CS6700 PA1: Q Learning and Sarsa for Windy Gridworld

Kankan Jana (CS21M026), Richa Verma (CS20D020)

March 9, 2022

The problem statement involves solving a windy gridworld environment, with and without stochasticity and with different starting states. We also get to vary the exploration strategy between ϵ-greedy and softmax action selection. The task is to design a tabular RL agent using Q learning and SARSA algorithms to solve the given environment and evaluate its performance across different environment and algorithm configurations. The problem statement also asks us to tune the hyperparameters for both the algorithms across these 16 configurations and use the best ones to plot the final results. These configurations are tabulated below for each algorithm, along with their optimal values which are arrived at after performing hyperparameter tuning. We use the open-source tool wandb.ai to perform this tuning. It uses a Bayes startegy for tuning which explores the hyperparameter region randomly at first, and then goes into regions with high return, gradually.

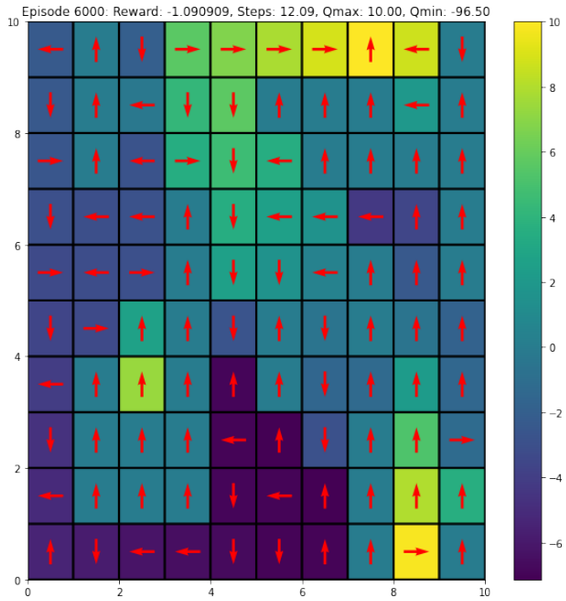
1 Q Learning

We first describe our findings for the Q Learning algorithm in the given environment. We run each experiment for 6000 episodes based on our observations of the convergence behaviour of the algorithm. The below table specifies the optimal values of hyperparameters that are found after performing tuning sweeps for each of the 16 configurations. Table 1 shows the values.

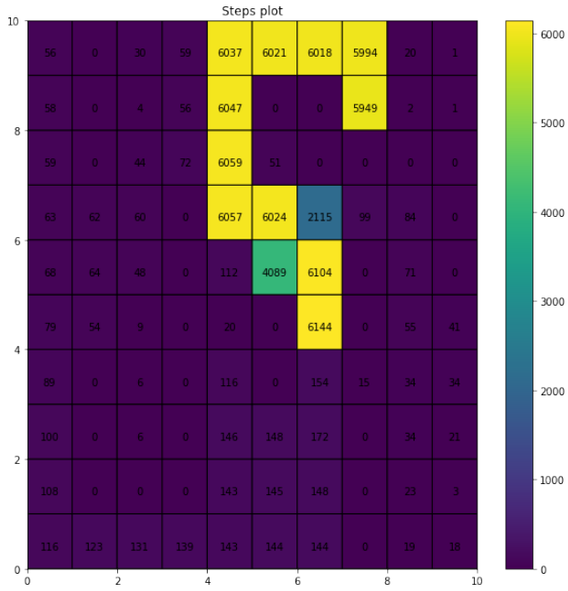
Config	Start	p-val	Wind	Explore	α	β	ϵ	γ	Return
1	(3, 6)	1.0	F	ϵ-greedy	0.2	-	0.01	0.99	-1.09
2	(3, 6)	1.0	F	softmax	0.4	0.2	-	0.9	-1.0
3	(3, 6)	1.0	T	ϵ-greedy	0.3	-	0.01	0.99	-4.3
4	(3, 6)	1.0	T	softmax	0.3	0.5	-	0.99	-5.9
5	(3, 6)	0.7	F	ϵ-greedy	0.3	-	0.01	0.99	-22.4
6	(3, 6)	0.7	F	softmax	0.3	0.1	-	0.99	-19.0
7	(3, 6)	0.7	T	ϵ-greedy	0.2	-	0	1.0	-55.3
8	(3, 6)	0.7	T	softmax	0.2	0.2	-	0.99	-44.3
9	(0, 4)	1.0	F	ϵ-greedy	0.4	-	0.01	0.99	-6.2
10	(0, 4)	1.0	F	softmax	0.2	0.5	-	0.99	-6.0
11	(0, 4)	1.0	T	ϵ-greedy	0.6	-	0.01	0.99	-8.4
12	(0, 4)	1.0	T	softmax	0.6	0.2	-	0.99	-7.6
13	(0, 4)	0.7	F	ϵ-greedy	0.6	-	0.01	1.0	-19.7
14	(0, 4)	0.7	F	softmax	0.4	0.1	-	0.99	-16.7
15	(0, 4)	0.7	T	ϵ-greedy	0.2	-	0.01	0.99	-24.4
16	(0, 4)	0.7	T	softmax	0.2	0.01	-	0.99	-20.1

Table 1: Hyperparamater Tuning Results for Q Learning

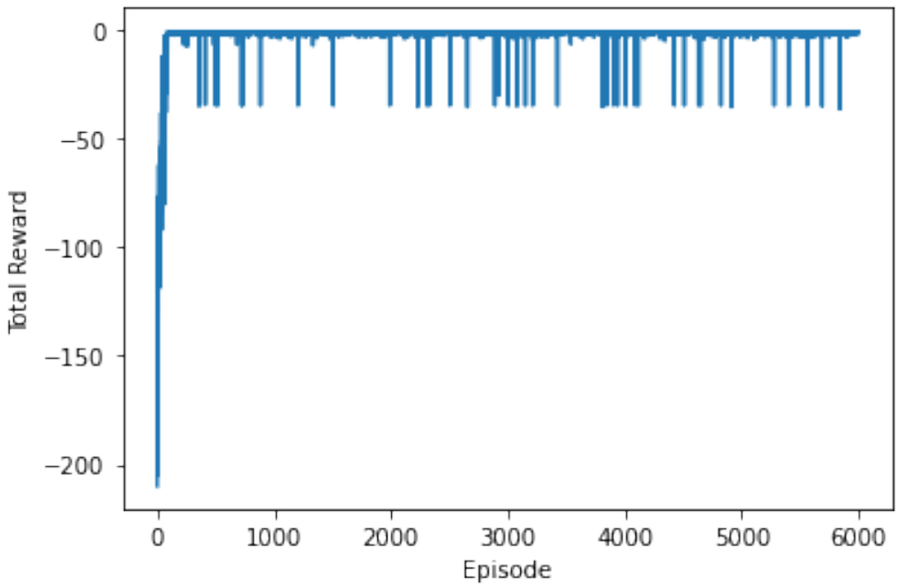
The following figures show the results for each of the configs mentioned in the above table.



