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

Mariia Vidmuk, Daria Yakovchuk

1 Introduction

The goal of this mini-project is to use deep-Q-learning to train an artificial agent that can make decisions about how to mitigate an epidemic process. Throughout the project, we evaluated the performance of different methods based on their action and observation spaces.

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

Running the epidemic simulation for one episode (30 weeks), without epidemic mitigation (no action is taken, i.e. all values in the action dictionary are set to False). Produced plots:

1. A 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, where time is measured in weeks and all the variables share the y axis scaling.
2. A plot of variables $i^{[w]}_{\text{total}}, d^{[w]}_{\text{total}}$ over time, where time is measured in weeks and all the variables share the y axis scaling.
3. A set of plots of variables $i^{[w]}_{\text{city}}, d^{[w]}_{\text{city}}$ over time, where time is measured in weeks (one subplot per-city, variables share the y-scaling per-city).

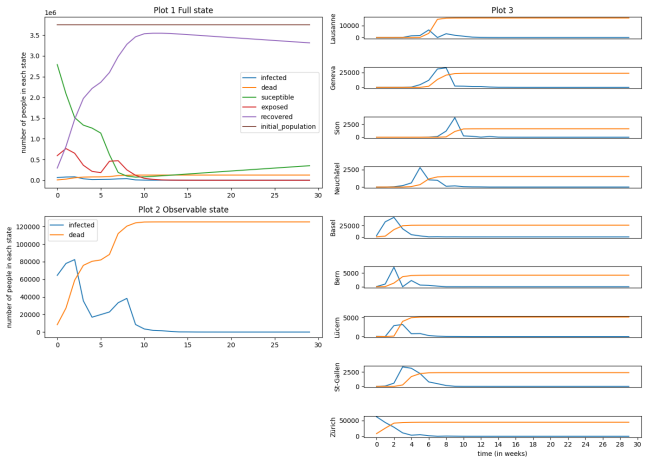

Figure 1: Unmitigated scenario

Discussion:

When an agent does nothing to stop the disease from spreading the pandemic is likely to spread quickly and finish with a certain amount of deaths. This is precisely shown in the second and third plots from the infected and dead: no more infection, thus, no more deaths (infected -> 0 and dead -> constant). That is what we can observe in plots (2 and 3): the number of infected and exposed people in total has a peak at the beginning of epidemic, which leads to increased deaths and recoveries numbers.

In the first plot, one can see an increase in susceptible people at the end of the simulation episode, which is due to the loss of immunity over time.

2 Professor Russo's Policy

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

Running the epidemic simulation for one episode(30 weeks) using Pr. Russo's Policy to pick actions. Produced plots: Plot 1, Plot 2, Plot 3 from the previous question (1.a) and Plot 4. A plot of the action taken by the policy over time (whether the policy chooses to confine or not).

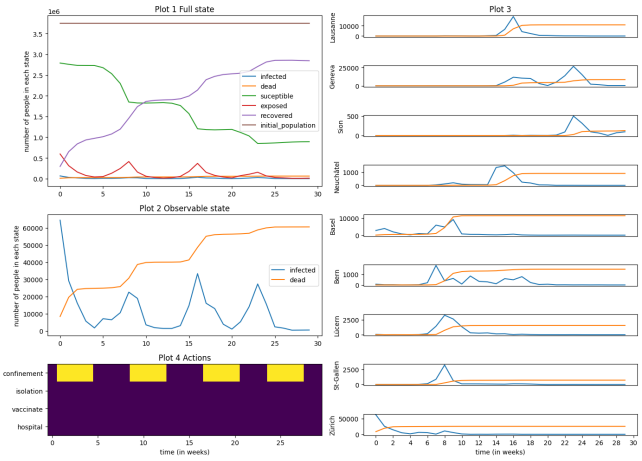

Figure 2: Pr. Russo's Policy

Discussion:

Pr. Russo's Policy aims to reduce opportunities for the virus to spread and thus reduce the overall number of infections but periodically.

