Homework 1

Gridworld

The following hyperparameters were used for all algorithms unless otherwise specified:

- $\gamma = 0.95$
- $\theta = 10^{-16}$
- $\alpha = 0.1$
- $\epsilon = 0.1$

Policy Iteration

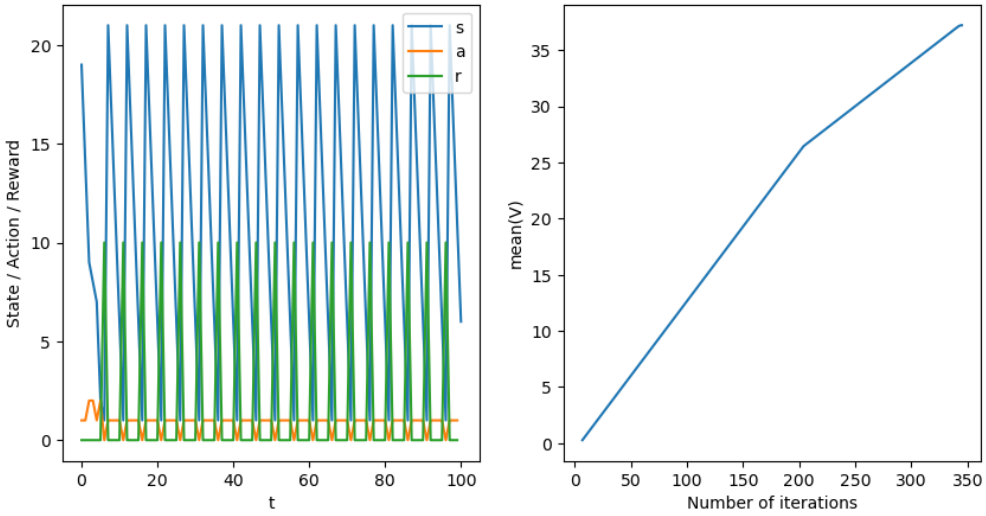

Figure 1: Gridworld Trajectory and Learning Curve for Policy Iteration

For the gridworld policy iteration, an example trajectory and learning curve are shown in Figure 1. The state value function and policy are shown in Figure 2. The policy iteration algorithm converges to the optimal policy in approximately 350 iterations.

Value Iteration

For the gridworld value iteration, an example trajectory and learning curve are shown in Figure 3. The state value function and policy are shown in Figure 4. The value iteration algorithm converges to the optimal policy in approximately 140 iterations.

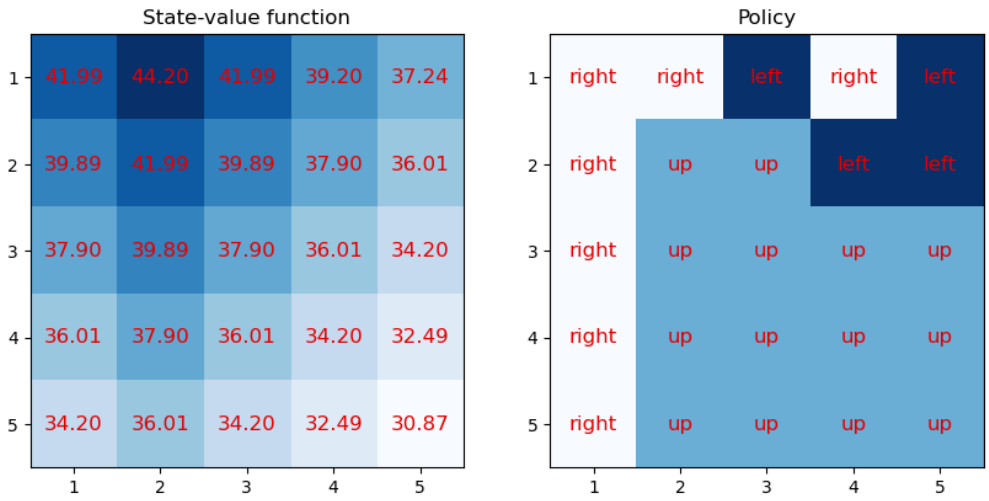

Figure 2: Gridworld State Value Function and Policy for Policy Iteration

SARSA

For the gridworld SARSA algorithm, an example trajectory and learning curve are shown in Figure 5. The state value function and policy are shown in Figure 7. The SARSA algorithm shown here was run for 5000 episodes. Additionally, TD(0) was applied to the SARSA algorithm to derive a state value function. An example trajectory and learning curve are shown in Figure 6. The state value function is nearly identical to the one generated by the value iteration and policy iteration algorithms. Additionally, the effect of varying ϵ and α on the learning curve is shown in Figure 8.

