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

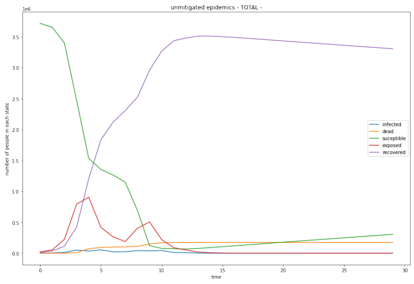
Maxence Hofer, Giacomo Mossinelli

May 2023

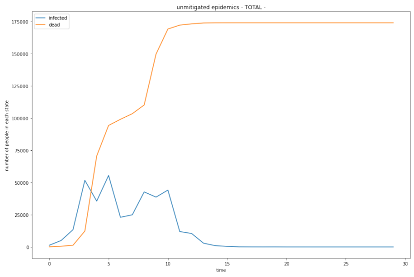
1 Introduction

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

In this section, we observe the behavior of the epidemic without any sort of mitigation. First of all, we focus on the information we can deduce from 1a. In this case, it is possible to notice that the number of susceptible people quickly reduces, since a large component of the population is splitted in the other categories already in the first weeks. As can be guessed, a very large portion falls into the category of recovered, but a not negligible increase in deaths can also be noted. The number of people exposed to the virus (as well as, in an extremely more moderate way, the number of infected people) shows the highest values in the first weeks, when the number of susceptibles is still high. Finally, it can be seen (also by looking at Figure 1b and 1) that the number of deaths experience the most significant rises when the number of infected is high, as well as declines during the quietest periods. This is an indication of the rationality of these plots, which seem to intuitively and convincingly model the spread of the epidemic.



(a) $s_{total}^{[w]}$,$e_{total}^{[w]}$, $i_{total}^{[w]}$, $r_{total}^{[w]}$ and $d_{total}^{[w]}$ over time.



(b) $i_{total}^{[w]}$ and $d_{total}^{[w]}$ over time.

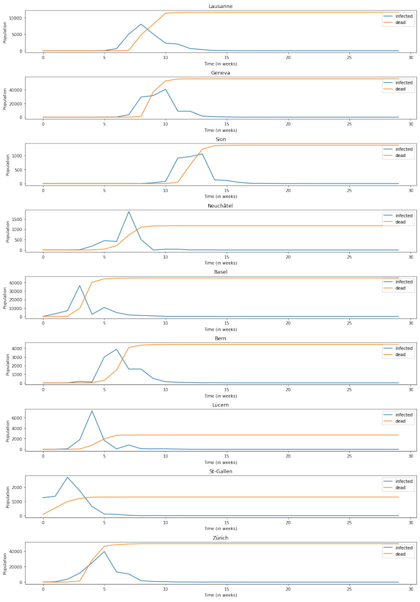


Figure 1: $i_{city}^{[w]}$ and $d_{city}^{[w]}$ for each city

2 Professor Russo's Policy

The Professor Russo's policy is pretty simple. It states that if there are more than 20000 infected at the end of the week, there must be a 4-week long period of confinement.

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

Figure 2 shows the main information on the spread of the epidemic and the actions taken accordingly. Firstly, it can be seen that the policy has been respected: the confinement periods last four weeks and begin the week following an observation of 20,000 infected. The algorithm seems to fulfil its purpose when compared to the unmitigated case. In particular, the final number of deaths is much lower (about one third) and the susceptible

