Report Emidemics DQN

Alexia Dormann, Alexander Popescu

June 5, 2023

1 Introduction

Question 1 a.)

When no actions are taken, the epidemic develops freely until people are no more susceptible. Indeed, they are either dead or recovered and therefore immune to further infection. As expected it follows more or less a generalized logistic growth model. What is also interesting to see, is that in the biggest cities like Zurich and Geneva the number of deaths peaks before the number of infected and generally has more death when adjusting for the number of infected indicating logistical problems in hospitals. What's also amusing to see is the so-called Rösti Graben or the separation of the french speaking and German-speaking cities, since the Swiss German cities evolve a little faster than the French-speaking cities. Additionally, cities that are close to the Austrian and German borders get infected first showing that the infection made its way probably from the German-speaking part of Europe into Switzerland. (Fig. 1)

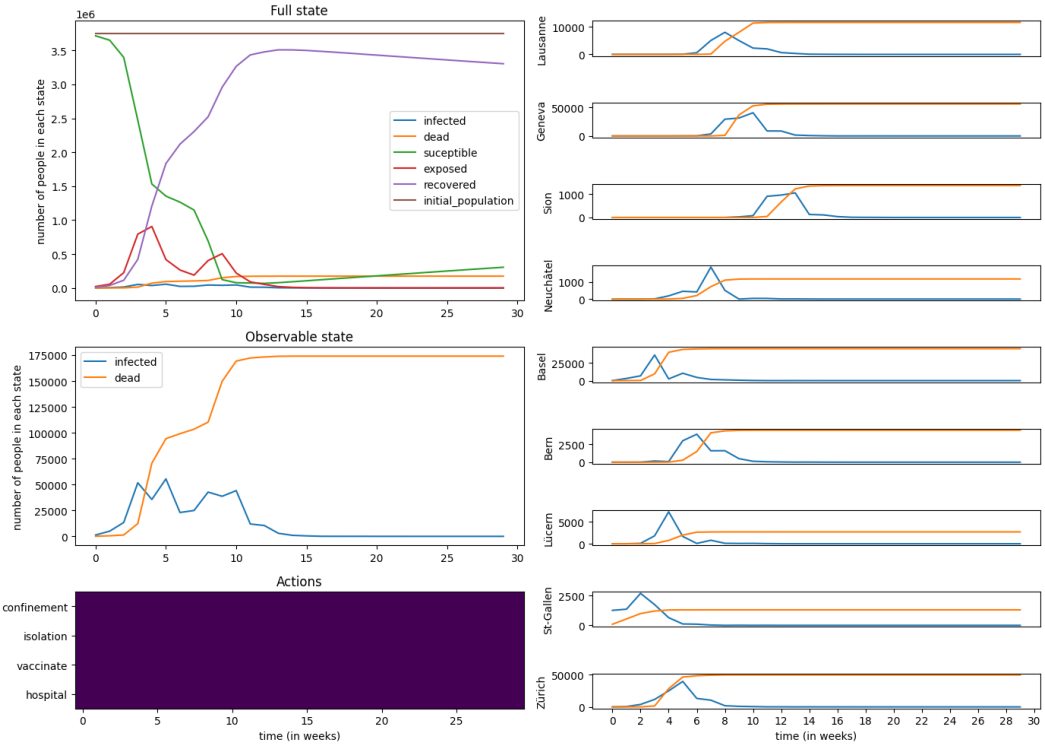

Figure 1: Epidemic evolution without actions

2 Professor Russo's Policy

Question 2a

Clearly, we can see that there are fewer deaths and infections at one time and that in general the profile of infected and dead people is more stretched out over time meaning that hospitals and other infrastructures are less overloaded. The less overloaded hospitals also imply that people can get treated faster and better which decreases the total number of death in an episode. Additionally, we see that some cities like Lausanne or Geneva, get first infected later on. (Fig. 2)