Here we can observe the periodic increase and decrease of infected individuals in total for plots 1 and 2 and per city for plot 3 (Basel is the only city where the boost of deaths was higher than for infections).

Also, the number of deaths decreased using Pr. Russo's Policy (60K) if compare to the first unmitigated scenario (120K). Over 30 weeks, the increase in death was mildly growing. However, in the unmitigated scenario, we saw a boost in the first 10 weeks.

Overall, Pr. Russo's Policy could potentially help mitigate the spread of an epidemic, but still, we need other important actions to be done such as vaccination.

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

Evaluation procedure:

- run 50 simulation episodes where actions are chosen from Pr. Russo's Policy
- for each episode, save the following values:

1. the **number of total confined days** $N_{\text{confinement}} = 7 \cdot$ number of confined weeks ,
2. the **cumulative reward** (the sum of all rewards collected during the episode) $R_{\text{cumulative}} = \sum_{i \in [0,\ldots,30]} R^{[i]}$.
3. the **number of total deaths** $N_{\text{deaths}} = d_{\text{total}}^{[30]}$,

The Values above could be seen in the plot below. (Fig. 3)

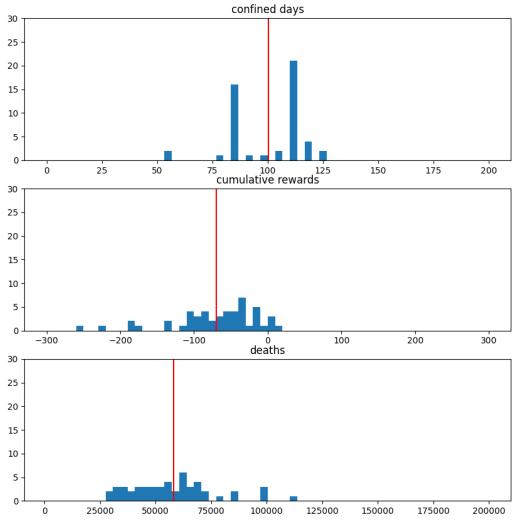

Figure 3: Evaluation of Pr. Russo's policy

3 A Deep Q-learning approach

Question 3.a) implementing Deep Q-Learning

Implementing and training the Deep Q-Learning agent π_{DQN} for 500 training episodes, with $\epsilon = 0.7$.

Plots of the training trace (for all 3 training runs, distinguished with different colours) and the evaluation trace (averaged across 3 training runs).

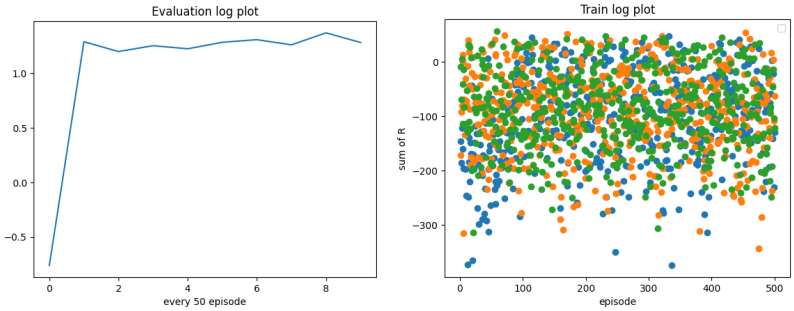

Figure 4: DQN training

Discussion: Does your agent learn a meaningful policy?

An agent with $\epsilon = 0.7$ acts randomly during its exploration phase, but doing so with such big ϵ is harmful to learning a meaningful policy if the agent continues to take actions based on mostly random experience.

That is what we can observe in the training trace plots. Yet, the evaluation trace shows that even with huge ϵ the agent succeeded in learning.

Plot one of the episodes of the best DQN Policy.