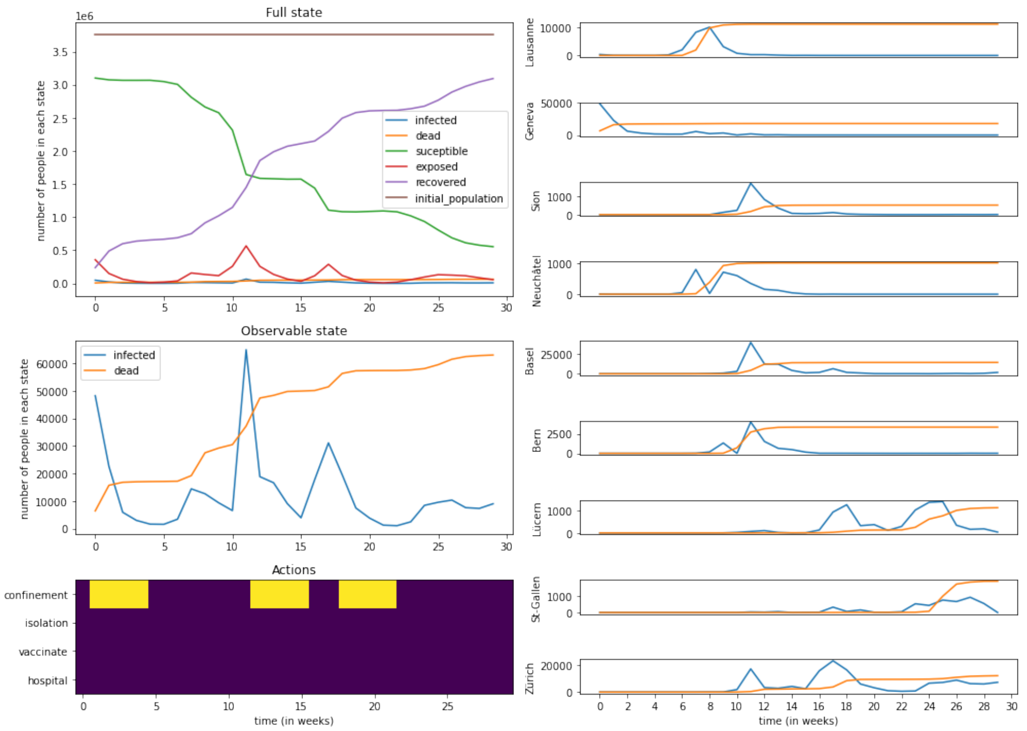


Figure 2: One episode, Pr.Russo's policy

portion of the population falls much more slowly, which is indicative of the lower exposure to the virus (which is also confirmed by the "exposed" variable, which is smaller than in the previous case). It is also worth noting that the quantity of infected people tends to be lower, which leads to a less rapid and more moderate increase in the curve of deaths.

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

We plot histograms (see Figure 3) which collect the total number of confined days, the cumulative reward and the number of deaths of 50 episodes. It is also possible to observe the average values of these quantities (denoted by the red vertical line), which shows how the average number of deaths through these 50 episodes is 58922.9, the average number of days in confinement is 98.98 and finally the average reward is -70.28965721607209.

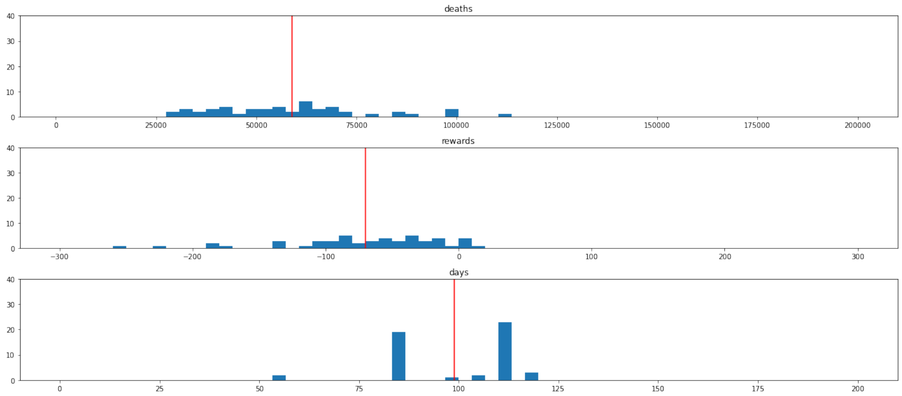
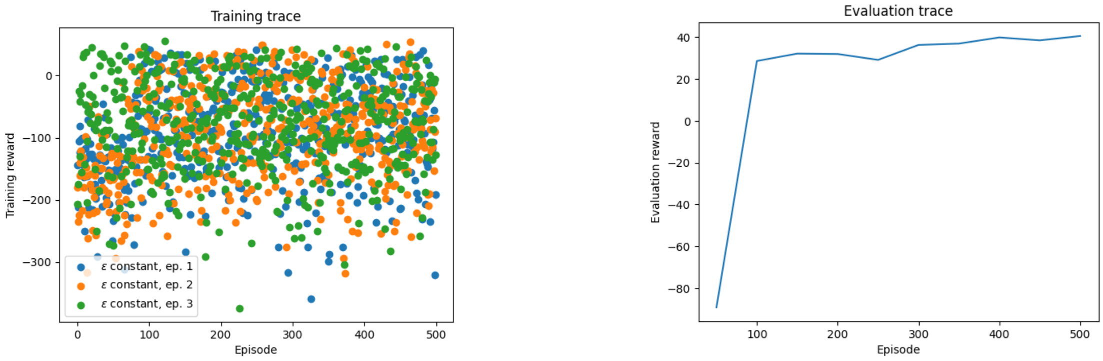


Figure 3: Histogram of collected values, Pr.Russo's policy

3 Deep Q-Learning with a binary action space

Question 3.a) Implementing Deep Q-Learning

In this section, we implement the DQN agent. Figure 4a shows the training trace (defined as the total reward accumulated on an episode) represented for three different training processes. Each 50 episodes, we calculate an average reward over 20 episodes with $\epsilon = 0$. This is shown on figure 4b. We can observe that, even though the training trace looks like a random cloud of points, we quickly obtain good evaluation reward.



(a) Training trace, $\epsilon = 0.7$

(b) Evaluation trace, $\epsilon = 0.7$

Figure 4: Traces for point 3a

We also plot the results on one episode using the learned policy. This is represented in figure 5. We observe that the agent confine when the number of infected people rise. The policy looks meaningful, even though the number of confinements is quite high. One way to improve this would be to give a lower value to the confinement in the calculation of the reward.