(a) Q value heatmap

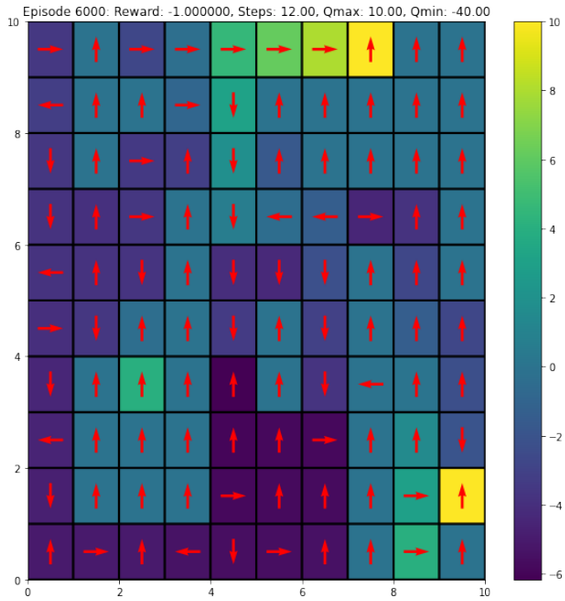


(b) State occupancy heatmap

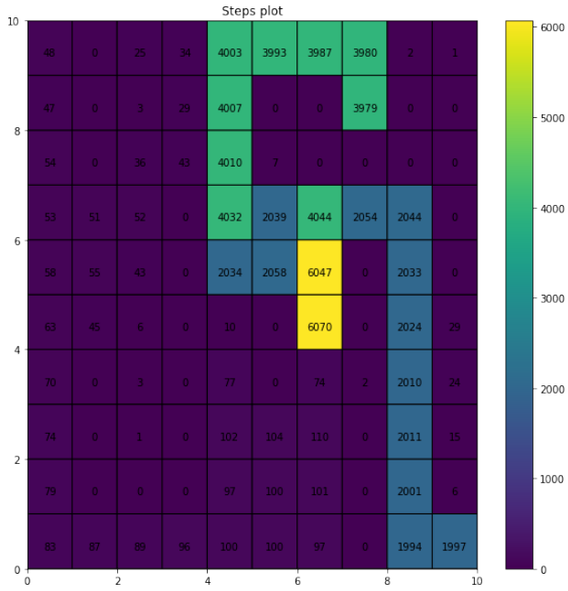


(c) Reward vs. episodes

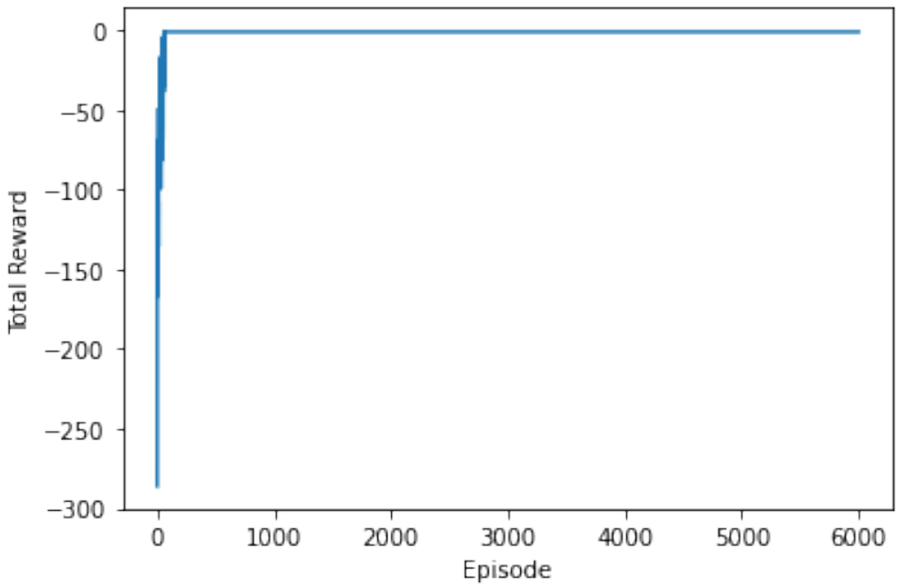
Figure 1: Q Learning for config 1



(a) Q value heatmap

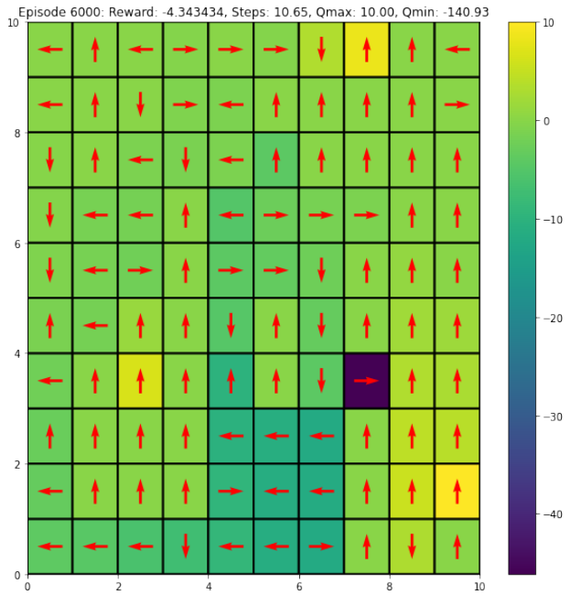


(b) State occupancy heatmap

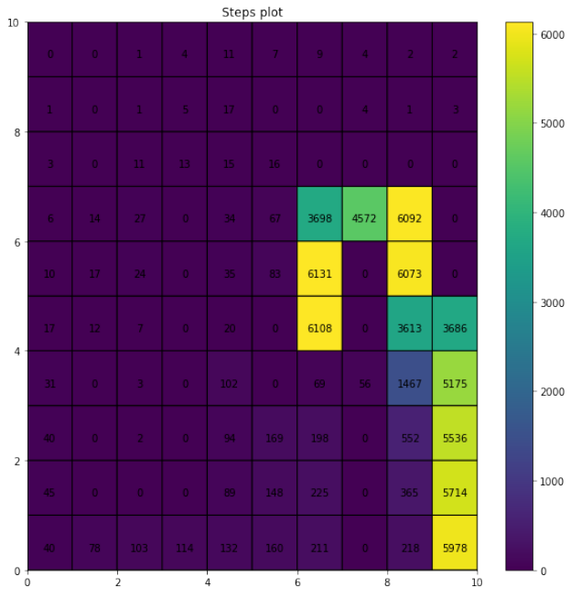


(c) Reward vs. episodes

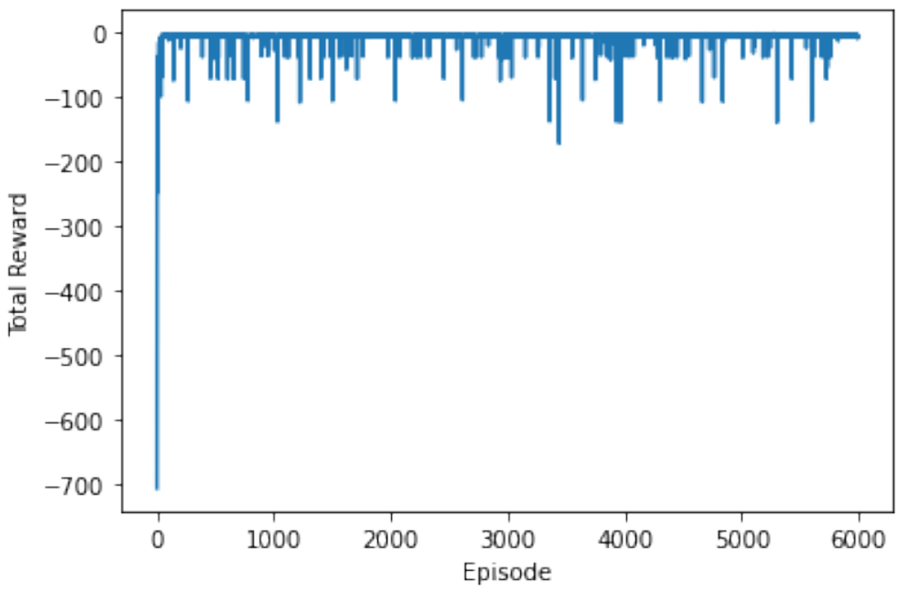
Figure 2: Q Learning for config 2



(a) Q value heatmap

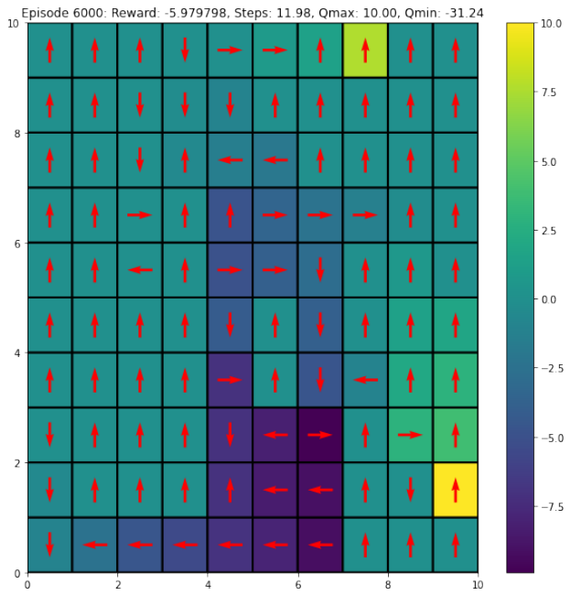


(b) State occupancy heatmap

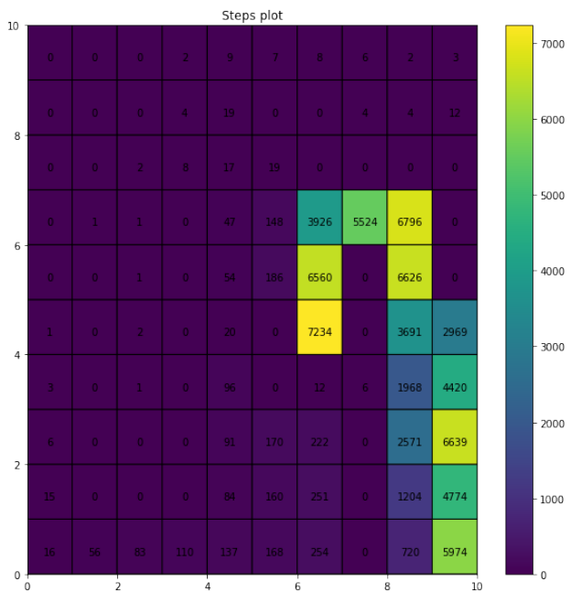


(c) Reward vs. episodes

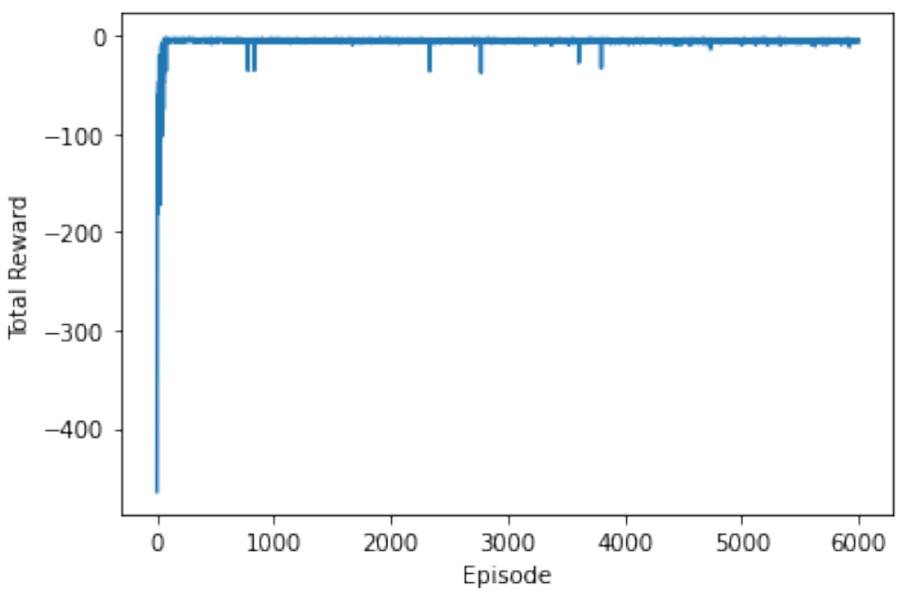
Figure 3: Q Learning for config 3



(a) Q value heatmap

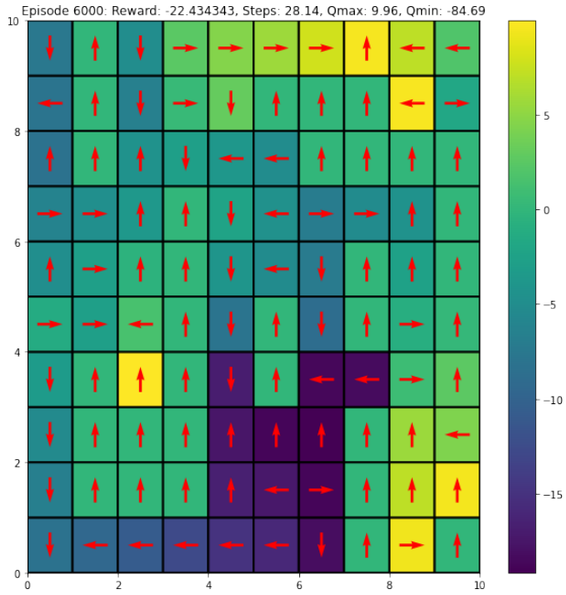


(b) State occupancy heatmap

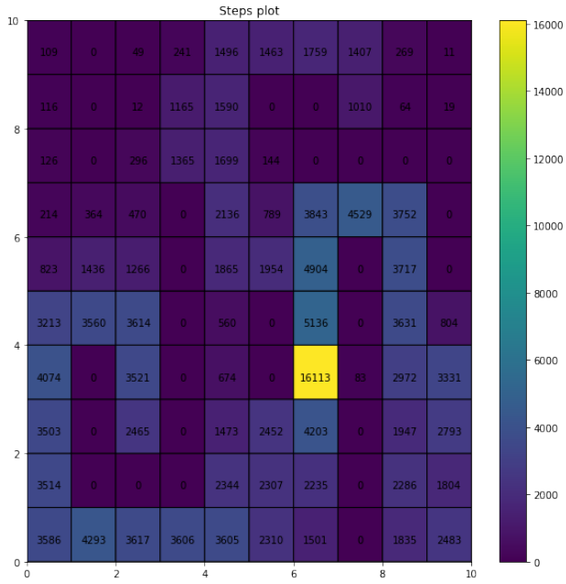


(c) Reward vs. episodes

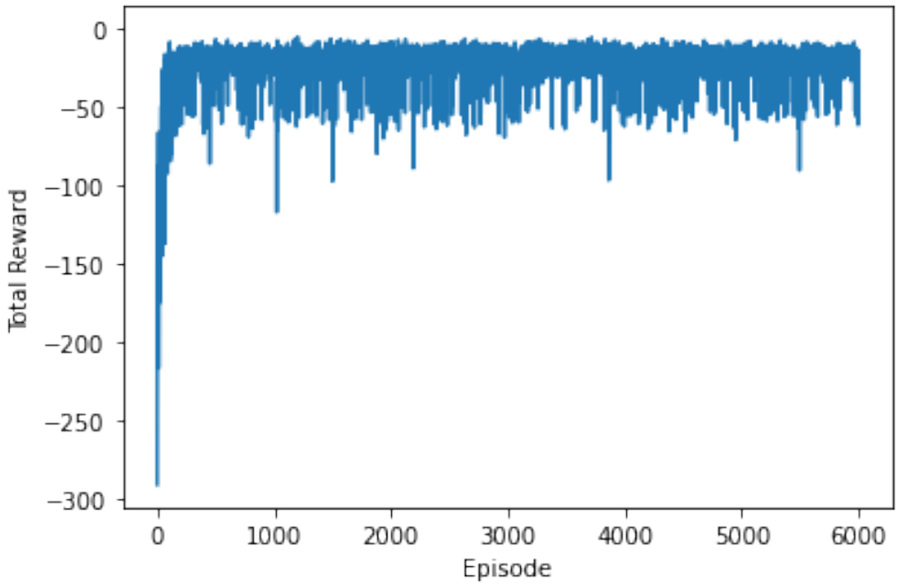
Figure 4: Q Learning for config 4



(a) Q value heatmap

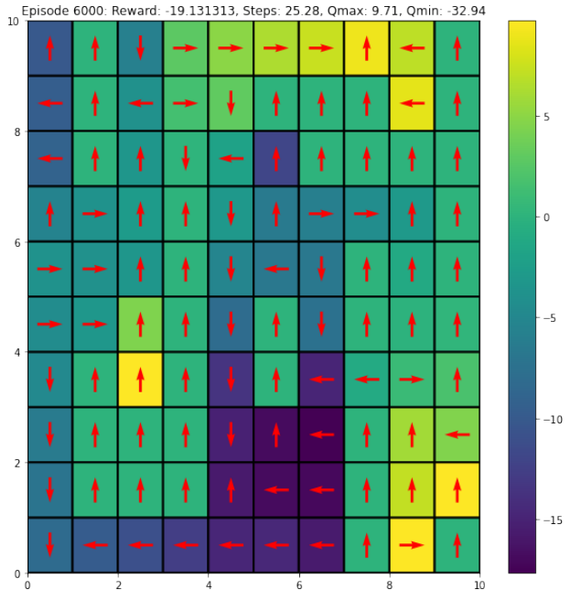


(b) State occupancy heatmap

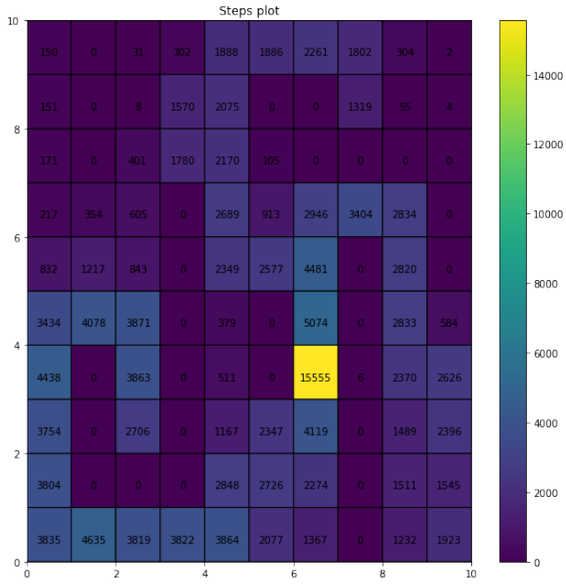


(c) Reward vs. episodes

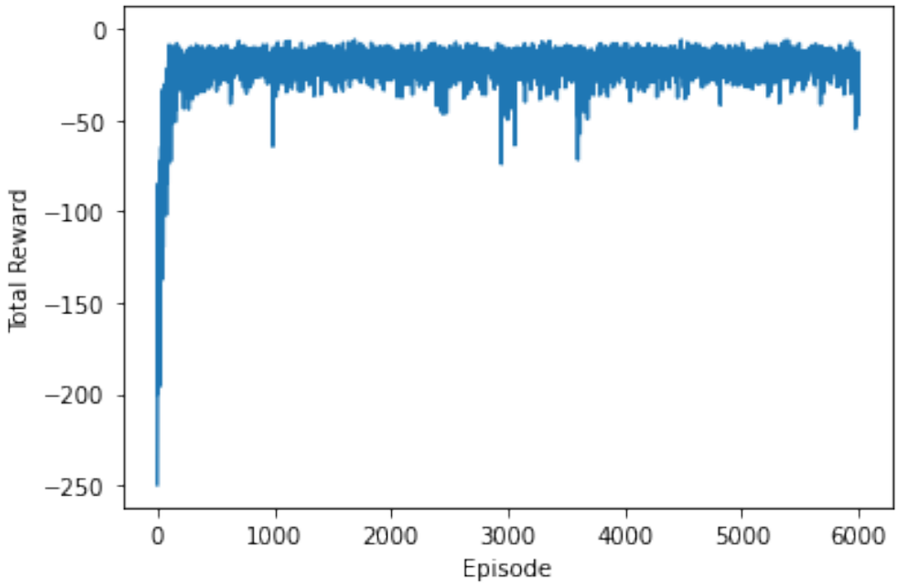
Figure 5: Q Learning for config 5



(a) Q value heatmap

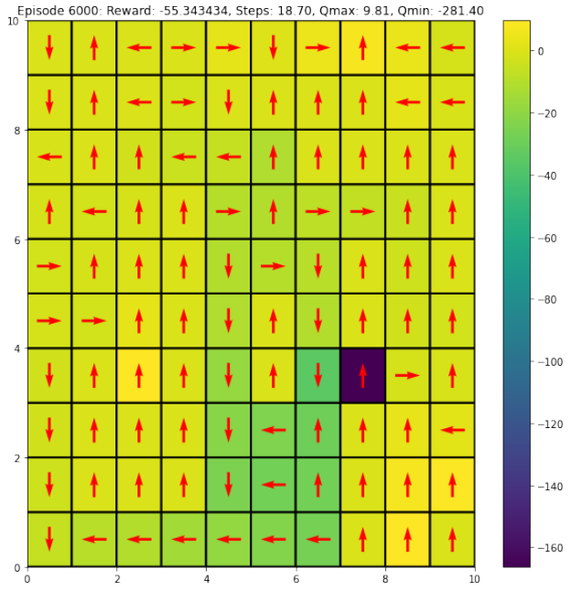


(b) State occupancy heatmap

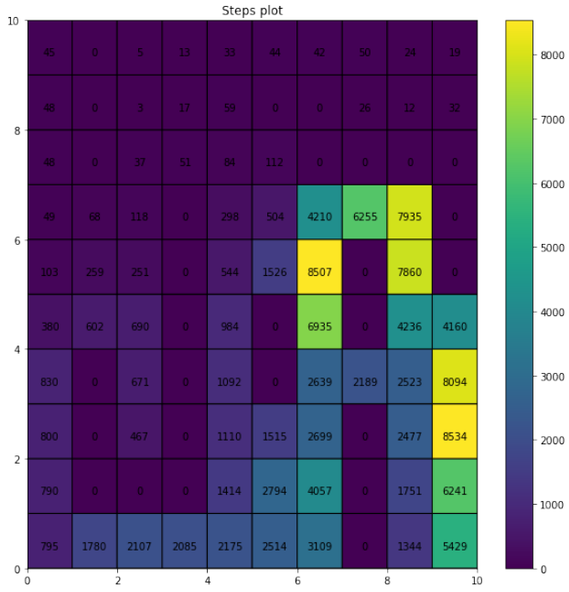


(c) Reward vs. episodes

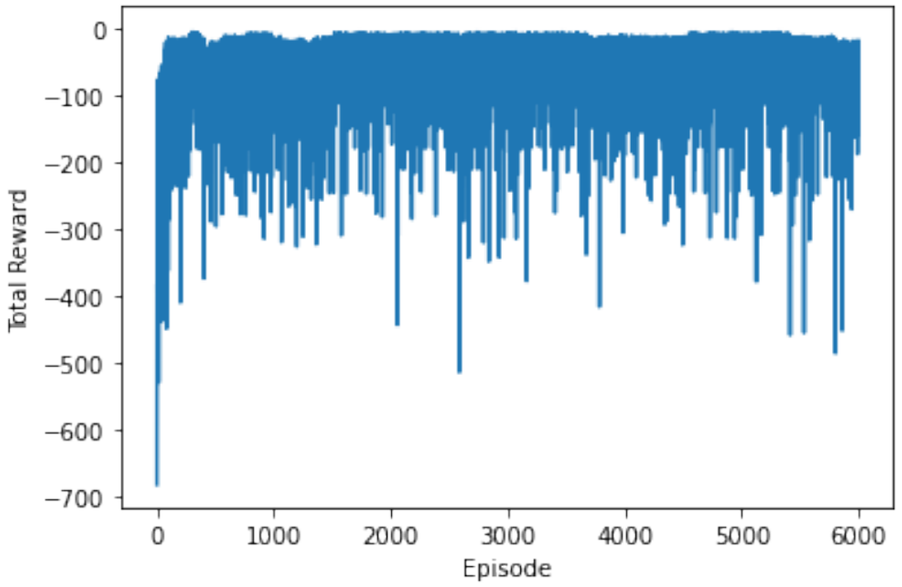
Figure 6: Q Learning for config 6



(a) Q value heatmap

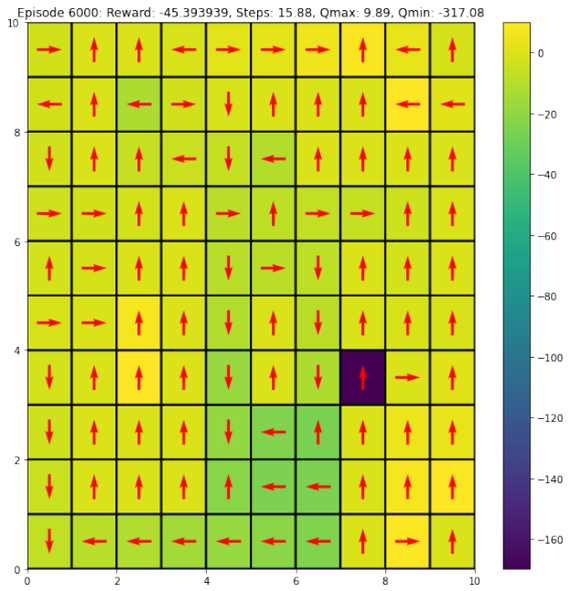


(b) State occupancy heatmap

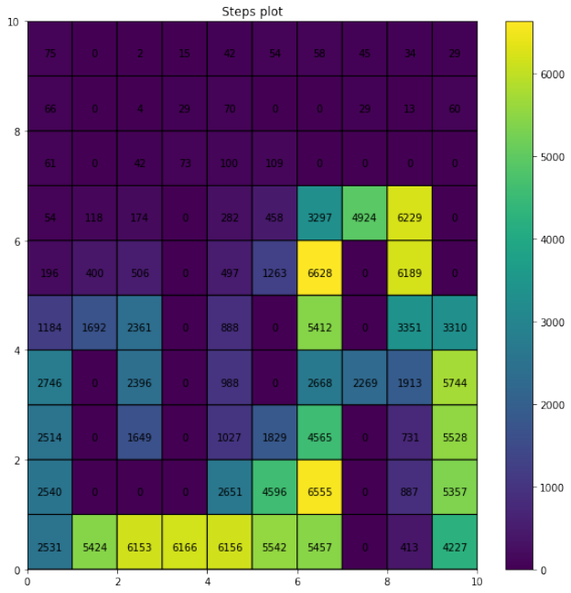


(c) Reward vs. episodes

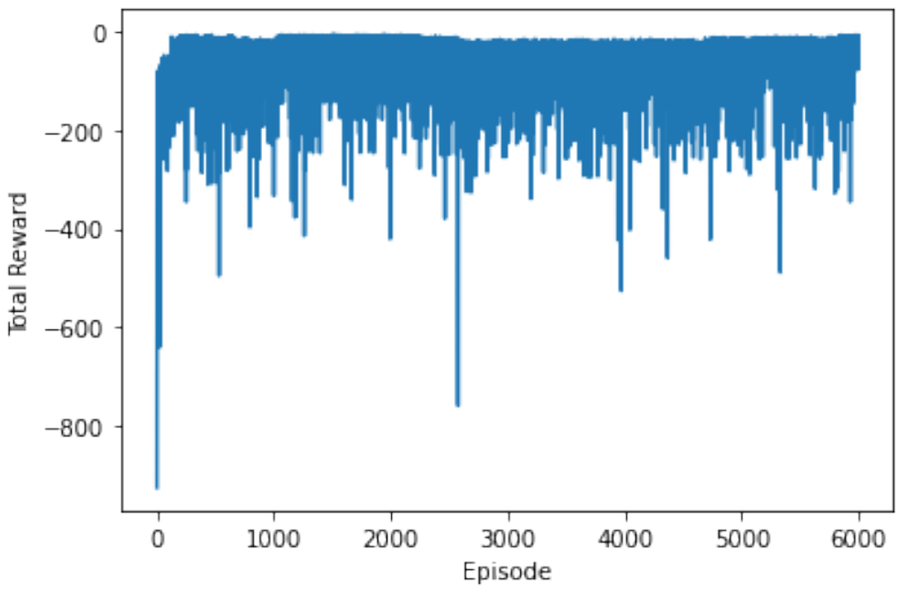
Figure 7: Q Learning for config 7



(a) Q value heatmap

