Assignment 4

for **Deep Reinforcement Learning WS2024**

by
Kamil Bannasch
405231

Fynn Castor
540055

Eik Weißhaar
507068

the **8. Dezember 2024**

1 Excercise 1.

1.1 a)

- *Visualization: To understand the resulting policy, plot the final estimated action-values $Q(s, a)$ of each grid position as a heat-map. To acquire knowledge about the learning progress over time and the convergence behaviour, plot the total reward per episode. Evaluate the role of importance sampling by comparing the learning performance with and without its use.*

The Output was generated over 1000 Episodes of a length of 50 steps, with policy improvement and evaluation after each step. For the behaviour policy a ε-soft policy with $\varepsilon = 0.4$ was chosen. Furthermore $\gamma = 0.9$ was used. In the case of non-importance sampling we used standard first-visit Monte-Carlo for an ε-soft with the same parameters as the behaviour policy.

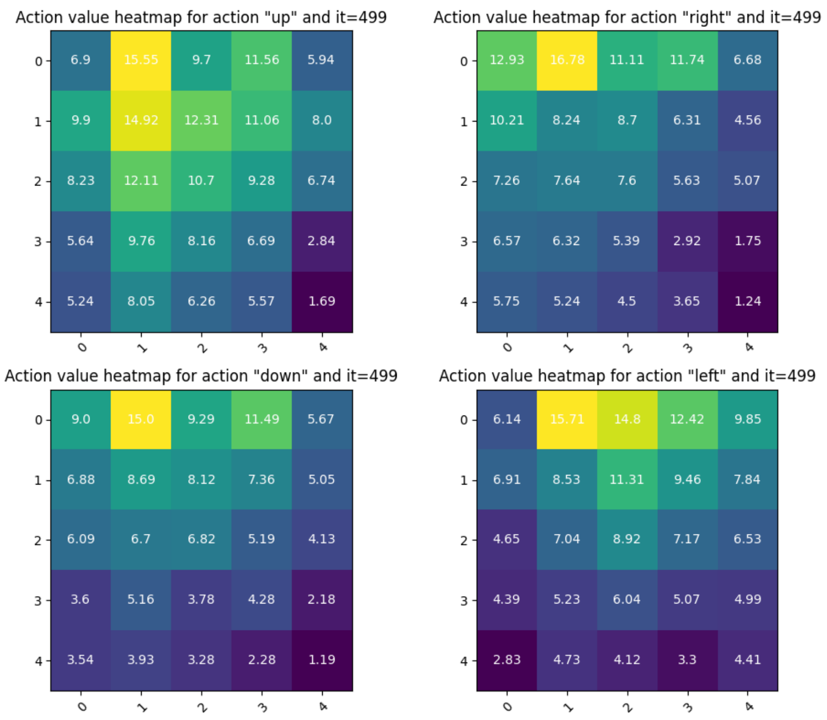
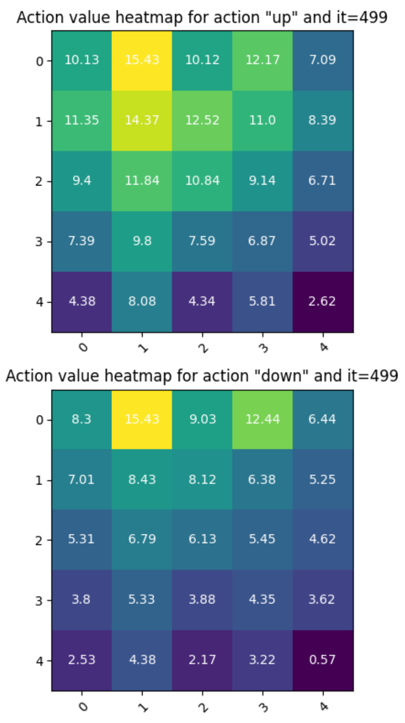


Abbildung 1: Visualization of action-values with importance sampling



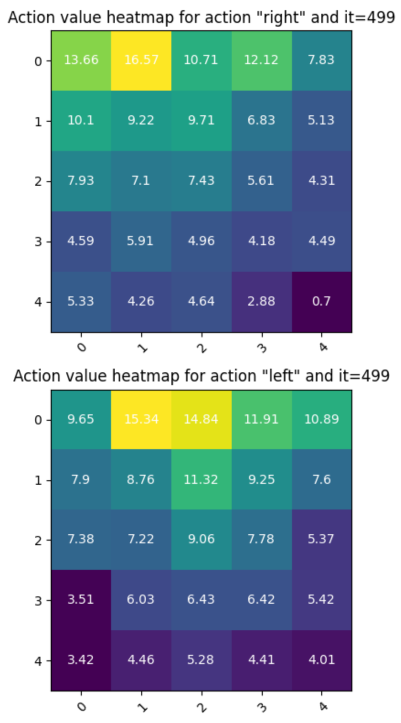
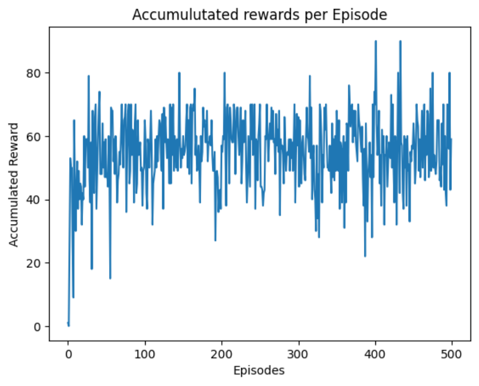
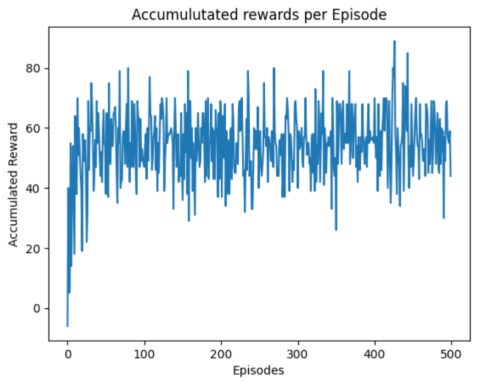


