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

Amin Asadi Sarijalou
IC, EPFL
amin.asadisarijalou@epfl.ch

Ilker Gül
IC, EPFL
ilker.gul@epfl.ch

Question 1: No Epidemic Mitigation

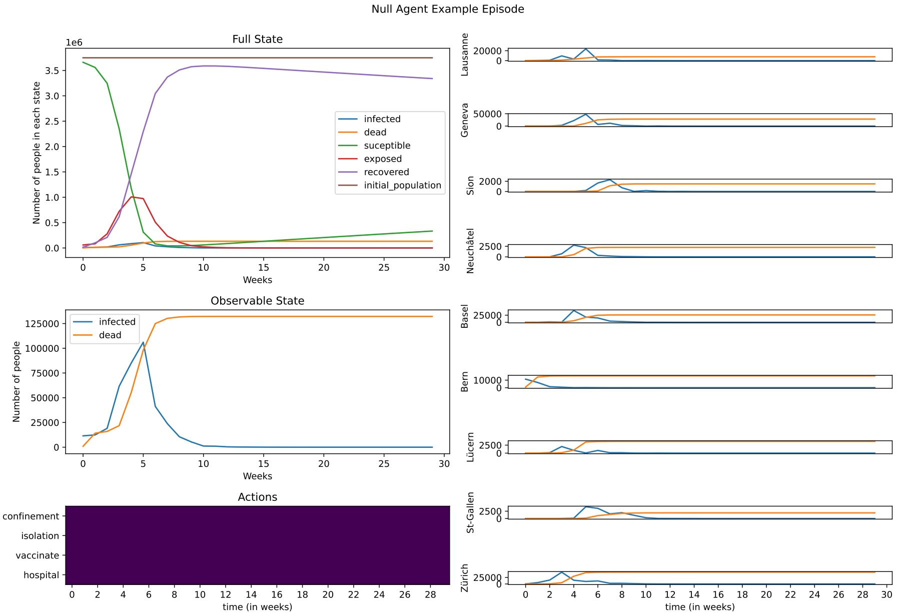

Figure 1: As we can see in this plot, most of the change in the variables occur in the first 10 weeks: The number of susceptible individuals drop significantly because they get exposed and possibly infected and afterwards they get recovered or they die. In this regard, we can observe a notable increase in the number of recovered individuals. This metric peaks at week 10 where it gets very close to the initial population size. This means, a significant portion of the population gets infected in the first 10 weeks. The recovered people gain immunity against the virus; thus, for a long time afterwards, no significant change occurs in the variables. We also observe that the number of susceptible and recovered people start to increase and decrease, respectively, after around week 10. This is because some recovered people lose their immunity and become susceptible again. The number of exposed and infected individuals both peak at week 5; nevertheless, the number of exposed is much higher (≈ 1 M) than the number of infected (≈ 0.1 M), which means only a small portion of the exposed individuals become infected. Then, they both start to decrease (both eventually become recovered or dead). Finally, the number of dead people grow sharply in the first 7 weeks, but following that, as people get recovered and gain immunity, it almost levels off at around $130,000$, which is relatively high for a population of size $\approx 3.7M$.

Question 2: Prof. Russo's Policy

2.a: Implement Pr. Russo's Policy

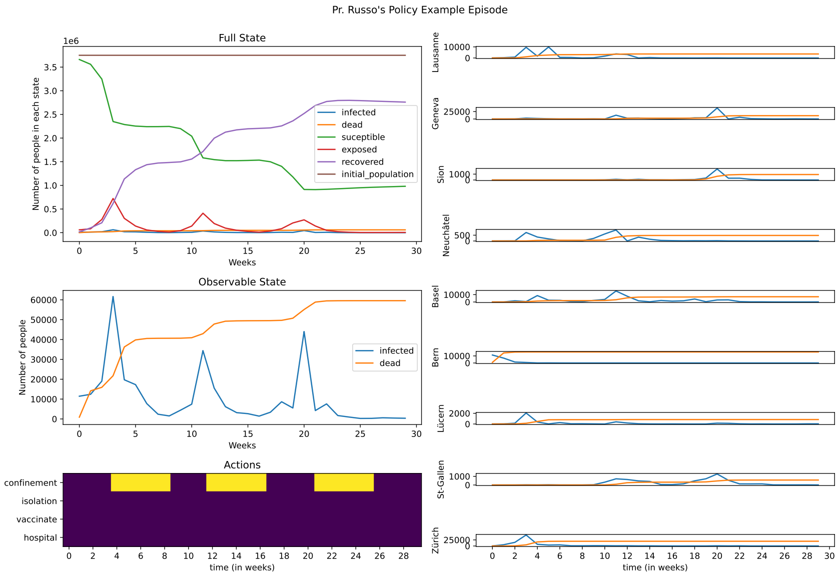

Figure 2: We observe that as expected, each time the number of infected reaches above $20,000$ individuals, the confinement action is taken for the next 4 weeks, which results in a sharp decrease in the number of infected and exposed variables. The death toll is almost half of the death toll compared to the unmitigated scenario. Also, the increase in the number of deaths happens slower and more gradually with the new policy. Another important observation is that the number of susceptible people levels off at $\approx 1M$ people (after week 20) compared to the no mitigation policy where this metric reaches zero. Please note that the action space here is only 'confinement' and 'no confinement'.