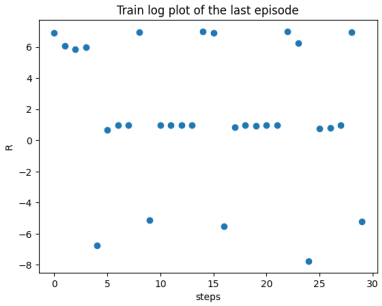

Figure 5: Episode of the best DQN Policy

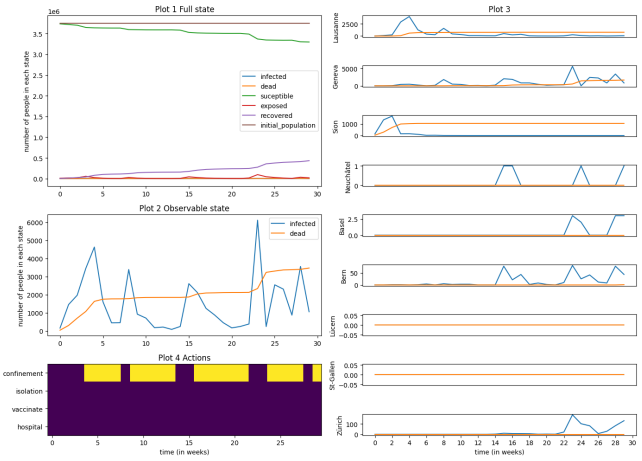

Figure 6: Additional plot DQN

Interpretation:

Here we can observe that as soon as the agent does nothing (no confinement), the reward goes down with an increase of infected individuals, and the agent does its action of confinement. It is shown in plot 4 of actions and on the train log plot of the last episode of the best policy we obtained. Also, one can observe that the agent with the DQN policy shows significantly better results than the agent with Pr. Russo's Policy, because the number of deaths in total was reduced to 3000 in the simulation episode.

Question 3.b) decreasing exploration

Implementing and training Deep Q-Learning agent for 500 training episodes with decreasing ϵ.

Plots of the training trace (for all 3 training runs, distinguished with different colours) and the evaluation trace (averaged across 3 training runs).

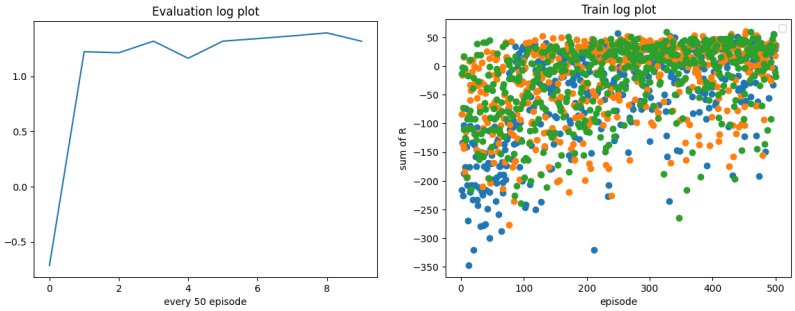

Figure 7: DQN decreasing exploration training

Plot one of the episodes of the best DQN decreasing Policy.

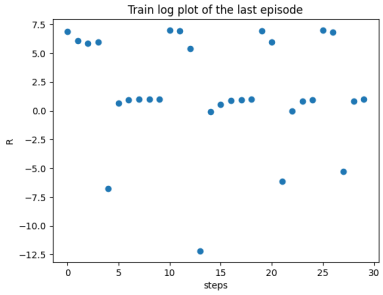

Figure 8: Episode of the best DQN Policy with decreasing exploration

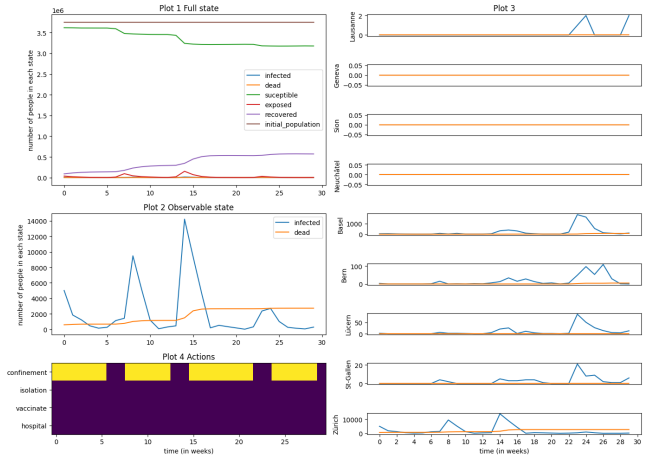

Figure 9: Additional plot DQN with decreasing exploration

Discussion: Compare and discuss the results between questions 3.a) and 3.b). Which policy gets the best results and why?