Question 3.b) Decreasing exploration

In this section, we compare the results obtained previously with epsilon defined as

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

Where t is the number of the episode, $T_{max} = 500$, $\epsilon_0 = 0.7$ and $\epsilon_{min} = 0.2$. Fig. 6a represents the training trace and Fig. 6b represents the evaluation trace as defined above. We observe that the training trace is slightly better for the simulation with ϵ varying. However, the results look similar for the evaluation trace. This is due to the fact that with ϵ constant the next action in the training episodes is 70% of the time chosen randomly, while between 70% and 20% with the ϵ varying. On the contrary, in the case of the evaluation, the next action is chosen deterministically, so it does not make a difference between both ϵ because they are trained on the same number of episodes.

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

In our case, the best obtained model is the one corresponding to the third training process with constant ϵ. As in section 2, we plot the histograms (Figure 13) to evaluate the number of deaths, the reward and the days of confinement. If we use the number of deaths as a term of comparison, this result is better when compared to the previous policy. Not only the histogram qualitatively shows this fact, but there is also the mean value to prove it (the average number of deaths is 4320.72). This result is also achieved due to the longer confinement time imposed by this policy, which no longer has to respect the 4-weeks limit imposed by the previous method (the average number of confined days is 154.28). Finally, it can be seen that the value of the reward is significantly higher (average value is 40.08507274627686). Overall, it can be said that this policy tends to achieve more satisfactory results than Pr.Russo's policy.

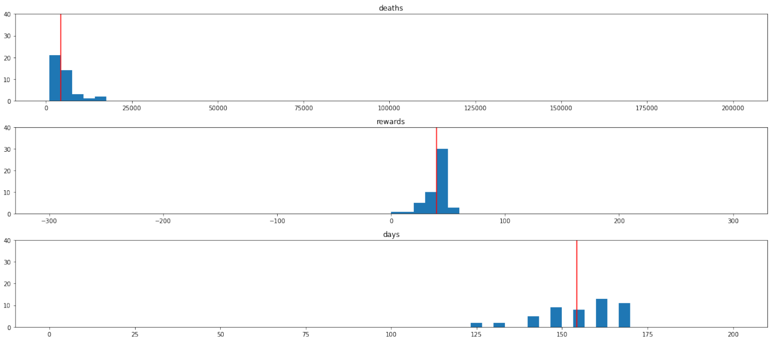


Figure 7: Histogram of collected values, DQN policy

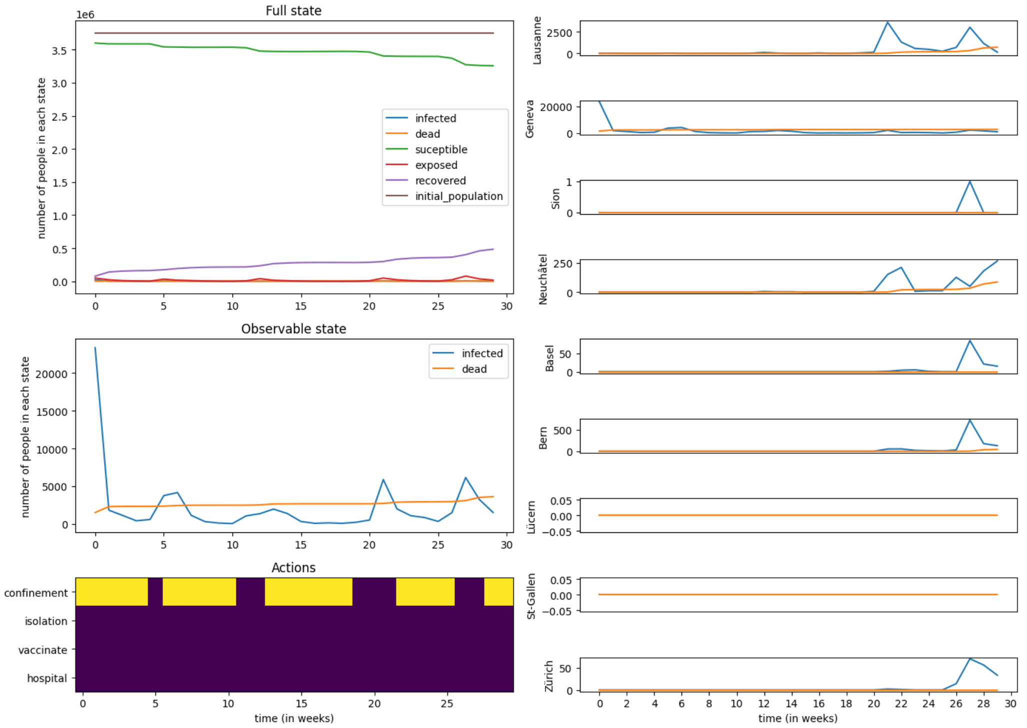
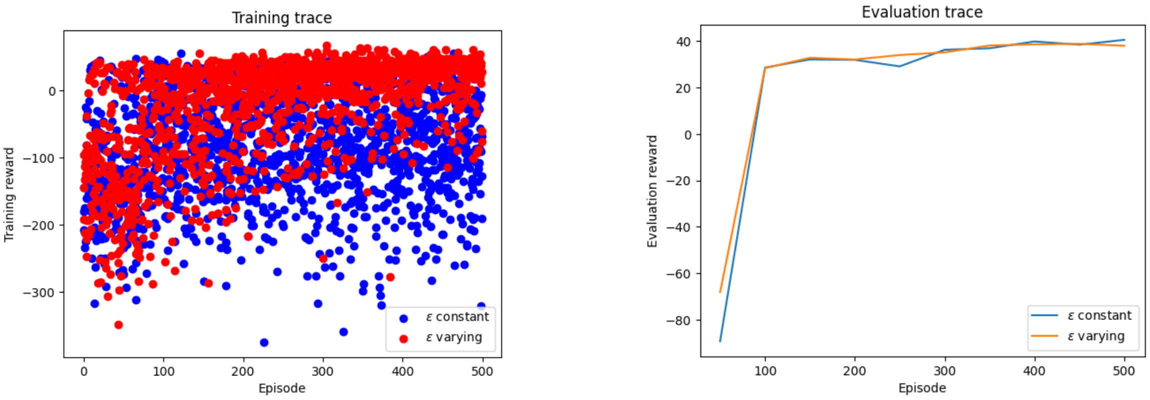


