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

Paul Devianne, Paul Boulenger

May 2023

State variable	Quantity represented by the variable
s_{city}	number of *susceptible* individuals in city "*city*".
e_{city}	number of *exposed* (infected but not yet contagious) individuals in city "*city*".
i_{city}	number of *infected* (and contagious) individuals in city "*city*".
r_{city}	number of *recovered* (cannot be infected) individuals in city "*city*".
d_{city}	number of *dead* individuals in city "*city*".

Table 1: State variables of the dynamical model.

1 Introduction

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

We simply create an agent NoAgent which does not act i.e its act methods always returns 0.

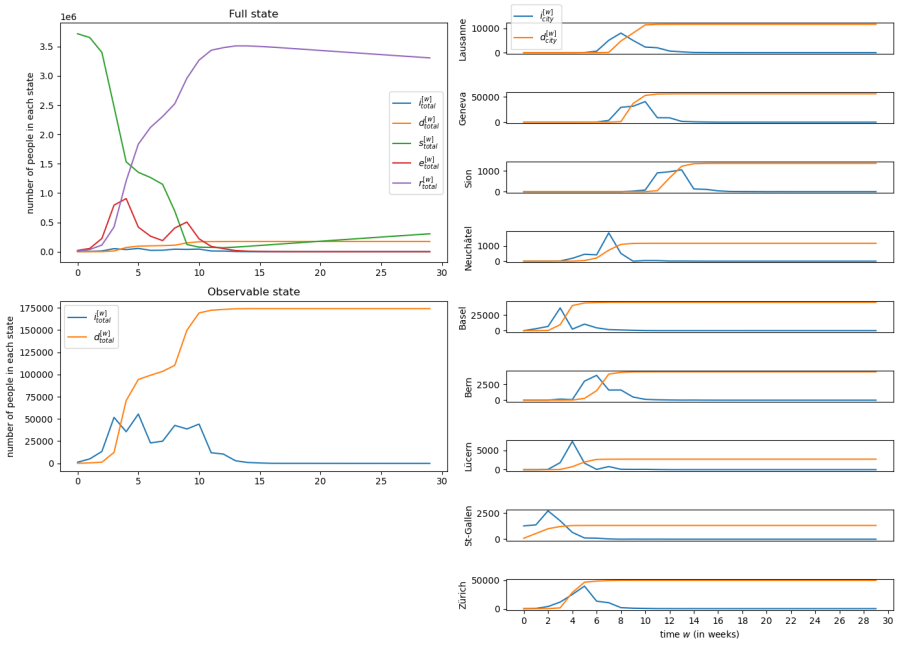


Figure 1: Epidemic simulation without mitigation

Answer The number of death highly increases in the first 10 weeks and then stabilises at 175000. This is related to number of infected which is spiky between 25000 and 75000 during the first 11 weeks. Similar trends are observed in each cities, possibly with a time shift. Also the number of exposed, fits quite well the number of infected, with higher values, a bit like if the number of infected was a proportion of the number of exposed

Logically, the number of susceptible is maximal in the beginning and then drops as people get contaminated ; it re-increases from week 12, as people may loose their immunity. The number of recovered follows the exact opposite trend.

2 Professor Russo's Policy

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

We implement the agent as described and get the following results:

