CS6700 Reinforcement Learning
Programming Assignment-1

K V Vikram (CS19B021), Kanishkan M S (ME19B192)

February 24 2023

Introduction

This assignment familiarized us with two popular Temporal Difference Learning algorithms, SARSA and Q-Learning. We implemented these algorithms and ran experiments using them on multiple gridworld environments. We also learned how to use wandb for making sweeps to get optimal hyper-parameter configurations.

Totally, 32 configurations $\{2\times$algorithms$, 2\times$action selection policies$, wind =$ True/False$, start = (0,4)/(3,6)$ and action fail probability $p = 1.0/0.7\}$ were analyzed and the best hyper-parameters for them were found. For every configuration, we used the below process to tune the hyper-parameters.

Wandb Sweep Technique

For each configuration, a wandb sweep is made that runs 729 different hyper-parameter combinations through a grid search. For each hyper-parameter combination, we do 5 runs of the learning algorithm with 2000 episodes (for epsilon greedy action selection) or 4 runs of the algorithm with 1600 episodes (for softmax action selection - as it is slower) to calculate the below metrics.

- **average test steps:** We compute this metric as the average number of steps to finish an episode during the test phase. It is averaged over 100 different episodes to account for environment stochasticity.
- **average test reward:** We compute this metric as the average reward earned in an episode during the test phase. It is averaged over 100 episodes. This is our **primary ranking metric.**
- **average train steps:** We compute this metric as the average number of steps to finish an episode during the train phase. It is averaged over all episodes of all runs.
- **average train reward:** We compute this metric as the average reward earned in an episode during the train phase. It is averaged over all episodes of all runs. This is a form of regret measure. This is our **secondary ranking metric.**

We will only show the results of the top 10 hyper-parameter combinations for each configuration in the report. However, the **results of all** $32 * 729 = 23328$ **tests** can be accessed at our team's wandb dashboard. The relevant projects are `rl_a1_configX` where $1 \leq X \leq 32$ is the configuration number as in the report.

Configuration Description

We will first give the parallel co-ordinates plot for all the 729 runs with the top 10 combinations (based on primary and secondary ranking metrics) in yellow. The values of the metrics for these top 10 hyper-parameter combinations are also shown. Out of these 10, we take the top 4 combinations to plot comparative rewards curve and comparative steps curves. From these curves, we will decide the best hyperparameter combination.

For this best combination, we make the below plots.

1. average reward per episode plot
2. average steps per episode plot
3. heatmap of grid with Q-values with arrows showing the optimal action
4. heatmap of grid with state visit counts

We do 3 runs with 2000 episodes to get stable curves. The Q table from these 3 runs that performs the best over 100 test episodes is defined as the best Q table. This is used to make the heat maps. Additionally, we also provide 2 renderings of the agent's trajectory in the gridworld when it acts greedily using the best Q table. Finally, we give reasons to justify the above plots and behaviour.

The Gridworld

Gridworld with state labels

Configuration 1

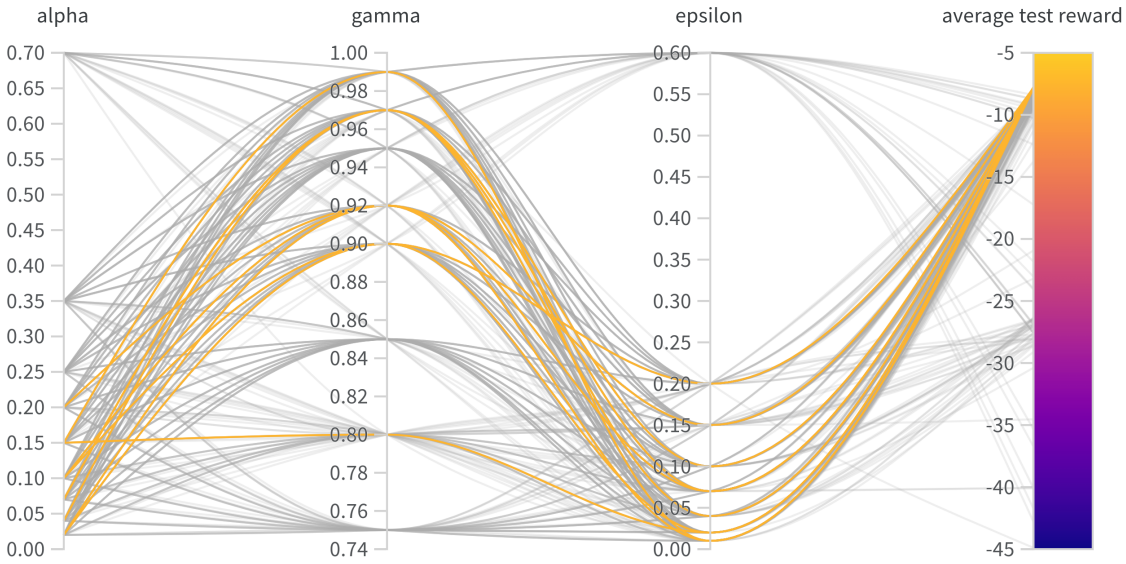
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 1.0

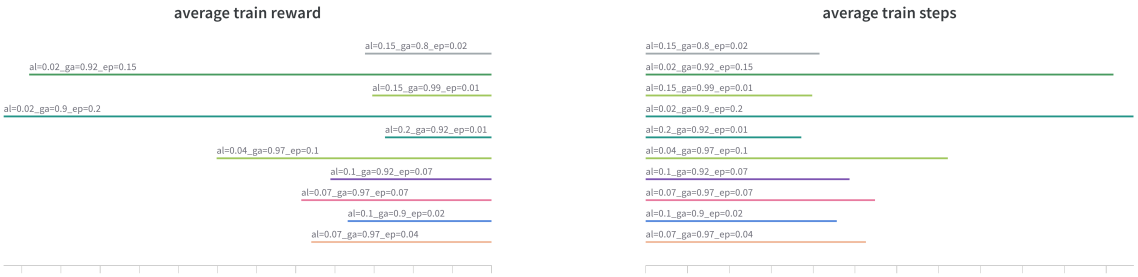
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



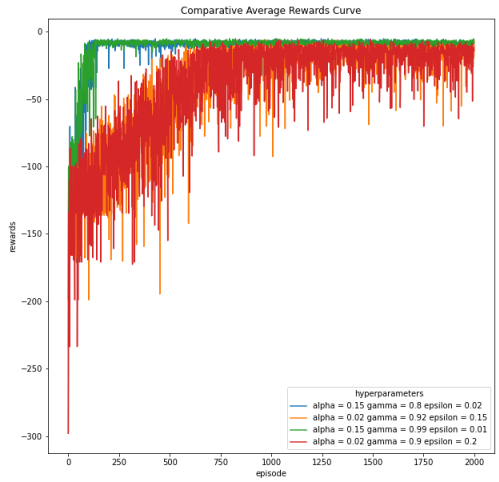
Recorded Metrics

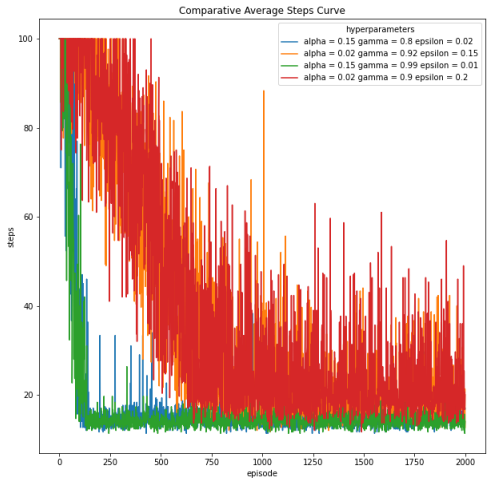


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



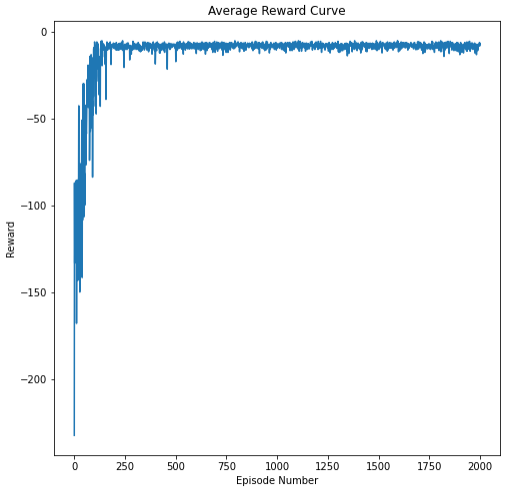


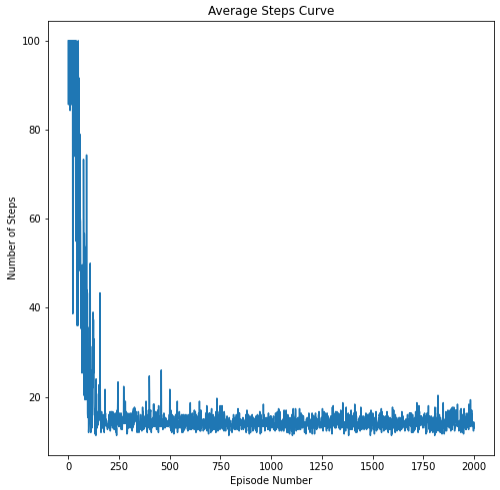
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.15, 0.99, 0.01)$ are the best hyper-parameters according to our sweep.

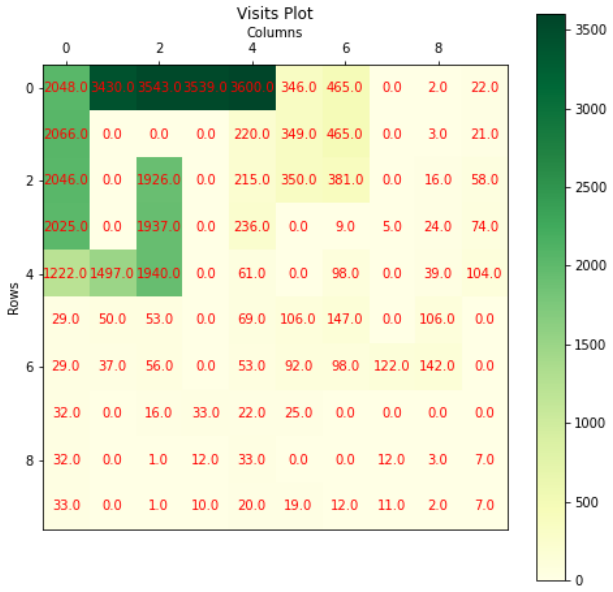
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

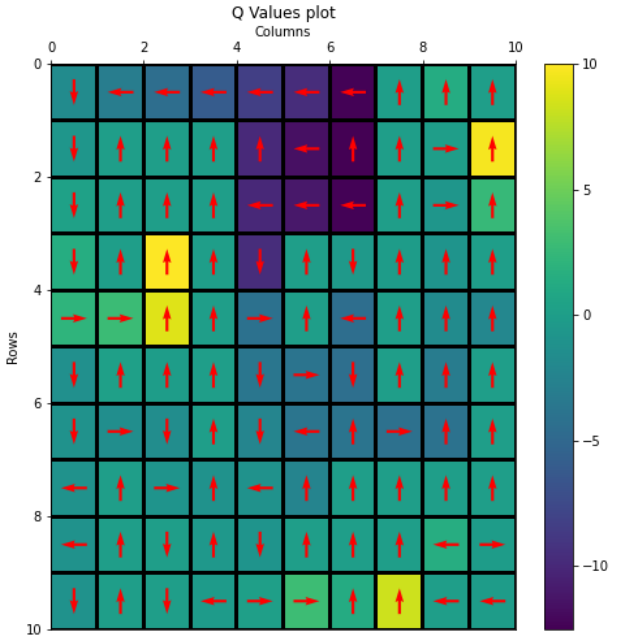




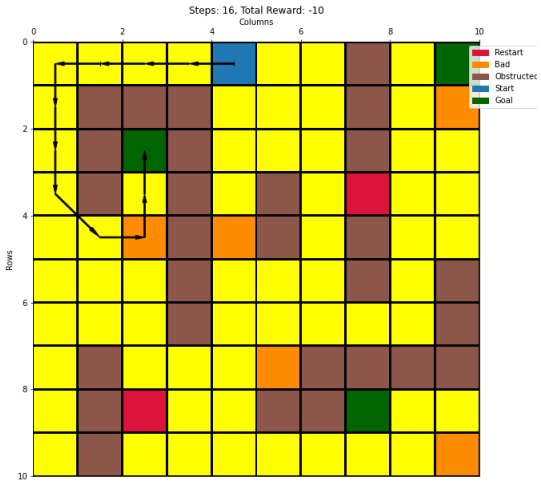
Heatmap with state visit counts

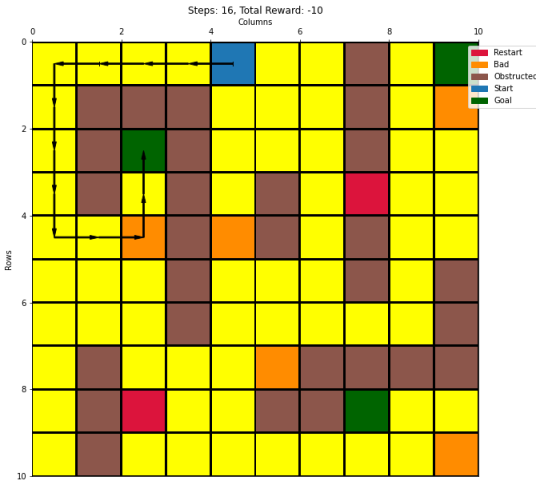


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- In this configuration, the only source of stochasticity is the wind.
- The nearest goal state is at $(2, 2)$ for this start state. The wind will be against the agent when moving along the first row. We can see that the rightward wind helps the agent move diagonally in rendering 1 and reduces the negative reward earned.

Configuration 2

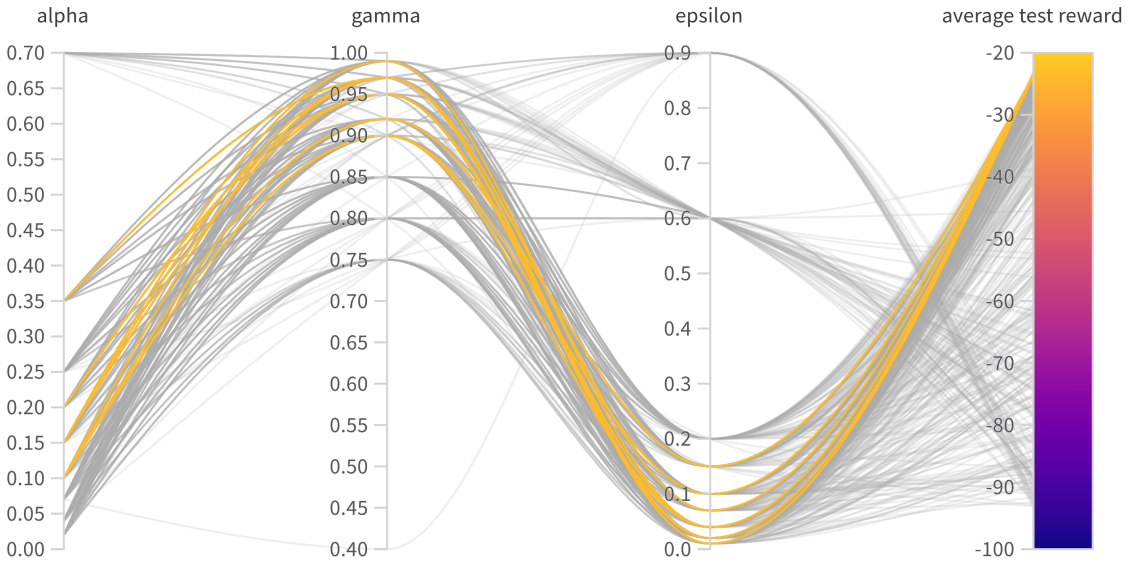
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 0.7

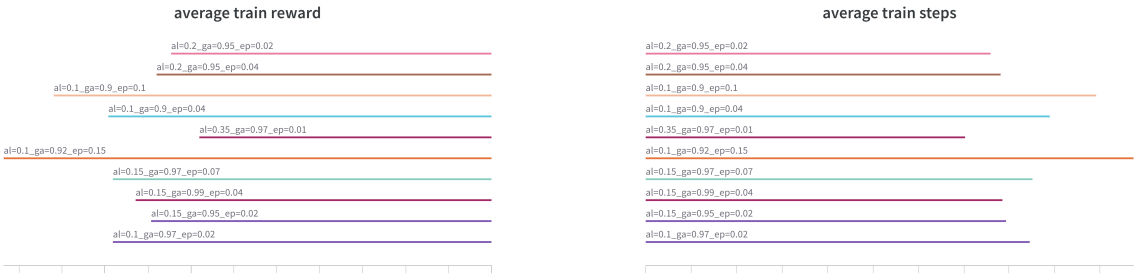
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

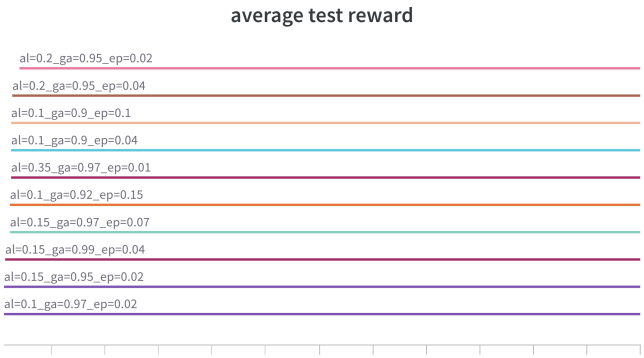
Parallel Co-ordinates Plot



Recorded Metrics

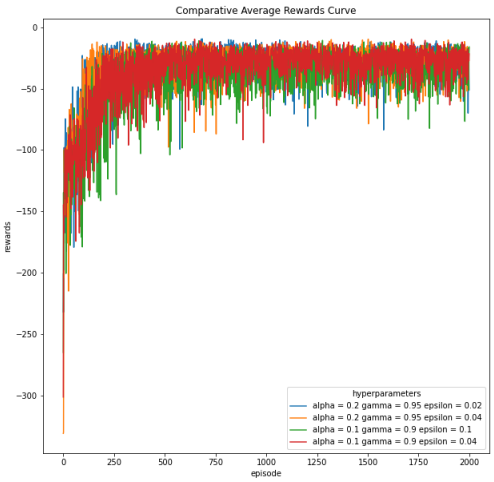


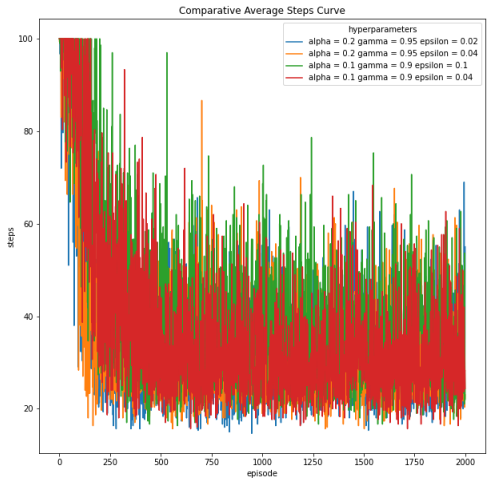
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



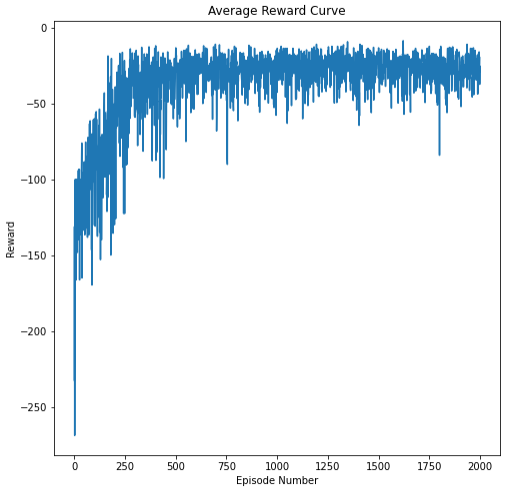


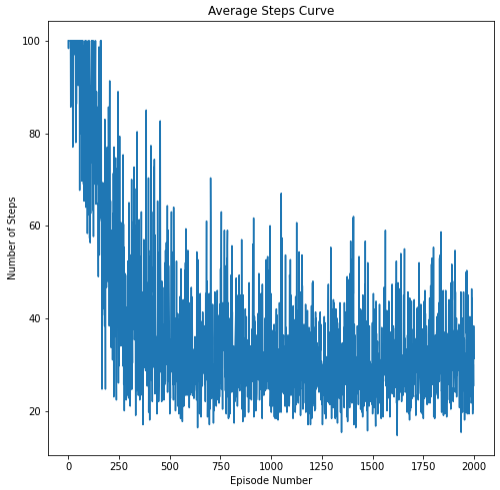
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.1, 0.9, 0.04)$ are the best hyper-parameters according to our sweep.

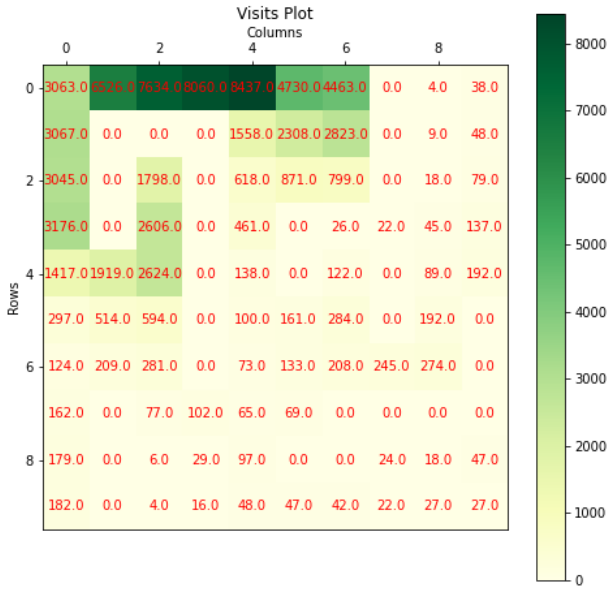
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

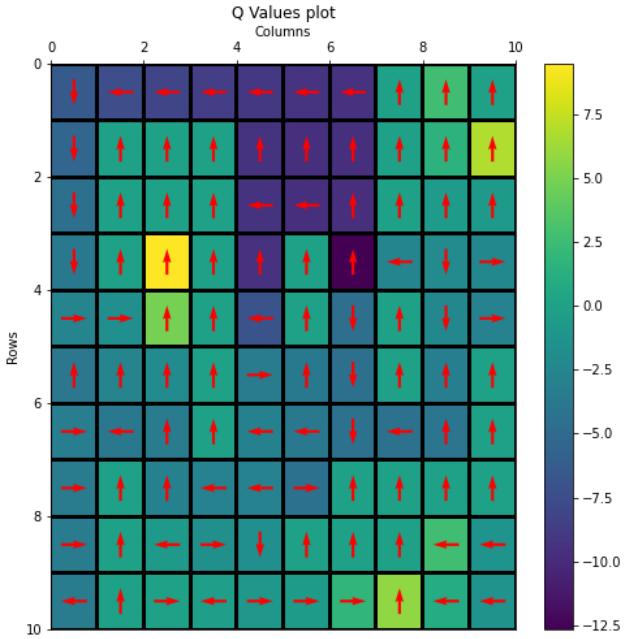




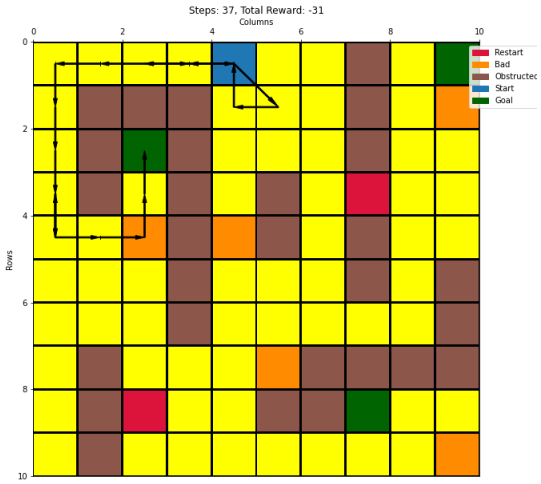
Heatmap with state visit counts

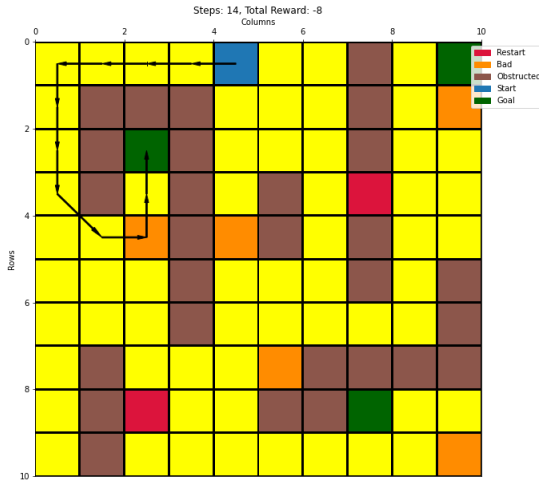


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- In this configuration, as $p = 0.7$, the agent may move to the west or the east of the chosen action (we call this event *action failure*). Here, the chosen action is considered as north. This causes large fluctations in the reward and steps curve.
- We can see that the wind here will sometimes support the agent and sometimes oppose it in its way to $(2, 2)$.
- Once again, the agent chooses the nearest goal state.

Configuration 3

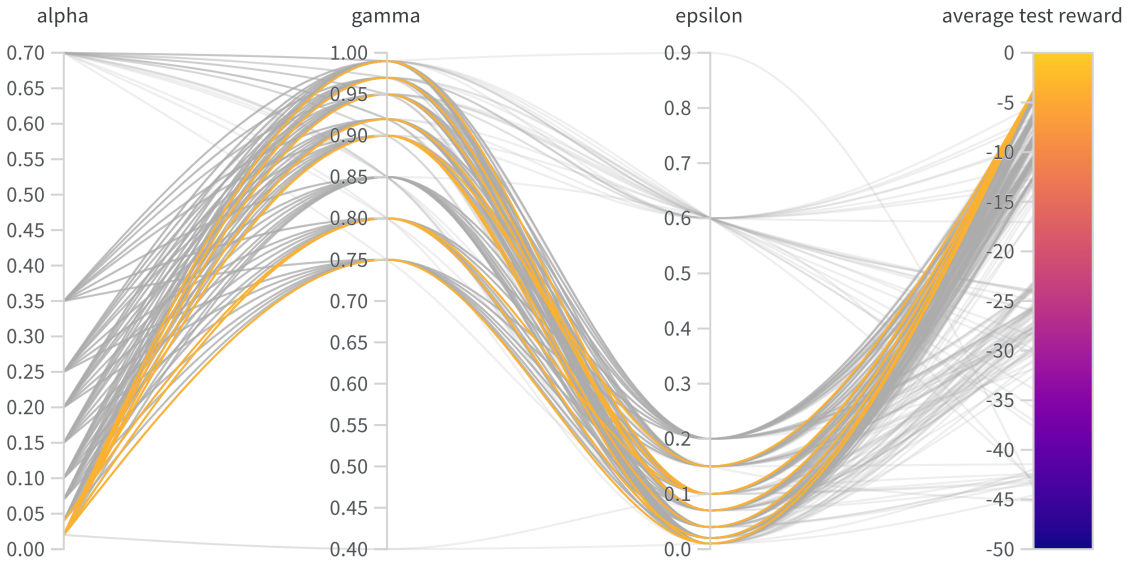
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 1.0

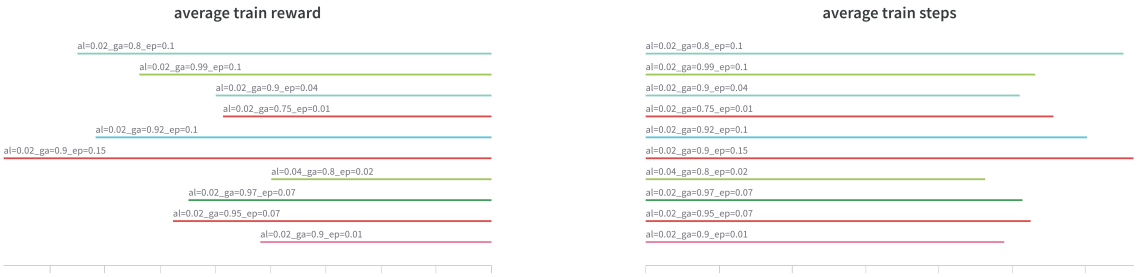
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

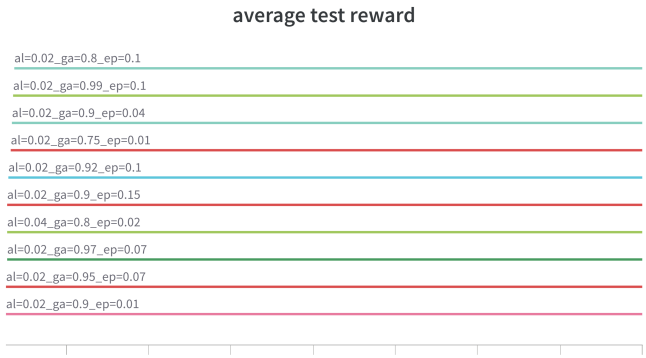
Parallel Co-ordinates Plot



Recorded Metrics

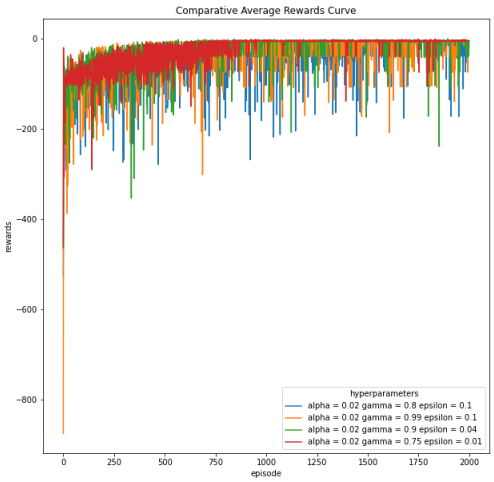


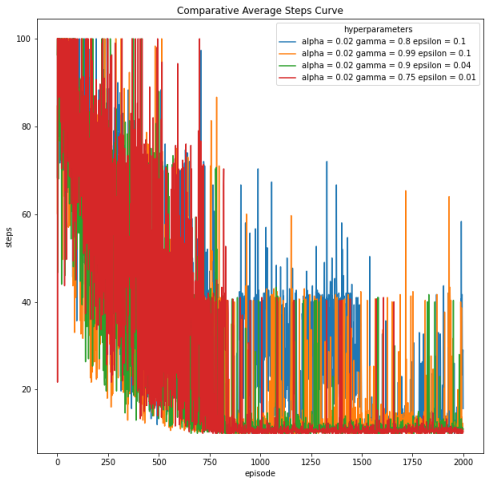
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



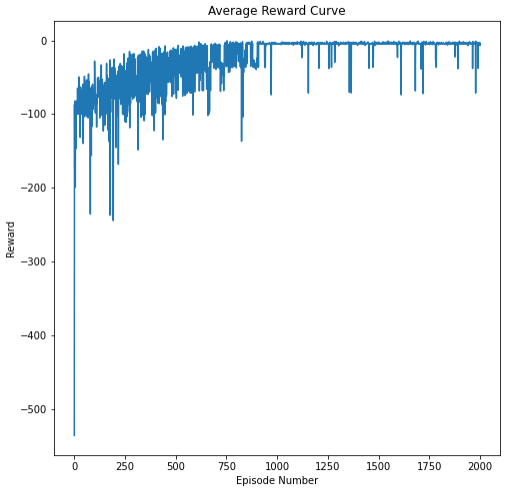


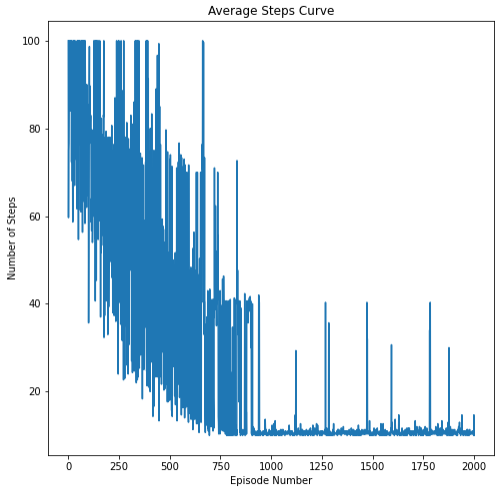
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.02, 0.75, 0.01)$ are the best hyper-parameters according to our sweep.

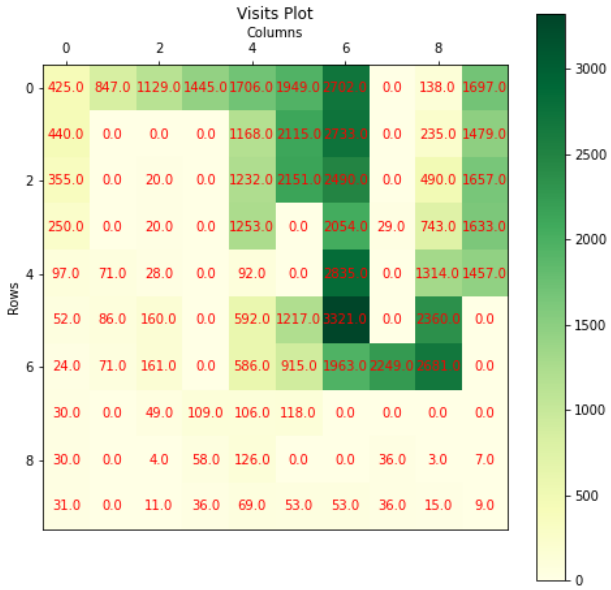
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

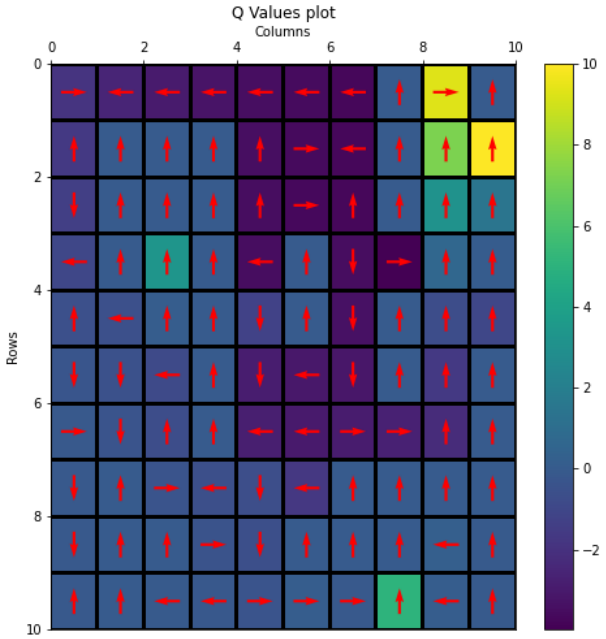




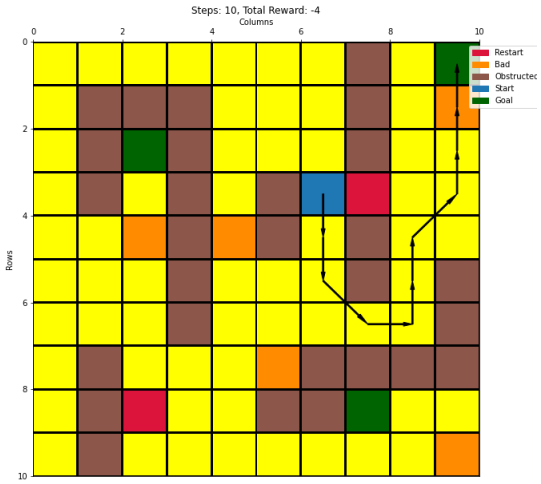
Heatmap with state visit counts

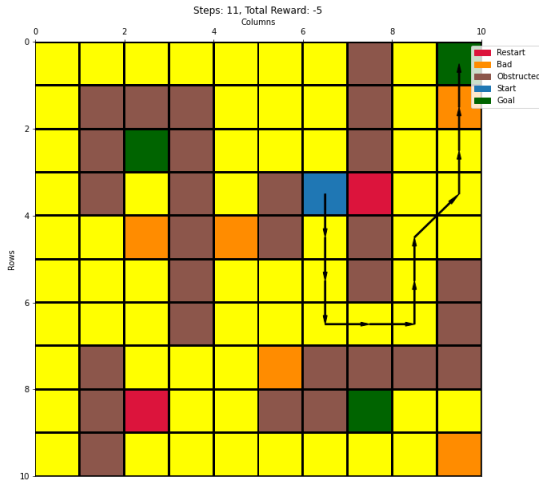


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present. Even though the goal states at $(0, 9)$ and $(8, 7)$ are equidistant from the start state, the agent biases towards $(0, 9)$ because of the wind.
- Because of the wind, the agent gets pushed to the last column and is forced to go through the bad state below the goal.
- The agent may be pushed into the restart state next to the start state and get a large negative reward. This has not happened in the above renderings but it may happen.

Configuration 4

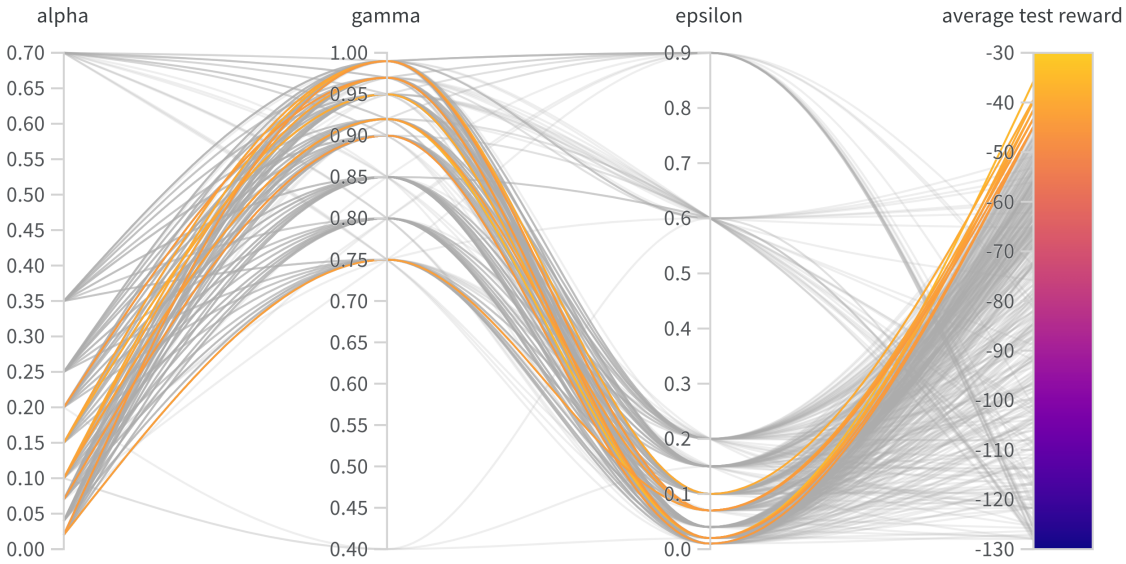
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 0.7

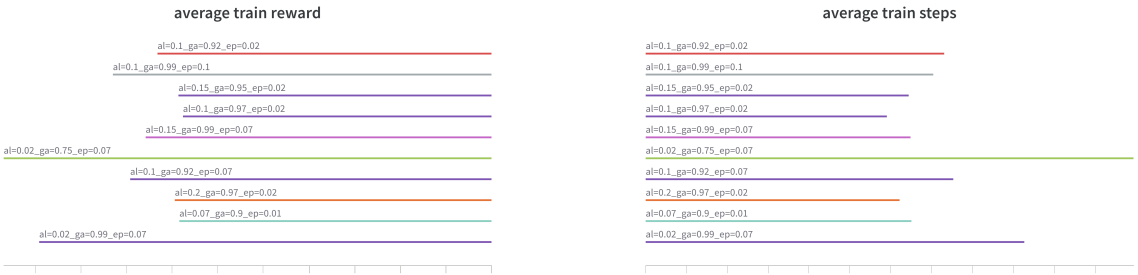
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

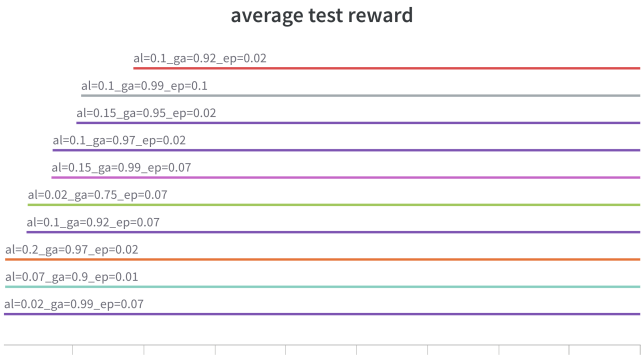
Parallel Co-ordinates Plot



Recorded Metrics

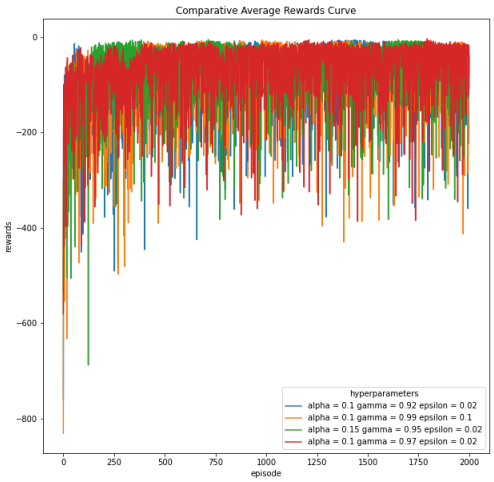


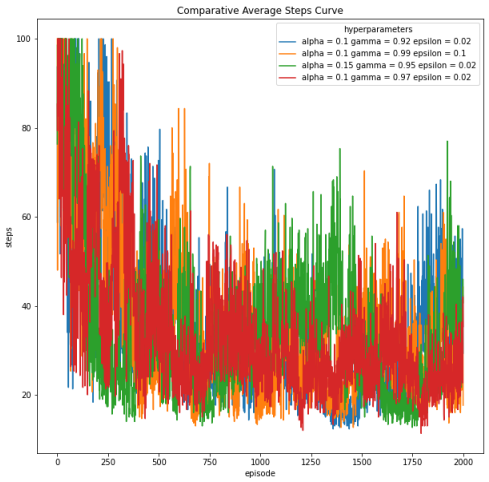
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



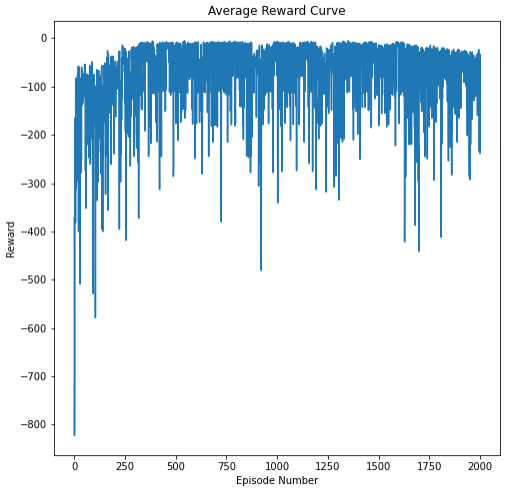


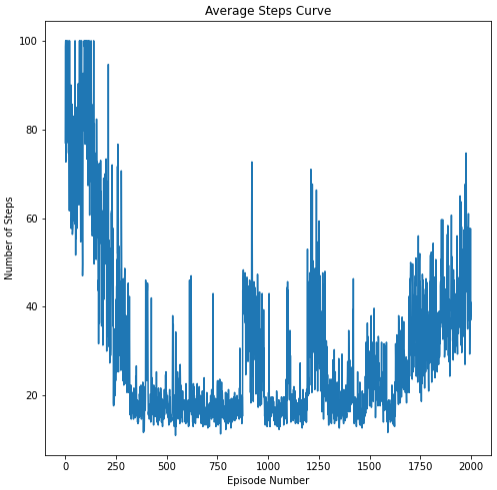
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.1, 0.97, 0.02)$ are the best hyper-parameters according to our sweep.

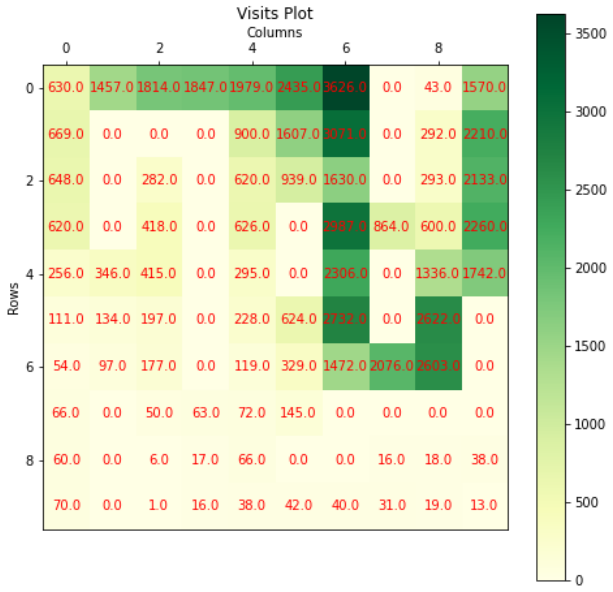
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

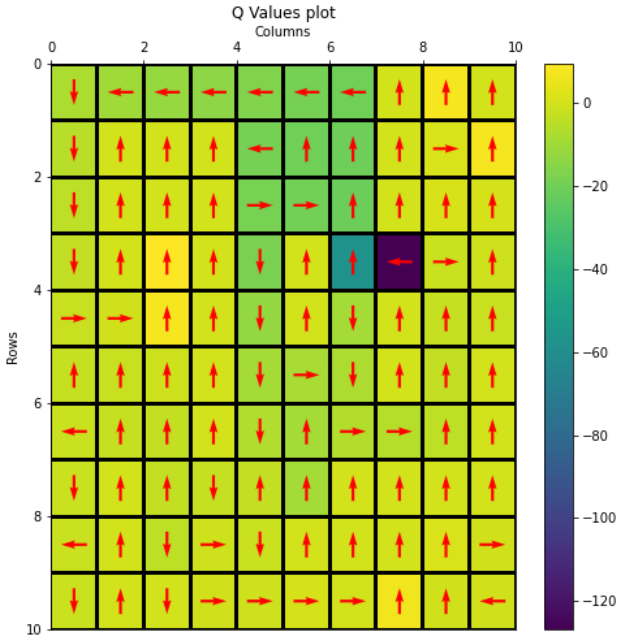




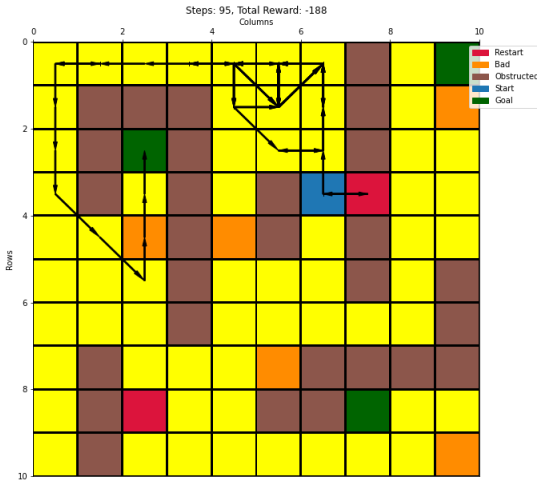
Heatmap with state visit counts

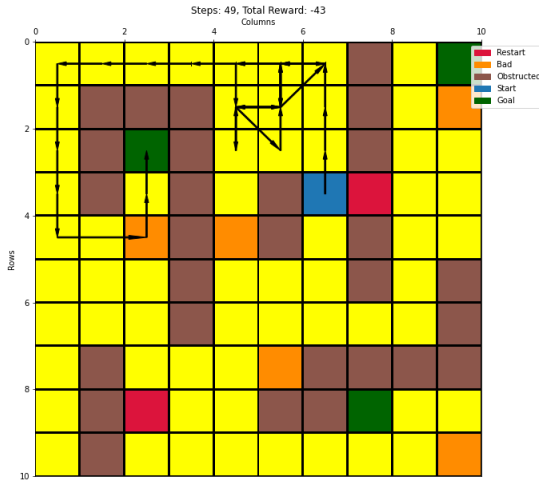


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present along with the possibility of action failure. This is the reason for the large fluctations in the reward and steps curves once again.
- Because of the wind, the agent sometimes moves into the restart state next to the start state resulting in a very high negative reward, as in the renderings.
- The action failure probability also causes the agent to move in loops occasionally as seen above.