Figure 5: One episode, single action, $\epsilon = 0.7$



(a) Training traces

(b) Evaluation traces

Figure 6: Comparison of training and evaluation traces for two different types of ϵ

4 Dealing with a more complex action Space

In this part, the learning rate was asked to be imposed to 10^{-5}, but, after trying with several values, we have observed that the results were much better with a learning rate equal to 10^{-3}, especially for what concern part 4.2, where the results were unstable with the lower learning rate. Therefore, it has been decided to use 10^{-3} as learning rate.

4.1 Toggle-action-space multi-action agent

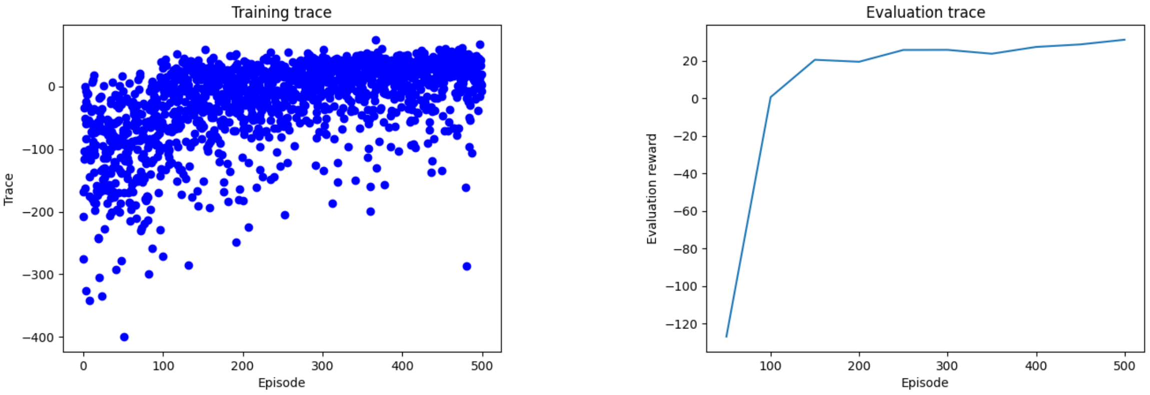
In this section, we present a method which makes possible to choose one action at a time between five different actions (do nothing, confine, isolate, vaccinate or add hospital bed). At each step, the agent is only allowed to choose one action among them, and when this happens, the state of the action will be changed (for example, if the agent picks confine and confine was already True, it will be changed to False).

Question 4.1.a) (Theory) Action space design

The main difference between this implementation and another one, for what concern the calculation of the different Q-values, is that the Toggle method requires a smaller neural network output, which is 5 in our case and would be 16 (2^4, all the possible combination of the actions) in the case of the naïve action space with four different actions. This will improve training, as the number of parameters of the neural network to be adjusted is smaller.

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

In this section, we present the results obtained using the aforementioned method. Figure 8 illustrates the training and evaluation traces, as defined in section 3. We observe a slightly lower best evaluation reward compared to the previous section with the binary action space. This difference is likely due to the limited training time, as the neural network has not had enough episodes to reach optimal values. We also observe that the evaluation trace continues to show improvement over 500 episodes. It would be intriguing to explore whether a higher number of episodes can yield a higher value than the binary action space. Nevertheless, the agent demonstrates proper learning behavior.



(a) Training trace, ϵ varying

(b) Evaluation trace, ϵ varying

Figure 8: Traces for point 4.1

We can also plot one simulation using the best policy learned. This is shown in figure 9. We observe that the policy does seem meaningful, as it is a combination of the different actions. At the same time, we observe that it corresponds to a toggle policy, i.e. only allowing one change between two episodes. However, we could expect to obtain better performance with a longer training process.