Abbildung 2: Visualization of action-values without importance sampling



(a) With importance sampling



(b) Without importance sampling

Abbildung 3: Total rewards per Episode

- *Reflection: How does importance sampling affect the sample size and resulting value function in terms of convergence speed and accuracy. Consider scenarios where using importance sampling might have clear advantages or disadvantages.*

Importance sampling allows for unbiased estimation of a deterministic (greedy) policy, while still reaping some of the benefits of explorative policy, e.g. ε-soft during the training. With a more explorative behavior policy than the target policy, one can get easier information over so far sparsely visited states, while still being able to use them to update an exploiative policy. Unfortunately, the benefits here are not really visible as shown by the quite similar heatmaps and reward plots of the generated data. In practice importance sampling might also be practicle to reuse already generated samples in order to improve a current policy, lessening the overall computation cost and decreasing the needed sample generation. If the target and behavior policies are too different, though, this might lead to some skewed reweighting decreasing usefulness.

1.2 b)

- *Visualization: Similar to the previous task, plot the progression of your learned policy by visualising the reward over sampled episodes, and by visualising the final action-values Q(s, a).*

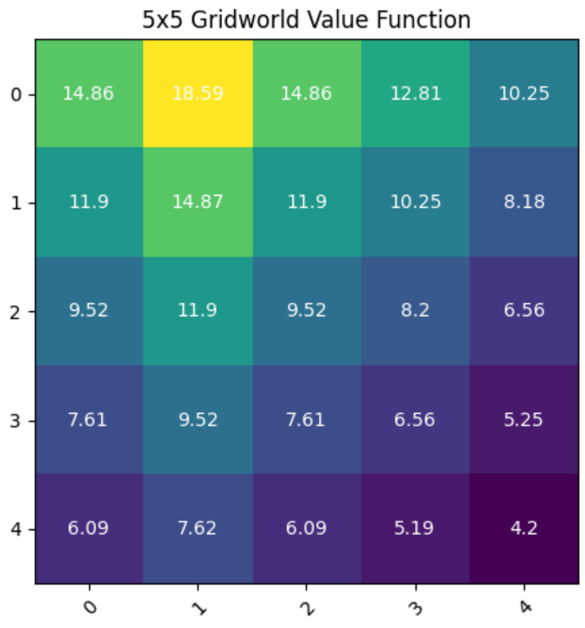


Abbildung 4: State Values (n=1)

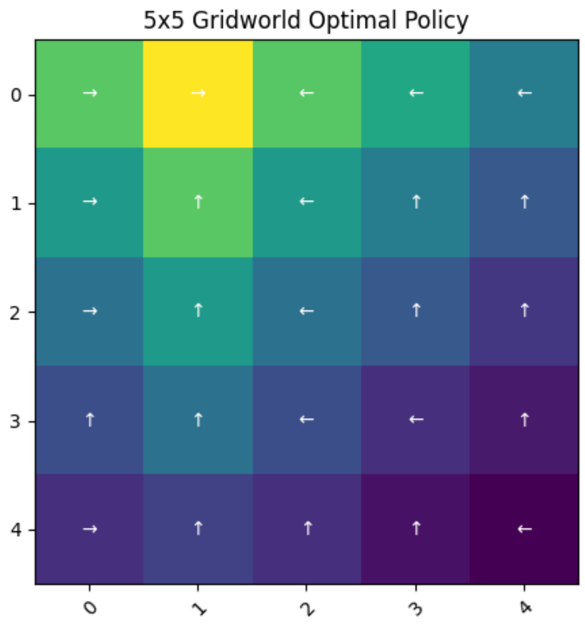


Abbildung 5: Optimal Policy (n=1)

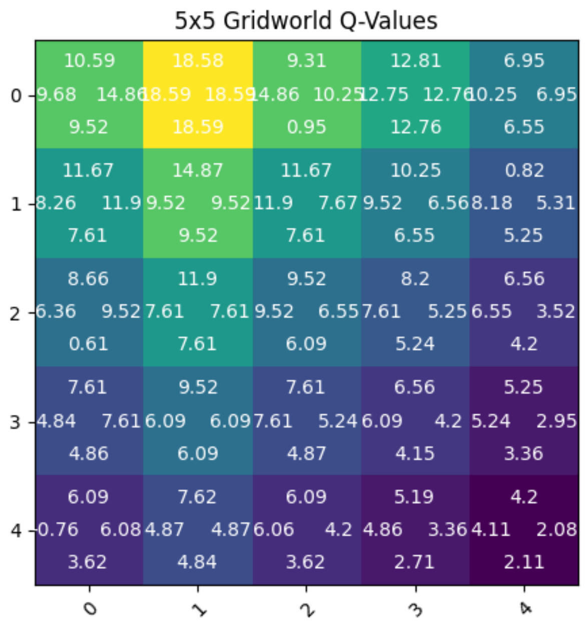


Abbildung 6: Calculated Q-Values (n=1)

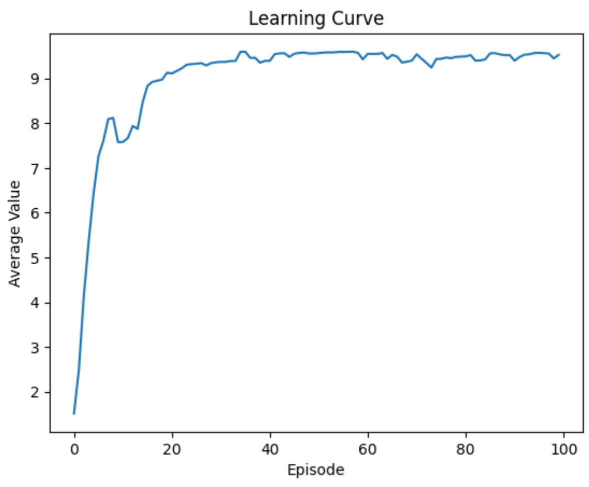


Abbildung 7: Learning Curve depicting the average state value in each episode, which converges after around 30 episodes.