2.b: Evaluate Pr. Russo's Policy

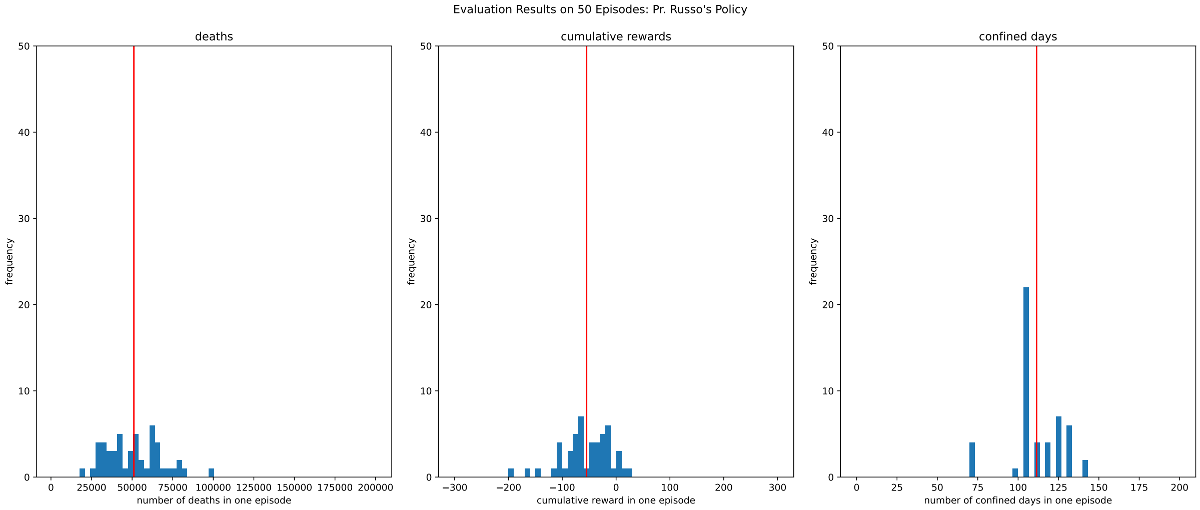

Figure 3: The histograms resulting from the evaluation of the Pr. Russo's Policy for 50 episodes. Average death number: 51079.26, Average cumulative reward: -54.80, Average number of confined days: 111.44. We can see that, although the agent takes a considerable amount of confinement days, the death toll is very high and the cumulative reward is negative, thus the policy is not effective enough.

Question 3: A Deep Q-learning approach

3.a: Implementing Deep Q-Learning

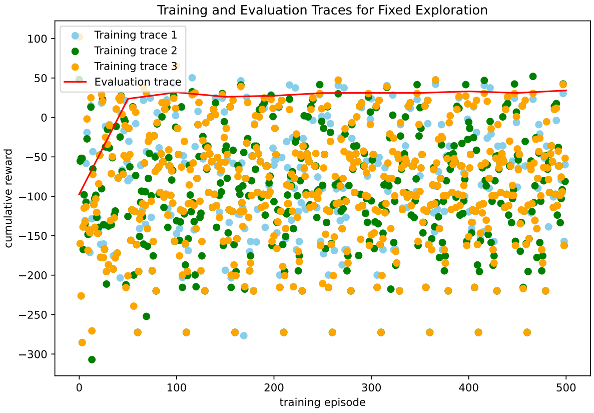

Figure 4: 3.a: As we can see here, the training successfully converges to a good cumulative reward.