The difference between 3.a and 3.b can be distinguished from the train trace plot. The plot of 3.a is noisier because of the constant and huge exploration. The decreasing exploration policy can be better because with

time we won't rely on random actions but only on those which increase a reward. The reward of the agent with DQN decreasing policy on the training log plot of the last episode of the best model is slightly higher. And the number of deaths is roughly the same at the end of the simulation episode.

On the evaluation trace plot, we see that the 3.b policy gives roughly the same results on average.

However, to conclude, the result in 3.b is better than in 3.a.

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

Running the best-performing policy π^*_{DQN} through the evaluation code in 2.b), generating the same histogram plots and comparing the results.

The Values above could be seen in the plot below.

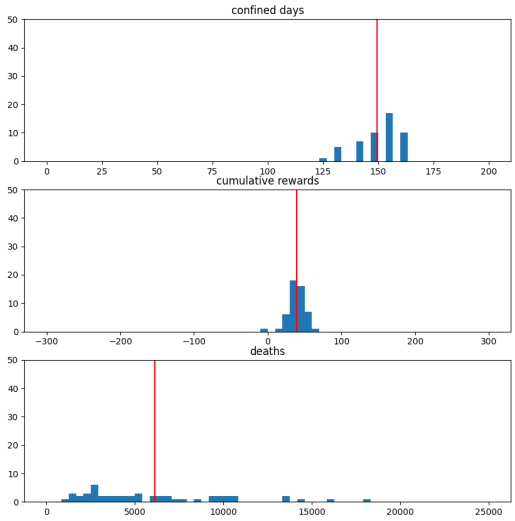

Figure 10: Evaluation of DQN decreasing exploration policy

Discussion: Compare the results. Did the reinforcement learning policy outperform Pr. Russo's, if so in what sense?

Yes, the reinforcement learning policy outperforms Pr. Russo's. Because the average number of deaths is way smaller (6000 vs. 60 000) and the average cumulative reward is way higher than with Pr. Russo's policy (40 vs.-80). But the number of confined days is slightly bigger(150 vs. 100).

4 Dealing with a more complex action Space

4.1 Toggle-action-space multi-action agent

Question 4.1.a) (Theory) Action space design

Why would one want to use such an action-observation space as the one above, rather than directly compute $Q(s, a)$ for each action? **Discuss, the impact on network architecture and on training.**

Answer:

Having such action space we might take several actions at the same time and not change the network output. But this can make the learning process slower and more unstable.

The change of the observation space, which includes the current state of each action as an observation, enforce us to have a larger input layer size.

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

Implementation of the toggled-action and observation spaces and training Deep Q-Learning agent on it.

Plots of the training trace (for all 3 training runs, distinguished with different colours) and the evaluation trace (averaged across 3 training runs).

Plot one of the episodes of the best Toggle Policy.

Discussion: Is the agent properly learning? Interpret the best policy π^*_{Toggle}

From the train log plot of the last episode, we can see that the agent learns how to quickly react in the environment. One can see the constant positive reward after a while on the same plot.