Q-Learning

For the gridworld Q-learning algorithm, an example trajectory and learning curve are shown in Figure 9. The state value function and policy are shown in Figure 11. The Q-learning algorithm shown here was run for 5000 episodes. Additionally, TD(0) was applied to the Q-learning algorithm to derive a state value function. An example trajectory and learning curve are shown in Figure 10. The state value function is nearly identical to the one generated by the value iteration and policy iteration algorithms. Additionally, the effect of varying ϵ and α on the learning curve is shown in Figure 12.

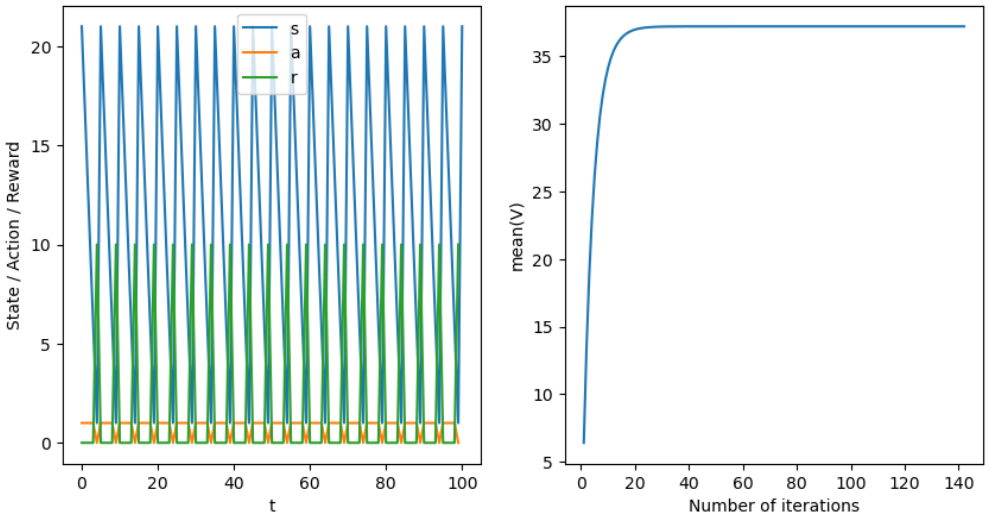

Figure 3: Gridworld Trajectory and Learning Curve for Value Iteration

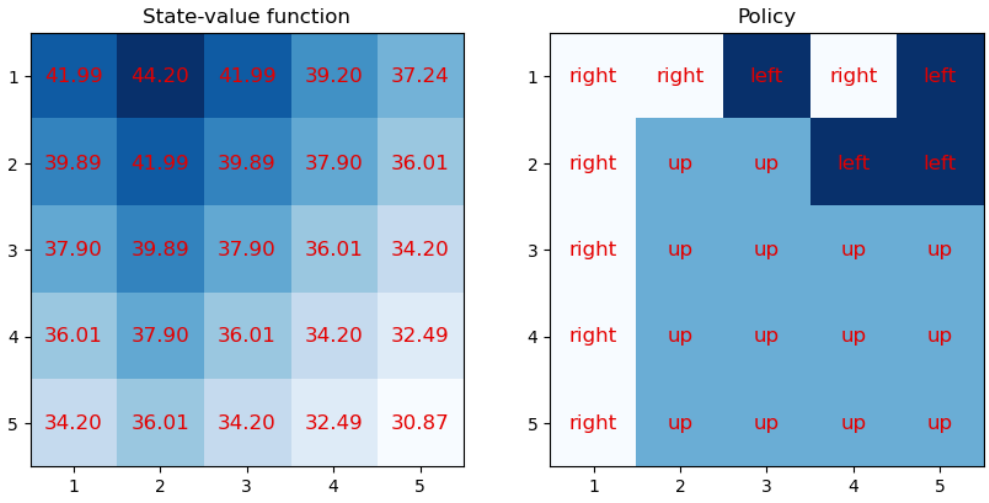

Figure 4: Gridworld State Value Function and Policy for Value Iteration

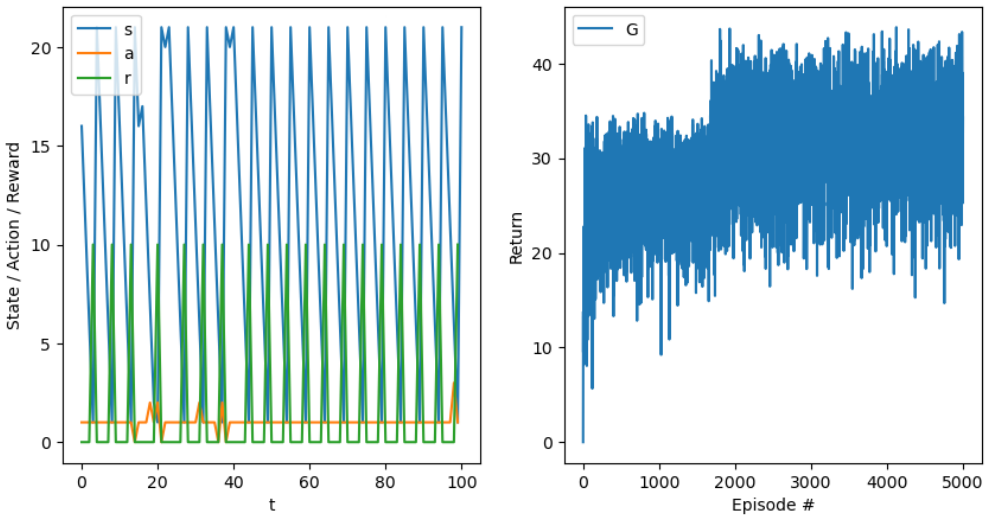

Figure 5: Gridworld Trajectory and Learning Curve for SARSA

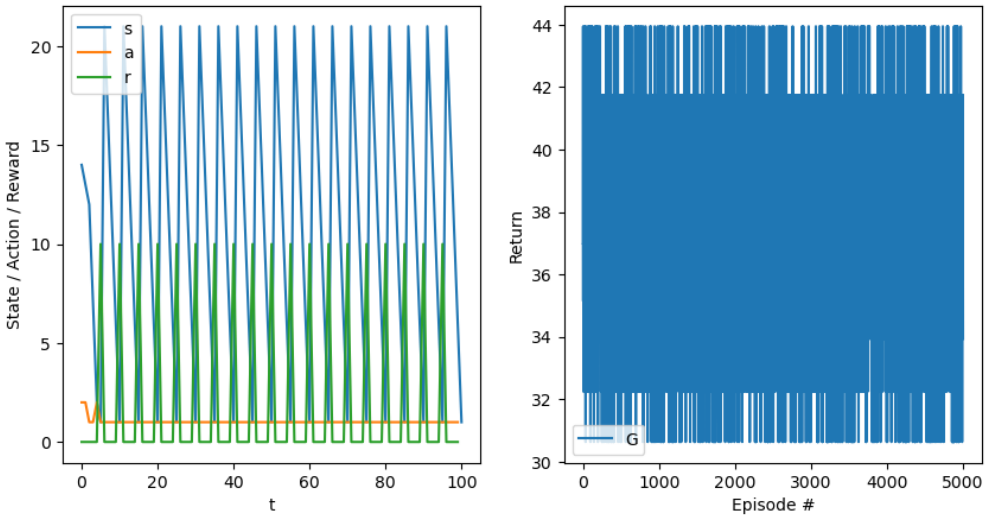

Figure 6: Gridworld Trajectory and Learning Curve for SARSA after TD(0)

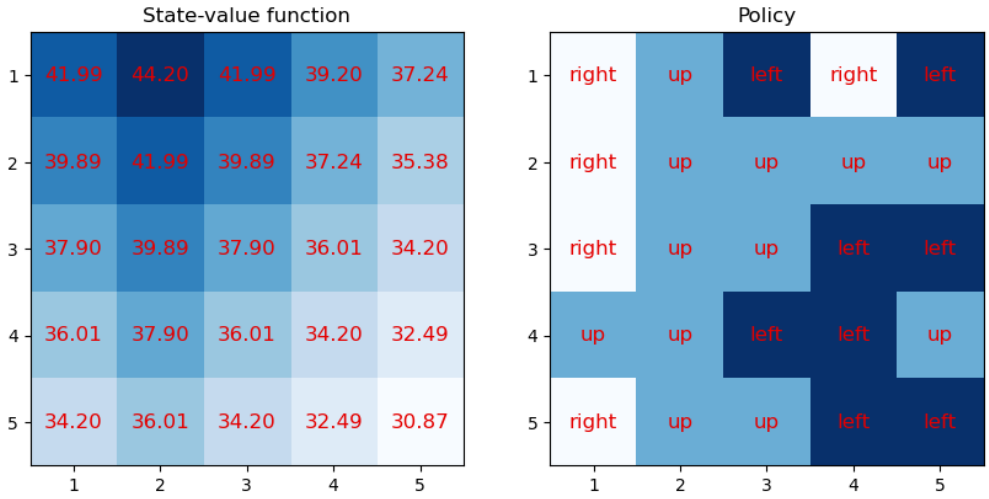

Figure 7: Gridworld State Value Function and Policy for SARSA using TD(0)