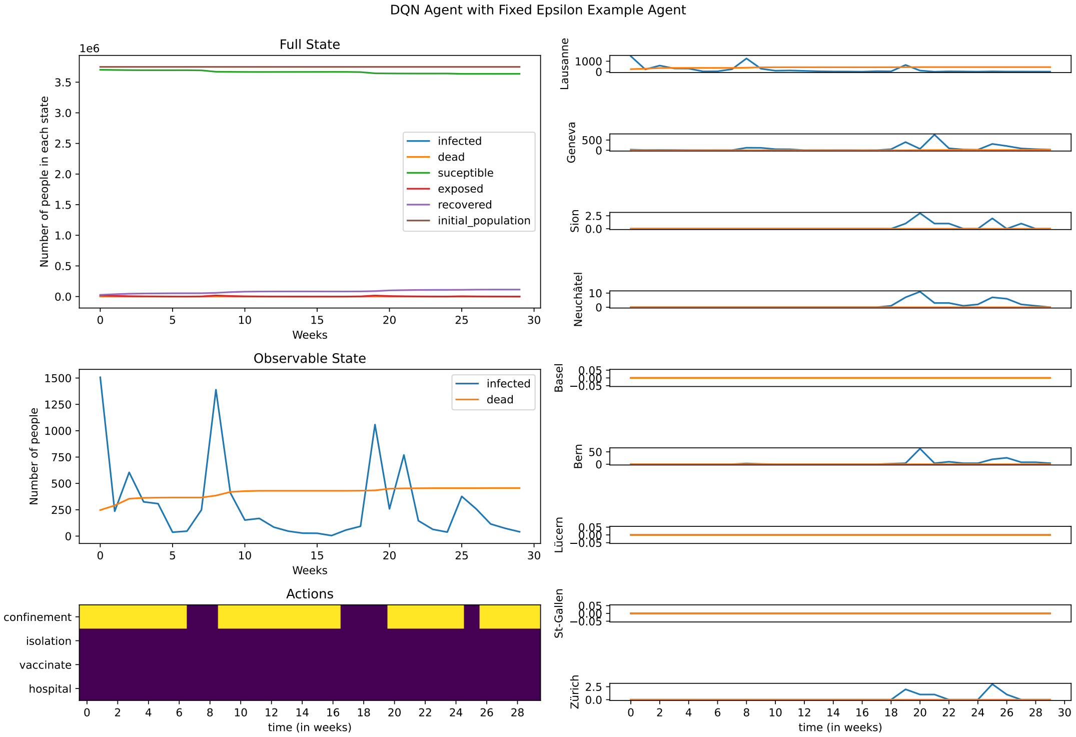

Figure 5: 3.a: An example episode of DQN policy. (Note that the action space is only *confinement* or *no-confinement*). As we can observe, the policy is semantically similar to Pr. Russo's policy, however, instead of waiting for the number of infected individuals to reach 20, 000, as soon as this number starts growing with the sharp rate, it takes the confinement action. It seems that, every time, it keep confining for at least 4 weeks depending on the observed number of infected people. As a result, most of the people never get infected by the virus and remain susceptible. Moreover, total number of deaths is significantly small and is ≈ 500.

3.b: Decreasing Exploration

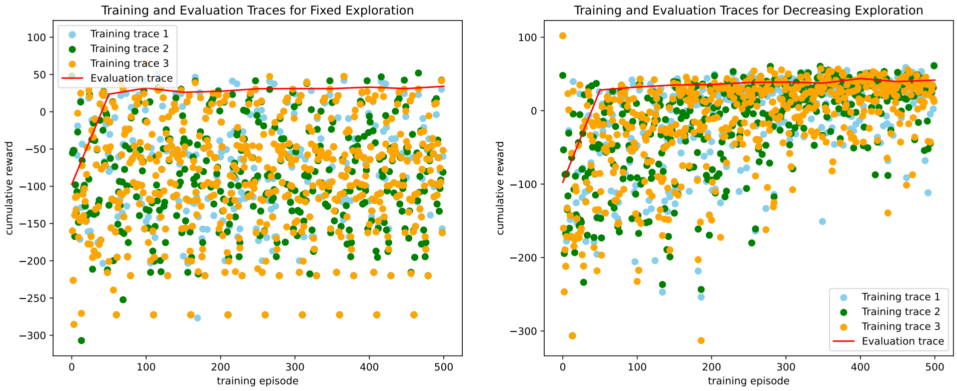

Figure 6: 3.b: Training and Evaluation Traces for fixed exploration (left) and decreasing exploration (right). The training trace of decreasing exploration policy is significantly more stable than that of fixed exploration policy. The reason is that with decreasing exploration, in early stages, the agent explores random actions in order not to get stuck with non-optimal policies. As it learns from the environment and its actions, it gradually decreases exploration, and instead, increases exploitation of its learned policy, which helps stabilize the training process. In contrast, fixed exploration until the end will prevent the agent from adequately exploiting its learned policy and the agent keeps exploring random strategies with fixed rate, hence the instability.