This output was generated over 100 episodes using values of 0.2 for exploration rate, 0.8 for gamma and an alpha of 0.9.

- *Experiment: Test different n-step predictions in TD Learning (from 1-step predictions in TD(0) to longer n-step returns) and compare the results between them by plotting their learning curves in one diagram. Discuss the convergence speed and stability of the Q(s, a) values for different step size parameter n.*

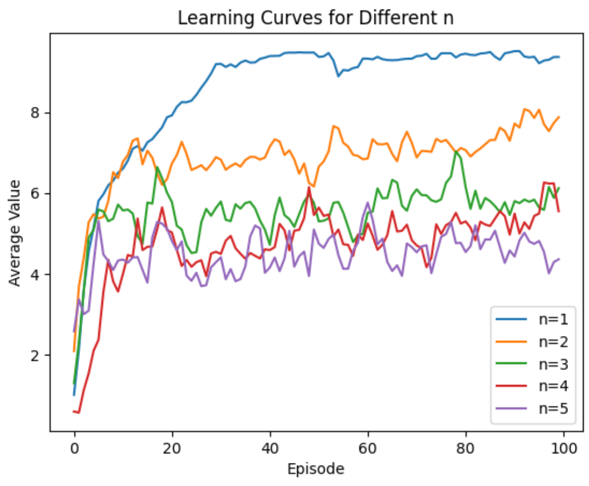


Abbildung 8: Learning curves for different n. These show the average state value and are used to display convergence.

With a larger step size, the average reward decreases. This is because of the added variance caused by including more estimations for the value calculations.

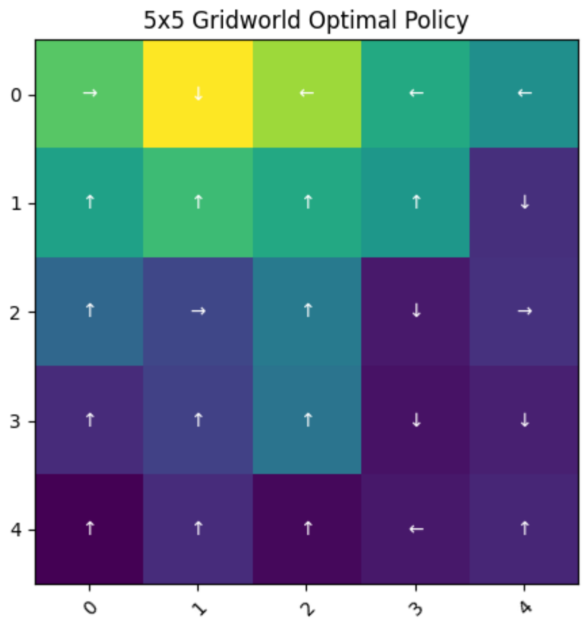


Abbildung 9: Calculated optimal policy with n=5

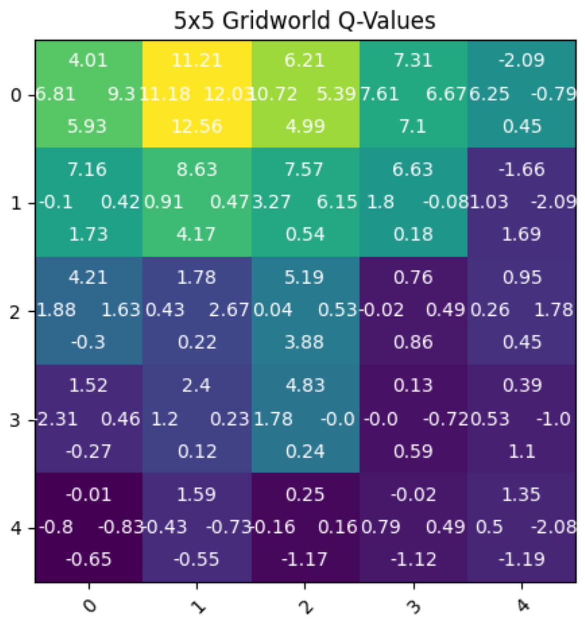


Abbildung 10: Q-Values (n=5)

When comparing the output of n=1 to n=5, it is obvious that both convergence speed and stability decrease with larger step sizes. Additionally, even after 100 episodes, the optimal policy for many states still has not been found.

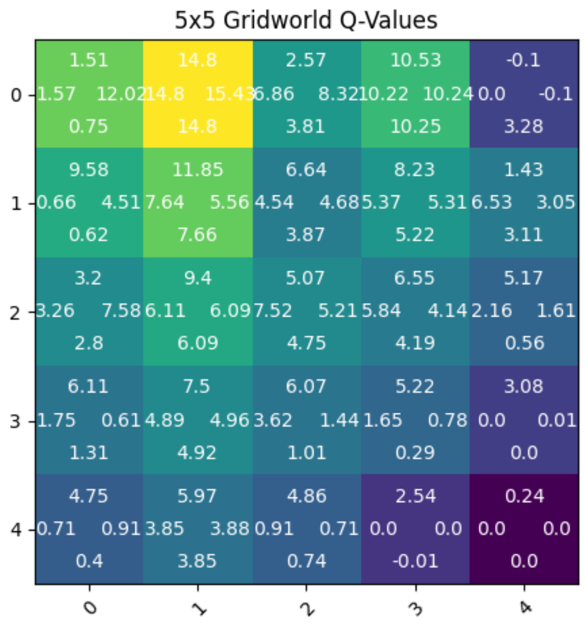


Abbildung 11: Q-values after termination

1.3 c)

- *Visualization: As with the other algorithms, visualize the converged estimates of the actionvalues by plotting Q(s, a) for each state as a heat map. Additionally, track the cumulative reward per episode to capture the learning dynamics over time, and combine these plots with those from previous tasks (SARSA and MC approach) to compare performance.*

This output was generated over 1000 episodes using values of 0.1 for exploration rate, 0.8 for gamma and an alpha of 0.1.