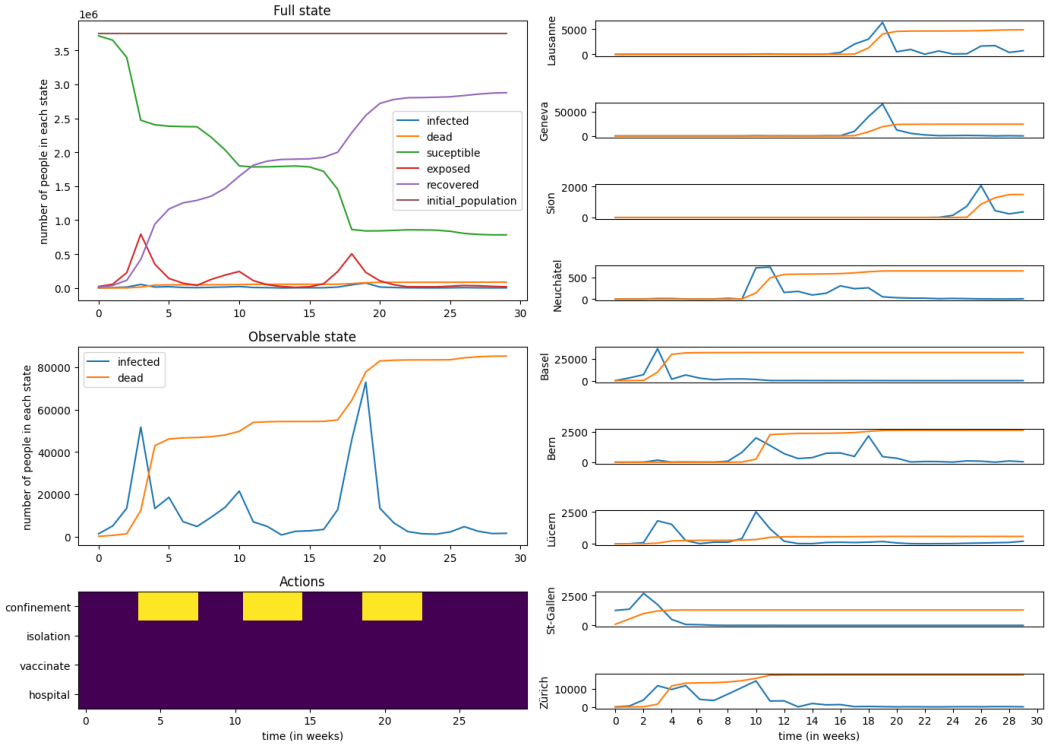

Figure 2: Epidemic evolution with Russo policy

Question 2b

The number of deaths never surpasses 112'000 in any of the episodes which is lower than the 175'000 we got in the simulation without actions. Furthermore, the mean deaths per episode (about 58'000) is significantly lower. (Fig 3)

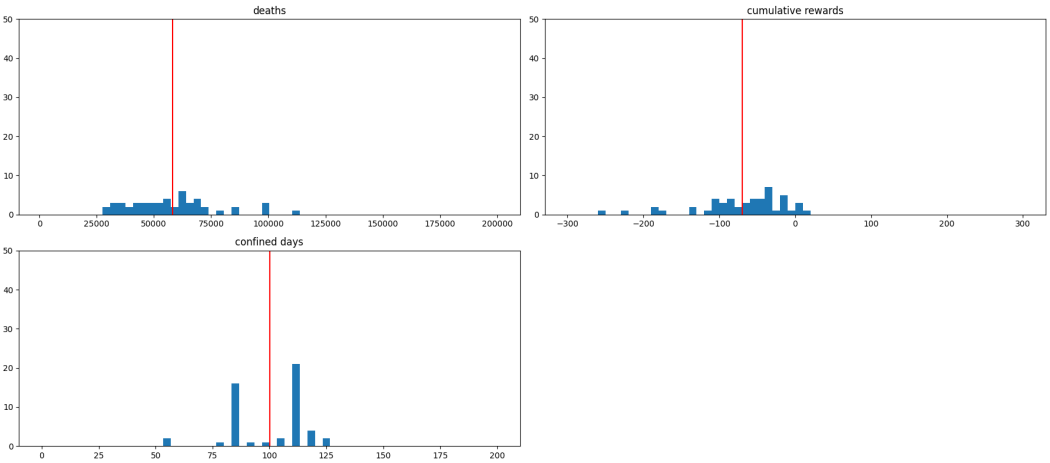

Figure 3: Histograms of the evaluation averaged over 50 episodes of the Russo policy

3 A Deep Q-learning approach

Question 3a

The DQN agent trained as described in the project description seems to perform a lot better than the Russo policy. Thanks to the agent's action, the number of deaths did not even reach 2000. In fact, the virus did not even manage to spread to the french speaking parts of Switzerland. No infections or deaths were recorded there. So at first sight it seems to perform exceptionally. However, this performance comes at a cost. Looking at the actions taken it becomes evident that the agent excessively performs the action of confinement which clearly helps with the number of deaths but is not sustainable in the long run. Moreover, the number of susceptible is still quite high. Therefore, as soon as the confinement is lifted off a new wave of infection will come. So it seems that the model is relying too much on the reward of low deaths to achieve its performance. To avoid this issue, we may consider increasing the cost of confinement. (Fig. 6)

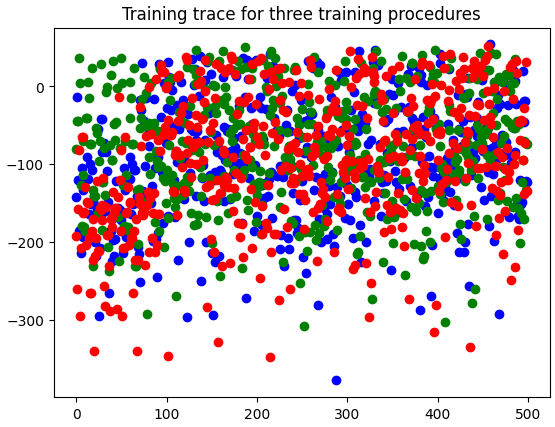

Figure 4: Scatter plot of cumulative reward (y-axis) with respect to the episode(x-axis) during training of binary-DQN and epsilon=0.7