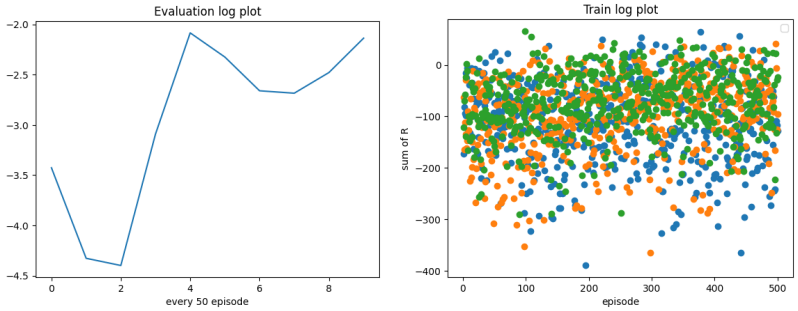

Figure 11: Training with toggle action space

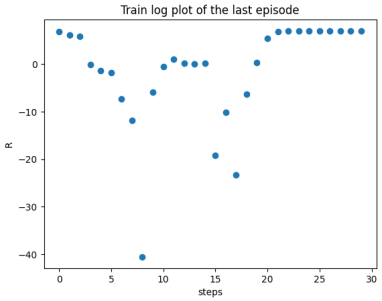

Figure 12: Episode of the best DQN Policy with with toggle action space

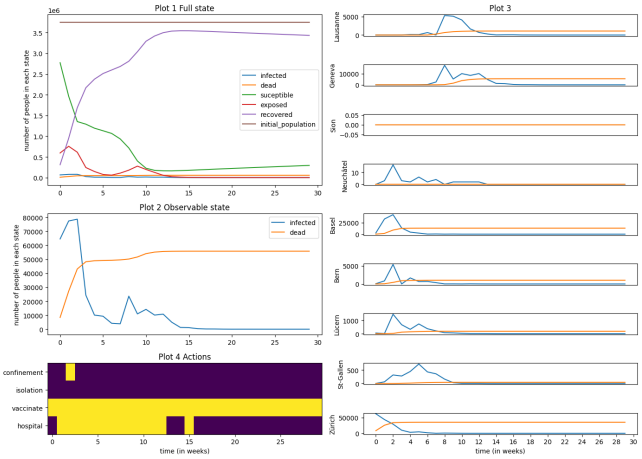

Figure 13: Additional plot DQN policy with toggle action space

Also, the epidemic seems to be mitigated, because it didn't reach all of the cities (Sion). However, the number of deaths is much higher (55K) than for the previous policy. One way to interpret the actions that were taken is that the agent wants to stop the spread of the epidemic using vaccination all the time, which increases immunity directly and makes people recover without them being infected. After the decrease in infection, the agent observes the situation and doesn't change the actions. The confirmation of this hypothesis can be seen in the plots above.

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

Evaluation of the π^*_{Toggle} policy trained in question 4.1.b), using the evaluation procedure that was previously defined.

Discussion: How does the policy perform compared to the binary action policy evaluated in question 3.c?

The toggle-action-space multi-action policy performs poorly compared to the binary one. Because the average number of deaths is higher(60K vs. 5K) and the average cumulative reward is way lower than with the binary action policy(-40 vs. 40). But the number of confined days is smaller on average(30 vs. 150).

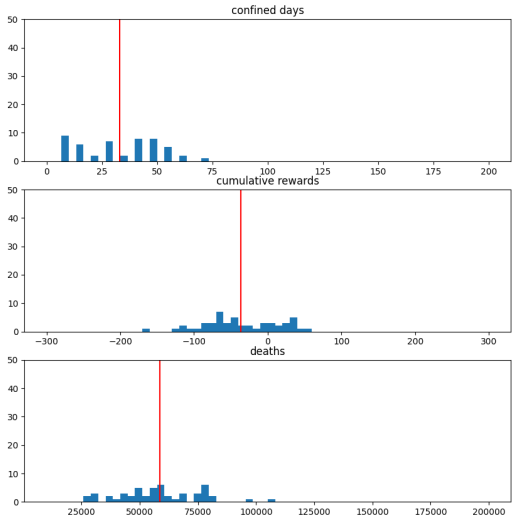

Figure 14: Evaluation of Toggle-action-space multi-action policy

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

What assumptions does the use of such a technique make on the action space? Could you think of an action space for which toggling the actions would not be suitable? Discuss.

Answer:

The main assumption that we made is that any action can be taken with any another one at the same time.