- *Reflection: (1) Why is Q-learning categorized as an off-policy algorithm. Reflect on its ability to learn optimal policies independently of the current exploration strategy, and highlight the advantages and potential challenges when updating through maximal future returns. (2) Compare Q-learning to other presented methods in terms of learning stability and data efficiency, particularly emphasizing scenarios where Q-learning may offer better performance due to its off-policy nature.*

1. Q-Learning is off-policy, because it separates the target and behaviour policies. This allows for calculating the values of the target policy while also allowing exploration through the behaviour policy (in this case epsilon-greedy). As long as each action is taken infinitely many times by the behaviour policy, it will converge on the optimal policy. The clear advantage is that, given any starting configuration, as long as the above statement is true, the optimal policy will be found. However, if the underlying model changes, this approach will take a long time to adapt, making it better for (mostly) static situations.

2. Compared to TD-Learning and MC, Q-Learning features a lower data efficiency, as it can take a long time to converge on the optimal policy. Its stability depends on the

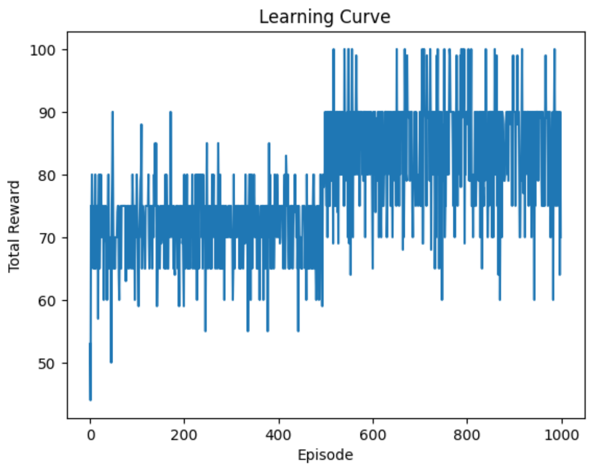


Abbildung 12: Total reward of each episode. The halfway shift is caused by the fact that upon initialisation, the right path is favored until the left one is found by exploration.

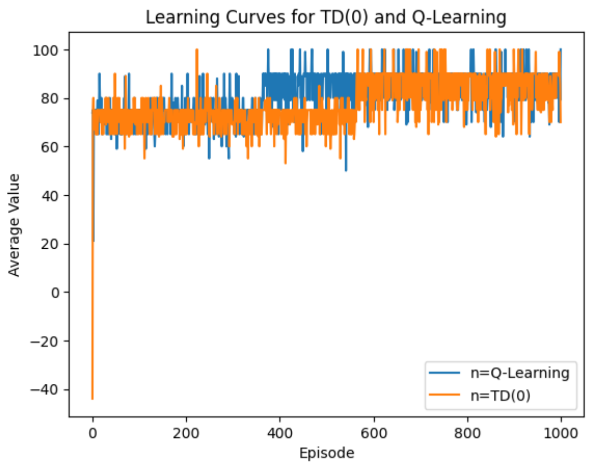


Abbildung 13: The different rewards of TD(0) and Q-Learning. Q-Learning performs a little bit better here, as it has a higher average reward.

used exploration strategy, with a high exploration rate leading to faster convergence, but also suboptimal decision making. In a dynamic environment, it is less stable than SARSA, because it estimates the reward of future actions. It is best used in scenarios with the goal of determining the optimal policy while also ensuring sufficient exploration in case there are external changes.

2 Excercise 2.

2.1 a)

Visualisation: To compare the behaviours of SARSA and Q-learning, visualize the paths each algorithm generates in the Cliff Walking environment. This will highlight the decision-making processes and navigation strategies of the final policies. Additionally, plot the cumulative rewards for each episode, across at least 500 episodes, to evaluate the learning performance of both algorithms.

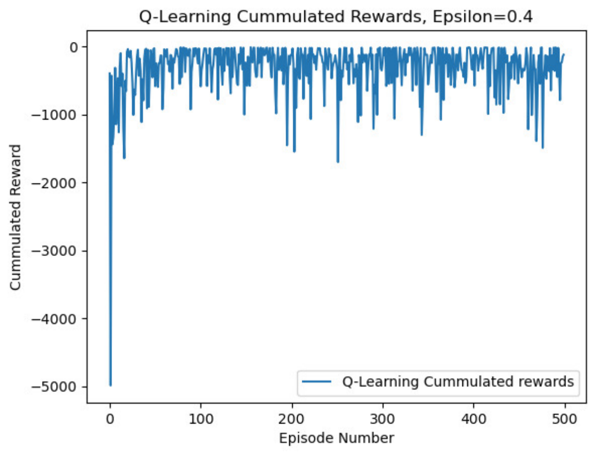


Abbildung 14: Cummulated Rewards after 500 episodes with Q-Learning with epsilon 0.4

