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

UMLIL Erwan, CAUTRU Antoine

June 5, 2023

Introduction

Using deep Q-learning rather than simple Q-learning enables to deal with more complicated environments, without having to store the Q-values for each state-action pair. Specifically, using a neural network to predict the optimal action from a current state enables to work in a continuous action space. Here, we aim at observing the effect of political choices on epidemic propagation and training a DQN agent to achieve satisfying epidemic mitigation.

1 Epidemics unmitigated

Figure 1 shows the epidemic propagation when no action is taken. The epidemy has reached every city within 10 weeks and a total of 120k people were killed after two waves: the first one in German-Switzerland, and then in Romand-Switzerland. Once people are infected, the number of dead people increases at a high rate.

2 Professor Russo's Policy

2.1 Implementation of the policy

The epidemic mitigation with Prof. Russo's policy is visible on figure 2. The epidemy still reaches Sion but after about 24 weeks instead of 10 weeks for the unmitigated scenario, and the total number of dead reaches 60k. Overall, the confinement enables to strongly reduce the number of infected people. The epidemy rises again after each de-confinement and starts again to propagate.

2.2 Evaluation of the policy

The results of evaluation of this policy are presented in Figure 4a, where histograms show the number of total confined days, the cumulative reward as well as the number of total deaths.

3 Deep Q-learning with a binary action space

3.1 Constant exploration rate

The training and evaluation traces of the binary agent with a constant exploration rate are visible on figures 3a and 3b. Figure 5 shows the epidemic mitigation during 30 weeks with the output policy. We can interpret the policy as follows: the agent finds a balance between the number of dead people and the number of confined day, also considering that it is better to confine for two consecutive weeks rather than confine one week, deconfine and confine again (the announcement cost in the reward is integrated). **The learned policy seems to be meaningful.**

3.2 Decreasing exploration rate

The training and evaluation traces of the binary agent with a decreasing exploration rate are visible on figures 3c and 3d. This policy achieves **better results than the one that used constant**

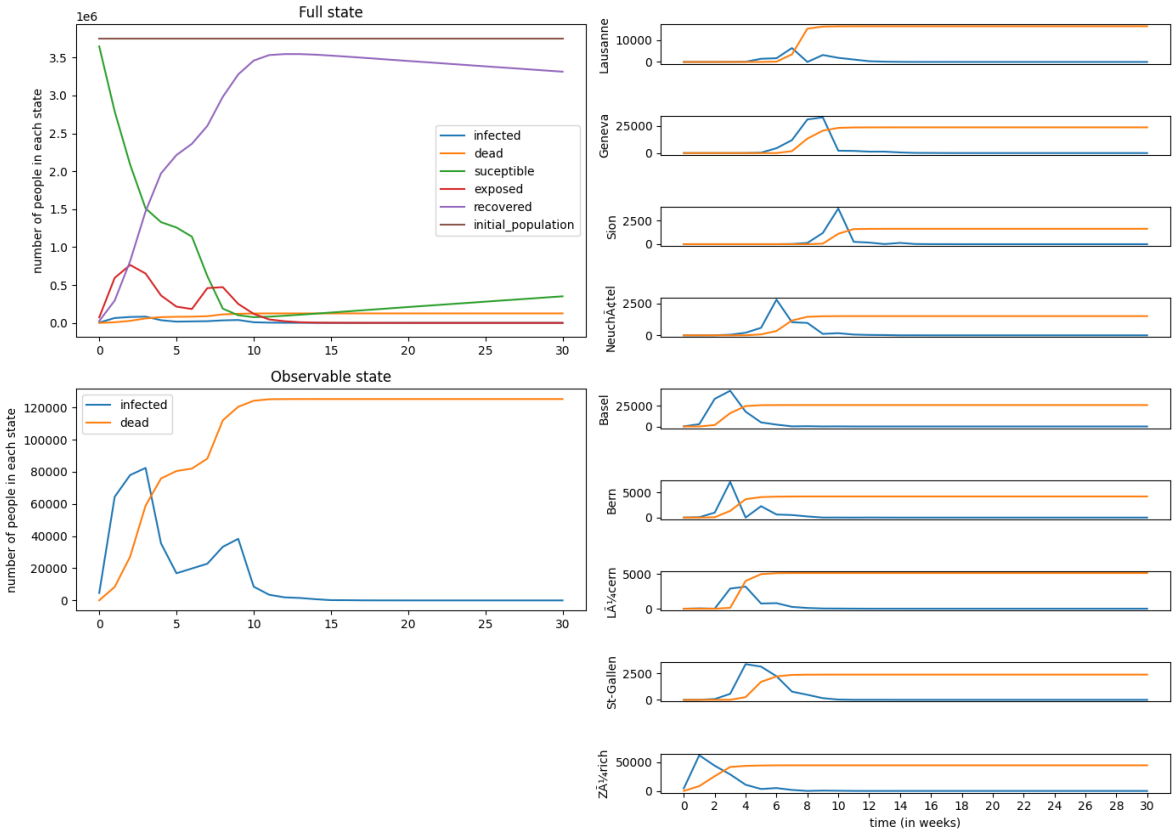


Figure 1: Epidemic propagation when unmitigated.