Configuration 5

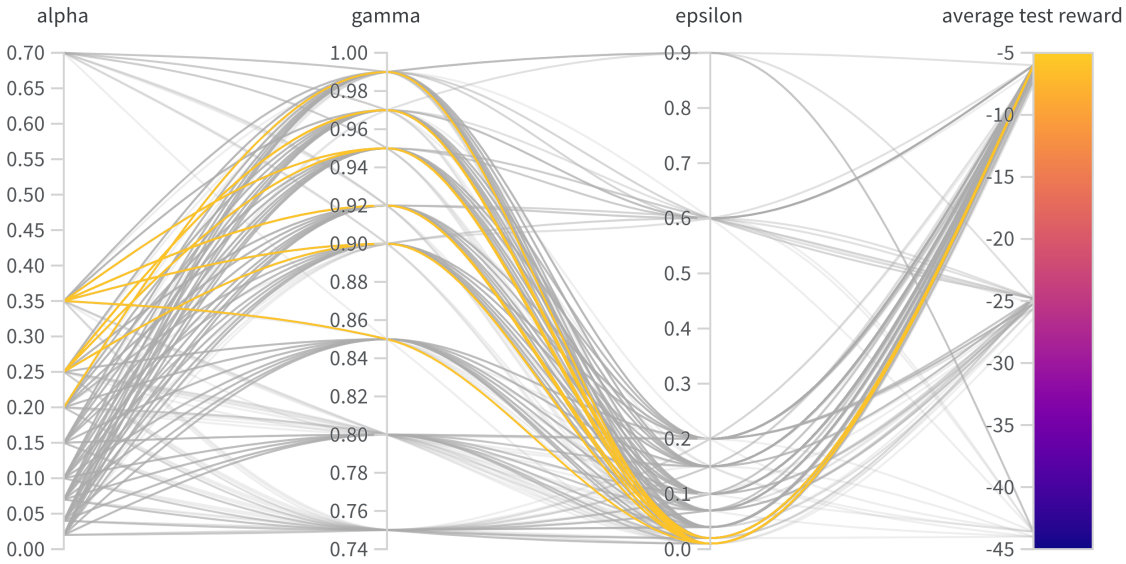
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 1.0

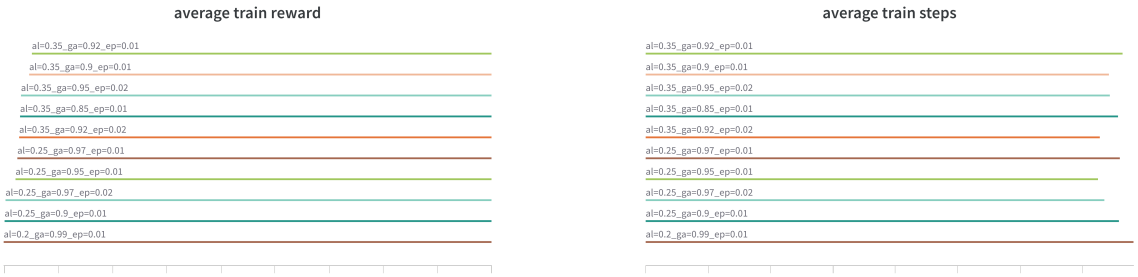
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

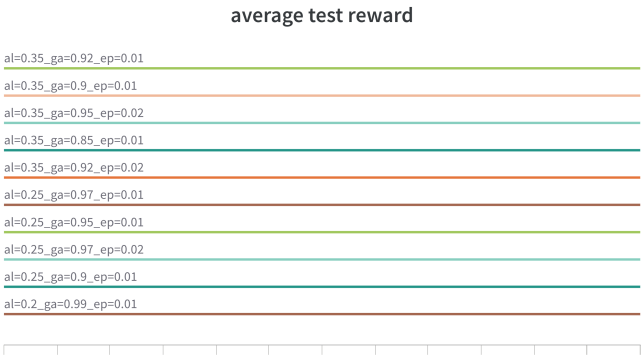
Parallel Co-ordinates Plot



Recorded Metrics

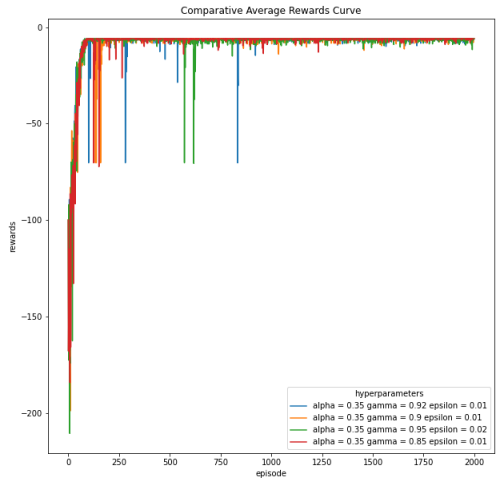


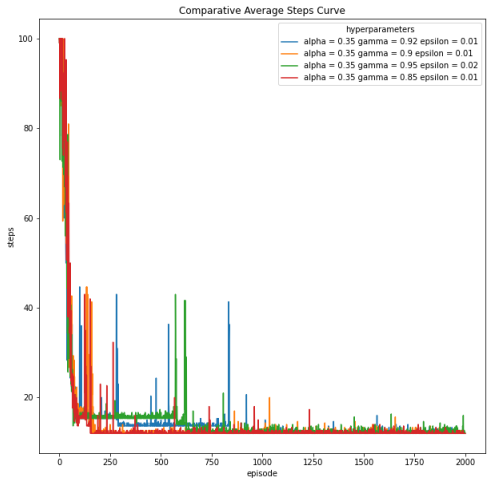
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



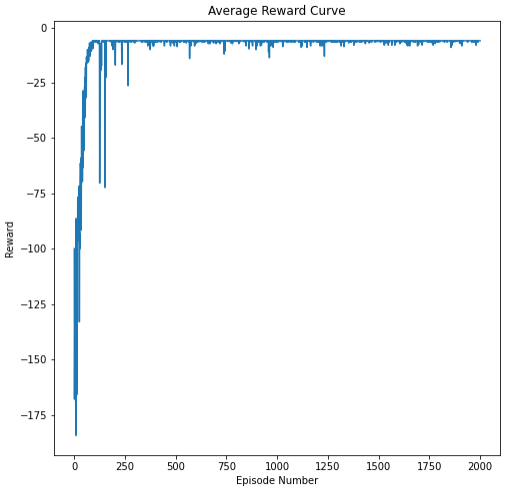


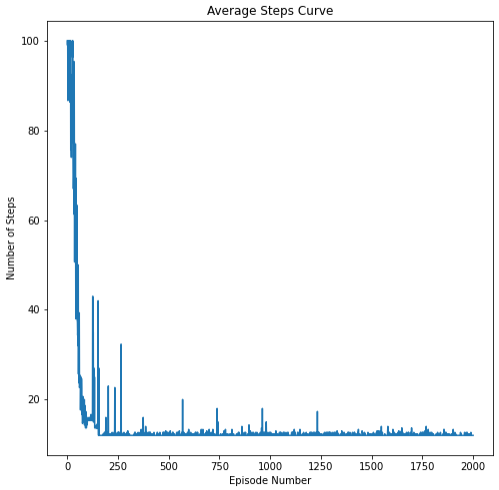
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.35, 0.85, 0.01)$ are the best hyper-parameters according to our sweep.

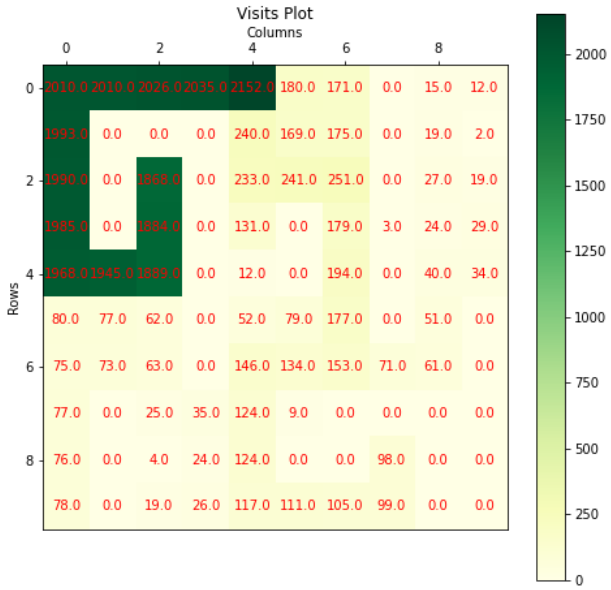
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

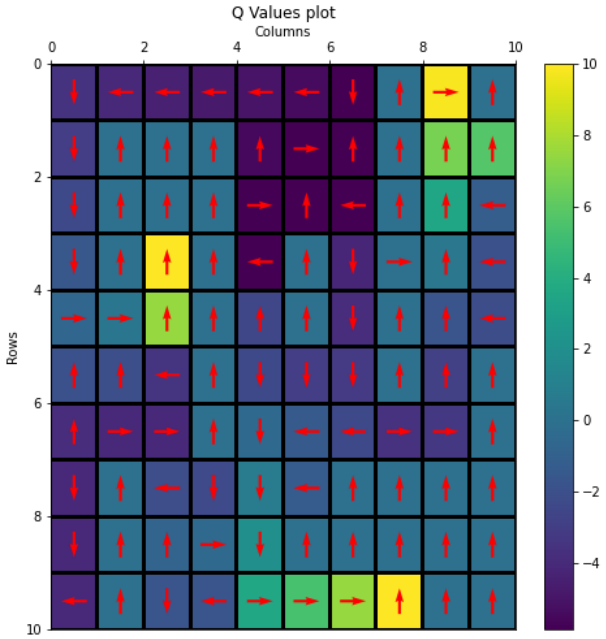




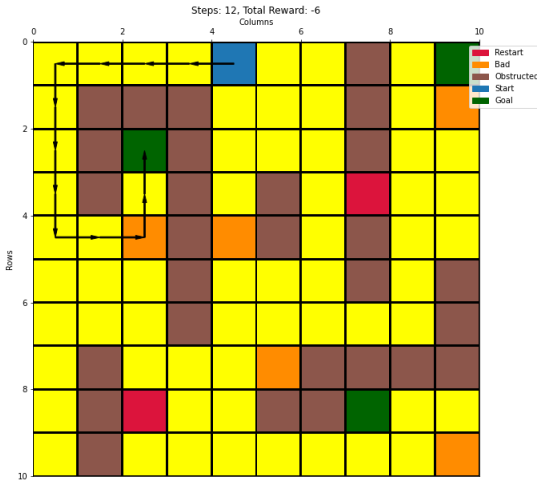
Heatmap with state visit counts

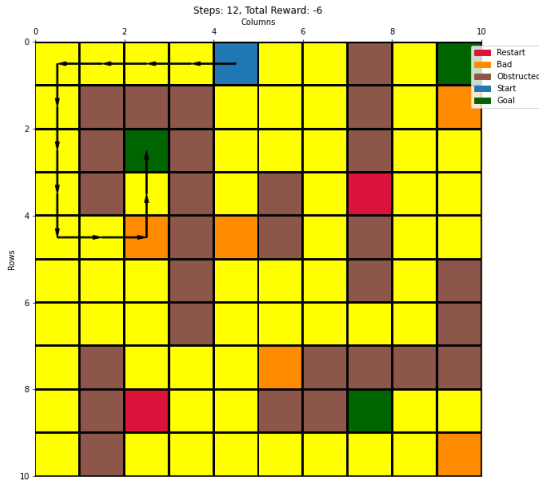


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path with a bad state in the way as it is optimal (in terms of reward earned).

Configuration 6

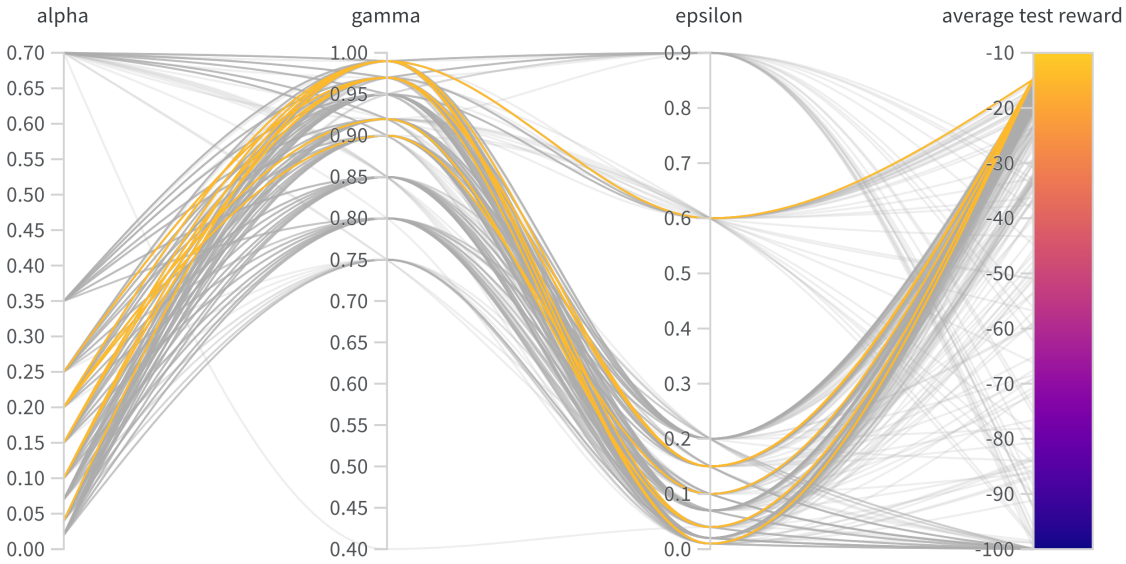
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 0.7

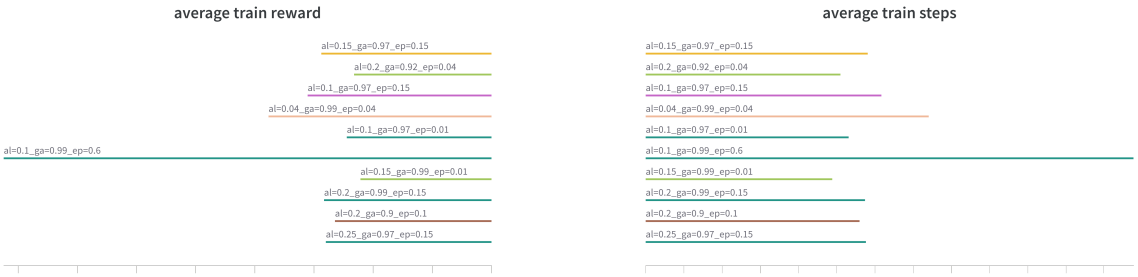
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

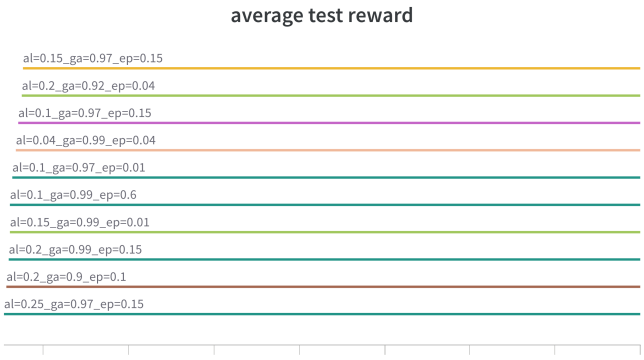
Parallel Co-ordinates Plot

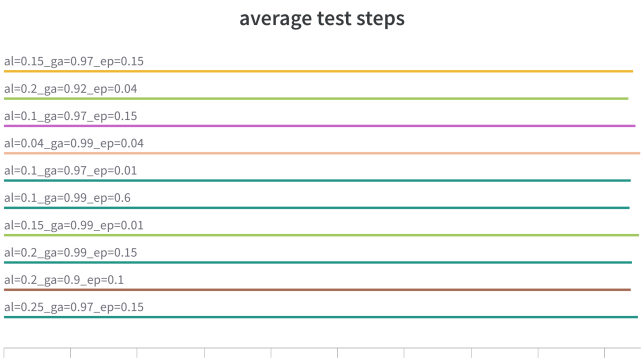


Recorded Metrics



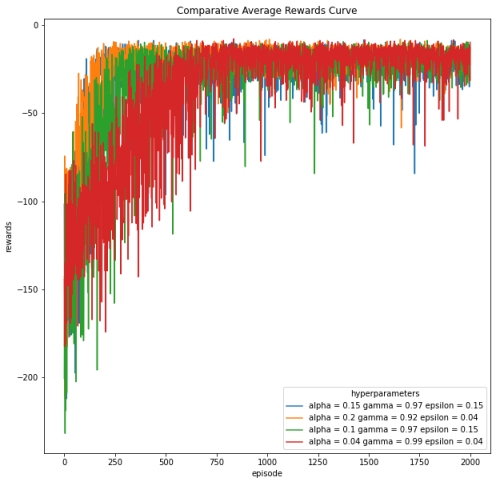
Train Metrics

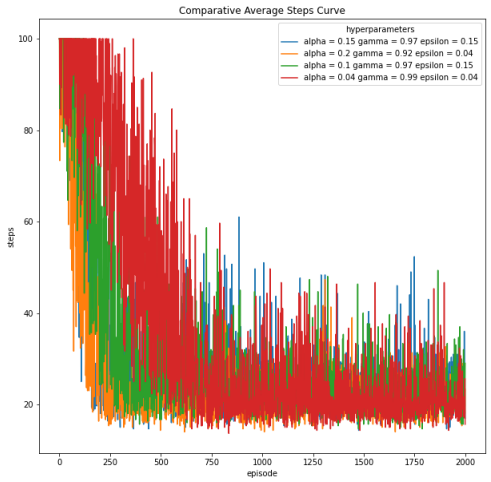




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



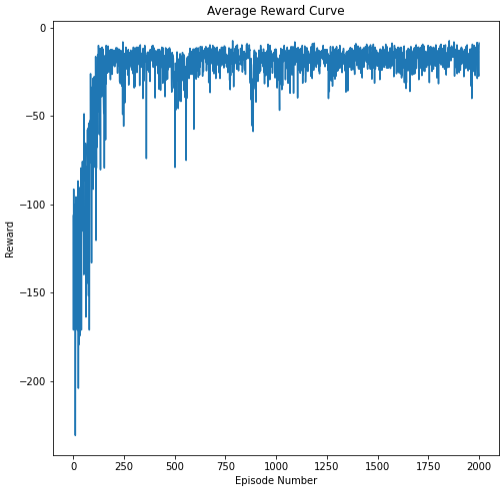


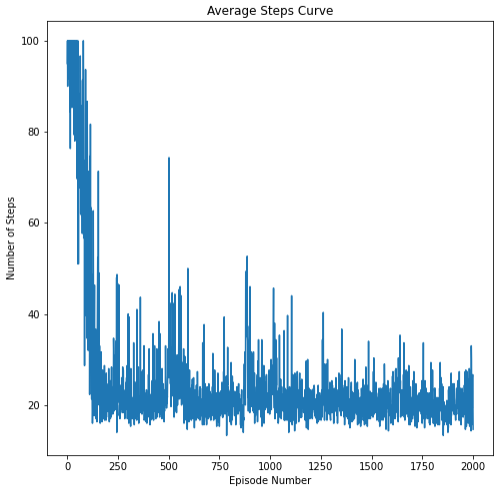
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.2, 0.92, 0.04)$ are the best hyper-parameters according to our sweep.

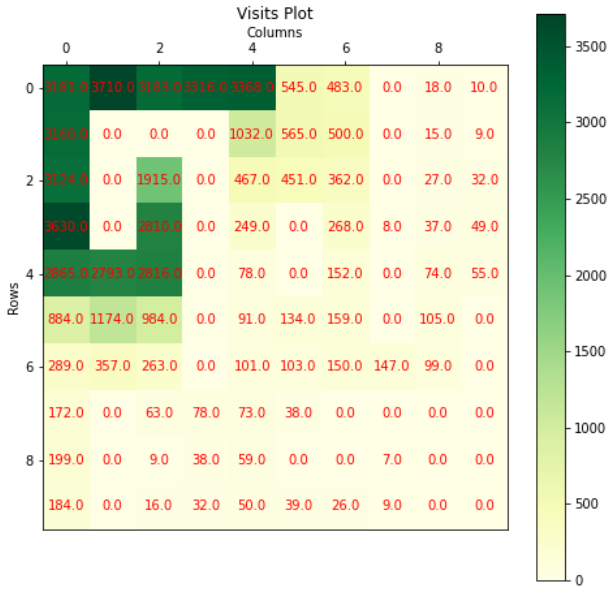
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

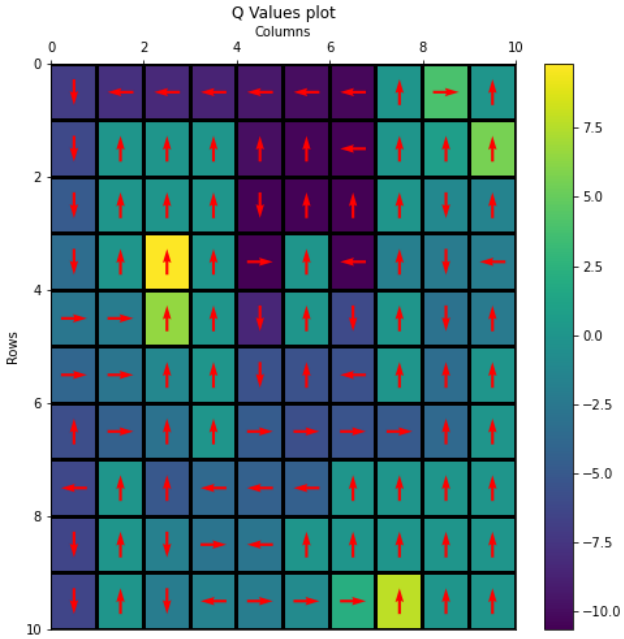




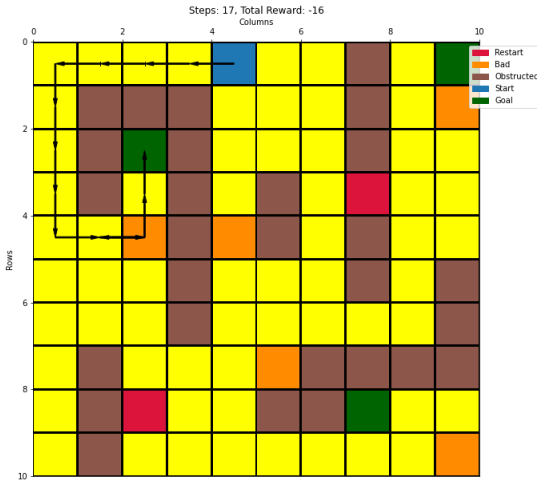
Heatmap with state visit counts

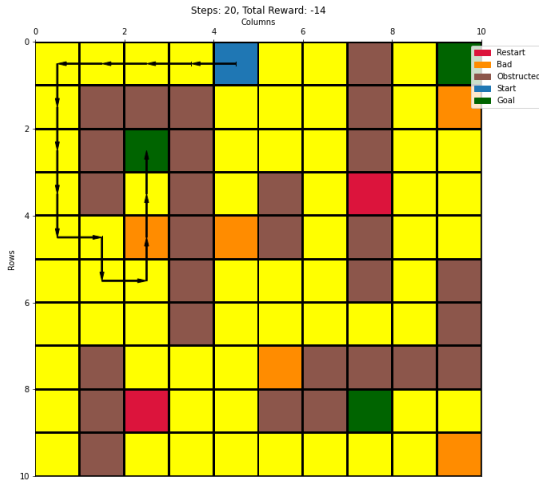


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but there is action failure.
- The agent always tries to take the path to $(2, 2)$ directly. But action failure may increase the number of steps.

Configuration 7

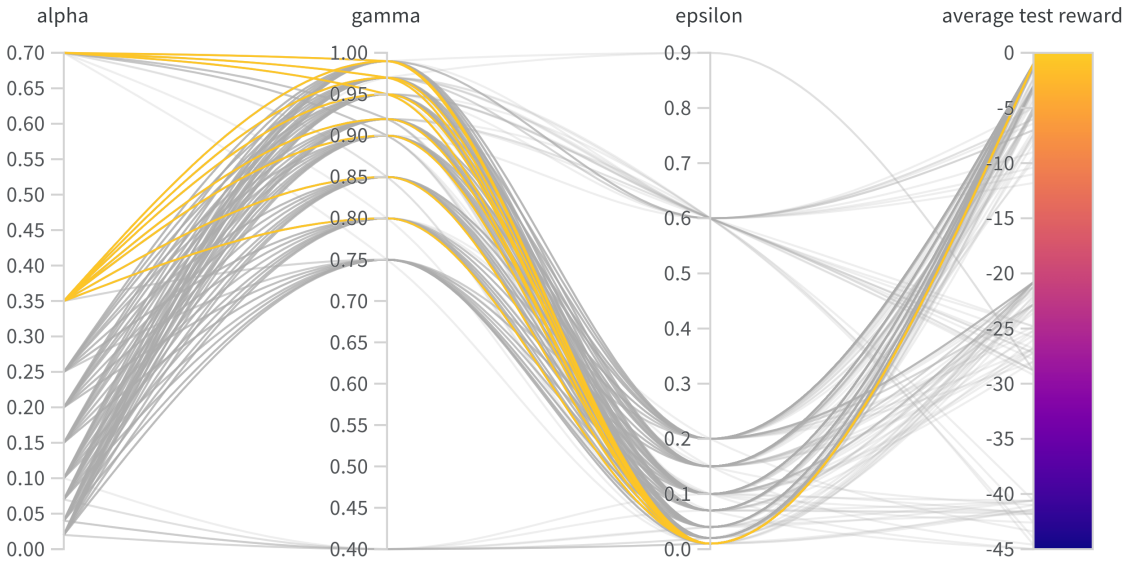
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 1.0

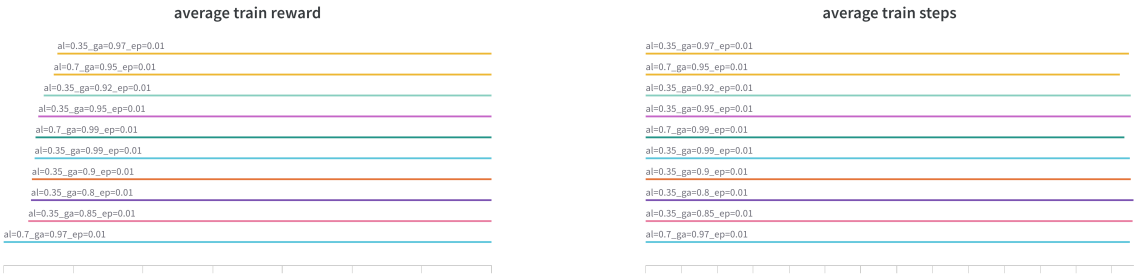
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



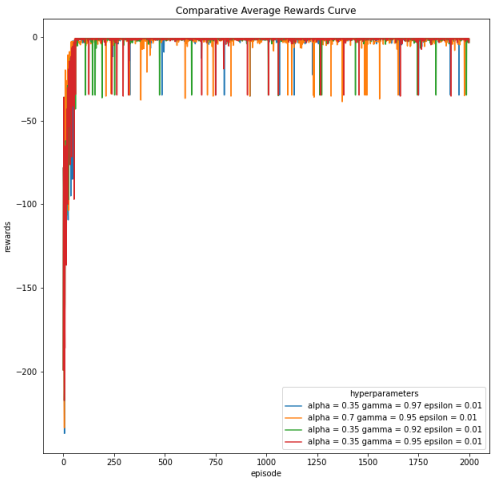
Recorded Metrics

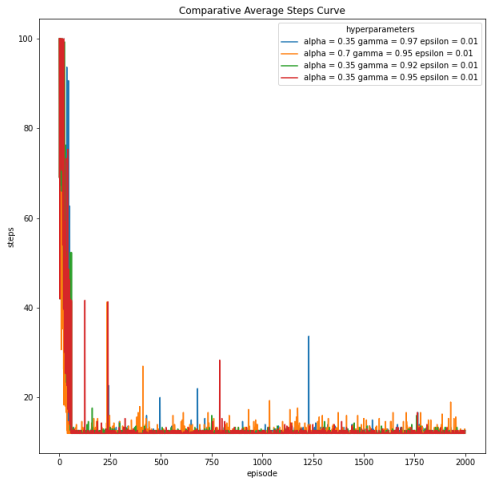


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



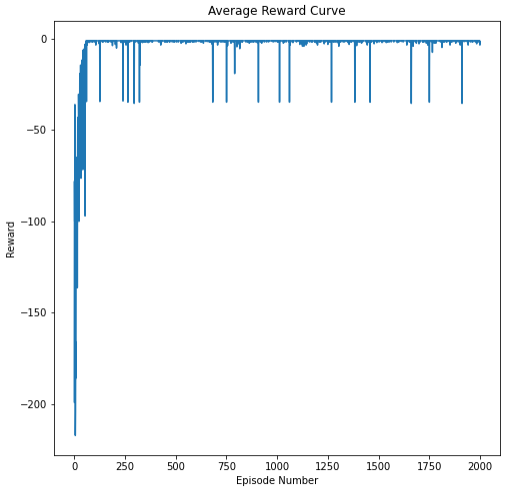


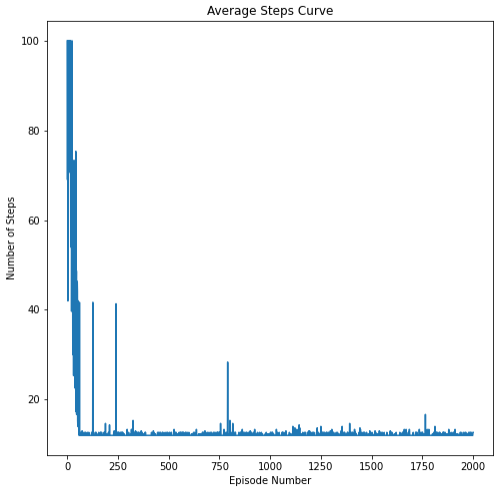
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.35, 0.95, 0.01)$ are the best hyper-parameters according to our sweep.

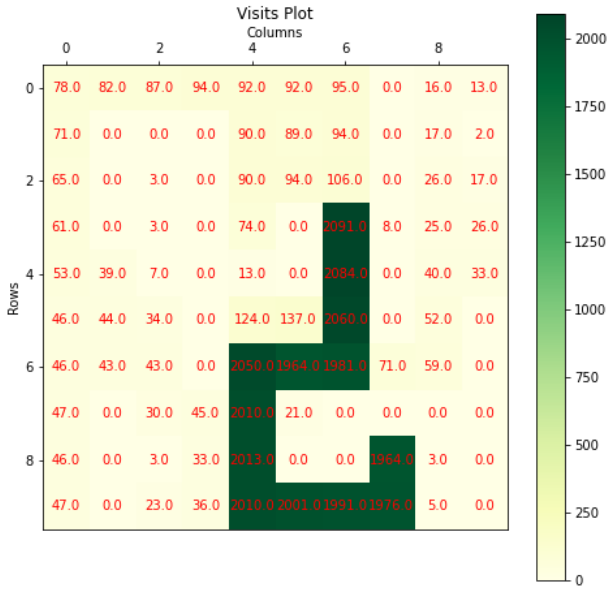
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

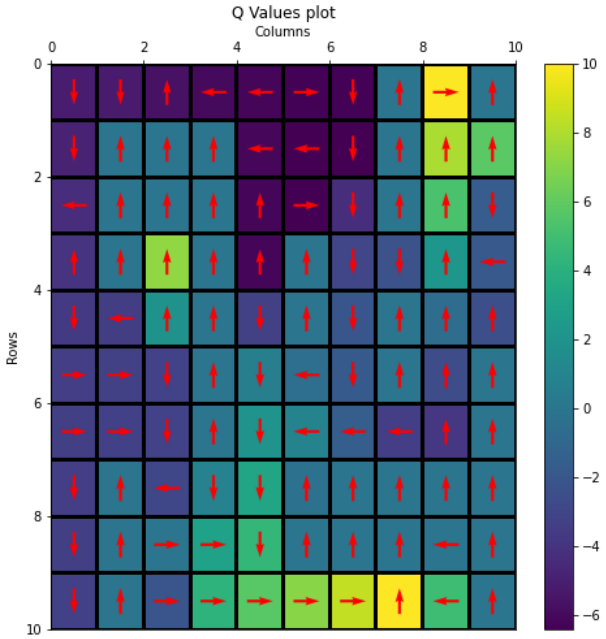




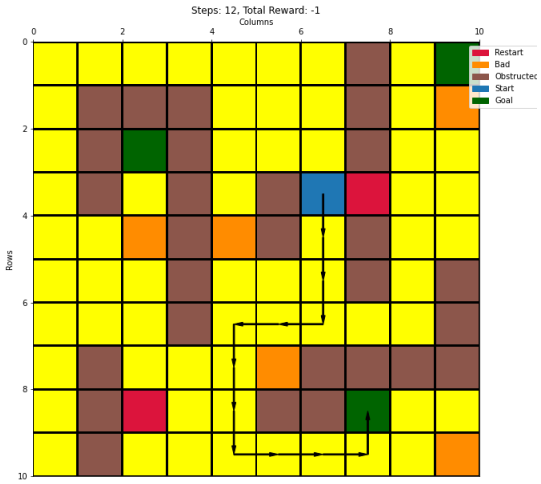
Heatmap with state visit counts

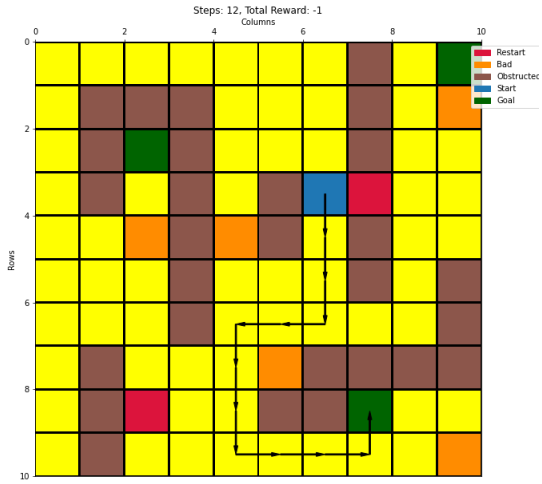


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path to $(8, 7)$.

Configuration 8

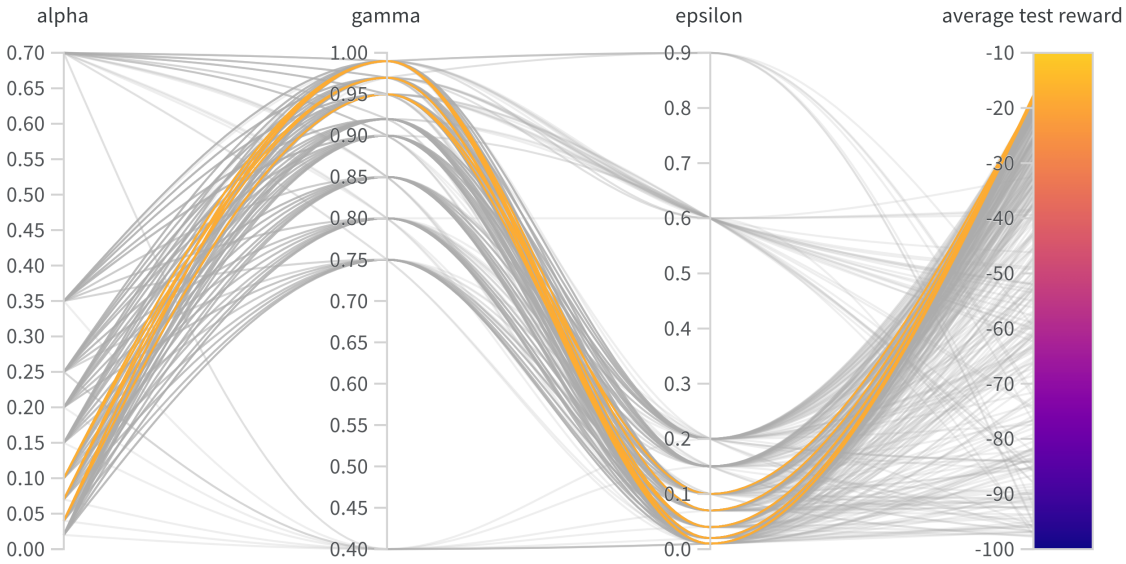
Configuration Description

- **learning -** SARSA
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 0.7

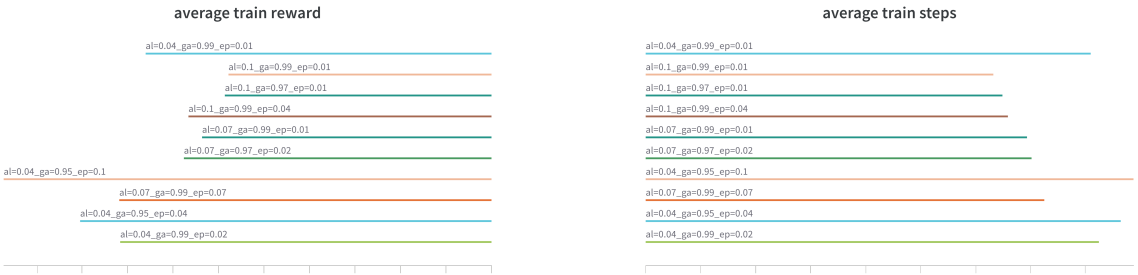
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



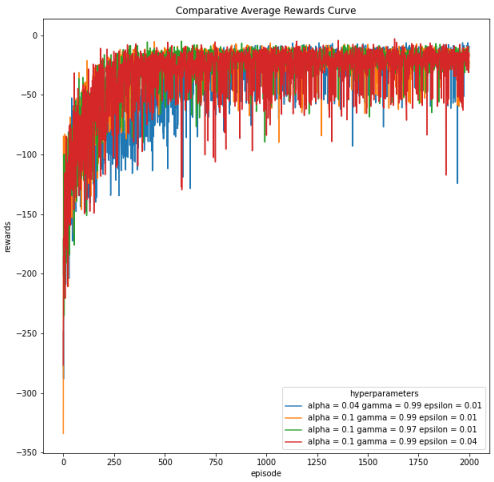
Recorded Metrics

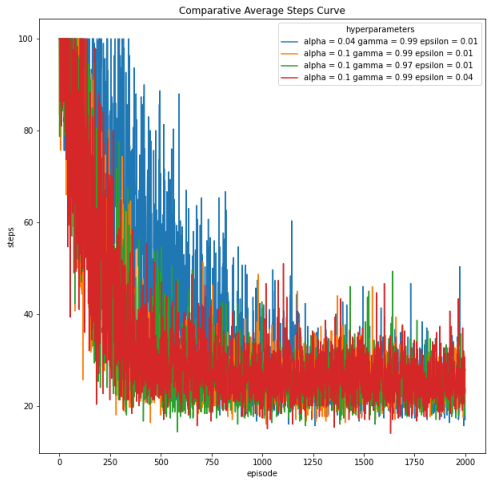


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



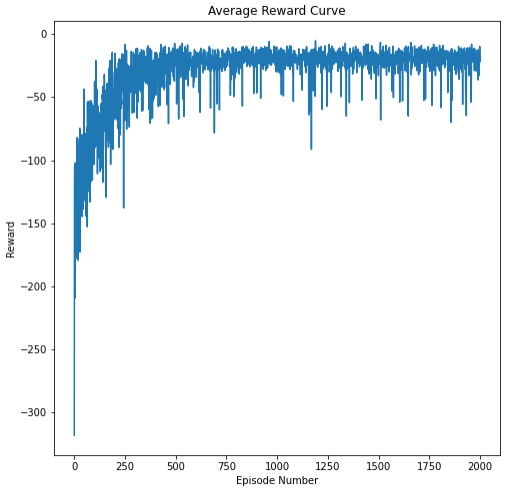


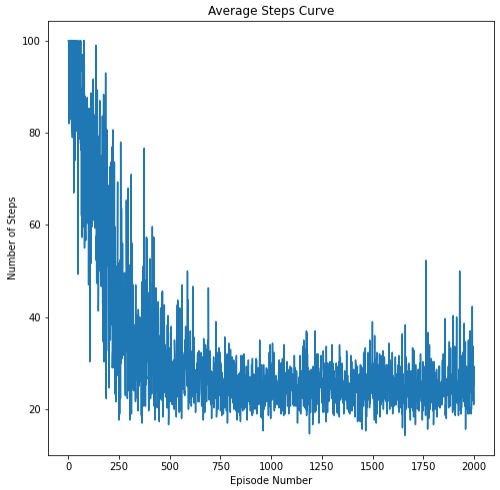
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.1, 0.97, 0.01)$ are the best hyper-parameters according to our sweep.

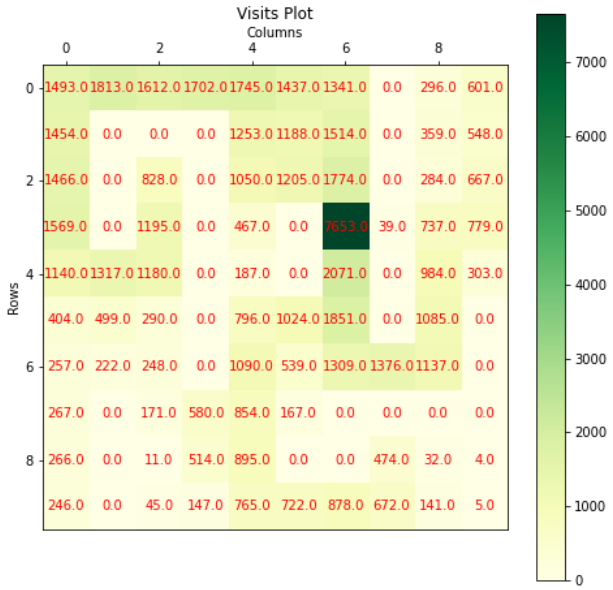
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

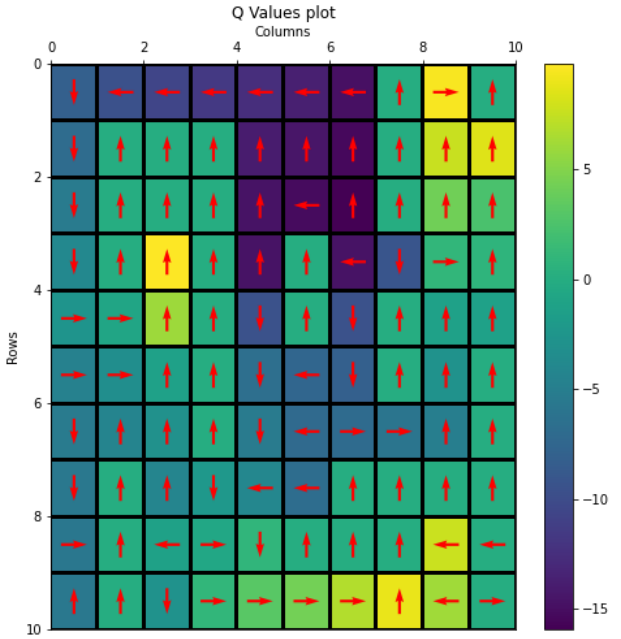




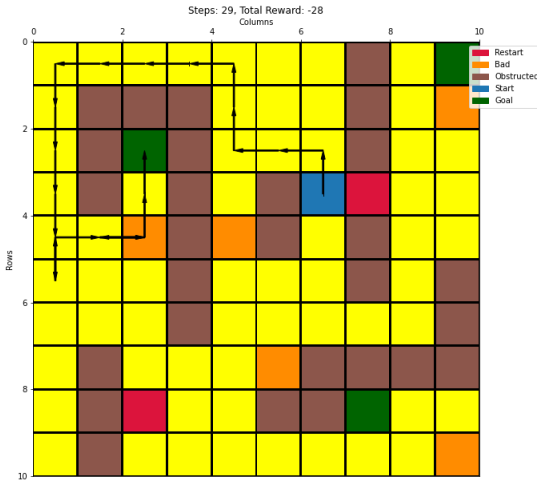
Heatmap with state visit counts

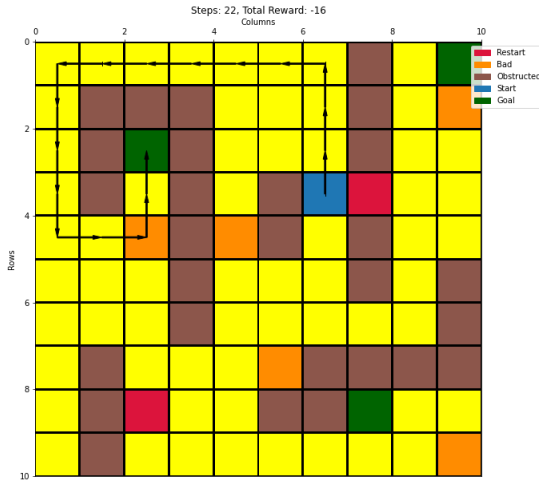


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but there is action failure.
- Action failure is the reason for the large variations in the reward and steps curves.
- From the heat map, all goal states have been visited considerable number of times but by chance, the renderings show paths to the same goal state.