3.c: Evaluation

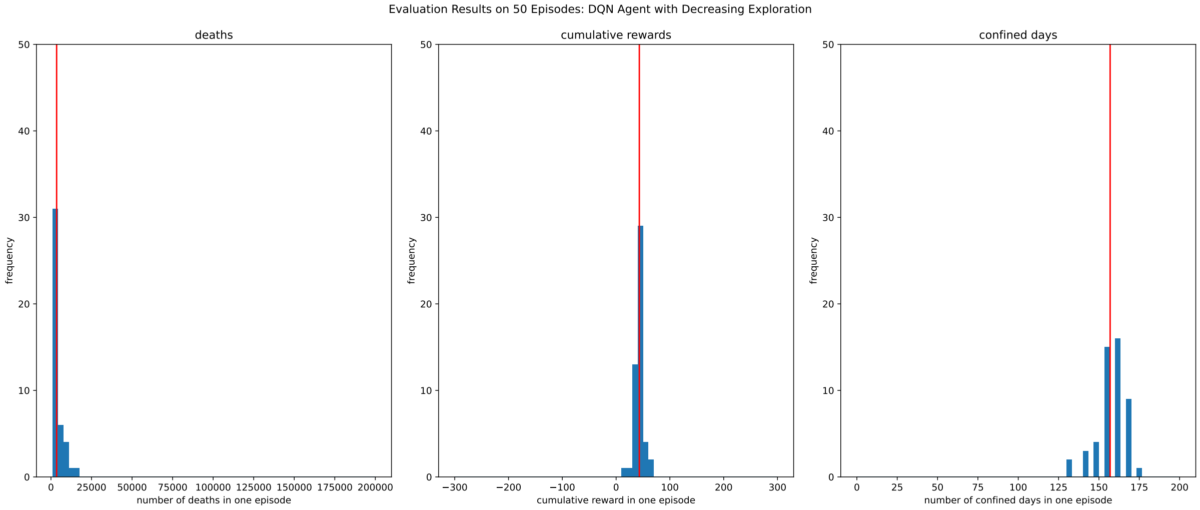

Figure 7: 3.c: The histograms resulting from the evaluation of the π^*_{DQN} for 50 episodes. Average death number: 3496.52, Average cumulative reward: 43.16, Average number of confined days: 156.94. By comparing these plots to Figure 3, we notice that the DQN agent significantly outperforms the Pr. Russo's policy: 1-The number of deaths is mach smaller (≈ 3496 in DQN vs. ≈ 51000 in Pr. Russo's policy) and 2-The mean reward is significantly higher (≈ 44 in DQN vs. ≈ -54 in Pr. Russo's policy). These improvement are mainly because the DQN policy takes confinement actions with smaller threshold of number of infected and for a longer period (as explained in Figure 5). The total confined days is 156 compared to 111 with Pr. Russo's policy, which is 45 days more (rather notable).

Question 4

4.1.a: Action space design

We can definitely use a model with 16 outputs to estimate a Q-value for each possible action, however, such a model with large action-space cardinality might be harder to train and the training might be less stable as switching between one action to the other could change the environment drastically. The toggled action-space technique, although has less power, can be easier to train, since at each time step, at most one possible decision can change its state. Moreover, by designing the action-space in this way, the model can more easily learn the costs associated with switching the state of the decisions from False to True. In this technique, for training, the observation space includes the state of the actions as well, but, the network architecture does not change compared to single-action (confinement or no-confinement) DQN model.

4.1.b: Toggle-action-space multi-action policy training

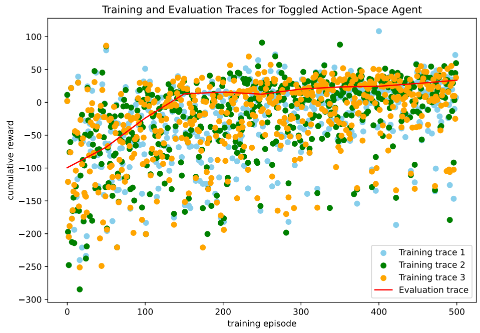

Figure 8: 4.1.b: Training and evaluation traces for training the toggle DQN policy. As we can see, the agent is successfully learning as the training and the evaluation curve both converge, but not to a cumulative reward as high as the single-action DQN policy. Also, not that it seems that it is possible to achieve higher cumulative reward, if trained for more episodes.)