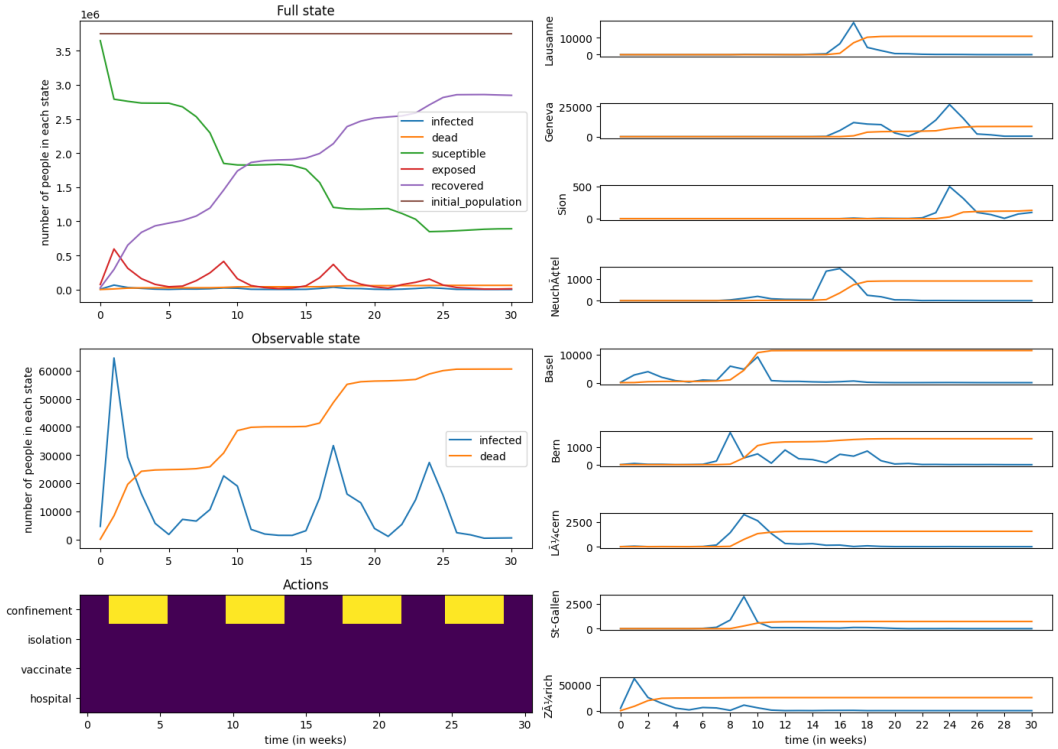
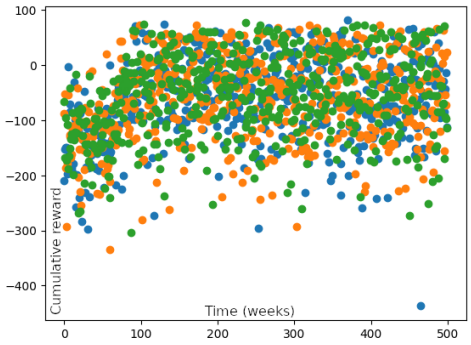
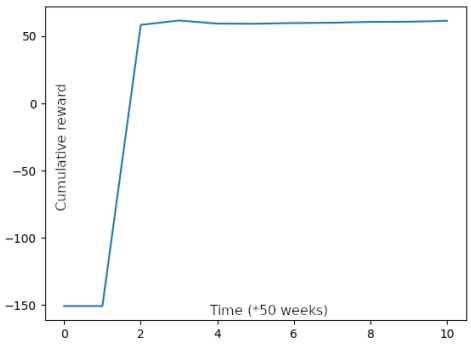


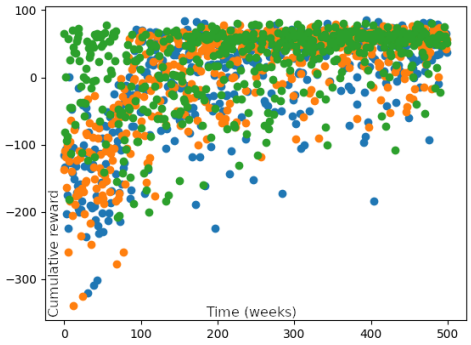
Figure 2: Implement Professor Russo's policy.