Configuration 9

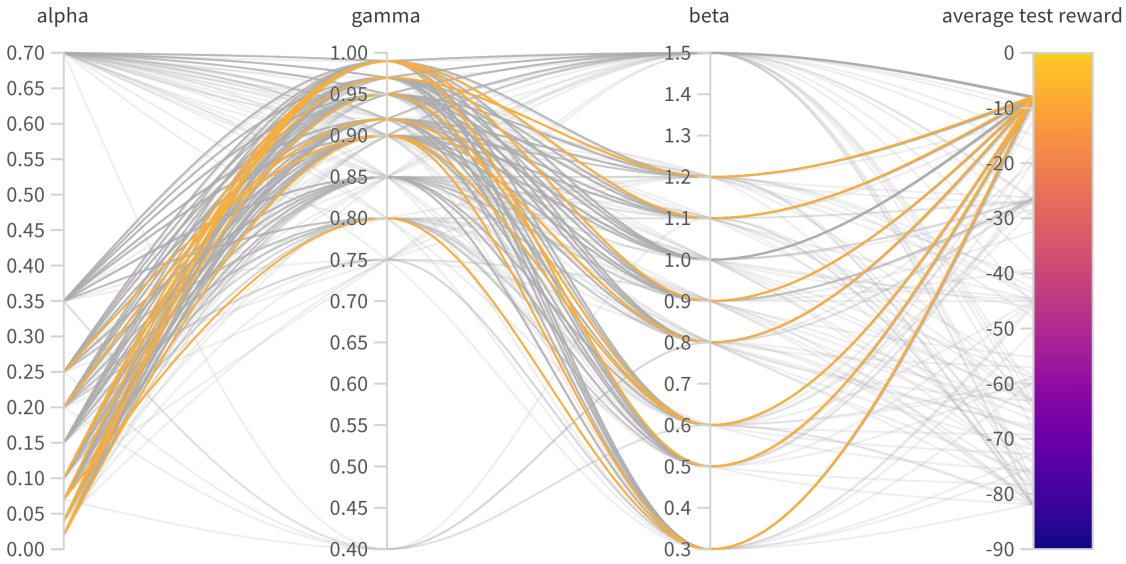
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 1.0

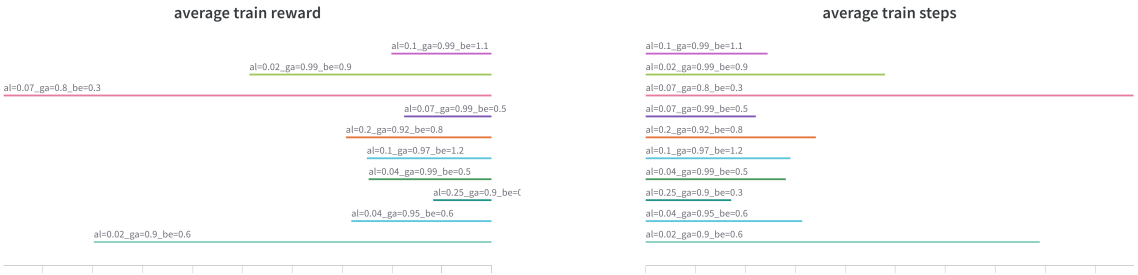
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



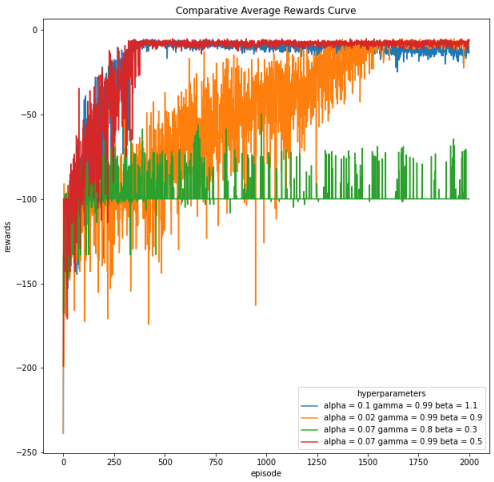
Recorded Metrics

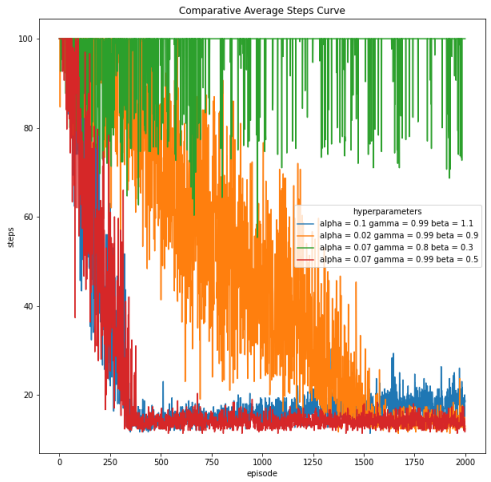


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



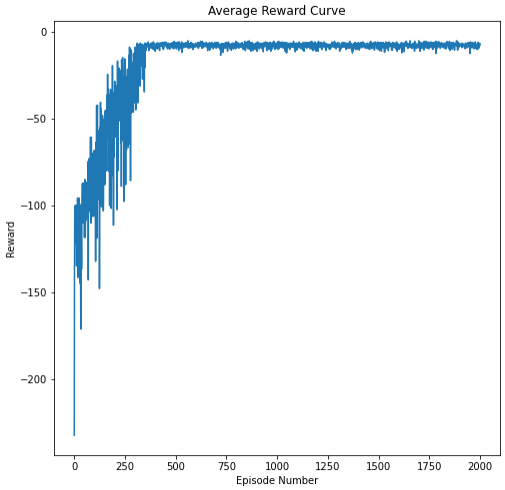


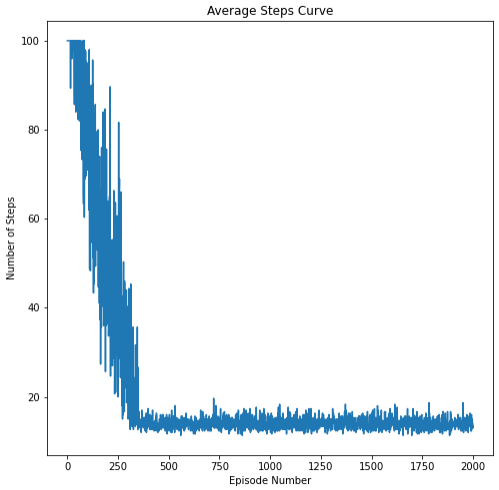
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.07, 0.99, 0.5)$ are the best hyper-parameters according to our sweep.

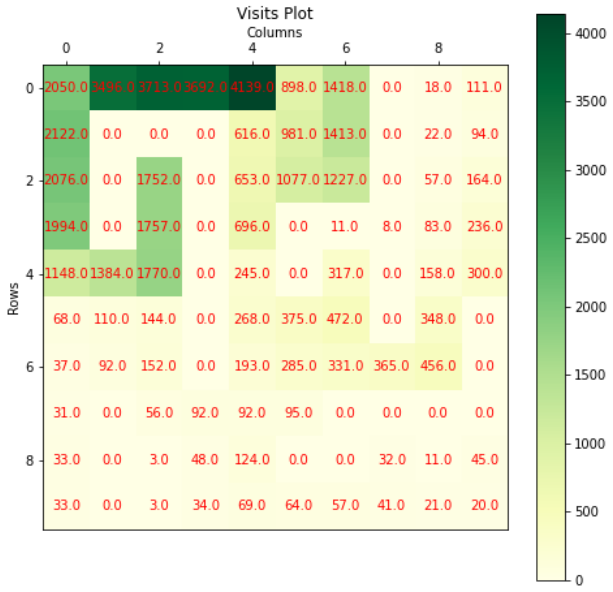
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

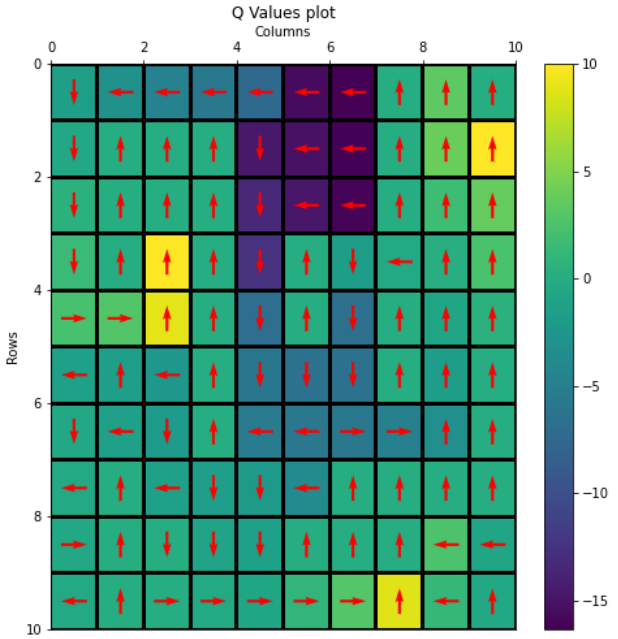




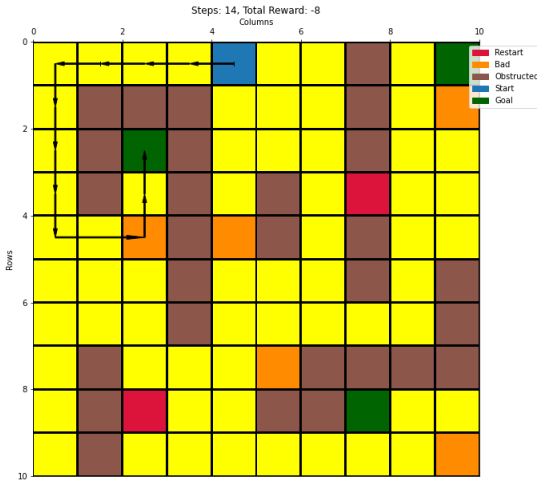
Heatmap with state visit counts

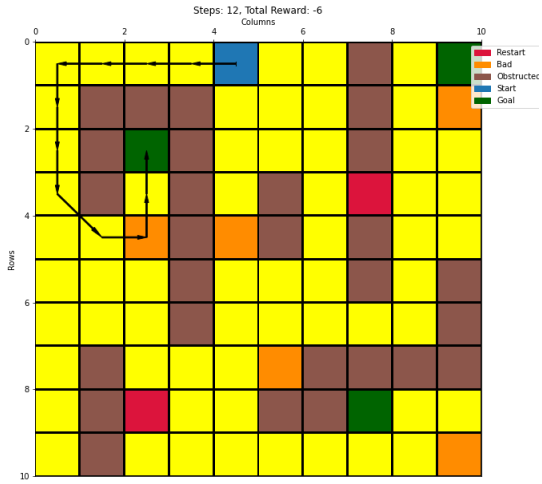


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- In this configuration, the only source of stochasticity is the wind.
- The nearest goal state is at $(2, 2)$ for this start state. The wind will be against the agent when moving along the first row. We can see that the rightward wind helps the agent move diagonally in rendering 2 and reduces the negative reward earned.

Configuration 10

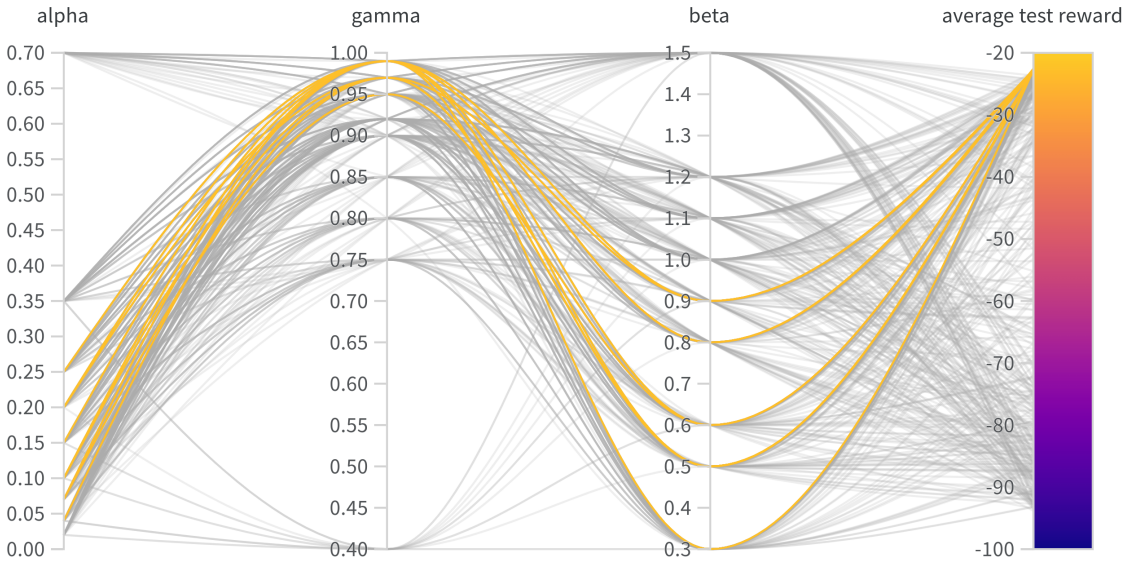
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 0.7

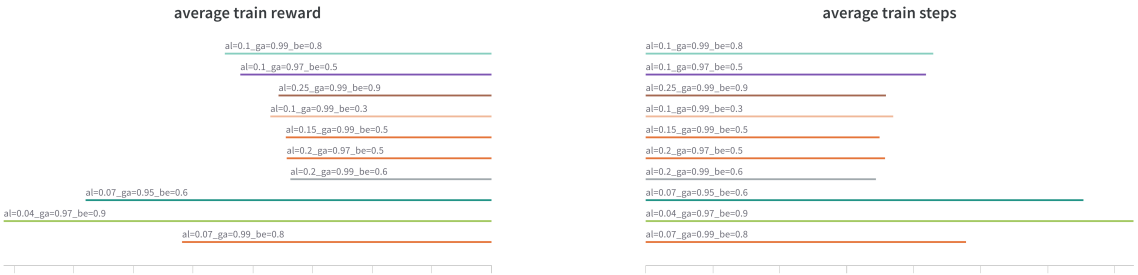
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

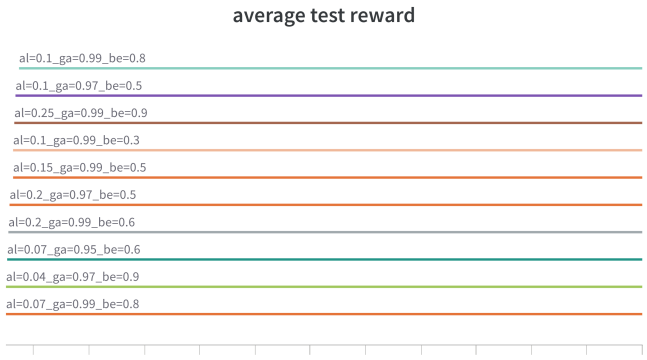
Parallel Co-ordinates Plot



Recorded Metrics

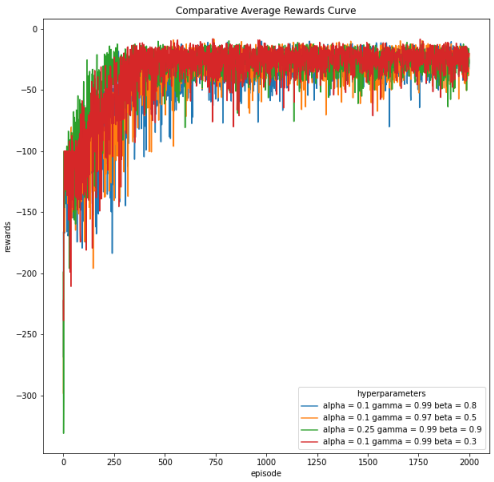


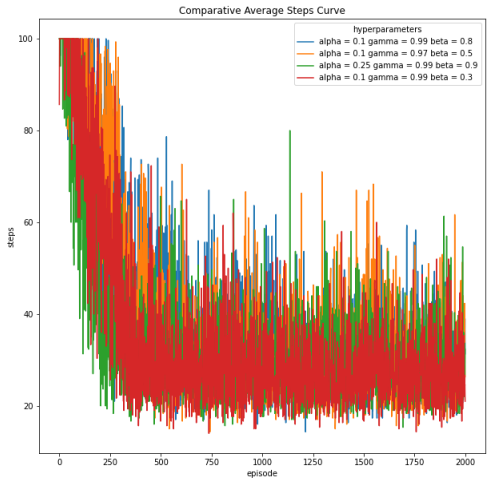
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



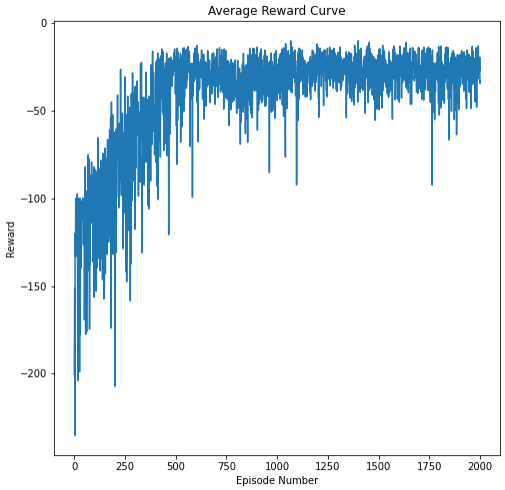


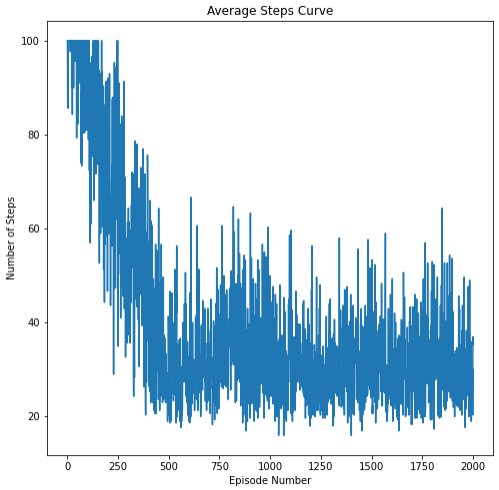
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.1, 0.99, 0.8)$ are the best hyper-parameters according to our sweep.

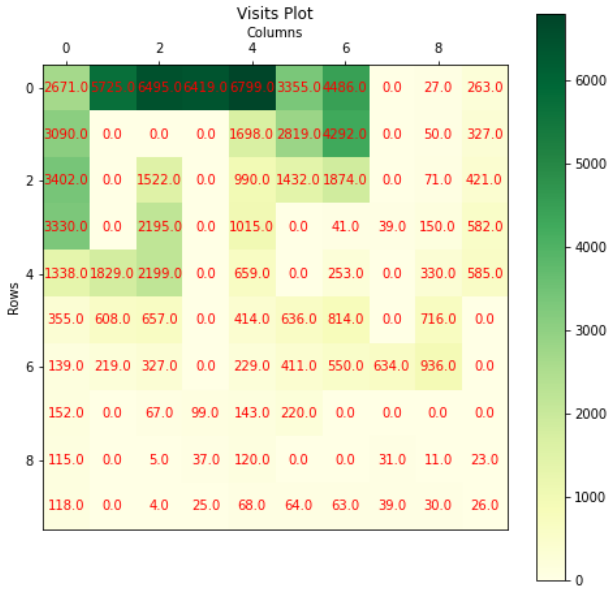
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

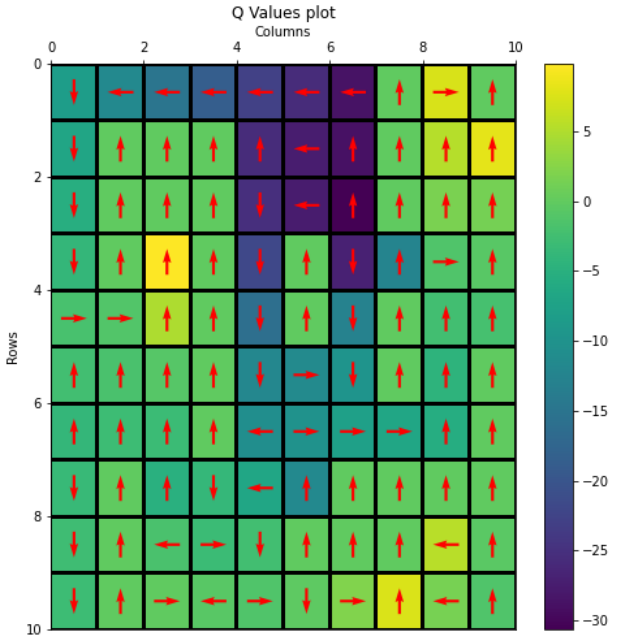




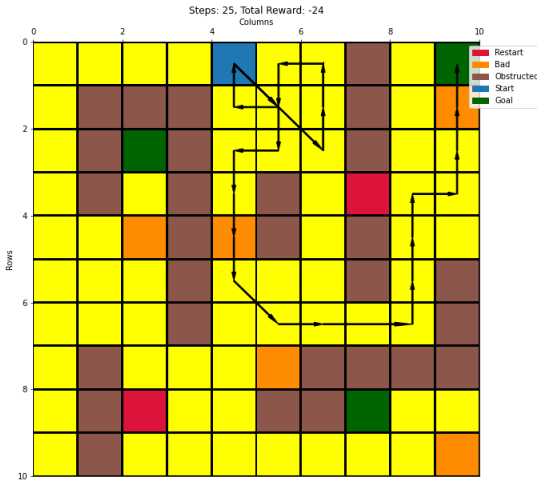
Heatmap with state visit counts

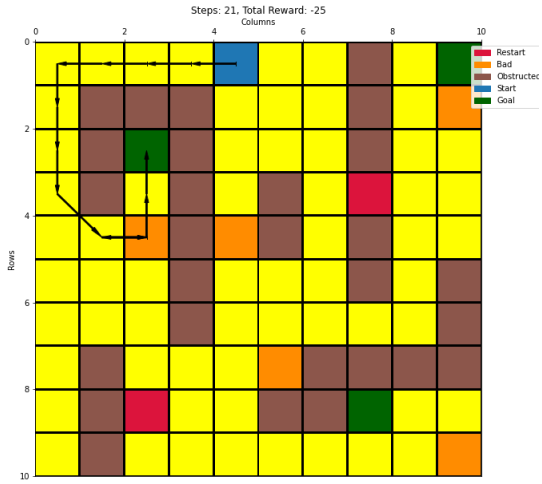


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here there is wind and action failure chance. This causes large fluctations in the reward and steps curve.
- We can see that the wind here will sometimes support the agent and sometimes oppose it.
- From the renderings, we can see that the effect of action failure is apparent as it has taken paths to 2 different goals.

Configuration 11

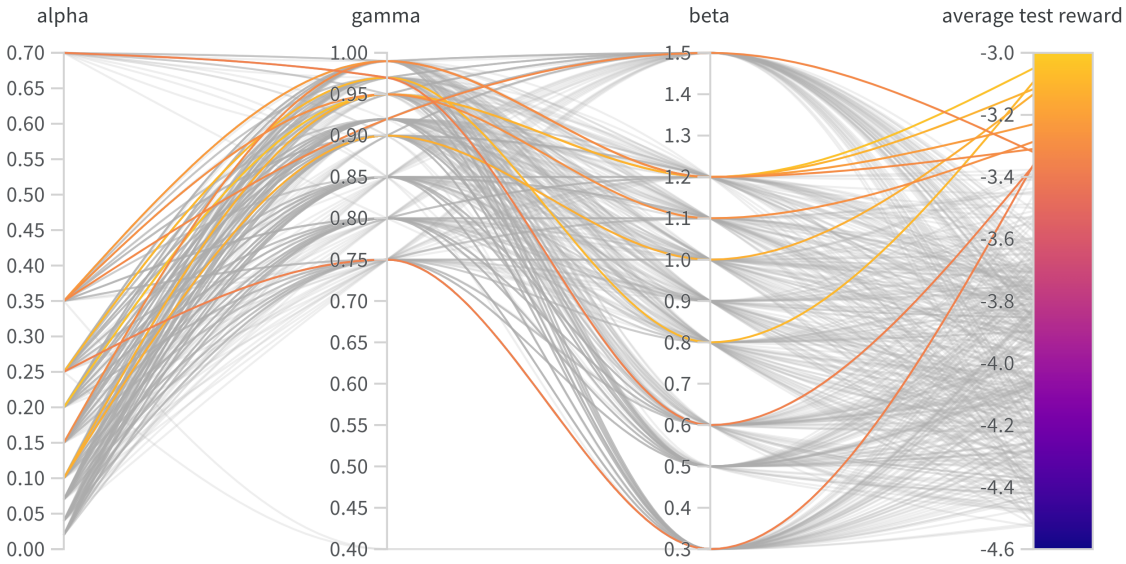
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 1.0

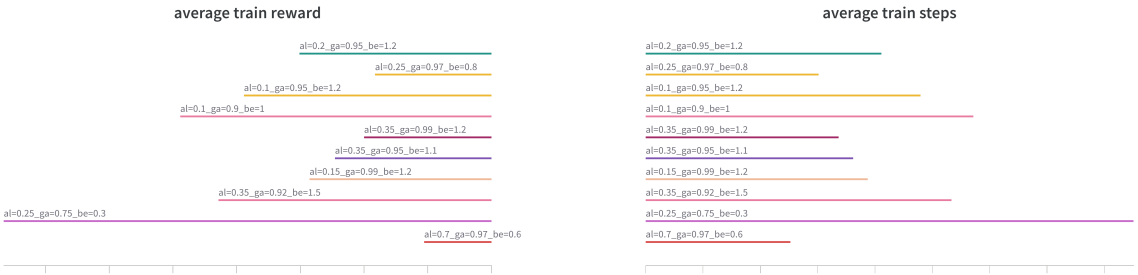
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

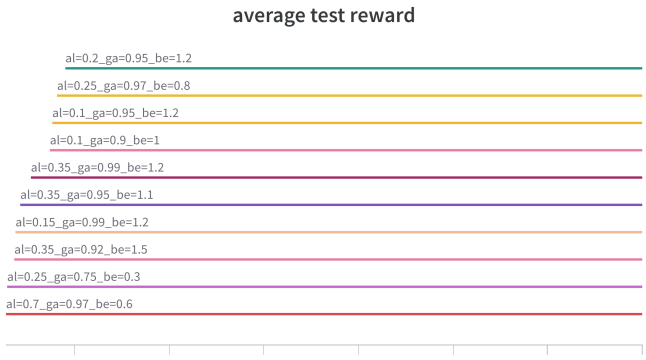
Parallel Co-ordinates Plot

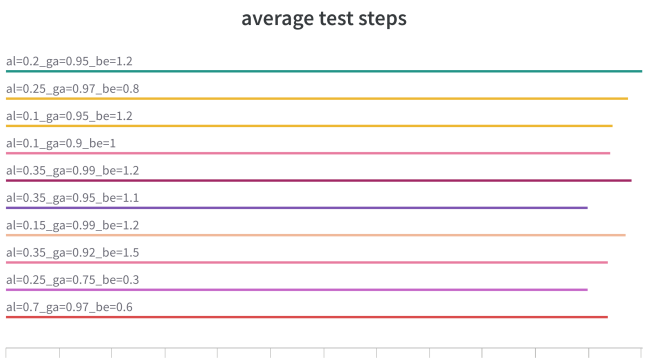


Recorded Metrics



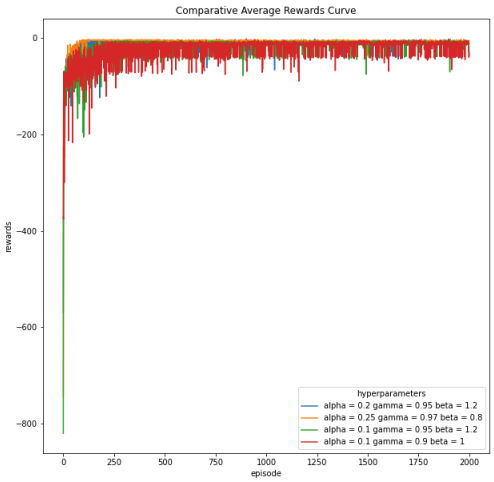
Train Metrics

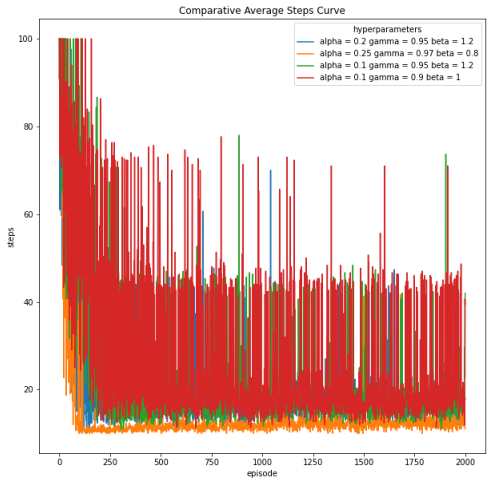




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



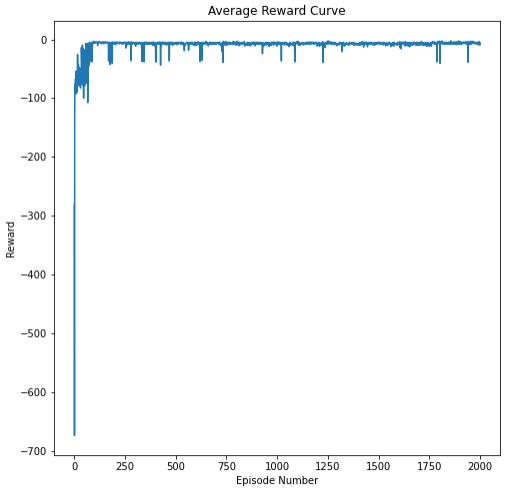


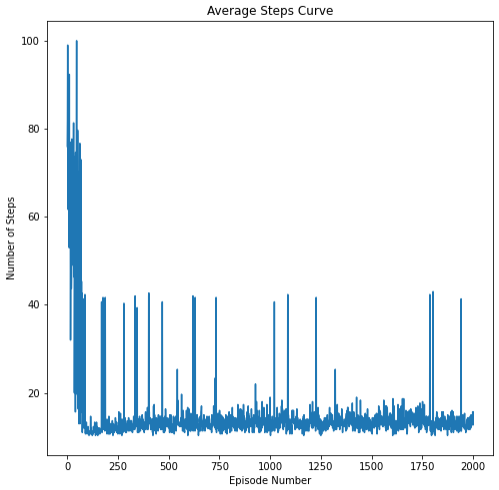
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.25, 0.97, 08)$ are the best hyper-parameters according to our sweep.

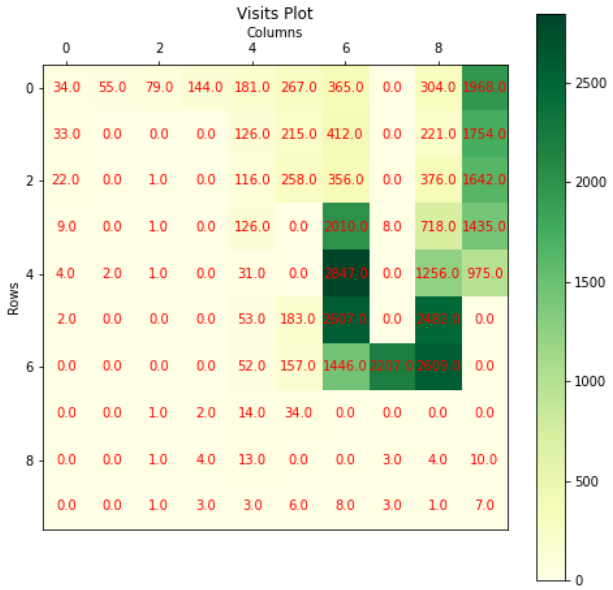
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

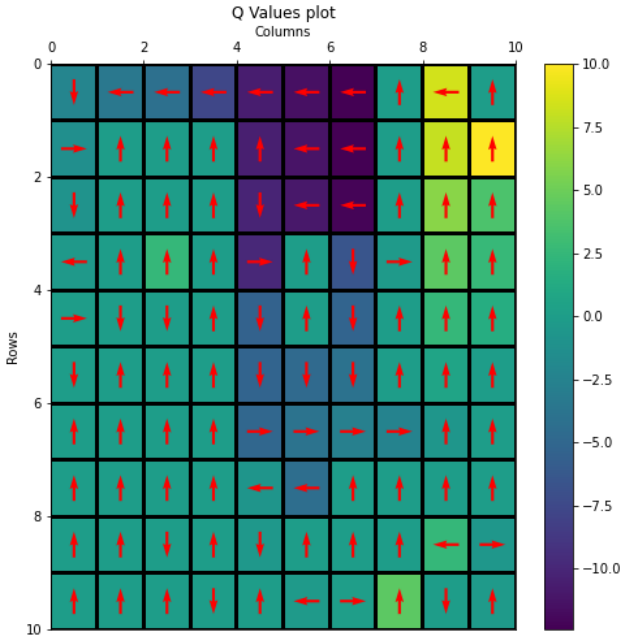




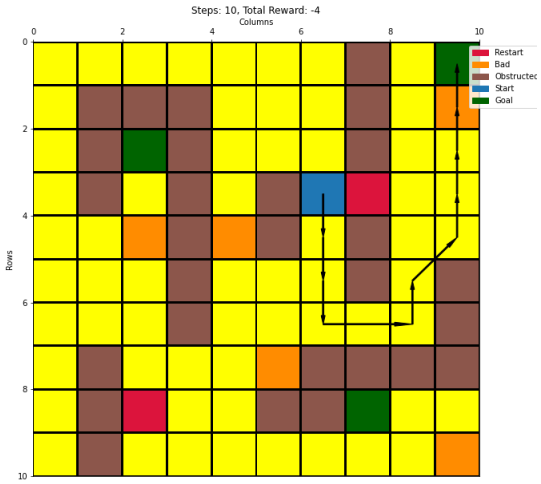
Heatmap with state visit counts

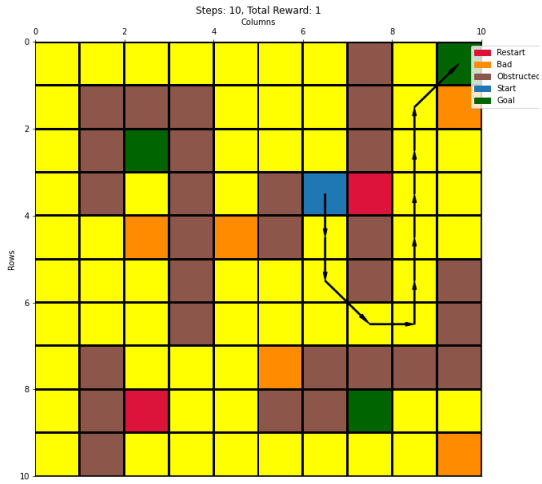


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present. Even though the goal states at $(0, 9)$ and $(8, 7)$ are equidistant from the start state, the agent biases towards $(0, 9)$ because of the wind.
- Because of the wind, the agent gets pushed to the last column and is forced to go through the bad state below the goal.
- The agent may be pushed into the restart state next to the start state and get a large negative reward. This has not happened in the above renderings but it may happen.

Configuration 12

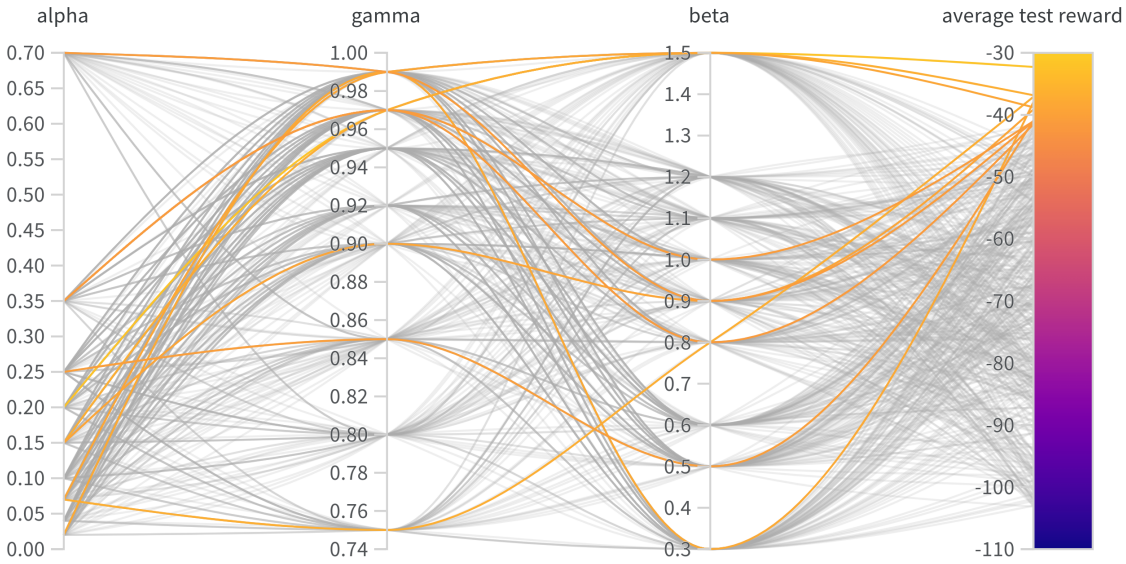
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 0.7

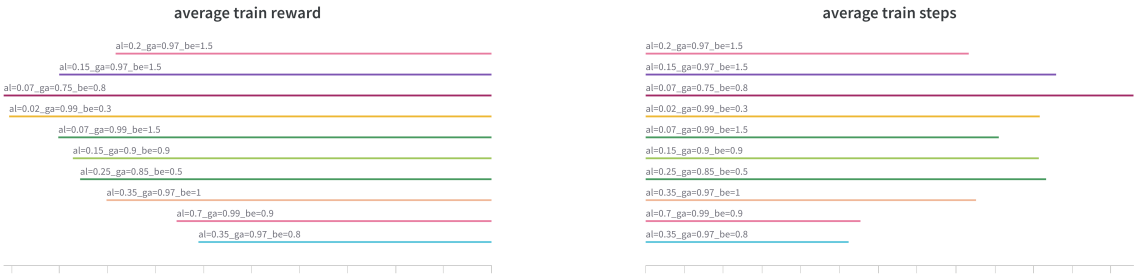
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

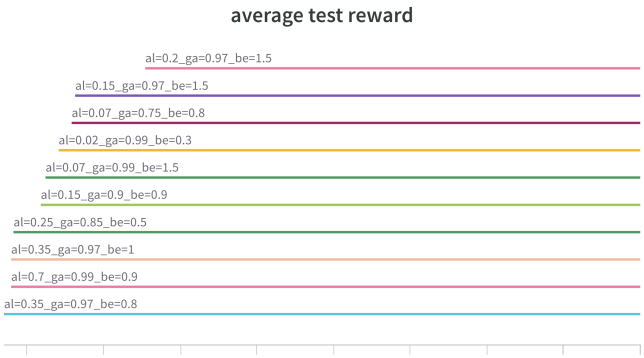
Parallel Co-ordinates Plot



Recorded Metrics

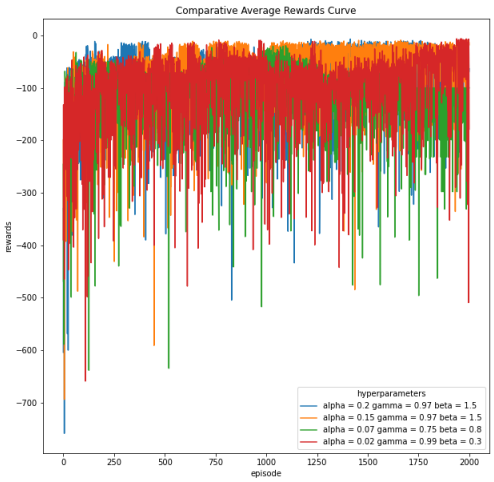


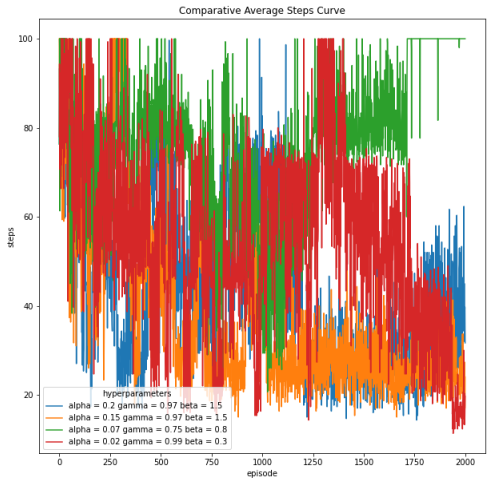
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



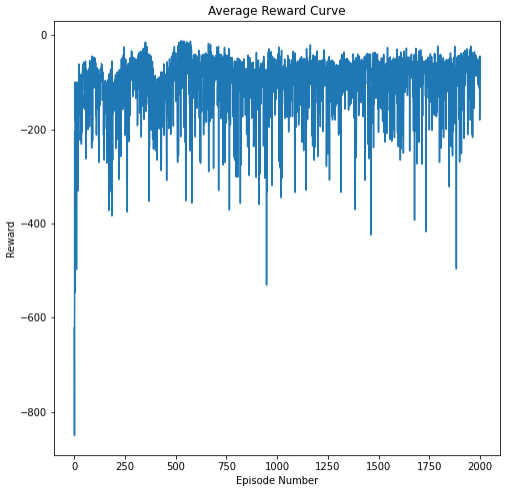


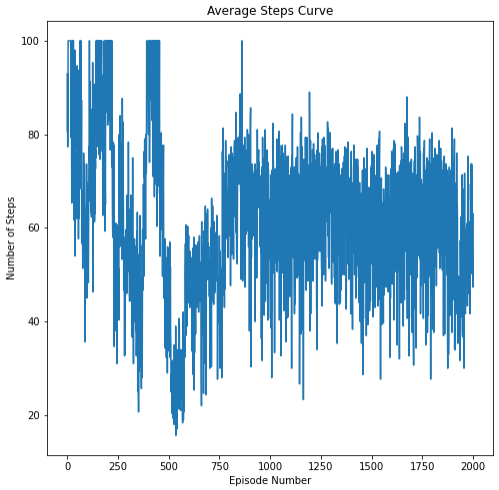
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.15, 0.97, 1.5)$ are the best hyper-parameters according to our sweep.

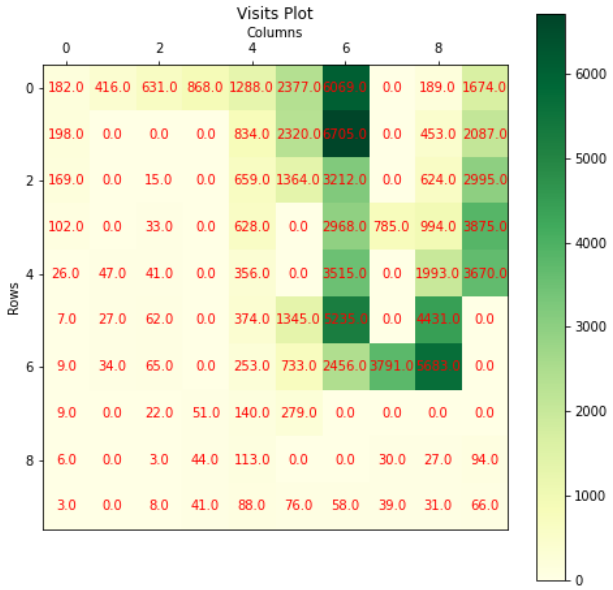
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

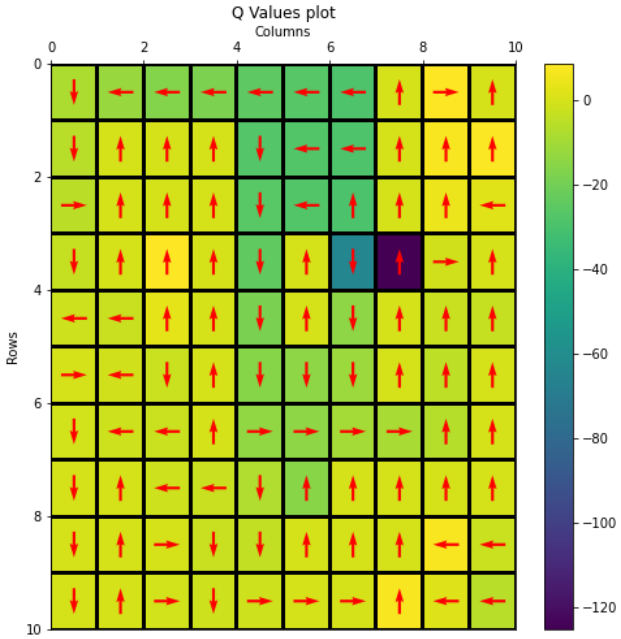




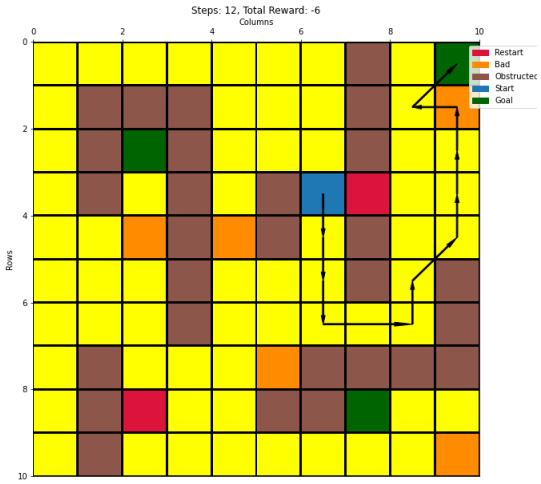
Heatmap with state visit counts

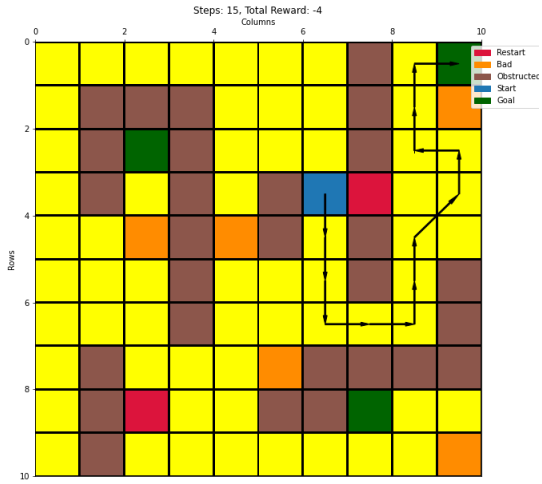


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present along with the possibility of action failure. This is the reason for the large fluctations in the reward and steps curves once again.
- Because of the wind, it is also biased towards $(0, 9)$ as well.