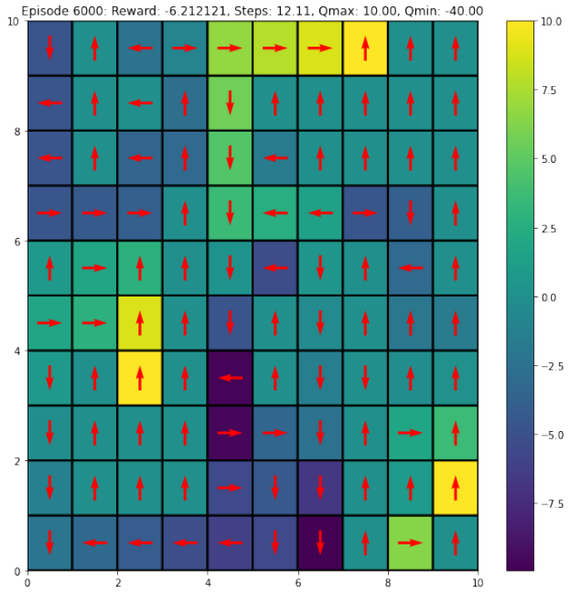


(b) State occupancy heatmap

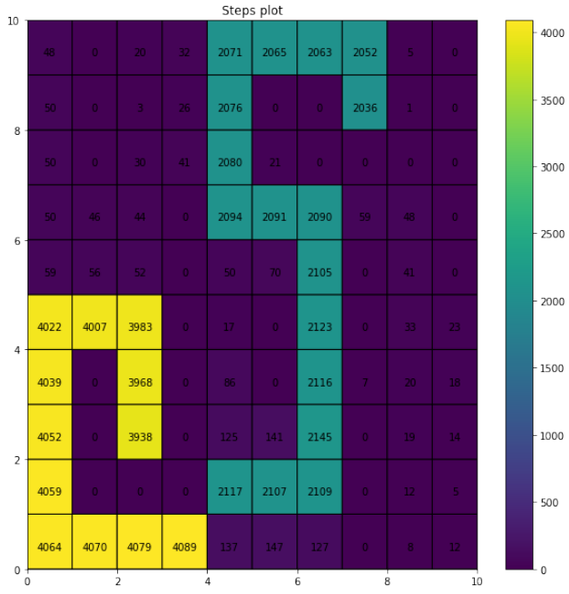


(c) Reward vs. episodes

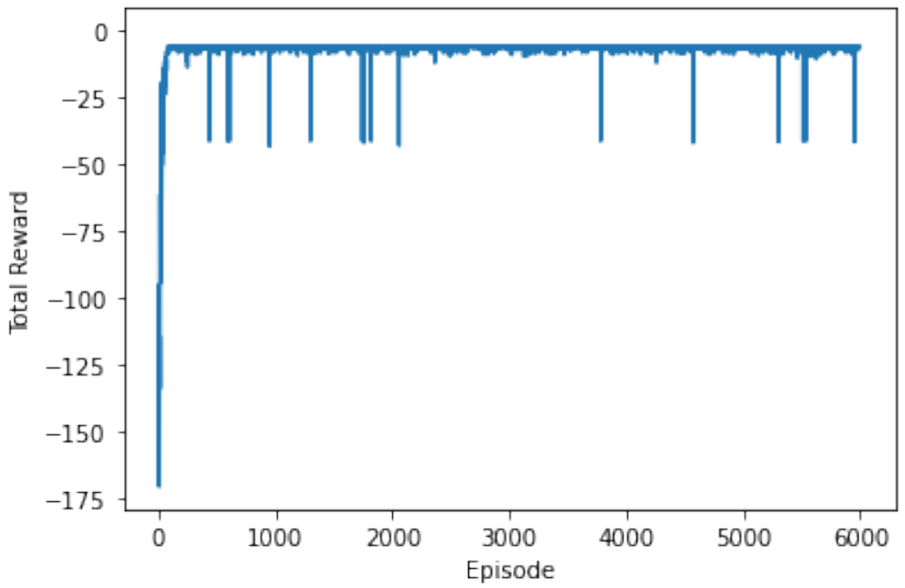
Figure 8: Q Learning for config 8



(a) Q value heatmap

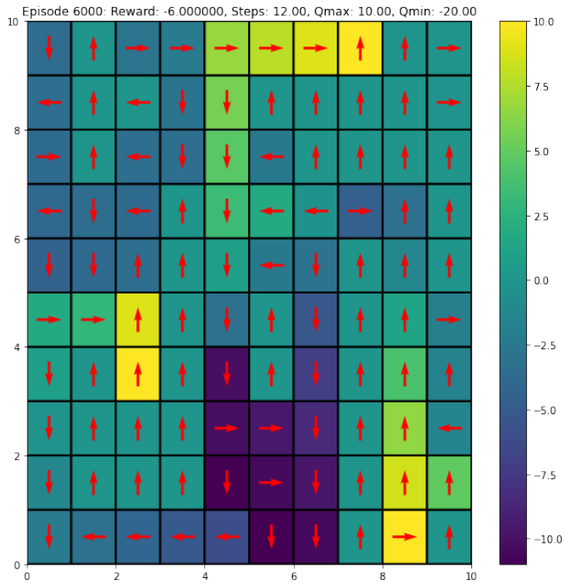


(b) State occupancy heatmap

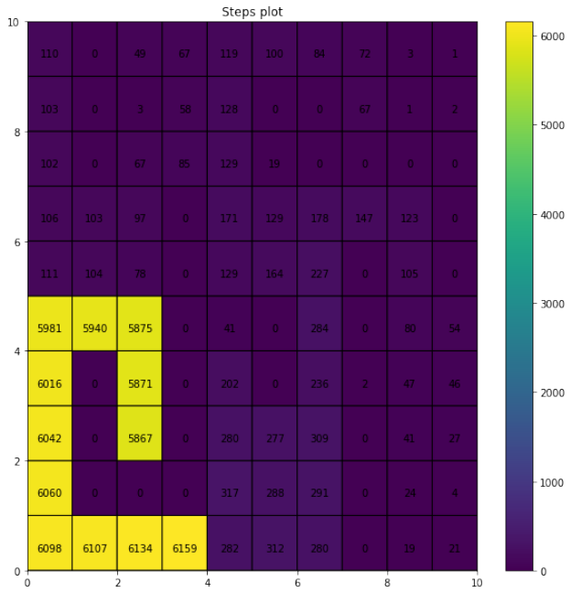


(c) Reward vs. episodes

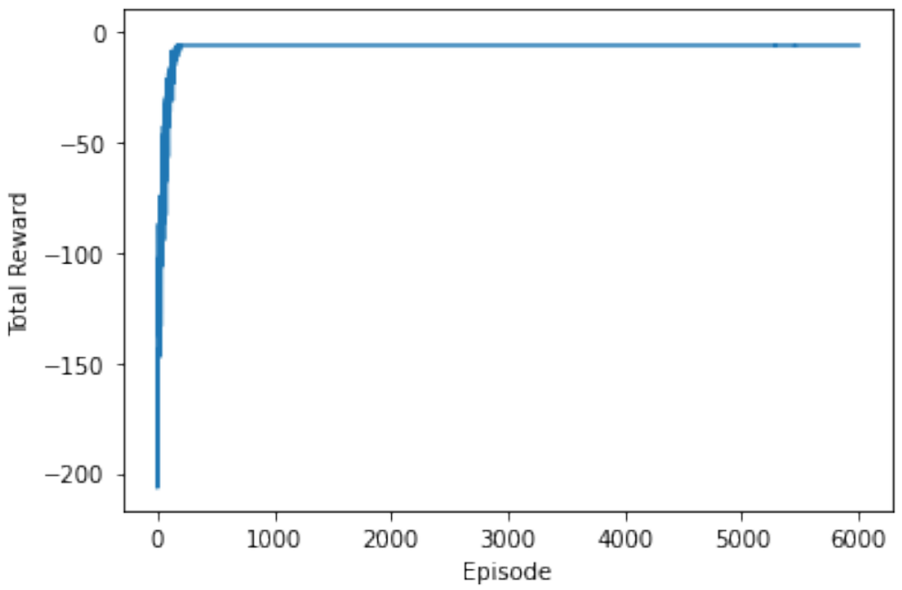
Figure 9: Q Learning for config 9



(a) Q value heatmap

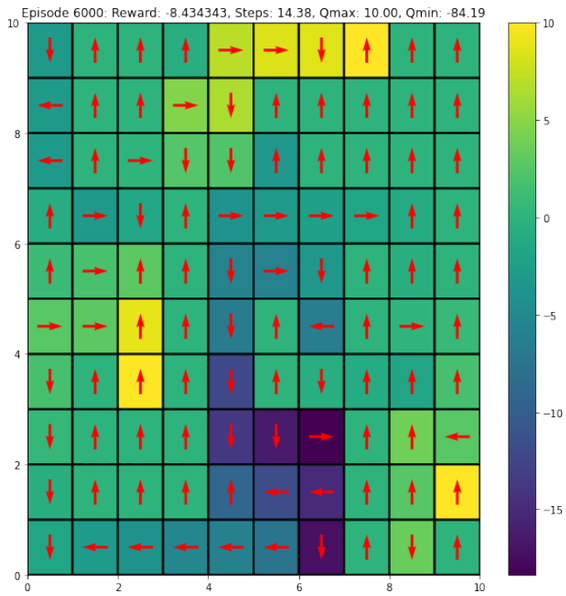


(b) State occupancy heatmap

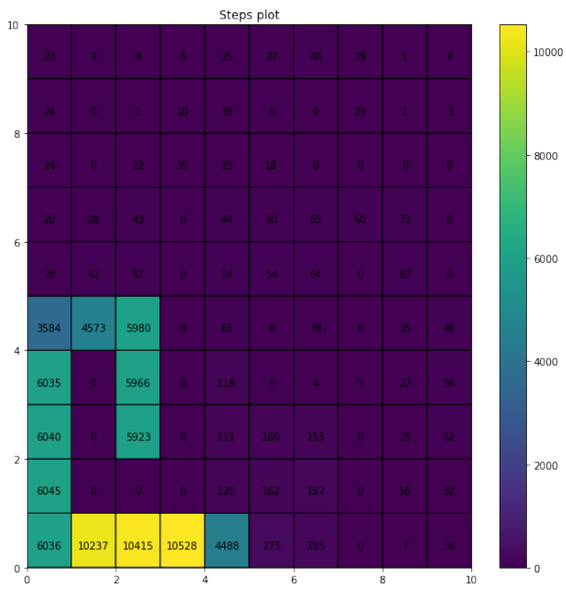


(c) Reward vs. episodes

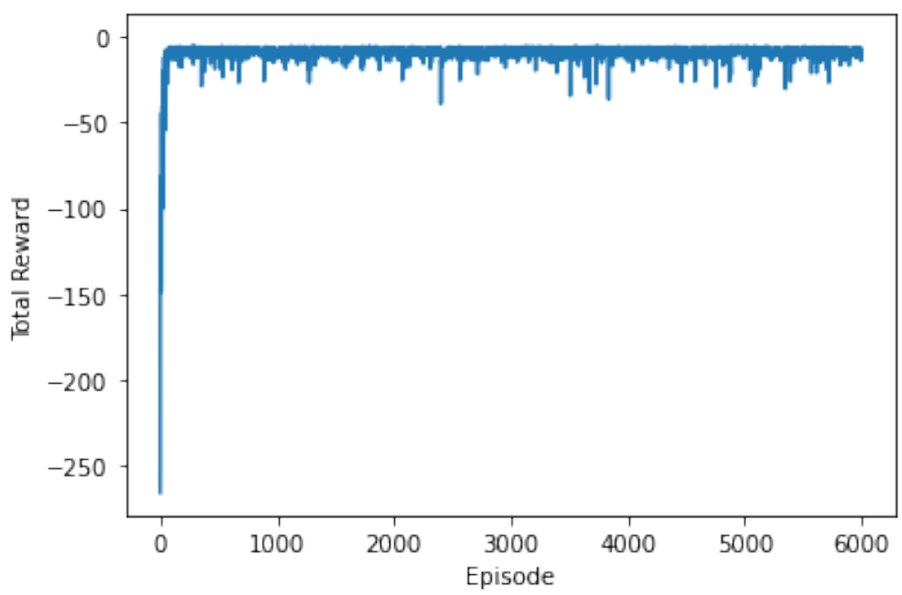
Figure 10: Q Learning for config 10



(a) Q value heatmap

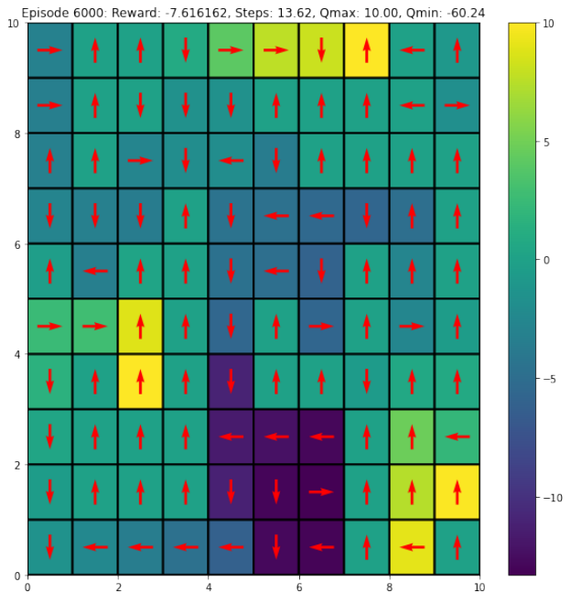


(b) State occupancy heatmap

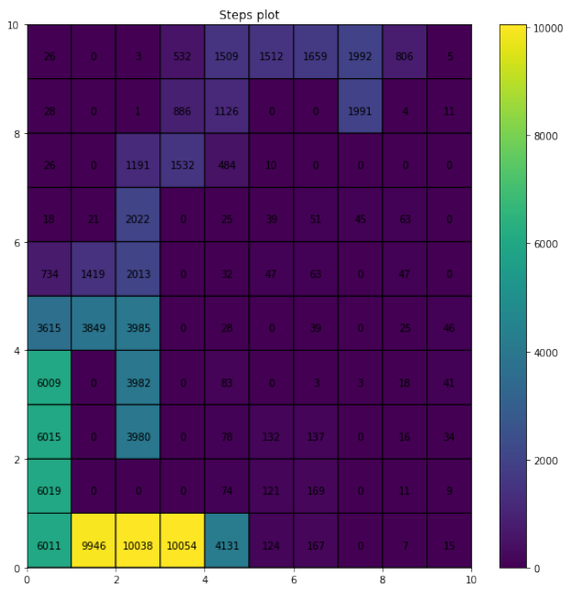


(c) Reward vs. episodes

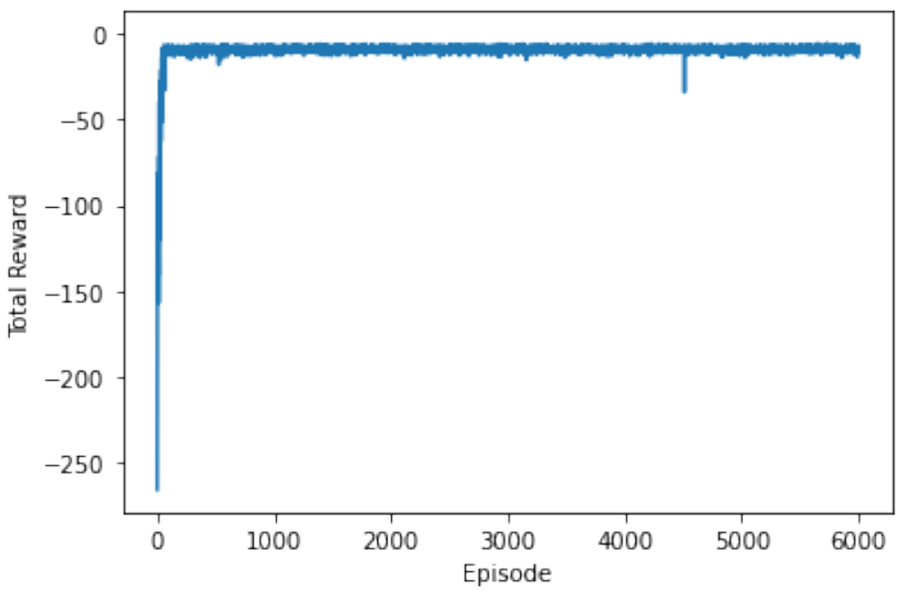
Figure 11: Q Learning for config 11



(a) Q value heatmap

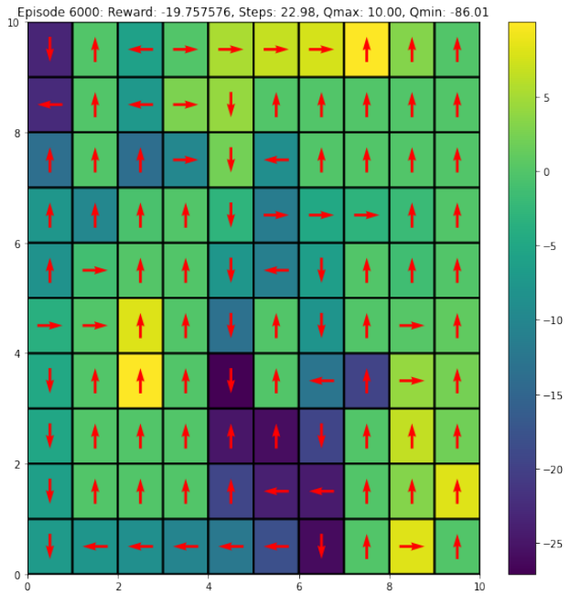


(b) State occupancy heatmap

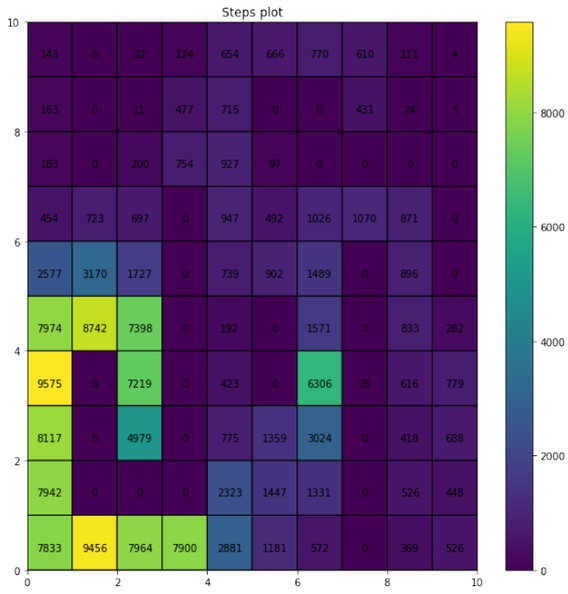


(c) Reward vs. episodes

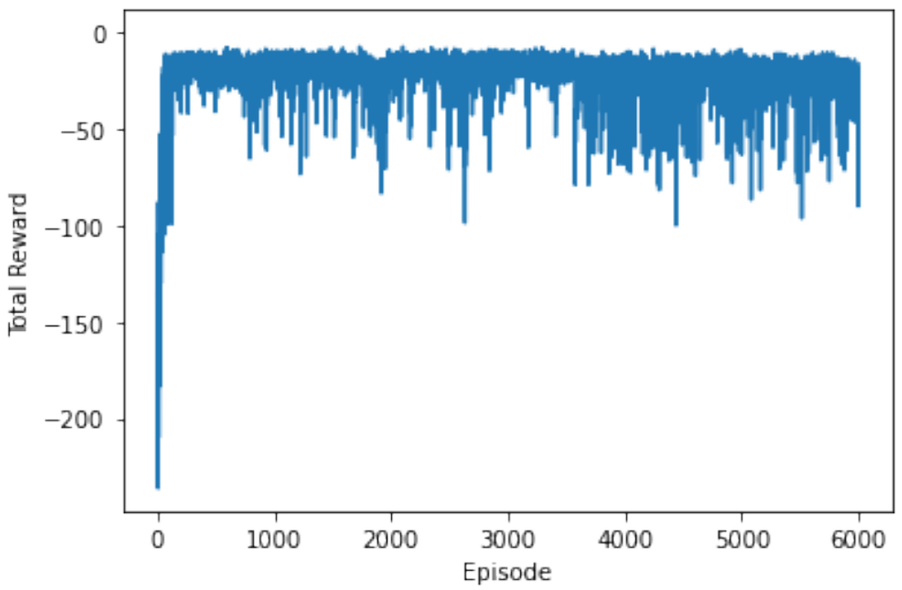
Figure 12: Q Learning for config 12



(a) Q value heatmap

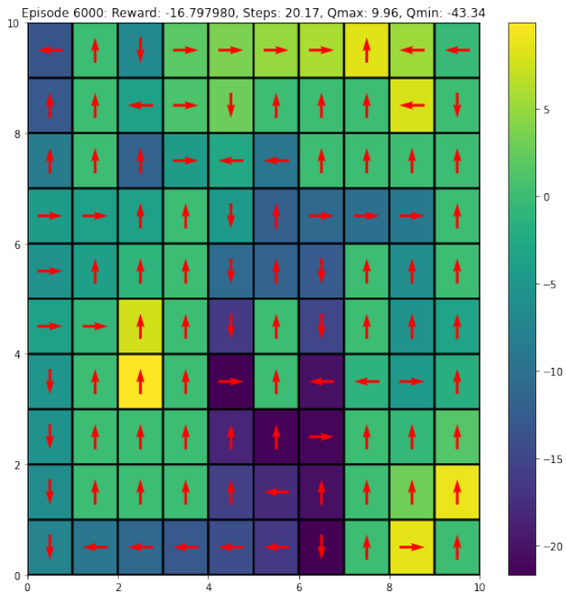


(b) State occupancy heatmap

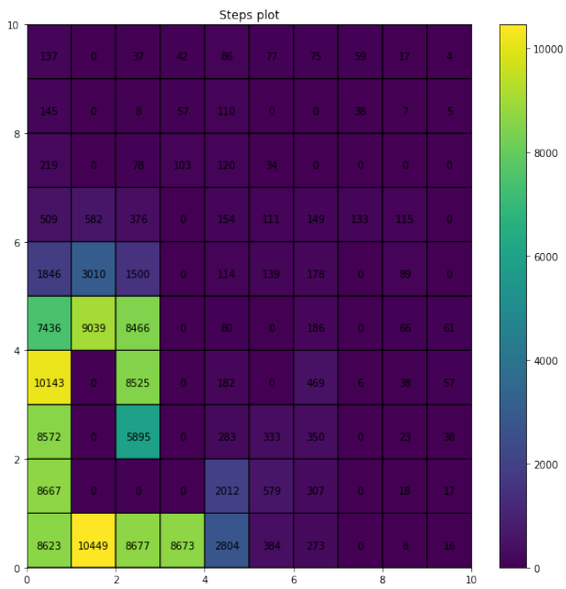


(c) Reward vs. episodes

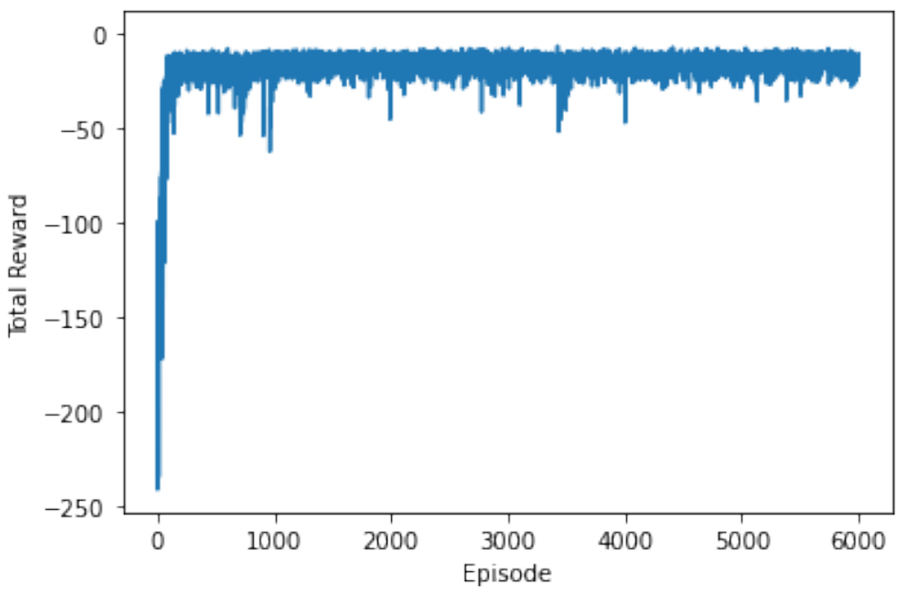
Figure 13: Q Learning for config 13



(a) Q value heatmap

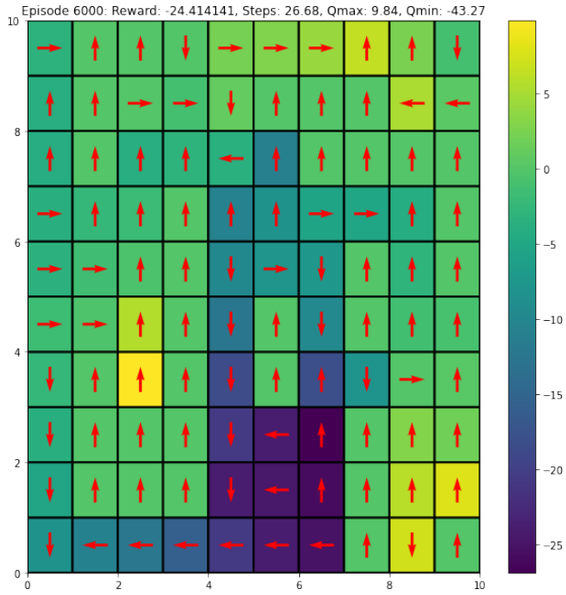


(b) State occupancy heatmap

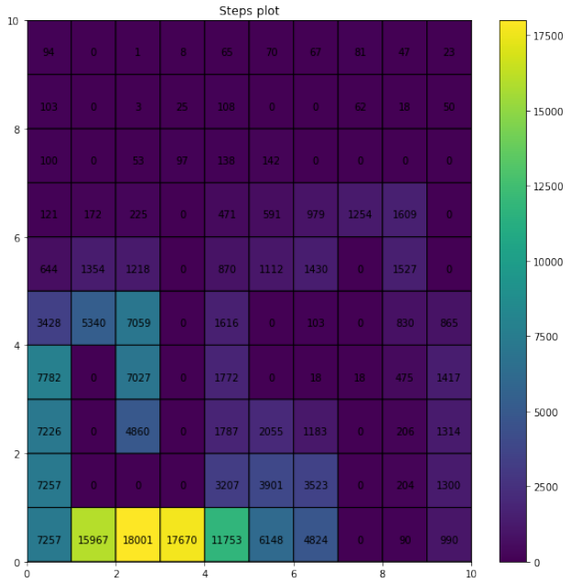


(c) Reward vs. episodes

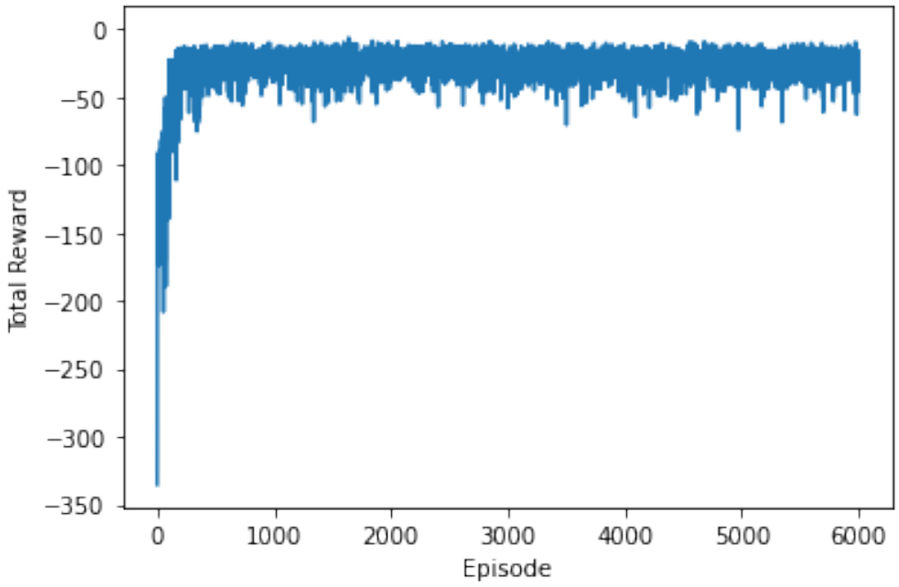
Figure 14: Q Learning for config 14



(a) Q value heatmap

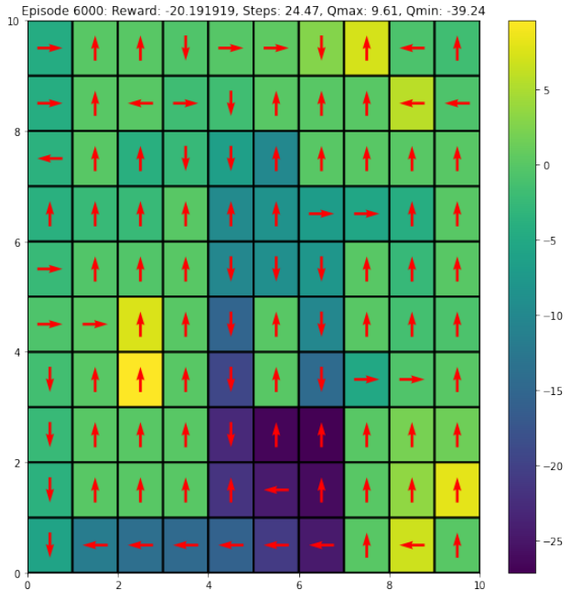


