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

Mohamad Fakhouri (Sciper number: 336125) Laurynas Lopata (Sciper number: 315134)

Question 1

1.a

1) We can see in 1 that over the 30 weeks, the number of susceptible people decreases rapidly while on the opposite end the number of recovered increases. This follows the fact that recovered people are not susceptible anymore hence the inverse correlation. We can also see a slight bump in the number of exposed people over the first couple of weeks, coinciding with the pandemic's beginning.

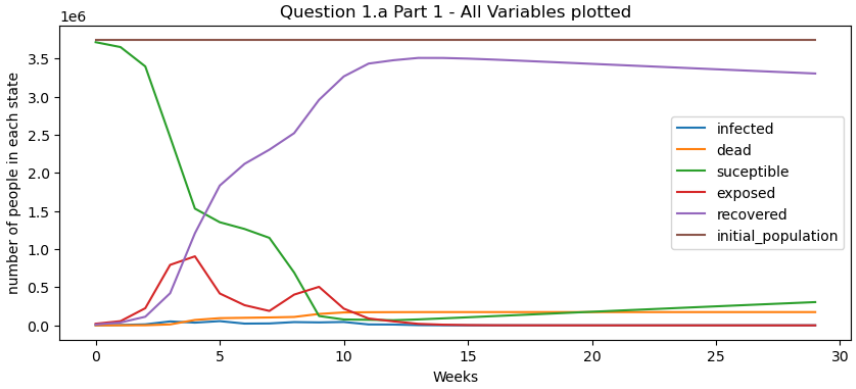

Figure 1: Plot of all variables over 30 weeks without any mitigations

2) As expected we can see that the rise in infected precedes the rise in deaths. In other words, there is a lag between infection and death. Furthermore, we can observe that over time as the infected population decreases towards zero while the number of deaths converges to a constant value.

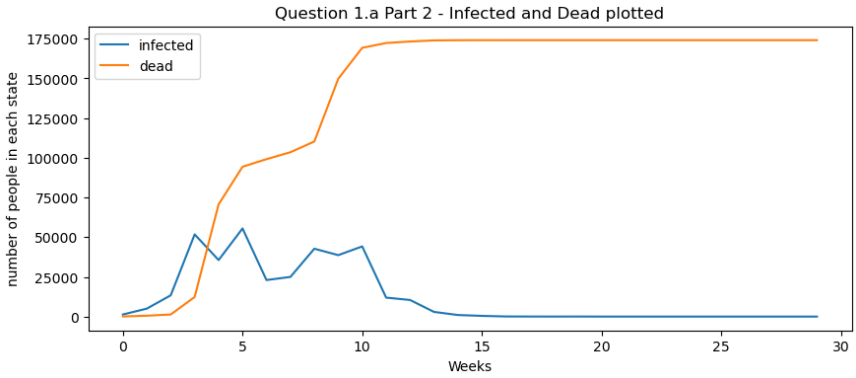

Figure 2: Plot of the number of infected and dead over 30 weeks without any mitigations

3) Similarly to the previous plot, we can observe the expected behaviour where infections preceded deaths and both converge to constant values over time.

Question 2

2.a

1) Compare to Question 1 in 4 we can see that the number of susceptible people decreases less rapidly and conversely the number of recovered increases was not as substantial as before. We also note that the three peaks in the number of exposed signify that the number of exposed rose after the end of the confinement periods

2) The effect of the policy is also clearly observable in 5, where we see three bumps just before the confinement was initiated and sharp decreases after its inception