Configuration 13

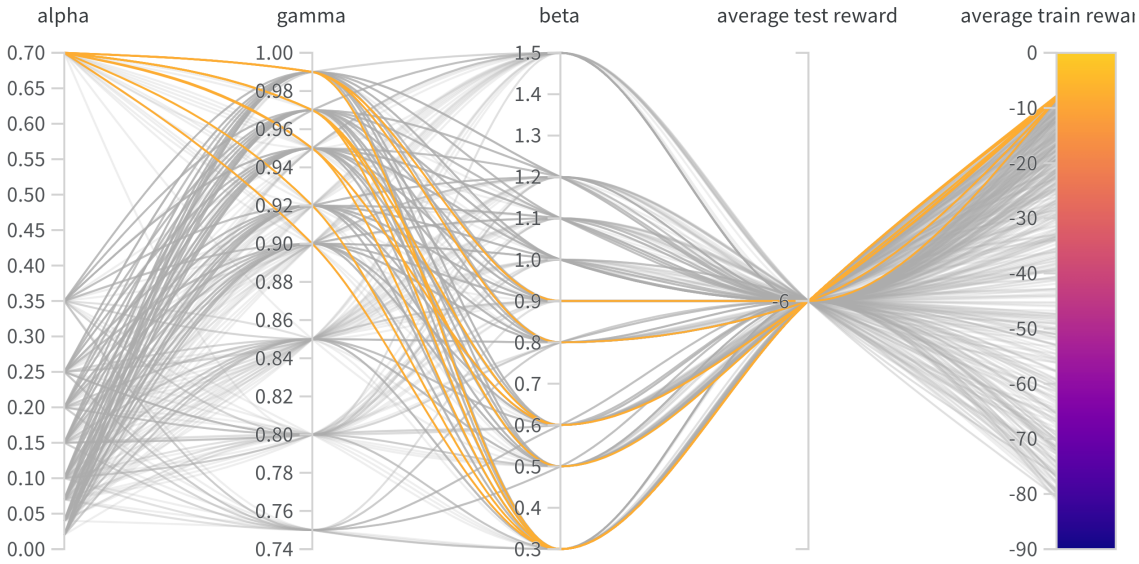
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 1.0

Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot

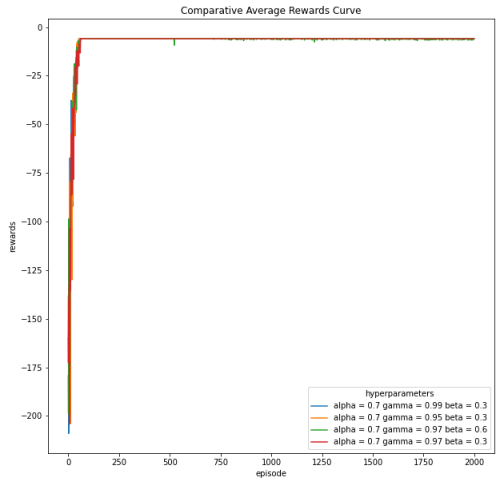


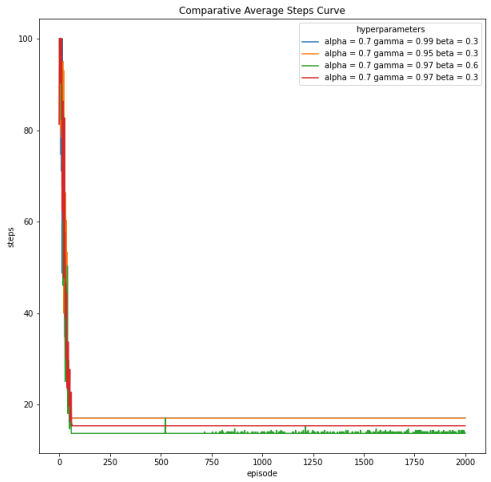
Recorded Metrics

Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



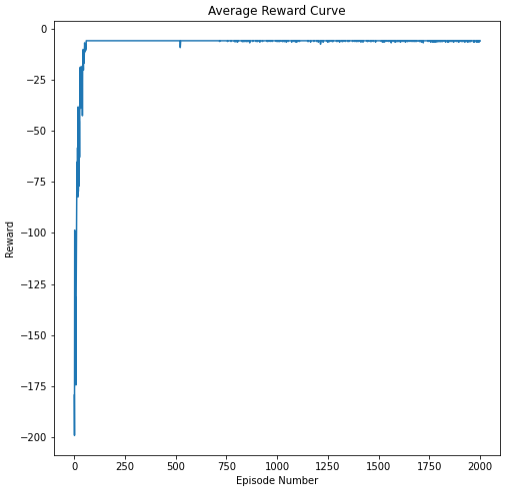


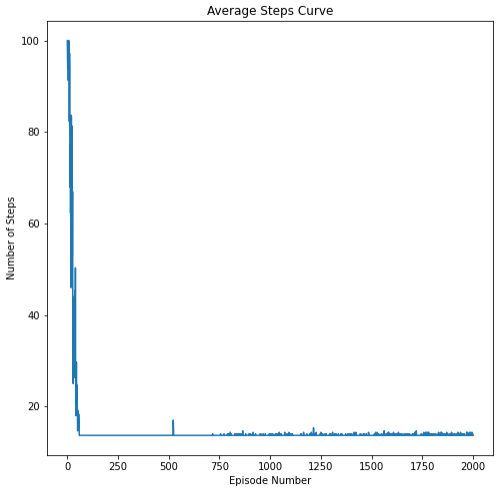
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.7, 0.97, 0.6)$ are the best hyper-parameters according to our sweep.

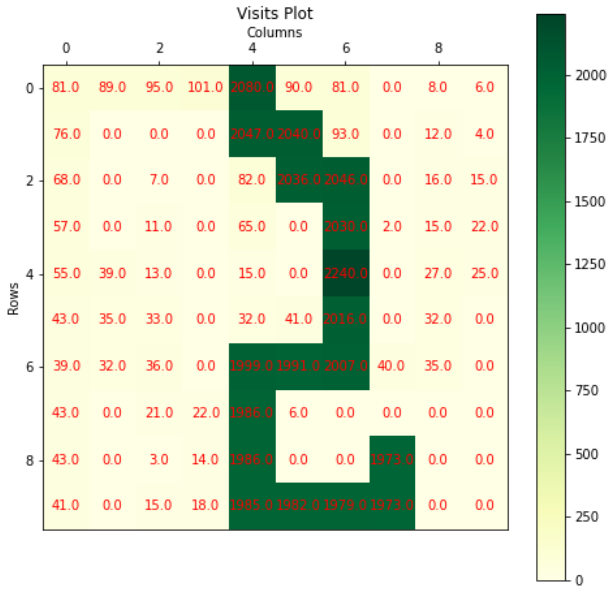
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

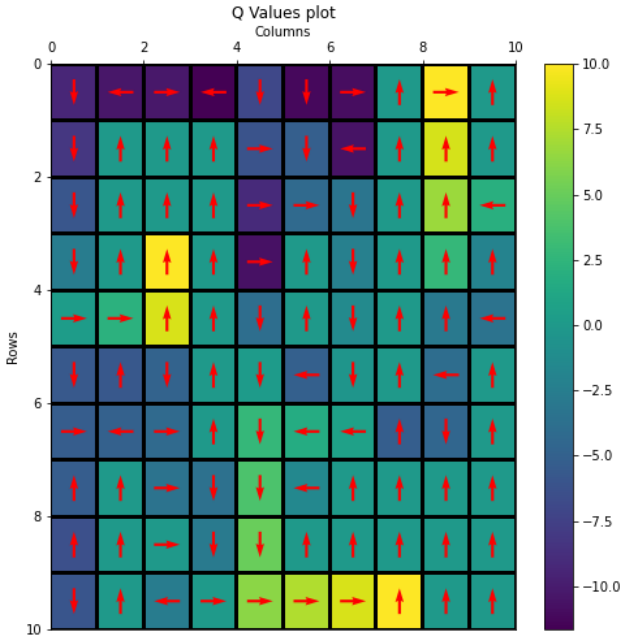




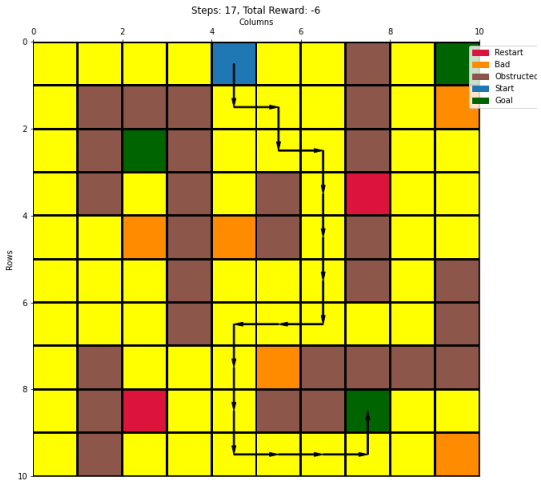
Heatmap with state visit counts

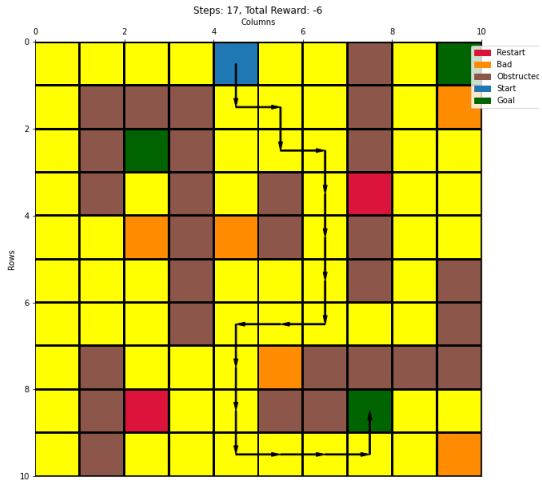


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path towards the goal (8,7).

Configuration 14

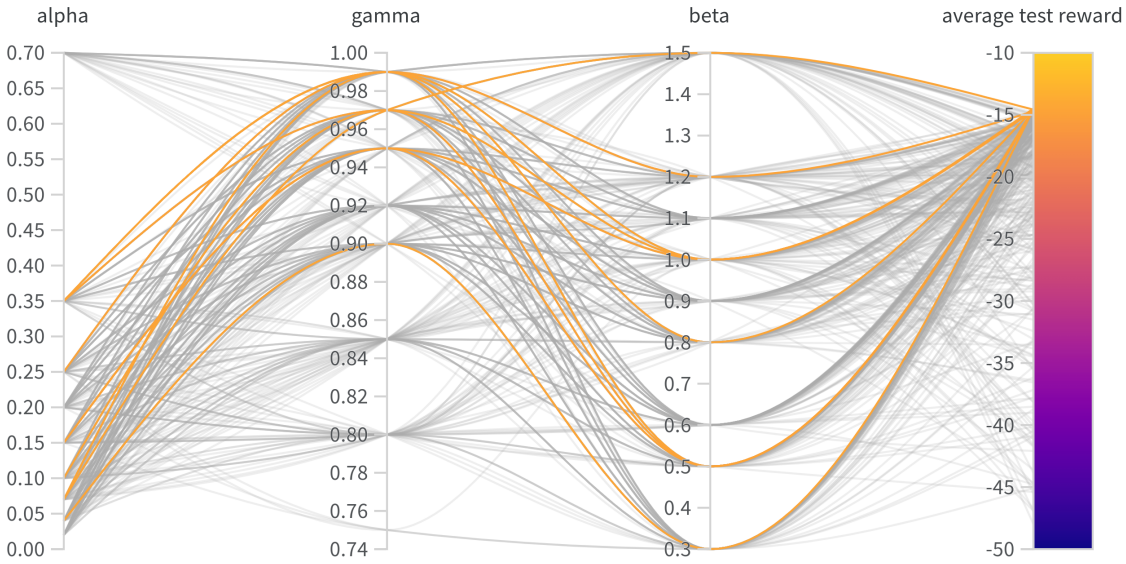
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 0.7

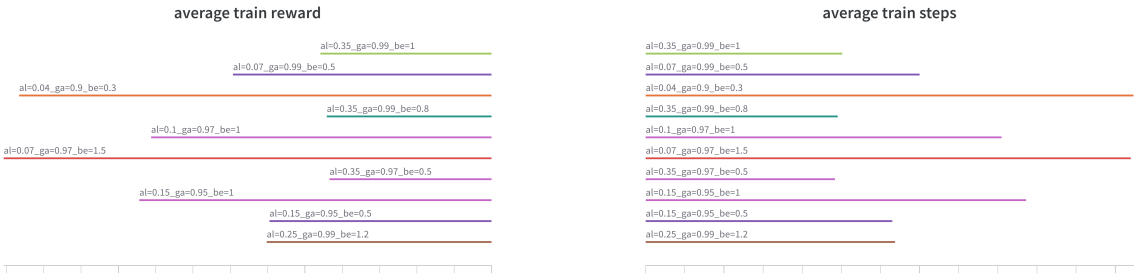
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

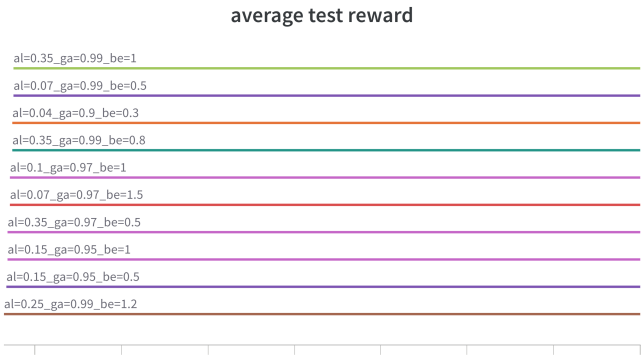
Parallel Co-ordinates Plot

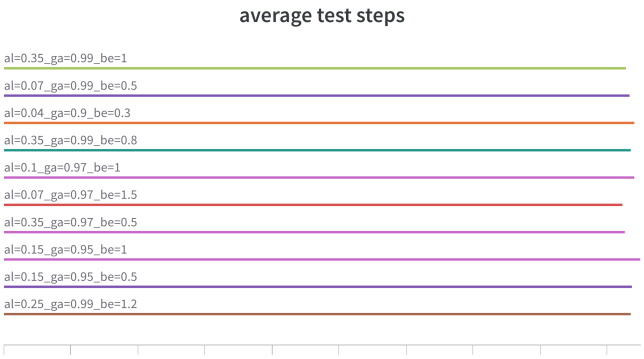


Recorded Metrics



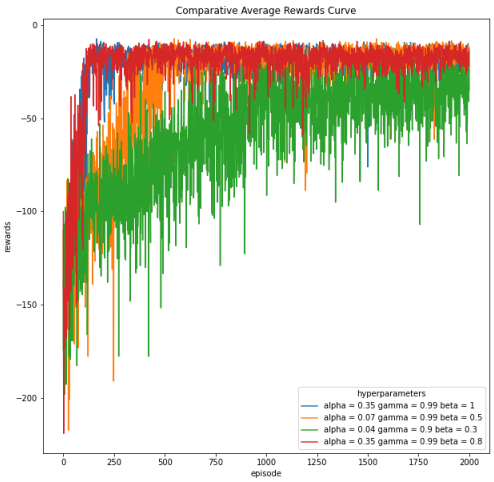
Train Metrics

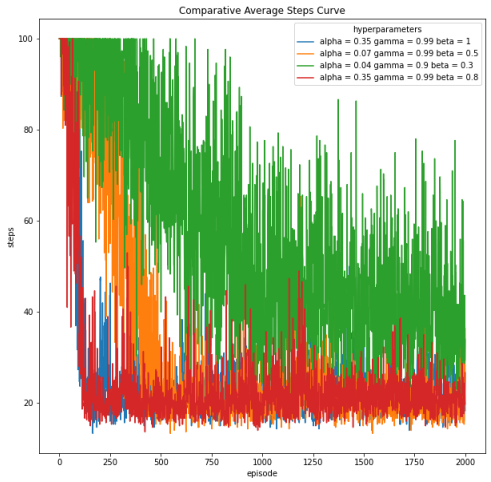




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



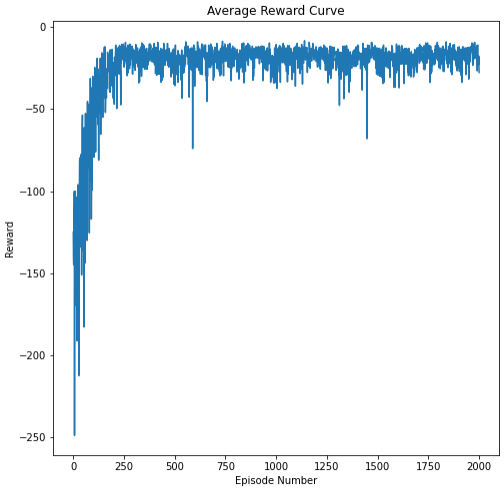


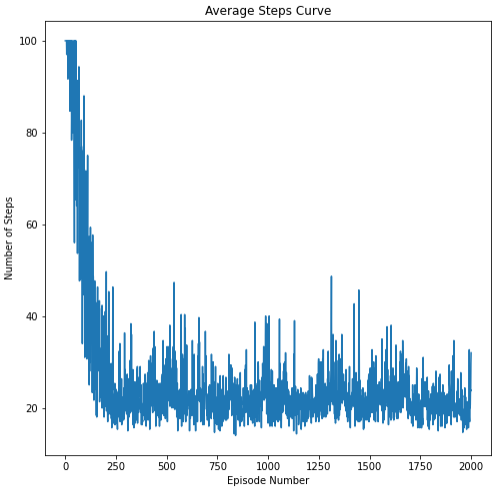
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.35, 0.99, 1.0)$ are the best hyper-parameters according to our sweep.

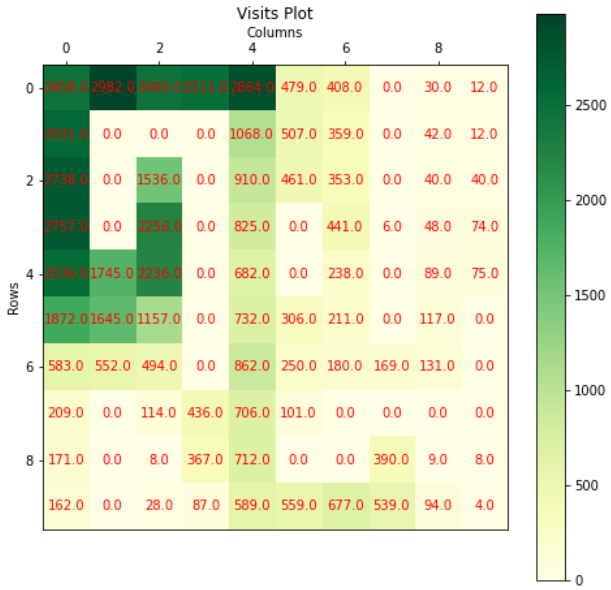
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

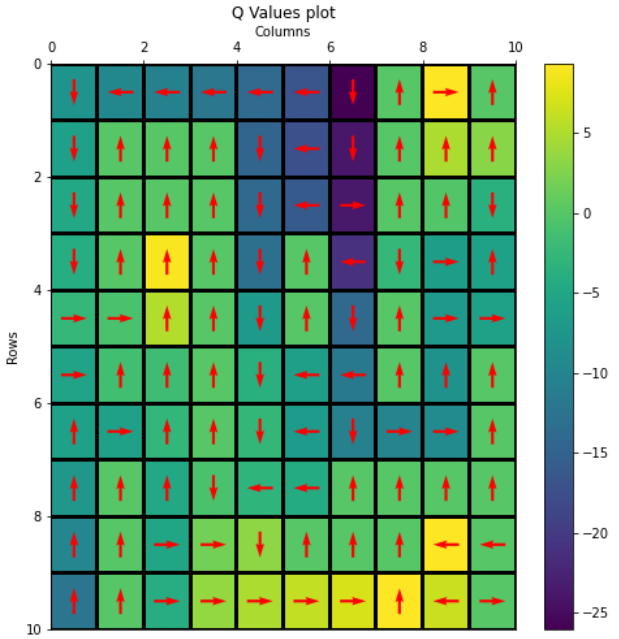




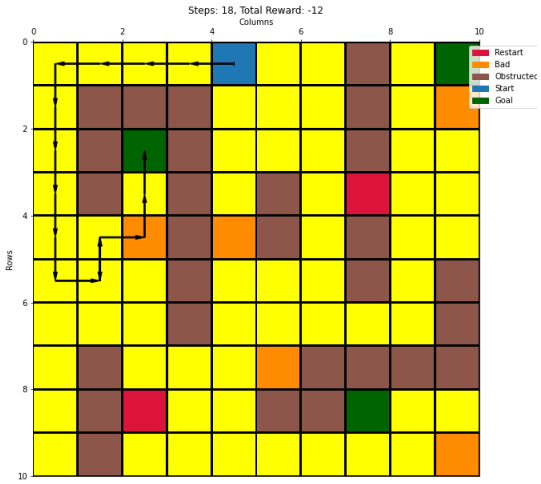
Heatmap with state visit counts

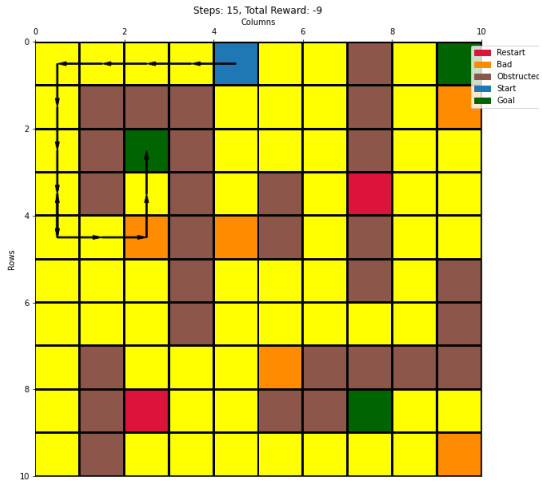


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but there is action failure.
- The agent always tries to take the path to $(2, 2)$ directly. The paths taken in the renderings show a bias towards $(2, 2)$ because the shown path is highly constrained where movement along the first row and first column is not affected by action failure.

Configuration 15

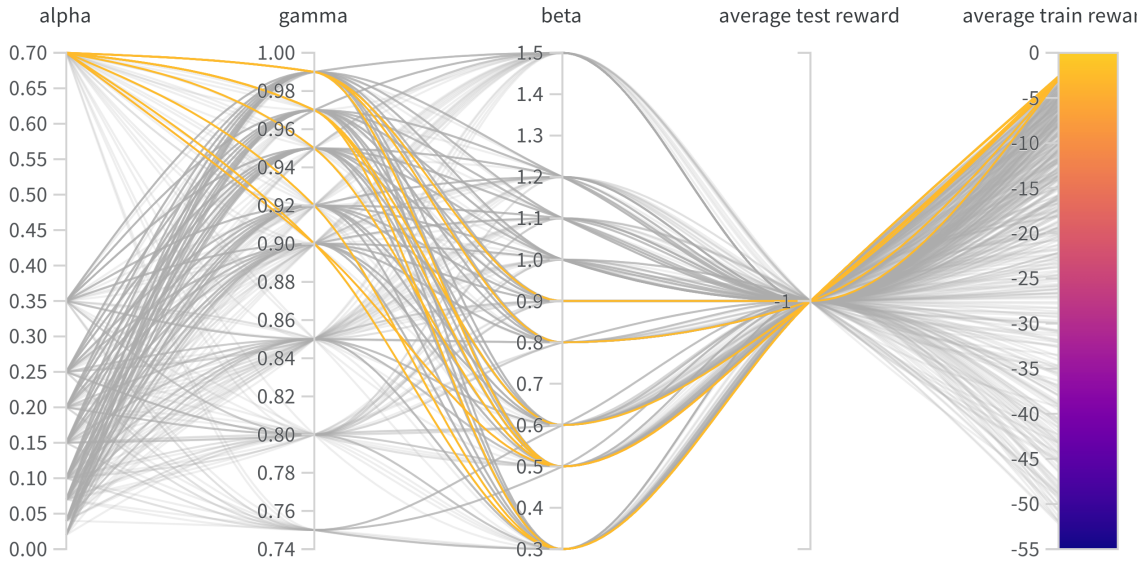
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 1.0

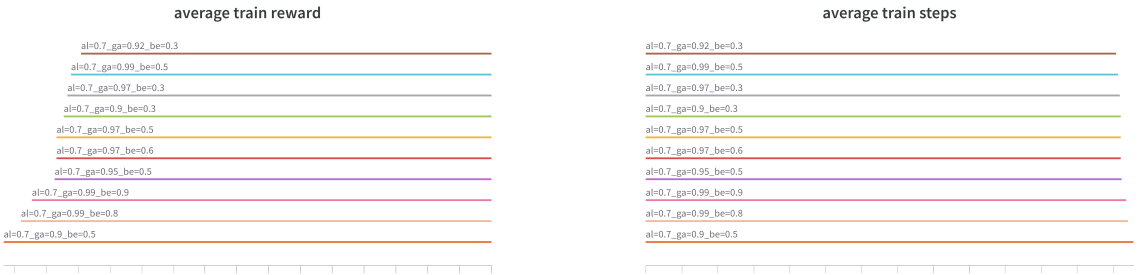
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

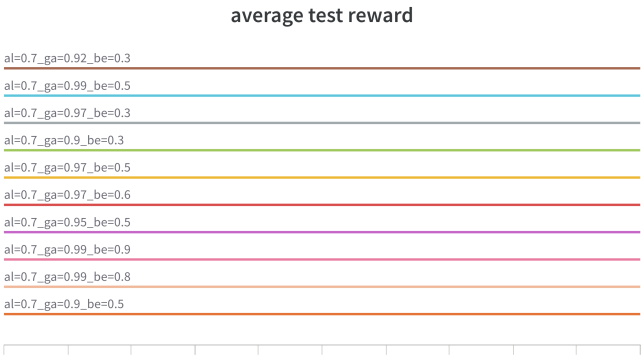
Parallel Co-ordinates Plot



Recorded Metrics

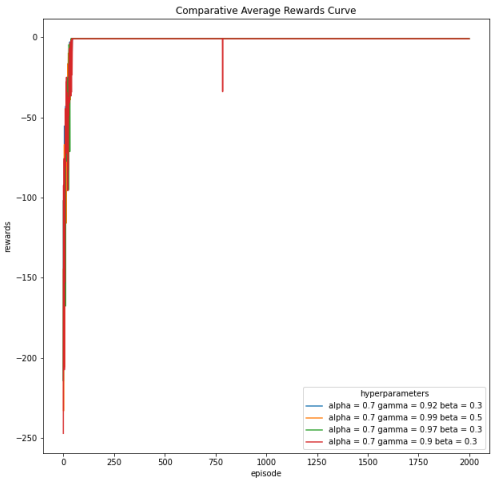


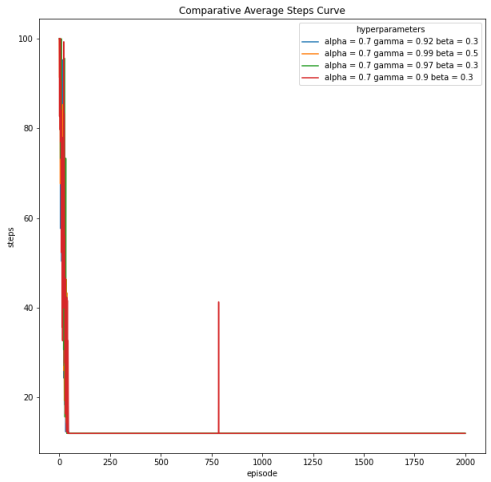
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



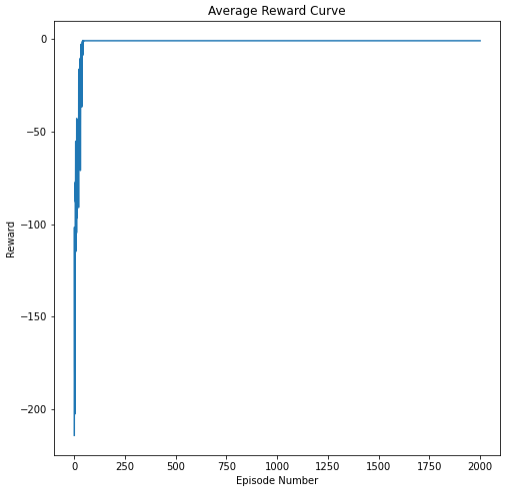


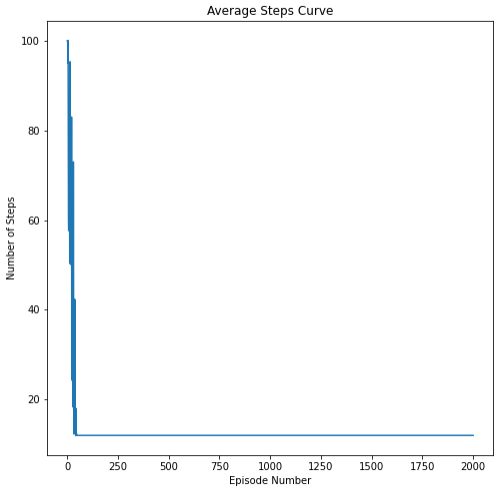
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.7, 0.92, 0.3)$ are the best hyper-parameters according to our sweep.

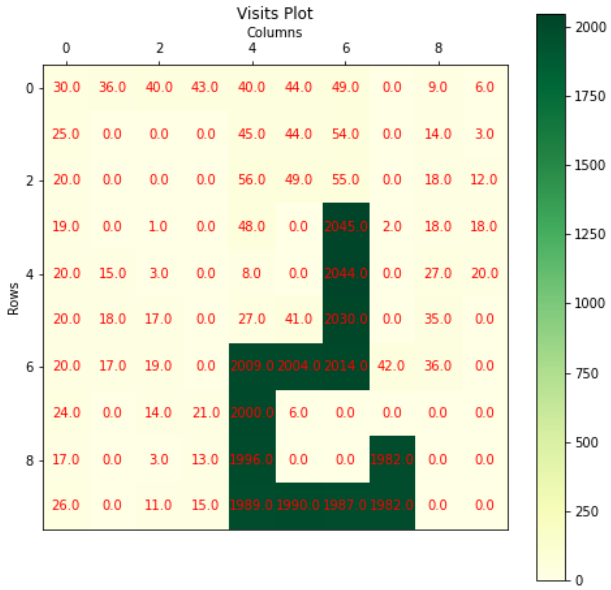
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

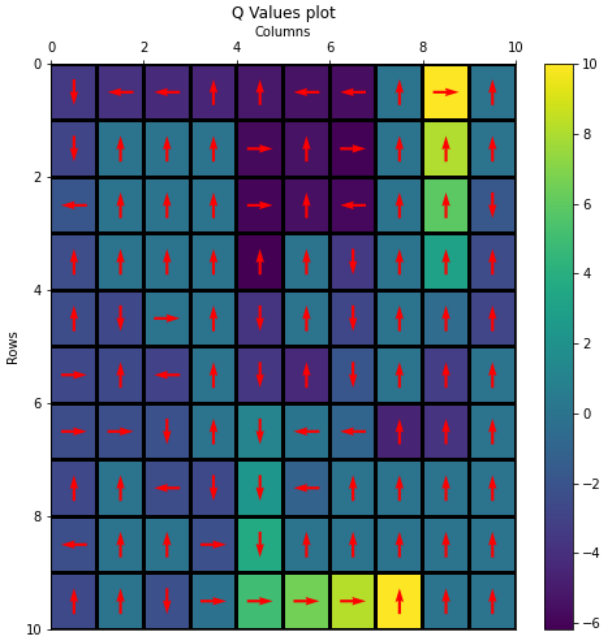




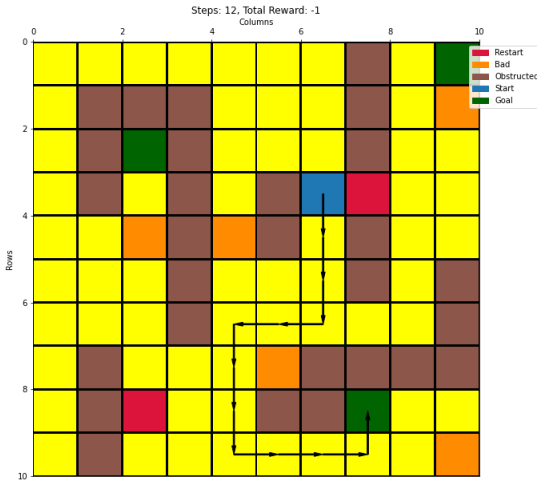
Heatmap with state visit counts

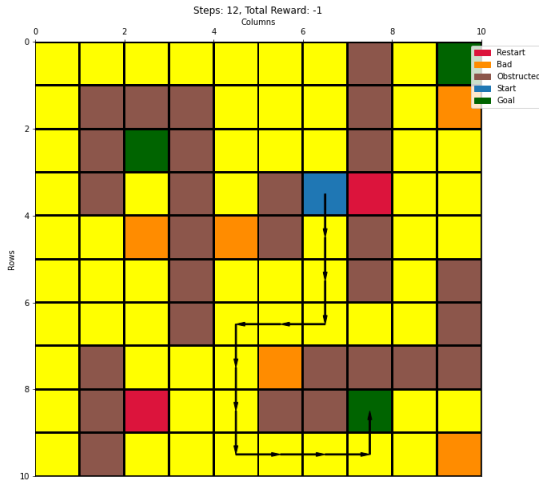


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path with a bad state in the way as it is optimal (in terms of reward earned).

Configuration 16

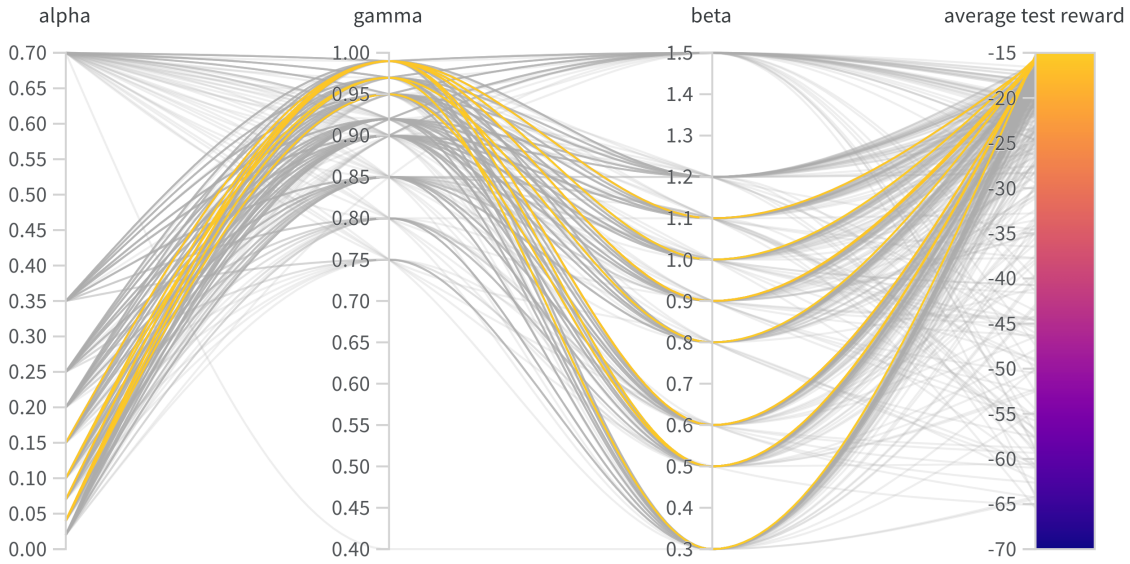
Configuration Description

- **learning -** SARSA
- **action -** Softmax Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 0.7

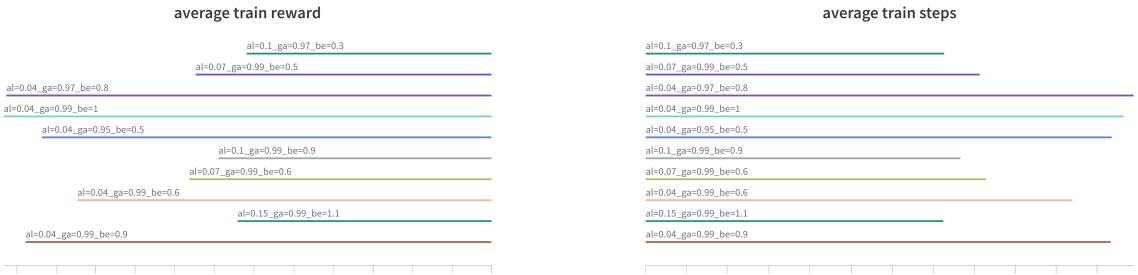
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

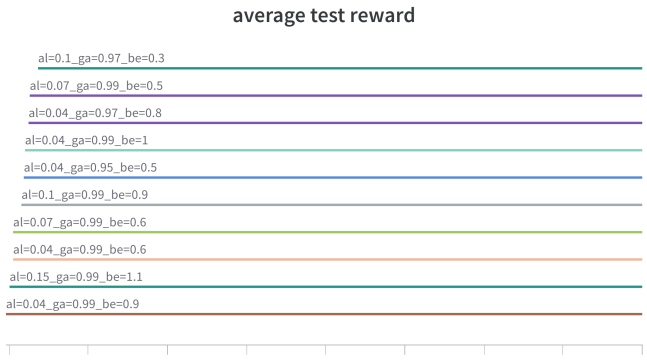
Parallel Co-ordinates Plot

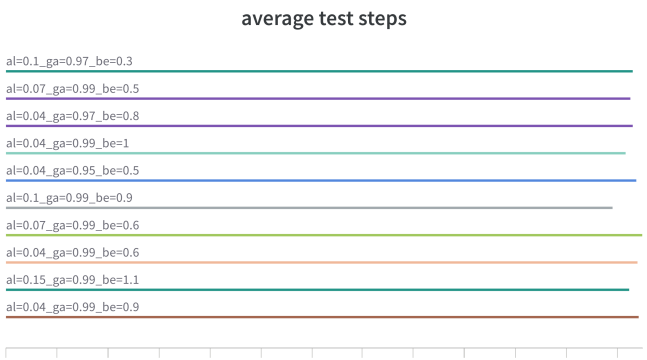


Recorded Metrics



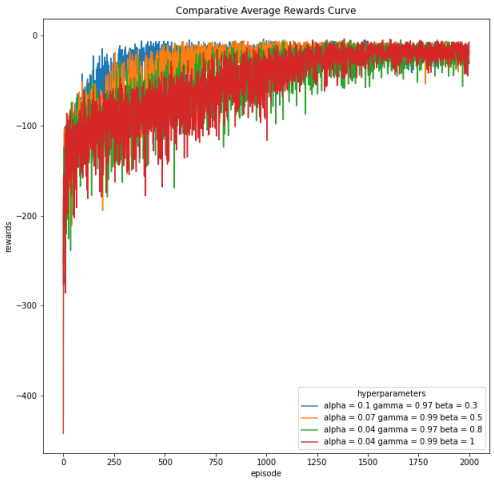
Train Metrics

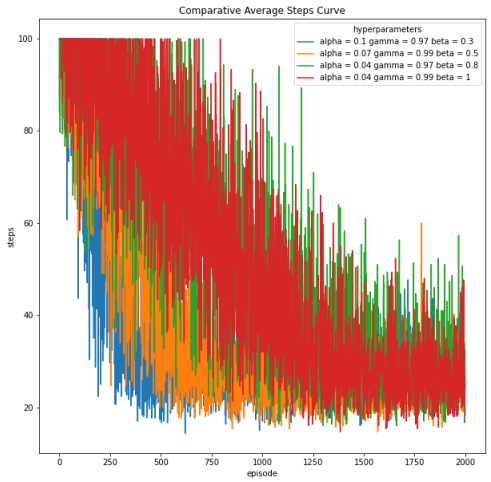




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



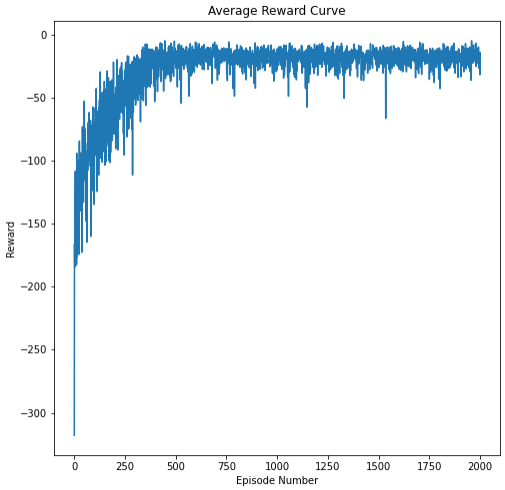


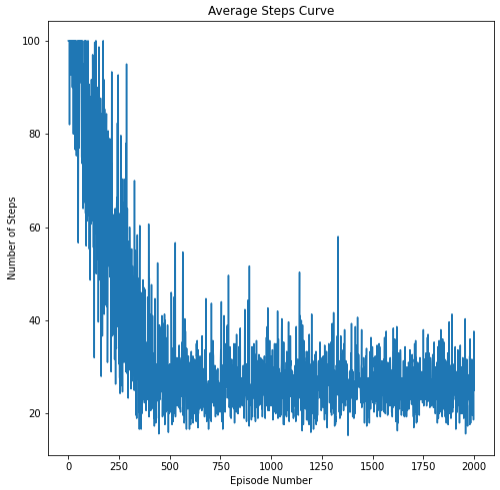
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.1, 0.97, 0.3)$ are the best hyper-parameters according to our sweep.

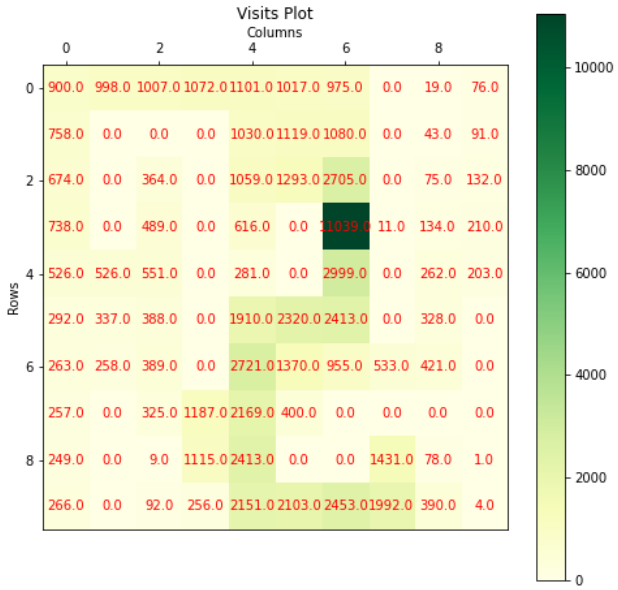
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

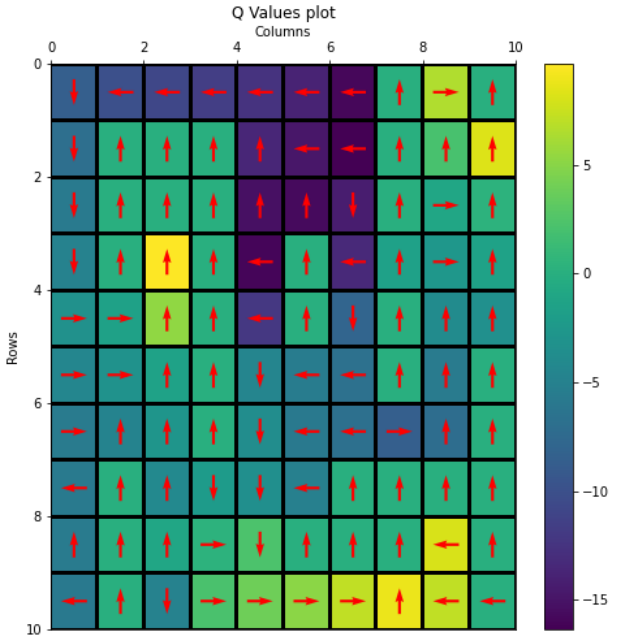




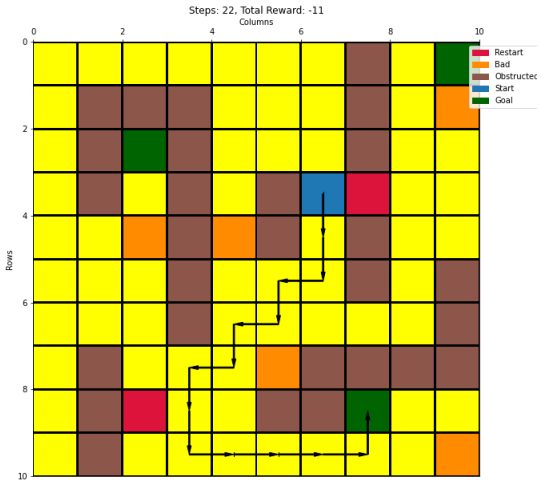
Heatmap with state visit counts

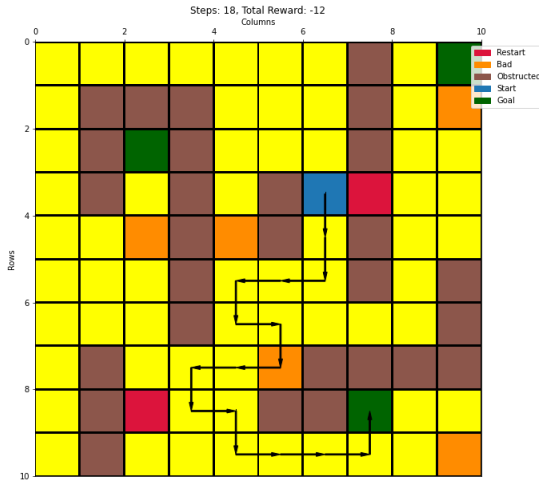


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but there is action failure.
- Action failure is the reason for the large variations in the reward and steps curves.
- Compared to epsilon-greedy action selection in confifuration 8, we can see that $(0, 9)$ is not explored significantly by softmax action selection.