(b) State occupancy heatmap

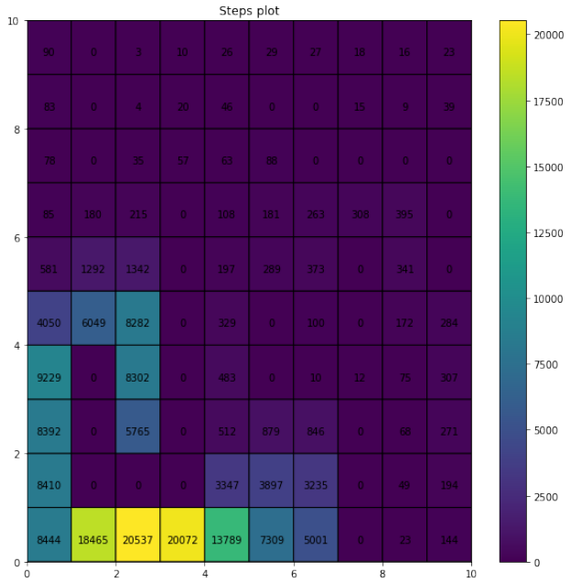


(c) Reward vs. episodes

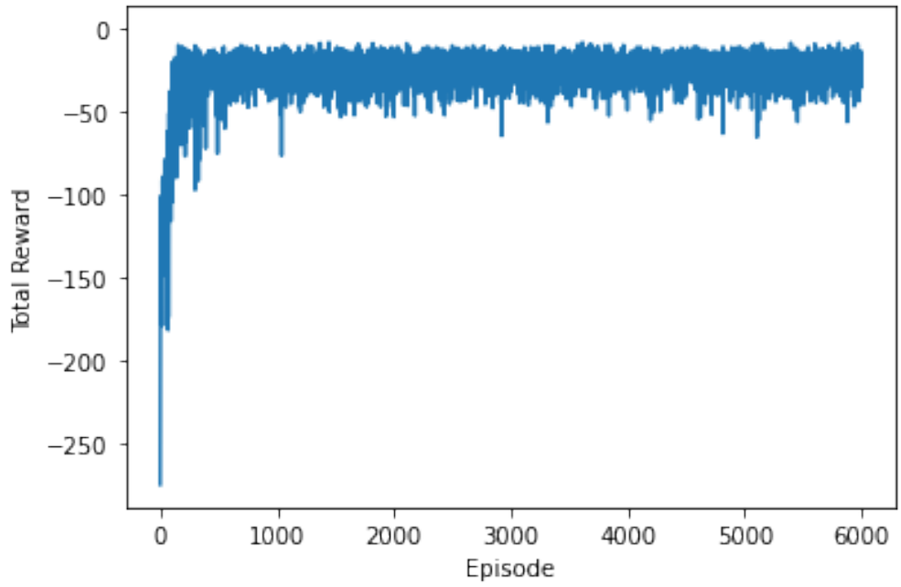
Figure 15: Q Learning for config 15



(a) Q value heatmap



(b) State occupancy heatmap



(c) Reward vs. episodes

Figure 16: Q Learning for config 16

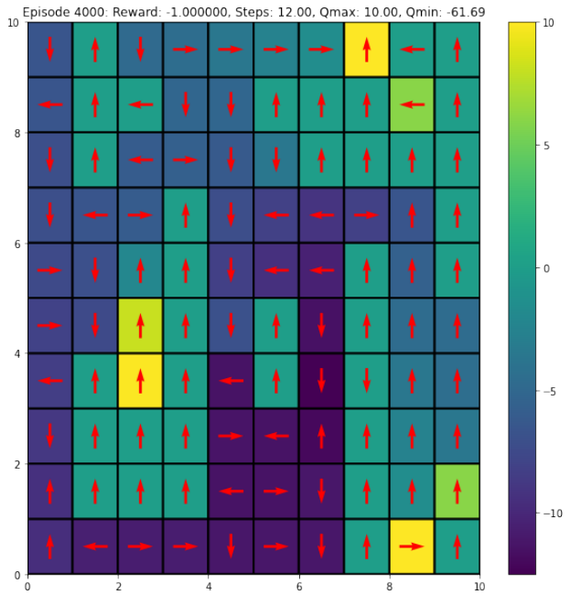
2 SARSA

We now describe our findings for the SARSA algorithm in the given environment. We run each experiment for 6000 episodes based on our observations of the convergence behaviour of the algorithm. The below table specifies the optimal values of hyperparameters that are found after performing tuning sweeps for each of the 16 configurations. Table 2 shows the values.

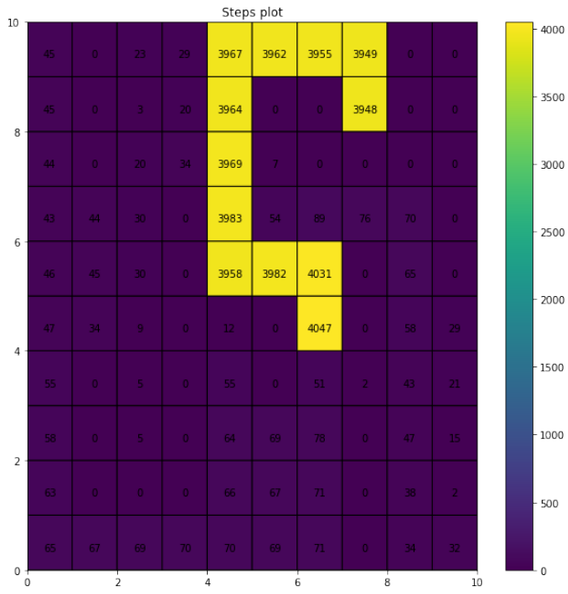
Config	Start	p-val	Wind	Explore	α	β	ϵ	γ	Return
1	(3, 6)	1.0	F	ϵ-greedy	0.6	-	0	1.0	-1.0
2	(3, 6)	1.0	F	softmax	0.3	0.1	-	0.99	-1.3
3	(3, 6)	1.0	T	ϵ-greedy	0.2	-	0	0.99	-4.09
4	(3, 6)	1.0	T	softmax	0.3	0.1	-	0.99	-3.8
5	(3, 6)	0.7	F	ϵ-greedy	0.2	-	0.01	0.99	-21.9
6	(3, 6)	0.7	F	softmax	0.2	0.3	-	0.99	-20.5
7	(3, 6)	0.7	T	ϵ-greedy	0.1	-	0	1.0	-46.2
8	(3, 6)	0.7	T	softmax	0.1	0.01	-	1.0	-36.5
9	(0, 4)	1.0	F	ϵ-greedy	0.6	-	0.01	0.8	-6.04
10	(0, 4)	1.0	F	softmax	0.6	0.01	-	1.0	-6.0
11	(0, 4)	1.0	T	ϵ-greedy	0.4	-	0.001	1.0	-9.5
12	(0, 4)	1.0	T	softmax	0.6	0.1	-	1.0	-9.7
13	(0, 4)	0.7	F	ϵ-greedy	0.3	-	0.01	0.99	-18.6
14	(0, 4)	0.7	F	softmax	0.2	0.3	-	0.99	-17.8
15	(0, 4)	0.7	T	ϵ-greedy	0.2	-	0.01	1.0	-35.7
16	(0, 4)	0.7	T	softmax	0.2	0.3	-	1.0	-25.4

Table 2: Hyperparamater Tuning Results for SARSA

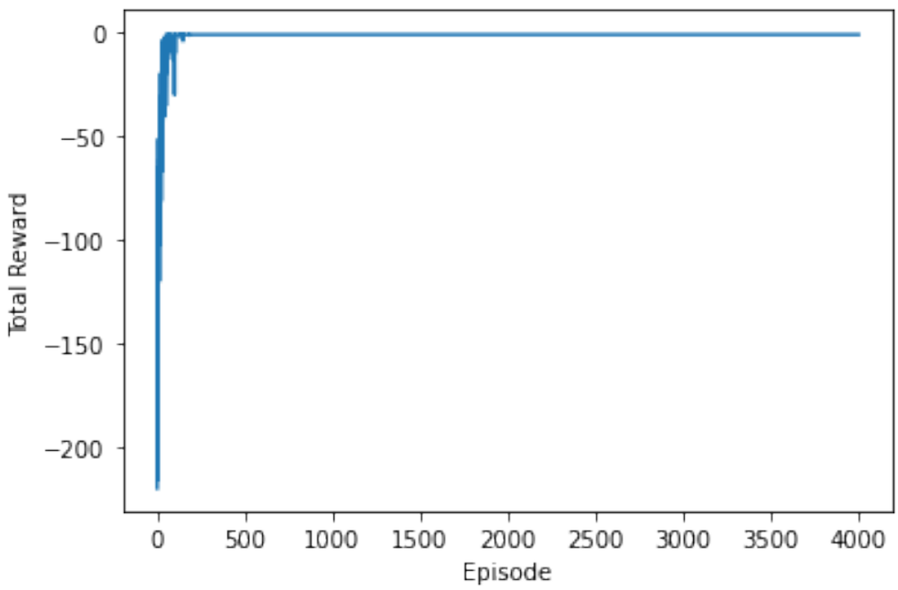
The following figures show the results for each of the configs mentioned in the above table.