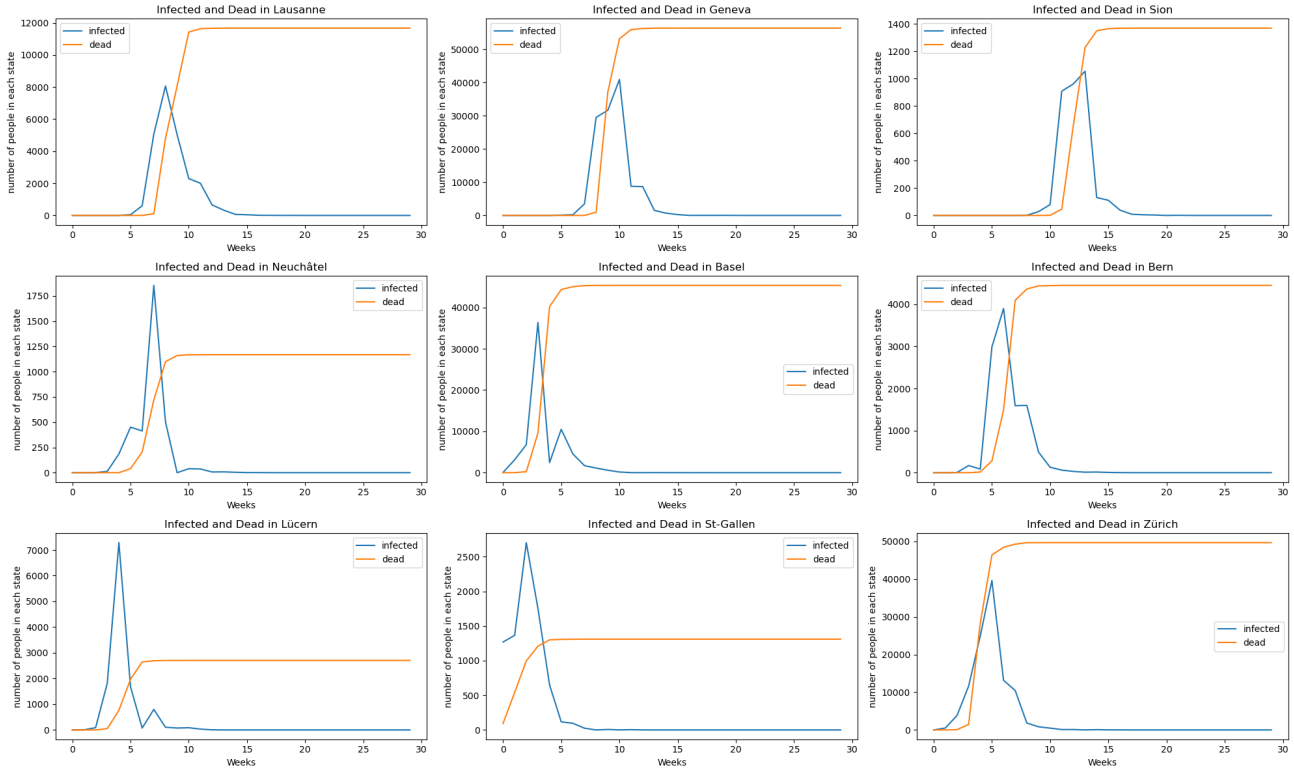

Figure 3: Plot of the number of infected and dead over 30 weeks across all cities without any mitigations

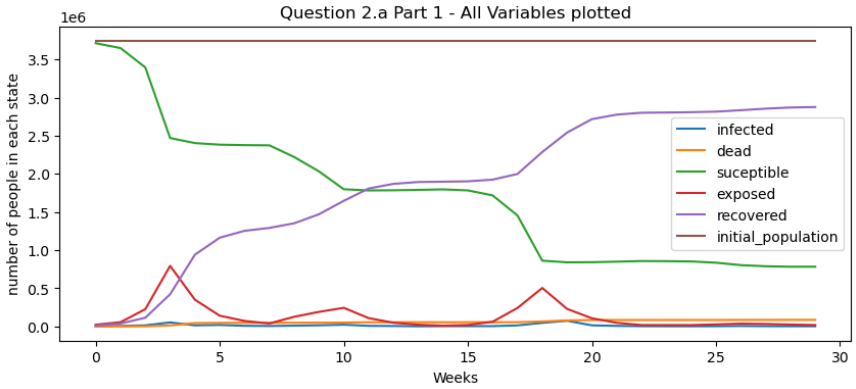

Figure 4: Plot of all variables over 30 weeks using the π_{Russo} policy

3) In 6 we see the city plots. In these plots, we see similar developments with multiple peaks followed by sharp declines in infected people. We note that these peeks are more pronounced in German-speaking cities compared to their French counterparts.

4) In 7 we see the Russo policy in action. As expected in confines the week after the number of infected hits 20'000 (gray dashed line).

2.b

We can see from 8 and 9 the histograms, the means and the deviations of Total days confined, Total dead and Cumulative rewards.

Question 3

3.a

In 11 we see the actions the policy chooses. From it, we can see that the policy chooses to confine after a sharp increase of infected patients and after a sufficient reduction of infection, the confinement is stopped.

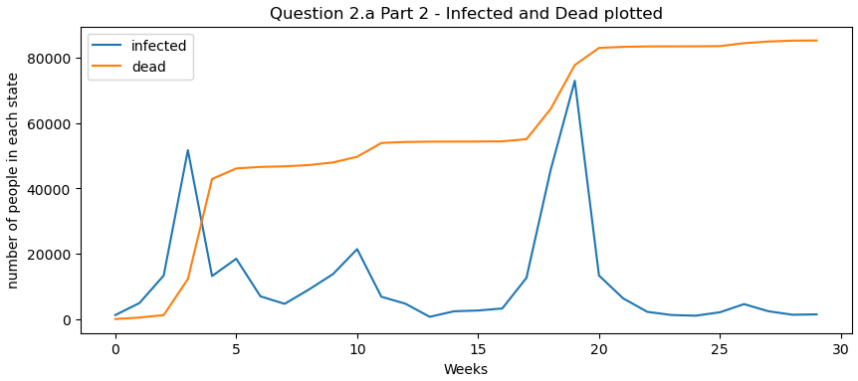

Figure 5: Plot of the number of infected and dead over 30 weeks using the π_{Russo} policy