Configuration 17

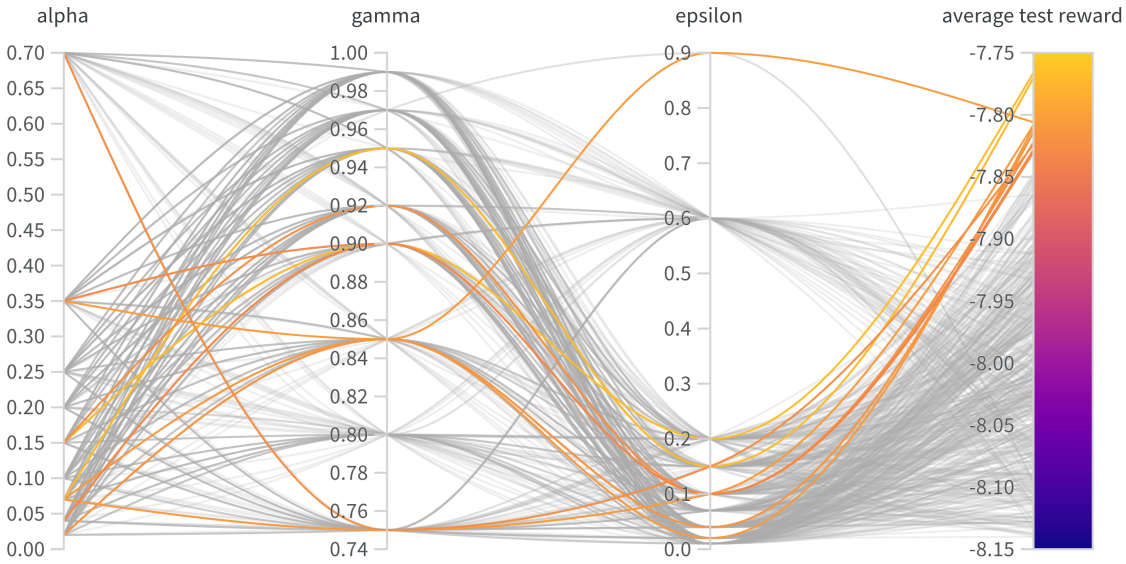
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 1.0

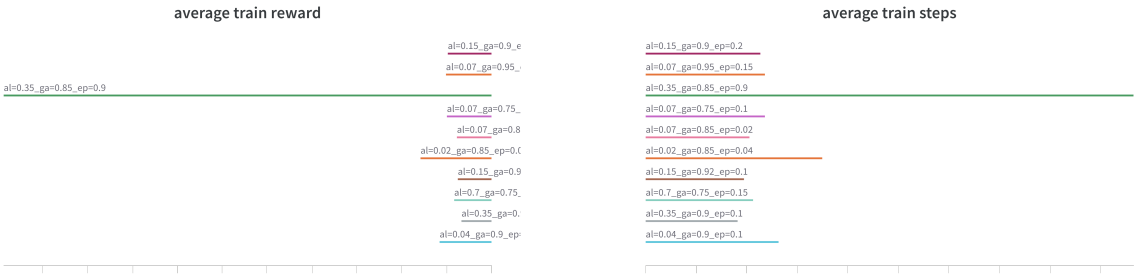
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



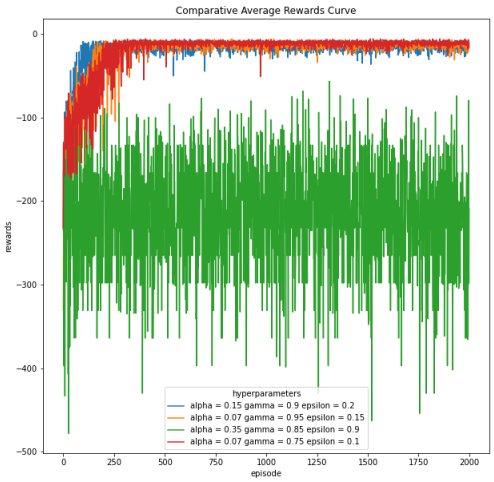
Recorded Metrics

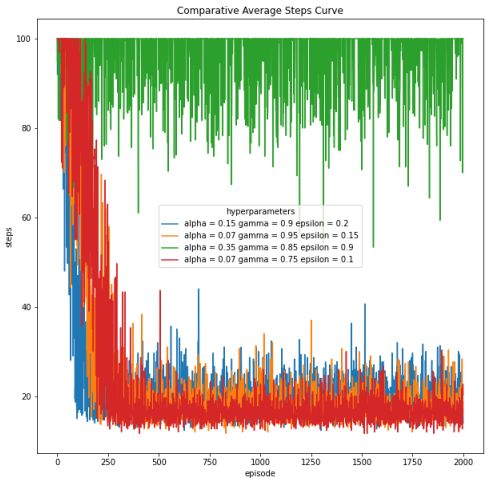


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



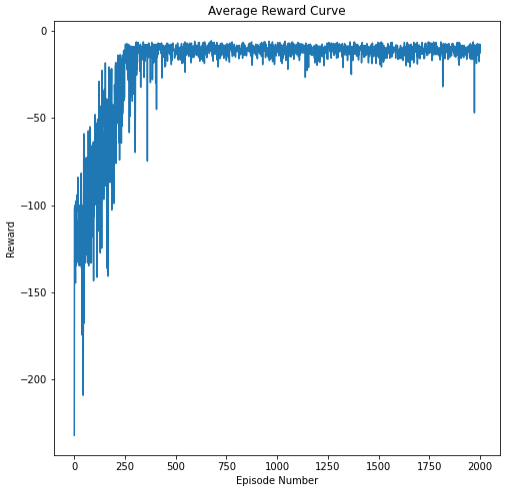


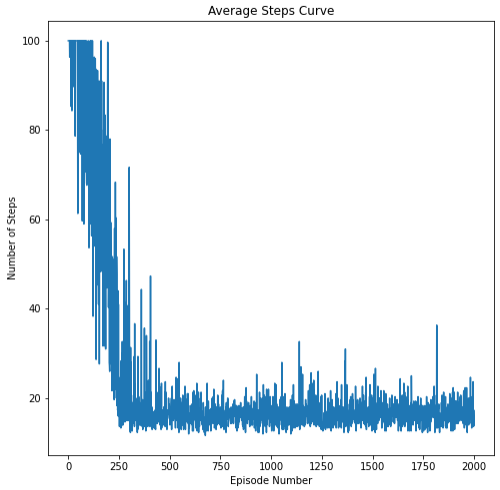
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.07, 0.75, 0.1)$ are the best hyper-parameters according to our sweep.

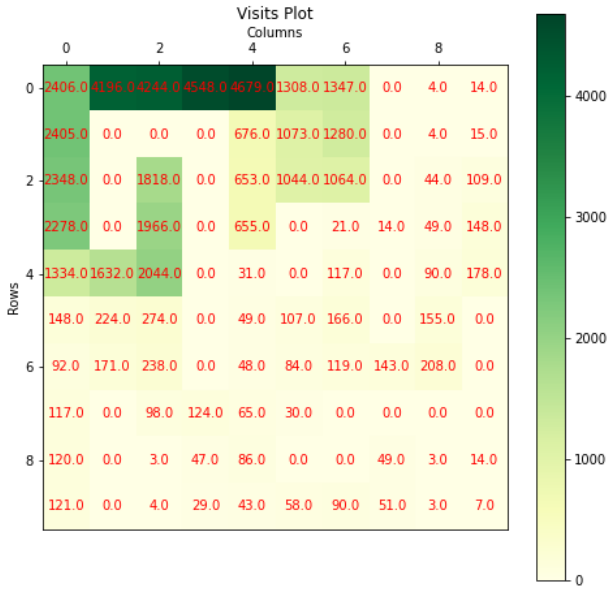
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

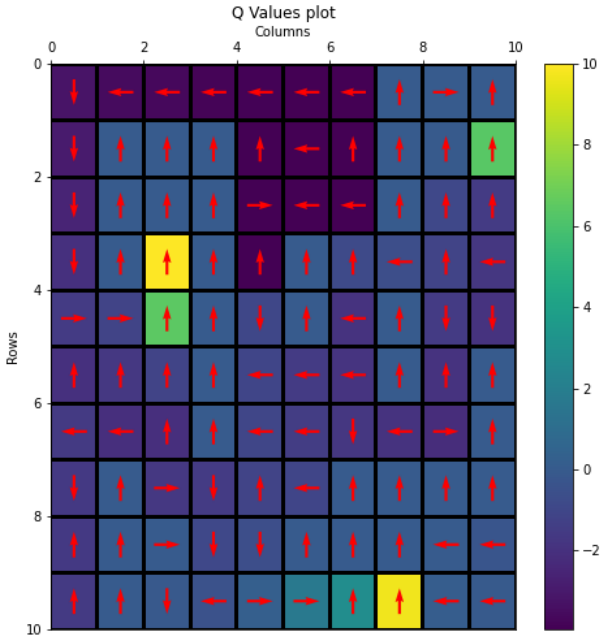




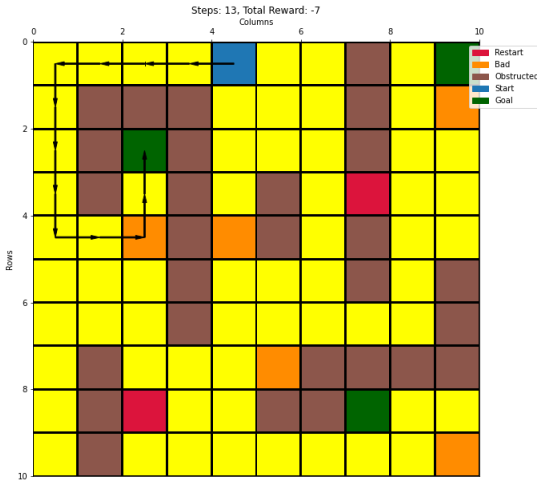
Heatmap with state visit counts



Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy



Justifications and Inferences

- In this configuration, the only source of stochasticity is the wind.
- The nearest goal state is at $(2, 2)$ for this start state. The wind will be against the agent when moving along the first row. We can see that the rightward wind helps the agent move diagonally in rendering 2 and reduces the negative reward earned.

Configuration 18

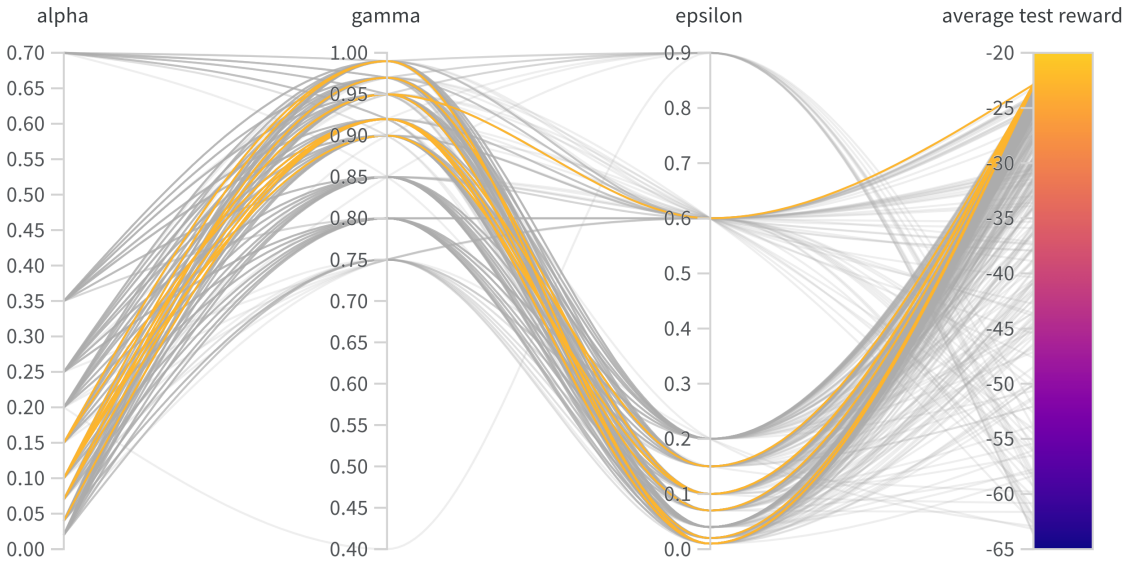
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 0.7

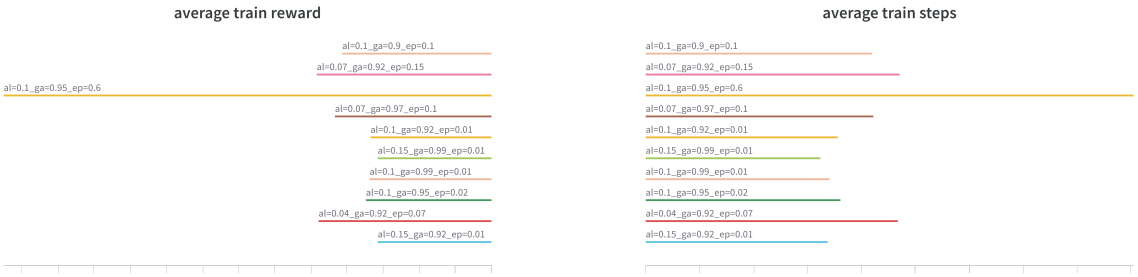
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

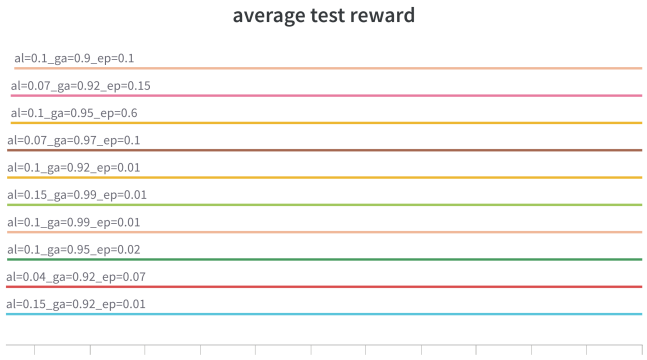
Parallel Co-ordinates Plot

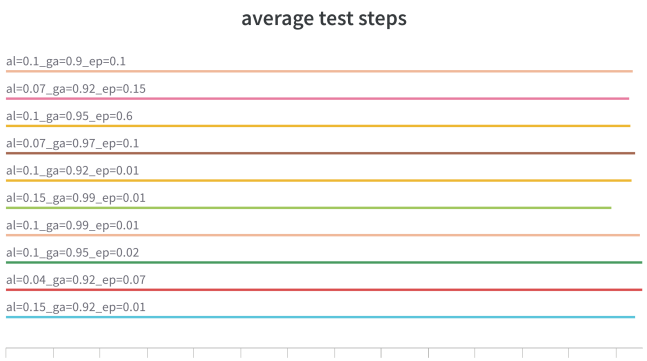


Recorded Metrics



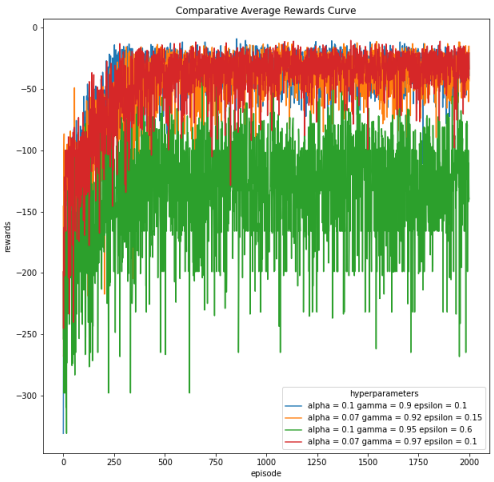
Train Metrics

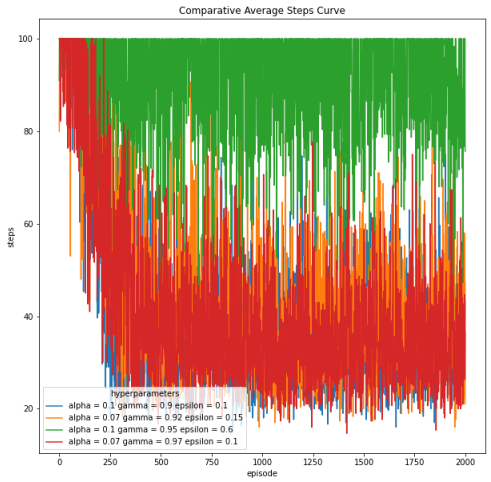




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



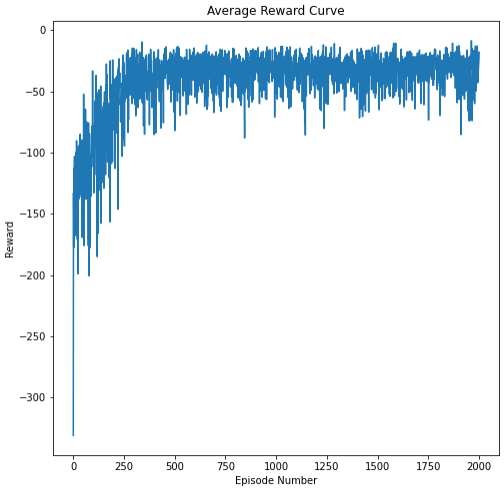


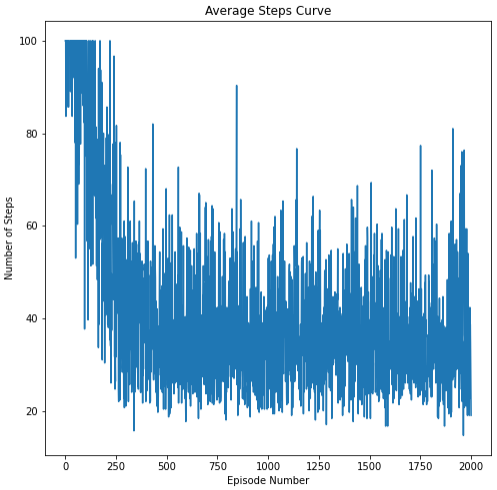
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.1, 0.9, 0.1)$ are the best hyper-parameters according to our sweep.

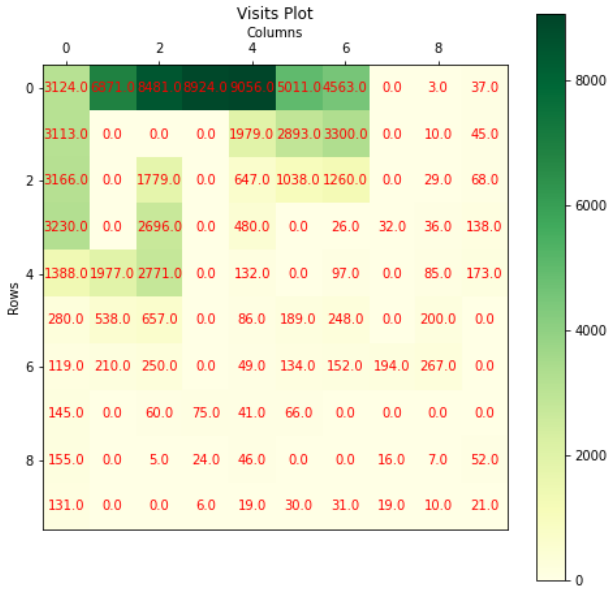
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

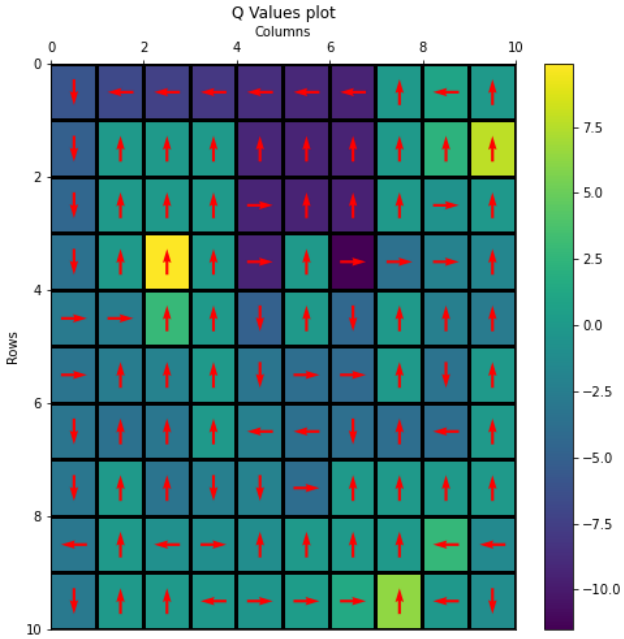




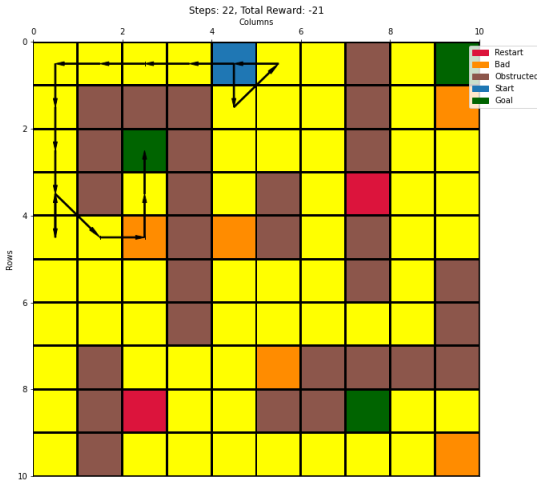
Heatmap with state visit counts

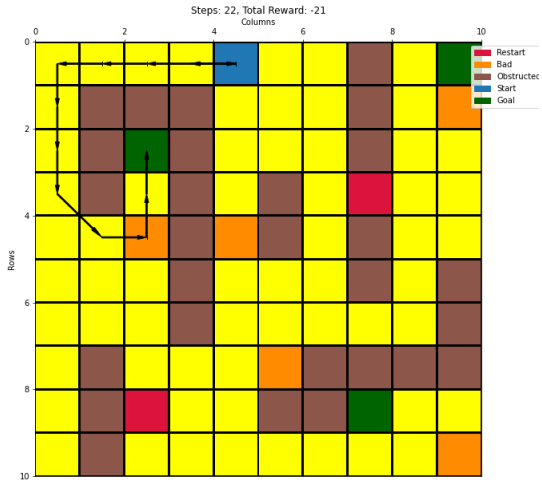


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- In this configuration, there is chance of action failure. This causes large fluctations in the reward and steps curve. This causes agent to go around in loops as well.
- We can see that the wind here will sometimes support the agent and sometimes oppose it in its way to $(2, 2)$.
- Once again, the agent chooses the nearest goal state.

Configuration 19

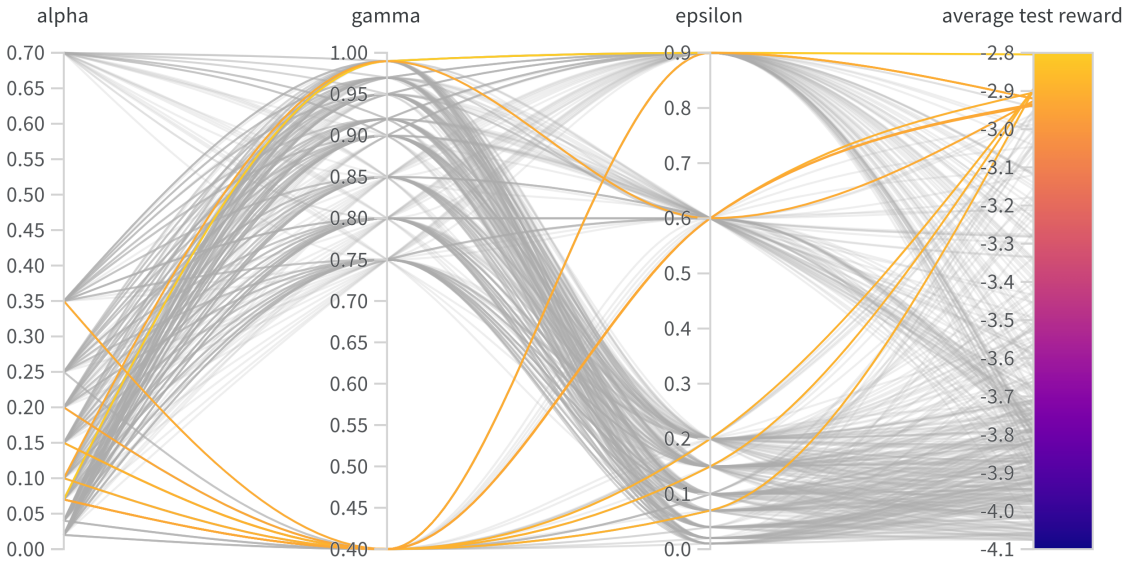
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 1.0

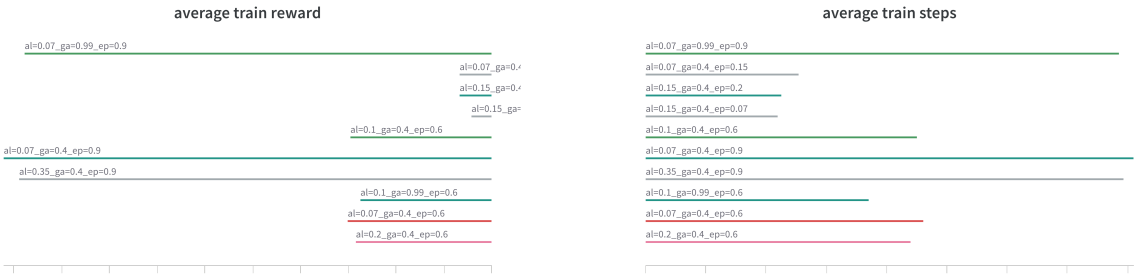
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



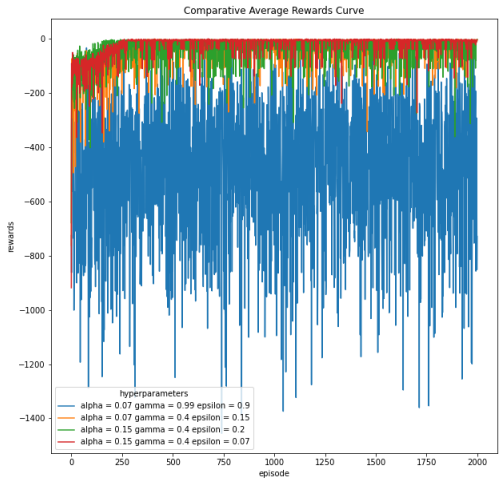
Recorded Metrics

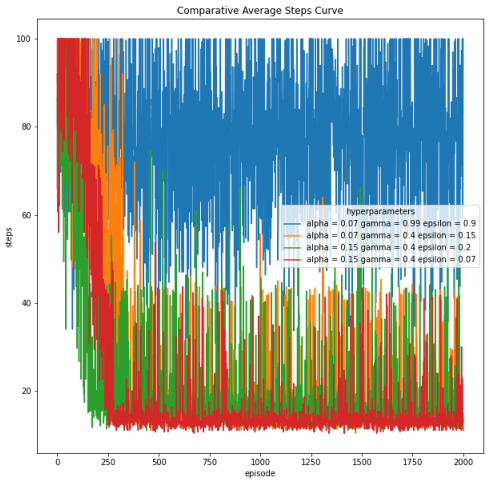


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



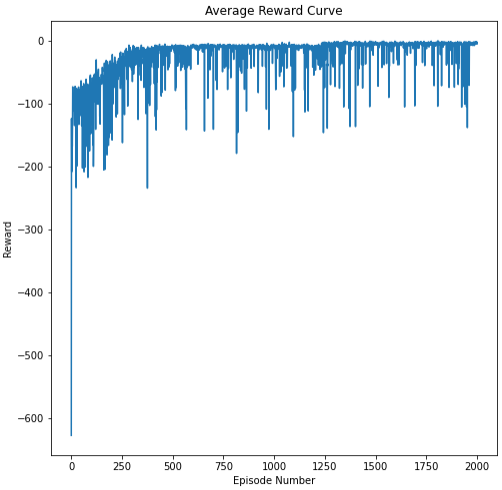


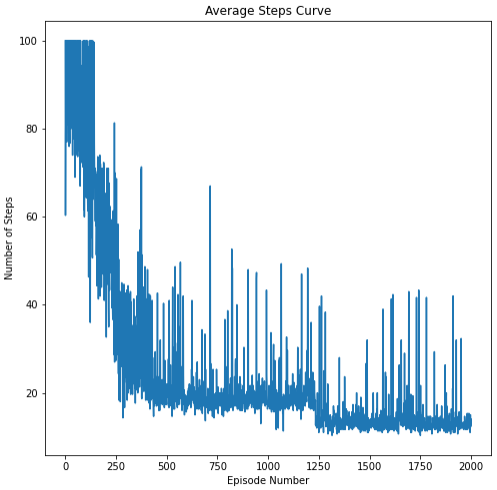
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.15, 0.4, 0.07)$ are the best hyper-parameters according to our sweep.

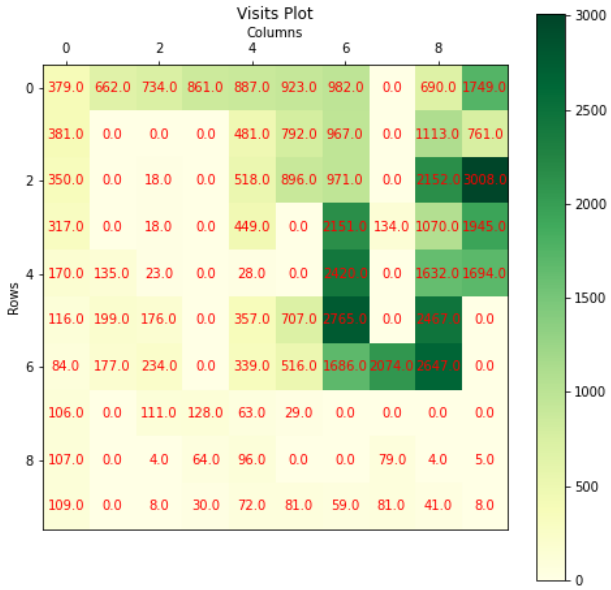
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

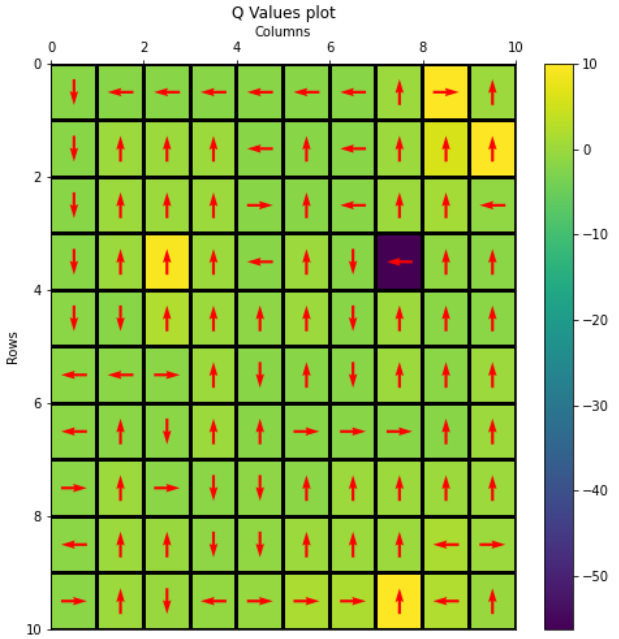




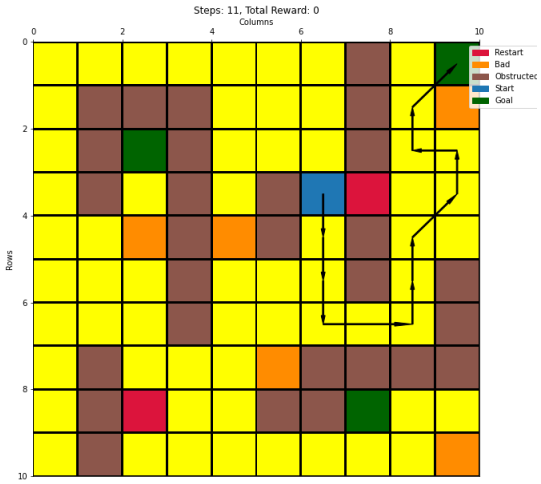
Heatmap with state visit counts

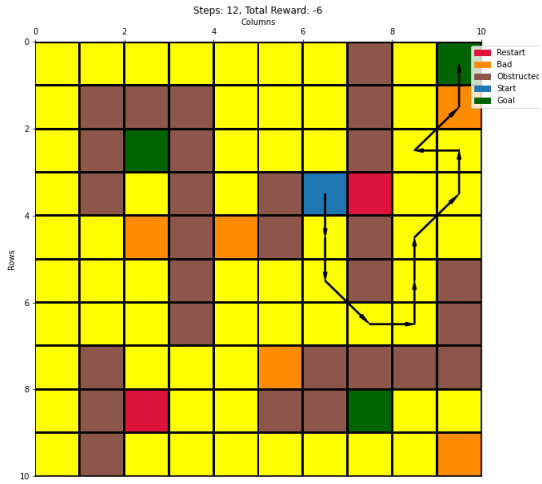


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present. Even though the goal states at $(0, 9)$ and $(8, 7)$ are equidistant from the start state, the agent biases towards $(0, 9)$ because of the wind.
- Because of the wind, the agent gets pushed to the last column and is forced to go through the bad state below the goal.
- The agent may be pushed into the restart state next to the start state and get a large negative reward. This has not happened in the above renderings but it may happen.

Configuration 20

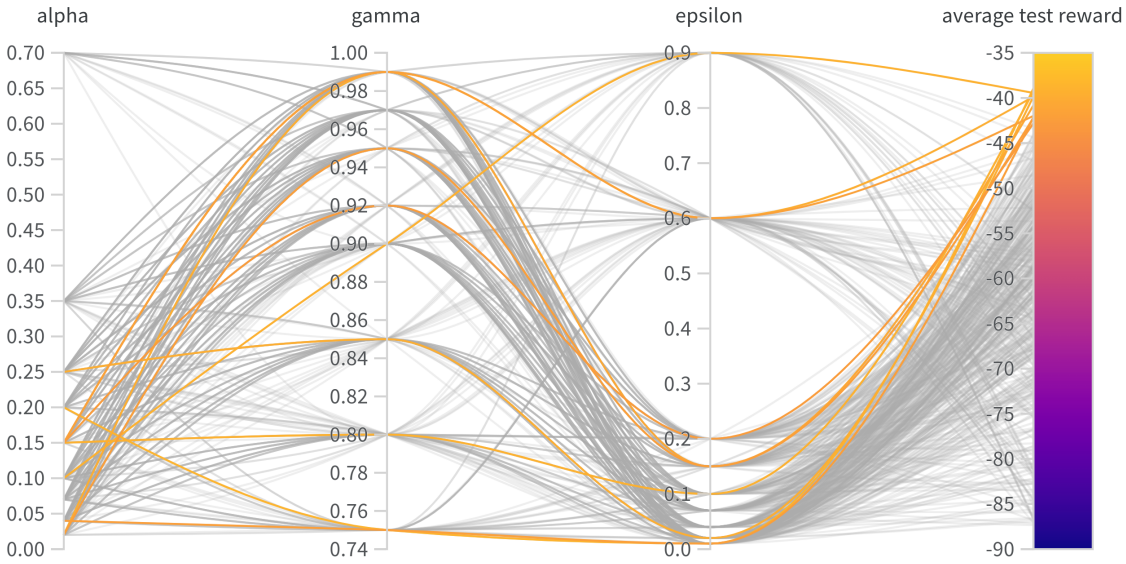
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 0.7

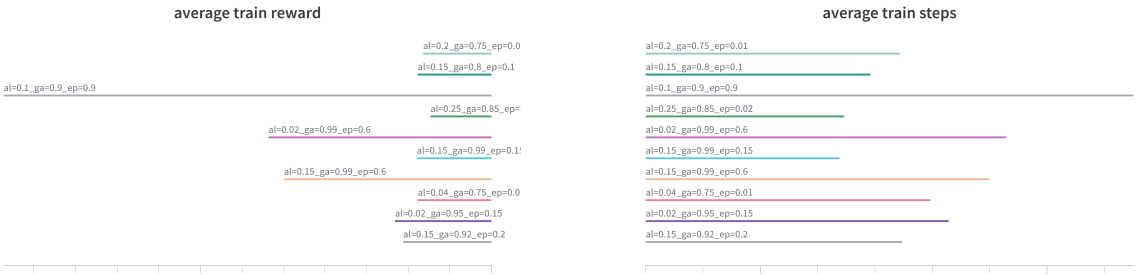
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



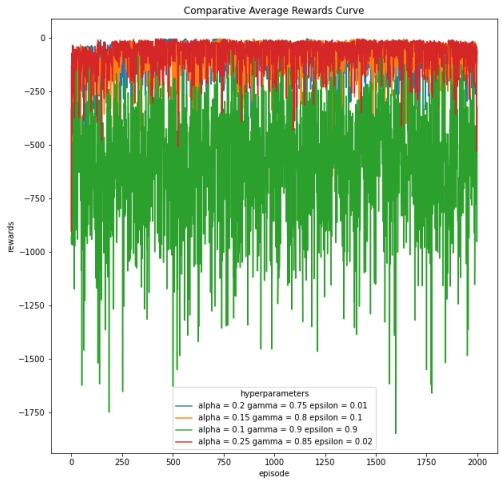
Recorded Metrics

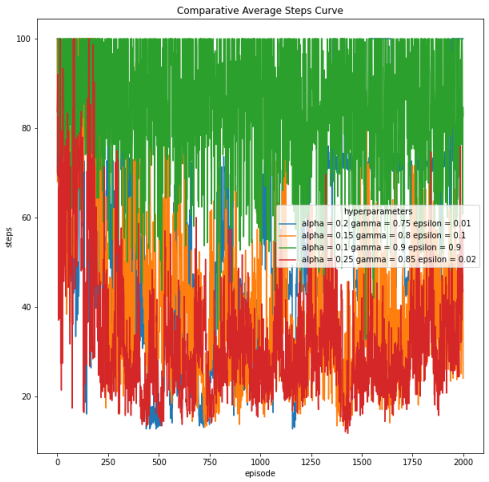


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



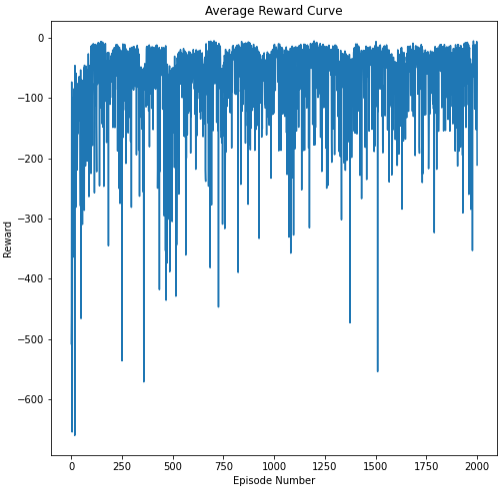


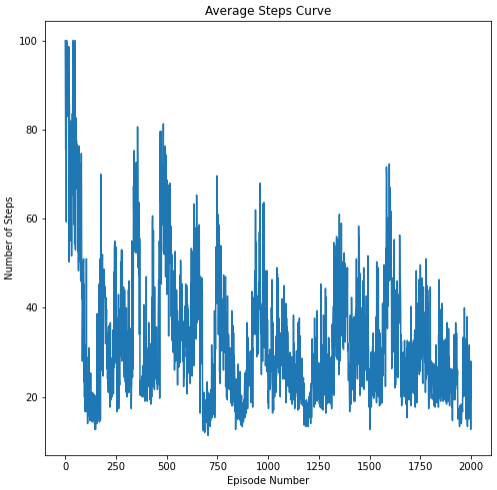
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.25, 0.85, 0.02)$ are the best hyper-parameters according to our sweep.

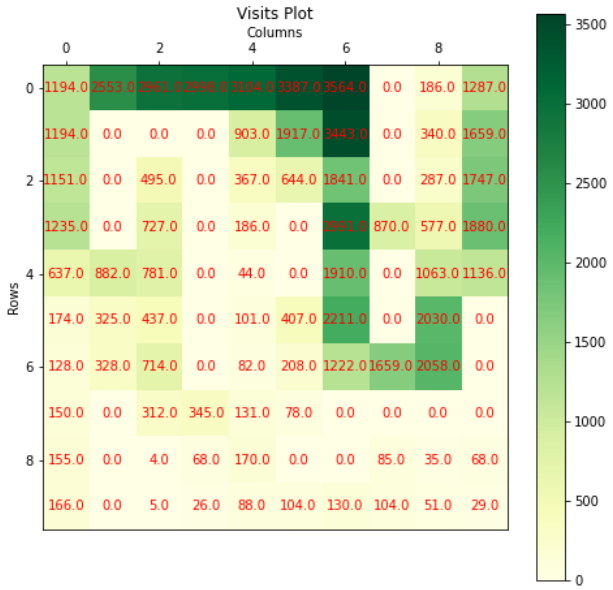
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

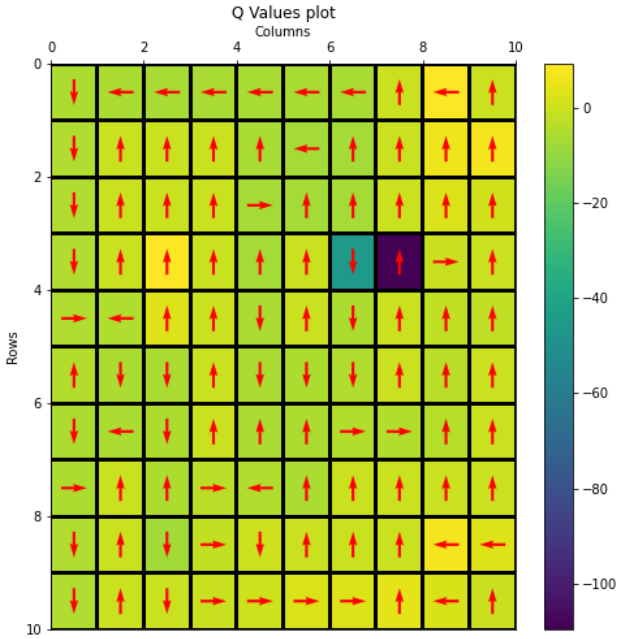




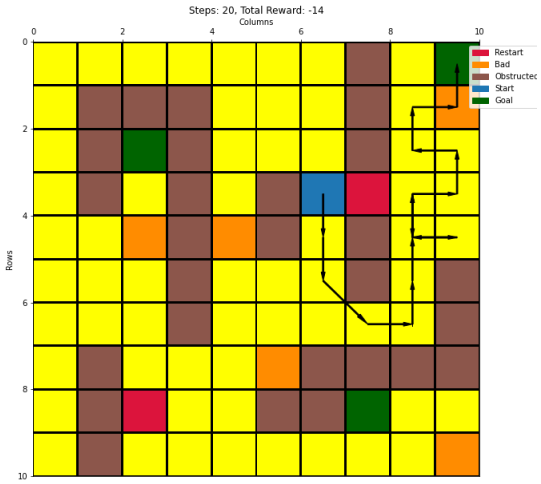
Heatmap with state visit counts

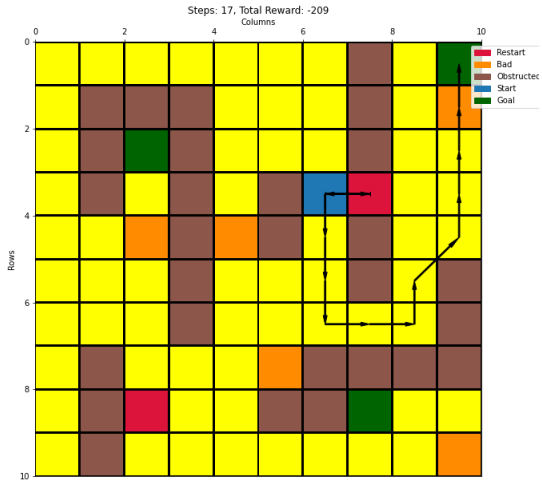


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present along with the possibility of action failure. This is the reason for the large fluctations in the reward and steps curves once again.
- Because of the wind, the agent sometimes moves into the restart state next to the start state resulting in a very high negative reward, as in the renderings. Wind also causes bias towards $(0, 9)$.
- The action failure probability also causes the agent to take longer routes. The agent also has explored the first row significantly many times because action failure has no effect there.