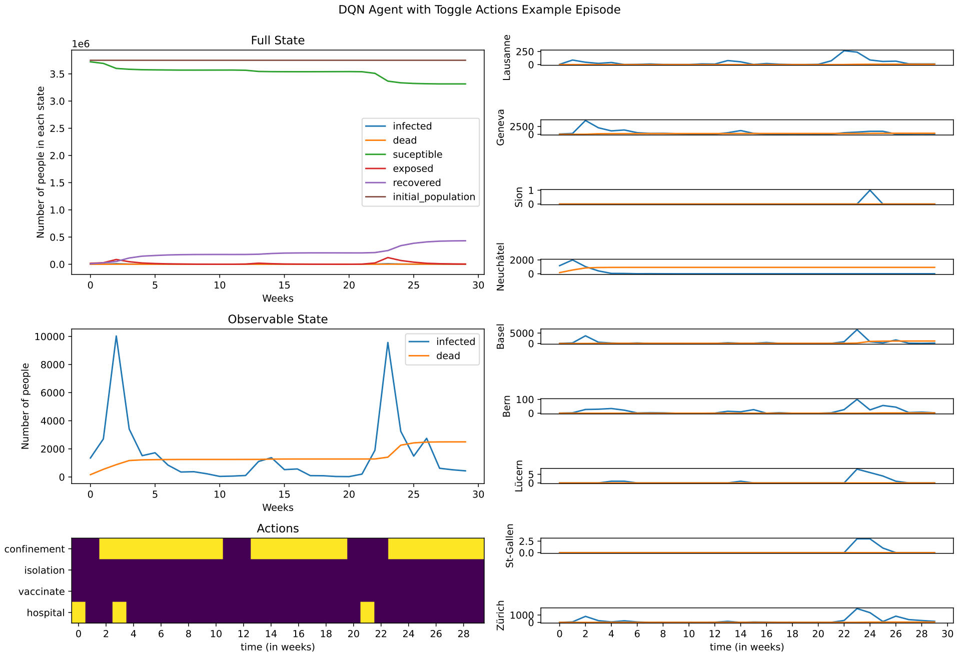

Figure 9: 4.1.b: An example episode of Toggle DQN policy. The policy takes the *confinement* action every time a surge of number of infected people occurs to prevent the virus from further spreading. The duration of the confinement seems to be proportional to the number of infected people. The agent also decides to add hospital beds three times, two of which are when the number of infected people are very high, thus we hypothesise that the agent takes this action so that there are enough hospital beds for infected people, which can decrease the number of deaths.

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

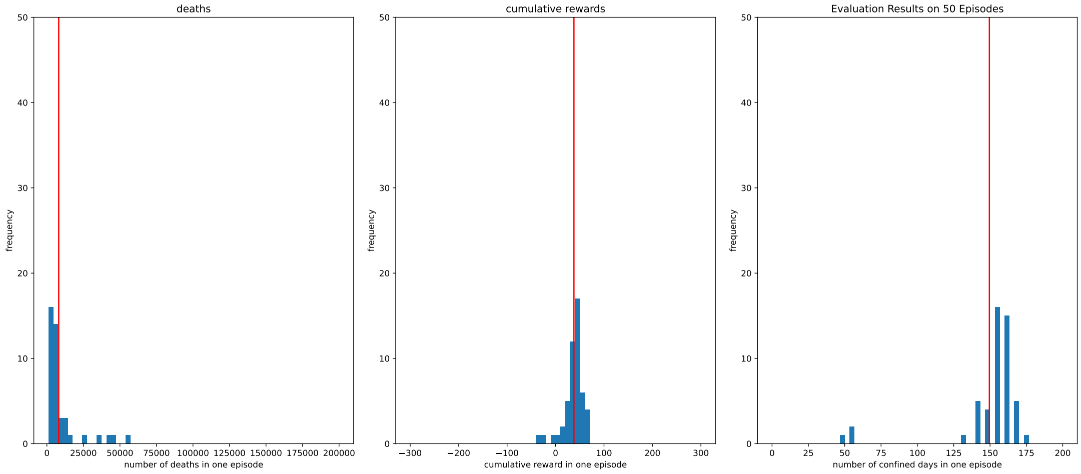

Figure 10: 4.1.c: The histograms resulting from the evaluation of the π^*_{toggle} Policy for 50 episodes. Average death number: 7110.6, Average cumulative reward: 33.9, Average number of confined days: 153.16. We observe that the compared to the single-action policy, the reward is slightly less (33.9 vs 43.16), total deaths are approximately doubled (3496 to 7110) and the number of confinements is roughly similar (153 vs 156). Thus, the toggle action space agent performs worse than binary-action space (=single-action) agent. Our hypothesis is that this is because the toggle policy does not have direct control over deciding the actions and can only do so indirectly by choosing to toggle.

4.1.d: question about toggled-action-space policy, what assumption does it make?

Such an action-space assumes that at each time step, at most one action can change its status compared to its current status, and all other actions keep their current status. This technique would not be suitable for an action-spaces where the agent must change two actions at the same time after making a particular observation. For example, assume that the agent is a self-driving car. Assume that two of the actions of this agent are: *speed* which can take the values in the set: IDLE, INCREASE, DECREASE and *direction* which can take the values: IDLE, TURN RIGHT, and TURN LEFT. Assume that both actions are in IDLE state. If the agent observes that a human is on the way and an accident may happen, it must immediately decide to slow down and change direction of the car at the same time(if possible). In such a scenario, the toggle behavior would not be suitable as it changes the states of the actions one at a time.