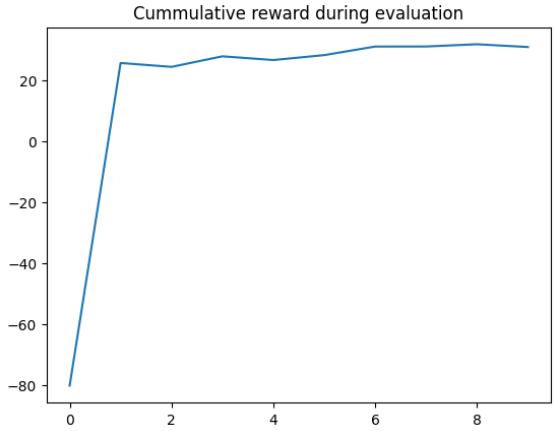

Figure 5: Cumulative reward during evaluation (average cumulative reward: y-axis, every 50 episodes: x-axis during training of binary-DQN and epsilon=0.7

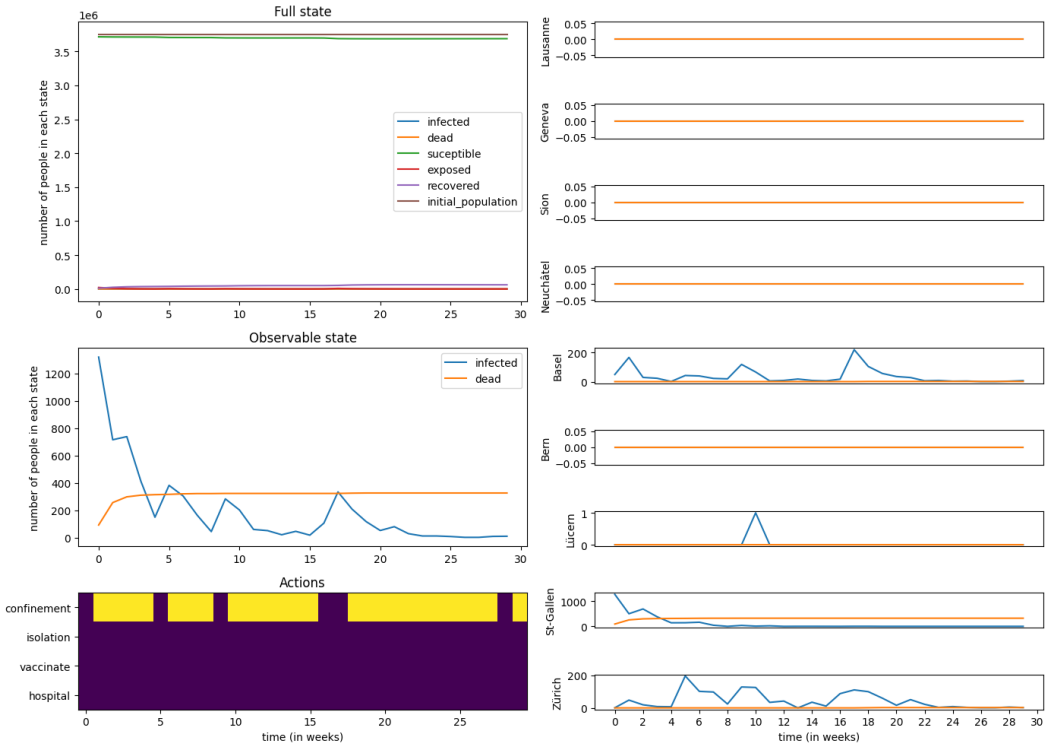

Figure 6: Evolution with DQN-agent with epsilon=0.7 and only 2 possible actions, namely confinement or no confinement

Question 3b

The model with the decaying epsilon achieves a better cumulative reward (22 vs 27). (Fig. 5, 8) Indeed, the decaying epsilon achieves better results as it allows for early exploration. Indeed, the initial network has not yet learnt anything about the environment, therefore, it makes little sense to follow its decisions. Instead, it is better to randomly explore the environment. Additionally, at the later stages of the training, the epsilon with be reduced, therefore, more often the actions recommended by the network will be followed. By doing so we are able to explore the states near this learnt optimal. Note that exploration remains relevant as it might help the network to not get stuck in a local minimum.