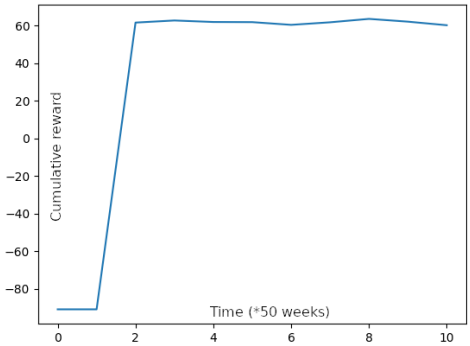
(a) Training trace, binary agent, constant exploration rate



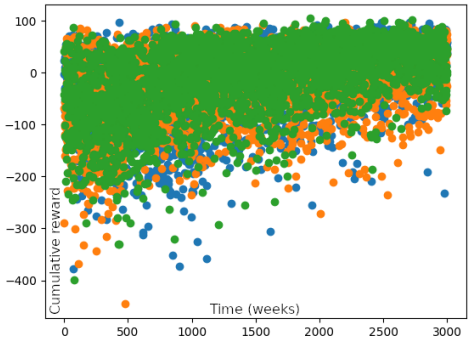
(b) Evaluation trace, binary agent, constant exploration rate



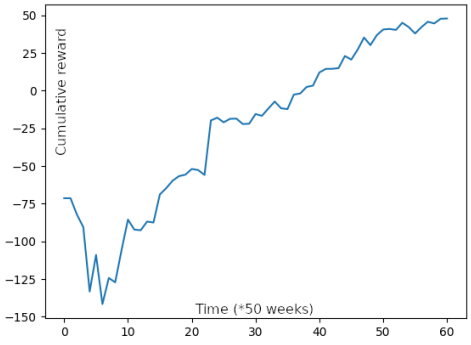
(c) Training trace, binary agent, decreasing exploration rate



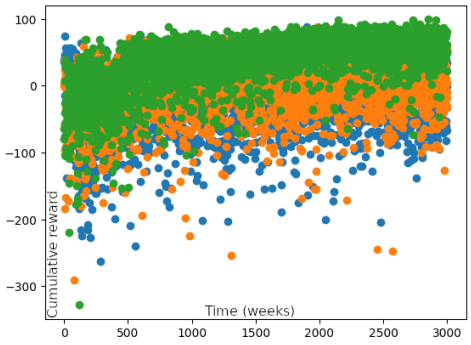
(d) Evaluation trace, binary agent, decreasing exploration rate