4.2.a: Multi-action factorized Q-values policy training

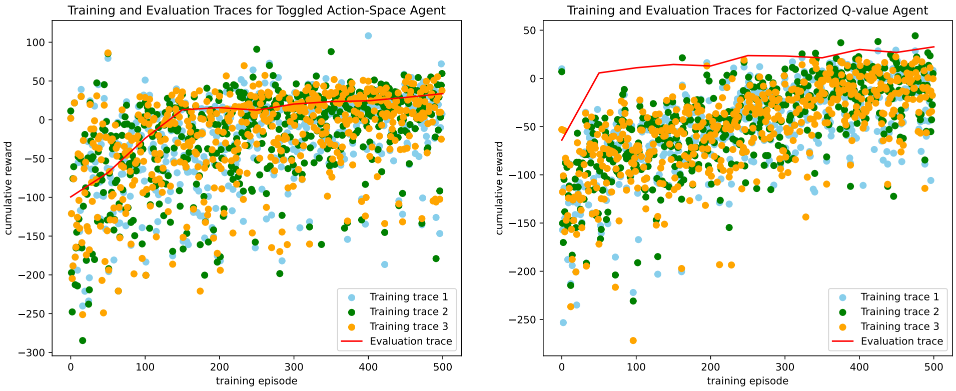

Figure 11: 4.2.a: Yes, the policy successfully learns and converges. We observed that training this policy was more difficult than other policies as the action-space is larger. Moreover, the training and evaluation traces for this policy had several ups and downs. We hypothesise that this behaviour is do to the fact that in factorized policy the chosen actions from episode to another can be very different and can result in very different rewards, as the actions are more fine-grained in this policy.

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

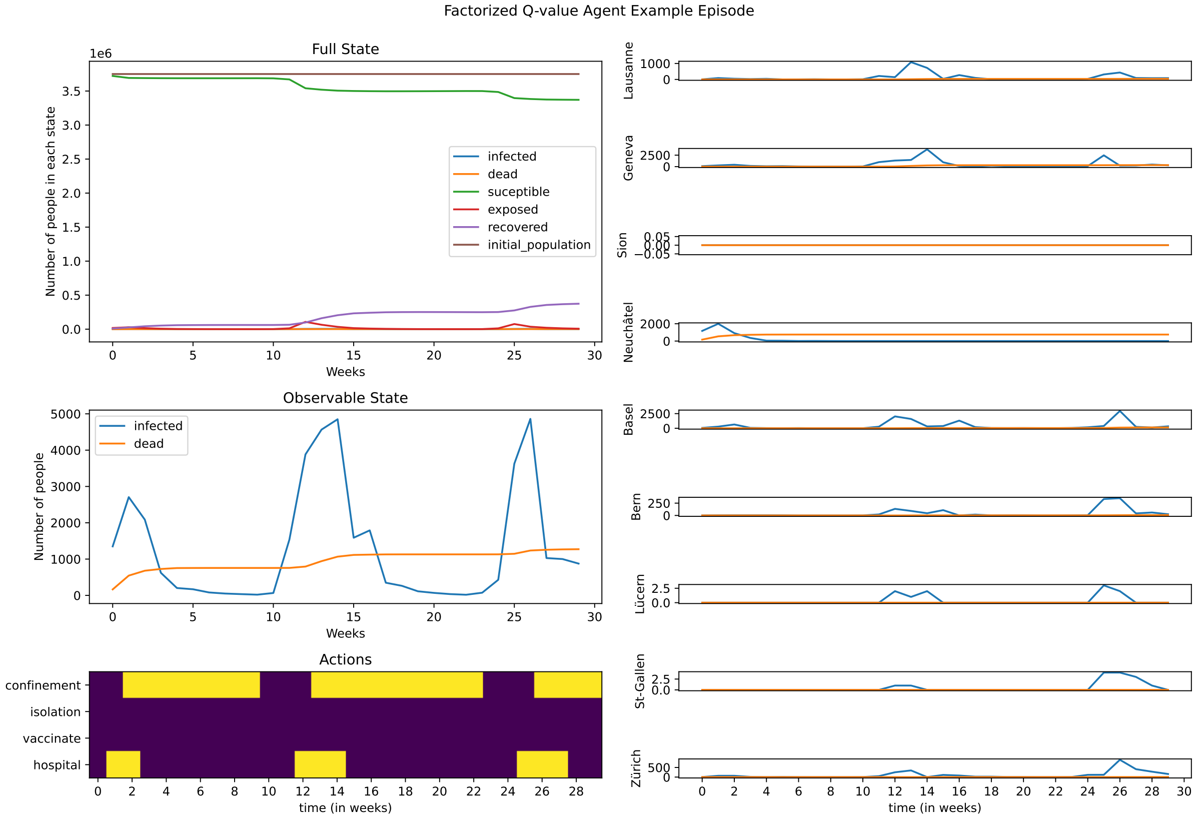

Figure 12: 4.2.a: An example episode of the factorized Q-values policy. Similar to the toggle policy, this policy also uses only the confinement and the hospital actions. It decides to add hospital beds when the number of infected is high, which slows down the growth of this number, and then decides confinement, which sharply decreases the number of infected; shortly after, it stops adding hospital beds. As one of the confinement periods is very long (10 weeks), this policy might not be realistic.