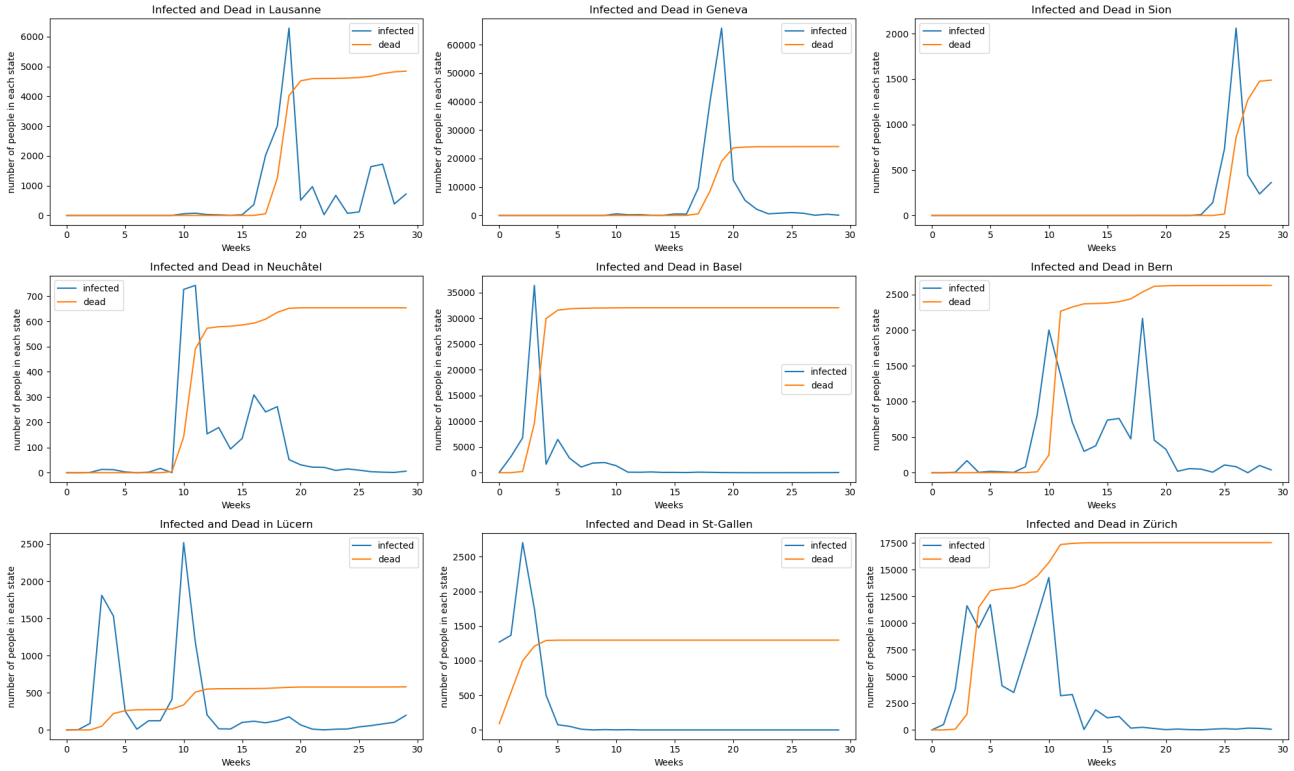

Figure 6: Plot of the number of infected and dead over 30 weeks across all cities using the π_{Russo} policy

3.b

From 10 and 13 we can see that the training rewards converge and are more stable towards the end of the training. We conclude that π_{DQN} with decreasing exploration outperforms π_{DQN} without decreasing exploration. This was expected since over-training reducing exploration reduces suboptimal actions. Another reason is that DQN is an off-policy algorithm that uses an experience replay buffer to learn from past experiences. Without decreasing exploration, the agent might continue to explore excessively, even after it has already learned a good policy.

3.c

It is evident that this is a meaningful policy and it achieves a significantly higher reward than π_{Russo}. It would confine the country with a lower case count compared π_{Russo} and would also confine for longer in some scenarios. Compared to π_{Russo} π_{DQN} reduced the number of deaths.

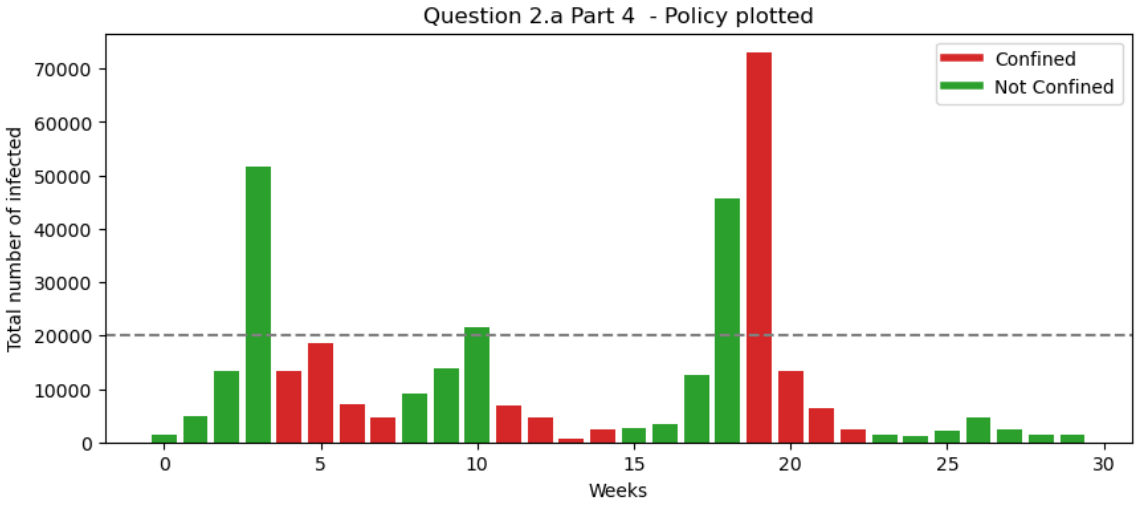

Figure 7: Plot of the action the π_{Russo} policy chose over an example episode

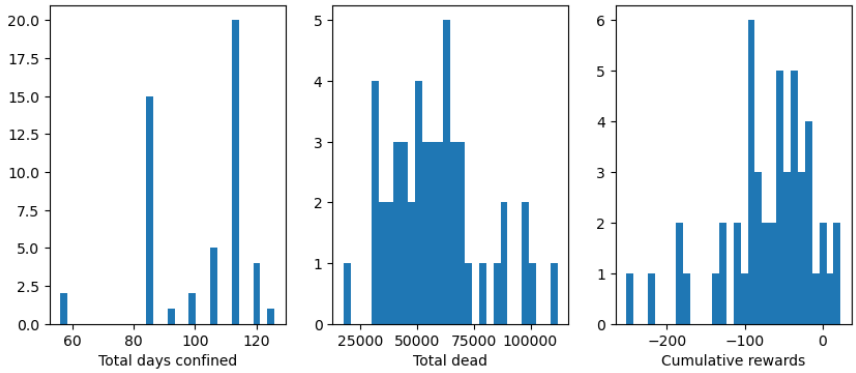

Figure 8: Plot of Total days confined, Total dead and Cumulative rewards histograms of the π_{Russo} policy simulation

Question 4.1

4.1.a

The main advantage of using the described action-observation space is that from the observation the model sees the previous actions taken so it can potentially learn the announcement costs of imposing a new measure and minimise them. Comparing the network architecture to the previous question we conclude that this action space leads to a more complex model since the output dimension is higher. This impacts training as well since the model is more complex training it for longer or with a finer training step is required.