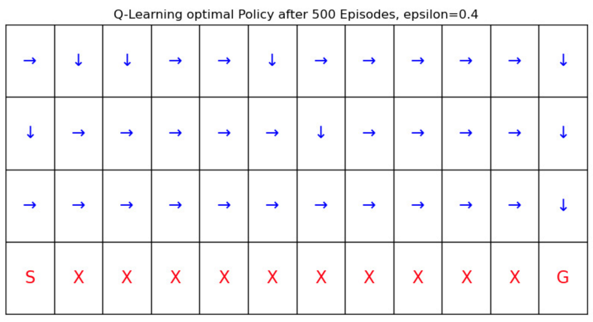


Abbildung 15: Optimal Policy after 500 episodes with Q-Learning with epsilon 0.4

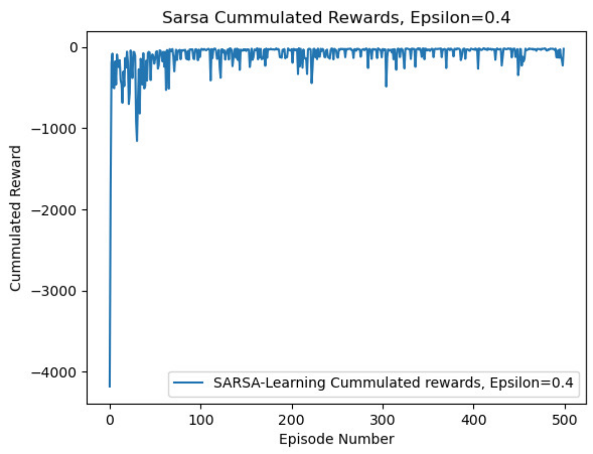


Abbildung 16: Cummulated Rewards after 500 episodes with SARSA with epsilon 0.4

Reflection: Consider following points
(1) In the context of reinforcement learning, what are the advantages and disadvantages of on-policy versus off-policy methods? Can you identify two scenarios where one approach might outperform the other?
(2) Why does SARSA tend to choose safer paths compared to Q-learning, which seeks the optimal path? During exploration, could Q-learning's greedy strategy lead to falling off the cliff?
(3) Why might Q-learning accumulate lower total rewards per episode compared to SARSA?
(4) Suppose action selection in Q-learning follows a purely greedy strategy. Discuss whether, under these conditions, Q-learning becomes identical to SARSA? Will they make exactly the same action selections and updates to their value functions?

(1) On-Policy updates the policy that is taken, while off-policy updates the target policy. Q-Learning updates the policy, even when not taking it, while SARSA takes it's policy and updates it when going through it.
(2) Because SARSA updates it's policy also when exploring and therefore a position close to the cliff would get big negative rewards. Q-Learning could fall off the cliff when close to it and when exploration is chosen. Therefore Q-Learning is "riskier"for this example.
(3) Because SARSAs sampling method might result in more steps taken, then the Q-Learning and therefore learns during an episode then the Q-Learnings
(4)

2.2 b)

Visualisation: Visualize the paths of the final on- and off-policies (for all epsilon values) to compare differences in behaviour. Additionally, plot the cumulative rewards for each episode to see how exploration affects the learning and performance over time.

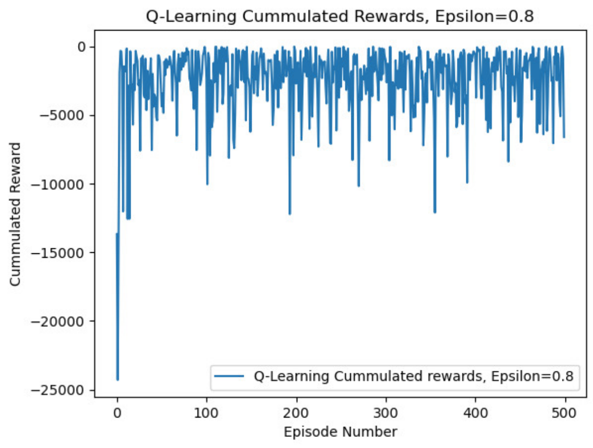


Abbildung 17: Cummulated Rewards after 500 episodes with Q-Learning with epsilon 0.8