(a) Q value heatmap

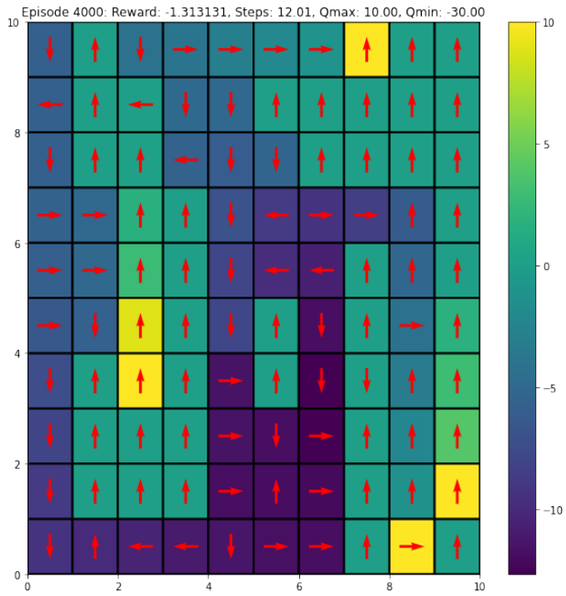


(b) State occupancy heatmap

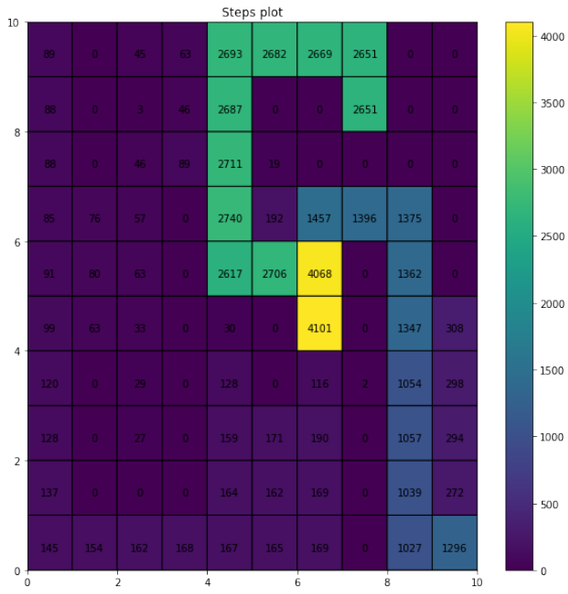


(c) Reward vs. episodes

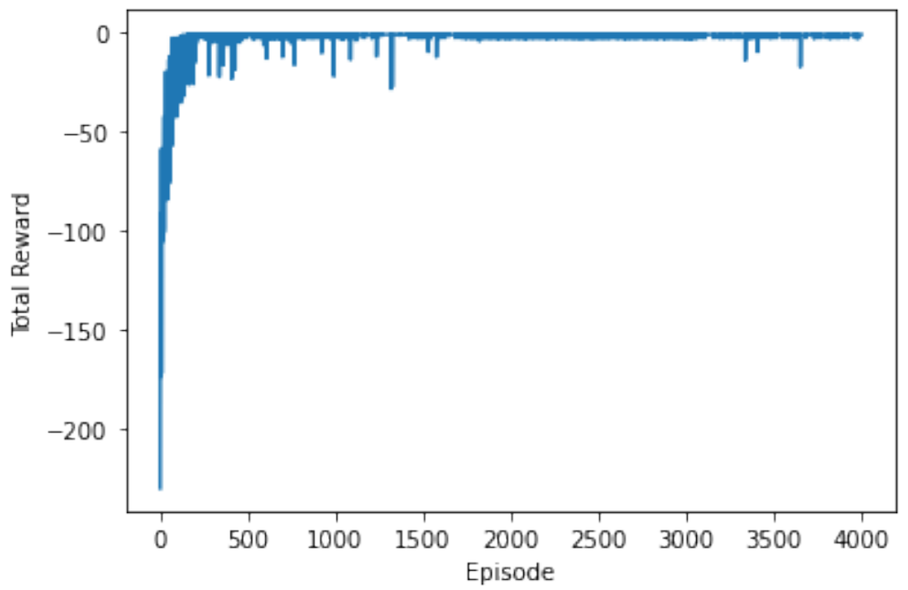
Figure 17: sarsa for config 1



(a) Q value heatmap

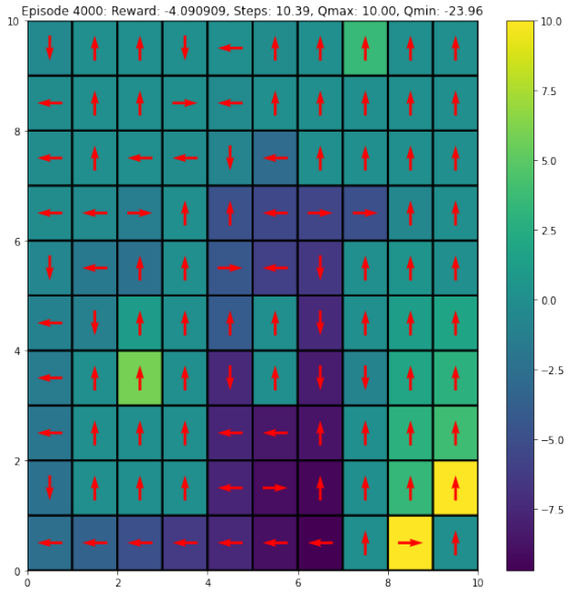


(b) State occupancy heatmap

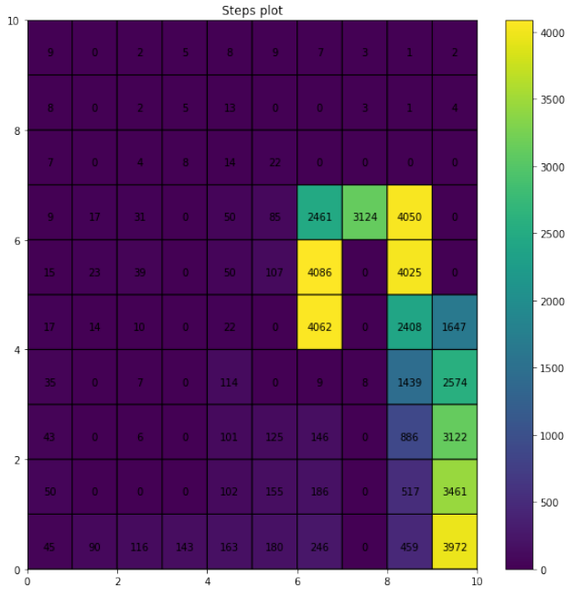


(c) Reward vs. episodes

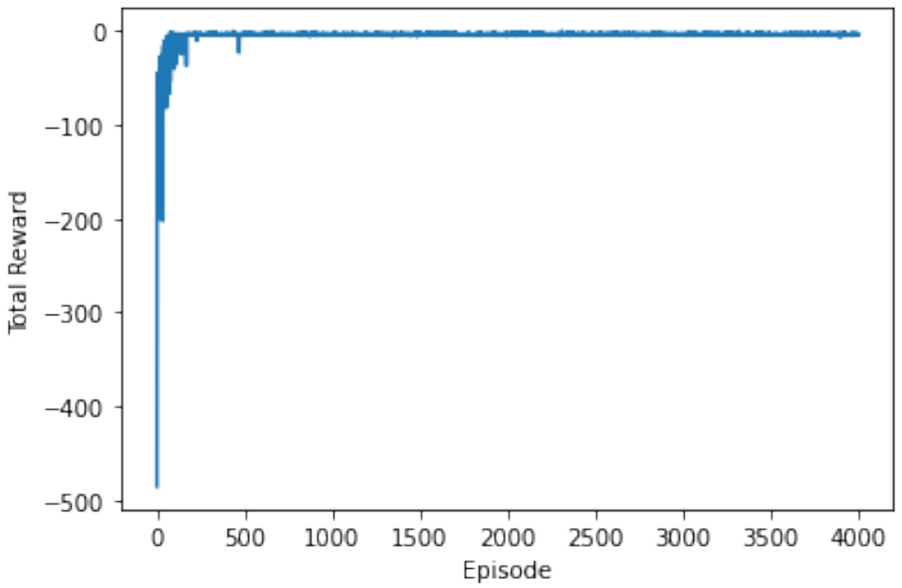
Figure 18: sarsa for config 2



(a) Q value heatmap

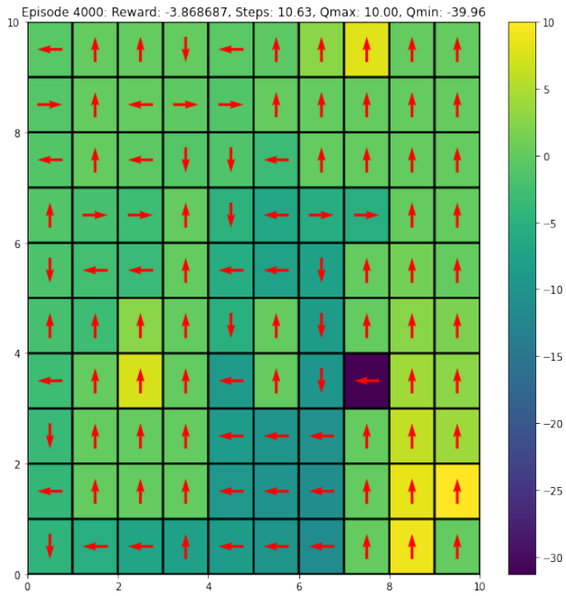


(b) State occupancy heatmap

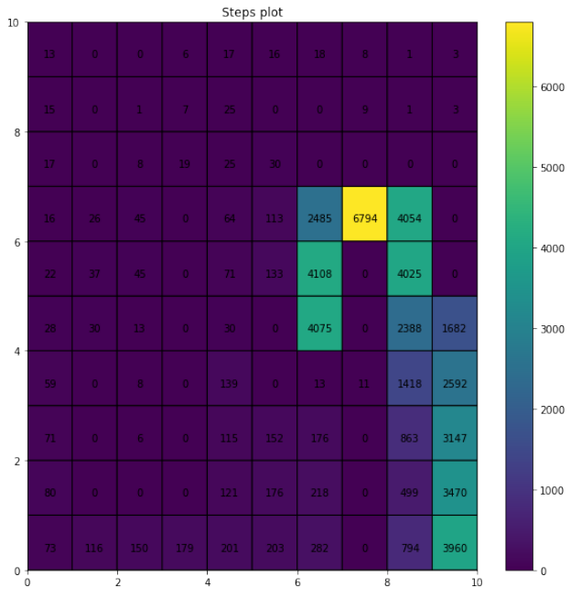


(c) Reward vs. episodes

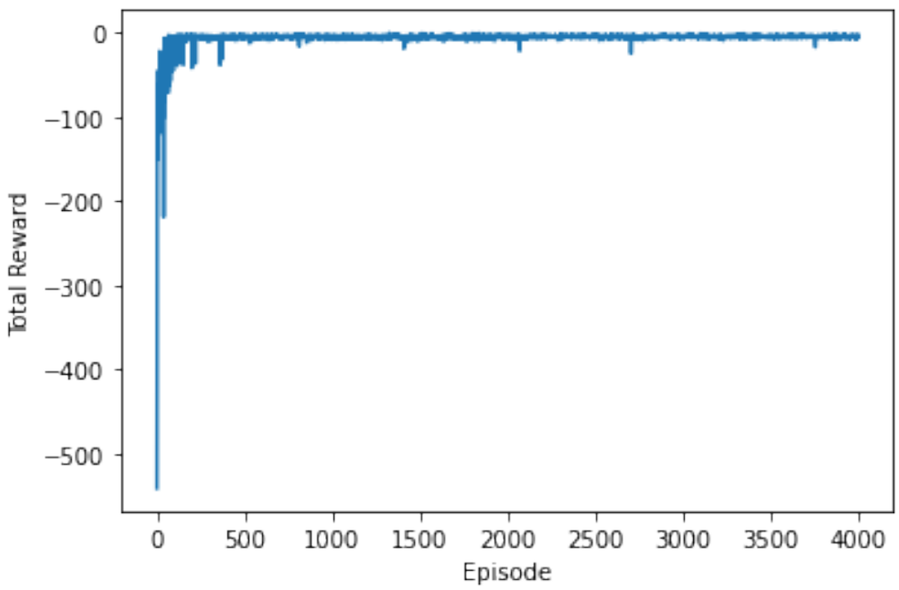
Figure 19: sarsa for config 3



(a) Q value heatmap

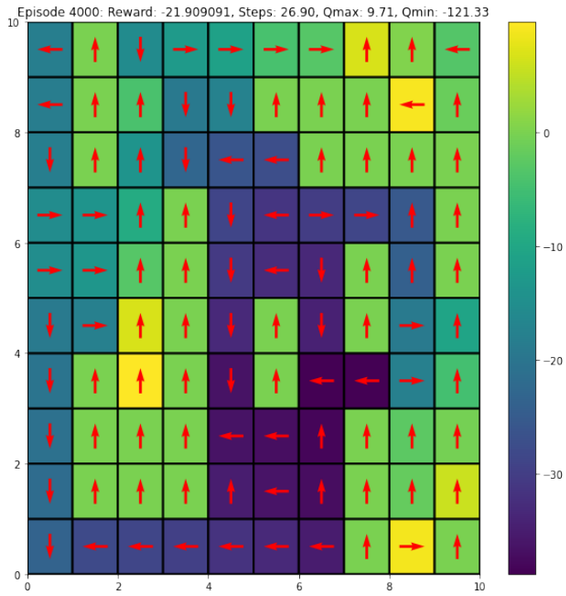


(b) State occupancy heatmap

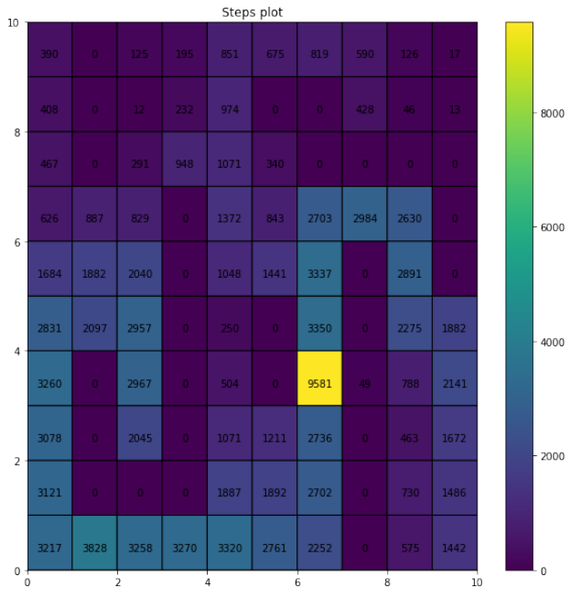


(c) Reward vs. episodes

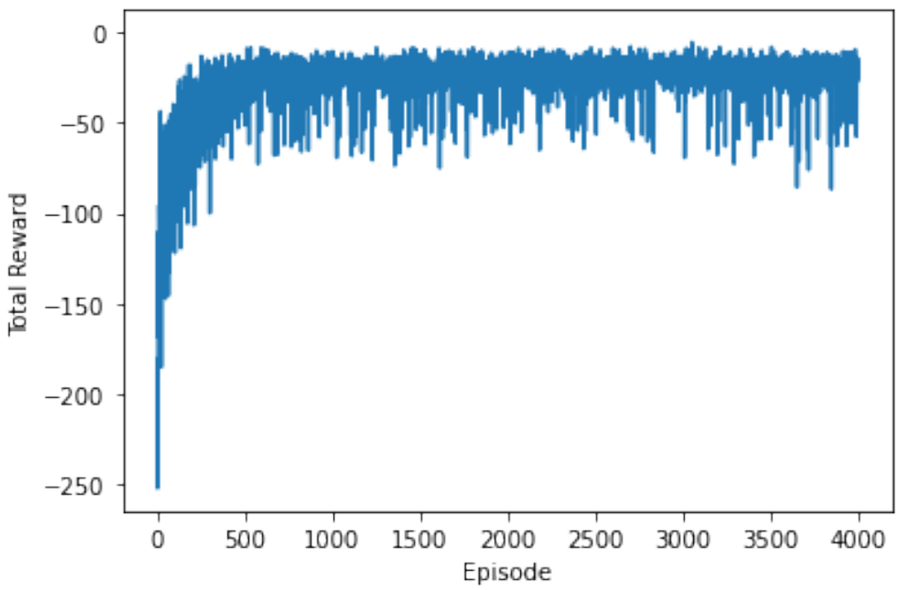
Figure 20: sarsa for config 4



(a) Q value heatmap

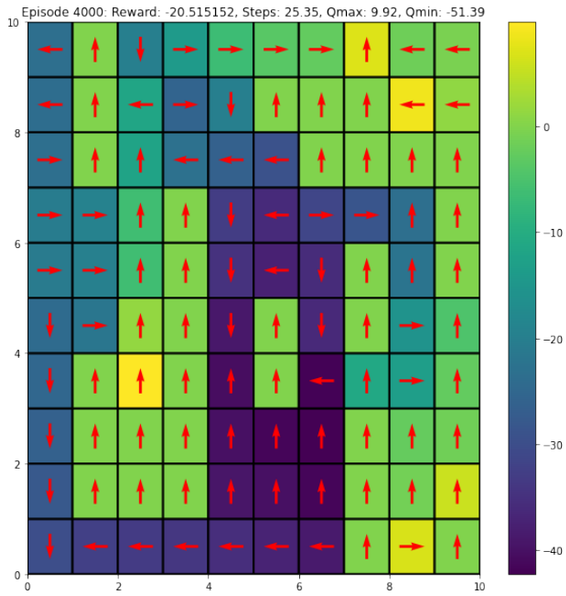


(b) State occupancy heatmap

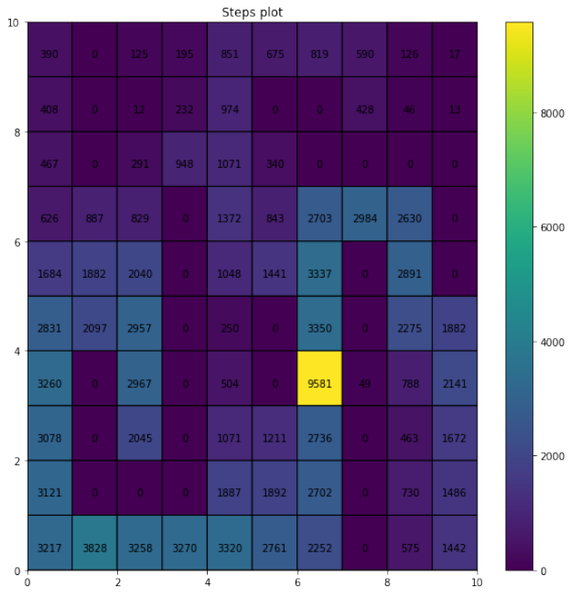


(c) Reward vs. episodes

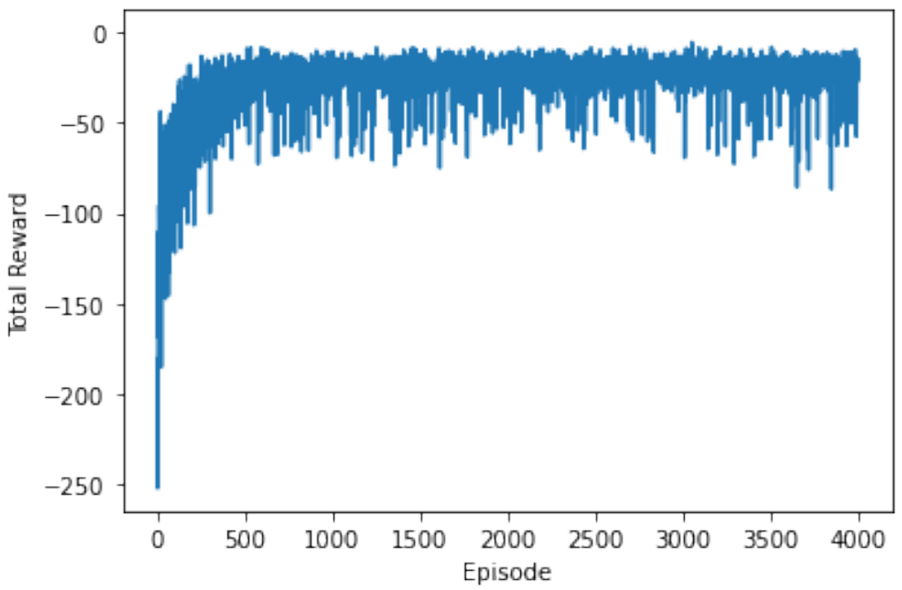
Figure 21: sarsa for config 5



(a) Q value heatmap

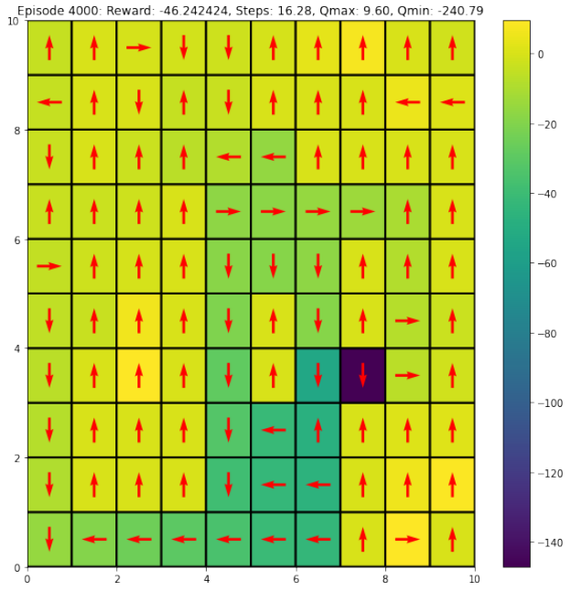


(b) State occupancy heatmap

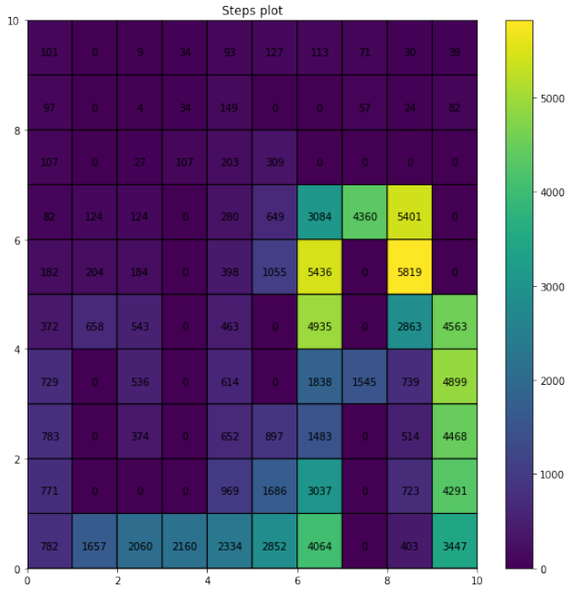


(c) Reward vs. episodes

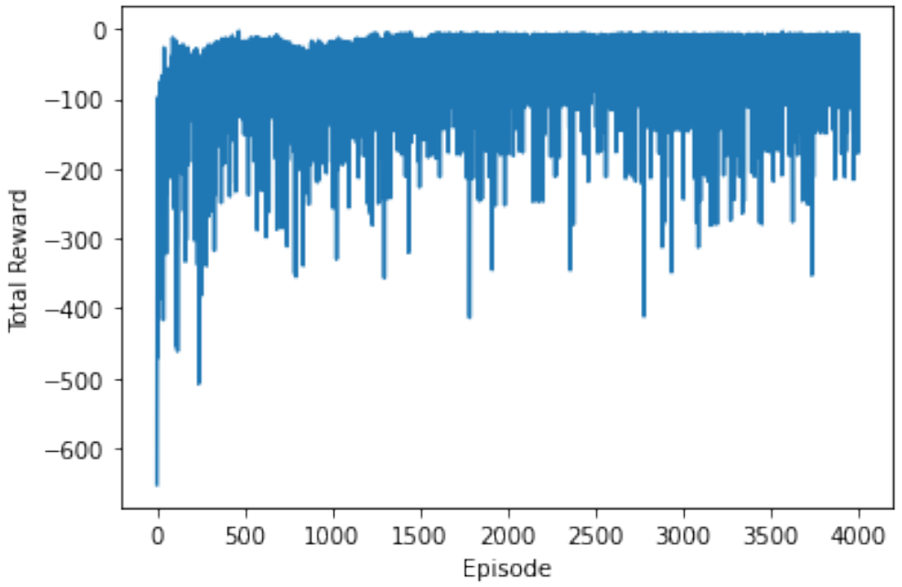
Figure 22: sarsa for config 6



(a) Q value heatmap

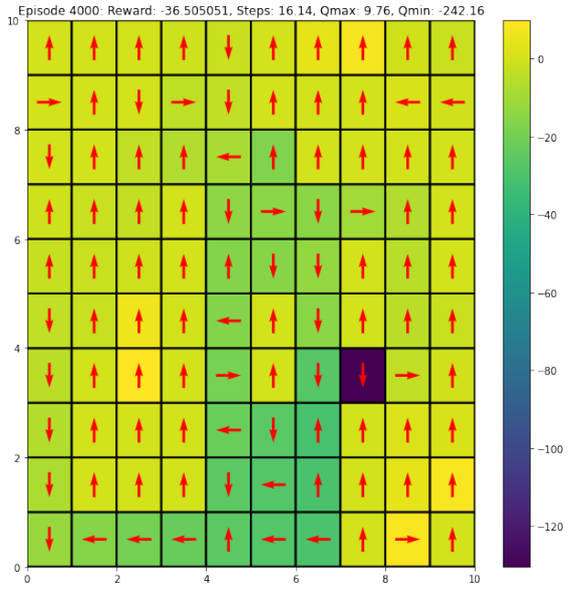


