

Question 5.c (Interpretability) Q-values

The heatmaps of the Q-values in a simulated episode are as Figure 19a and Figure 19b. The heatmap for both policies are generated using the same environment seed, and the red rectangles mark the actions guided by greedy policy.

Discussion: For DQN in Question 3, the agent chooses actions basing on the highest Q-values. As shown in the plot, at each step the policy compares two Q-values and chooses the action based on the one with higher Q-values. For π_{factor} policy, the agent chooses whether or not to adapt one action based on the corresponding neuron pair. As we can see in Figure 19b, for each decision time, the agent tends to act on *Confinement* and *Add hospital beds* over *Isolation* and *Vaccinate*, since the Q-values of performing the former two actions are generally larger in the heatmap. At the start of each *Confinement* period, the Q-value for implementing *Confinement* is significantly higher than not implementing it, and it gradually decreases over time.

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

If the training episodes are sufficient, increasing the number of actions will not yield worse results, which means the cumulative reward is an increasing function of the number of actions. This can be explained by the fact that policies involving fewer actions are encompassed within those involving a greater number of actions. Put differently, the agent has the option to refrain from executing specific actions while still attaining identical results as when limited actions are employed. However, in practice, especially when given limited number of training episodes, this conclusion may not apply, since increasing the number of actions will also increase the size of the action space. And the agent will be more likely to suffer from under-fitting issues.