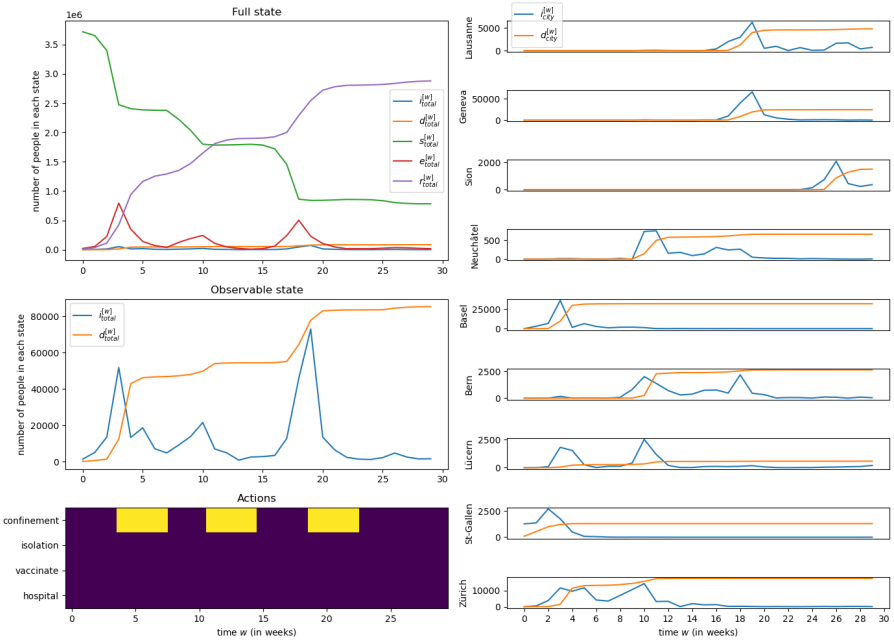


Figure 2: Epidemic simulation with Pr Russo's Policy

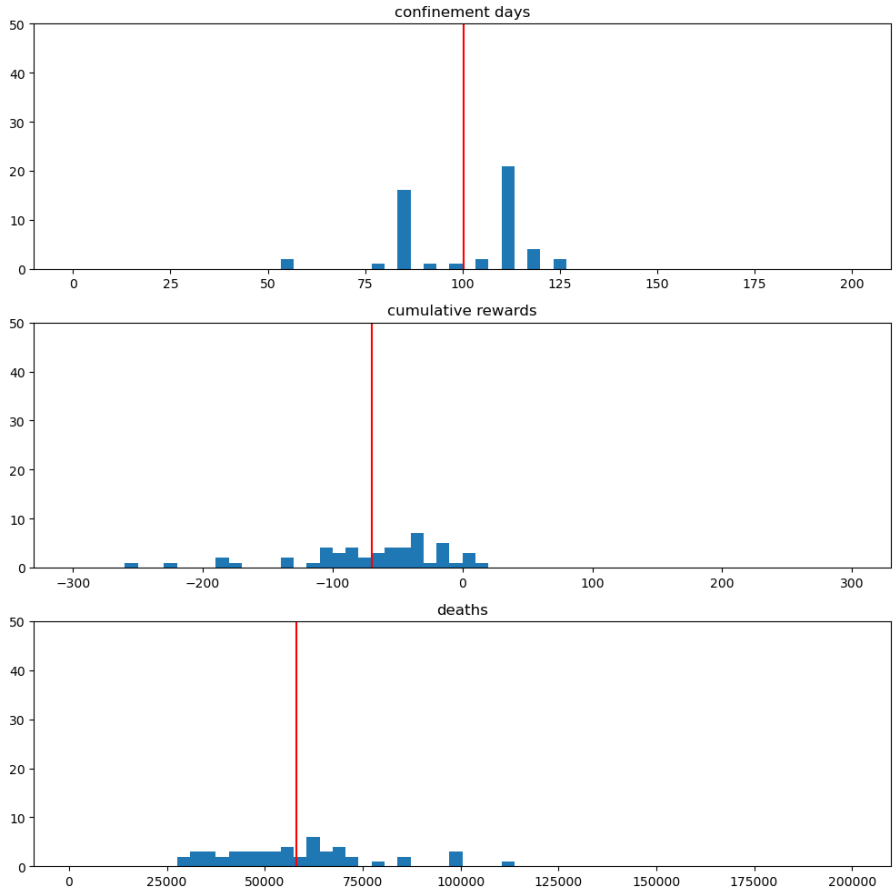


Figure 3: Pr Russo's Policy evaluation

See Figure 2. The professor's policy correctly confines population whenever the total population of infected is above 20'000 at the beginning of a week. The confinement are indeed four weeks long and the population is released when the total population of infected is below 20'000 at the end of the confinement. More specifically, three confinements are done in total. In comparison with the unmitigated case, the professor's policy reduces the total number of deaths by almost 100'000 people or 55%. Then, we clearly see the positive effect of the confinement rule on the epidemic development.

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

See Figure 3.

- Average death number: $\bar{N}_{deaths} = 58071.94$
- Average number of confined days: $\bar{N}_{confinment} = 100.24$
- Average cumulative reward: $\bar{R}_{cumulative} = -69.96$

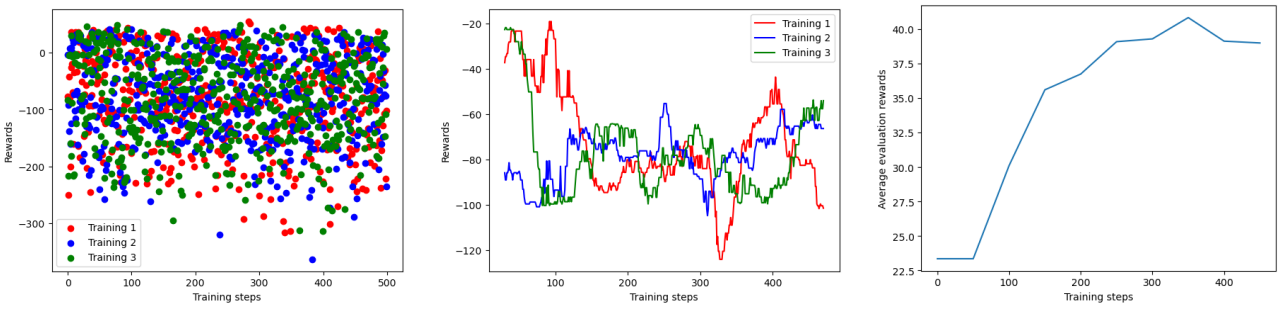
3 A Deep Q-learning approach

Unless stated otherwise all training/evaluation processes and all hyperparameters were chosen accordingly to the recommendations or the pytorch example for DQN. However we decided to remove the clip-grad statements, leading to better training. For the moving medians, the window size is 60

3.1 Deep Q-Learning with a binary action space

Question 3.a) implementing Deep Q-Learning

See Figure 4.



(a) Scatter plot **(b)** Centred moving medians **(c)** Training evaluation

Figure 4: Binary action DQN training

We train three full training processes. The results of the training are presented above. The agent learns a meaningful policy since the average evaluation trace is increasing with the number of training steps. The policy which achieves the highest evaluation rewards π^*_{DQN} is the first one. Then we use, the best policy to simulate episodes. See Figure 5.