4.1.b

From 15 we can observe that the model is indeed training as the average evaluation reward is increasing and converging while the training loss is noisy at the beginning but there's a clear downward trend and convergence.

In 16 we can see the actions taken by π_{toggle}. We can observe that the model would only take confinement and hospitalisation actions (one possible explanation for that would be that the model did not learn how to make use of the vaccination and isolation actions, or that they are too expensive for it to decide as useful. Moreover, the effects of vaccinations are long-lasting, and the agent cannot know how long ago it took an action). In the plot, we see three spikes of infections. All of these spikes are followed by the model confining and hospitalising, the strongest action it would take to combat the spread. We also note that in this particular episode, hospitalisation was almost always active. This can be explained by it being a cheap action to take while contributing positive rewards.

4.1.c

From 14 and 17 we can see that π_{Toggle} significantly outperforms πDQN in terms of the number of deaths. It is evident from the histogram since π_{Toggle} is skewed more to the left and have lower maximum values.

Figure 9: Policy comparison table. Bold is the best performing algorithm (using the mean as a metric)

Variable	Russo Policy	DQN Policy	Toggle Policy	Factor Policy
avg$[N_{conf}]$	$\mathbf{100.52 \pm 16.07}$	154.42 ± 8.78	157.5 ± 9.82	145.6 ± 13.5
avg$[N_{isol}]$	-	-	**0**	0.7 ± 2.88
avg$[N_{vacc}]$	-	-	**0**	0.42 ± 2.17
avg$[N_{hosp}]$	-	-	36.1 ± 65.4	$\mathbf{31.22 \pm 16.57}$
avg$[N_{dead}]$	$58,242.82 \pm 19,936.21$	$3,656.28 \pm 2,996.57$	$\mathbf{2,849.64 \pm 1,992.702}$	$10,121.92 \pm 10,726.53$
avg$[R_{cumul}]$	-69.96 ± 58.83	36.27 ± 9.76	$\mathbf{41.82 \pm 13.58}$	26.89 ± 28

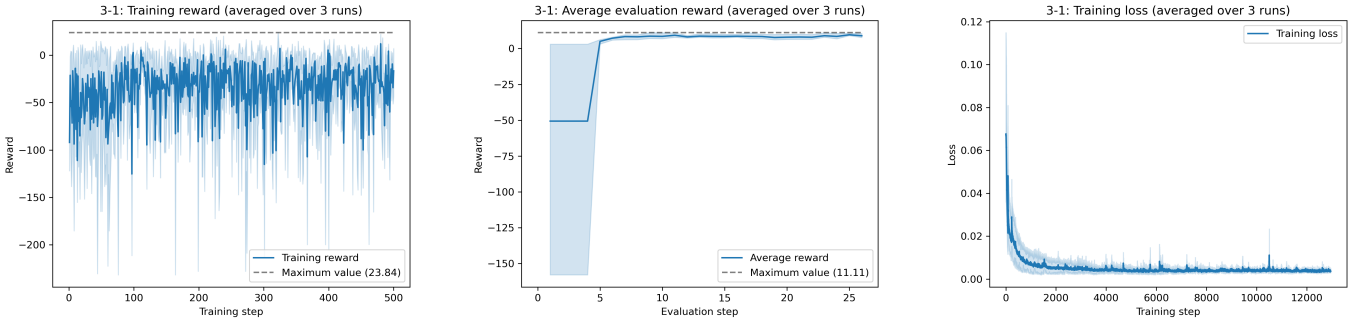

Figure 10: Plot of Training rewards, evaluation rewards and training loss of the π_{DQN} policy without decreasing exploration

4.1.d

The main assumption that this approach makes is that the action space consists of a set of discrete actions that can be toggled on and off. A concrete example of an action space where this approach would struggle can be described in the following way: Consider a 2D grid where a robot attempts to reach some target. In this case, toggling the actions would not be appropriate because the actions are not independent, and activating or deactivating individual actions would not make sense. The other strong assumption that it makes is that the agent can only change one action at a time (it cannot go from no action to confining and hospitalizing in one step). This is quite limiting, because in some cases due to an extreme spread in infected ratio, the agent needs to both confine (to stop the epidemic from spreading) and adding hospital beds (to heal the infected people) at the same time. The current Toggle space agent cannot do that, however, and it is quite limiting.

Question 4.2

4.2.a

From 18 we can make multiple observations. For starters, since the action space is more complex we trained the model for 600 episodes. However even after increasing the episode count we observe that the policy does not converge. This can be seen from the evaluation reward and train loss, both of them are substantially noisier than π_{DQN} or π_{Toggle}. This can be attributed to insufficient hyperparameter optimisation. Even though the model did not converge, the trend was positive and we can state that successful learning was present. We also note that this model can do at least what the DQN model can do, and it underperforms that, which supports our hypothesis that the model has not been optimized perfecrly yet. From 18 we can make multiple observations. For starters, since the action space is more complex we trained the model for 600 episodes. However even after increasing the episode count we observe that the policy does not converge. This can be seen from the evaluation reward and train loss, both of them are substantially noisier than π_{DQN} or π_{Toggle}. This can be attributed to insufficient hyperparameter optimisation. Even though the model did not converge, the trend was positive and we can state that successful learning was present. We also note that this model can do at least what the DQN model can do, and it underperforms that, which supports our hypothesis that the model has not been optimized perfectly.