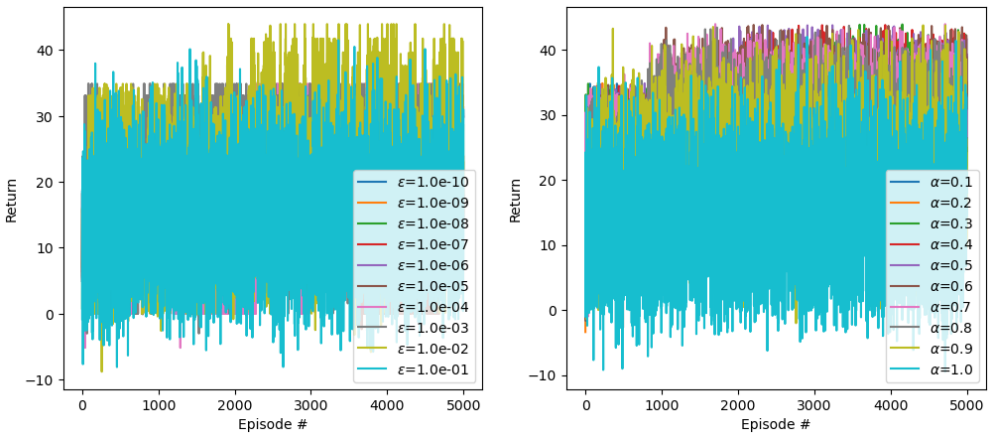

Figure 8: Gridworld Effect of ϵ and α on Learning Curve for SARSA

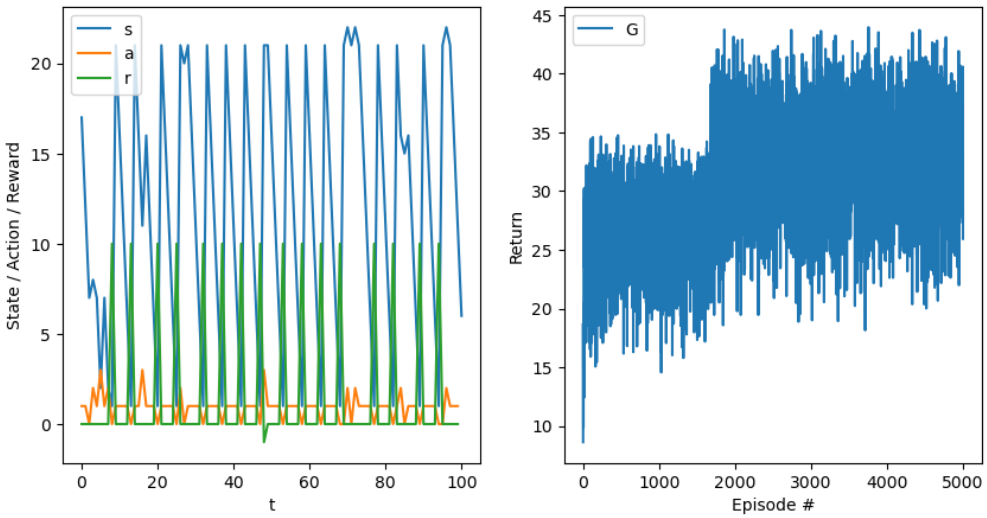

Figure 9: Gridworld Trajectory and Learning Curve for Q-Learning

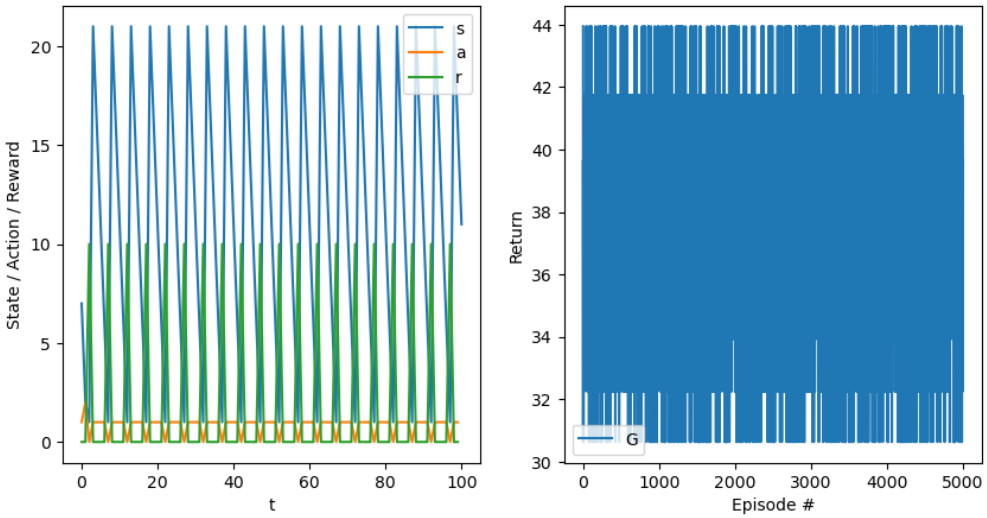

Figure 10: Gridworld Trajectory and Learning Curve for Q-Learning after TD(0)