One first observes the number of deaths at the end of the episode which is below 1200. Comparing with the professor's policy results we can confirm that this policy performs well for reducing the number of deaths and infected. This obviously has a cost. The confinement weeks represent 77 % of the episode time while it was only 40 % in the professor's episodes. The infected population stays low (below 5000) during the hole episode. Comparing the action table with the observable state graph, the policy has a recurrent behaviour. Whenever the spike in infections is too steep, the agent chooses to confine the population. It will remove the confinement whenever the infected population has recovered a low level. Then, we can confirm that the logic behind the policy seems adequate. Another interesting fact is that four cities were left with no death and no infections.

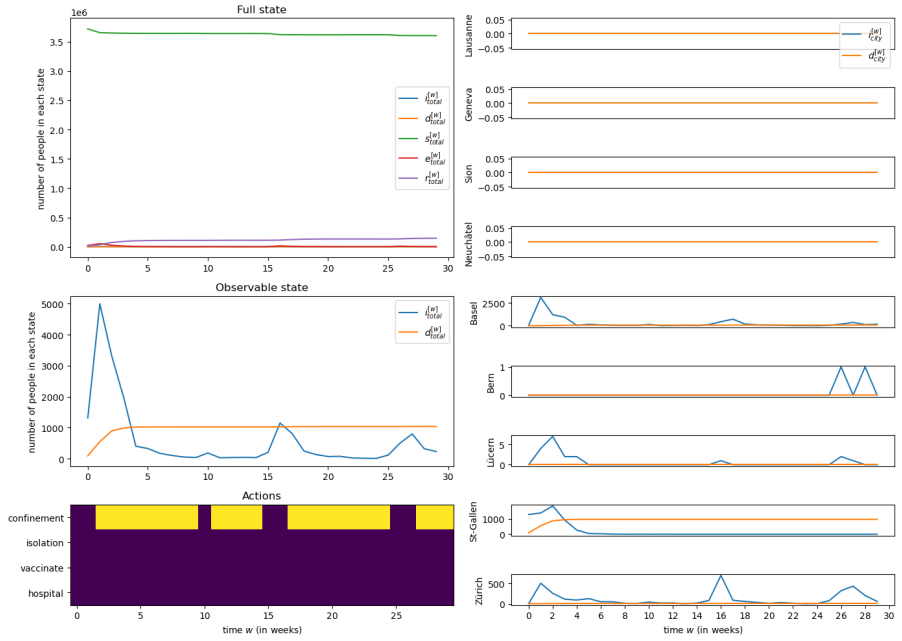
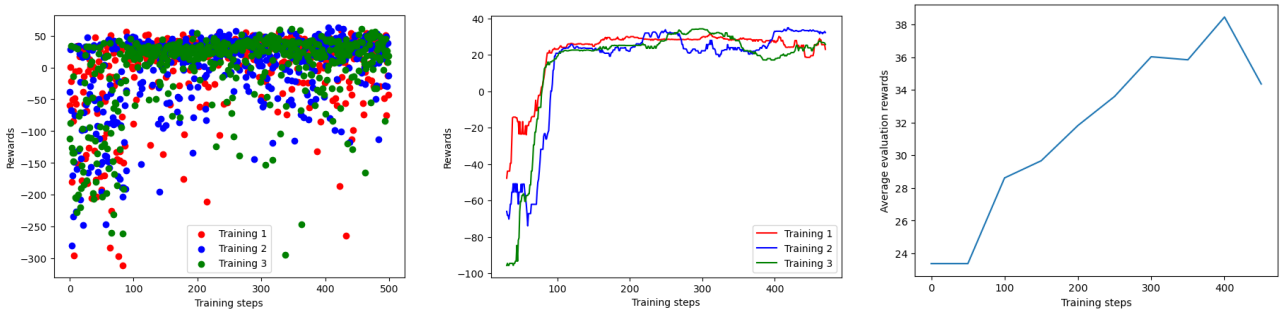


Figure 5: Epidemic simulation with Binary action DQN best policy



(a) Scatter plot **(b)** Centred moving medians **(c)** Training evaluation

Figure 6: Binary action DQN training with epsilon decay

Question 3.b) decreasing exploration

Answer See Figure 6. The policy without epsilon decay gets slightly better results (41 vs 38 of maximal evaluation reward). A possible explanation is that without decay, the agent is always exploring and reach higher rewards. Perhaps the decay we use is too fast and does not allow the agent to fully explore before stabilising.

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

See Figure 7.

- Average death number: $\bar{N}_{deaths} = 3718.26$
- Average number of confined days: $\bar{N}_{confinment} = 158.06$
- Average cumulative reward: $\bar{R}_{cumulative} = 38.91$

Comparing with the professor's policy, we see that our best DQN policy is more efficient in reducing the number of deaths. Indeed, the DQN policy reduces the number of deaths by 55'000 people in average compared to the professor's policy case, it represents a reduction of 95% in average which is remarkable. Apart from this, we have a positive average reward of 38.9 while the professor's policy had an average reward of -69. The DQN policy is then more efficient in reducing the number of deaths and in keeping the society functional (positive reward). The only drawback is the increased number of confined days which is higher in the DQN policy case.