Configuration 21

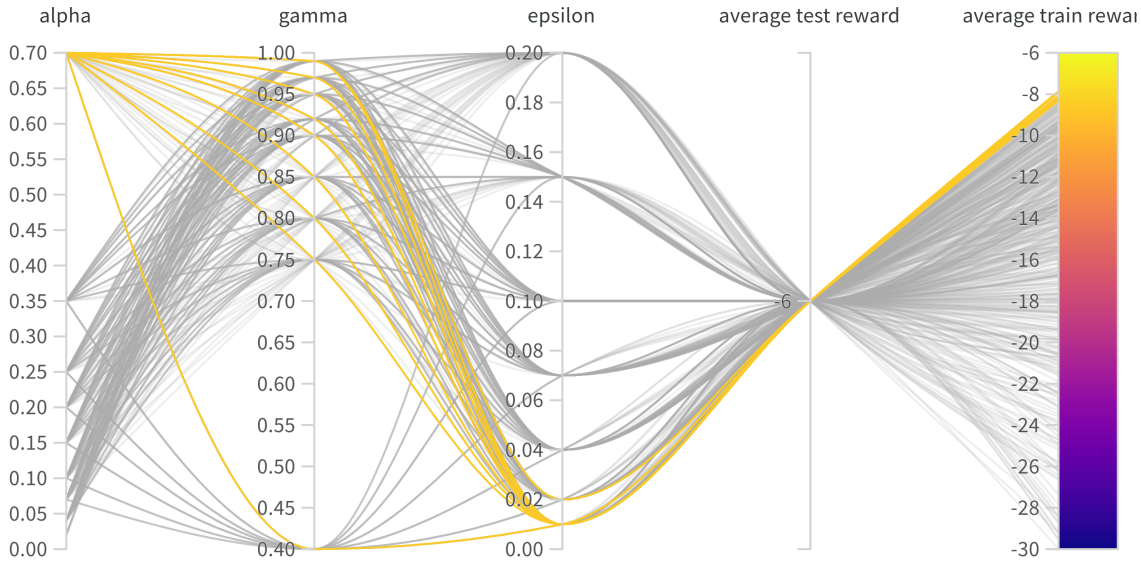
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 1.0

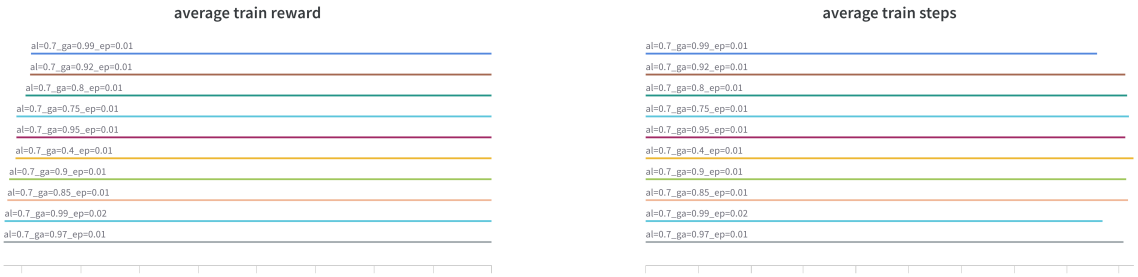
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

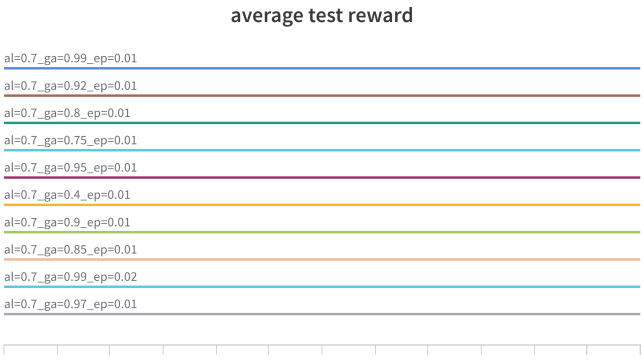
Parallel Co-ordinates Plot



Recorded Metrics

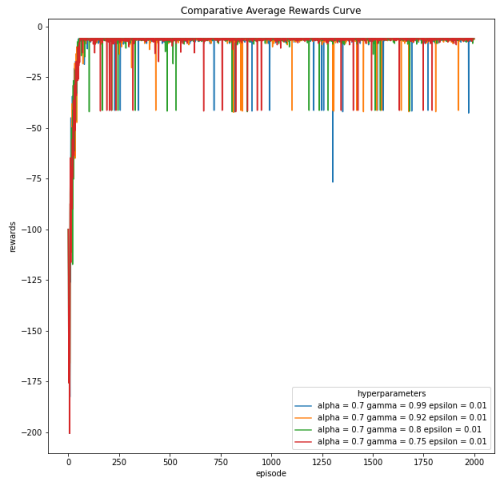


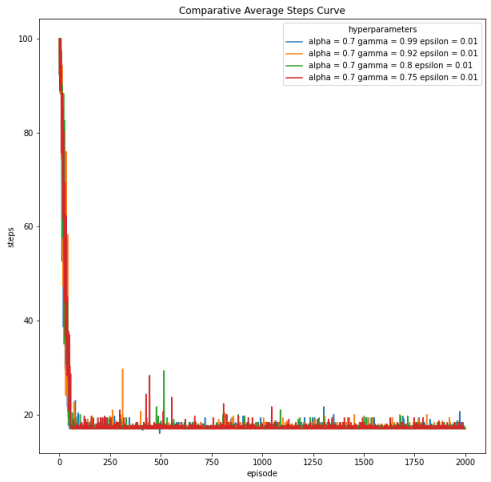
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



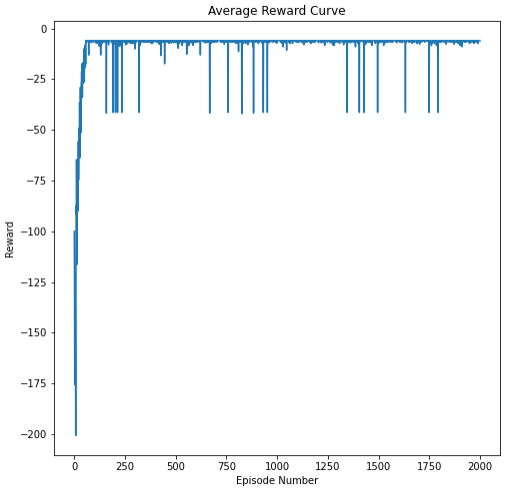


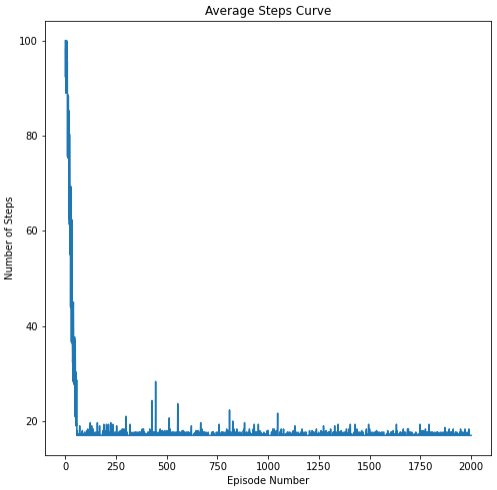
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.7, 0.75, 0.01)$ are the best hyper-parameters according to our sweep.

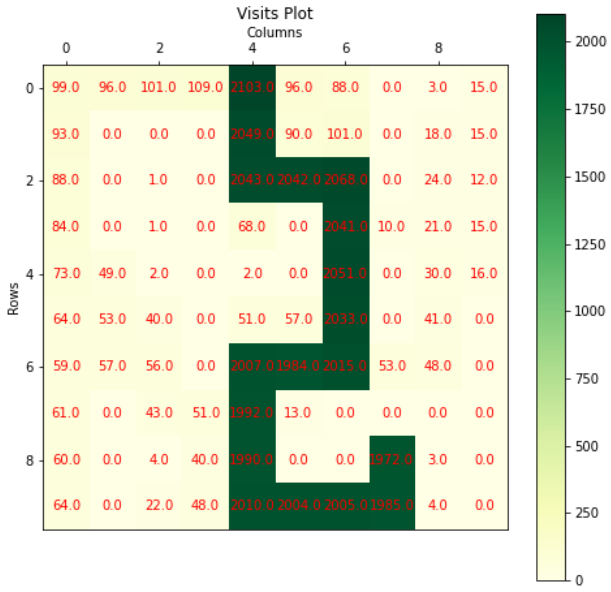
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

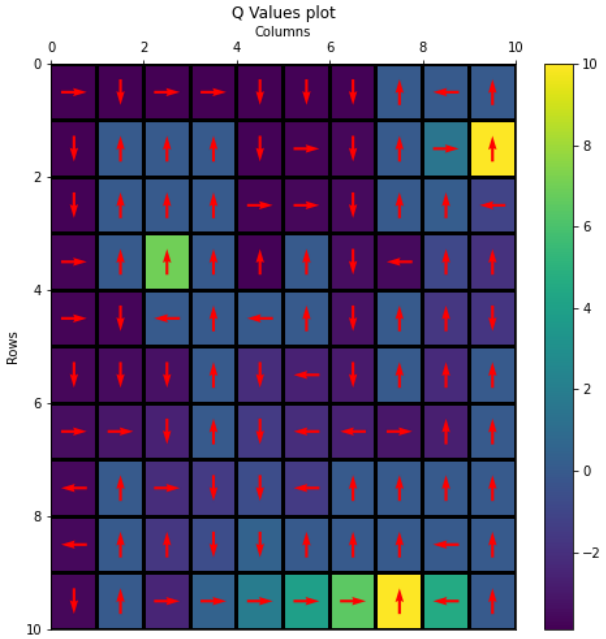




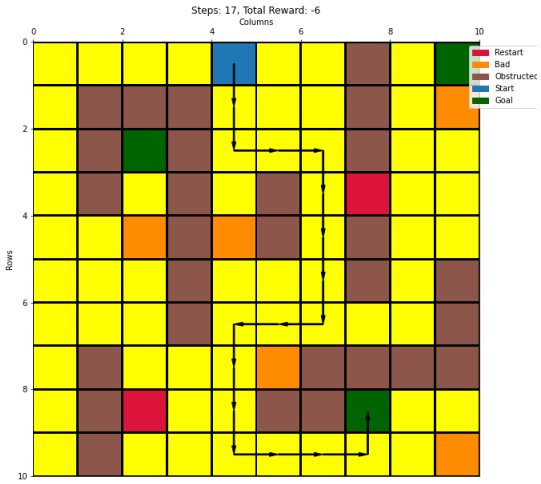
Heatmap with state visit counts

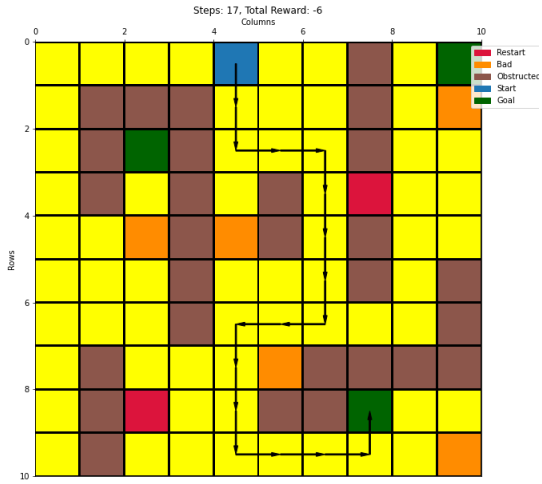


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path towards the goal (8,7).

Configuration 22

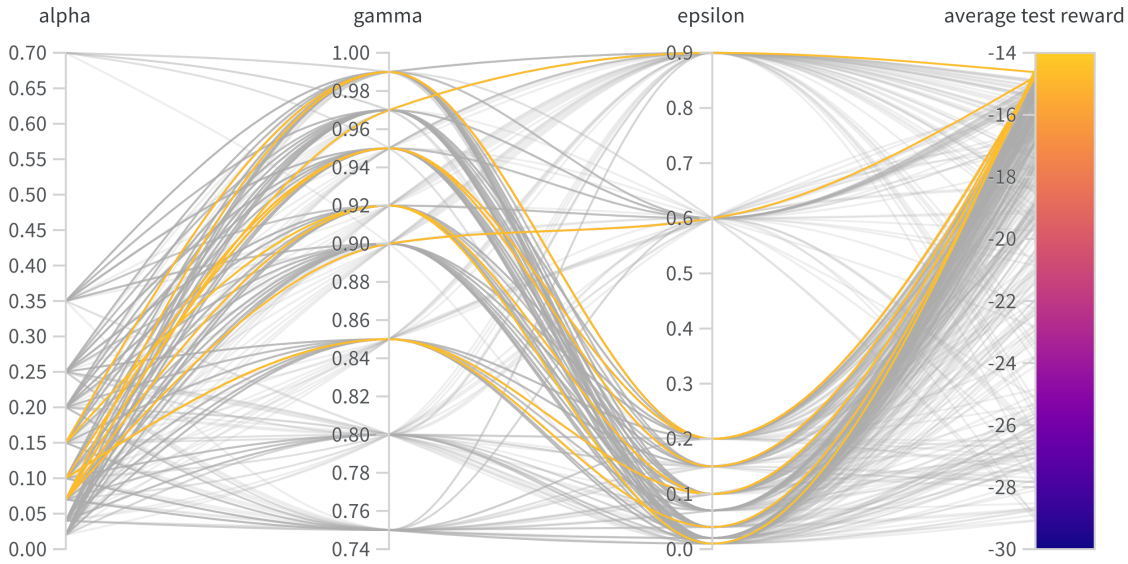
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 0.7

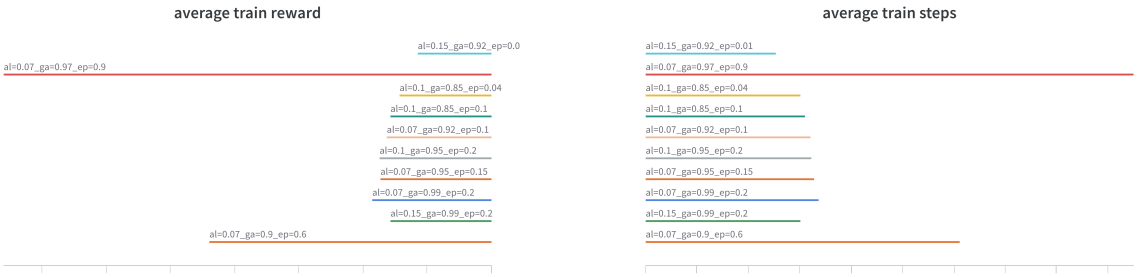
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

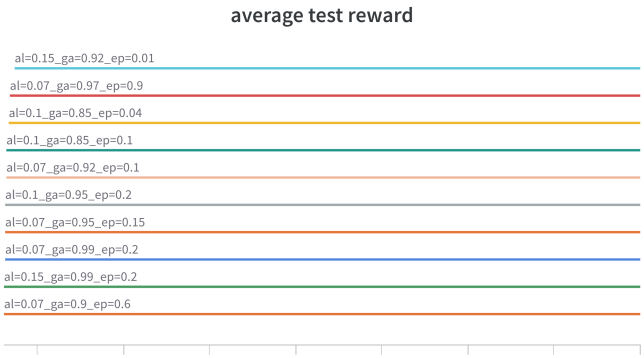
Parallel Co-ordinates Plot



Recorded Metrics

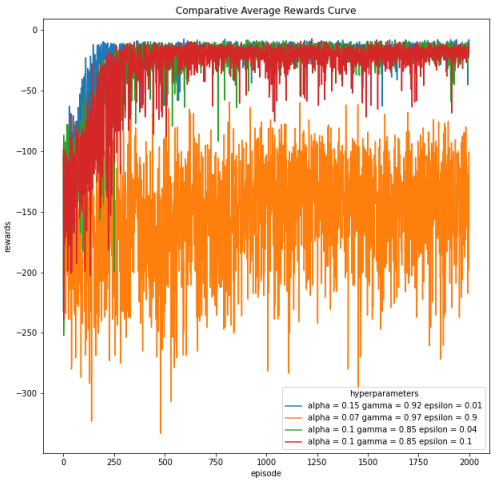


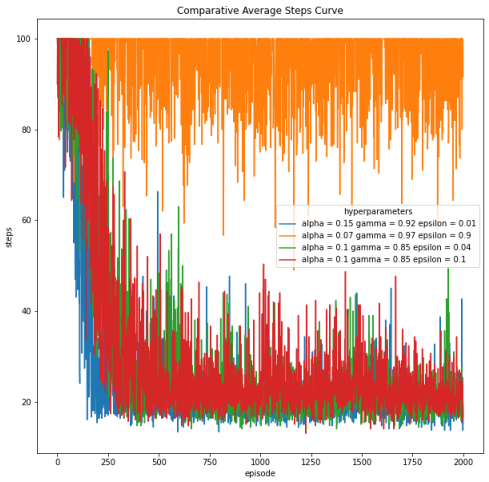
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



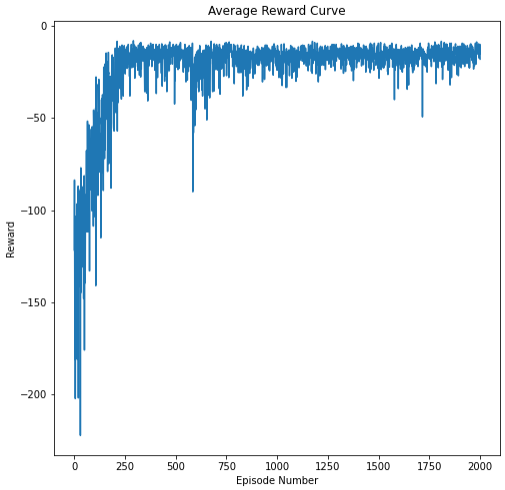


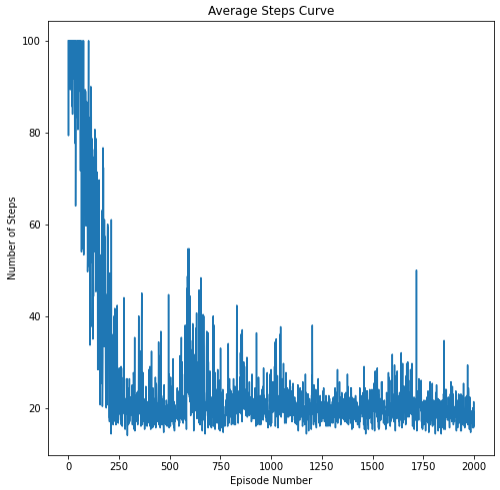
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.15, 0.92, 0.01)$ are the best hyper-parameters according to our sweep.

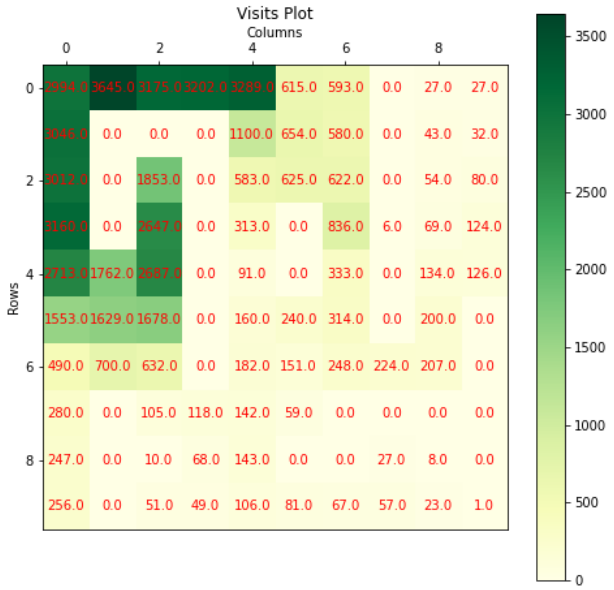
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

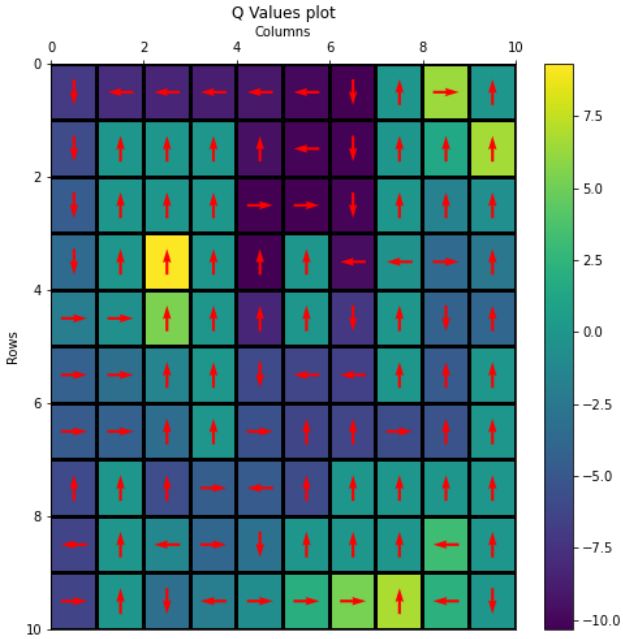




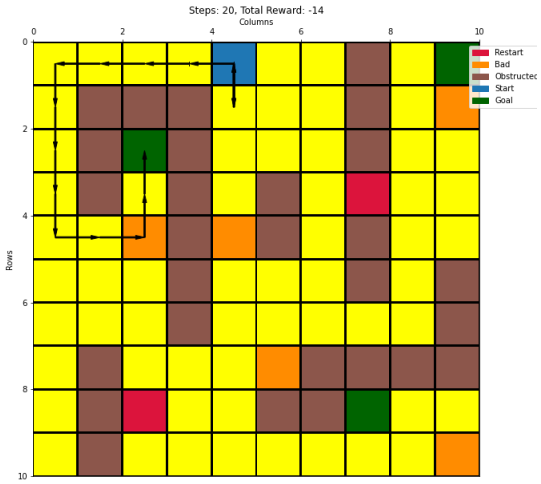
Heatmap with state visit counts

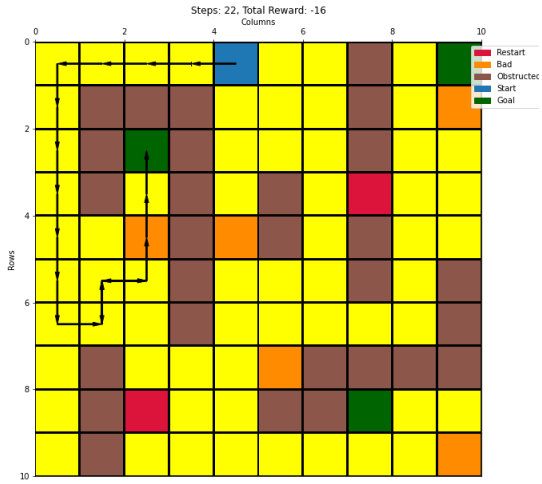


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but there is action failure. This is the reason for large fluctuations in the reward and steps curve.
- The agent always tries to take the path to $(2, 2)$ directly. But action failure may increase the number of steps.

Configuration 23

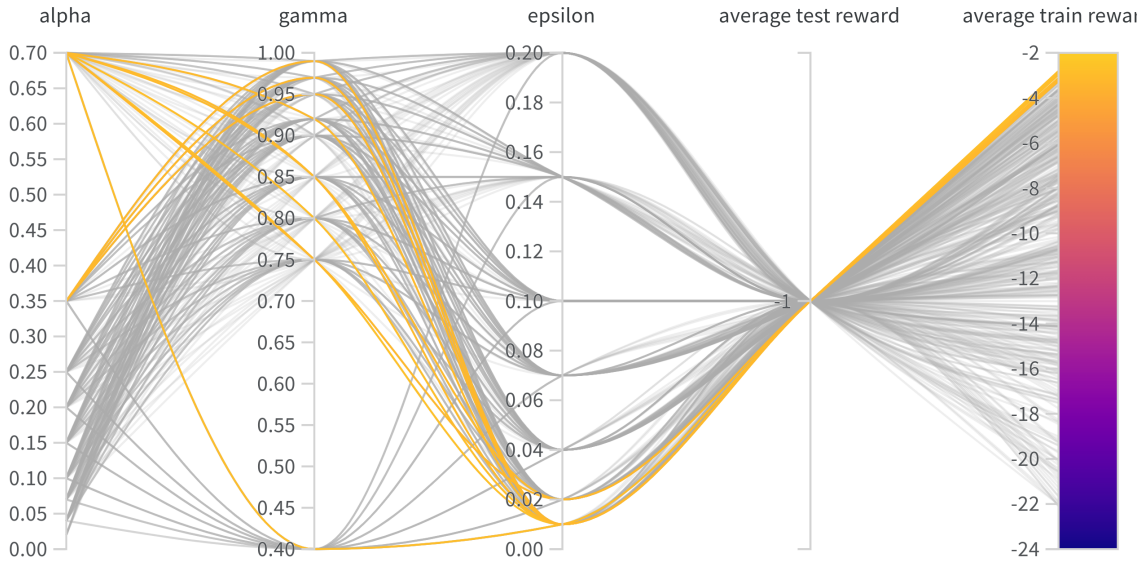
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 1.0

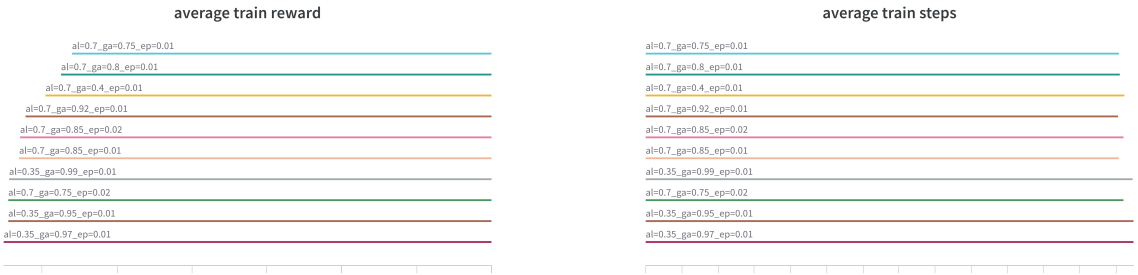
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



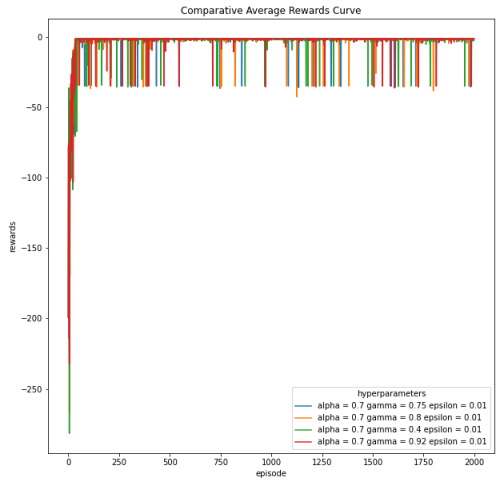
Recorded Metrics

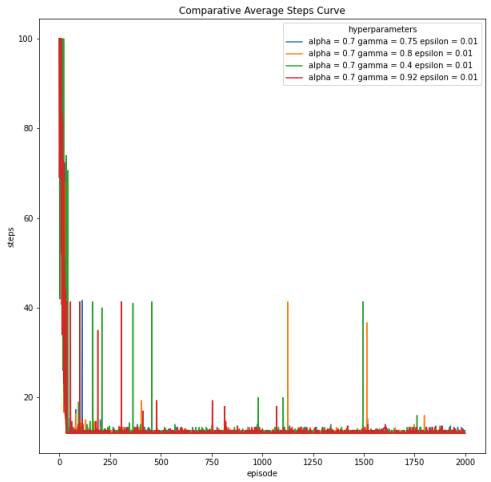


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



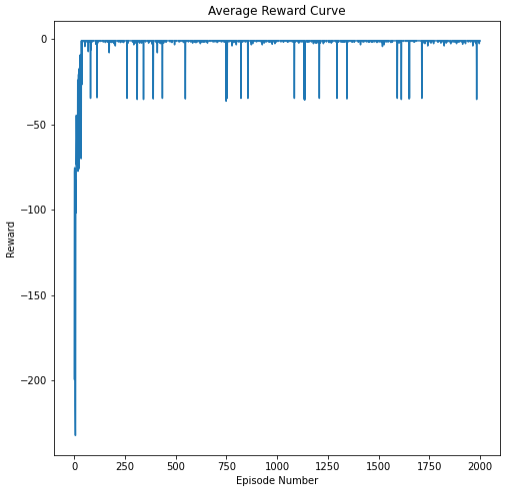


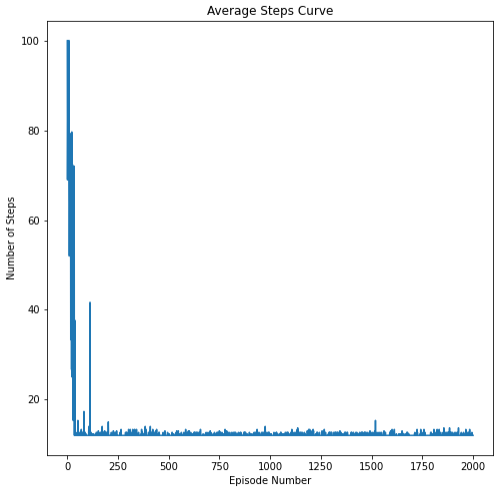
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.7, 0.75, 0.01)$ are the best hyper-parameters according to our sweep.

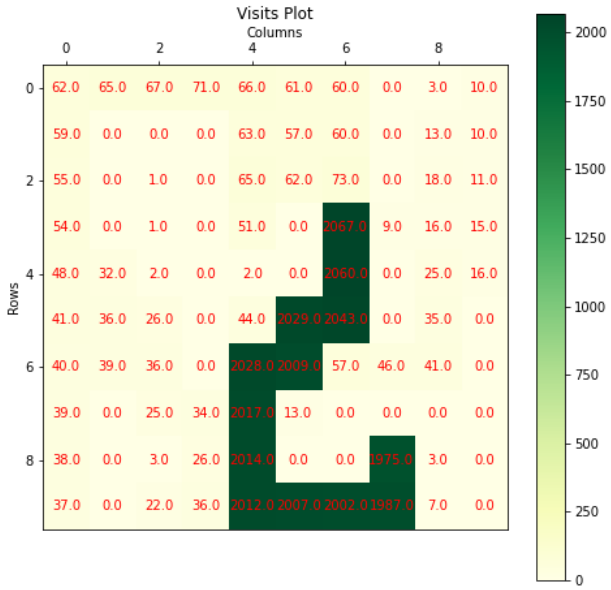
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

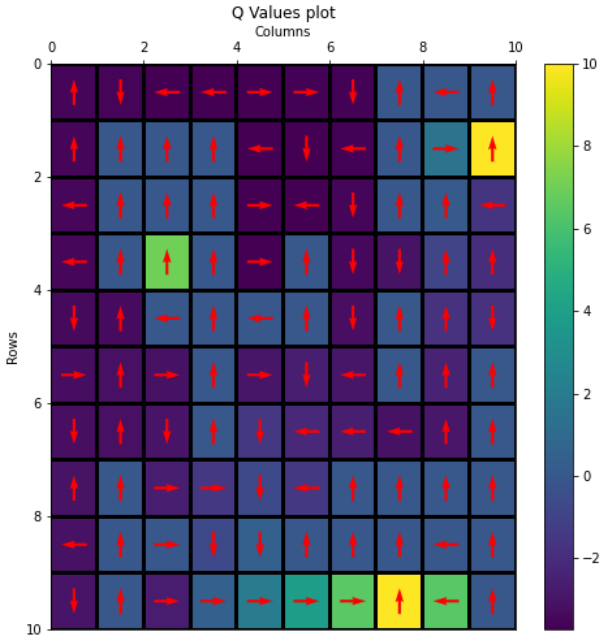




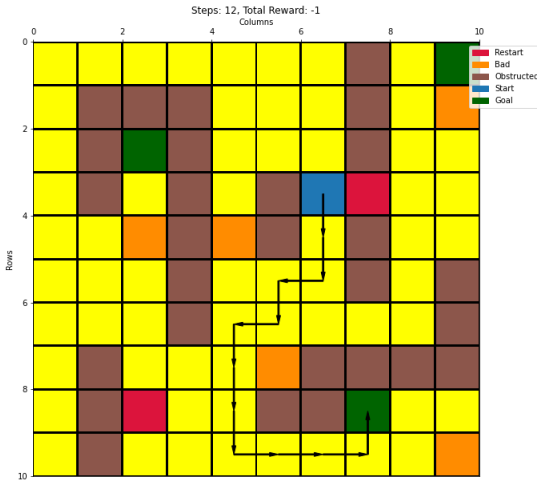
Heatmap with state visit counts

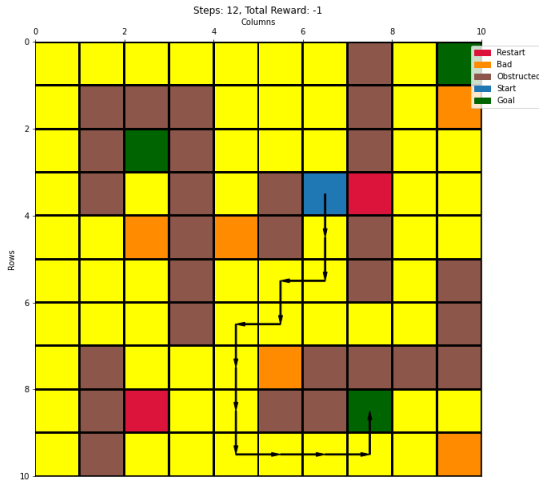


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path to $(8, 7)$.

Configuration 24

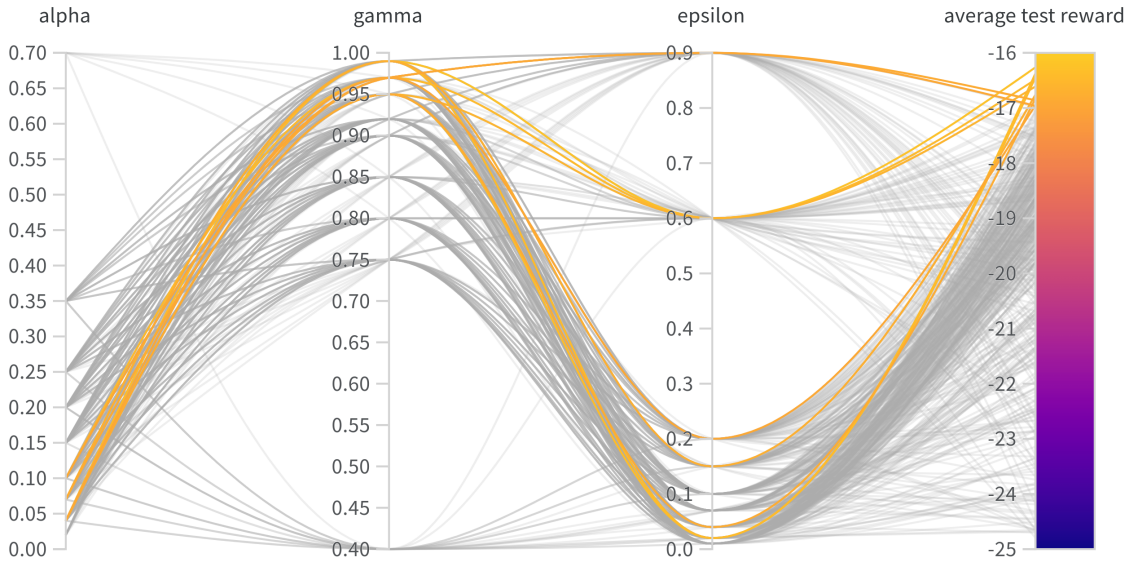
Configuration Description

- **learning -** Q-Learning
- **action -** Epsilon Greedy Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 0.7

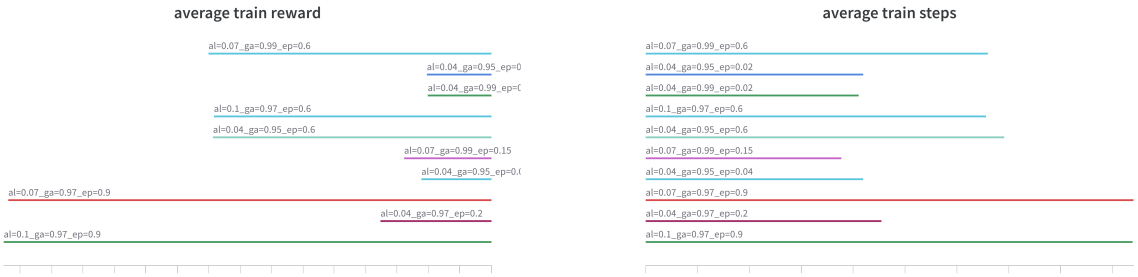
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

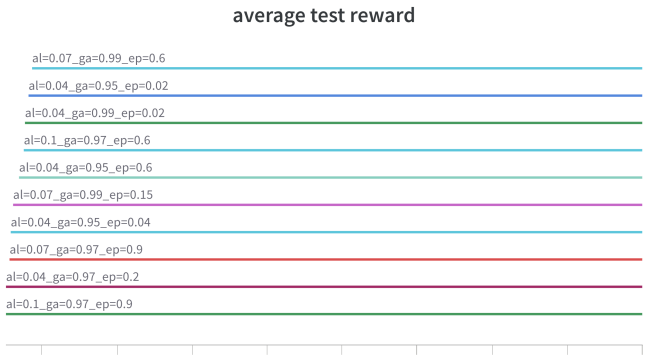
Parallel Co-ordinates Plot

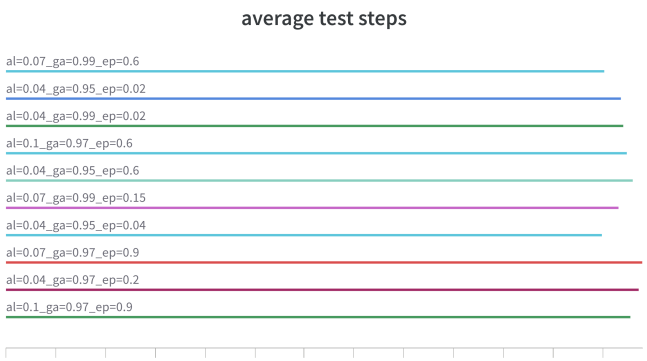


Recorded Metrics



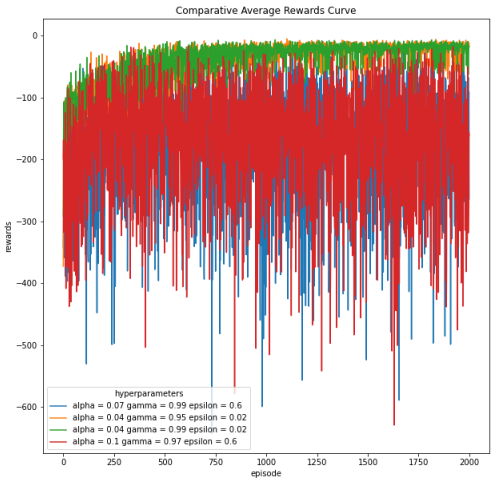
Train Metrics

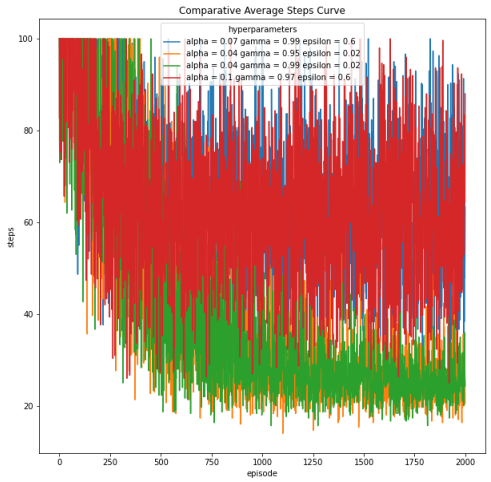




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



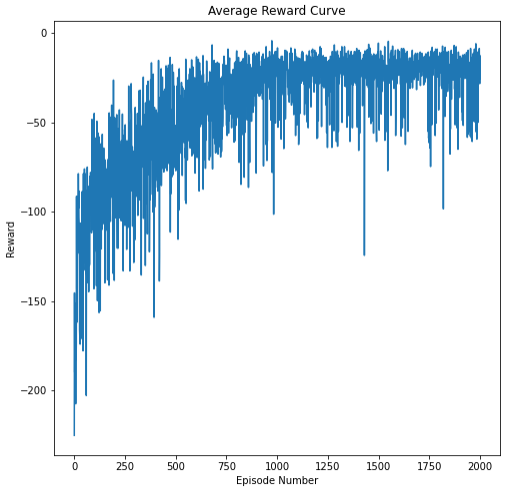


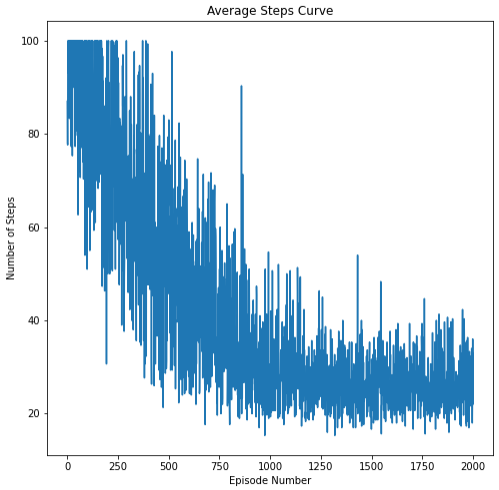
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \epsilon) = (0.04, 0.95, 0.2)$ are the best hyper-parameters according to our sweep.

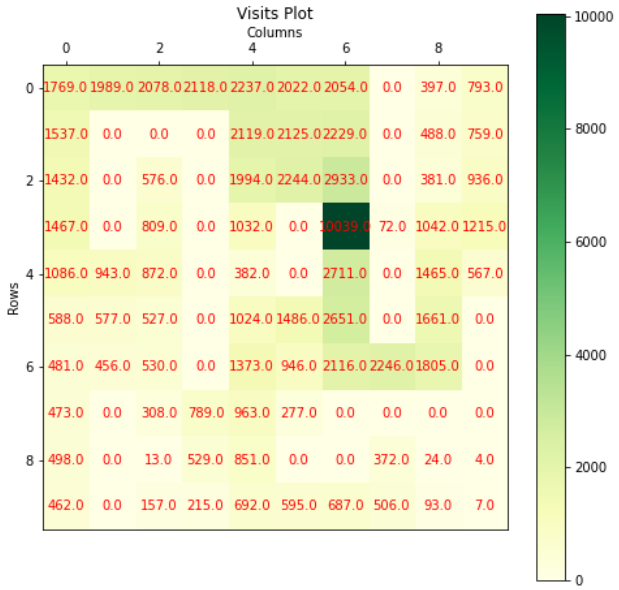
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

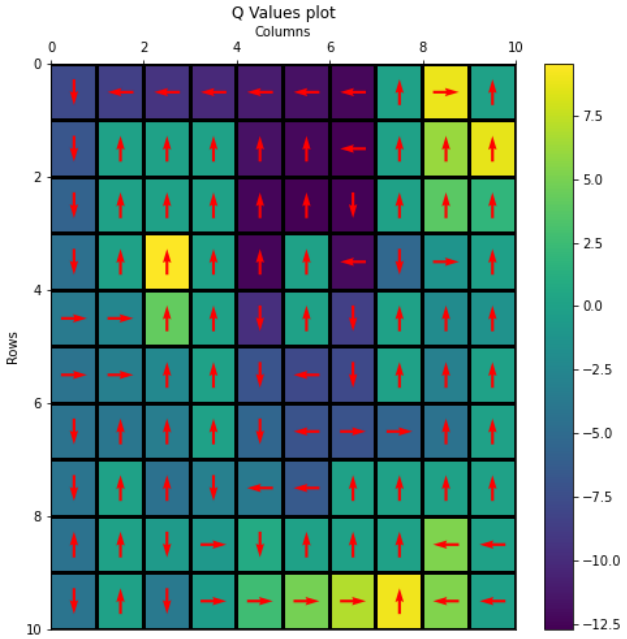




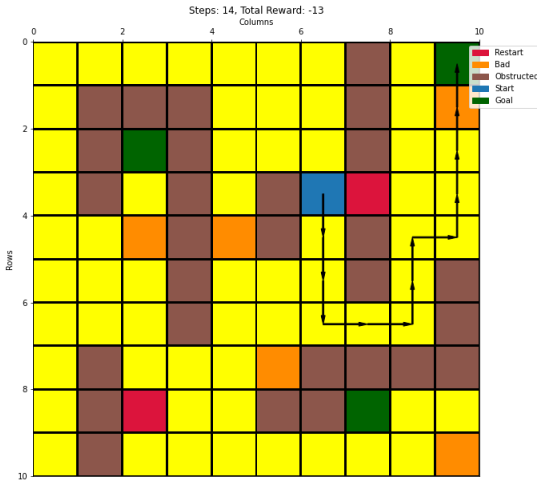
Heatmap with state visit counts

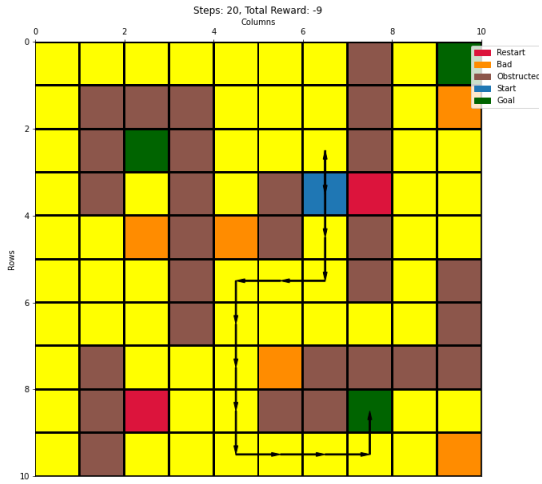


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but there is action failure.
- Action failure is the reason for the large variations in the reward and steps curves.
- From the heat map, all goal states have been visited considerable number of times. Clearly, an action failure event has caused the agent to move to 2 different goal states.