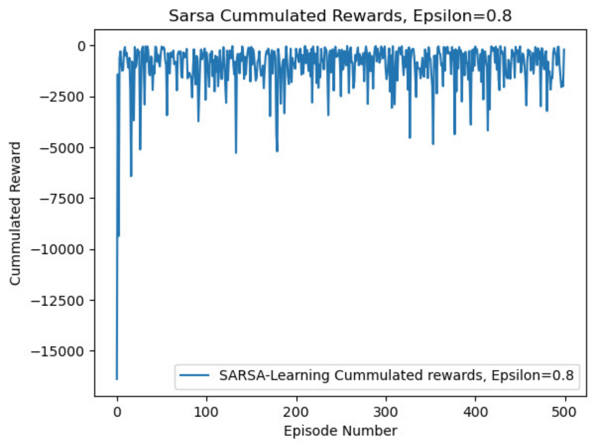


Abbildung 18: Cummulated Rewards after 500 episodes with SARSA with epsilon 0.8

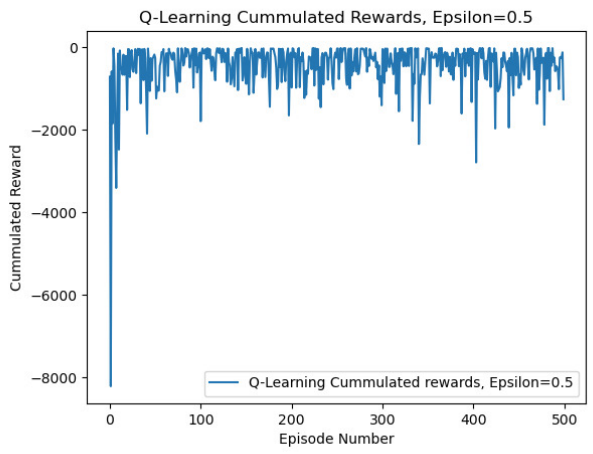
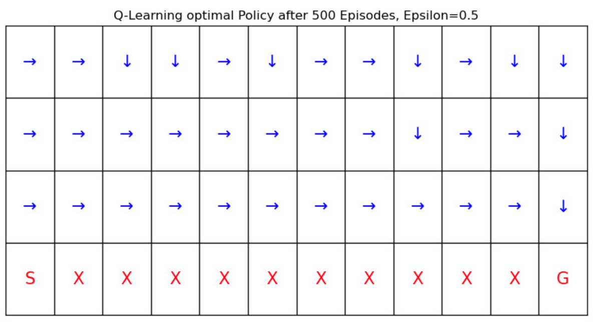
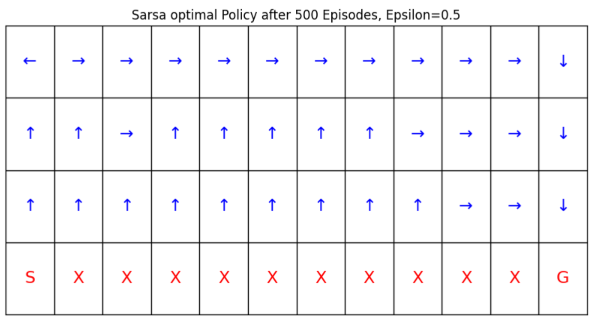


Abbildung 19: Cummulated Rewards after 500 episodes with Q-Learning with epsilon 0.5


Q-Learning optimal Policy after 500 Episodes, Epsilon=0.5

→	→	↓	↓	→	↓	→	→	↓	→	↓	↓
→	→	→	→	→	→	→	→	↓	→	→	↓
→	→	→	→	→	→	→	→	→	→	→	↓
S	X	X	X	X	X	X	X	X	X	X	G

Abbildung 20: Optimal Policy with Q-Learning with epsilon 0.5


Sarsa optimal Policy after 500 Episodes, Epsilon=0.5

←	→	→	→	→	→	→	→	→	→	→	↓
↑	↑	→	↑	↑	↑	↑	↑	→	→	→	↓
↑	↑	↑	↑	↑	↑	↑	↑	↑	→	→	↓
S	X	X	X	X	X	X	X	X	X	X	G

Abbildung 21: Optimal Policy with SARSA with epsilon 0.5

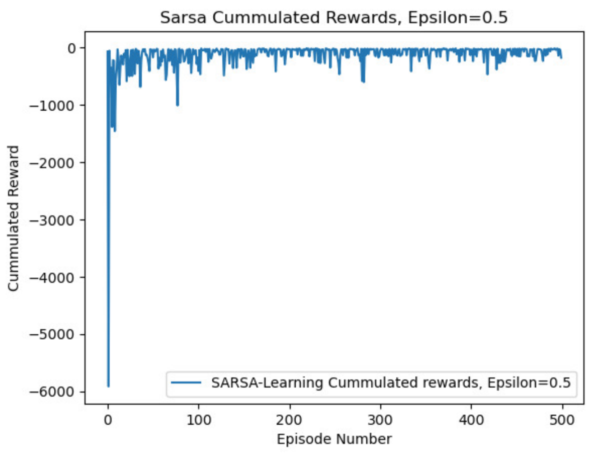


Abbildung 22: Cummulated Rewards after 500 episodes with SARSA with epsilon 0.5

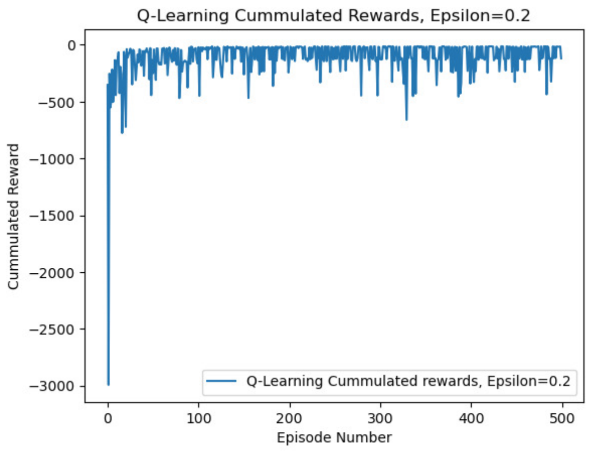
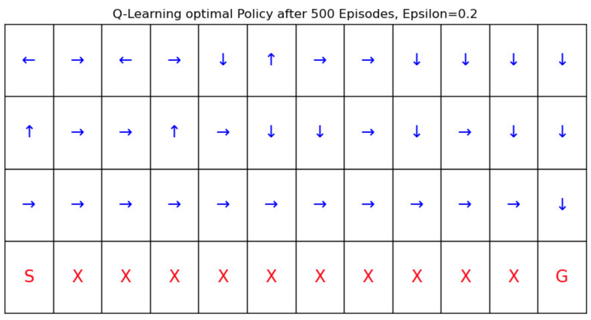


Abbildung 23: Cummulated Rewards after 500 episodes with Q-Learning with epsilon 0.2