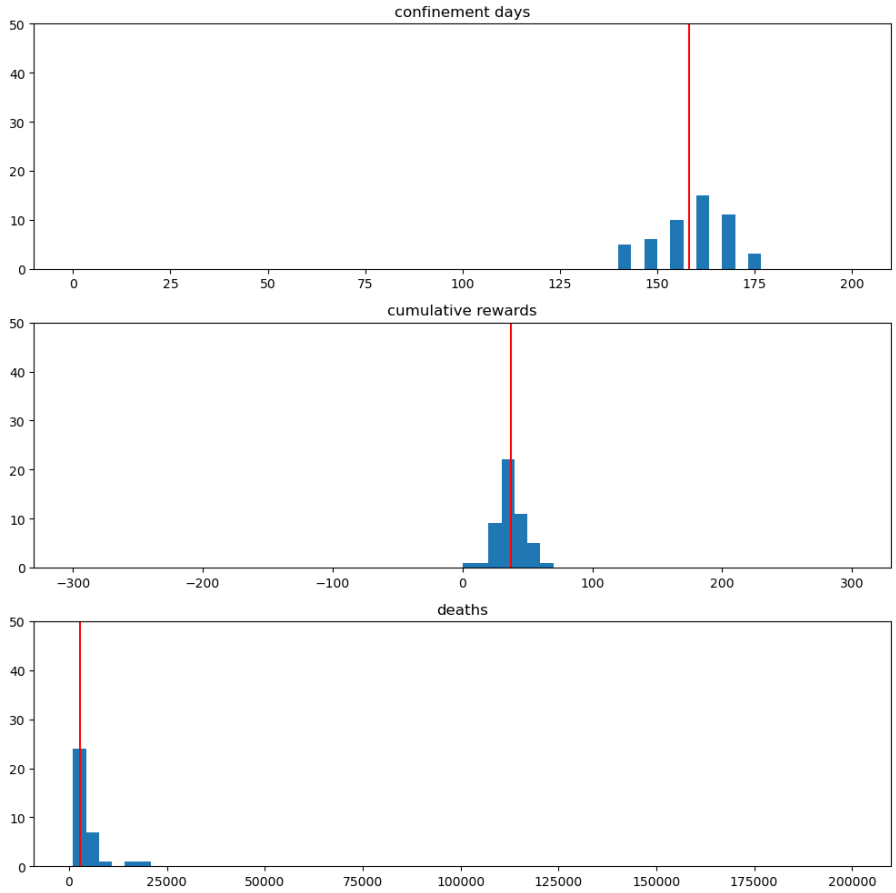


Figure 7: DQN best policy evaluation

4 Dealing with a more complex action Space

4.1 Toggle-action-space multi-action agent

Question 4.1.a) (Theory) Action space design

Answer The action-observation space is now representing the change to the system we want to make. It is not about how the agent intervene in the system. It is how the agent wants to change rules in the system. The action-space is now a 1D boolean array of size 5.

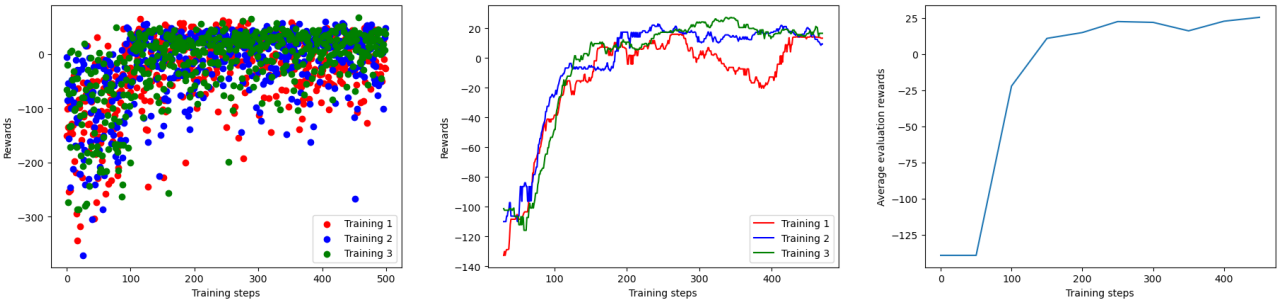
In a dynamical environment like this one, the optimal action changes over time. The agent has to adapt to the environment. Instead of having fixed actions with specific changes, the toggle action space is more flexible and allows higher maneuverability of the system.

However, this affects the network architecture which takes a more complex input. The input now also includes the state of the actions in addition to the state of the system. This implies that the network must be designed to accommodate these more complex preprocessors, which require additional layers and modifications to handle the augmented input.

Training with the action-observation space introduces a new challenge in learning the Q-values. The Q-values of toggle actions are dependent on the current state of each action, which means they can change dynamically during training. This introduces additional complexity in estimating accurate Q-values. The agent needs to learn not only the optimal Q-value for a given action but also how the state of that action affects its Q-value. This can lead to a more challenging training process, requiring longer training times and potentially more complex algorithms to converge to optimal policies.

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

Here we simply increased the number of input and output dimensions of the policy network (as well as the target network). We also choose a higher learning rate $lr = 1.0e-3$ *since the training which lead to a much better training*



(a) Scatter plot **(b)** Centred moving medians **(c)** Training evaluation

Figure 8: Toggle-action-space multi-action policy training with epsilon decay

The agent is learning properly since the evaluation trace is increasing with the number of training steps. The training episode with the best trained policy π^*_{Toggle} (the first one) is represented on Figure 9.