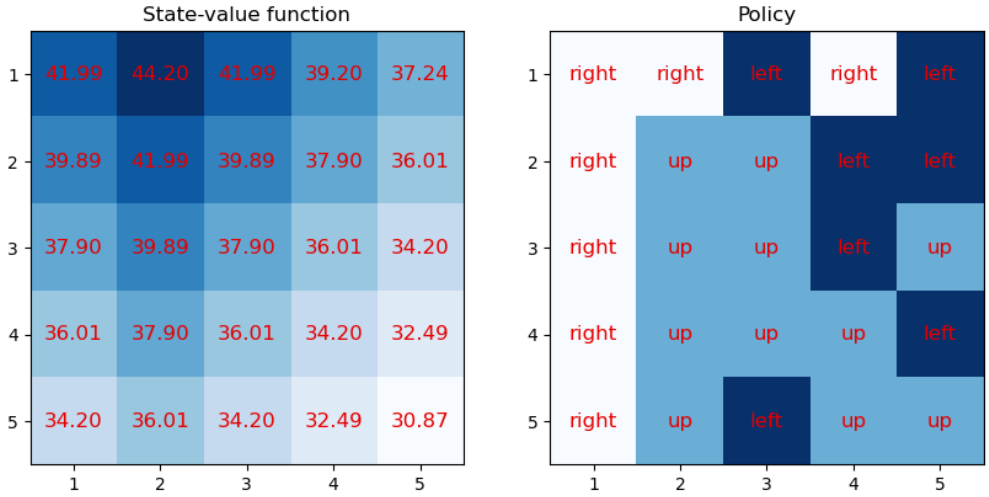

Figure 11: Gridworld State Value Function and Policy for Q-Learning using TD(0)

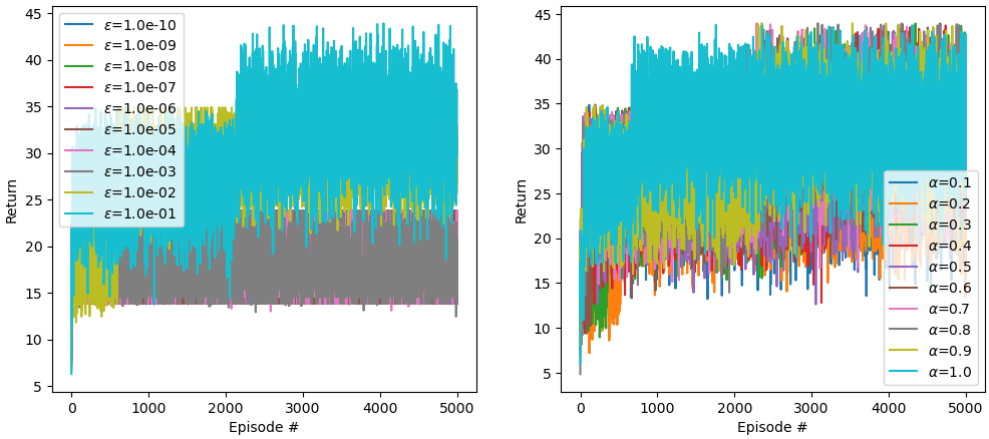

Figure 12: Gridworld Effect of ϵ and α on Learning Curve for Q-Learning

Pendulum

The following hyperparameters were used for all algorithms unless otherwise specified:

- $\gamma = 0.95$
- $\alpha = 0.3$
- $\epsilon = 0.8$
- $n_\theta = n_{\dot{\theta}} = n_\tau = 41$