Q-Learning optimal Policy after 500 Episodes, Epsilon=0.2

←	→	←	→	↓	↑	→	→	↓	↓	↓	↓
↑	→	→	↑	→	↓	↓	→	↓	→	↓	↓
→	→	→	→	→	→	→	→	→	→	→	↓
S	X	X	X	X	X	X	X	X	X	X	G

Abbildung 24: Optimal policy with Q-Learning with epsilon 0.2

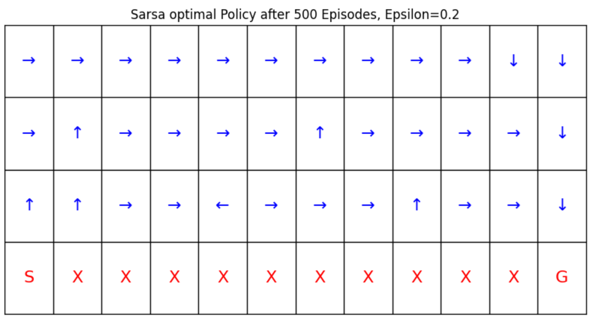


Abbildung 25: Optimal Policy with SARSA with epsilon 0.2

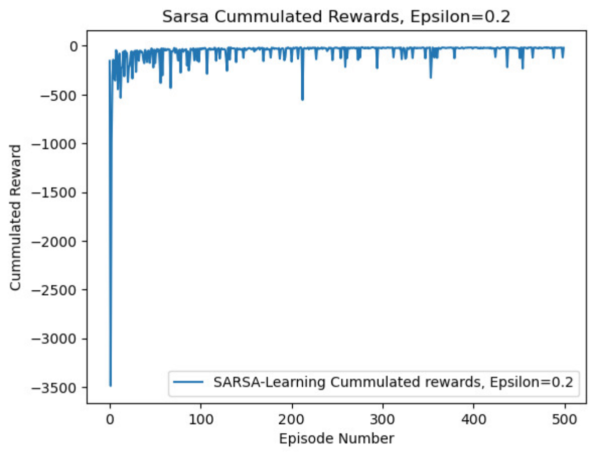


Abbildung 26: Cummulated Rewards after 500 episodes with SARSA with epsilon 0.2

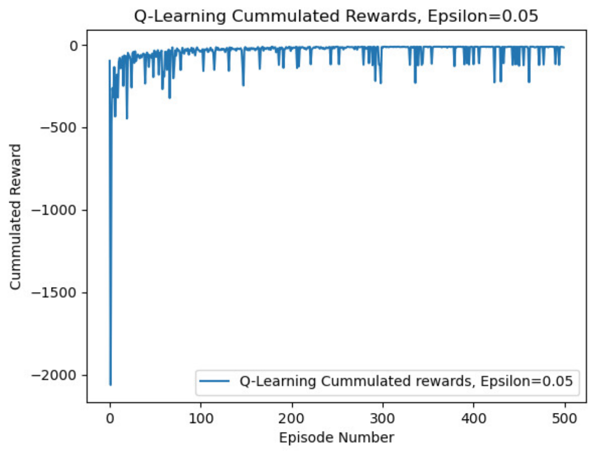


Abbildung 27: Cummulated Rewards after 500 episodes with Q-Learning with epsilon 0.05