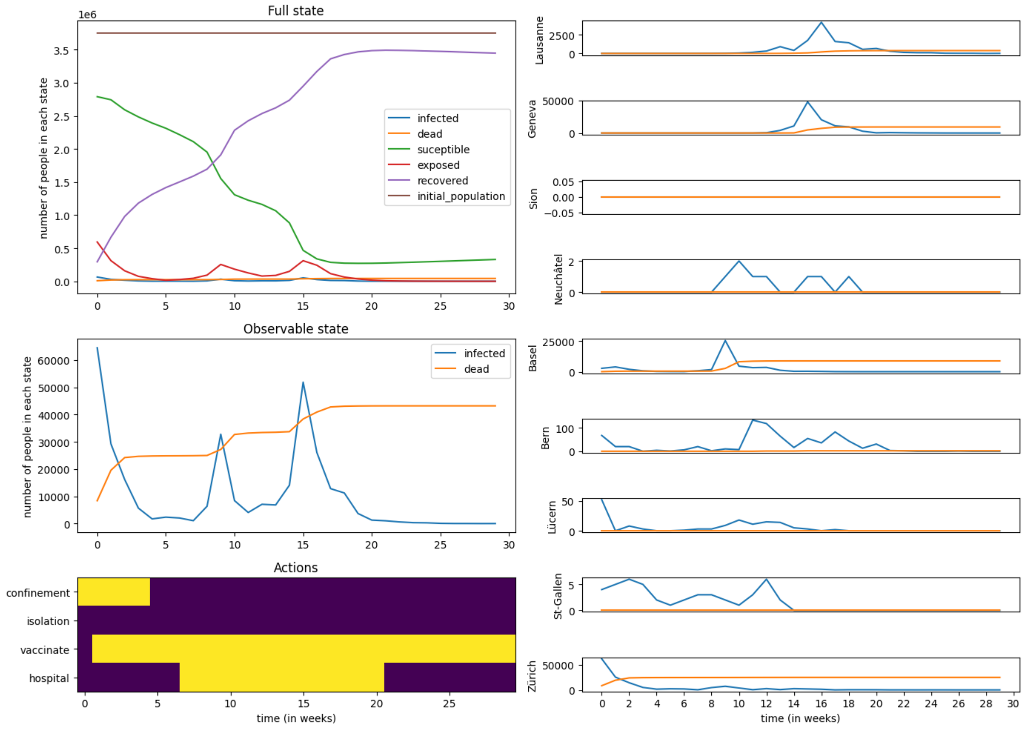


Figure 9: One episode, toggle action space, ϵ varying

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

Also in this case, histograms are plotted to evaluate the policy. It is possible to notice that the toggle policy presents a different amount of confinement actions depending on the episode, even if the average number of days is similar (average confinement days: 132.16). With respect to the previous policy, whose histograms are described in 3, the average number of death is strongly higher (12042.1), but the average reward is comparable to the previous one (more precisely, the value is 35.70223335266113). It is again possible to notice that, depending on the episode, we can expect fairly wide differences in these values. Considering the results represented in Figure 10, it is possible to say that the policy is performing worse than the binary action policy. The weakness of this policy could lie in the fact that the introduction of all actions could make the procedure more complex, thus requiring more training. Furthermore, being able to change the value of only one action at a time limits the possibilities provided by the actions introduced.

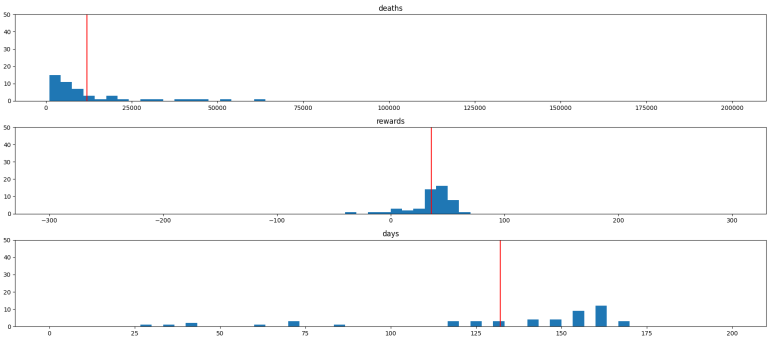


Figure 10: Histogram of collected values, toggle-action multi-action policy

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

We are assuming that we can only change the state of an action at a time but probably in this case, it would be better to have the possibility to have more than one change of action each time. Another example where it could be important to have this possibility would be the case of a game, where the player could need to use two actions at the same time (imagine a video game, where the player need to move and shoot at the same time).

4.2 Factorized Q-values, multi-action agent

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

In this part, instead of defining a toggle action space, we use a 4x2 output of our neural network. The Q-values can therefore be defined for a set of decision on each action σ as