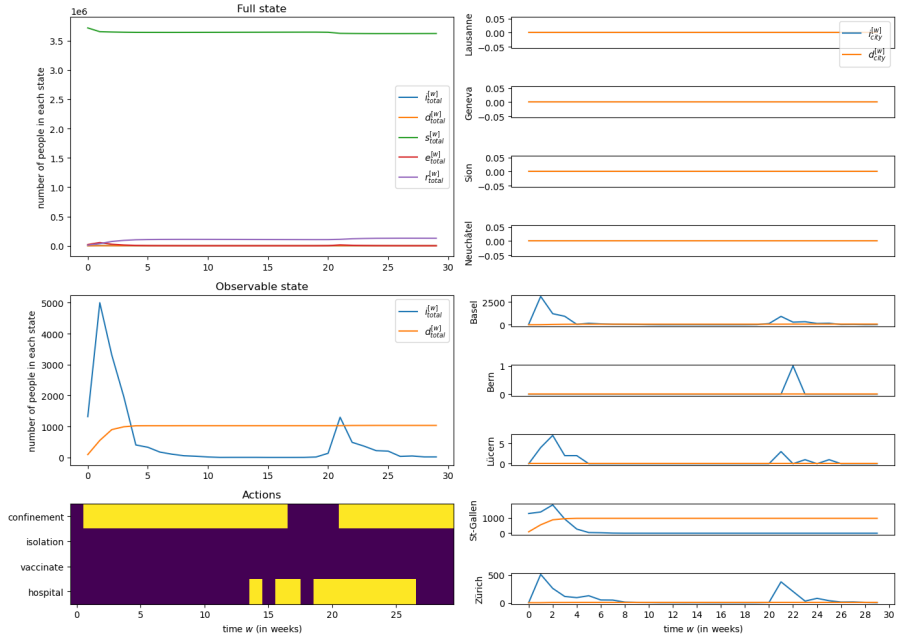


Figure 9: Epidemic simulation with Multi-action Agent Toggle Action Space Best policy

One can first observes a total number of deaths just below 20'000 at the end of the episode. The number of infected drops at the beginning, has a small peak value of 20'000 around week 9, and then drops to zero again. The policy follows long confinement periods with a few hospital bed adding actions. The best policy does not perform any vaccination or isolation. Two cities were left untouched.

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

See Figure 10.

- Average death number: $\bar{N}_{deaths} = 10868.66$
- Average number of confined days: $\bar{N}_{confinment} = 153.02$
- Average cumulative reward: $\bar{R}_{cumulative} = 27.72$

The average number of deaths is more than two times higher for the Toggle action than for the Binary action case. The rewards are also more than two time higher for the Binary agent case which means that even for a higher number of confined days $\bar{N}_{\text{confinement-Binary}} > \bar{N}_{\text{confinement-Toggle}}$, we still have $\bar{R}_{\text{cumulative-Binary}} > \bar{R}_{\text{cumulative-Toggle}}$. To sum-up, the Binary action agent performs better than the Toggle action space agent.