(b) State occupancy heatmap

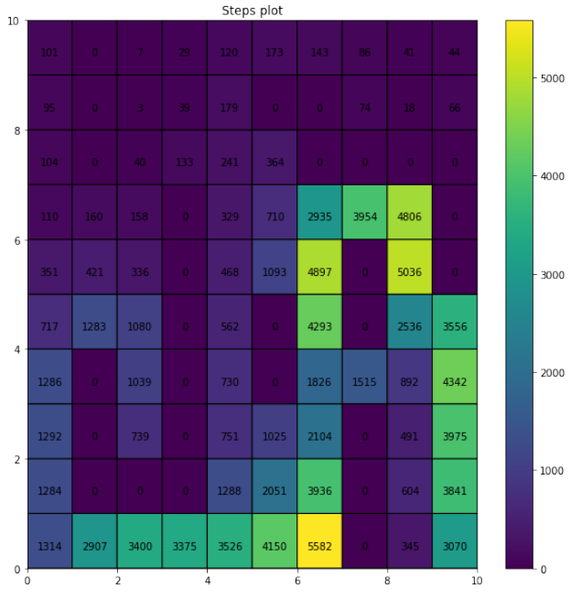


(c) Reward vs. episodes

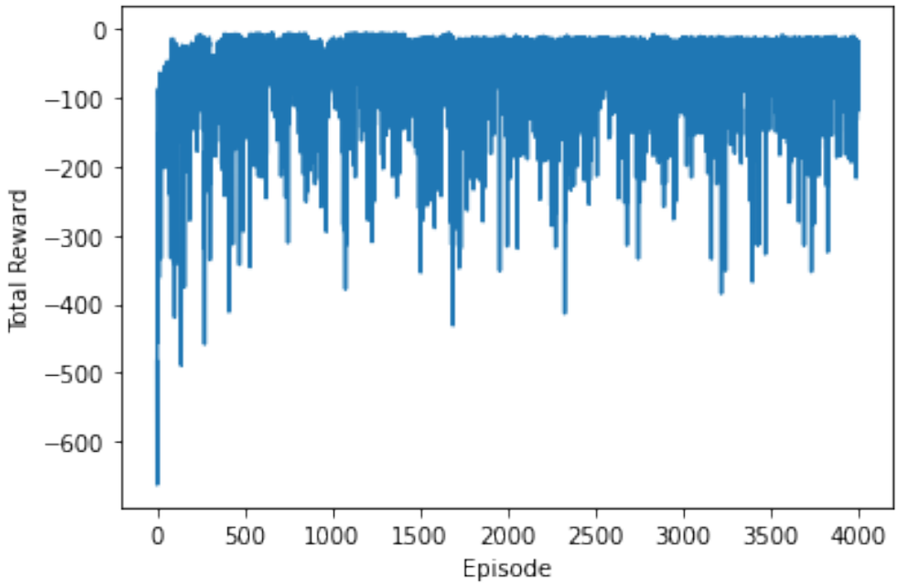
Figure 23: sarsa for config 7



(a) Q value heatmap

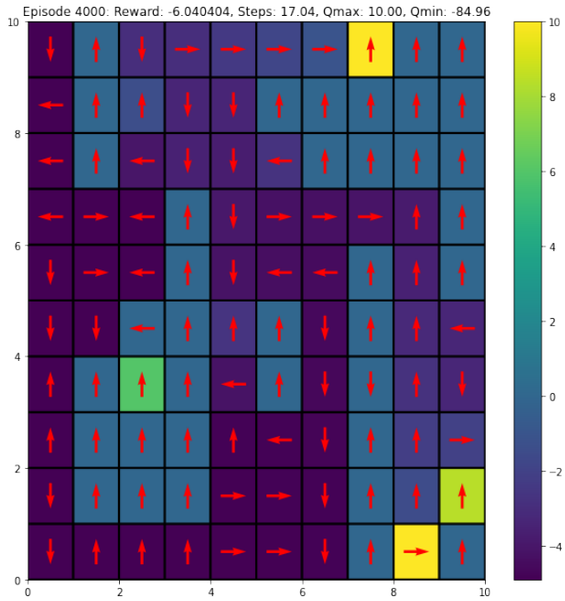


(b) State occupancy heatmap

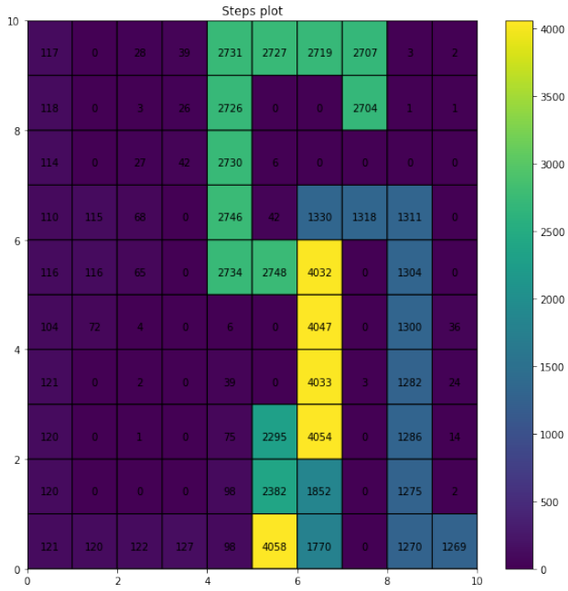


(c) Reward vs. episodes

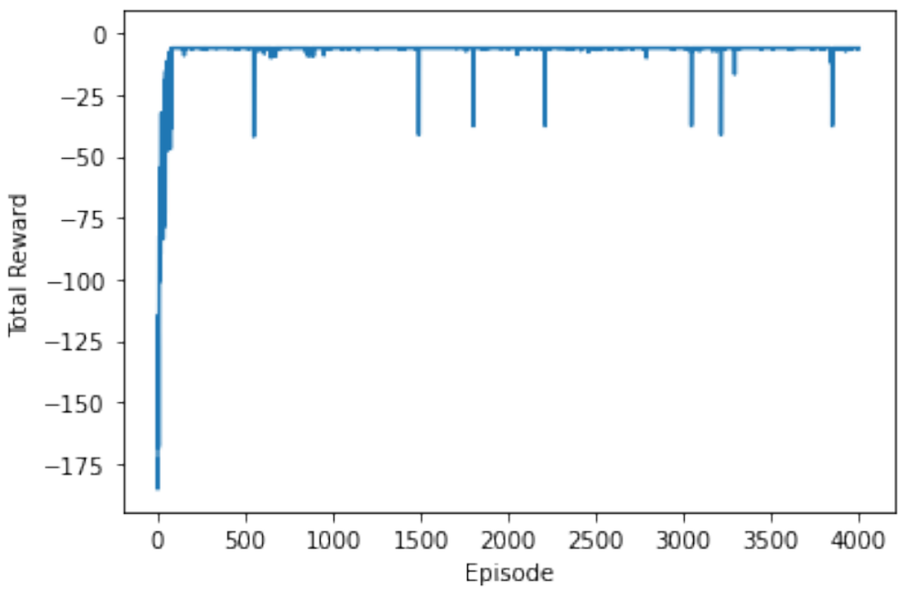
Figure 24: sarsa for config 8



(a) Q value heatmap

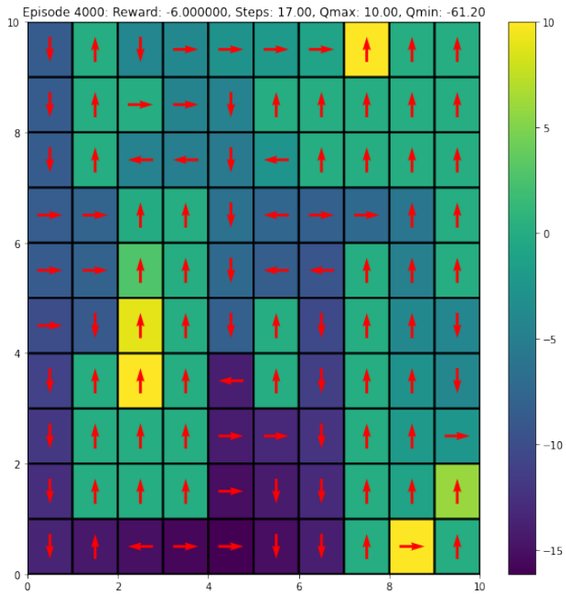


(b) State occupancy heatmap

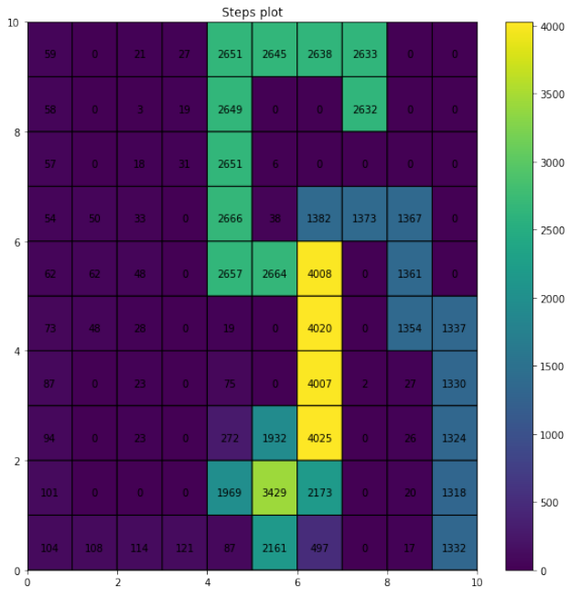


(c) Reward vs. episodes

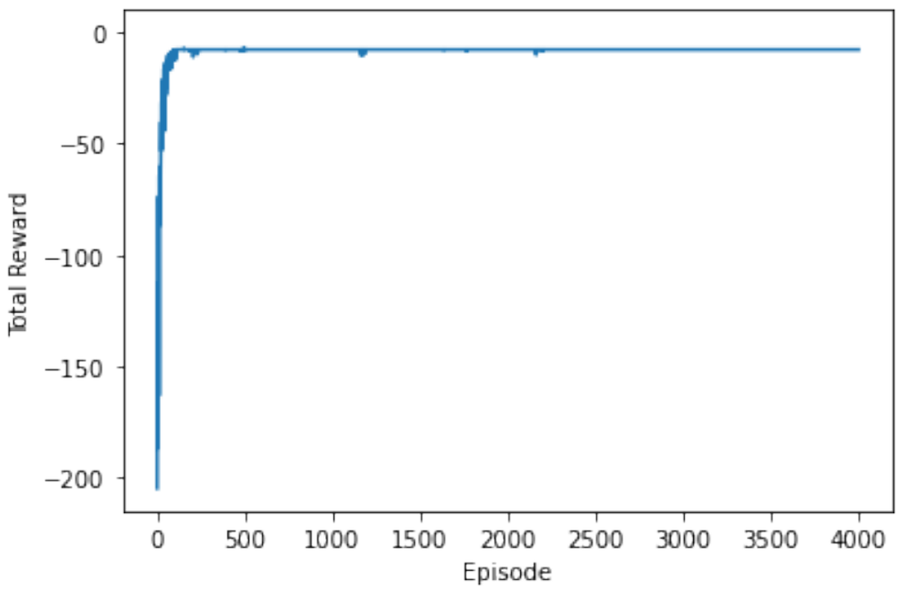
Figure 25: sarsa for config 9



(a) Q value heatmap

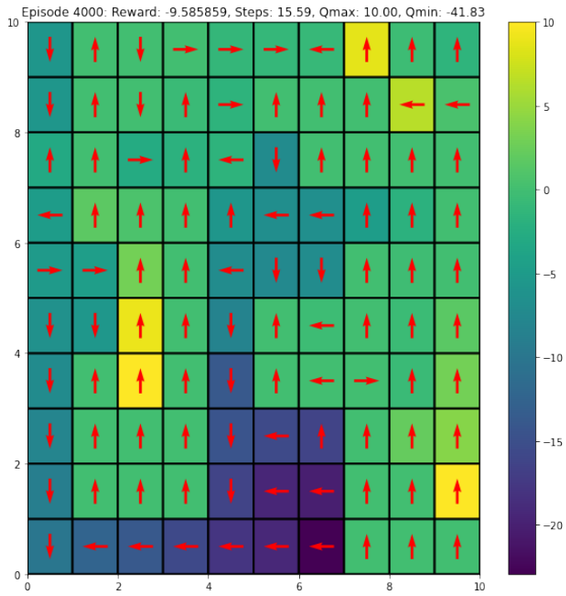


(b) State occupancy heatmap

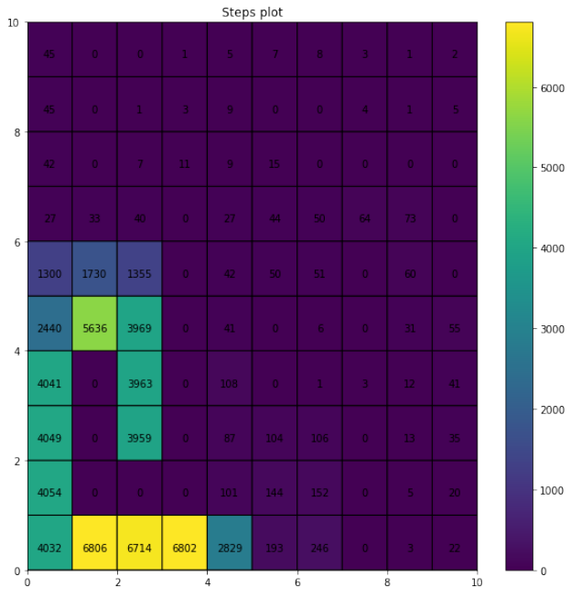


(c) Reward vs. episodes

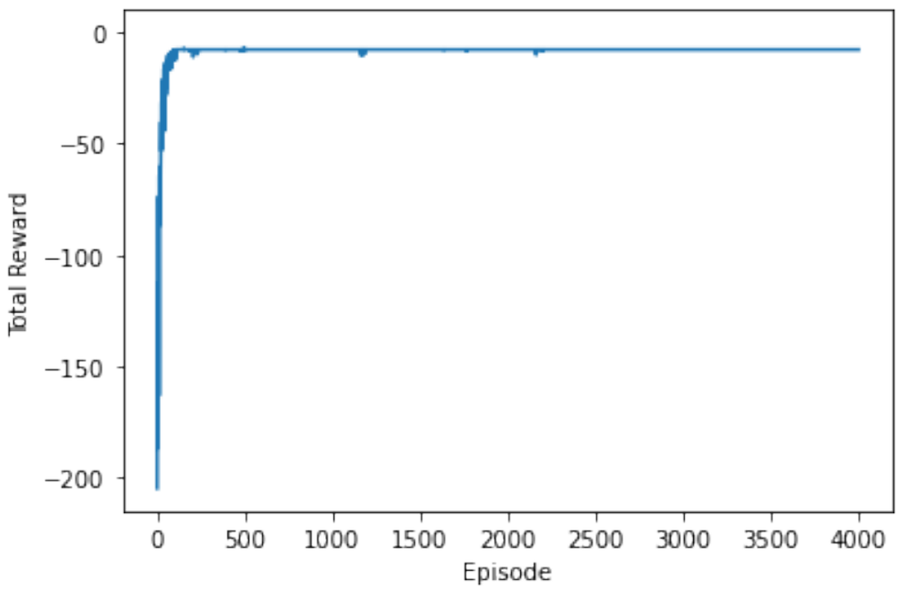
Figure 26: sarsa for config 10



(a) Q value heatmap

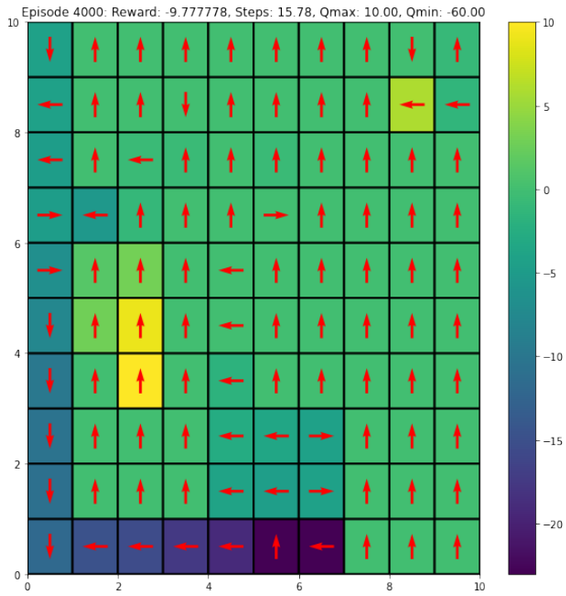


(b) State occupancy heatmap

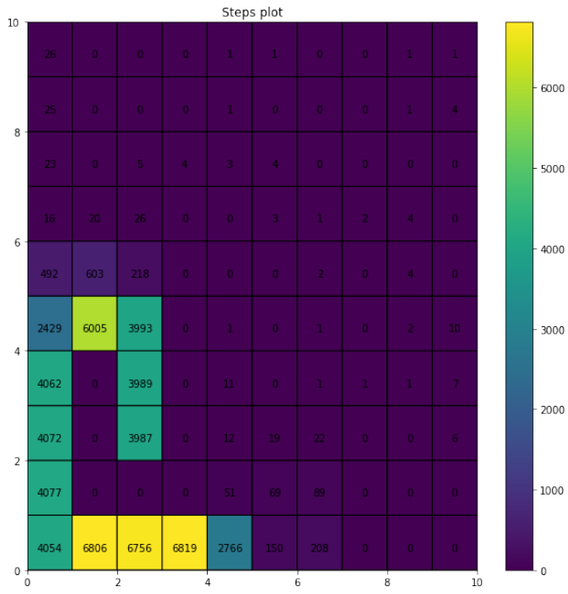


(c) Reward vs. episodes

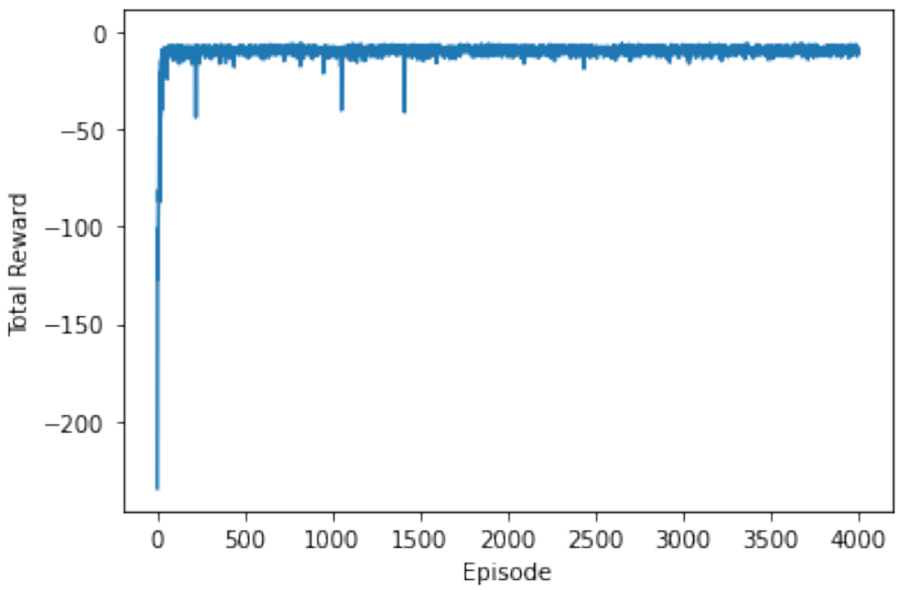
Figure 27: sarsa for config 11



(a) Q value heatmap

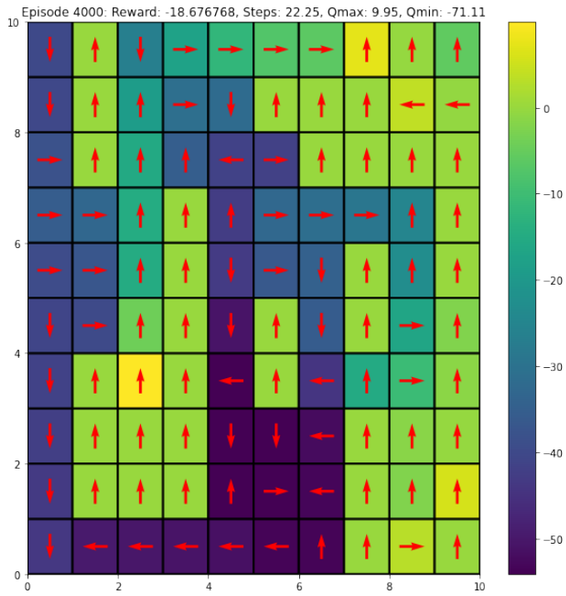


(b) State occupancy heatmap

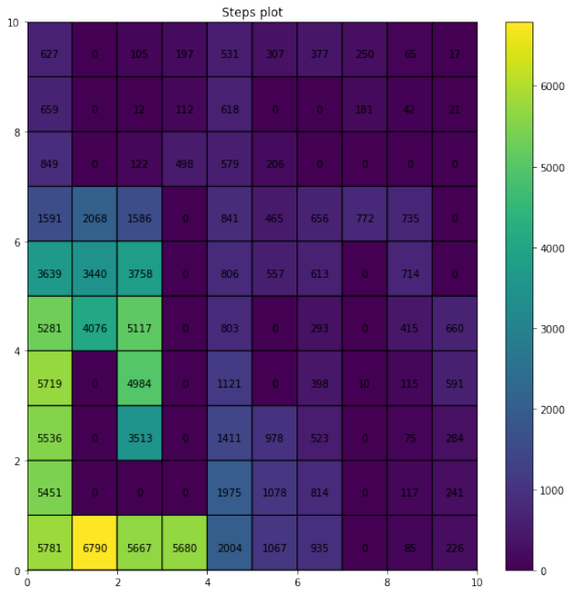


(c) Reward vs. episodes

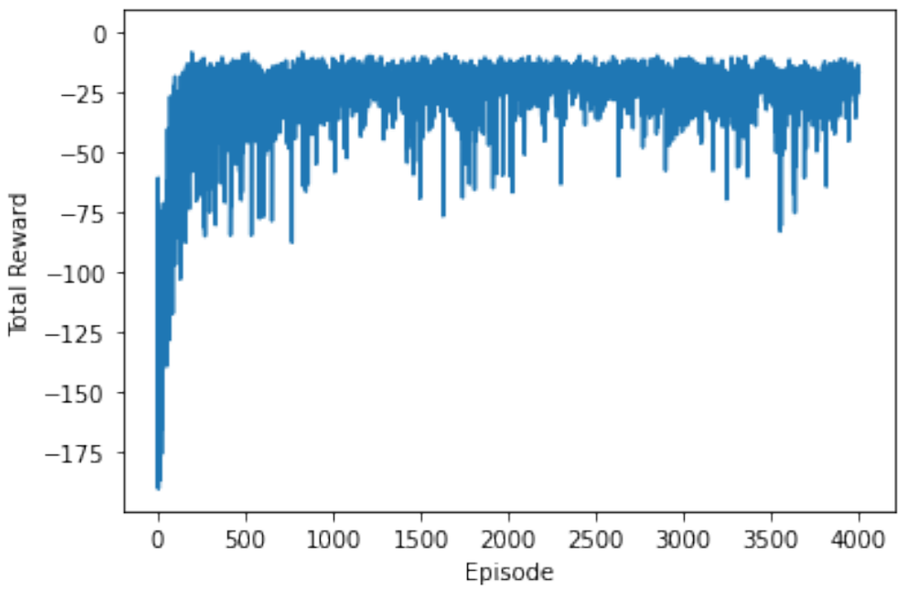
Figure 28: sarsa for config 12



(a) Q value heatmap

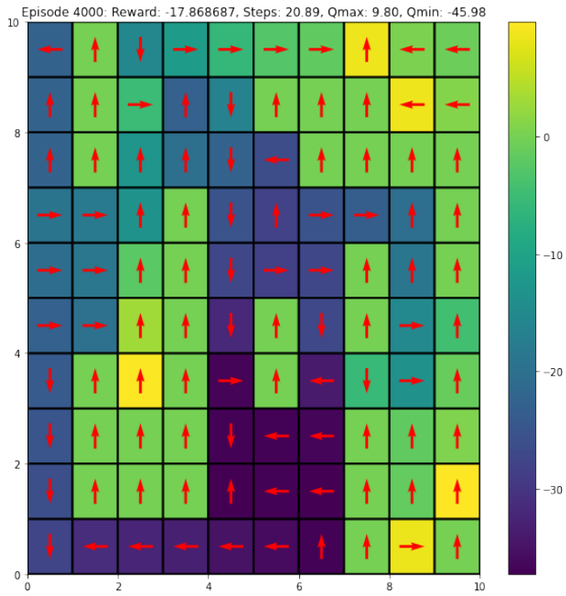


(b) State occupancy heatmap

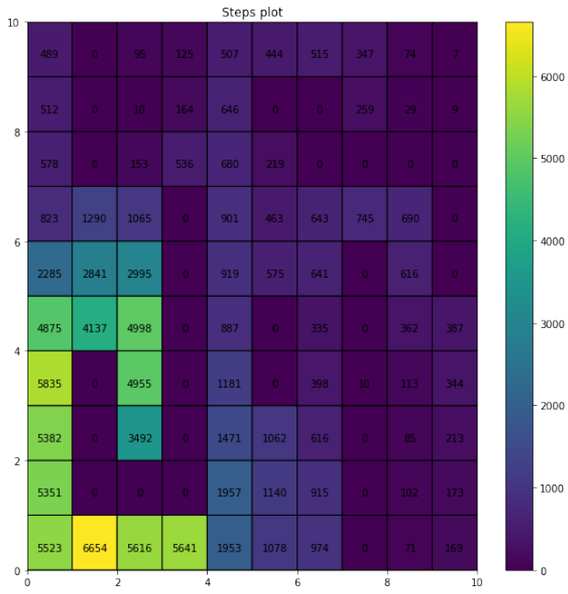