In 19 we can see an example episode and the actions taken by π_{Factor}. In the beginning, no action is taken while the case count remains low, when it gets closer to 10 000 the confinement action is taken. Later another spike in infections next to confinement hospitalisation is added. Finally, we see a 3rd spike that's substantially bigger than the previous ones around week 20. The model here takes all the action to reduce the spread which can be expected since this rise in infection is exceptionally high.

4.2.b

Comparing π_{Factor} and π_{Toggle}, we observe that the former performs better in terms of cumulative rewards. This can be observed from the rewards histogram, as the π_{Factor} one is skewed significantly more to the right. On the other hand, π_{Factor} performs substantially worse in the total number of deaths.

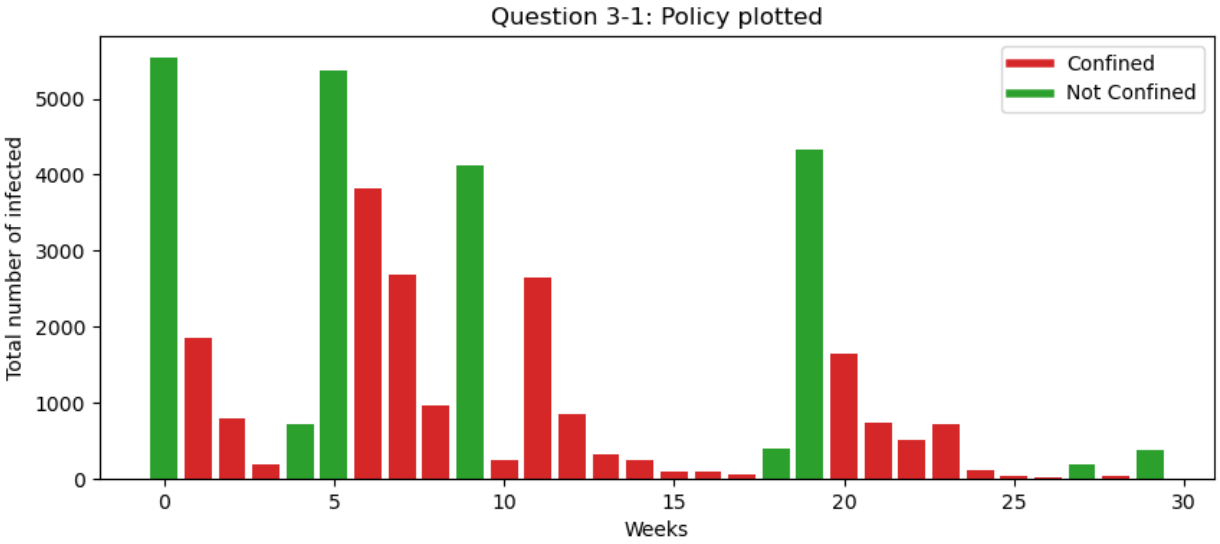

Figure 11: Plot of the action the π_{DQN} policy chose over an example episode

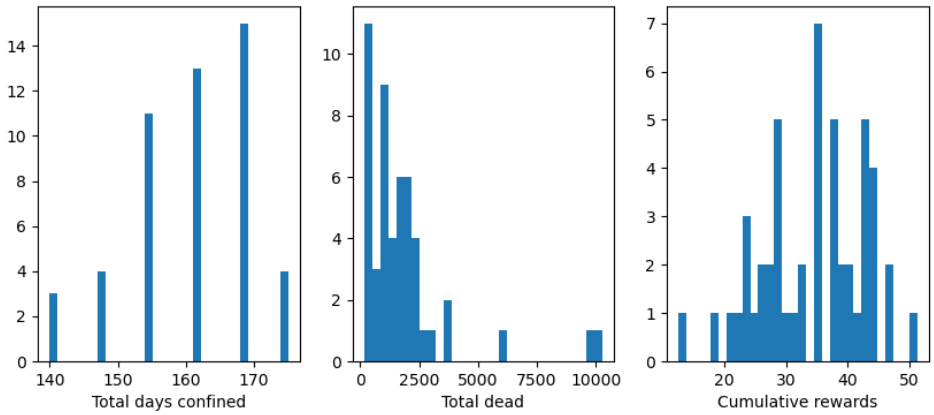

Figure 12: Plot of Total days confined, Total dead and Cumulative rewards histograms of the π_{DQN} without decreasing exploration policy simulation

4.2.c

The main assumption that is made by factorised $Q-values$ is the linearity assumption. Meaning:

$$Q(a_{conf}^{[w]} \cup a_{\text{isol}}^{[w]} \cup a_{\text{hosp}}^{[w]} \cup a_{\text{vacc}}^{[w]}, s) = \sum_{\mathfrak{d} \in \text{decisions}} Q(a_{\mathfrak{d}}, s). \tag{1}$$

This implies that the $Q-values$ of each action are independent of each other. Environments, where factorised $Q-values$, would not be a suitable approach are ones where actions are interconnected or would require coordination to achieve good results. For a more concrete example, we can imagine a robotic arm with multiple joints and continuous action spaces. Each joint can move independently. This setting would require cooperation between each joint to achieve success which yields factorisation of $Q-values$ difficult.