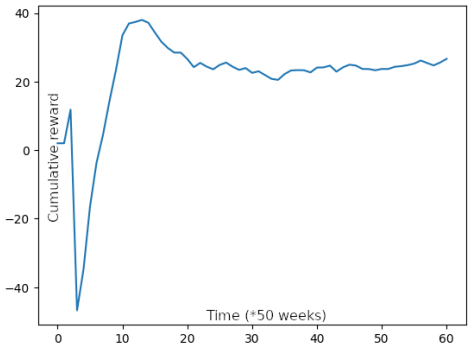
(e) Training trace, toggle agent



(f) Evaluation trace, toggle agent

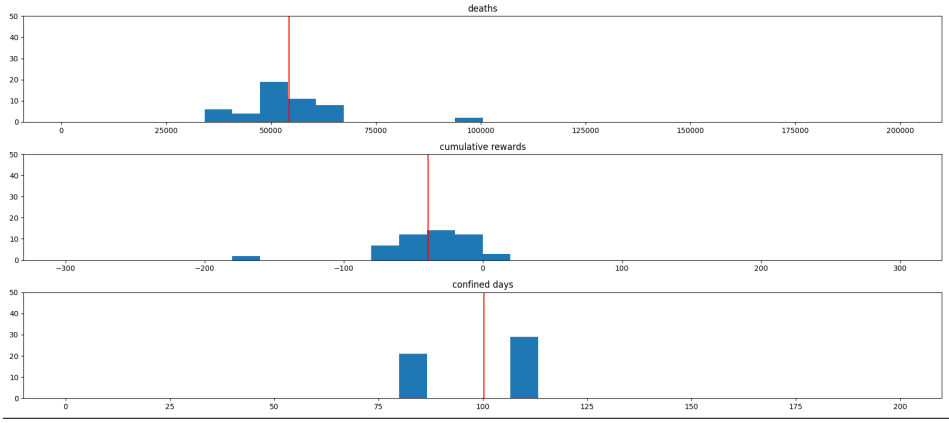


(g) Training trace, factorized Q-val agent

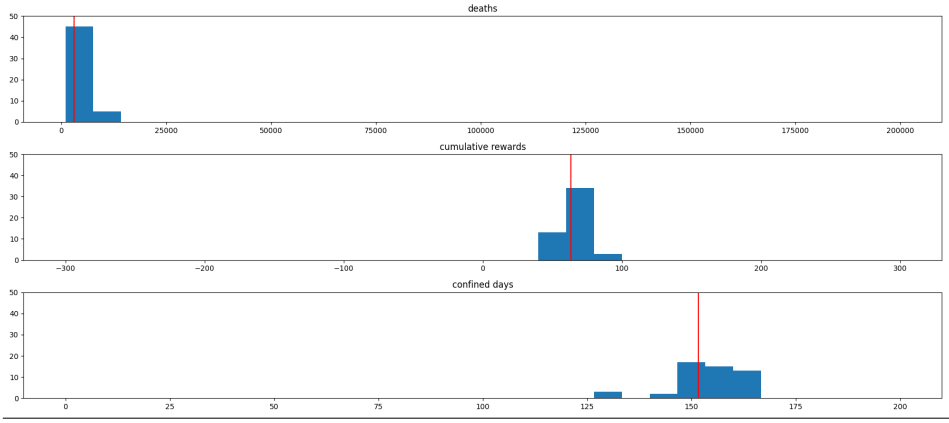


(h) Evaluation trace, factorized Q-val agent

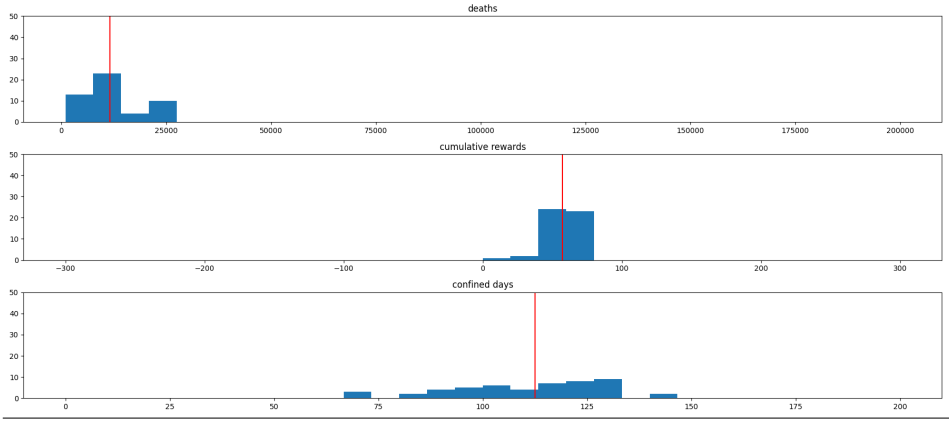
Figure 3: Training and evaluation traces of the various agents. Abscissa of left plots: episode number; abscissa of right plots: evaluation number (every 50 episodes). Ordinate: cumulative reward.