An example where this technique won't be suitable: Binary action space or spaces that have actions regarding the same attributes (Example: If we add to our action space: Reduce hospital beds). Considered policy will allow us to have actions to add and reduce hospital beds at the same time, which does not have much sense.

4.2 Factorized Q-values, multi-action agent

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

Implementation of the multi-action factorized Q-values agent and observation spaces and training Deep Q-Learning agent on it.

Plots of the training trace (for all 3 training runs, distinguished with different colours) and the evaluation trace (averaged across 3 training runs) for multi-action factorized Q-values policy and Toggle-action-space multi-action policy together.

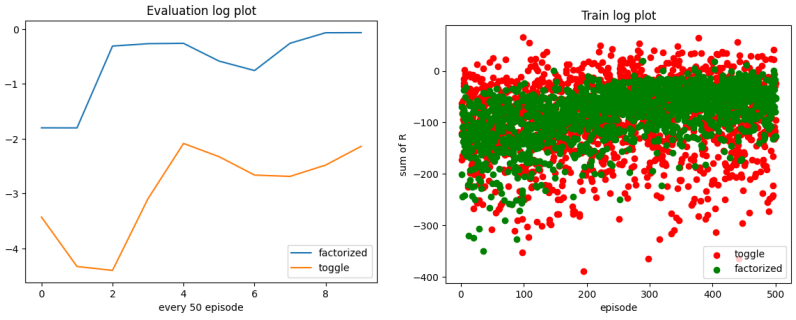

Figure 15: Multi-action factorized Q-values policy training

Plot one of the episodes of the best Factorized Q-values Policy.

Discussion: Does it successfully learn?

Looking at the small number of death($\leq$ 3000), we can conclude that the policy is performing well and successfully learn. Also, the epidemic didn't reach all the cities which is a good result. Looking at the evaluation log plot, we can see that the dynamic of rewards is similar to toggle action space policy, but with the difference of approximately 1.5. Since factorized Q-learning enables the independent representation and selection of several action, it can offer more degrees of freedom than a toggling action space.

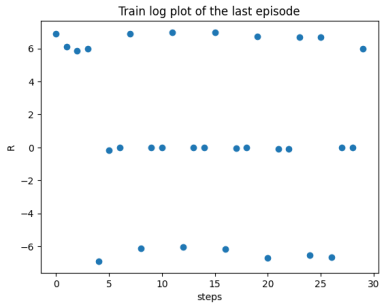

Figure 16: Episode of the best multi-action factorized Q-values policy

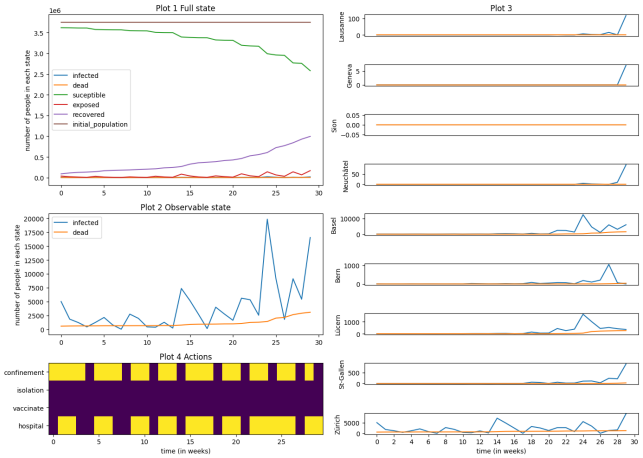

Figure 17: Additional plot multi-action factorized Q-values policy

Discussion: Is the policy realistic?

However, it doesn't think of a long-term perspective in relation to human immunity. One can't observe isolation and vaccination actions during the simulation episode. The main strategy is to confine the city and cure infected individuals. Looking at the plot 2 at Fig. 17, we can see increasing fluctuation of infected individuals, which indicates the inability of the policy to control the dissemination. So, it seems not realistic for us.

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

Evaluation of the best policy π^*_{factor} trained in question 4.2.a), using the evaluation procedure that we previously defined.