(c) Reward vs. episodes

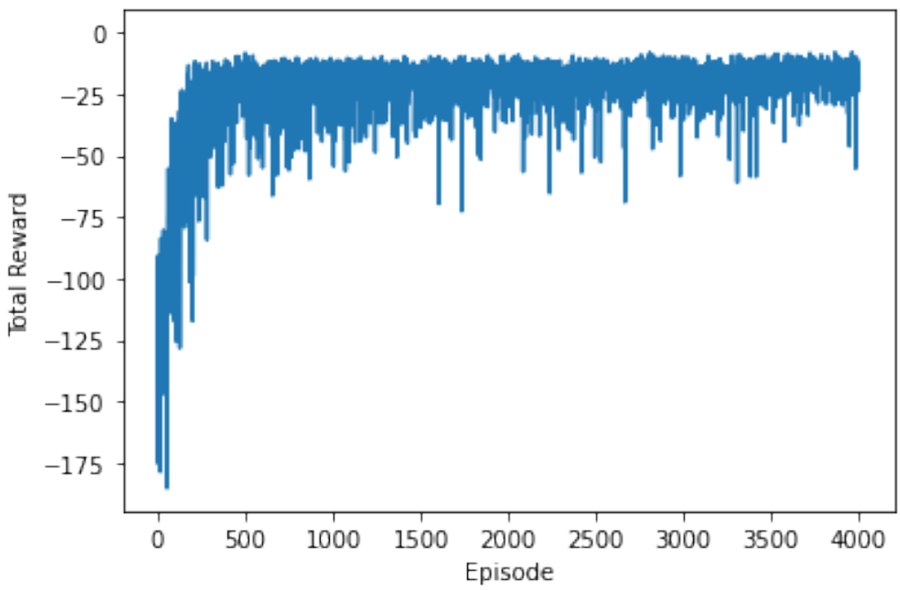
Figure 29: sarsa for config 13



(a) Q value heatmap

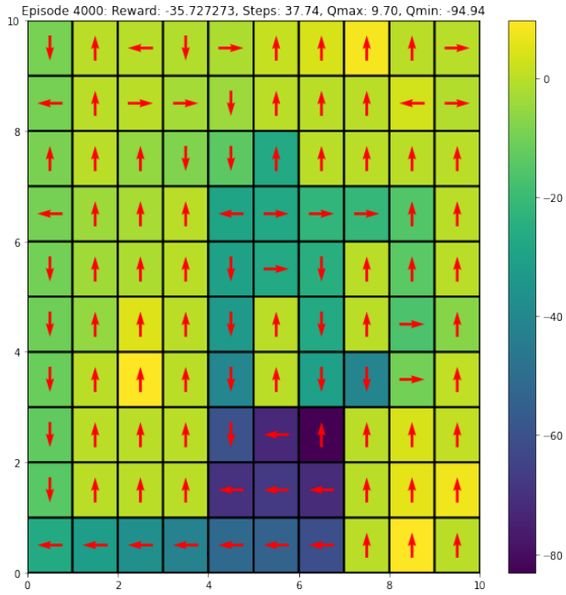


(b) State occupancy heatmap

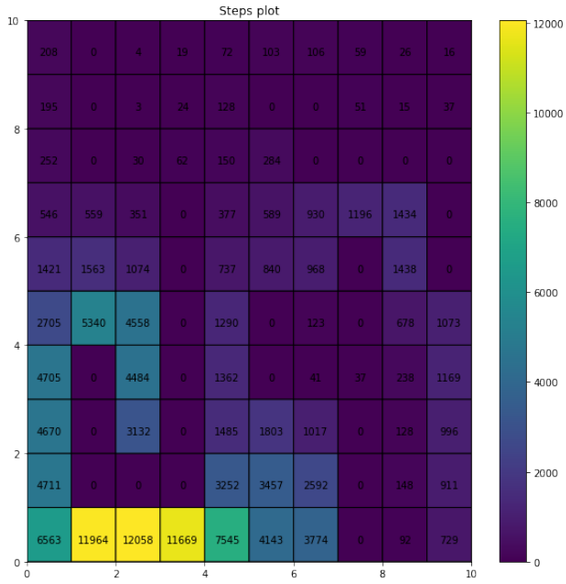


(c) Reward vs. episodes

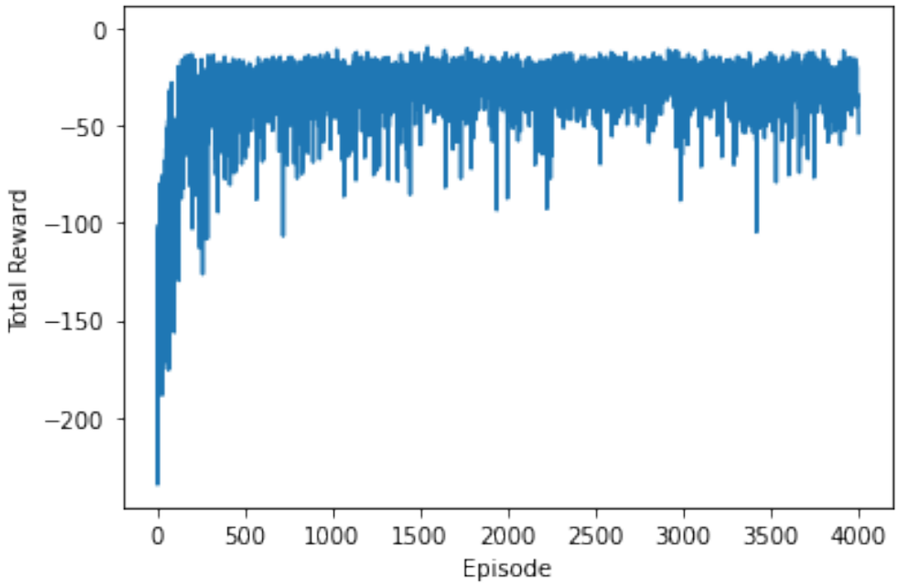
Figure 30: sarsa for config 14



(a) Q value heatmap

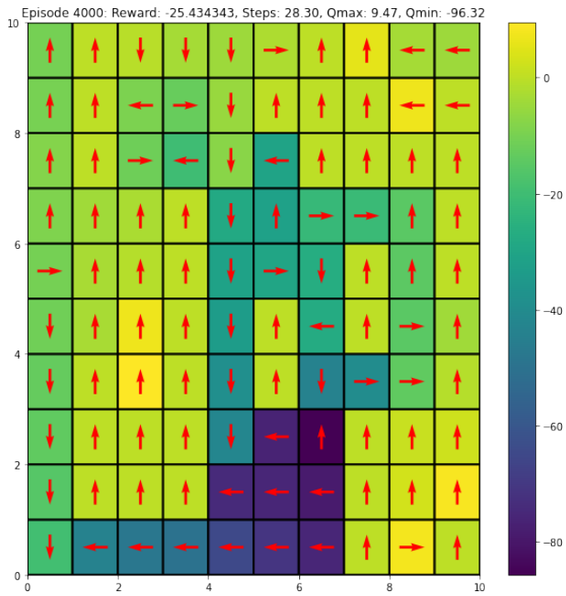


(b) State occupancy heatmap

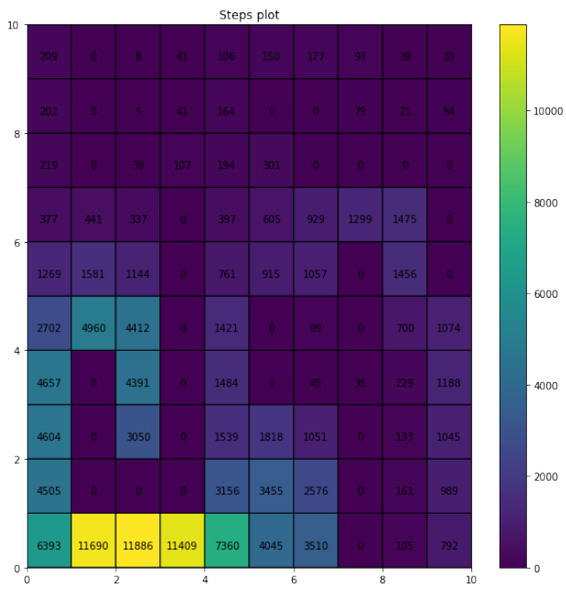


(c) Reward vs. episodes

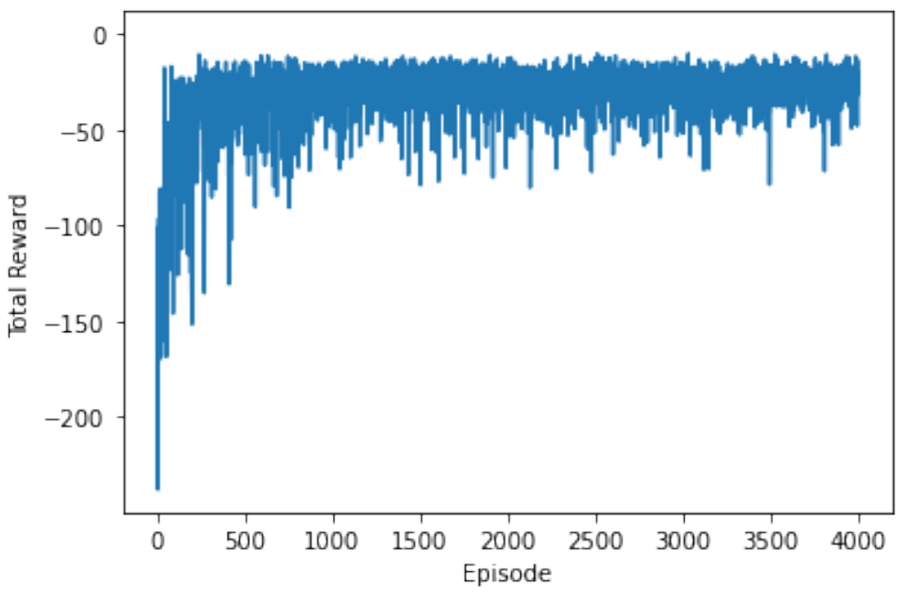
Figure 31: sarsa for config 15



(a) Q value heatmap



(b) State occupancy heatmap



(c) Reward vs. episodes

Figure 32: sarsa for config 16

2.1 Observations

Here we describe our learnings from the experiments:

1. Q Learning agent goes to the upper goal when wind is false and it is at starting state (3,6) because that is the closest goal. When wind=true, it goes to the lower right goal because it can take the rightmost column to reach the goal without being moved left by the wind.When starting in state (0,4), the agent prefers the bottom-left goal because it is the closest for both SARSA and Q Learning.

2. Softmax exploration strategy works well with Q Learning, whereas ϵ-greedy works better with SARSA.

3. SARSA agent avoids paths which are closer to restart states. It is evident that SARSA tries to stay away from those states with high negative reward, whereas, Q Learning agent tries to maximize the reward even if it has to venture close to the restart states.