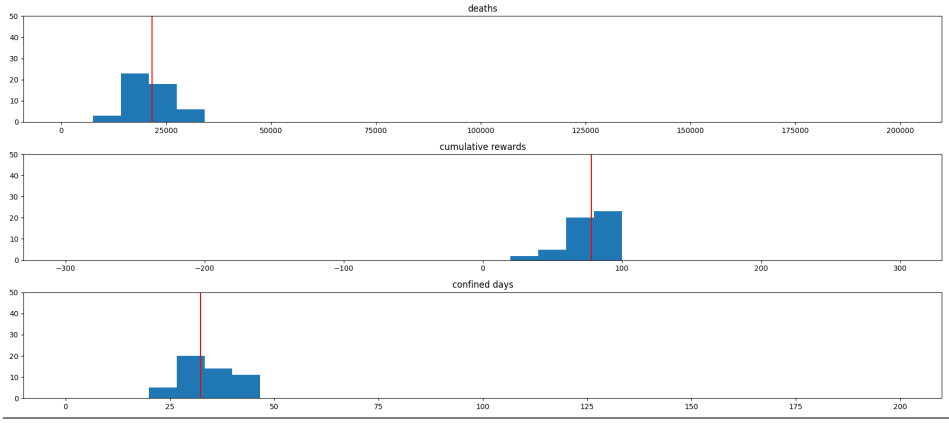
(a) Professor Russo's policy



(b) Binary agent with decreasing exploration rate



(c) Toggle agent



(d) Factorized Q-values agent

Figure 4: Evaluation of policies

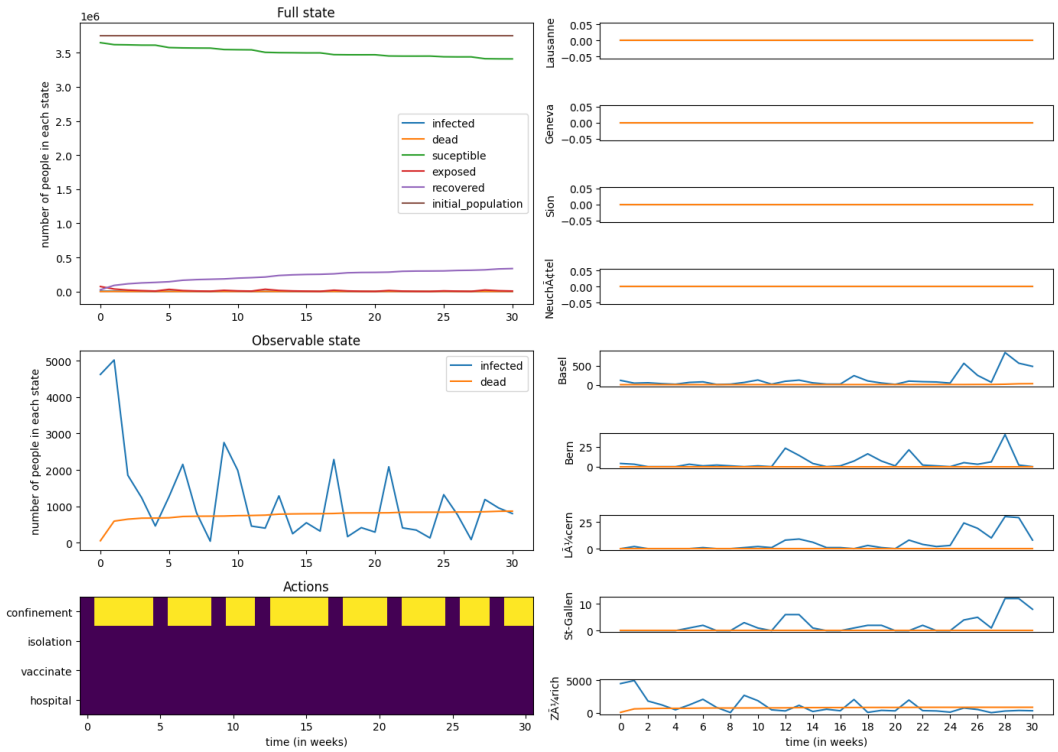


Figure 5: Implement binary agent with constant exploration rate.

exploration rate, since the cumulative reward seems to be higher if we look at evaluation traces. Besides, training traces show that this new agent achieves better training rewards than the previous one. Actually, it is logical because the agent progressively decreases its exploration, so it focuses on exploiting the best policy that was found.

3.3 Binary agent vs Pr. Russo's policy

We can observe the performance of the binary agent on figure 4b. **The reinforcement learning policy outperforms Professor Russo's policy.** Indeed, it obtains a lower mean number of deaths, and a higher mean cumulative reward. However, it is at the cost of more confined days in average.

4 Deep Q-learning with a more complex action space

4.1 Toggle-action-space multi-action agent

4.1.1 Action space design

First, such an action-observation space design allows to only consider five "toggle" actions rather than 2^4 actions that describe every possible combination, and the size of the input is slightly changed from 2x9x7 to 2x9x7 + 4. Moreover, the transition between actions play a role in the reward computing. (through the action cost). If we include the previous action in the state, the agent should learn not to switch too often from action. We should obtain a less noisy policy.

4.1.2 Toggle-action-space multi-action policy training

Training and evaluation traces are visible on Figures 3e and 3f. Both traces indicate that the agent properly learns. However this learning is slow: we need 3000 episodes to reach a reward that is equivalent to the one that is achieved by our binary agent after much less episodes.

We can interpret the policy from the episode on Figure 6. First, we notice that the agent never isolates during this episode. Moreover, it always starts vaccinating during the second half of a confine-