Question 5

5.a

We will refer to the table in Figure 9. For the Russo policy, the training and evaluation curves are constant (there is no training to be done). Moreover, the mean reward is quite low (-69.96). For the Single action DQN (see Figure 10, the training reward trace is quite noisy when using a fixed ϵ and stabilizes later when using a decreased ϵ in the greedy algorithm (see Figure 13). Moreover, both of these algorithms attain positive mean rewards (in the case of decreasing ϵ, it is 36.27. Toggle Policy also used a decreasing epsilon, and we see that in the training trace plot in Figure 15. As the policy had access to

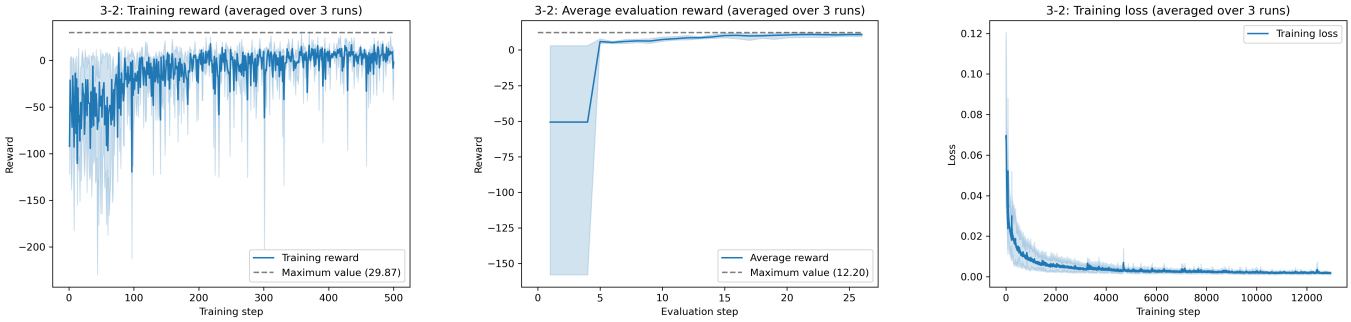

Figure 13: Plot of Training rewards, evaluation rewards and training loss of the π_{DQN} policy with decreasing exploration

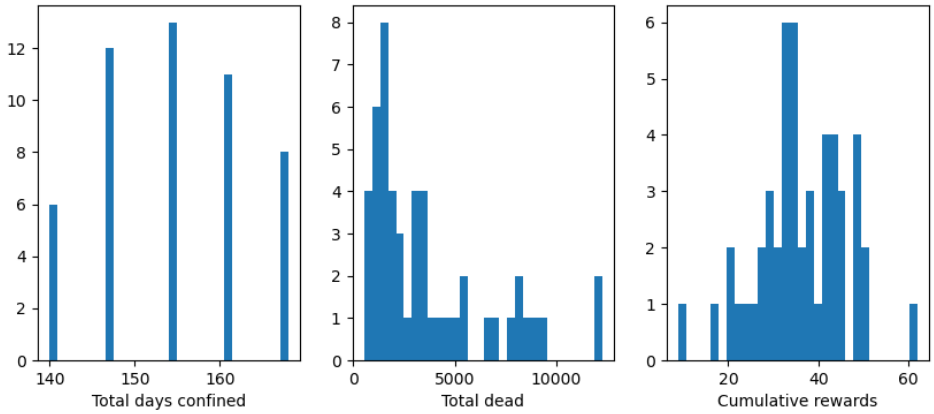

Figure 14: Plot of Total days confined, Total dead and Cumulative rewards histograms of the π_{DQN} with decreasing exploration policy simulation

more information about the environment (previous state) and more action outputs, it outperformed both the Russo and the DQN policies. For the Factorized Policy, the policy attained a positive mean reward but was worse than Toggle and DQN policies. This could be due to the fact that the Factor policy 1) did not converge as it had many more actions to take and 2) did not know the previous state of the system (as opposed to the Toggle policy).

5.b

We will rank each policy from left (worst) to right (best) based on their mean values for each observed metric.

1. $\text{avg}[N_{conf}] : \pi_{Toggle} < \pi_{DQN} < \pi_{Factor} < \pi_{Russo}$
2. $\text{avg}[N_{isol}] : \pi_{Factor} < \pi_{Toggle} < \pi_{Russo} = \pi_{DQN}$ (Russo and DQN Policy's cannot isolate)
3. $\text{avg}[N_{vacc}] : \pi_{Factor} < \pi_{Toggle} < \pi_{Russo} = \pi_{DQN}$ (Russo and DQN Policy's cannot vaccinate)
4. $\text{avg}[N_{hosp}] : \pi_{Toggle} < \pi_{Factor} < \pi_{Russo} = \pi_{DQN}$ (Russo and DQN Policy's cannot hospitalize)
5. $\text{avg}[N_{conf}] : \pi_{Toggle} < \pi_{DQN} < \pi_{Factor} < \pi_{Russo}$
6. $\text{avg}[N_{dead}] : \pi_{Russo} < \pi_{Factor} < \pi_{DQN} < \pi_{Toggle}$
7. $\text{avg}[R_{cumul}] : \pi_{Russo} < \pi_{Factor} < \pi_{DQN} < \pi_{Toggle}$

So we see that if we wish to optimize for decreasing the number of confined days (and hence optimizing population happiness :)), π_{Russo} is best. However, if we wish to save the most amount of lives, we should use π_{Toggle}, which confines the most. Only π_{Factor} increases isolation and vaccinates, and Toggle hospitalizes more than the Factor policy. If we wish to directly maximize the reward (which could also reflect financial rewards, not only moral rewards such as saving people), it seems that π_{Toggle} is the best.