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

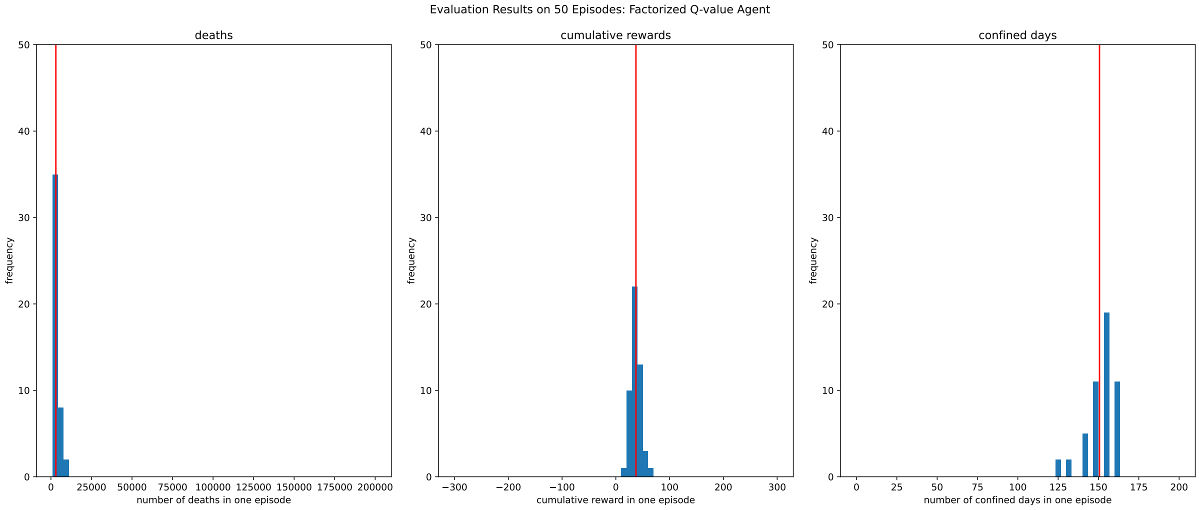

Figure 13: 4.2.b The histograms resulting from the evaluation of the π^*_{factor} Policy for 50 episodes. Average death number: 3033.7, Average cumulative reward: 37.8, Average number of confined days: 150.64. Average number of death is less than half of the average number of death in toggle policy. The average cumulative award is slightly better (37.8 vs 33.9) and the total confinement days is almost the same (150.64 vs 153). Therefore, overall, this policy performs better than the toggle policy of part 4.1.b.

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

First of all, we should note that the cardinality of the original action space is 16 (2 possibility for each of the four decisions). However, in this technique we have reduced the cardinality to 8 by designing the model in such a way that each decision contributes to the actual Q-value independently. As a result, factorizing the Q-values is not suitable in action-spaces where decisions cannot be assumed to be independent. For example, in a human-robot that can move forward, the possible decision for each of the legs is MOVE or NOT-MOVE. Thus, the original action space is: LEFT-MOVE, LEFT-NOT-MOVE, RIGHT-MOVE, RIGHT-NOT-MOVE. However, for a successful walking, one of the feet must remain steady, while the other must move, thus the decisions for each of the feet cannot be made independent of the other. As a result, factorized Q-values is not suitable for this action space. Such a technique may decide NOT-MOVE or MOVE for both legs, and therefore, instead of walking, the robot will remain still or will fall down, respectively.

Question 5

5.a

Training Curves: In terms of training curves, as also explained in part 3.b, the DQN policy with decreasing exploration is more stable. The training of single-action DQN policy and factorized policies achieved positive rewards faster (in evaluation) than the toggled policy (in 50 episodes compared to 100 episodes, respectively). Also, the single-action and toggled policies had a smoother growth in cumulative reward, compared to the factorized policy, which faced several ups and downs along the training.

Evaluation: The Pr. Russo's policy performs the worst among the policies. This is because this policy is a predefined policy and is not trained on environment. The *single-action DQN* policy performs much better than the Pr. Russo's policy and achieves much smaller number of deaths and much higher cumulative reward. In fact, this policy outperforms all models in terms of cumulative reward. The *factorized Q-values* performs slightly worse than the single-action DQN in terms of cumulative reward, but better in terms of total deaths. However, it should be noted that it is rather difficult to train this policy and that we trained it with a smaller lr than the single-action DQN but with the same number of episodes. Different learning rates would result in different performances. We chose $lr = 10^{-3}$ for training this policy. In fact, it might be possible to achieve better results for this policy using different hyper-parameters. Our expectation was that this policy should achieve the best results as it has the largest action-space and more fine-grained control over the actions. Nevertheless, one should note that this policy does not have access to the full cardinality of possible actions (16), but only to 8 of them. It assumes that the Q-value of a full-action is the sum of the score of the corresponding decisions, which is just an assumption and is not guaranteed to achieve best results. Finally, the *toggled-action-space* policy performs worse than the other DQN policies in terms of both cumulative reward and total deaths. We suggest two possible reasons for this. First, in order to converge, we trained the *toggle-action-space* policy with smaller learning rate(10^{-3}) compared to the *single-action DQN* policy but with the same number of episodes. The *toggled-action-space* policy might have needed more training episodes to converge to a better policy. Second, the *toggled-action-space* policy does not have direct control over picking or not picking each of the four possible decisions (as it can only choose to toggle), which can result in less power compared to the *single-action DQN* policy that could directly decide to do or not do the confinement action.