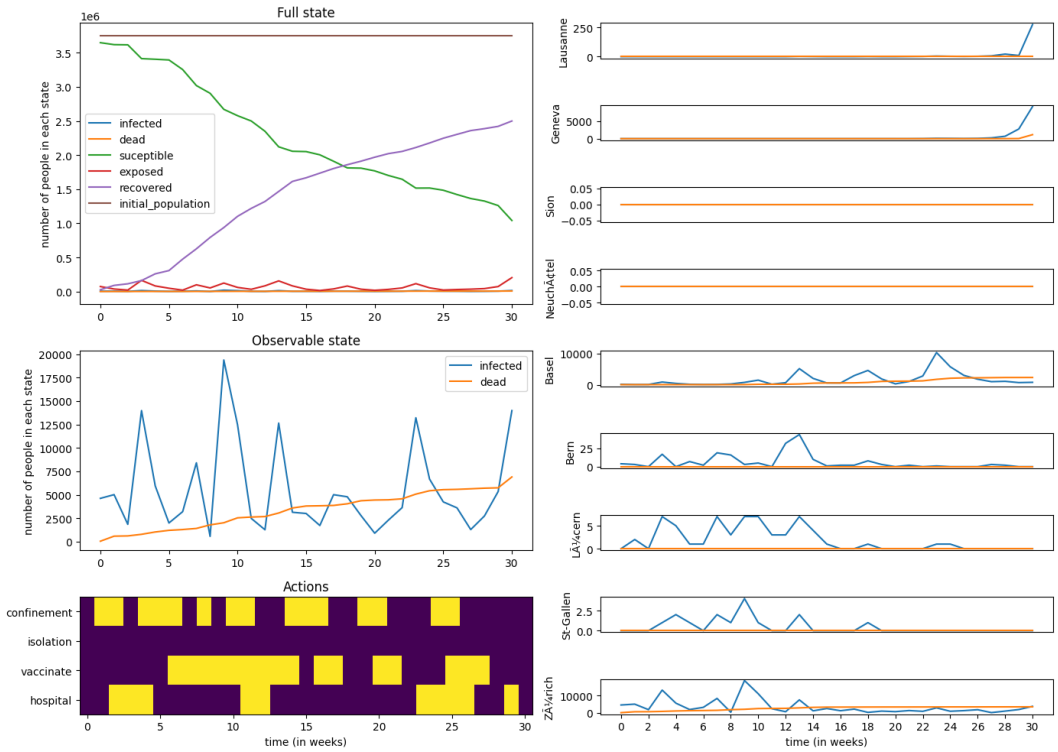


Figure 6: Implement toggle agent.

ment, probably to immunize people before stopping confinement. Finally, the agents seems to regularly add hospital beds from the beginning and until the end of the episode.

4.1.3 Toggle-action-space multi-action policy evaluation

The evaluation results are shown in Figure 4c. We can see that the agent achieves slightly more deaths than the previous binary policy, an equivalent cumulative reward, but **less confinement days** in average (112 instead of 151). So this agent seems to use its larger action space in order to reduce the number of confinement days while not increasing too much the number of deaths.

4.1.4 Question about toogle-action-space policy: what assumption does it make?

Using a toggle-action-space makes the assumption that the optimal state can be reached by taking consecutive toggled actions. It also assumes that the previous action is available.

For some action spaces, toggling the actions would not be suitable. For instance, if we had some kind of agent that can move left or right only at each step, we would have to use two actions in order to change direction of the movement: right and then left would become toggle right, toggle right, toggle left. Moreover, in this case, toggling the actions does not reduce action space.

4.2 Factorized Q-values, multi-action agent

4.2.1 Multi-action factorized Q-values policy training

Training and evaluation traces are presented in Figures 3g and 3h. We can see in the training traces that the agent seems to learn properly and faster than the toggle agent. However, we observe a kind of maximum value that is reached by the evaluation trace around epoch 500 (evaluation round 10).

We can interpret the policy from the episode in Figure 7. The agent is always adding new hospital beds and vaccinating. When the number of infected people becomes too high, it seems to confine. And isolation is almost always implemented as soon as there is no confinement, and cancelled when the agent confines. **This policy seems very relevant and realistic.**

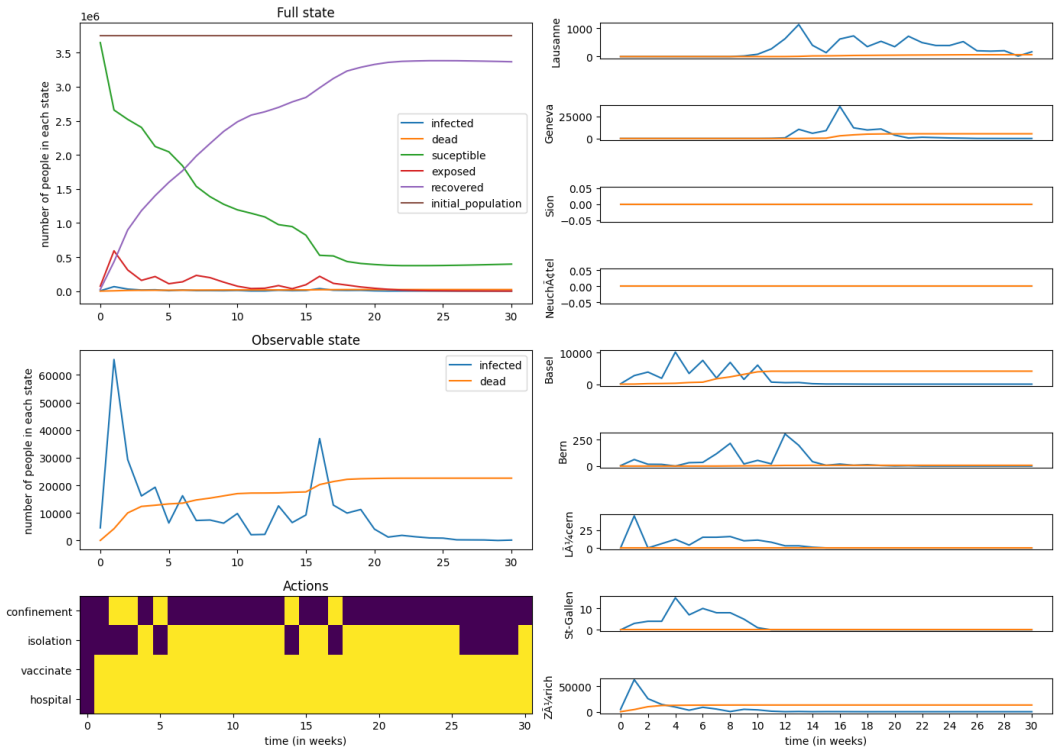


Figure 7: Implement factorized Q-values agent.