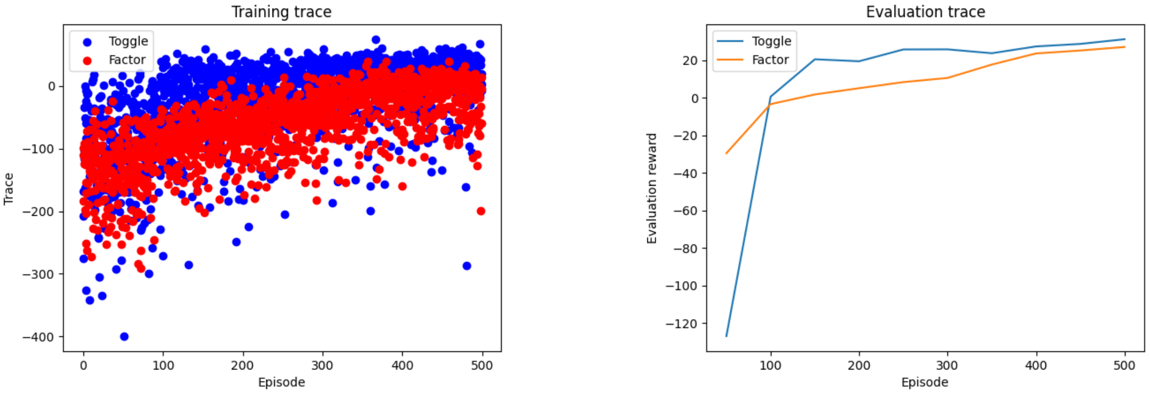
$$Q(a^{[w]}, s) = \sum_{\sigma \in decisions} Q(a_\sigma, s) \tag{2}$$

With $Q(a_\sigma, s)$ the output of the neural network corresponding to the decision σ. The training and evaluation traces are presented in figure 11, compared with the toggle policy. The agent seems to be properly learning, as the evaluation trace is increasing. We also observe that the evaluation traces is still increasing after 500 episodes, as in the case of the toggle action space, therefore a longer training process may yields better results.

As in the previous point, we can also plot one episode using the best trained policy. This is represented in figure 12. The policy seems to give a meaningful policy, using a combination of adding hospital bed and confine.

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

In this case, it is possible to observe a similar behavior to the toggled policy described in section 4.1. It is still possible to detect a variable value of the confinement days depending on the episode considered. It can suggest that, depending on the situation, the policy is able to indicate different strategies to deal with the epidemic. On average, 143.64 days are devoted to confinement, according to this policy. The average reward is very similar too (more precisely, 24.89776870727539), while the number of deaths is lower if compared with the previous case (the average here is 9924.48), but still higher than π_{DQN}.



(a) Training trace **(b)** Evaluation trace

Figure 11: Traces for point 4

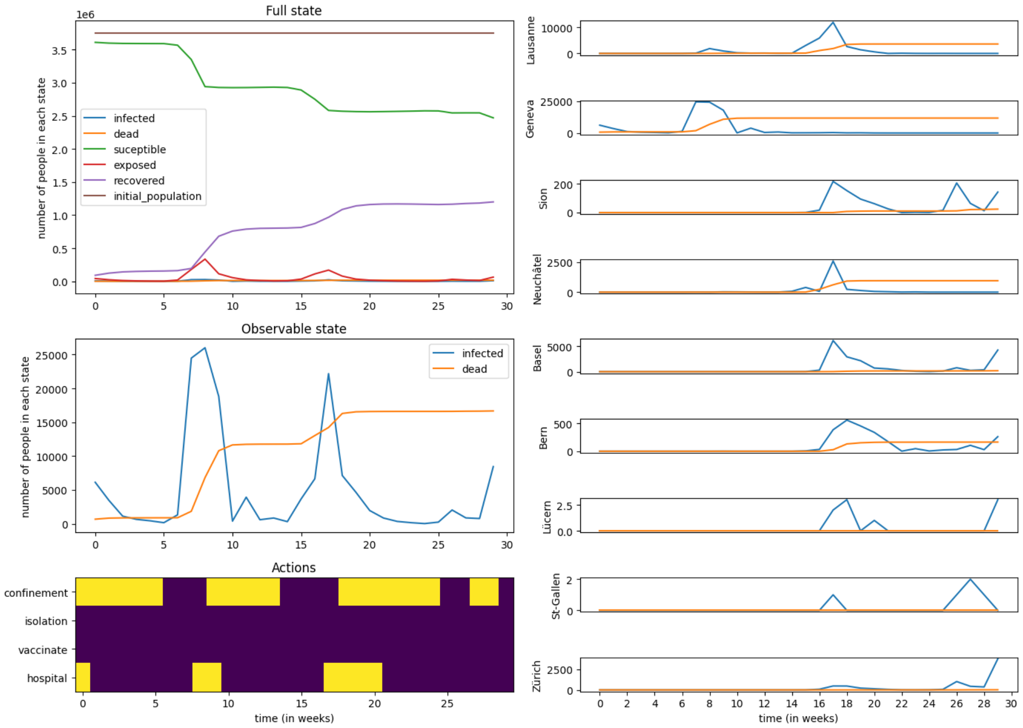


Figure 12: One episode, multi action factorized, ϵ varying

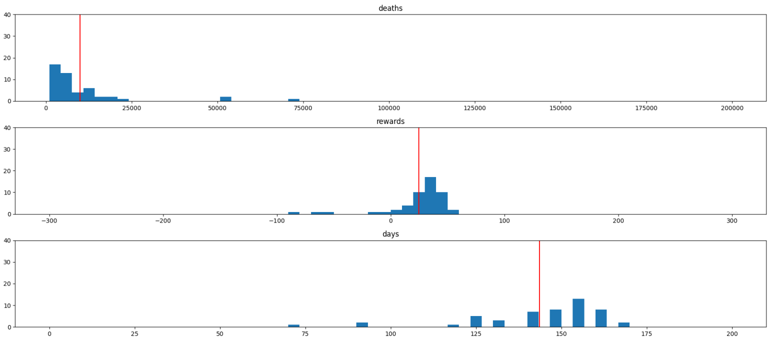


Figure 13: Histogram of collected values, multi-action factorized Q-values policy