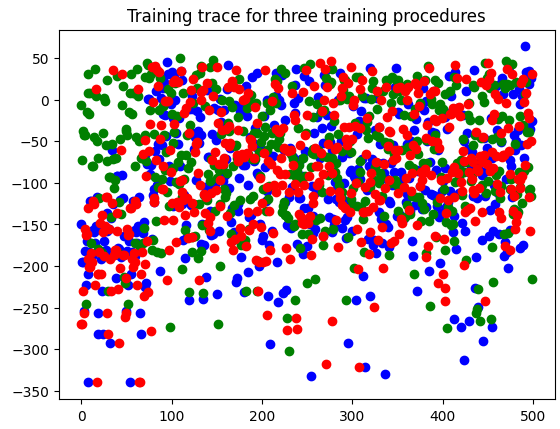

Figure 7: Scatter plot of cumulative reward (y-axis) with respect to the episode(x-axis) during the training of binary-DQN and decreasing epsilon

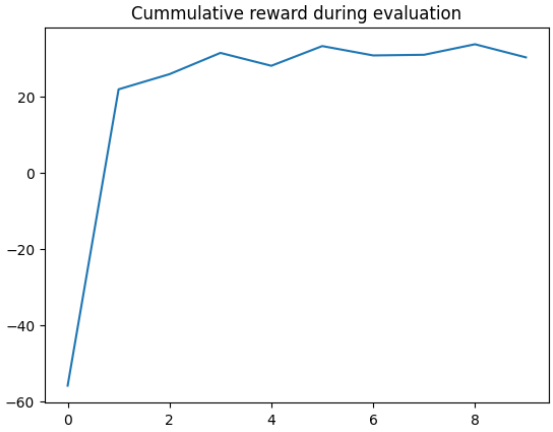

Figure 8: Cumulative reward during evaluation (average cumulative reward: y-axis, every 50 episodes: x-axis during the training of binary-DQN and decreasing epsilon

Question 3c

The reinforcement policy learnt in part 3 did improve compared to the Russo policy in the sense that it increases the cumulative reward per episode (Fig. 8). However, as mentioned before it learnt an unsustainable behaviour. Indeed, it relies too much on confinement to keep the number of infected and death low. But, as very few people are infected most of the population remains susceptible. Meaning that as soon as the confinement is lifted, they will be a new wave of infections. This highlights the need for additional actions such as vaccination, isolation, and increasing the number of hospital beds.

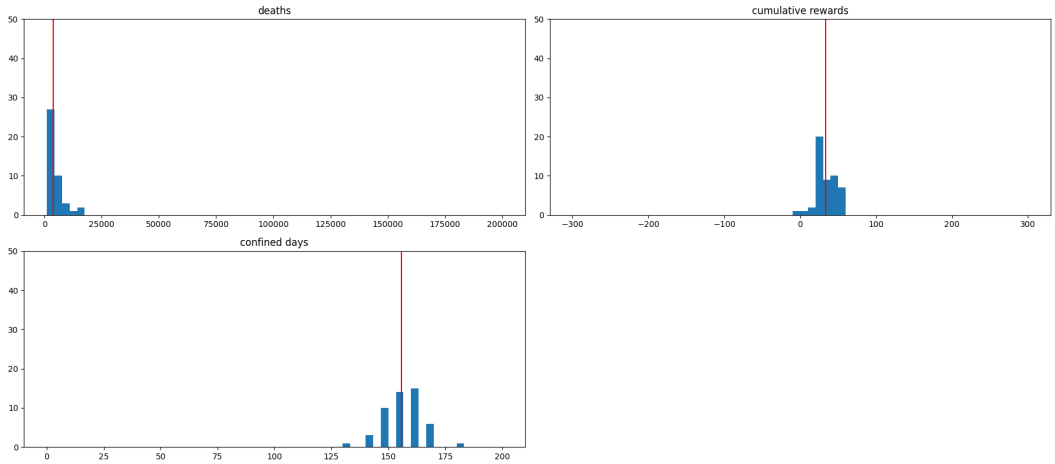

Figure 9: Evolution with binary DQN-agent with decreasing epsilon [0.7-0.2]

4 Dealing with a more complex action Space

Question 4.1.a

- *Reduced Action Space*: The proposed action space reduces the complexity of the action space by having "toggle" actions instead of having separate actions for each possible change (e.g., enabling and disabling each policy). This means the agent doesn't need to learn separate Q-values for "enable isolation" and "disable isolation"; instead, it learns a single Q-value for "toggle isolation".
- *Better Handling of Environment Dynamics*: In dynamic environments where the same action can have different effects depending on the current state of the environment (e.g. if isolation is currently enabled or disabled), it can be beneficial to include the current state of the actions in the observation space. This allows the agent to learn more effectively by making its policy contingent on the state of the environment and the action.
- *Increased Observability*: Including the current state of each action in the observation space helps to make the environment more fully observable. It provides the agent with additional context that can be important for decision-making.
- *Efficiency in Network Training*: The reduced action space can help to streamline the network architecture and facilitate faster and more efficient training. A smaller action space can result in a less complex Q-function, which can ease the computational requirements and potentially speed up convergence during training.

Question 4.1.b

We see that the model was able to learn a policy. Indeed, the cumulative reward increases over time (Fig. 11) However, the policy still seems to rely a lot on confinement and increase hospital beds actions to decrease the number of infections and the number of deaths. The vaccination and isolation are never toggled. Again we observe that the percentage of the population that is susceptible decrease very slowly. (Fig. 12). For some reason the agent did not explore other action than confinement and adding additional hospital beds which could be due to being stuck in a local minimum and having a to small epsilon for more exploration.