4.2.2 Multi-action factorized Q-values policy evaluation

The evaluation histograms for the factorized agent are shown on Figure 4d. We can notice that it achieves a **higher cumulative reward and less confinement days** compared to the toggle policy. However, twice more people die in average. Since the agents are trained using reward, from a reinforcement learning point of view, we can say that the factorized agent performs better than the toggle agent.

4.2.3 Factorized-Q-values, what assumption does it make?

It makes the following assumptions on the action space:

- **All combinations** of decisions are possible.
- The actions are **independent**. For example, if the best action is to confine only, the second best action cannot be to vaccinate only.

Starting from this observation, we can imagine an action space for which factorizing Q-values would not be suitable. For instance, if we added *establish a curfew* to our action space, we would not be able to choose it while still deciding to *confine* since the resulting rules would be conflicting. Moreover, the best action could be to *confine* and the second best one to *establish a curfew*, so these actions are dependent. As a consequence, we could not factorize Q-values.

5 Wrapping up

5.1 Comparing the training behaviors

The training curves of the different agents are presented in Figure 3. We first observe that for a binary agent, the training and evaluation traces abruptly increase during the first 200 episodes, and then stay quite constant. However, when we use a decreasing exploration rate, the training trace seems to keep slowly decreasing.

The toggle agent has a much more linear behaviour: both training and evaluation traces increase with a constant slope.

Finally, the factorized agent seems to learn very quickly at the beginning, to reach a kind of optimal state, and then to generate a quite constant cumulative reward over episodes.

According to these plots, the binary agent with decreasing exploration rate seems to be the most performing one because it learns fast and reaches high cumulative rewards.

5.2 Comparing policies

See Table 1 for comparison of agents performances.

Agent	Confined days	Isolation days	Vaccination days	Additional hospital beds days	total deaths	cumulative rewards
Russo	100	-	-	-	54321	-39
Binary	151	-	-	-	**3002**	63
Toggle	112	**10**	**117**	**69**	11615	57
Factorized	**32**	140	210	210	21628	**77**

Table 1: Comparison of agents over 50 episodes.

If we exclude Russo's policy which is the less performing one according to all criteria except the number of confined days, we notice that the toggle agent is the best one regarding isolation days, vaccination days and additional hospital beds days but has a relatively low cumulative reward. Besides, the best performing agent regarding cumulative reward is the factorized agent, which has the higher number of deaths if we exclude Russo's agent. So we can conclude that cumulative reward only gives back the costs we defined for each action or observation, but **does not guarantee that a policy is good or bad in general.** Nevertheless, from a theoretical reinforcement learning point of view that is based on the reward we aimed at optimizing, the Factorized agent seems to be the best one.

5.3 Interpretability: Q-values

Evolution of Q-values during an episode using our Binary agent is provided in Figure 8. The equivalent plot for the Factorized agent is presented in Figure 9. In the DQN case, the Q-value for confining seems to be almost always higher than the other one, and the few weeks when confinement is not chosen are characterized by **very close Q-values**, and that appears to be hard to interpret.

This observation is also true for the confinement pair of actions in the Factorized case. The chaotic period when confinement is often chosen and then lifted is characterized by very close Q-values. However, for other actions, such as vaccinate, the Q-value of True is clearly higher than the one of False during the whole episode. Consequently, **the Factorized policy seems to be more interpretable than the Binary DQN policy**.

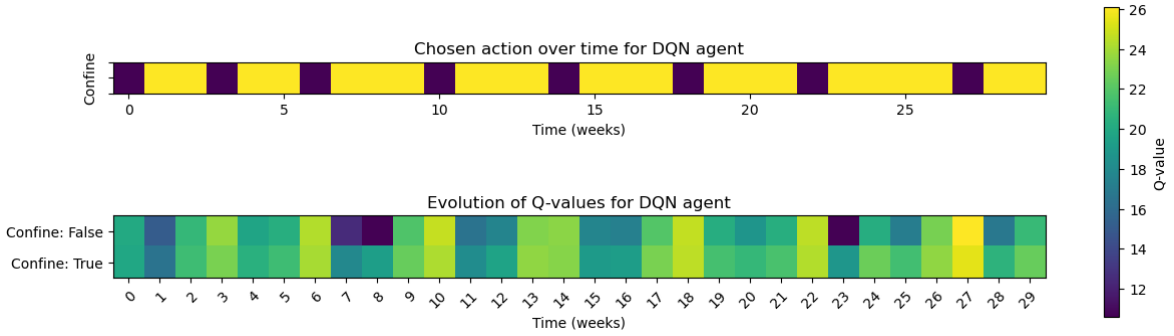


Figure 8: Q-values of Binary agent over time.