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

We assume that we can measure Q-values independently between each of them. It is not the case in practice, because any action will have repercussion on the others (if we confine, we will maybe not need to add hospital bed). An example of action that are dependent would be for example a driving simulation, the steering angle, throttle, and brake inputs are interdependent and affect the vehicle's behavior in a combined manner. Factorizing the Q-values into separate components would overlook the complex interactions between these actions. For instance, the effect of steering angle on the vehicle's trajectory is influenced by the throttle and brake inputs.

5 Wrapping Up

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

In order to compare the training behavior, we can plot the evaluation trace for each policy (with Russo's evaluation trace as a constant with an average reward calculated on 50 episodes, because there is no learning in the Russo's policy). Figure 14 represents these evaluation traces. We observe that all policies are better than Russo's policy at the end of the training. We also observe that the π_{DQN} policy goes quickly to high values of reward, which is not the case for the multi action space.

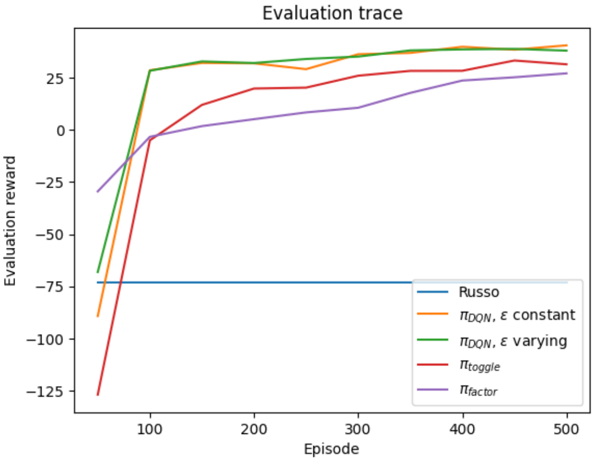


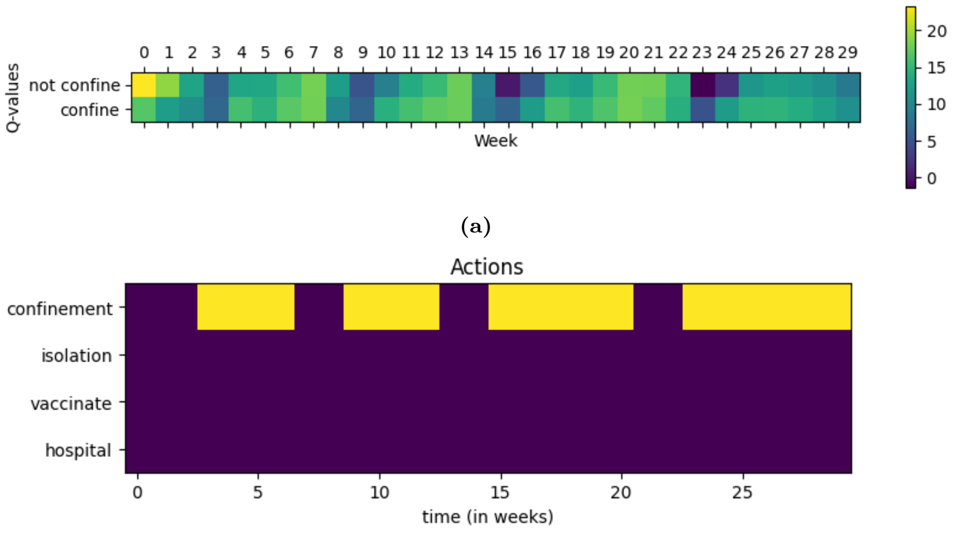
Figure 14: Evaluation traces for all policies

Question 5.b) (Result analysis) Comparing policies

In order to definitively compare policies, we report table 15 that explicitly compares the key metrics of the problem. In particular, it can be seen that:

- the policy with lower confinement days is π_{Russo}
- each policy never choose "Isolation"
- the policy with lower additional hospital bed days is π_{factor}
- the policy with lower vaccination days is π_{factor}
- the policy with lower number of deaths is π_{DQN}
- the policy with higher cumulative reward is π_{DQN}