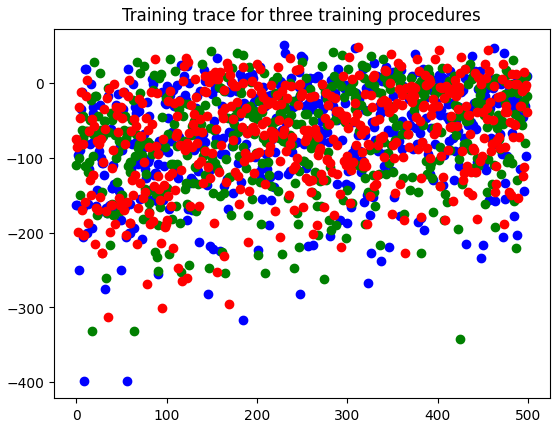

Figure 10: Scatter plot of cumulative reward (y-axis) with respect to the episode(x-axis) during training of toggle-DQN agent

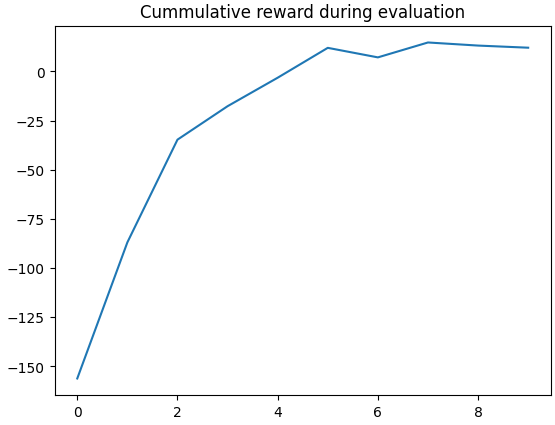

Figure 11: Cumulative reward during evaluation (average cumulative reward: y-axis, every 50 episodes: x-axis during training of toggle-DQN agent

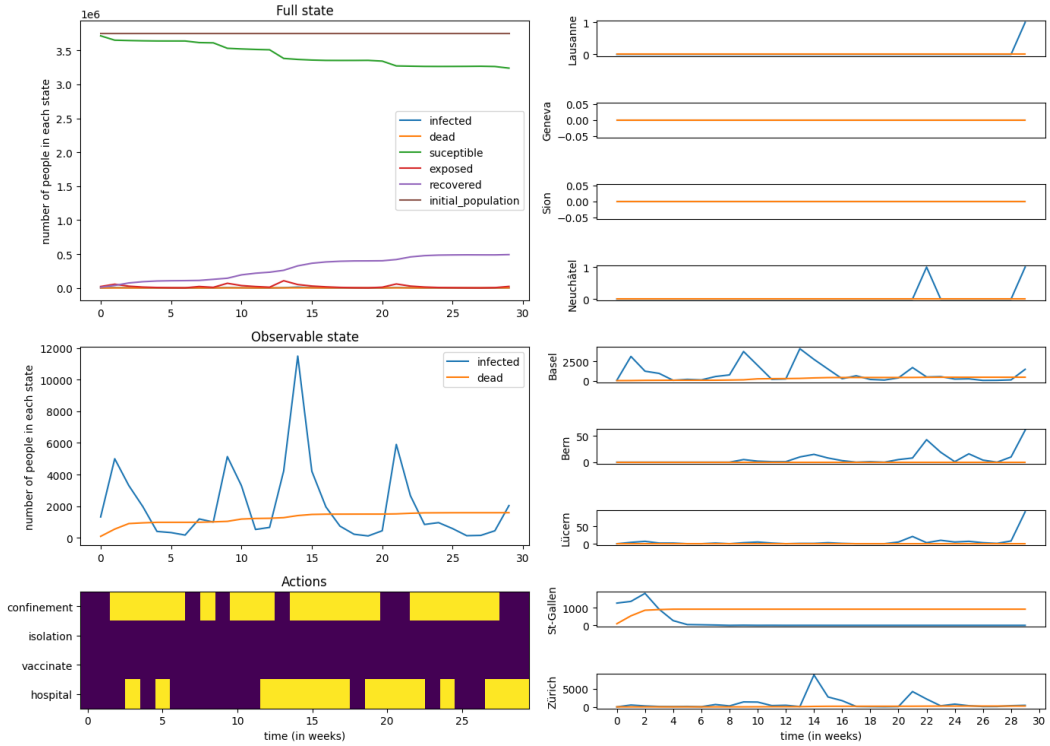

Figure 12: Evaluation of toggle-DQN on an episode

Question 4.1.c

We see that both policies achieve very similar rewards. They are both a bit above 0. (Fig. 9, 13) Interestingly, both policies rely on confinement to avoid the infections of new people, with about 155 days of confinement for both agents. However, as the toggle-DQN agent has access to a more complex action space, it also uses the hospital action, which decreases the number of deaths overall.