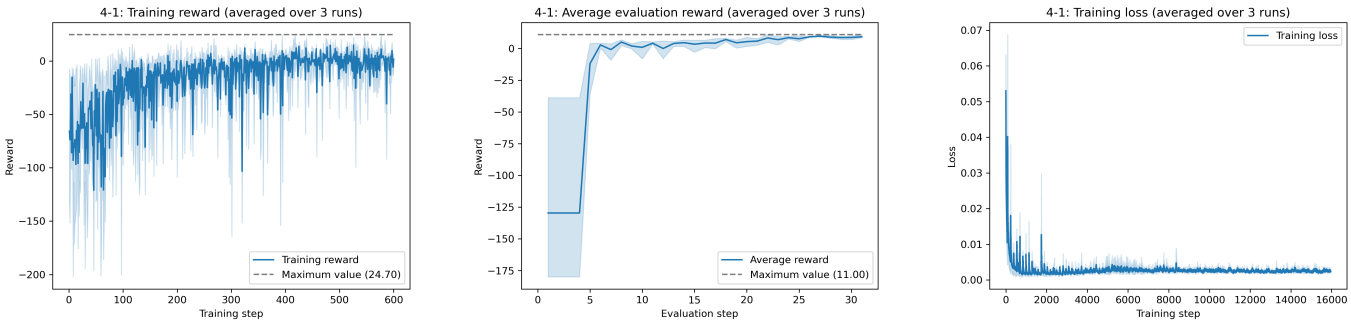

Figure 15: Plot of Training rewards, evaluation rewards and training loss of the π_{Toggle} policy

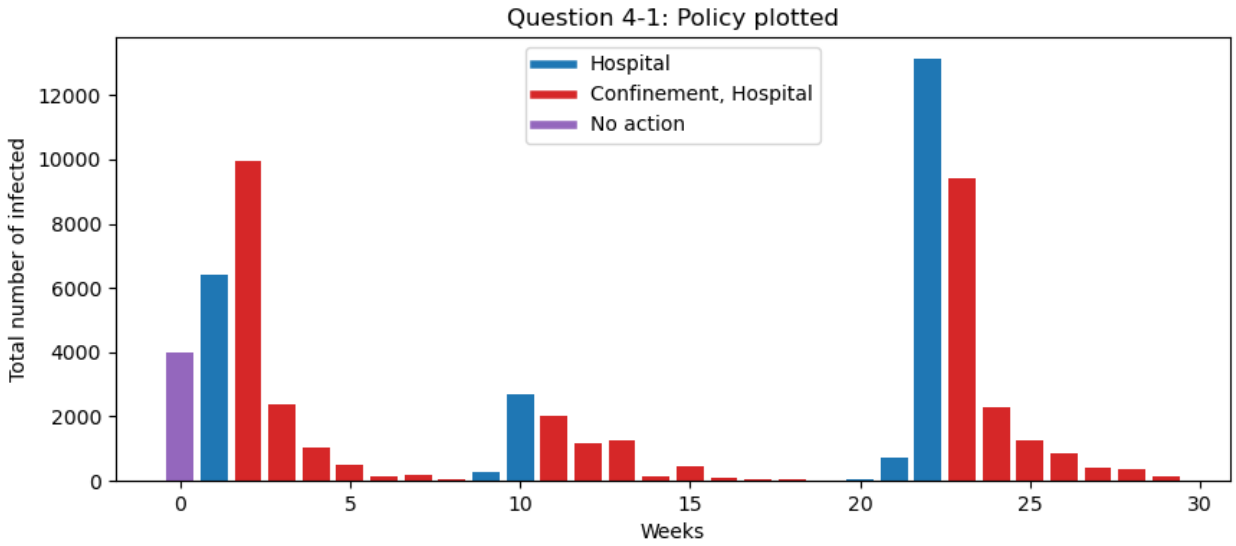

Figure 16: Plot of the action the π_{Toggle} policy chose over an example episode

5.c

We will discuss Figure 21. Both policies were run on the same seed. For the first week, they both reacted in the same way (no action). For the second week, the Factorized policy prefers to take no action, but the DQN policy begins confining and confines for the large majority of the time. The Factorized policy on the other hand, around week 20 when there is a sudden spike in the number of infected people, estimates the Q-value of no actions quite well (see purple sells in No-action rows), and reacts appropriately by confining, isolating, hospitalizing and vaccinating (all actions at the same time). However, Factorized Factorized only isolates and vaccinates for that bad week, and quickly goes back to no action. It keeps on hospitalizing for the majoriy of the weeks between week 21 until the end.

5.d

Theoritically, given the same input, and increasing the number of possible actions to take and assuming that the policy is optimized sufficiently, the policy that can take more actions should do better. However, we do not observe that that (the Factorized policy performs worse than DQN) and we suppose that it is due to optimization problems (see Question 4.2a). Now, if we talk about the cumulative reward strictly as the number of actions given a fixed state, it is not necessarily an increasing function. Consider this case: all the cities in the country have no infected people, no dead (essentially, there is no pandemic). This certainly could be one of the cases of our simulation. There, any taken action is unnecessary (there is no reason to confine, isolate, hospitalize or vaccinate) and taking any action will result in decreasing the reward. The same argument applies to why the reward is not a decreasing number of actions (sometimes, when the pandemic is widely spread, you want to take as many actions as possible as the factorized policy did in week 20). The cumulative reward seems to be a more complex function than that.

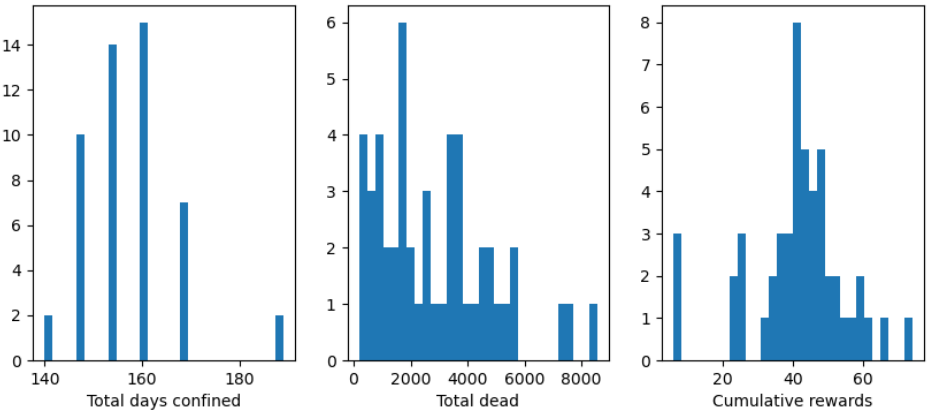

Figure 17: Plot of Total days confined, Total dead and Cumulative rewards histograms of the π_{toggle} policy simulation