Please note that for some actions (namely, "Isolation", "Hospital" and "Vaccination") π_{Russo} and π_{DQN} have not been considered, because they are not able to perform those actions. Looking at the results, it is possible to see that π_{factor} presents the best values for 2 metrics, more precisely in the choice of 2 actions. This is probably influenced by the fact that, since it has a wider choice than other policies, and since it can choose multiple options at the same time, it has the possibility of distributing the choice creating the most appropriate mix without particular constraints. On the other hand, the policy that performs best in terms of number of deaths and rewards is π_{DQN}. This is probably due to the simplicity of the model, which allows for quick but effective training, and also to the fact that the only possible action is confinement, which, by reducing the exposure to the virus, reduces the number of deaths. It is relevant to note that the values identified are similar to those described in the previous evaluation paragraphs. This is an indication that the actions taken into consideration do not depend on the seed we are using. Another important observation is that π_{toggle} is the policy which assumes the wider variety of actions. The results suggest that, if the training were longer in π_{toggle} and π_{factor} cases, then theycould have been better than π_{DQN} case. Having investigated the hyper-parameter gives better

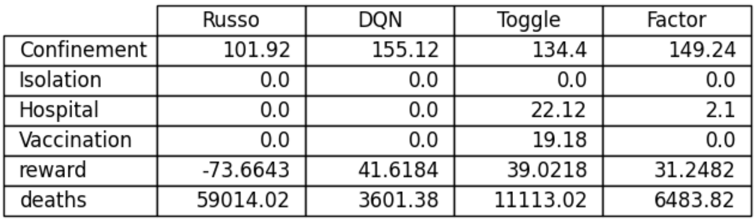


(a)

(b)

Figure 16: Q-values for π_{DQN} and corresponding episode

results and speeds up the process, but probably the complication of these models, which is much higher than in the previous case, makes the training phase slower. This makes these more elaborate cases comparable to the simpler DQN case in our tests.



	Russo	DQN	Toggle	Factor
Confinement	101.92	155.12	134.4	149.24
Isolation	0.0	0.0	0.0	0.0
Hospital	0.0	0.0	22.12	2.1
Vaccination	0.0	0.0	19.18	0.0
reward	-73.6643	41.6184	39.0218	31.2482
deaths	59014.02	3601.38	11113.02	6483.82

Figure 15: Table collecting metrics for each policy. Note that Russo's and DQN policies do not have the possibility to perform "Isolation", "Vaccination" and "add hospital beds" actions, so their values is always identically equal to 0

Question 5.c) (Interpretability) Q-values

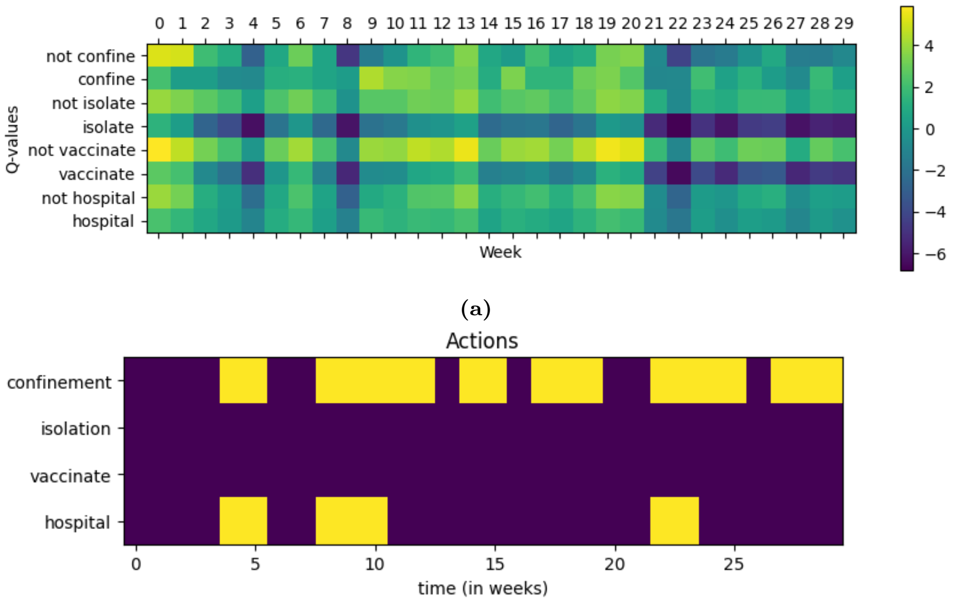
Figures 16a and 17a represent the Q-values for two episodes using respectively the DQN policy and the multi-action policy. Figures 16b and 17b represent the associated action. We observe that both figures are coherent, i.e. the action is taken if its Q-value is higher than the Q-value of not taking the action.

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

The relationship between the cumulative reward and the number of actions in a DQN network can vary depending on several factors. It is not necessarily true that the cumulative reward will always be an increasing function of the number of actions.

The number of actions in an environment affects the complexity of the decision-making process for the DQN network. With a larger action space, the agent has more choices to make at each step, which can make the learning problem more challenging. It may take longer for the agent to explore and find optimal actions, resulting in slower convergence and potentially lower cumulative rewards.

However, having a larger action space can also provide more opportunities for the agent to find better actions and explore alternative strategies. In some cases, a larger action space may allow the agent to discover



(a)

(b)

Figure 17: Q-values for π_{factor} and corresponding episode

more efficient paths to higher rewards, leading to an increasing function between the number of actions and the cumulative reward.