Discussion: How does it compare to the toggled policy?

Much higher number of confined days(150 vs.30). But the multi-action factorized Q-values policy succeeded to get a higher average cumulative reward(0 vs. -40) and a lower number of deaths(4K vs. 60K), which is a good indicator.

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

In question *4.2.a)*, you implemented a factorized-Q-value policy. **What assumptions does the use of such a technique make on the action space?** Could you **think of an action space for which factorizing Q-values would not be suitable**? Discuss.

Answer: For the factorized-Q-value policy, we assume that actions are independent. In the estimating final q-value we need the union of all Q-values of all chosen actions, but for practical purposes knowing that they are independent, we just sum them up). An example where this technique won't be suitable: Having dependent(overlapped) actions: Add beds to all hospitals and add beds to those hospitals that have expertise in treating infections.

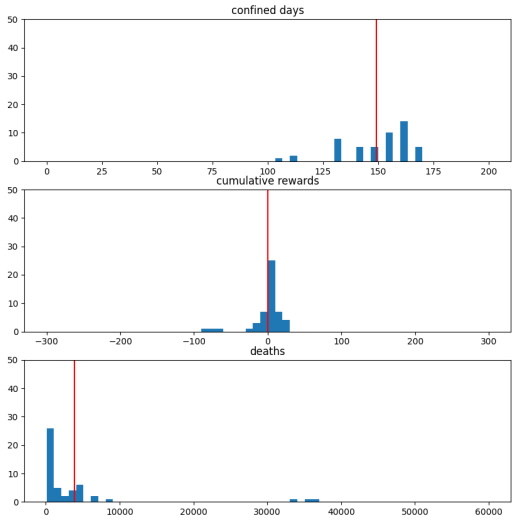

Figure 18

Figure 19: Evaluation of multi-action factorized Q-values policy

5 Wrapping Up

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

Compare the evaluations and training curves of **Pr. Russo's Policy**, **single-action DQN**, **factorized Q-values** and **toggled-action-space** policies. Discuss the performance differences, what do you observe? How do the two approaches compare ? What approach performs best? Why?

Looking at evaluations, we can see that Russo's policy performs quite poorly. But also, we observe that one Q-values based approach doesn't have also much success - with toggled action space. However, worth to notice, other DQN policies managed get quite good results. The first approach is implemented statically, but Q-values based policies aim to estimate Q-values of each action in each state. What we can conclude, that all DQN outperform Russo policy. Looking at the reward, we can conclude that binary DQN is the best policy.

Question 5.b) (Result analysis) Comparing policies

Running the evaluation procedure with each (π_{DQN}, π_{toggle}, π_{factor} as well as the original π_{russo}) trained policy (always pick the best-performing policy) for 50 episode.

Computed metrics (all of the averages are empirical means computed over the 50 episodes): the average number of **total confined days** avg$[N_{\text{confinement}}]$ (*lower is better*), the average number of **total isolation days** avg$[N_{\text{isolation}}]$ (*lower is better*), the average number of **total vaccination days** avg$[N_{\text{vaccination}}]$ (*lower is better*), the average number of **total additional hospital bed days** avg$[N_{\text{hospital}}]$ (*lower is better*), the average **number of total deaths** avg$[N_{\text{deaths}}]$ (*lower is better*), the average **cumulative reward** avg$[R_{\text{cumulative}}]$ (*higher is better*).

Policy	avg$[N_{confinement}]$	avg$[N_{isolation}]$	avg$[N_{vaccination}]$	avg$[N_{hospital}]$	avg$[N_{deaths}]$	avg$[R_{cumulative}]$
Russo	100.24				58071.94	-69.96
DQN	149.38				6113.82	39.37
Toggle	33.18	0.0	103.88	54.46	58697.28	-36.45
Factorized	149.24	0.0	0.0	136.92	3917.36	0.41

Discussion:

The best(green) and worst(red) performing policies with respect to each metric are emphasised in the table. Here we have to compare the pairs because the two policies don't have access to other actions but only to confinement. Here, DQN decreasing policy is better because the average number of deaths is way smaller and the average cumulative reward is way higher. However, the drawback is the higher average number of confined days, which led to efficient mitigation of the virus.