SARSA

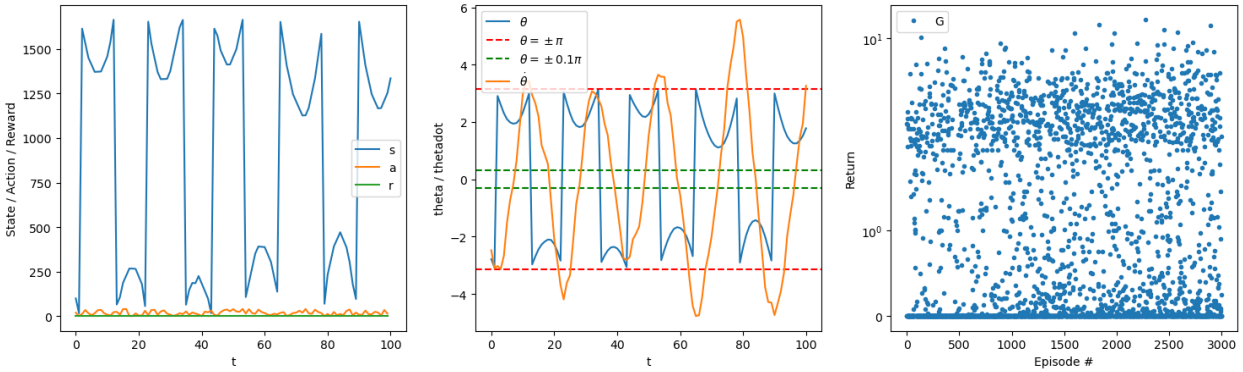

Figure 13: Pendulum Trajectory and Learning Curve for SARSA

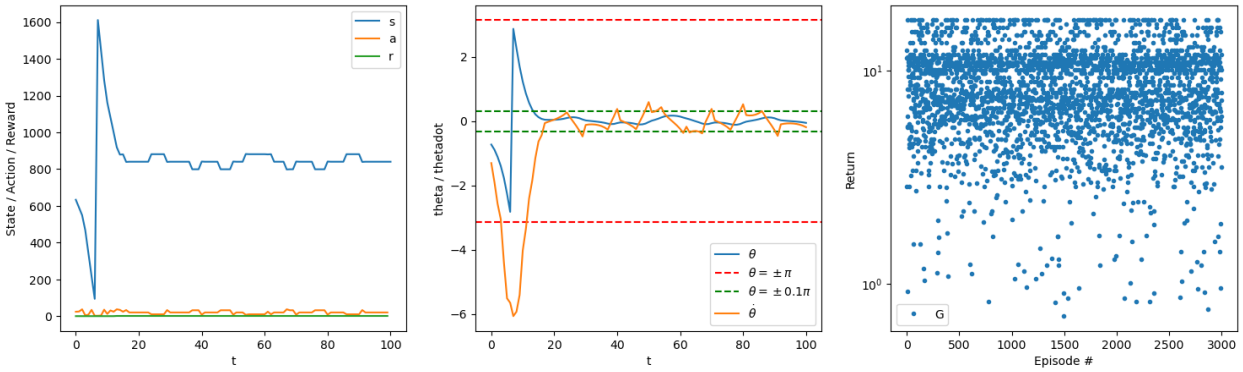

Figure 14: Pendulum Trajectory and Learning Curve for SARSA after TD(0)

For the pendulum SARSA algorithm, an example trajectory and learning curve are shown in Figure 13. The state value function and policy are shown in Figure 15. The SARSA algorithm shown here was run for 3000 episodes. Additionally, TD(0) was applied to the SARSA algorithm to derive a state value function. An example trajectory and learning curve are shown in Figure

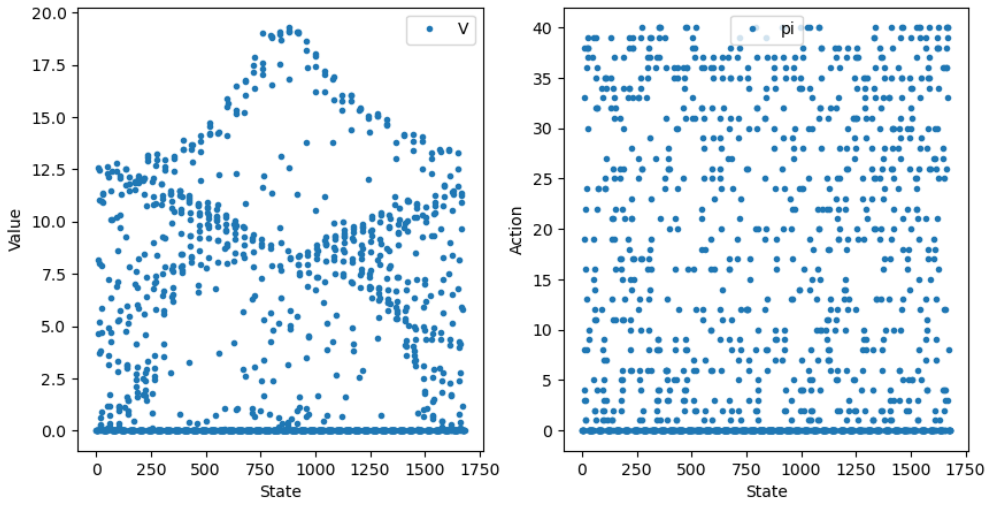

Figure 15: Pendulum State Value Function and Policy for SARSA using TD(0)