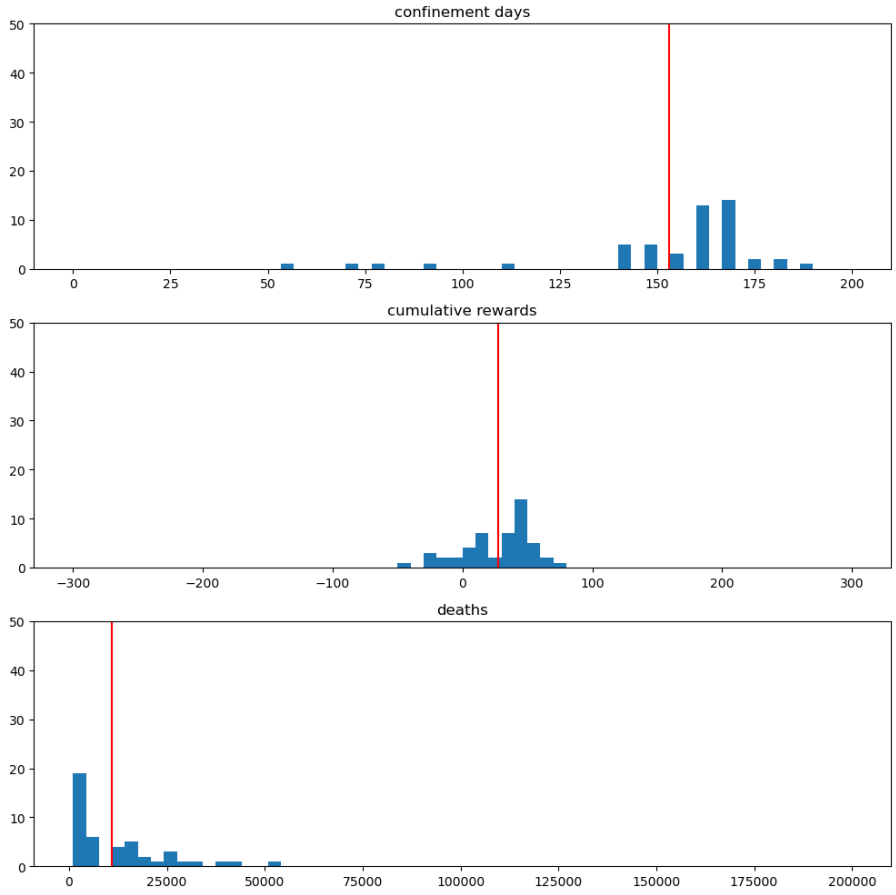


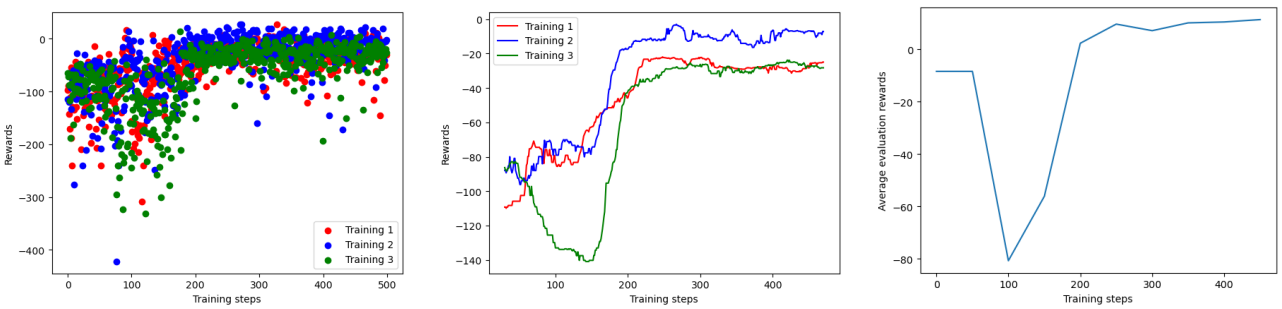
Figure 10: Toggle space action Agent best policy evaluation

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

Answer The assumptions made for the toggle action space is that there is only one action possible after each observation and the actions are binary. However, actions with continuous values would not be fitted for the toggle space. For instance, one could vaccinate people from a certain age range which would not be possible with the toggle action space because it would add a very high number of possible actions for all possible age ranges. But this observation can be said for the algorithms in 3. and 4.2 action-spaces. What is mostly particular here is the single action at each step.

4.2 Factorized Q-values, multi-action agent

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



(a) Scatter plot **(b)** Centred moving medians **(c)** Training evaluation

Figure 11: Multi-action factorized Q-values policy training

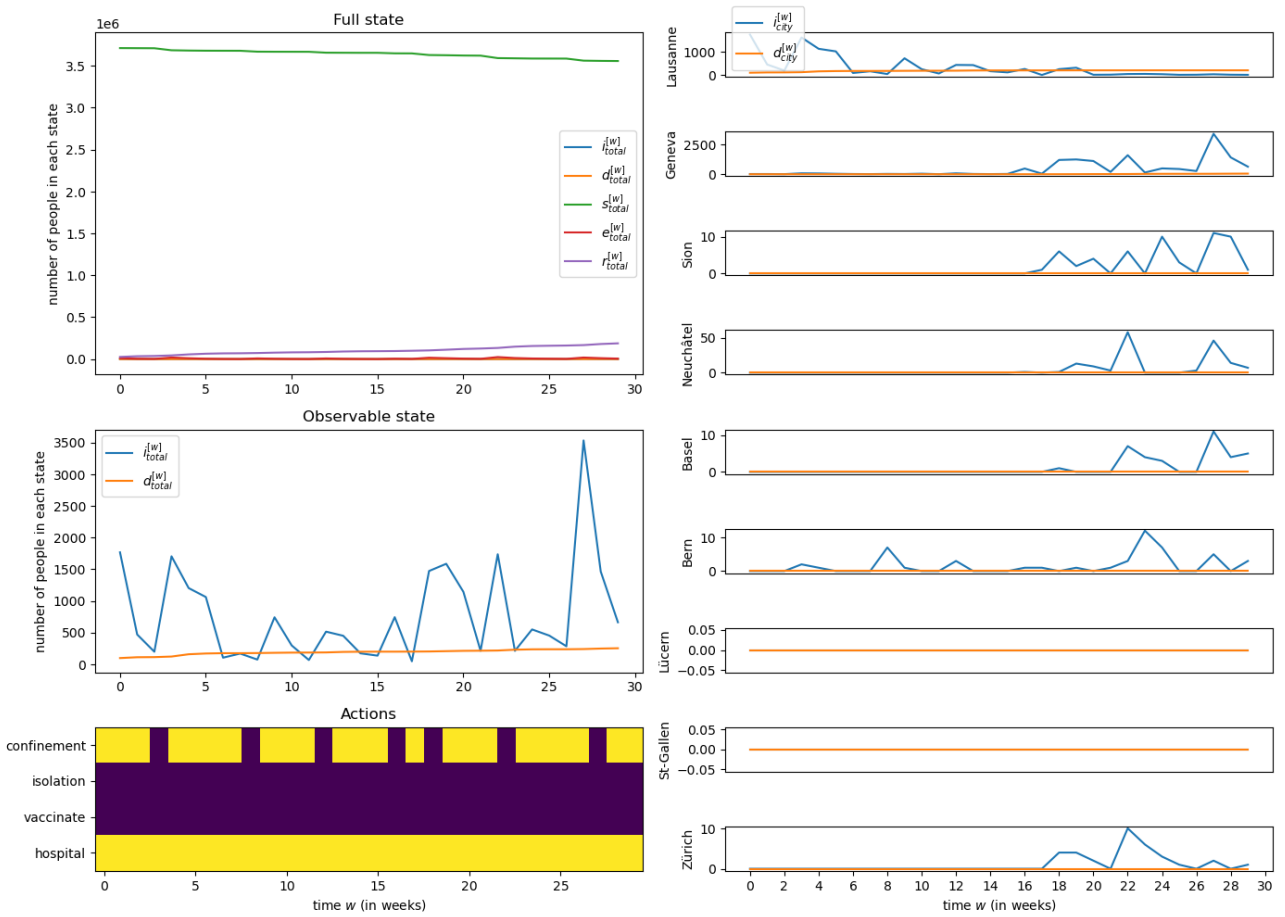


Figure 12: Epidemic simulation with Multi-action Agent factorized-Q-Values Best policy