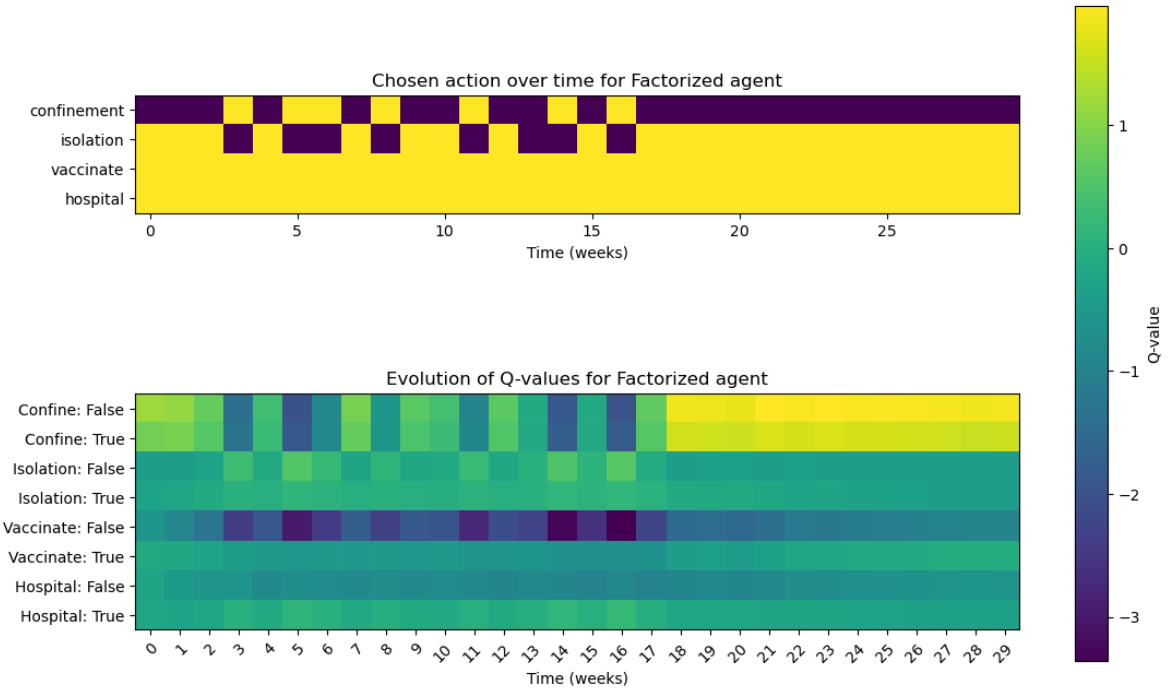


Figure 9: Q-values of Factorized agent over time.

5.4 Is cumulative reward an increasing function of the number of actions?

We can observe in Figure 1 that our Binary agent achieves a higher cumulative reward than the Toggle agent while it has only 2 actions (compared to 5). So cumulative reward does not seem to be an increasing function of the number of actions.

This previous observation comes from experimentation. However, it might be possible that it results from the fact that the Binary agent learns faster and easier than the Toggle agent, so **given the trainings we have done**, this observation is true.

In general, if we assume that two agent A and B have been trained in the exact same environment with a different number of actions (A has strictly less actions than B and **actions of A are included in actions of B**), we claim that B achieves a higher cumulative reward than A **in expectation** and **in the fully greedy case**. Indeed, let's consider a step t of an episode and, at first, fully greedy agents. Given state s_t, agent A would choose the action $a^\star$ that maximizes $Q_A(s_t, a)$ which is the expected reward of choosing action a given state s_t. Since B is fully trained, and can choose any action that A can choose, we know that $a^\star$ is in B's action space. So the action $b^\star$ that B chooses verifies $Q_B(s_t, b^\star) \geq Q_B(s_t, a^\star)$. Moreover, since the environment is exactly the same, we have $Q_B(s_t, a^\star) \geq Q_A(s_t, a^\star)$, because B acts *at least* as well as A for each future step. So we have:

$$Q_B(s_t, b^\star) \geq Q_A(s_t, a^\star)$$

Since this is true for every step t, we can conclude in this case that **B has a higher expected cumulative reward than A**.

However, if we add exploration using a ϵ-greedy policy, it seems not to be true! Indeed, let's consider A and B fully trained agents, with B being able to choose any action of A as well as another action a_B which always lead to a terrible reward (as small as necessary). Then the greedy part of the policy will always choose the same action as A, but the random part will choose a_B with a certain probability p. As soon as $p \neq 0$ and if we imagine that the reward we obtain when acting with a_B is always strictly less than other possible rewards, we get that **the expected cumulative reward for B is strictly smaller than the one for A**. So the discussed claim is not true in general.