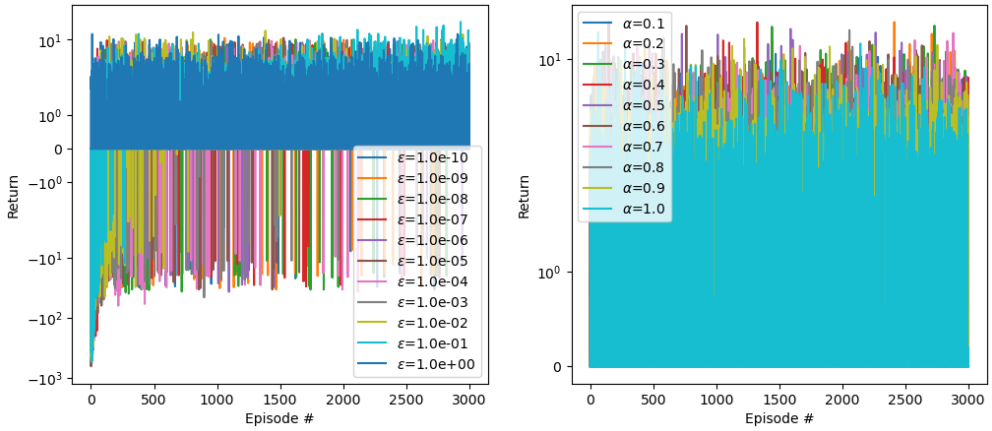

Figure 16: Pendulum Effect of ϵ and α on Learning Curve for SARSA

14. For convenience's sake and to understand the learning at different points of the process, Figure 13 only shows the trajectory at the last iteration of SARSA. Therefore, this case does not perform well. However, the real fruits of the SARSA algorithm can be seen in Figure 14, which is a trajectory that uses the policy derived from SARSA. We see here that the pendulum is controlled in 15 time units and remains controlled for the rest of the trajectory. Finally, the effect of varying ϵ and α on the learning curve is shown in Figure 16. We see that the learning curve is very sensitive to ϵ and α, especially needing high exploration values for ϵ to gain useful returns.

Q-Learning

For the pendulum Q-Learning algorithm, an example trajectory and learning curve are shown in Figure 17. The state value function and policy are shown in Figure 19. The Q-Learning