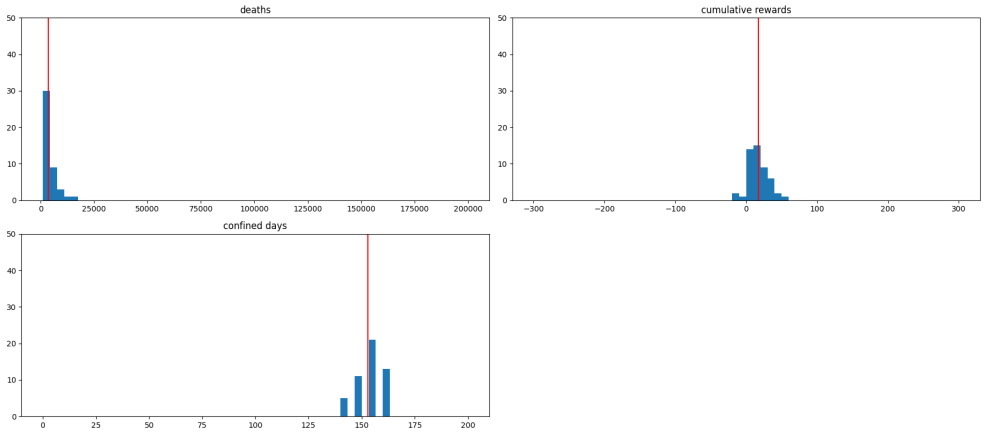

Figure 13: Histograms of the evaluation of toggle-DQN on 50 episodes

Question 4.1.d

- *Reversibility*: Actions are reversible, i.e., an action that changes the state from 'on' to 'off' can also change it back from 'off' to 'on'.
- *State Dependence*: The outcome of the action depends on the current state. This means that the effect of the toggle action is context-dependent, changing based on whether the current state is 'on' or 'off'.

- *Independence*: The actions do not interfere with each other, meaning that the state of one action does not affect the state of another action.

There are many action spaces where toggling would not be suitable:

- *Continuous actions*: For example, in a driving simulation, actions could include steering angle or acceleration level. These aren't binary and can't be effectively toggled - you can't just 'toggle' the steering wheel or the acceleration, because these have a range of possible states, not just 'on' or 'off'.
- *Multi-state actions*: Actions that have more than two possible states. For instance, in a game, you might have an action that can be 'stopped', 'walking', or 'running'. Toggling doesn't make sense here because there are three states to consider.
- *Irreversible actions*: Some actions cannot be undone by simply performing the action again. For example, in a simulation of chemical reactions, once certain chemicals are mixed, you can't 'unmix' them by performing the same action.
- *Interdependent actions*: If the outcome of one action affects the state of another action, then toggling could become problematic. For instance, in a simulation where turning on a heating system could automatically turn off the cooling system, toggling one action would affect the state of another, which could complicate the learning process.

Question 4.2.a

As one can see in Fig.16 and 15, the loss decreases over time and the cumulative reward increases. Therefore, the network seems to be learning a policy. However, even at the end of training the loss still seems to be decreasing. This indicates that the model has not reached convergence and further training would be needed.

In fig. 17, we see that the learnt policy is a bit unrealistic indeed, it always keeps the cities isolated, vaccinate the population, increases the capacity of hospitals and declares confinement very often. Indeed, it is not sustainable to forbid the population to travel between cities for so long. However, compared to the policy in part 3, the number of people susceptible does decrease over time. Meaning that at some point lifting some of the measures will no longer be associated with such a high cost.

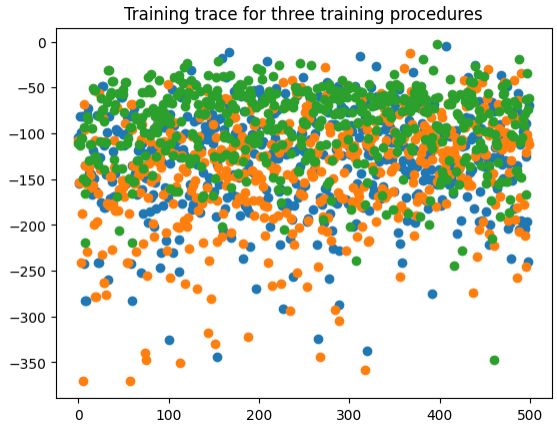

Figure 14: Scatter plot of cumulative reward (y-axis) with respect to the episode(x-axis) during training of the DQN-factorized agent

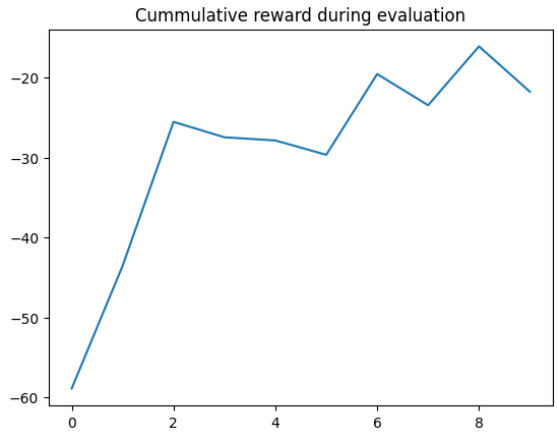

Figure 15: Cumulative reward during evaluation (average cumulative reward: y-axis, every 50 episodes: x-axis during training of the DQN-factorized agent