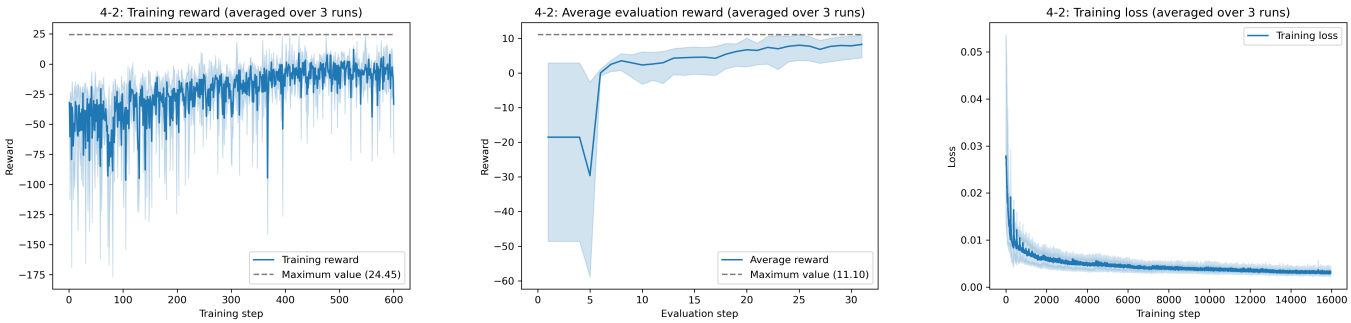

Figure 18: Plot of Training rewards, evaluation rewards and training loss of the π_{factor} policy

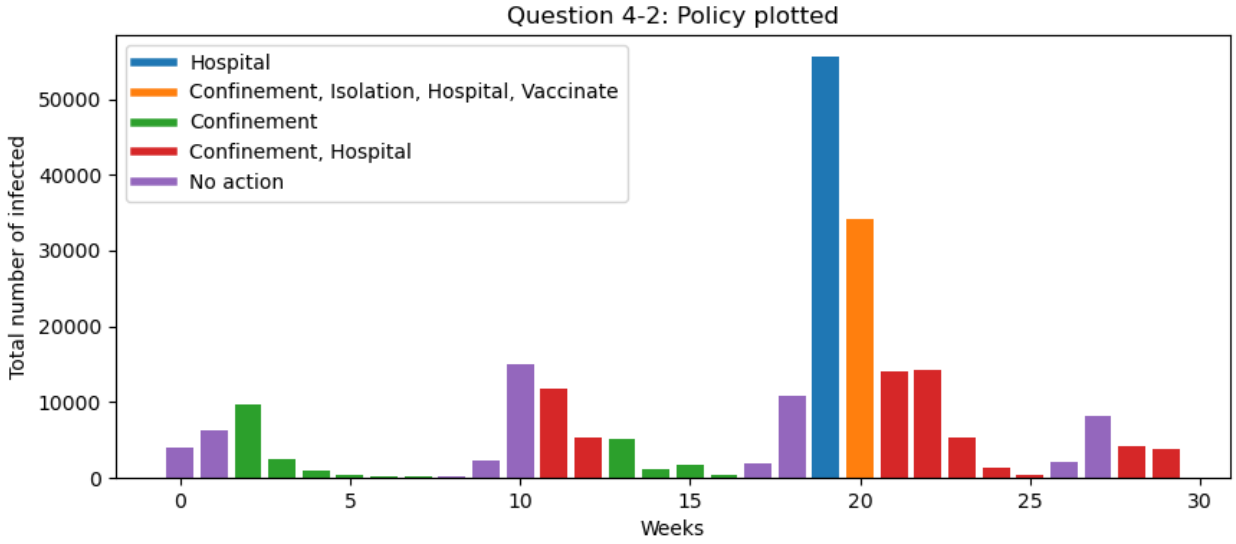

Figure 19: Plot of the action the π_{Factor} policy chose over an example episode

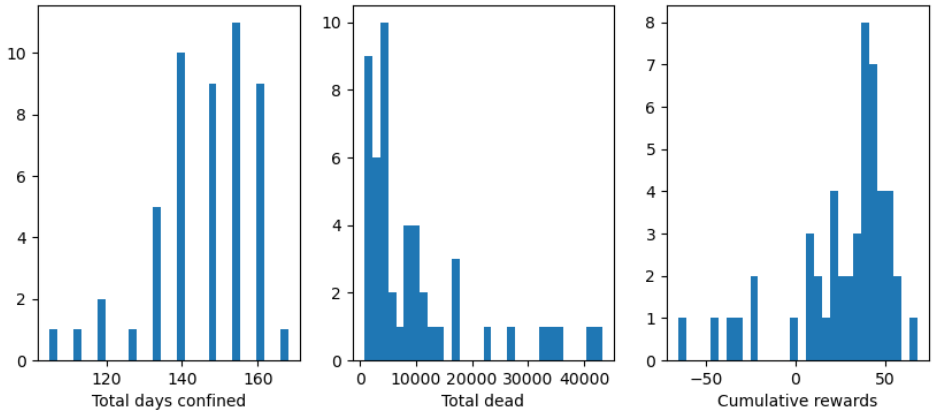

Figure 20: Plot of Total days confined, Total dead and Cumulative rewards histograms of the π_{factor} policy simulation

Figure 21: Plot of π_{Factor} and π_{DQN} heatmaps