Configuration 25

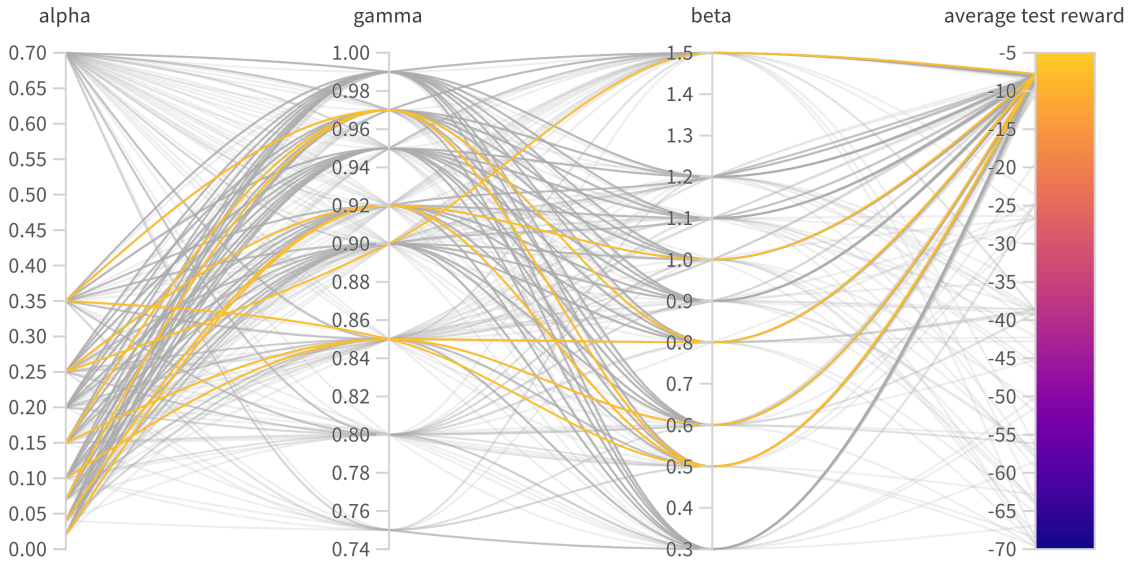
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 1.0

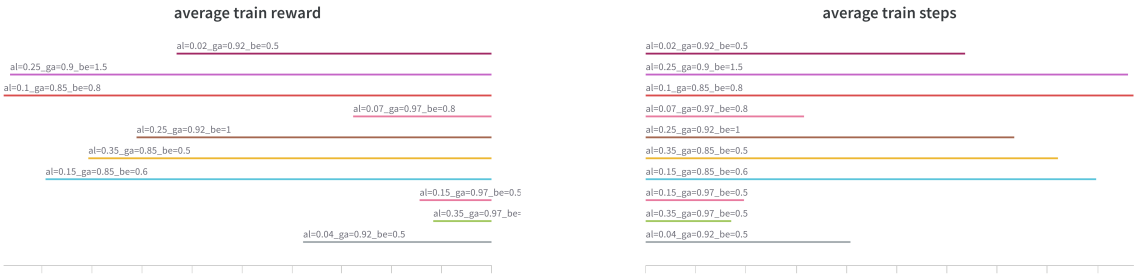
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

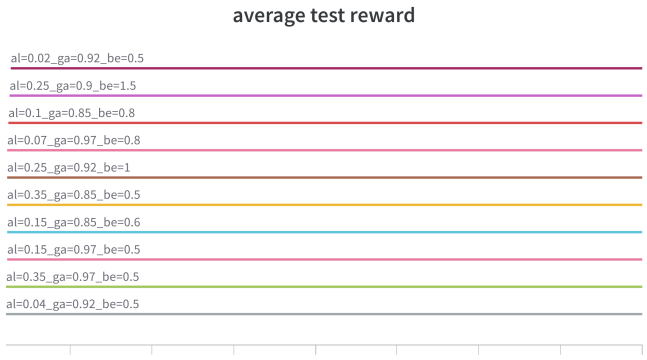
Parallel Co-ordinates Plot



Recorded Metrics

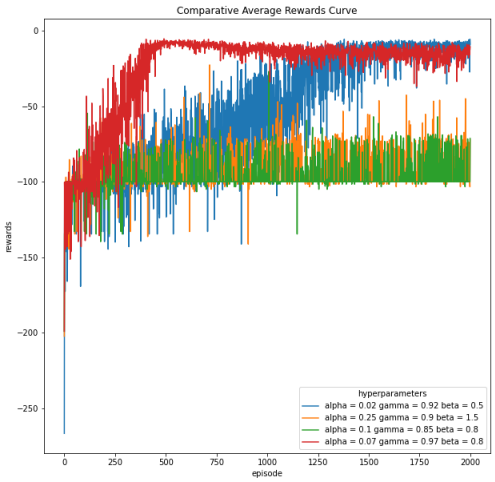


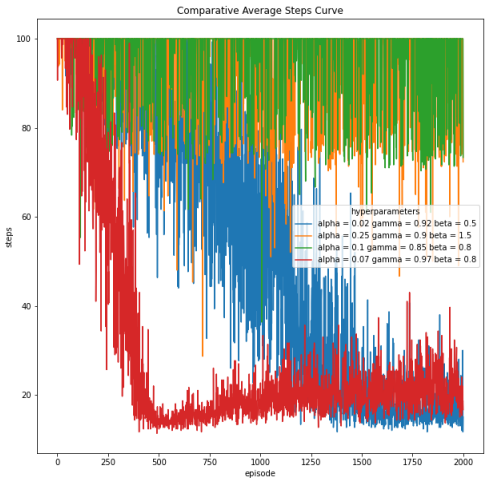
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



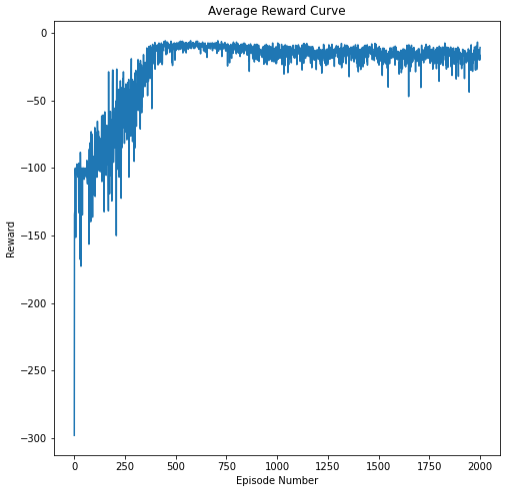


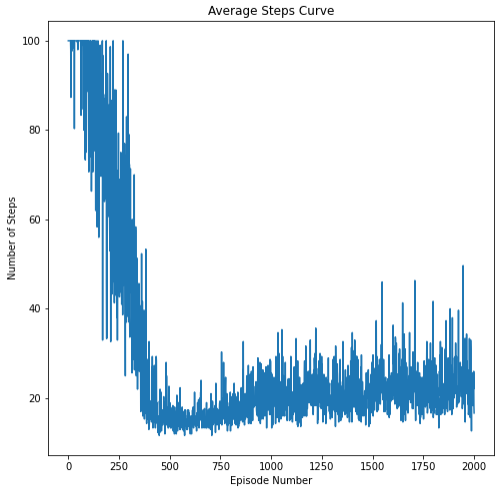
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.07, 0.97, 0.8)$ are the best hyper-parameters according to our sweep.

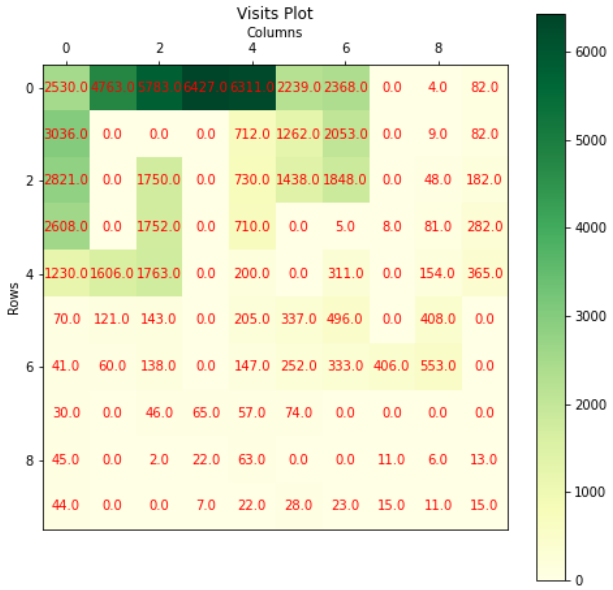
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

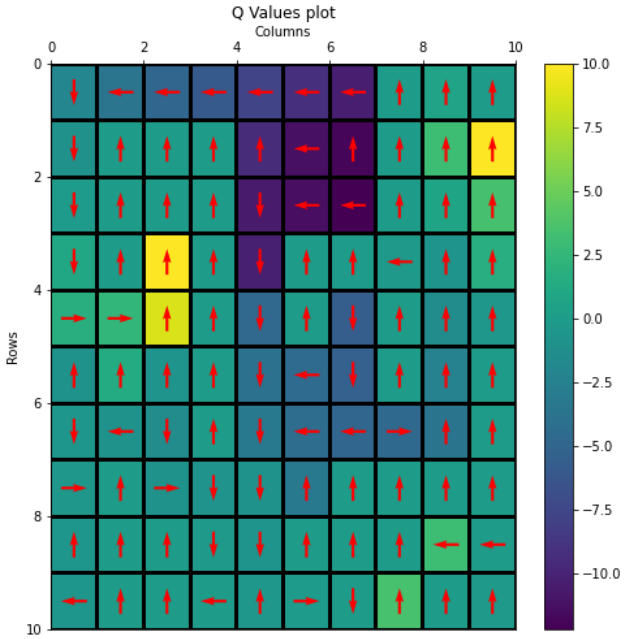




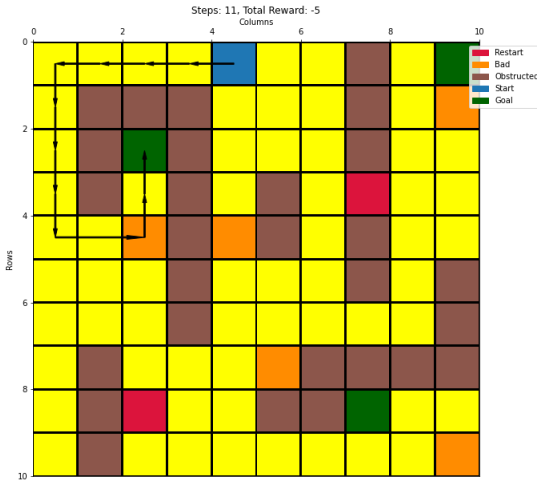
Heatmap with state visit counts



Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy



Justifications and Inferences

- In this configuration, the only source of stochasticity is the wind.
- The nearest goal state is at $(2, 2)$ for this start state. The wind will be against the agent when moving along the first row. We can see that the rightward wind helps the agent move diagonally in rendering 2 and reduces the negative reward earned.

Configuration 26

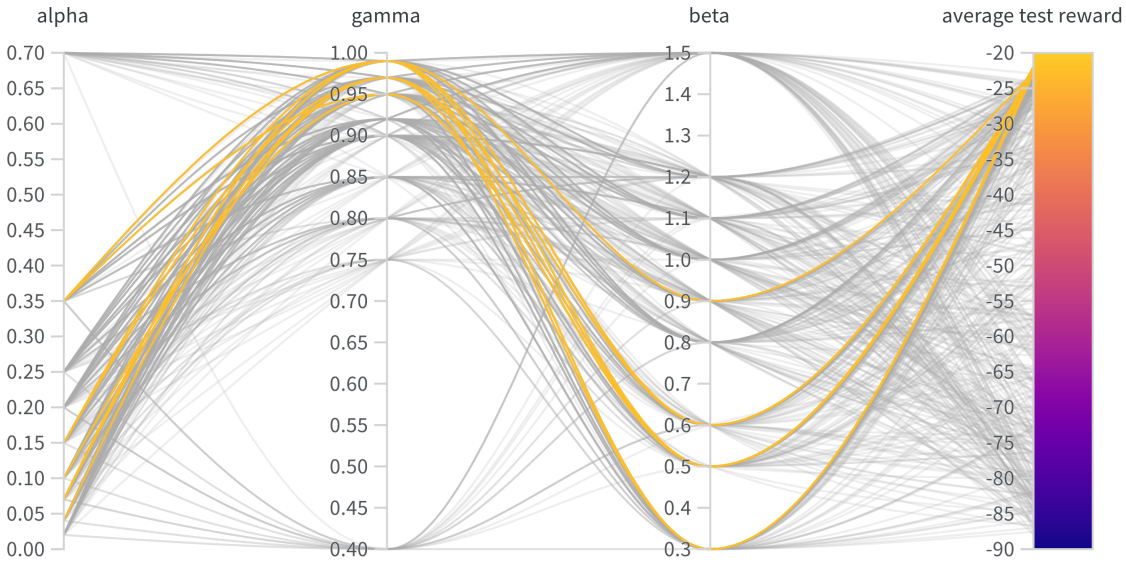
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** True
- **start -** $(0, 4)$
- **p value -** 0.7

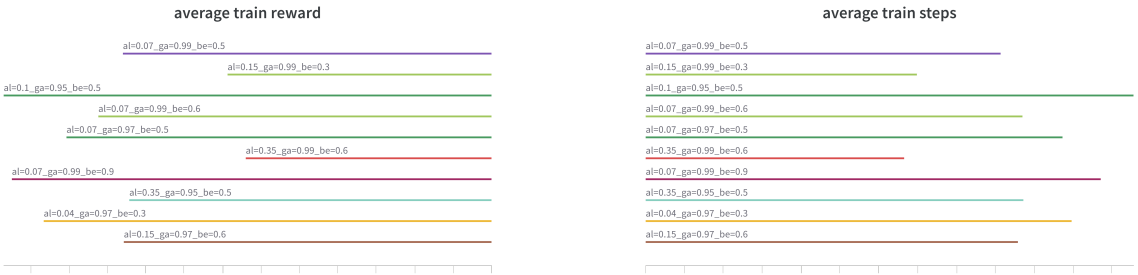
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

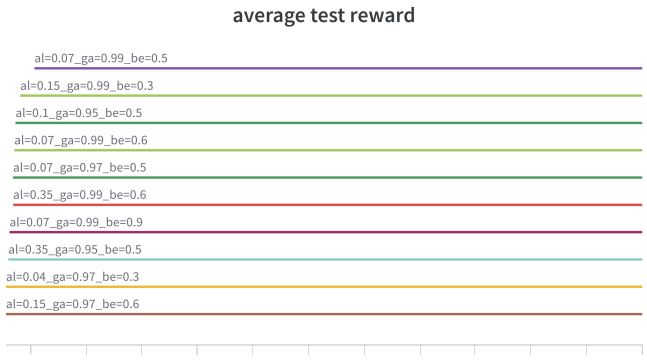
Parallel Co-ordinates Plot

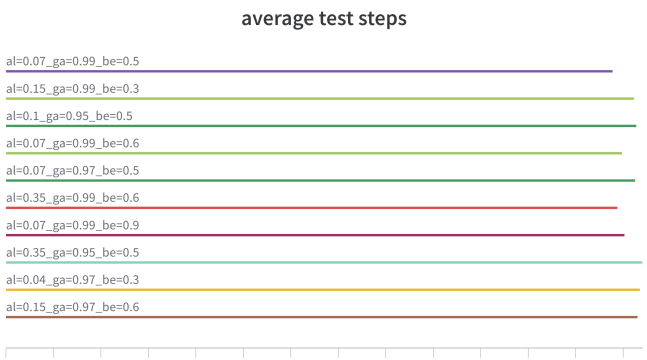


Recorded Metrics



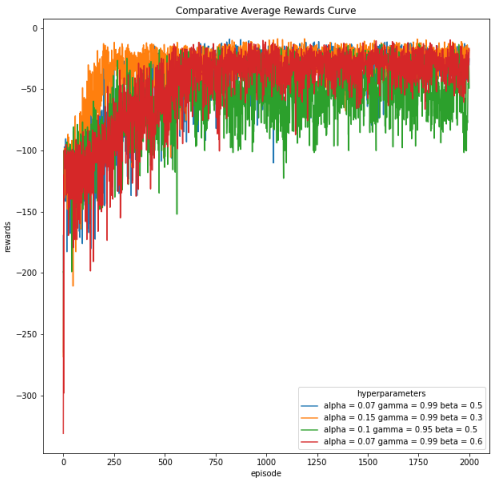
Train Metrics

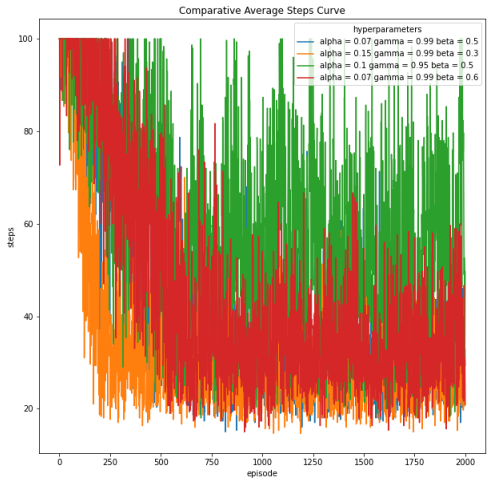




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



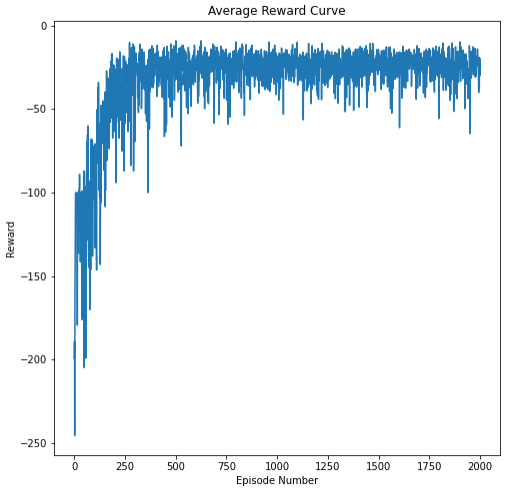


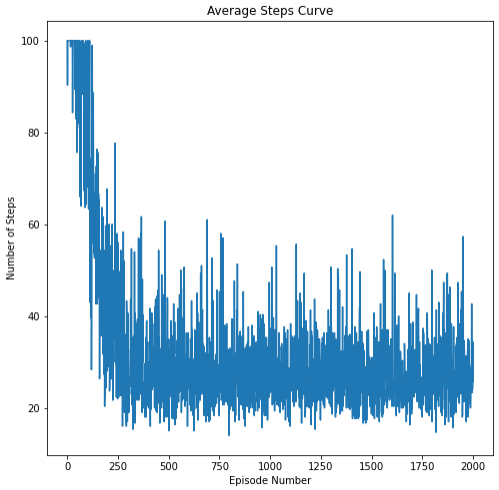
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.15, 0.99, 0.3)$ are the best hyper-parameters according to our sweep.

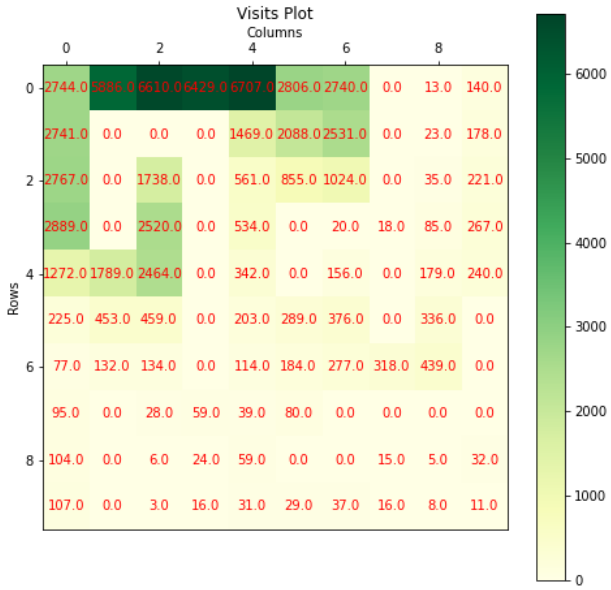
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

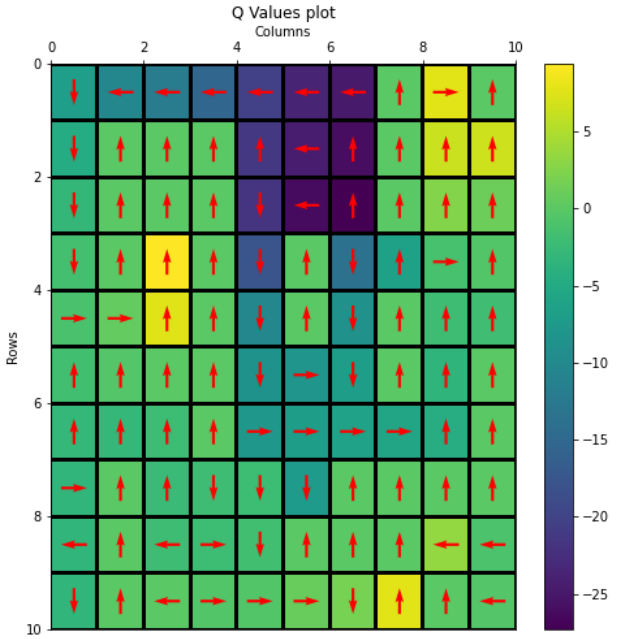




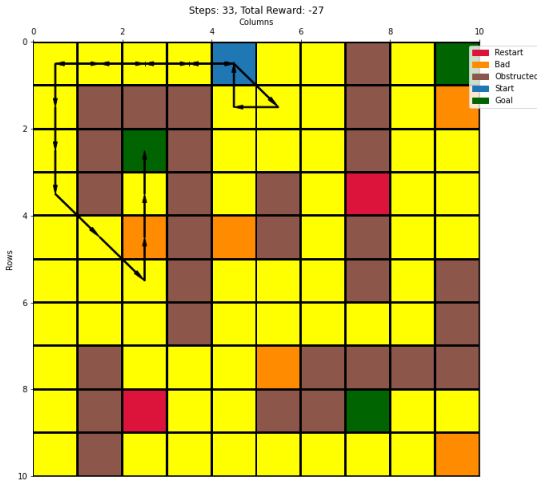
Heatmap with state visit counts

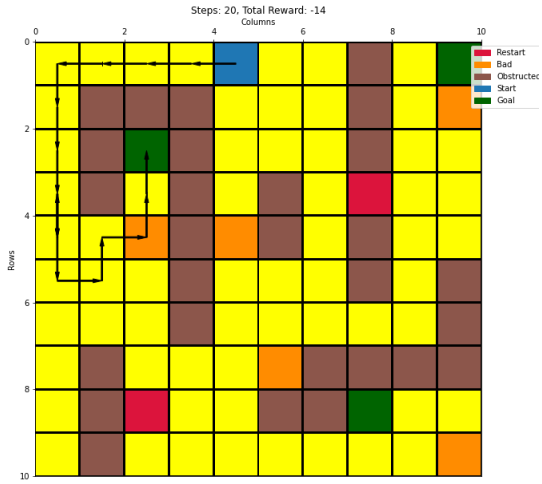


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here there is wind and action failure chance. This causes large fluctations in the reward and steps curve.
- We can see that the wind here will sometimes support the agent and sometimes oppose it.
- From the renderings, we can see that the effect of action failure is apparent as it has gone in loops. The effect of winds is also visible from the diagonal movements.

Configuration 27

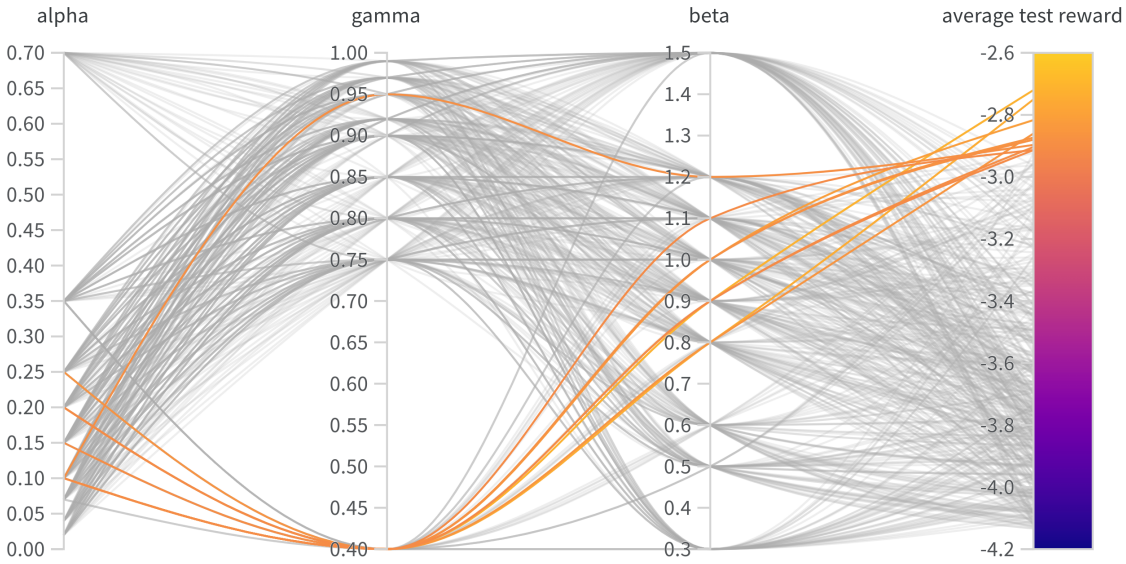
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 1.0

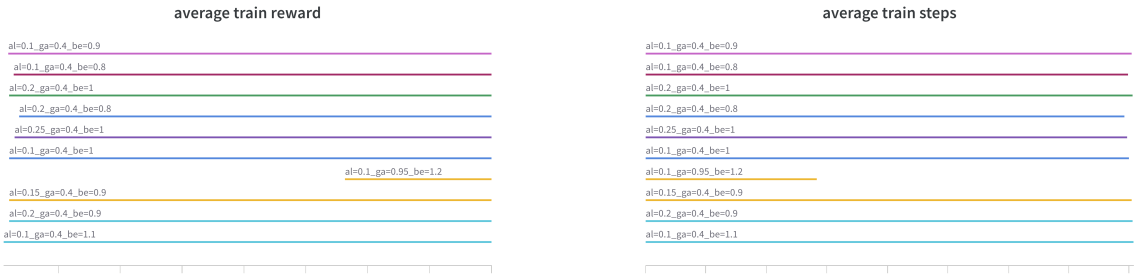
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

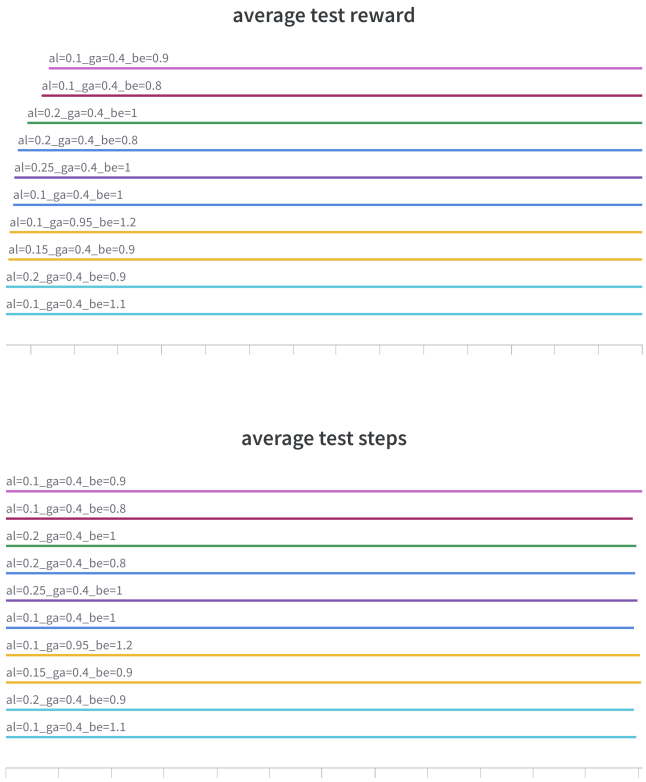
Parallel Co-ordinates Plot



Recorded Metrics

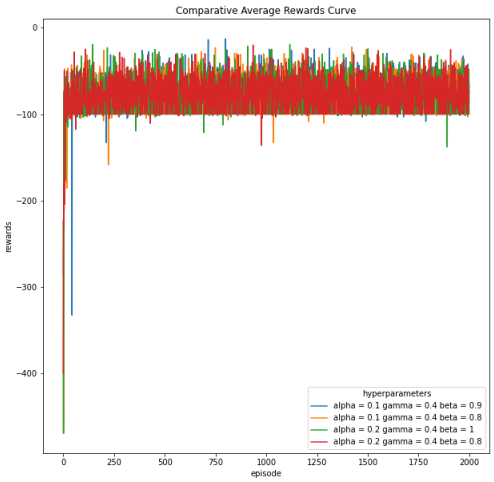


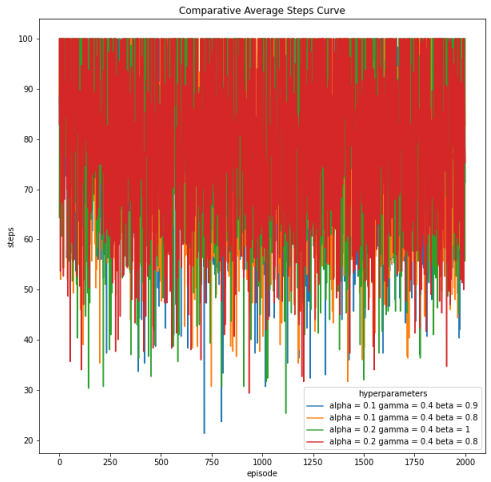
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



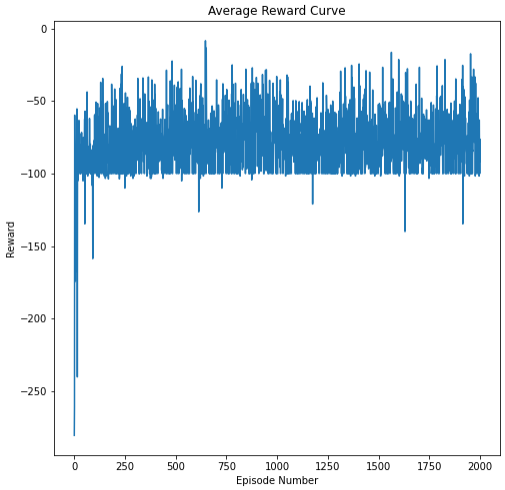


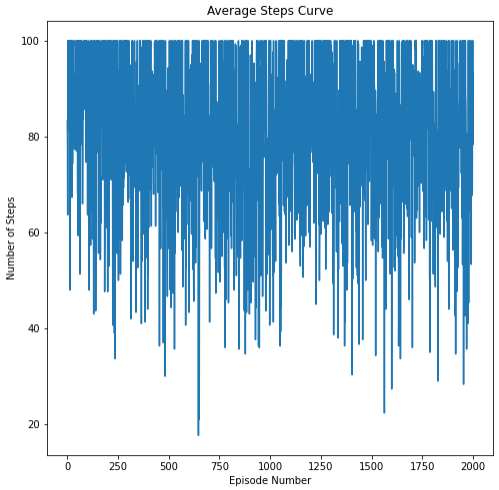
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.1, 0.4, 0.9)$ are the best hyper-parameters according to our sweep.

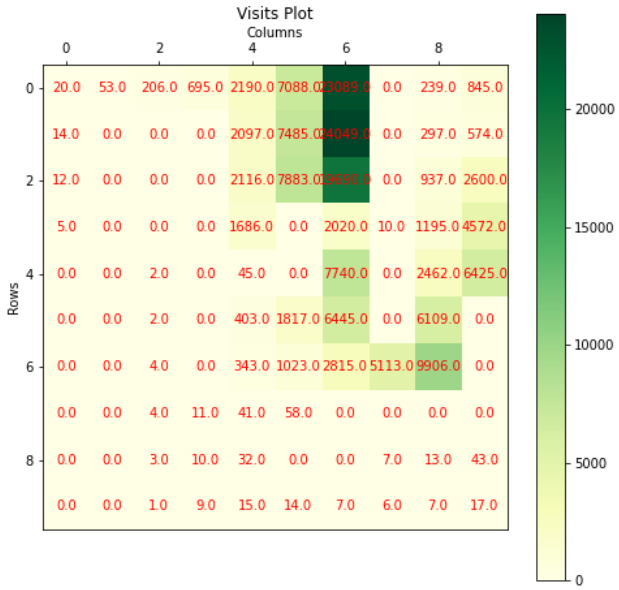
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

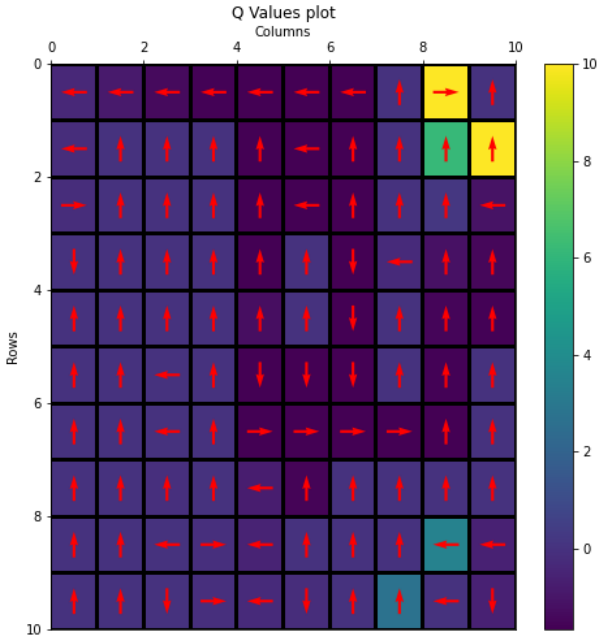




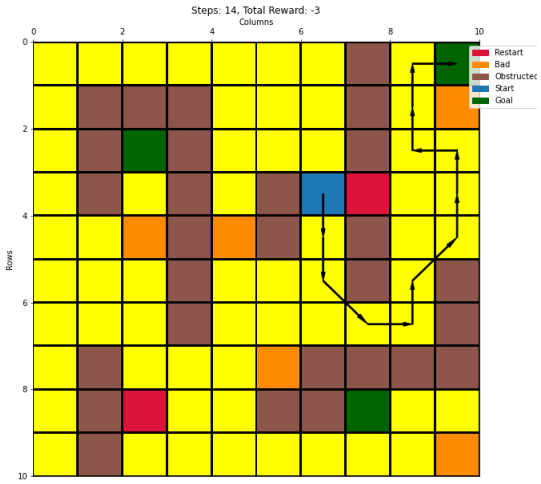
Heatmap with state visit counts

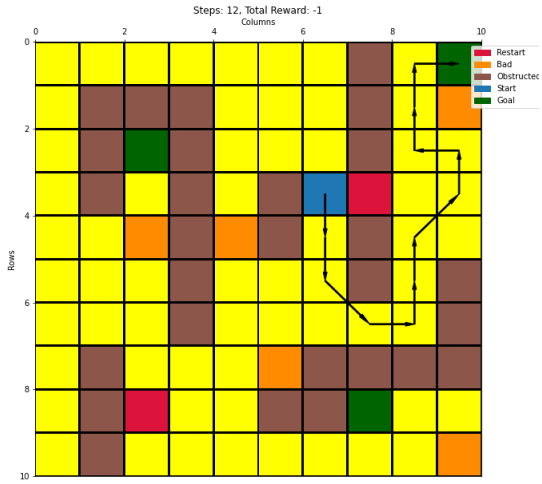


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present. Even though the goal states at $(0, 9)$ and $(8, 7)$ are equidistant from the start state, the agent biases towards $(0, 9)$ because of the wind.
- Even after hyperparameter finetuning, the agent exceeds the step count per episode most of the time. This is an outlier that escaped our tuning methods.

Configuration 28

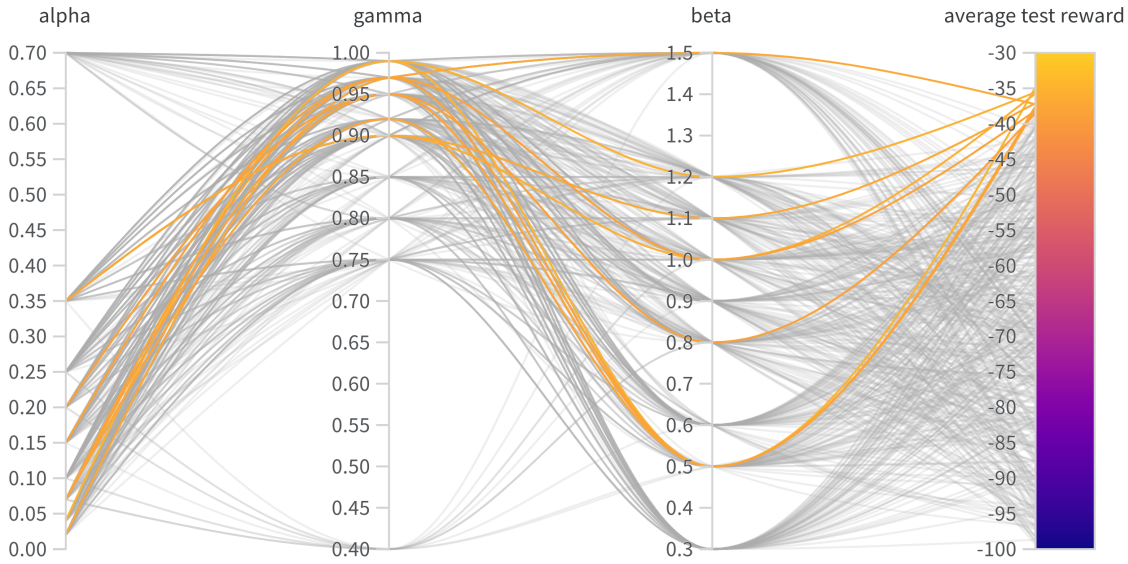
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** True
- **start -** $(3, 6)$
- **p value -** 0.7

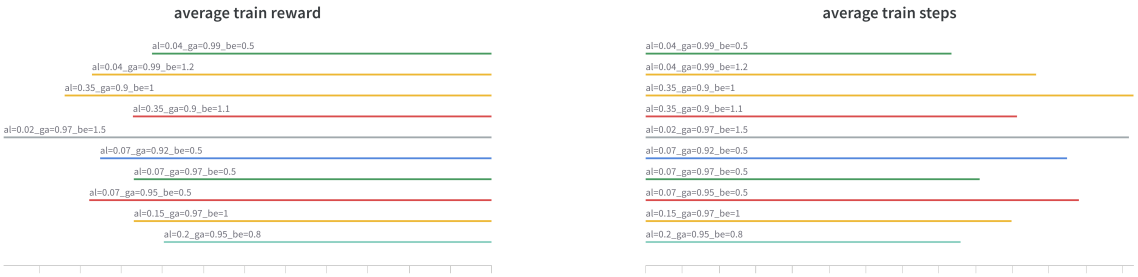
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

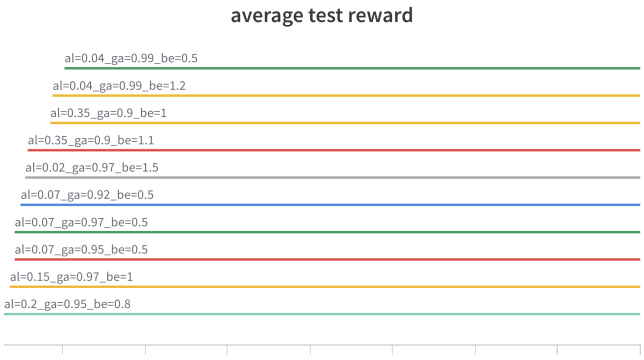
Parallel Co-ordinates Plot

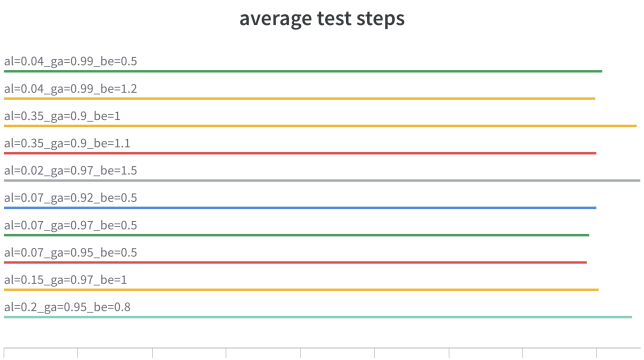


Recorded Metrics



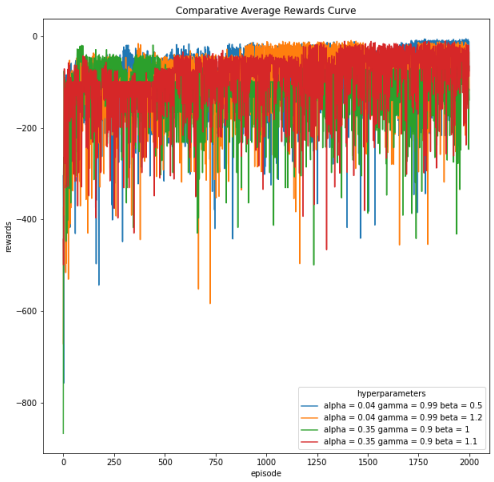
Train Metrics

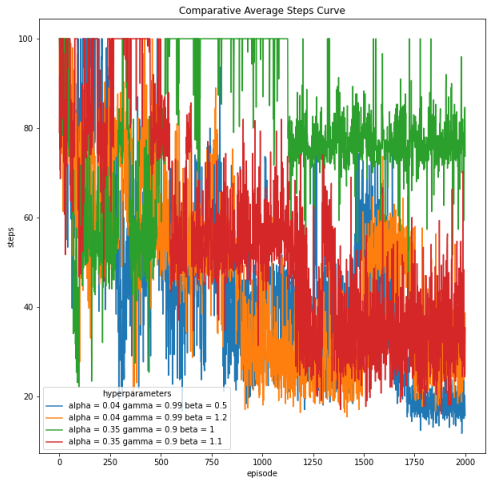




Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



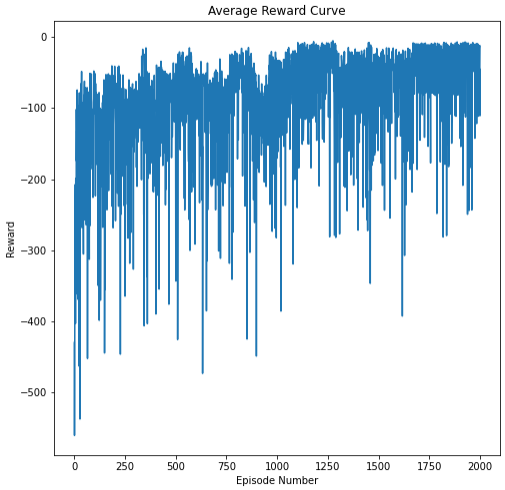


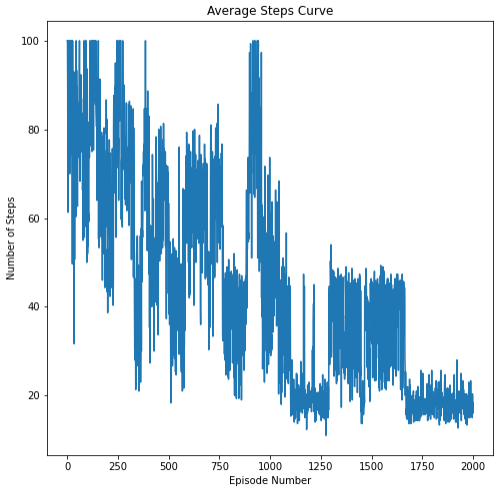
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.04, 0.99, 0.5)$ are the best hyper-parameters according to our sweep.

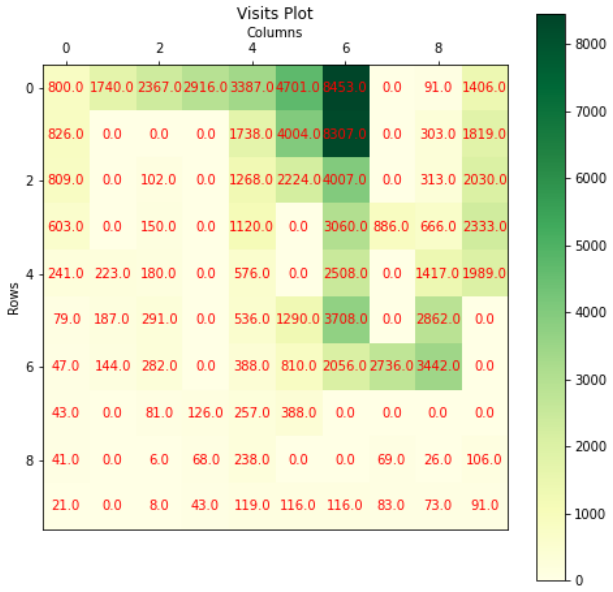
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

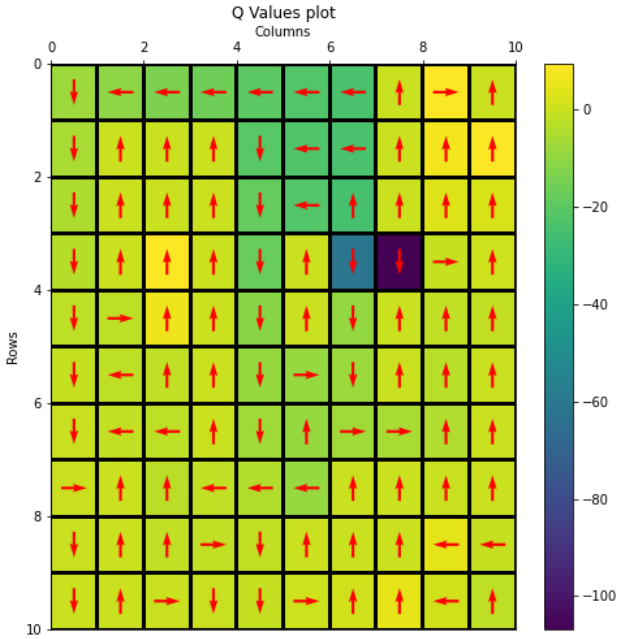




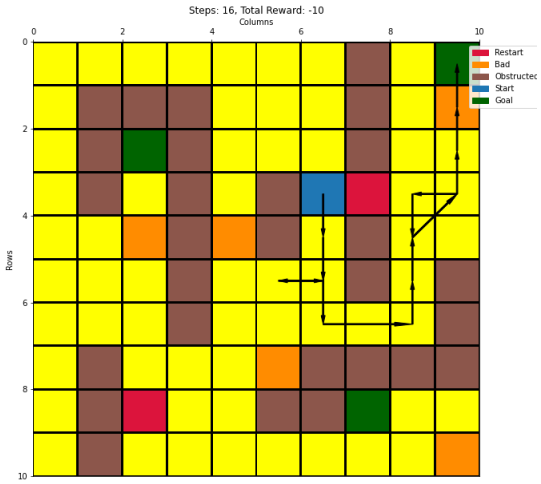
Heatmap with state visit counts

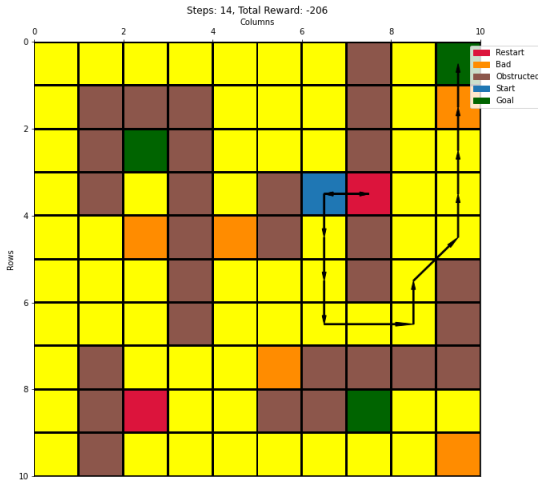


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, we have rightward wind present along with the possibility of action failure. This is the reason for the large fluctations in the reward and steps curves once again.
- Because of the wind, it is also biased towards $(0, 9)$ as well. It also goes into the restart state next to the start state because of the wind and takes longer routes due to action failure.