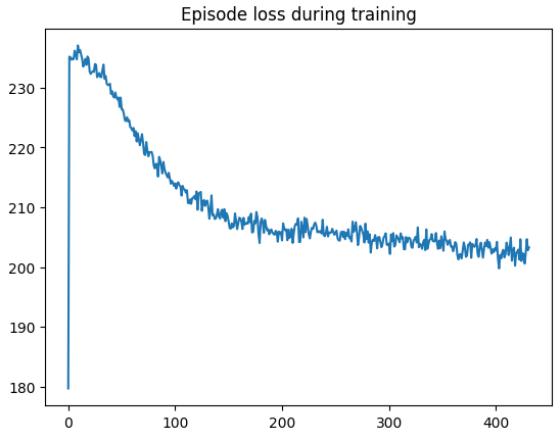

Figure 16: Loss per episode during training for the DQN-factorized agent

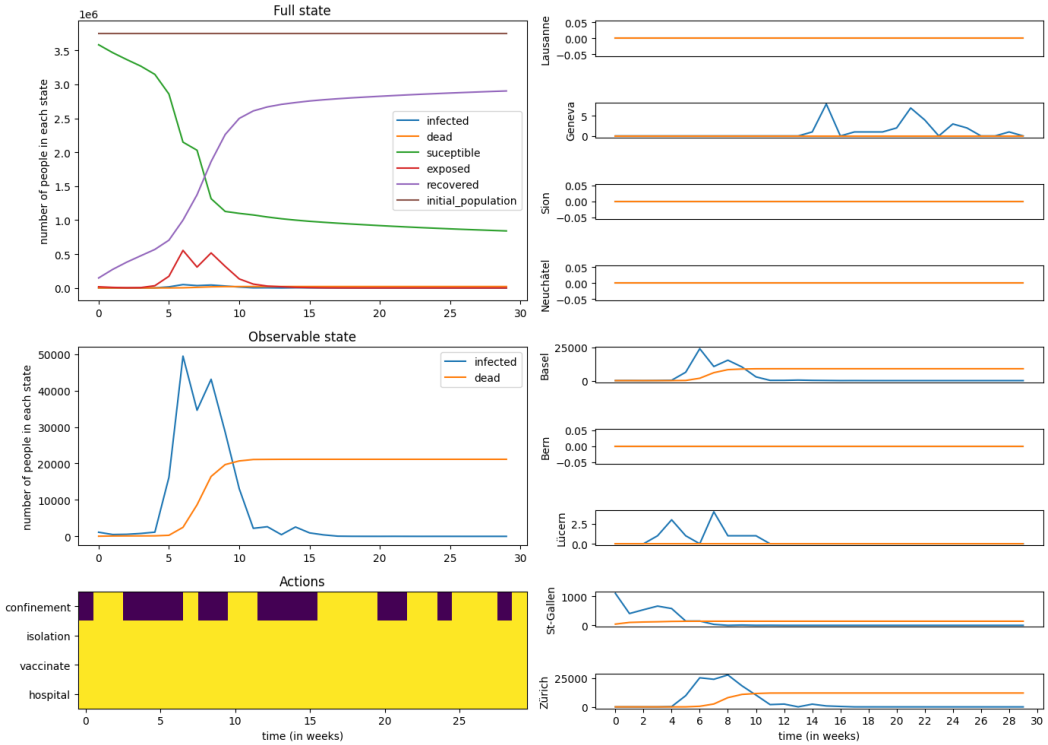

Figure 17: Evaluation of the factorized-DGN policy

Question 4.2.b

We can see that the reward is slightly smaller and that there are many more deaths with the factor DQN. However, by looking at the evolution of one episode it can be noticed that the number of susceptibles at the end of the episode is much lower for the factor-DQN indicating that it makes the epidemic "advance faster". However, this is only one episode so not really significant. Another explanation for the lower reward could be that the agent has a slightly more complex action space and similarly to the toggle version found itself trapped in a local minima of the loss without being able to explore all the actions.