In comparison between the two multi-action policies, the factorized Q-values policy is better to be chosen. The average number of deaths is smaller in all 4 policies and the average reward is positive and higher than for Toggle multi-action policy. But factorized Q-values policy agent does many actions, especially confinement and adding more hospital beds.

Question 5.c) (Interpretability) Q-values

For both π_{DQN} and π_{factor}, produce a plot for visualizing the estimated Q-values for one episode. **Discuss your results.** How interpretable is your policy?

Answer:

It is easy to notice that Q-values almost in each state for binary DQN policy are very similar. Every 6-8 weeks sudden decrease of the Q-value of no confine action strongly prompted the selection of a confinement action. So, looking at this distinct pattern of actions taken we can conclude that this model has converged to a stable solution.

For Factorized Q-values multi-agent policy, it is worth noticing a quick increase in values for confinement and a decrease in vaccination. Looking at the big absolute value of the confinement factor (both taking and not taking the action), we can conclude that it is essential for the decision-making process. At the end of the episode, the increasing difference between the final Q-values of every factor indicates that it is better not to take any action. Thus, it is reasonable to assume that policy manages to mitigate the epidemic. Based on these observations and distinct patterns, it does not appear that the Q-values provided exhibit significant instability or a lack of convergence in the learning process.

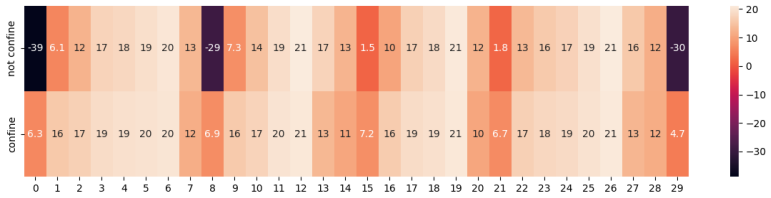

Figure 20: Heatmap: DQN policy

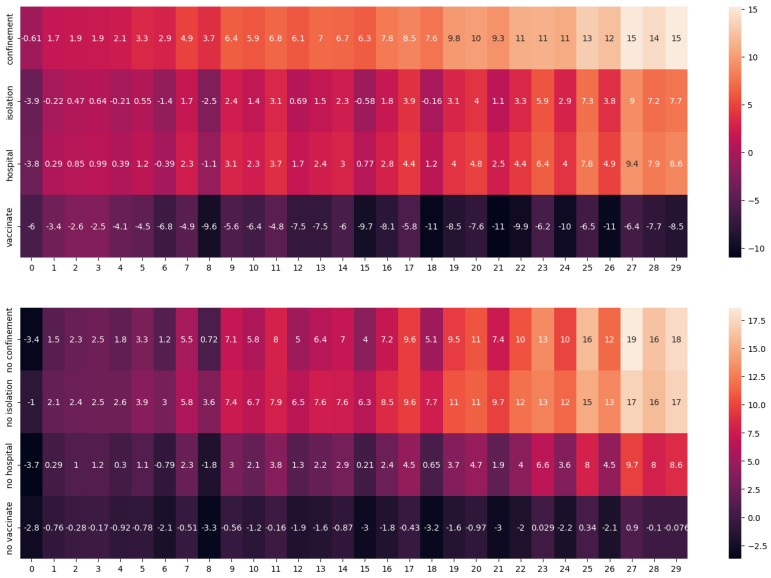

Figure 21: Heatmap: multi-action factorized Q-values policy

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

In the following project you have implemented different policies acting on the exact same environment with a different number of actions. **Is cumulative reward an increasing function of the number of actions?** (*In other words, does adding an action always yield a better reward?*)

Answer: Taking into the account the results of toggled-action space policy (binary DQN has higher average reward), we can conclude that cumulative reward is not always the increasing function of the number of actions.