See Figure 11. The policy successfully learn with an increase and stabilization of the training trace. Same observation for the evaluation rewards averaged over the three trainings. See Figure 12. Plotting for one episode our best performing policy for the factorized Q-value policy, we observe a very low number of deaths at the end of the episode. We also observe a constant action of adding the hospital beds. This is obviously unrealistic as there is a limit in the number of hospital beds we can add. To sum-up, this policy reduces considerably the number of deaths and infections, with the drawback of taking heavy actions with many hospital beds adding and confinement days.

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

See Figure 13.

- Average death number: $\bar{N}_{deaths} = 1011.76$
- Average number of confined days: $\bar{N}_{confinment} = 157.5$
- Average cumulative reward: $\bar{R}_{cumulative} = 4.77$

Compared to the toggle-space agent, the agent here decides more confinement days linked to a lower number of death. It also has a lower cumulative reward. It is positive in average but often slightly negative for half of the results.

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

Here we assume that the Q-value of an set is the sum of the Q-values of each action in the set, e.g.

$$Q(obs, \{\text{confine, not isolate, vaccinate, not hospital}\}) = \\ Q(obs, \text{confine}) + Q(obs, \text{not isolate}) + Q(obs, \text{vaccinate}) + Q(obs, \text{not hospital})$$

Which is assuming that these action have an independent effect on the environment (and on the reward). This probably wrong e.g isolate is probably ineffective if there is already a confinement

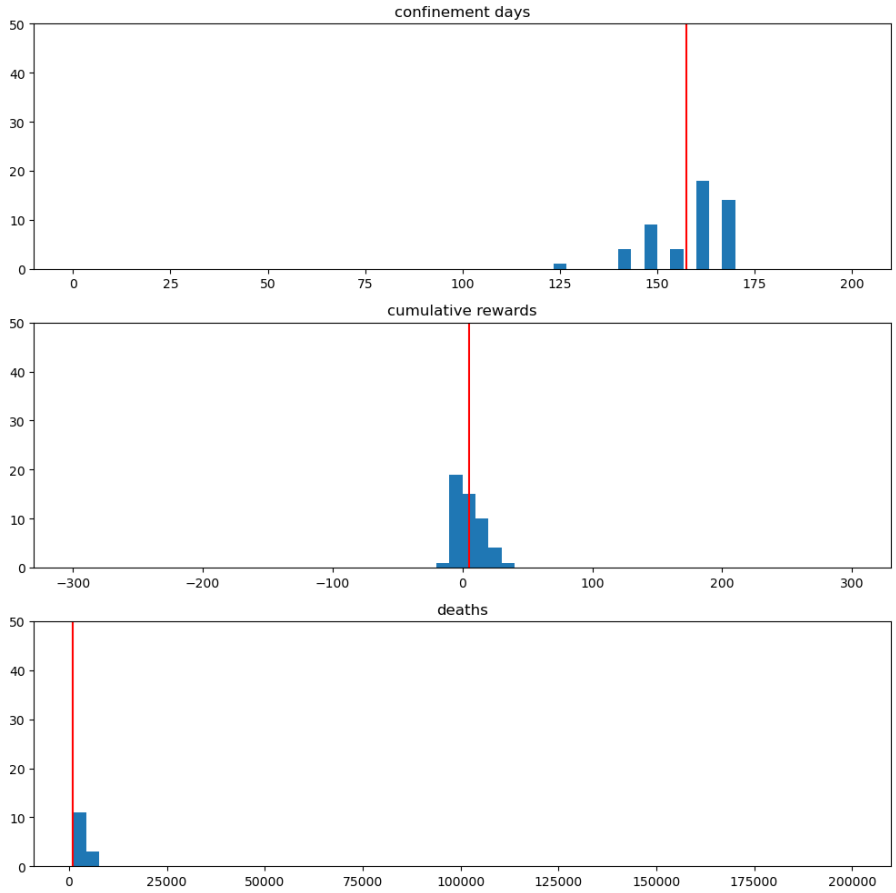


Figure 13: Factorized-Q-values Agent best policy evaluation

5 Wrapping Up

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

Answer The performance differences among Professor Russo's Policy, single-action DQN, factorized Q-values, and toggled-action-space policies can be observed from their evaluation and training curves. Professor Russo's Policy has constant evaluation rewards, making it computationally efficient. The single-action DQN agent shows rapid initial improvement and then a slower increase in evaluation rewards, depicting a fast convergence towards optimal policy. Despite a lesser noisy training for the toggle and factorized case, their evaluation trace converge slower. This difference with the single action DQN agent shows that the latter has a simpler action space, resulting in faster exploration with the same training steps and learning rate parameters (the lr is enven higher for the two other cases).

Question 5.b) (Result analysis) Comparing policies

Answer The best performing policy for the different metrics are presented on the last row of Table 2.