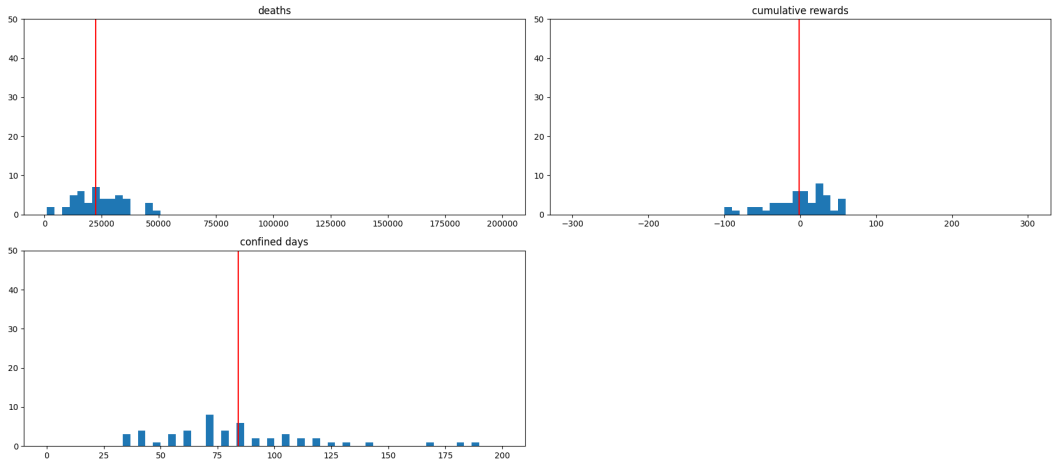

Figure 18: Histograms of the evaluation of the factorized-DQN policy

Question 4.2.c

- *Independence*: The primary assumption here is that the actions are independent of each other, i.e., the Q-value or the result of taking one action doesn't affect the Q-value or outcome of another action.
- *Additivity*: Another important assumption is that the total Q-value is the sum of the individual Q-values. This suggests that the total reward received from taking multiple actions concurrently is just the sum of the rewards of the individual actions.
- *Absence of interaction effects*: Factorized Q-values also assume that there's no interaction effect between actions. This means that the outcome of performing multiple actions together is the same as performing them independently.

There are definitely scenarios where factorizing Q-values would not be suitable:

- *Dependent actions*: If the outcome of one action depends on the outcome of another action, then factorizing Q-values isn't suitable. For instance, in a game where a door must be unlocked (action 1) before it can be opened (action 2), the Q-value of action 2 depends on action 1, so they can't be estimated independently.

Question 5.a

During the evaluation done during the training, we see that the best cumulative reward is achieved by the binary-DQN agent. However, even though this reward increases steadily during training, as discussed above it seems that for the agents with more complex action spaces, the training was more complicated. Indeed, either the training seems to have not reached convergence after the 500 episodes or it appears to get stuck in a local minima.

Question 5.b

We see that overall the agent with the more complex action space performed better. in particular, the factorDQN agent achieved the best cumulative reward on the 50 episodes evaluation. It also performs best in terms of the total number of deaths. In addition, as

	Russo	BinaryDQN	ToggleDQN	FactorDQN
Confinement	7.84	9.98	9.16	15.92
Isolation	-	-	0	0
Vaccinate	-	-	0	16
Hospital	-	-	7.42	13.1
Cummulative reward	-79.874	-20.71	-15.37	-10.58
Deaths	94244	38876	32933	290.4

Question 5.c

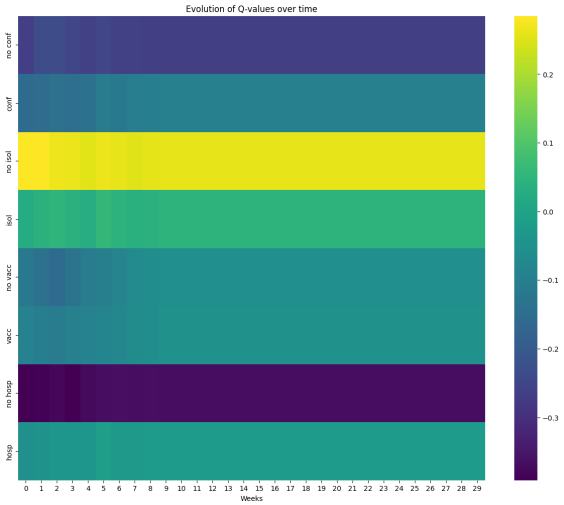

Figure 19: Q-value Heatmap of factor-DQN

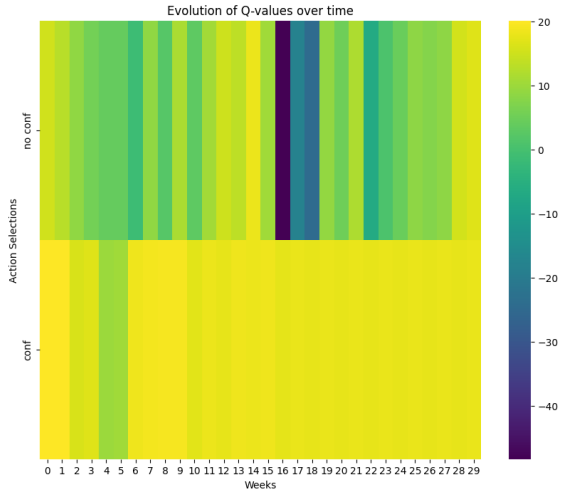

Figure 20: Q-value Heatmap of binary-DQN

4.1 Question 5 d.