5.b

Policy	total confined days	total isolation days	total additional hospital bed days	total vaccination days	number of total death	cumulative reward
π_{russo}	**111.44**	–	–	–	51079.26	−54.80
π_{DQN}	156.94	–	–	–	3496.52	**43.16**
π_{toggled}	153.16	**0**	**25.2**	**0**	7110.6	33.93
π_{factor}	150.64	**0**	62.16	**0**	**3033.7**	37.38

Table 1: Comparison of policies with respect to several metrics averaged across 50 sample episodes. We observe that the Russo agent has the least confinement days, however, this low number has resulted in a very large number of deaths and negative cumulative reward. Other agents have approximately same number for this metric (about 150). The toggle agent has decided to add hospital beds less than half of that of the factorized policy. None of the toggle and factorized policies ever decide to isolate or vaccinate. Both toggle and factorized agents have learned that confinement is the cheapest and yet most effective action to do. This is not very realistic. Perhaps, the cost for confinement action in the network dynamics is cheaper than what it should have been, thus, the networks have learned to do only this action almost always. On the other hand, the networks have learned that isolation and vaccination are expensive actions, and thus, they never pick these two actions. The number of total deaths is the least for the factorized Q-value policy. The cumulative award was best for the single-action DQN policy.

5.c

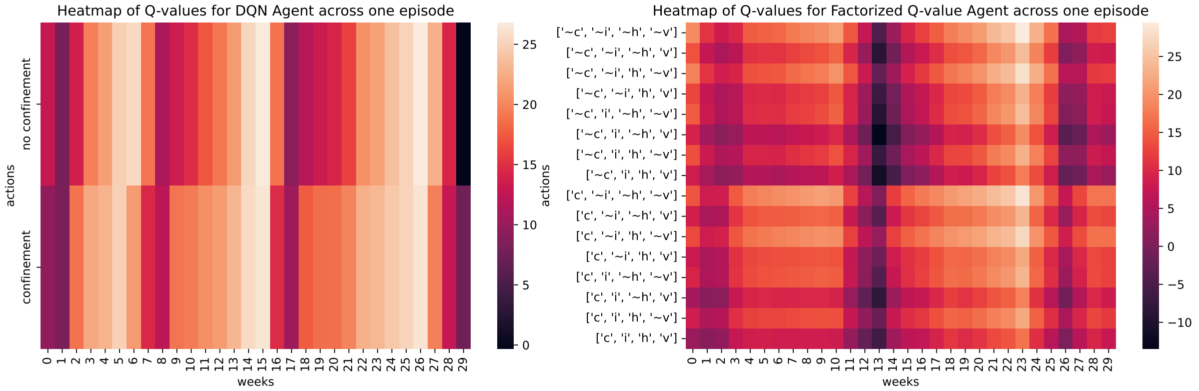

Figure 14: 5c: The heat map for the DQN agent shows that during several intervals, the Q-value assigned to the 'confinement' is larger than that of 'no confinement'; in between these intervals, the Q-value of the 'no confinement' becomes larger than that of 'confinement' so the confinement stops. This is aligned with our conclusions in part 3, that the agent performs 'confinement' in several intervals (each interval for a couple of consecutive weeks). In the heat map for the factorized Q-value agent, the variables i, c, h, v denote 'isolation', 'confinement', 'adding hospital beds', and 'vaccination' respectively, and ∼ behind a decision, means not picking that decision. We can see that the two actions ['c', '∼i', 'h', '∼v'] and ['c', '∼i', '∼h', '∼v'] have the largest Q-values, which means that either the agent chooses confinement, or confinement and adding hospital beds, which is aligned with the observations in part 4.2.

5.d

No, as we saw in the previous parts, the single-action DQN policy achieved a higher cumulative reward compared to the toggled-action-space policy which has more actions. Therefore, the statement is not always true. As mentioned before, one possible explanation for this is that the toggled-action-space policy does not have direct control on activating or deactivating the decisions (confinement, isolation, etc.), and it can only indirectly do so by choosing to toggle them. That being said, it is possible that if we train the toggled-action-space policy more, or with different learning rates, it can achieve higher cumulative reward.