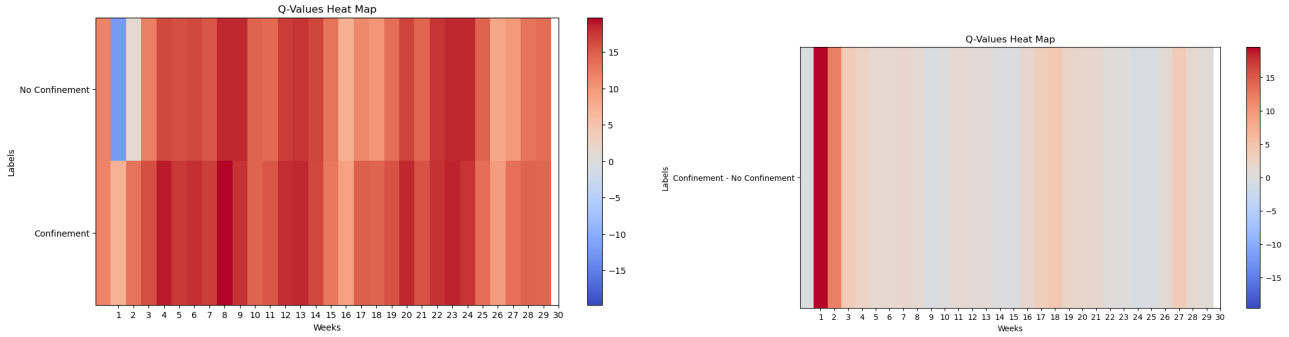
Policy	$\overline{N_{\text{confinement}}}$	$\overline{N_{\text{isolation}}}$	$\overline{N_{\text{vaccination}}}$	$\overline{N_{\text{hospital}}}$	$\overline{N_{\text{deaths}}}$	$\overline{R_{\text{cumulative}}}$
π^*_{Russo}	100.24	UA	UA	UA	58071.94	-69.96
π^*_{DQN}	158.06	UA	UA	UA	3718.26	38.91
π^*_{Toggle}	153.02	3.92	24.92	17.64	10868.66	27.72
π^*_{fact}	157.50	0.00	0.00	183.40	1011.76	4.77
Best Policy	π^*_{Russo}	π^*_{fact}	pi^*_{fact}	π^*_{Toggle}	π^*_{fact}	π^*_{DQN}

Table 2: Average episode performance metrics for the best policies of each algorithms (UA: Unavailable action)

The two parameters of isolation and vaccination were not used for the factorised Q-values case. This is probably due to the cost of taking such actions. It might be interesting to study how this metric changes with the cost of the action. The cumulative reward is much higher for the DQN case, and negative for the Russo policy. The factorised case displays the lowest number of death but with the cost of heavier action over the system with a lot of hospital beads adding and the maximum number of confinement days. The Toggle space has a number

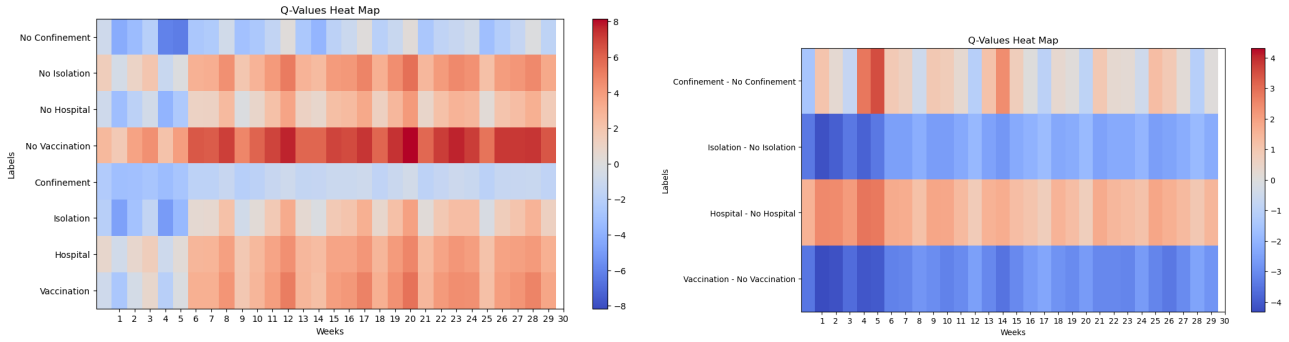
of confinement days approximately equal to the one of DQN but the number of deaths is almost three times higher while it had access to multiple actions. To sum-up DQN agent has better performance just by using the binary confinement action.

Question 5.c) (Interpretability) Q-values



(a) Heat-map of Q-values **(b)** Heat map of the differences of antagonist action Q-values

Figure 14: Binary DQN Q-values heat-map



(a) Heat-map of Q-values **(b)** Heat map of the differences of antagonist action Q-values

Figure 15: Factorised Q-values heat-map

For binary DQN, we observe that all Q-values are, most of the time, greater than 5-10. The Q-values for *confinement* are often greater than the ones for *no confinement* which is coherent with the observed policy. The *no confinement* Q-values are very same at week 1. i.e when the number of contagions rises, which is coherent. For the factorised Q-values, we observe that the Q-values for *hospital* are always greater than the ones for *No hospital*. Hence new hospital bed are added every week, which is what we observed in the policy evaluation. We observe the opposite for the *vaccination* and isolation.

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

With a similar network architecture and similar training time, the cumulative rewards IS NOT an increasing function of the number of actions. This was observed for the DQN single action agent which outperforms the other multi-action agents. Small set of actions can lead to an easier and faster convergence towards the optimal policy.