Configuration 29

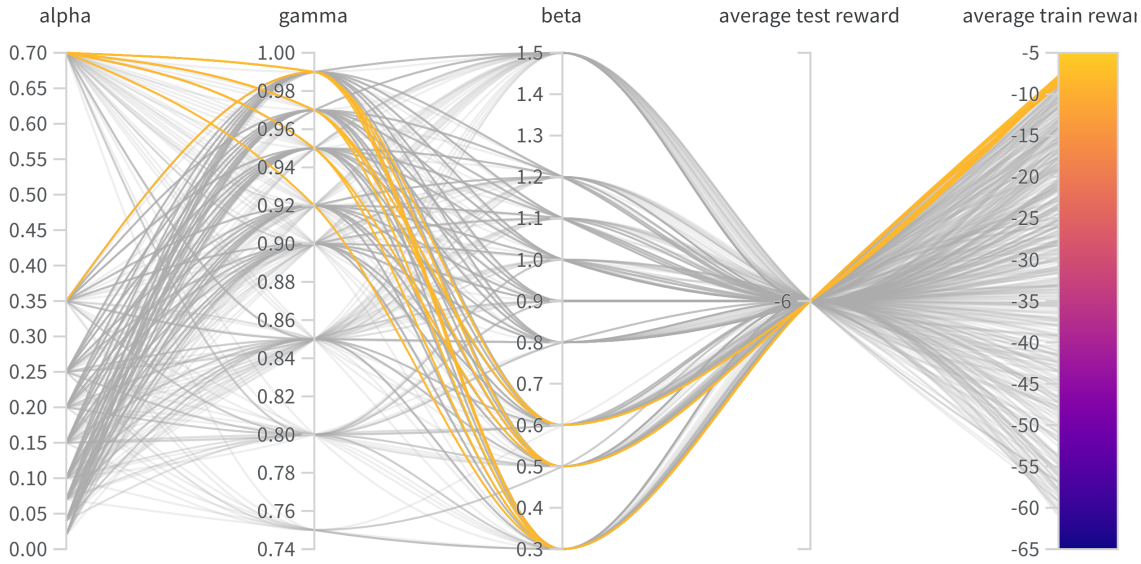
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 1.0

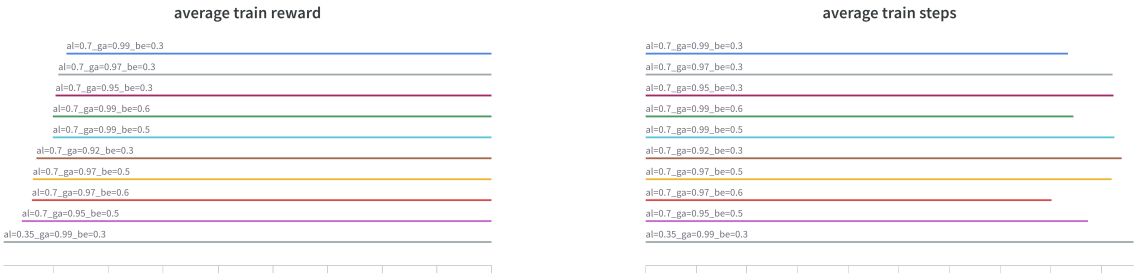
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



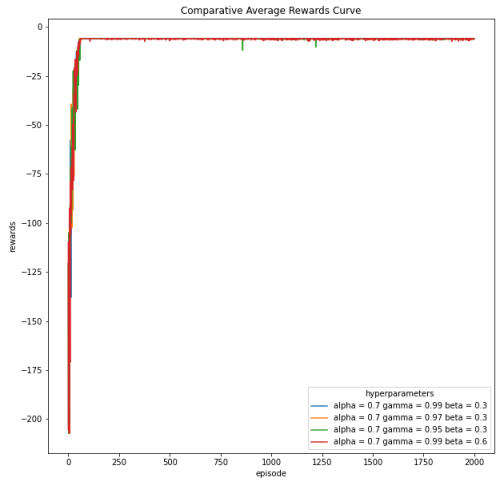
Recorded Metrics

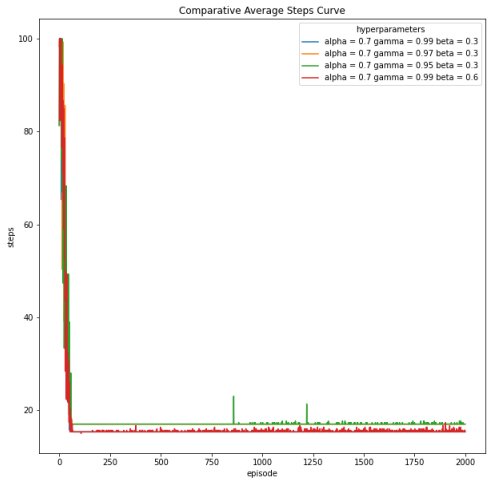


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



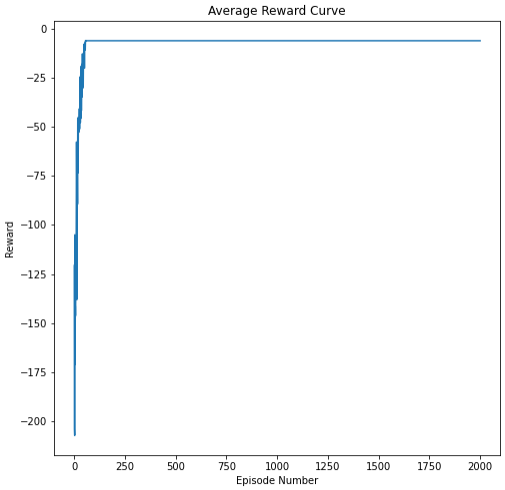


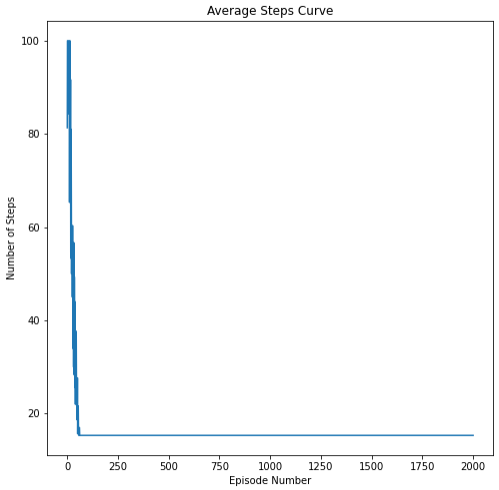
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.7, 0.99, 0.3)$ are the best hyper-parameters according to our sweep.

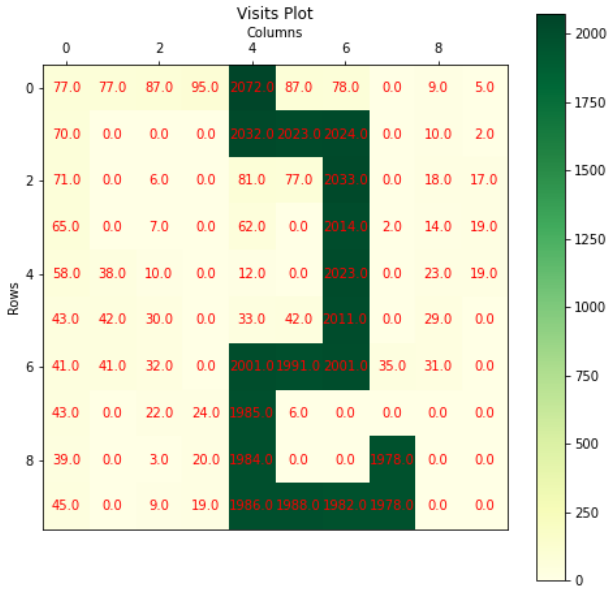
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

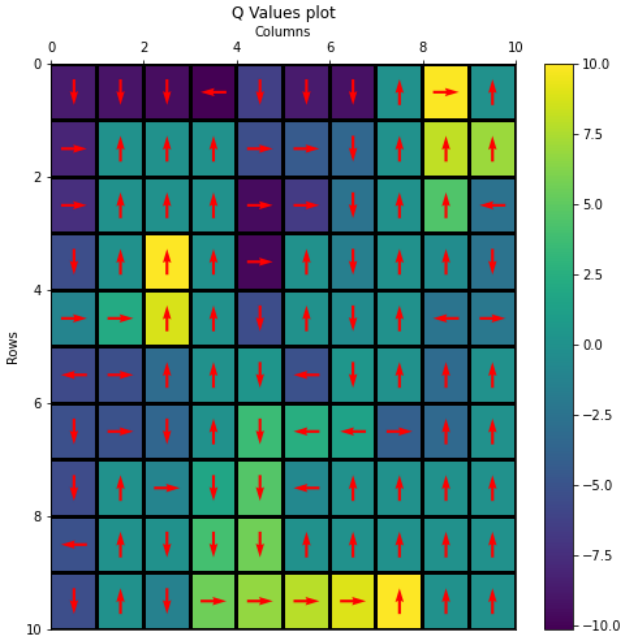




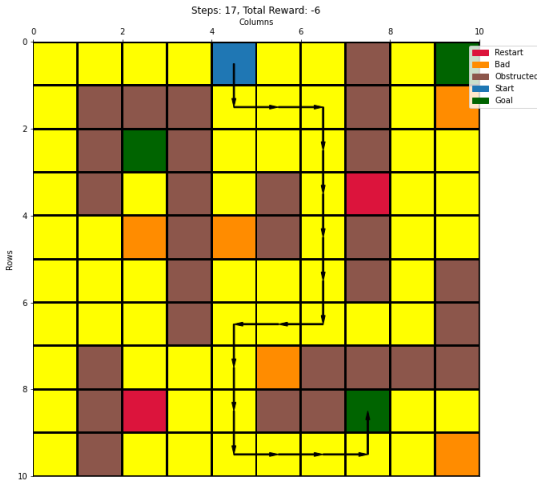
Heatmap with state visit counts

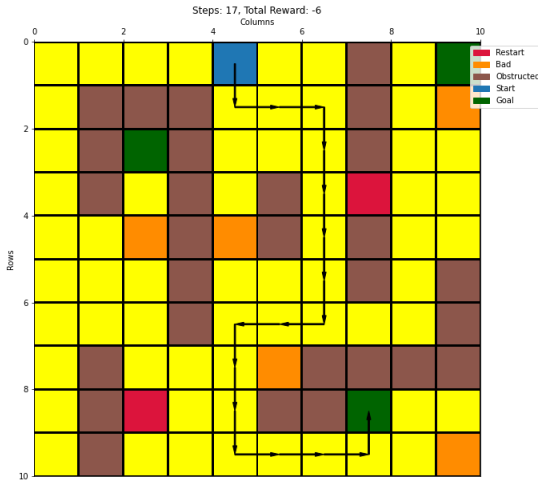


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path towards the goal (8,7).

Configuration 30

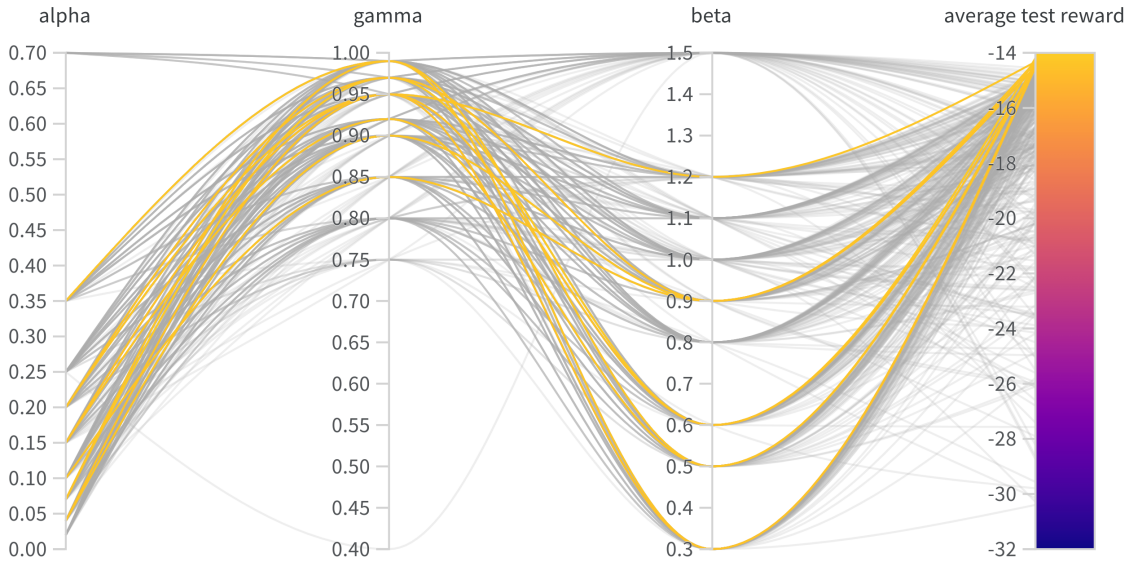
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** False
- **start -** $(0, 4)$
- **p value -** 0.7

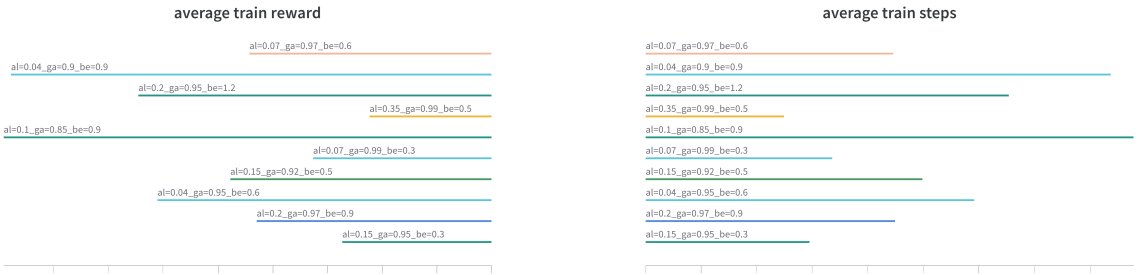
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



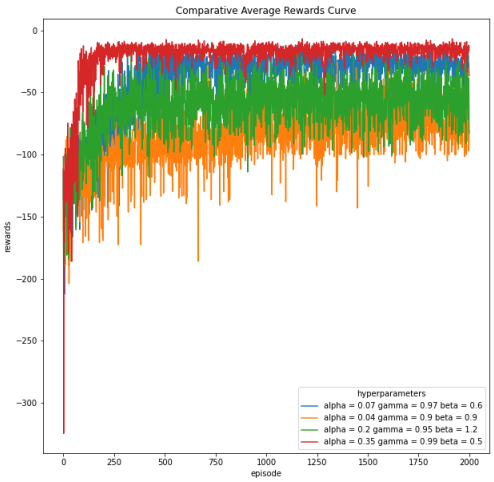
Recorded Metrics

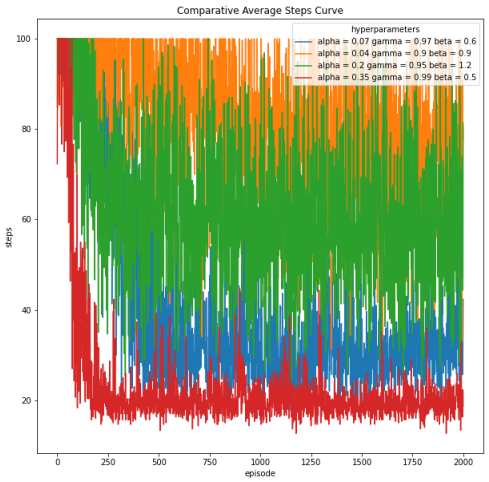


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



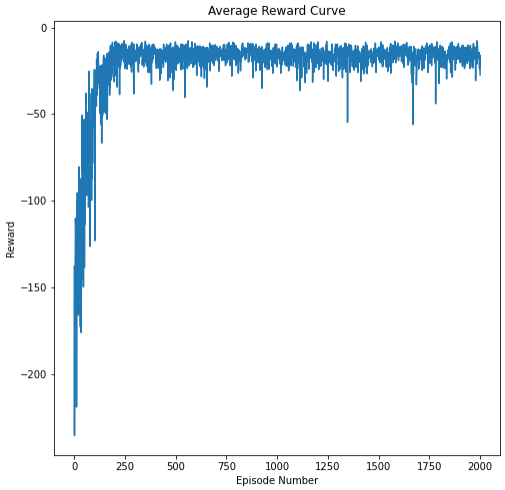


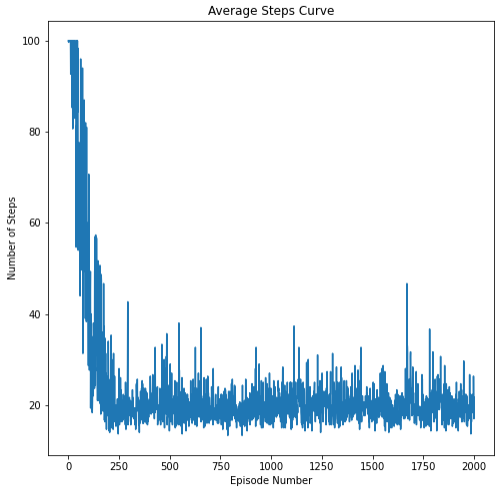
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.35, 0.99, 0.5)$ are the best hyper-parameters according to our sweep.

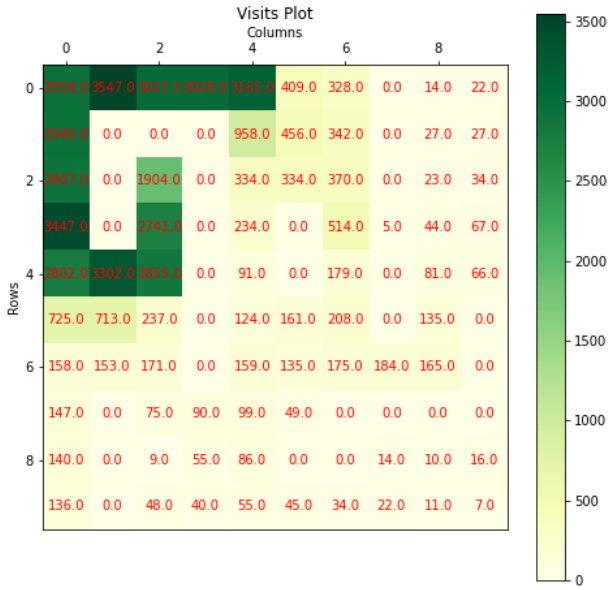
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

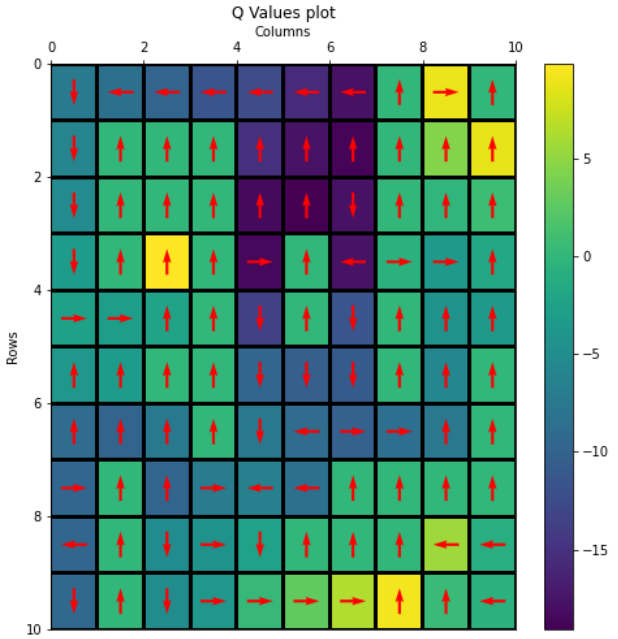




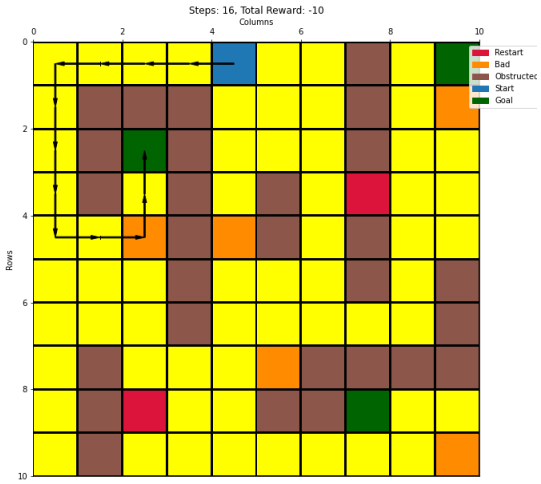
Heatmap with state visit counts

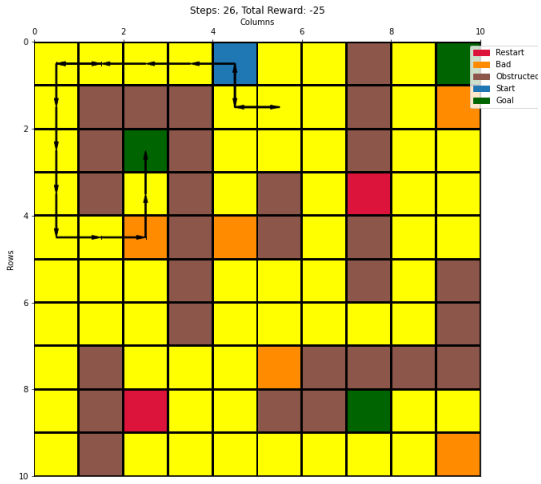


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but there is action failure.
- The agent always tries to take the path to $(2, 2)$ directly. The paths taken in the renderings show a bias towards $(2, 2)$ because the shown path is highly constrained where movement along the first row and first column is not affected by action failure.

Configuration 31

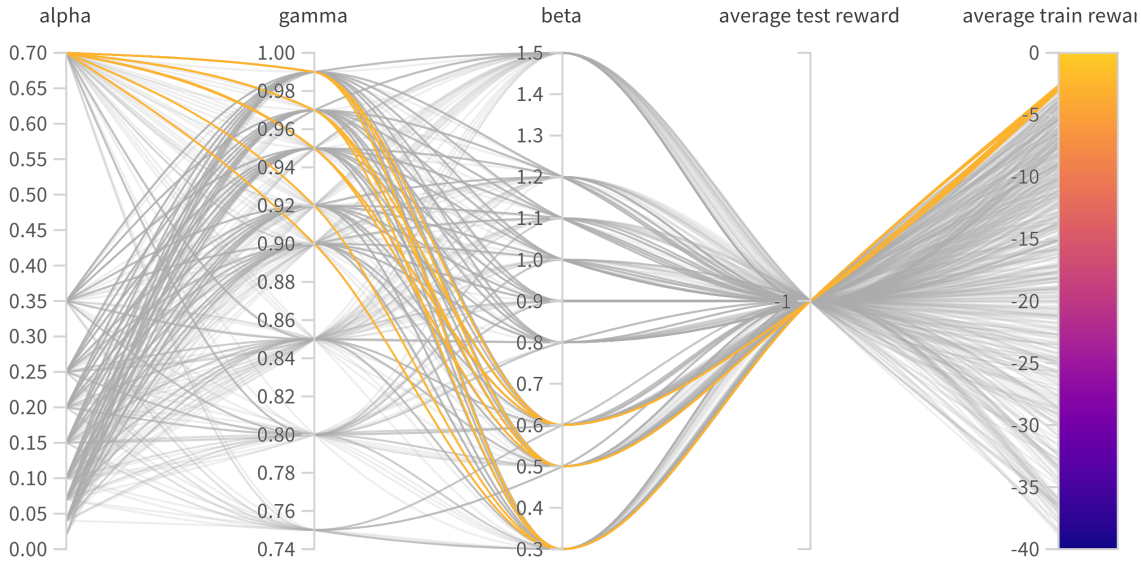
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 1.0

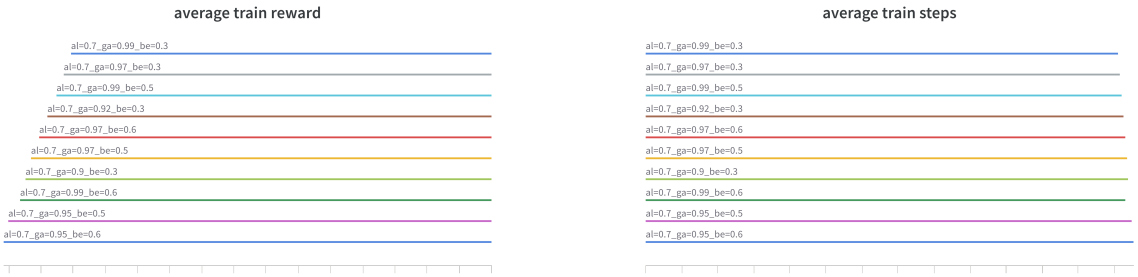
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

Parallel Co-ordinates Plot



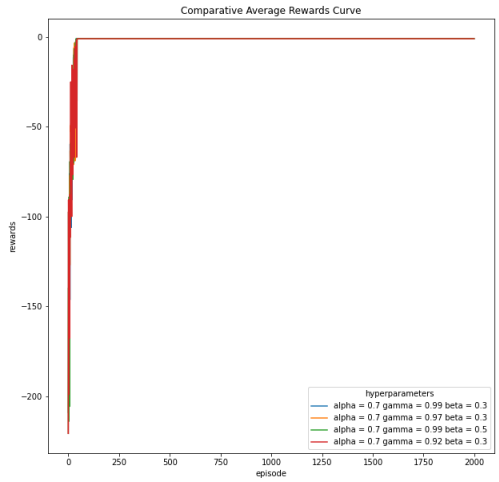
Recorded Metrics

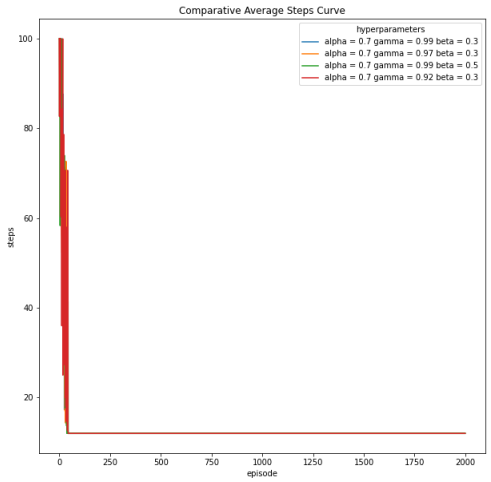


Train Metrics

Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



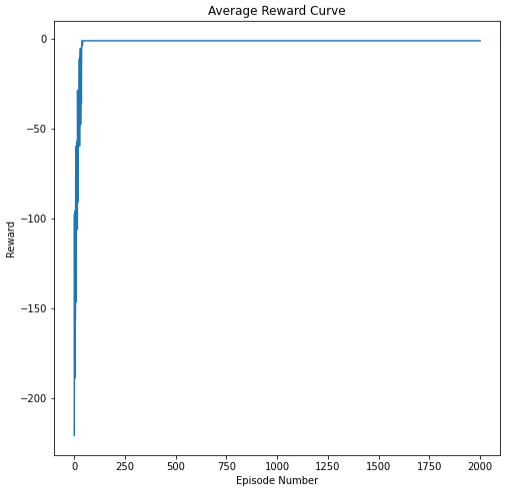


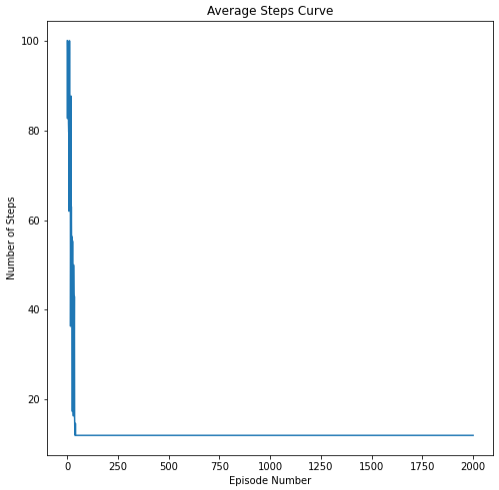
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.7, 0.99, 0.3)$ are the best hyper-parameters according to our sweep.

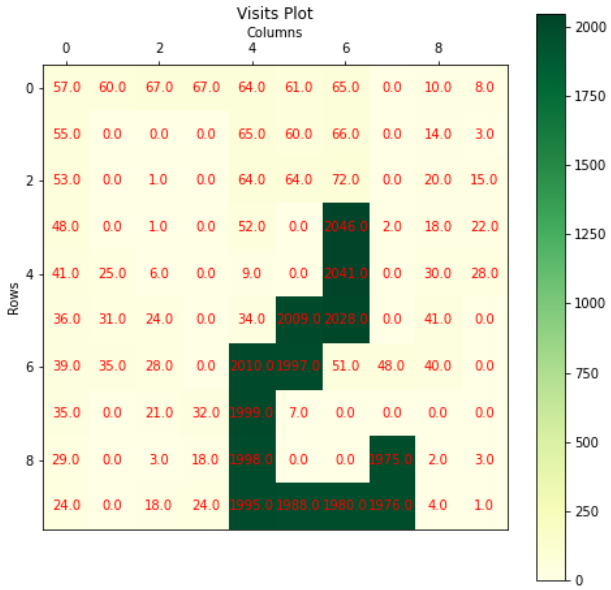
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

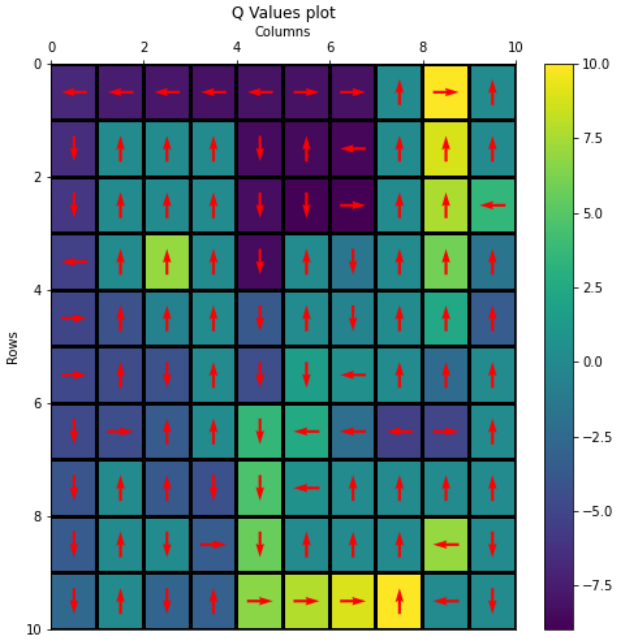




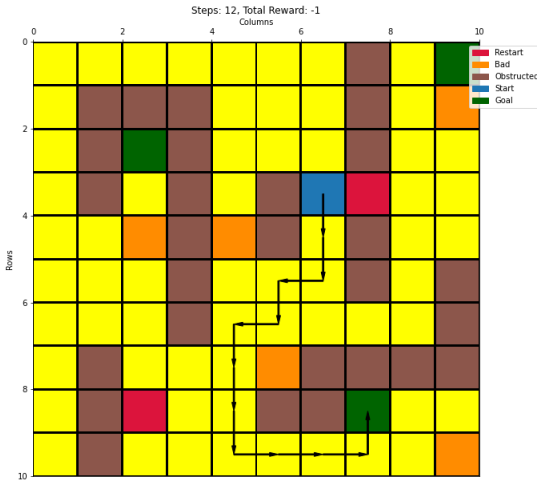
Heatmap with state visit counts

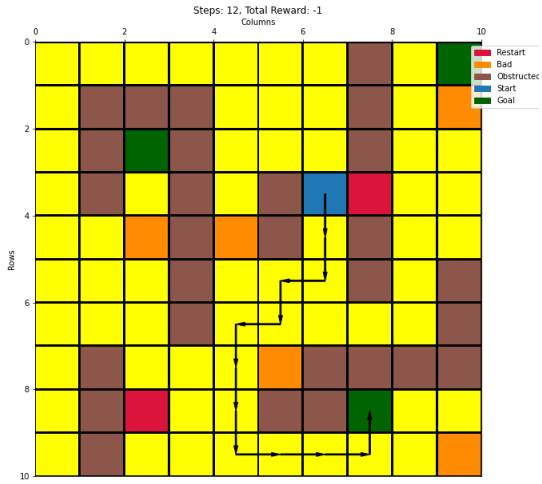


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind and no action failure. Therefore, there is no stochasticity in the environment.
- The agent always takes the shown path to $(8, 7)$.

Configuration 32

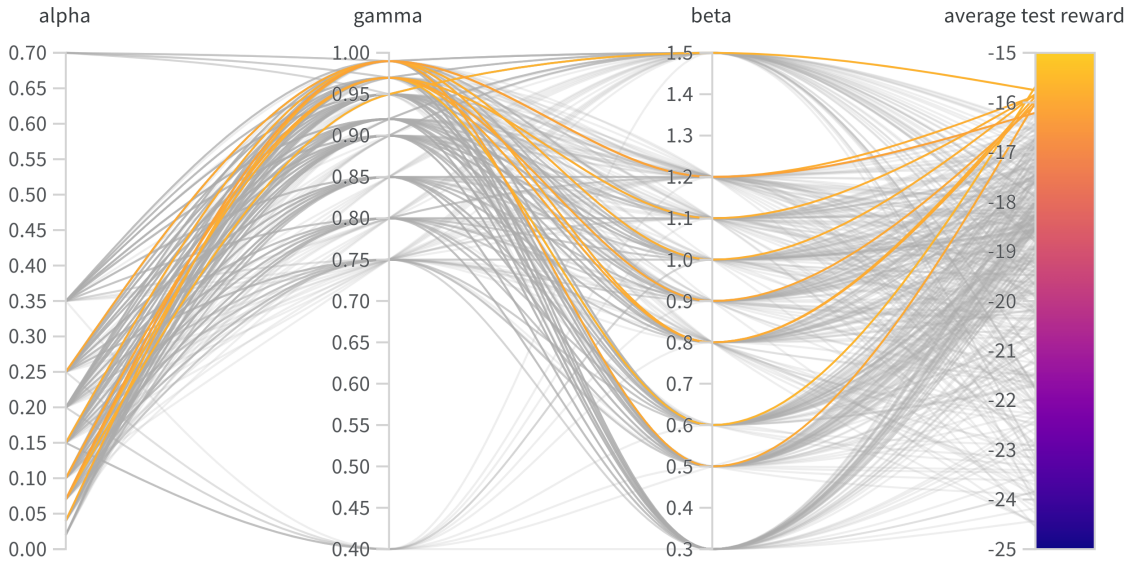
Configuration Description

- **learning -** Q-Learning
- **action -** Softmax Action
- **wind -** False
- **start -** $(3, 6)$
- **p value -** 0.7

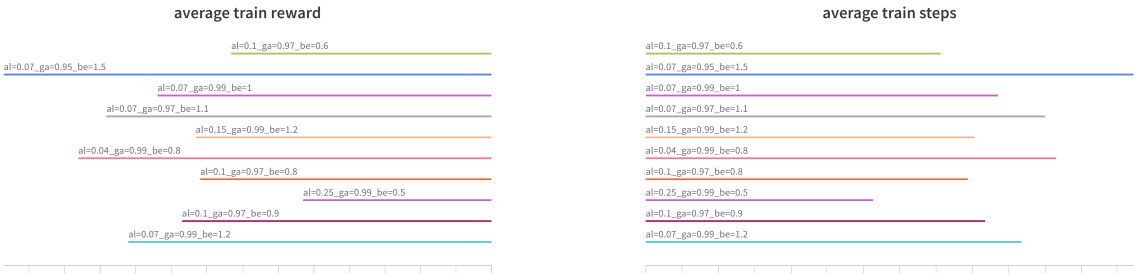
Wandb Results

We show the parallel co-ordinates plot with best 10 combinations colored along with recorded metrics during our sweep.

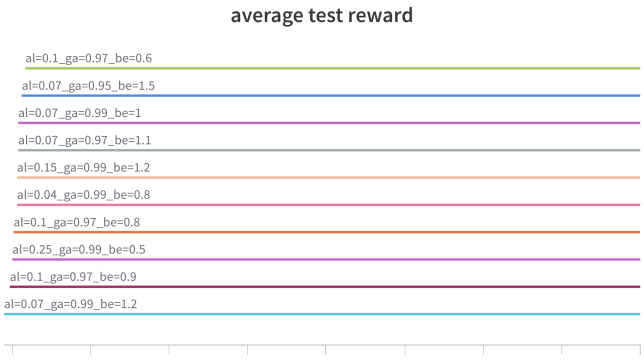
Parallel Co-ordinates Plot



Recorded Metrics

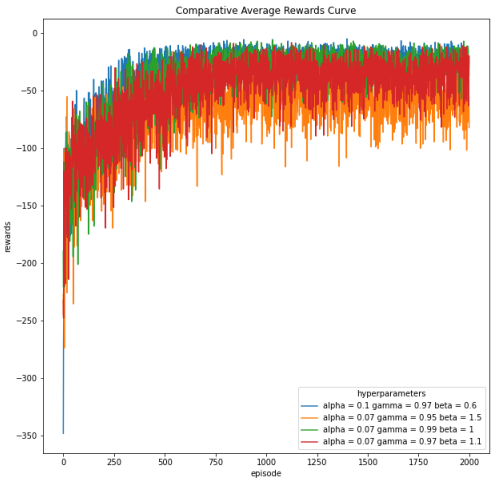


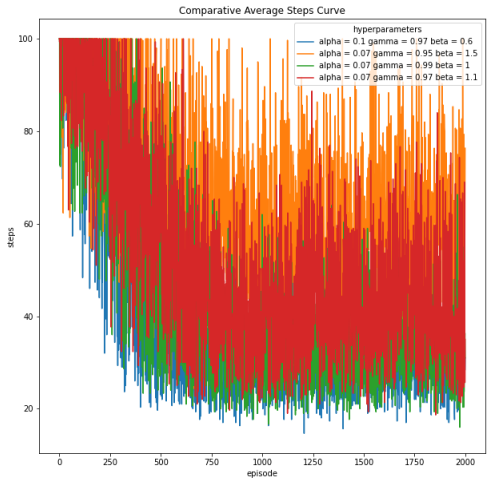
Train Metrics



Test Metrics

Comparative Average Reward Curve and Comparative Average Steps Curve



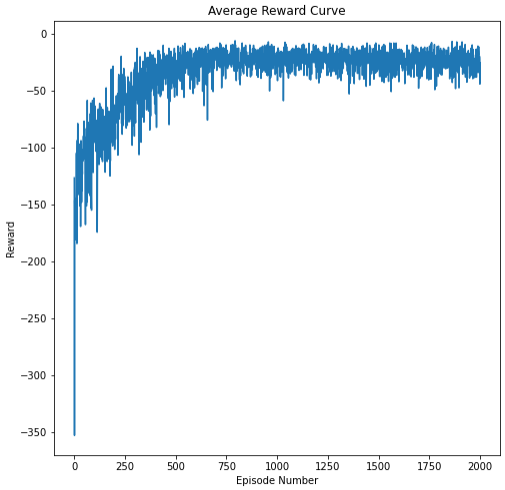


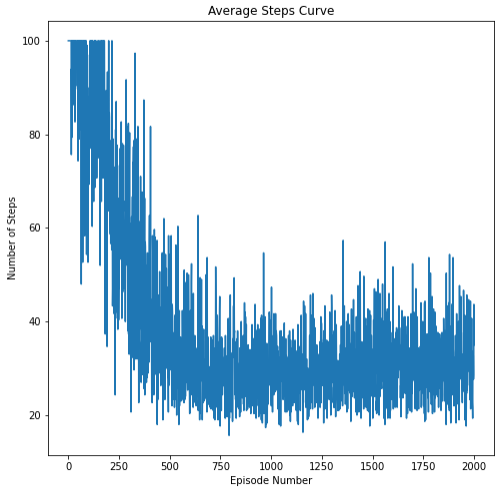
Best hyper-parameter Combination

We can see that $(\alpha, \gamma, \beta) = (0.1, 0.97, 0.6)$ are the best hyper-parameters according to our sweep.

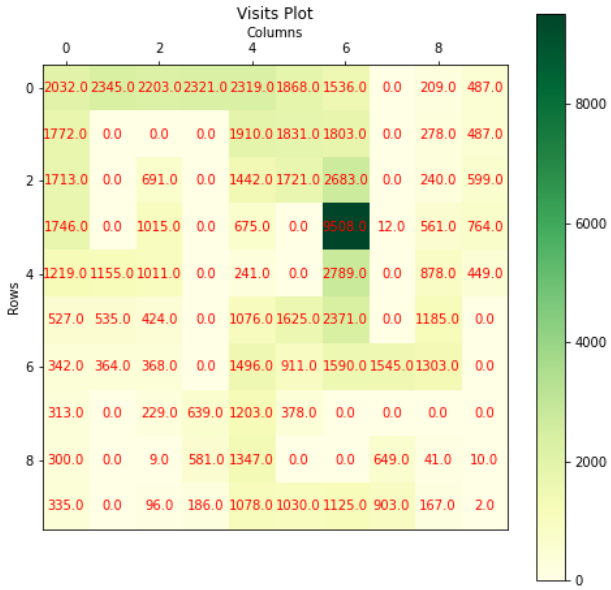
Plots for the best hyper-parameters

Average Reward Curve and Average Steps Curve

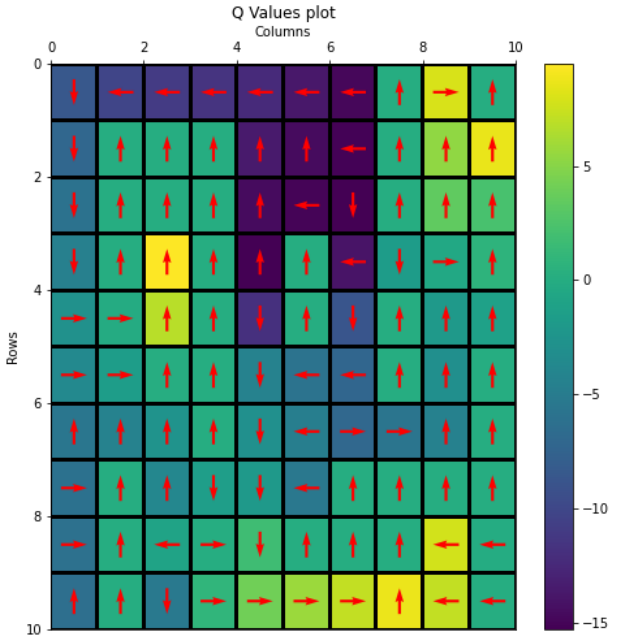




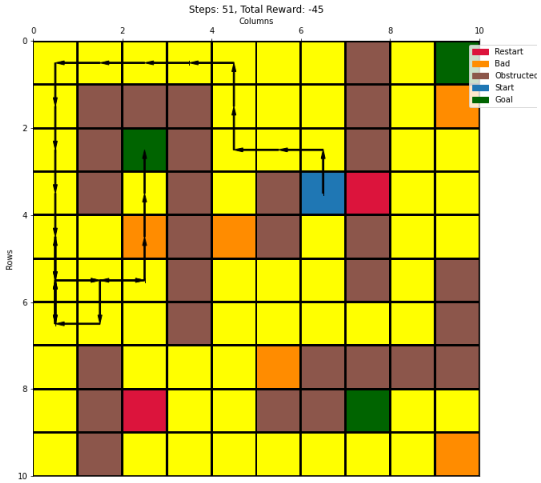
Heatmap with state visit counts

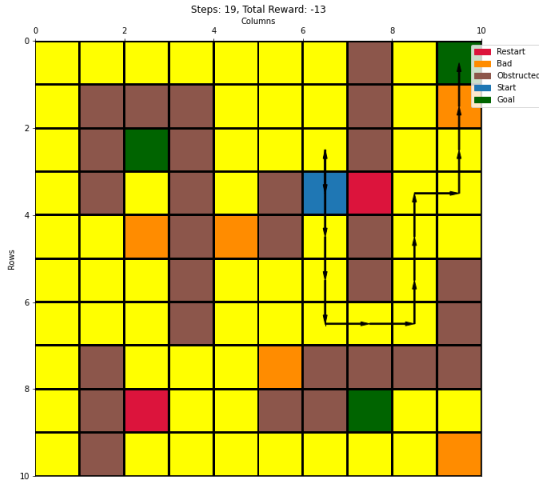


Heatmap of the grid with arrows showing the optimal action



Rendering 2 episodes of final policy





Justifications and Inferences

- Here, there is no wind but action failure may happen. From the optimal policy map, we can see that two consecutive action values at the begining of the episode has caused the agent to go to $(2, 2)$ goal state.
- One important observation is that the optimal action at start state is left. If the optimal action at start had been up or down, then the agent may move into the restart state if action failure happens. Whereas failure of left at start only results in safe up or down moves. We can see this phenomenon in configurations 8, 16 and 24 as well.

Conclusion

Risk averse nature of SARSA in comparison to Q-Learning

- SARSA is a ON-policy algorithm where the behaviour policy is the same as the target policy. If the behaviour (next action selection) policy is exploratory enough, then SARSA will actually be risk-averse because it makes updates to Q values based on its next action.
- Q-Learning uses a behaviour policy to make updates which is different from the target policy that it learns (which is the optimal policy). The update is made with the return-estimate as the maximum among all choices in the next state. Hence, it is OFF-policy. It is not risk-averse as the next action choice (given by its behaviour policy) is not considered while making the update.
- If we care about the rewards that are obtained while learning (when we are in real-world than an environment simulator), then being risk-averse while learning is necessary.
- In configurations 5 (SARSA) and 21(Q-Learning), we can see that the respective best combinations have $\epsilon = 0.01$. There are two paths to the 2 goals at $(2, 2)$ and $(8, 7)$ with total

reward as -6. Remember that we have no wind here and no action failure as well. We can see that SARSA in config. 5 has explored paths to both the goals from the heatmap. But its risk averse nature has caused it to prefer the goal $(2, 2)$ as there is a dangerous restart state next to the optimal path to $(8, 7)$.

- SARSA has its behaviour policy as its target policy. So, the behaviour policy being ep-greedy, there is a chance that it might fall into the restart state on its way to $(8, 7)$. So, it avoids this path.
- On the other hand, Q-Learning makes its updates according to the optimal policy and doesn't bother about its behaviour policy. This is clear in the fact that it has chosen the path to $(8, 7)$. The initial selection of $(8, 7)$ is by chance but it has also visited $(2, 2)$ and did not revise its behaviour.

Epsilon greedy and Softmax comparison : Fluctuations even after convergence

- The nature of ϵ greedy policy is that it chooses random actions uniformly,even if certain actions are better than others. Whereas softmax action selection policy chooses random actions with a probability proportional to the current values.
- On comparing rewards curve of configurations 21(ϵ-greedy) and 29(softmax), it is clearly visible that, even after convergence of reward, due to ϵ-greedy policy's nature, there are still fluctuations in the reward output.This is not the case with softmax, because after convergence softmax is always going to choose best actions whose probability will be anyways high.
- In general, following the principle of GLIE (Greedy in the limit, infinite in exploration) is necessary for convergence to optimal policy. So, doing some form of parametrized decay of ϵ is a good idea.

Choice of hyperparameters

- In our code, we have used fixed lists of 9 values for $\alpha, \gamma, \epsilon, \beta$ and performed a wandb sweep on the resulting 729 combinations for each configuration (only ϵ or β is used for a configuration).
- We came up with these lists based on intuition and some experimentation as well.
- In general, we need to have a small learning rate ($\alpha < 0.2$) in order to ensure that any one reward sample does not have an outsized impact on Q-estimates. This is evident in our *alphalist* in the code, where we have added a few large values to demonstrate why a small learning rate is needed in our sweeps.
- Again in general, a good value of discount factor ($\gamma > 0.9$) is necessary so that the agent gives more importance to the return for a selected action in some state. This is necessary to make the agent give huge importance to the future return and not be short-sighted. This fact and some experiments have resulted in our *gammalist* where some small values are also present to highlight the general need for large γ.
- Also in general, we need to have a small ϵ ($\epsilon < 0.1$) in order to ensure that eventually the agent behaves greedily most of the time. In fact, we could have a decaying ϵ so that the agent is more exploratory in the beginning and becomes greedy in the limit. This explains our *epsilonlist*.

- In general, the larger the beta, the more uniformly random is the softmax action. But from the expression of softmax action, it is clear that β depends on the order of magnitude of the rewards (as Q-estimates depends on the magnitude as well). So, we came up with *betalist* purely by experimentation.