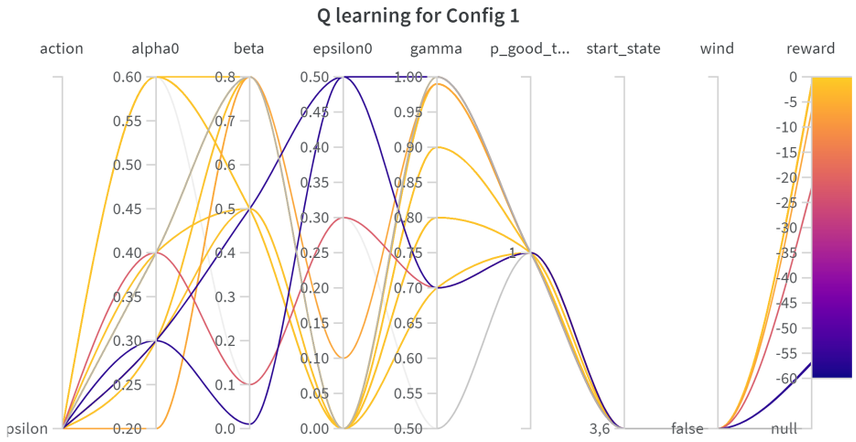
4. High exploration is not helping in this environment. It could be because of the small grid size and high number of obstacles.

5. SARSA favours the left-most goal over the other two goals because there is less chance to get stuck in a restart state when using the bottom-most rows to reach that goal.It prefers a safer policy as compared to Q learning under both wind settings.

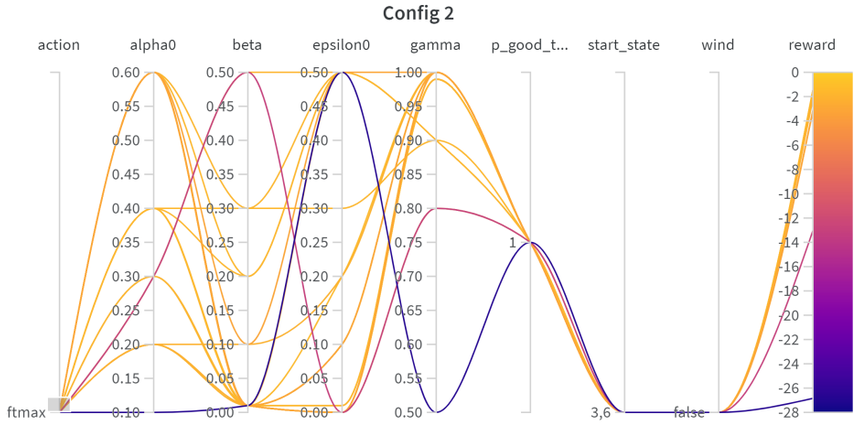
6. Upon adding further randomness to the environment in the form of p=0.7, the returns are not as high as with p=1.0. Both the algorithms are still able to reach one of the goals but they take a lot more steps to do so, as is expected. SARSA still tries to avoid getting closer to restart states.

3 Hyperparameter Tuning

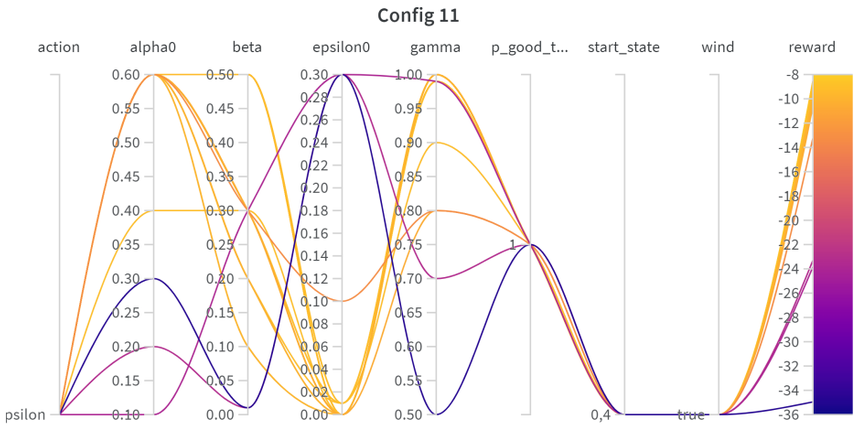
Here we present some hyperparameter tuning plots from wandb.ai.



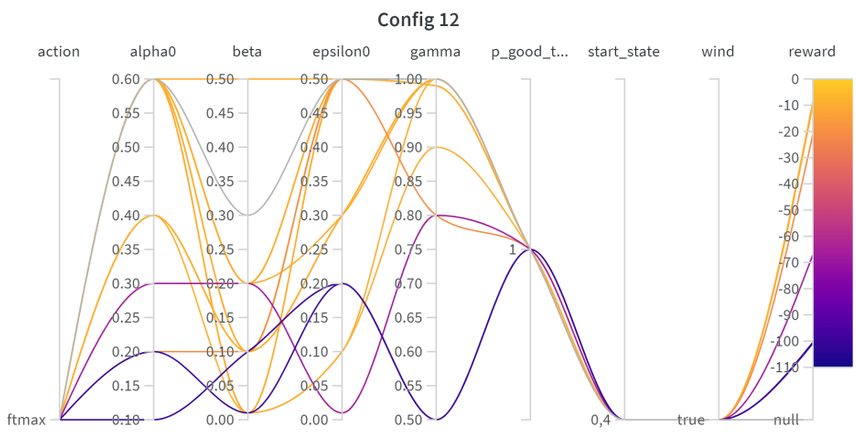
(a) Q Learning with config 1



(b) Q Learning with config 2

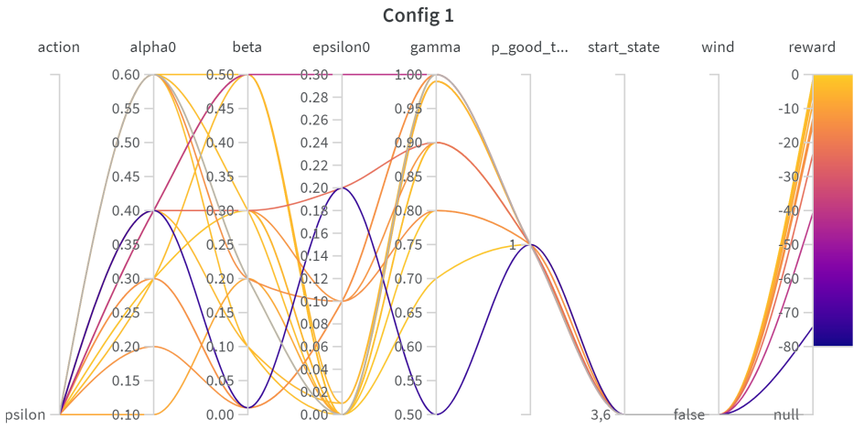


(c) Q Learning with config 11

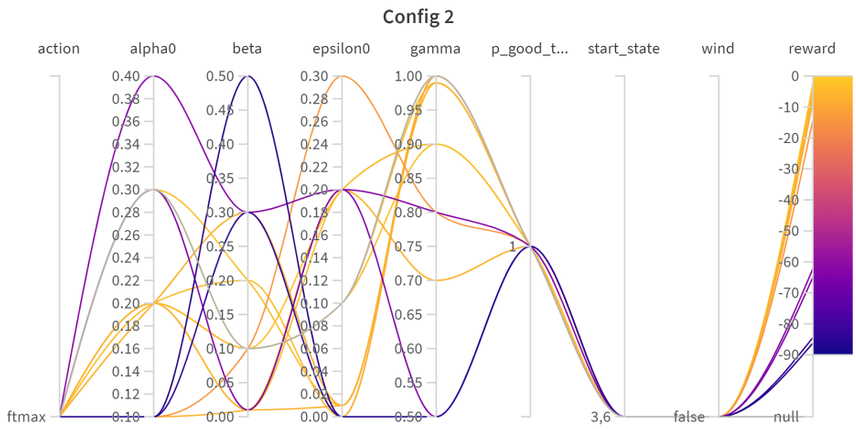


(d) Q Learning with config 12

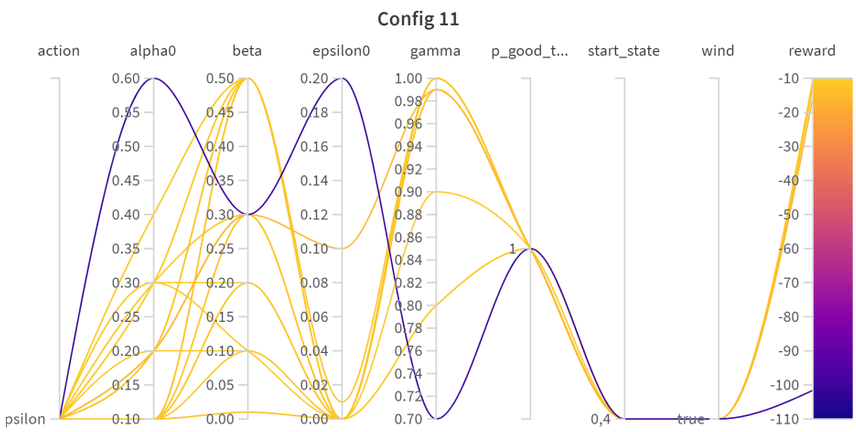
Figure 33: Tuning Q learning using WandB



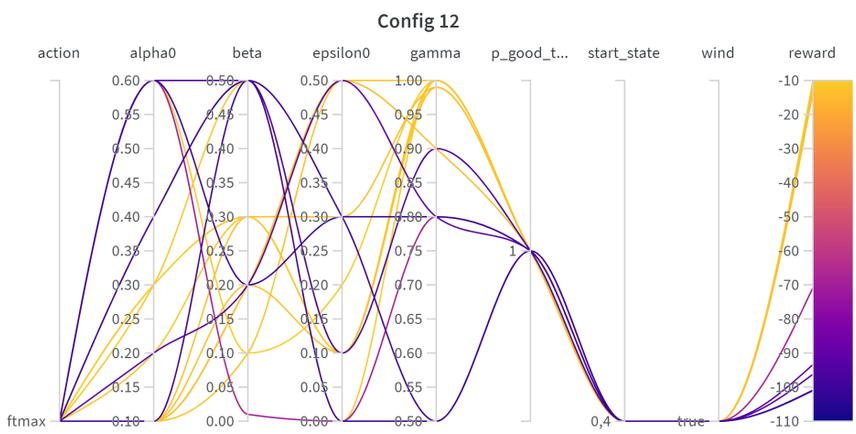
(a) SARSA with config 1



(b) SARSA with config 2



(c) SARSA with config 11



(d) SARSA with config 12

Figure 34: Tuning SARSA using WandB