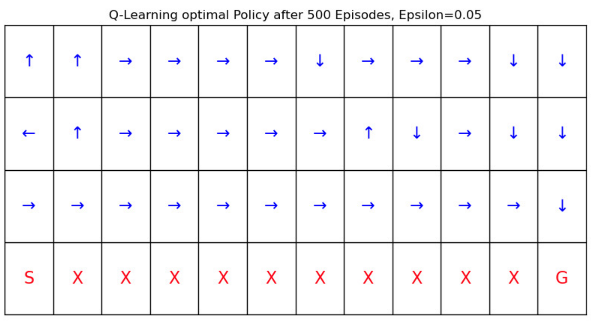


Abbildung 28: Optimal Policy with Q-Learning with epsilon 0.05

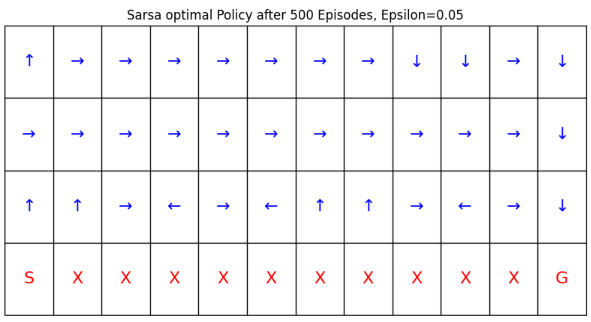


Abbildung 29: Optimal Policy with SARSA with epsilon 0.05

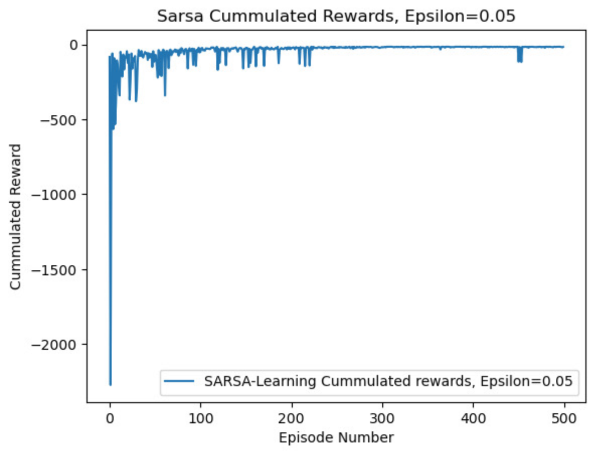


Abbildung 30: Cummulated Rewards after 500 episodes with SARSA with epsilon 0.05

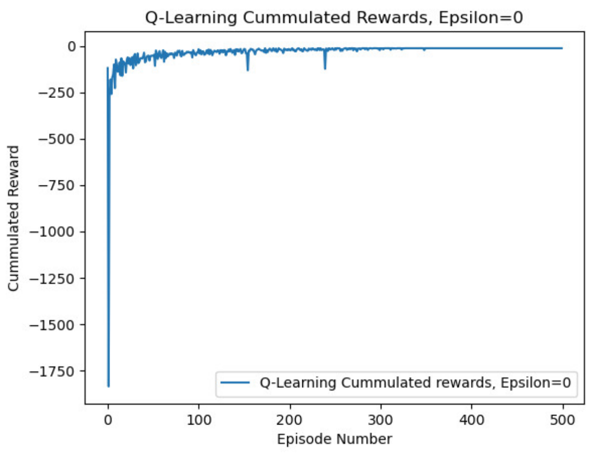
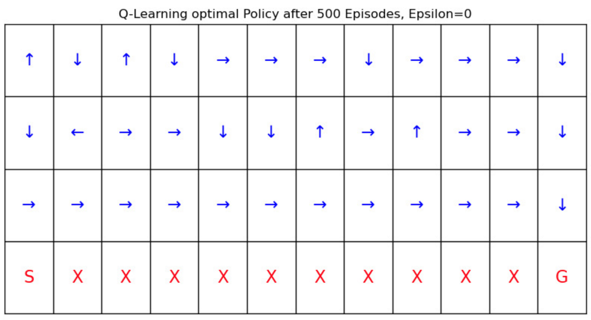


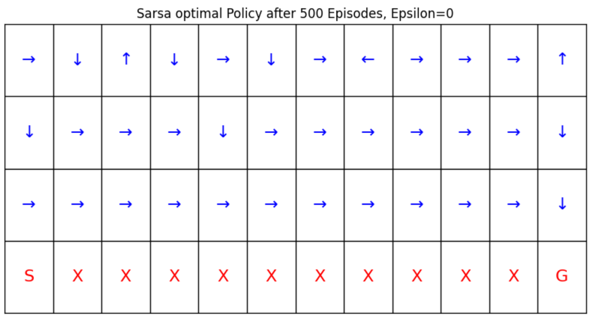
Abbildung 31: Cummulated Rewards after 500 episodes with Q-Learning with epsilon 0



Q-Learning optimal Policy after 500 Episodes, Epsilon=0

↑	↓	↑	↓	→	→	→	↓	→	→	→	↓
↓	←	→	→	↓	↓	↑	→	↑	→	→	↓
→	→	→	→	→	→	→	→	→	→	→	↓
S	X	X	X	X	X	X	X	X	X	X	G

Abbildung 32: Optimal Policy with Q-Learning with epsilon 0



Sarsa optimal Policy after 500 Episodes, Epsilon=0

→	↓	↑	↓	→	↓	→	←	→	→	→	↑
↓	→	→	→	↓	→	→	→	→	→	→	↓
→	→	→	→	→	→	→	→	→	→	→	↓
S	X	X	X	X	X	X	X	X	X	X	G

Abbildung 33: Optimal Policy with SARSA with epsilon 0

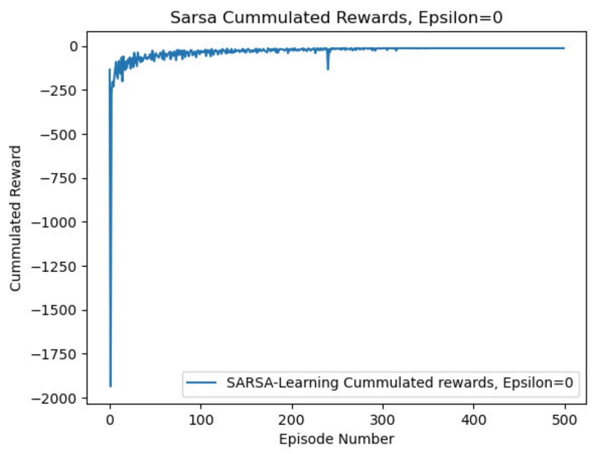


Abbildung 34: Cummulated Rewards after 500 episodes with SARSA with epsilon 0

Reflection: Consider following points
(a) Observe the paths chosen by both SARSA and Q-learning agents with different exploration parameters. Does the epsilon-greedy policy influence the behaviour of SARSA and Q-learning agents? Is there a connection between the amount of exploration and the path taken by the algorithm? Discuss how changes in epsilon affect the agents' decision- making and learning process by examining the final policies and analysing the variance in the sum of rewards during a sampled episode.
(b) Explain why SARSA with decreasing epsilon adjusts the policy more and more towards the optimal path. Compare this with how Q-learning's policy responds to changes in epsilon. How do the algorithms manage risky actions as epsilon decreases? Provide examples from your simulations to support your answer.
(c) What are the potential risks and benefits of completely eliminating exploration? Could a lack of exploration lead to suboptimal policies? Consider how this limitation might affect both SARSA and Q-learning. Suggest a modification or an alternative approach that incorporates inherent exploration capabilities, such as adaptive exploration strategies to maintain a balance between exploration and exploitation.

(a) It definitly effect the the behaviour of SARSA and Q-Learning. A higher exploration rate causes the path next to the cliff to be unsafe, leading to a more cautious optimal path. A lower exploration rate decreases the danger of that path, as the agent is more unlikely to take the down-action.

(b) Favoring the greedy strategy the SARSA agent sees the optimal path more and more with a decreasing epsilon, since exploration is smaller and therefore the path rarely flies off the cliff. The state values are therefore not updated with the negativ reward as often as they would be with a higher exploration rate. Comparing the Q-Learning it seems, that the algorithms both get to their converging paths quicker when the exploration probabilty is on the lower scale.

(c) Eliminating exploration will cause the agent to always take the current optimal path. While it might be a suitable idea for this problem, there are issues with it. If there was a second, longer path with a higher reward, it would not be explored, making the agent miss out on potential rewards. An idea would be to limit exploration to all actions not leading down the cliff in this case. If there was a hidden reward somewhere, it will eventually be discovered without negatively affecting the reward too much.