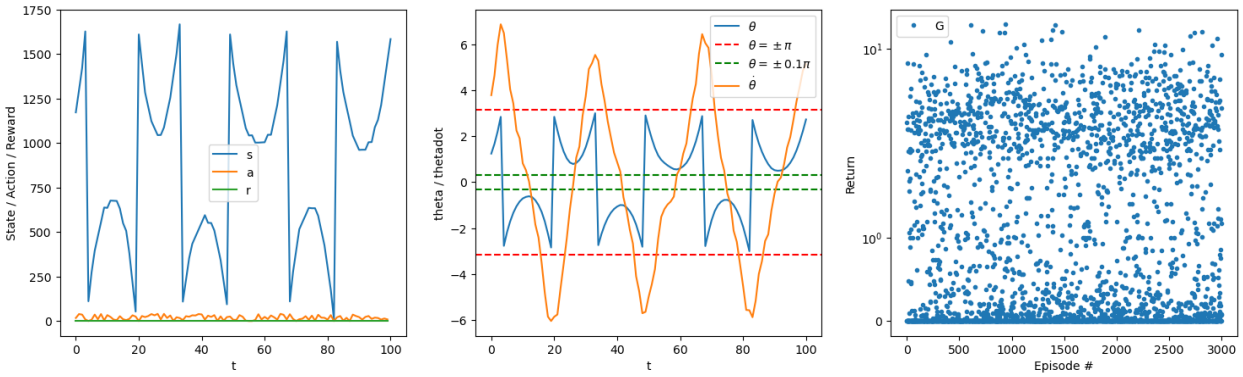

Figure 17: Pendulum Trajectory and Learning Curve for Q-Learning

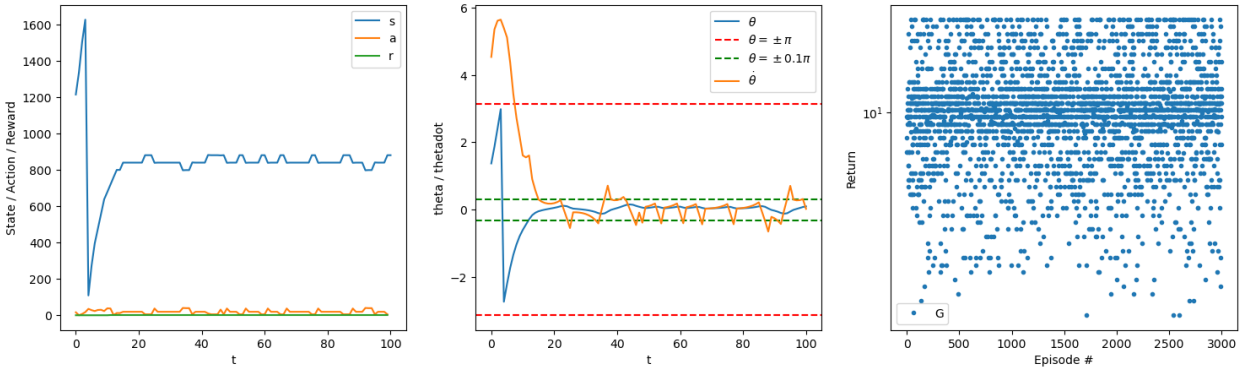

Figure 18: Pendulum Trajectory and Learning Curve for Q-Learning after TD(0)

algorithm shown here was run for 3000 episodes. Additionally, TD(0) was applied to the Q-Learning algorithm to derive a state value function. An example trajectory and learning curve are shown in Figure 18. For convenience's sake and to understand the learning at different points of the process, Figure 17 only shows the trajectory at the last iteration of Q-Learning. Therefore, this case does not perform well. However, the real fruits of the Q-Learning algorithm can be seen in Figure 18, which is a trajectory that uses the policy derived from Q-Learning. We see here that the pendulum is controlled in 15 time units and remains controlled for the rest of the trajectory. Finally, the effect of varying ϵ and α on the learning curve is shown in Figure 20. We see that the learning curve is very sensitive to ϵ and α, especially needing high exploration values for ϵ to gain useful returns.

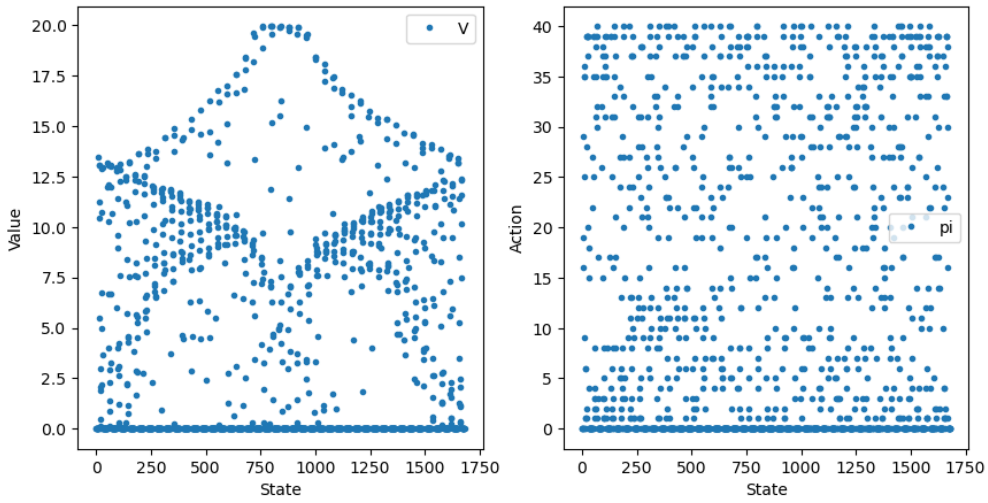

Figure 19: Pendulum State Value Function and Policy for Q-Learning using TD(0)

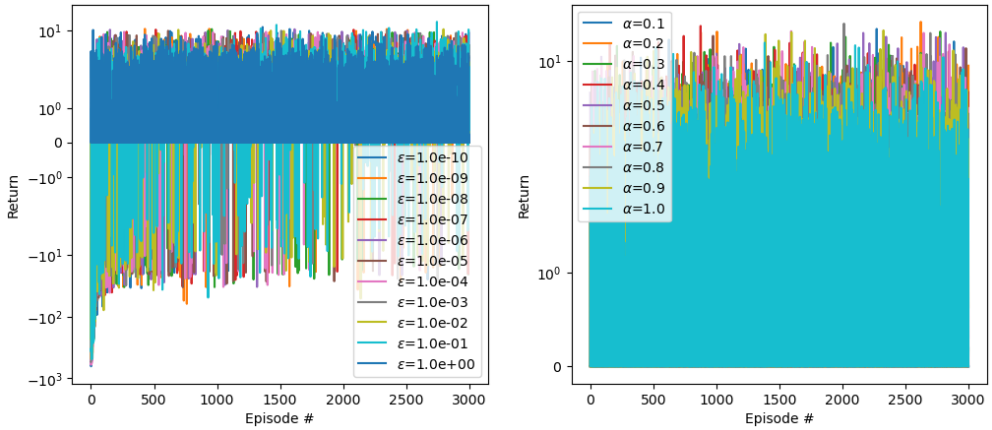

Figure 20: Pendulum Effect of ϵ and α on Learning Curve for Q-Learning