National Taiwan University
Department of Computer Science
Name: Yu Xiang, Luo
ID: B10902037

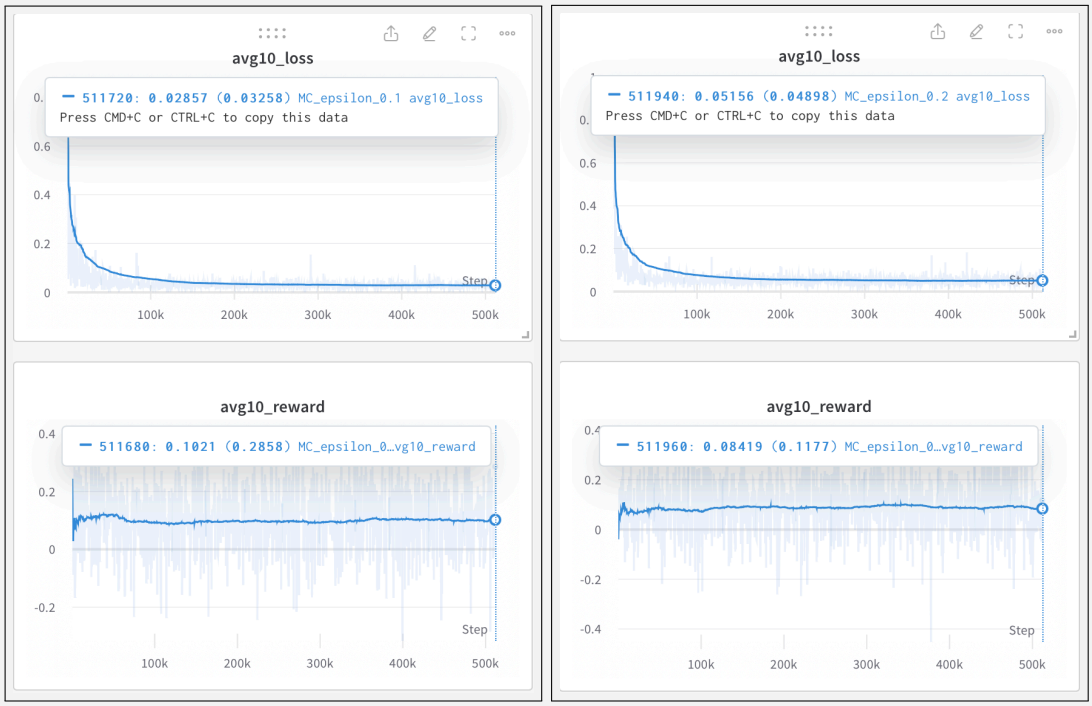
112-1 (Fall 2023)
Reinforcement Learning

Assignment 2 Report

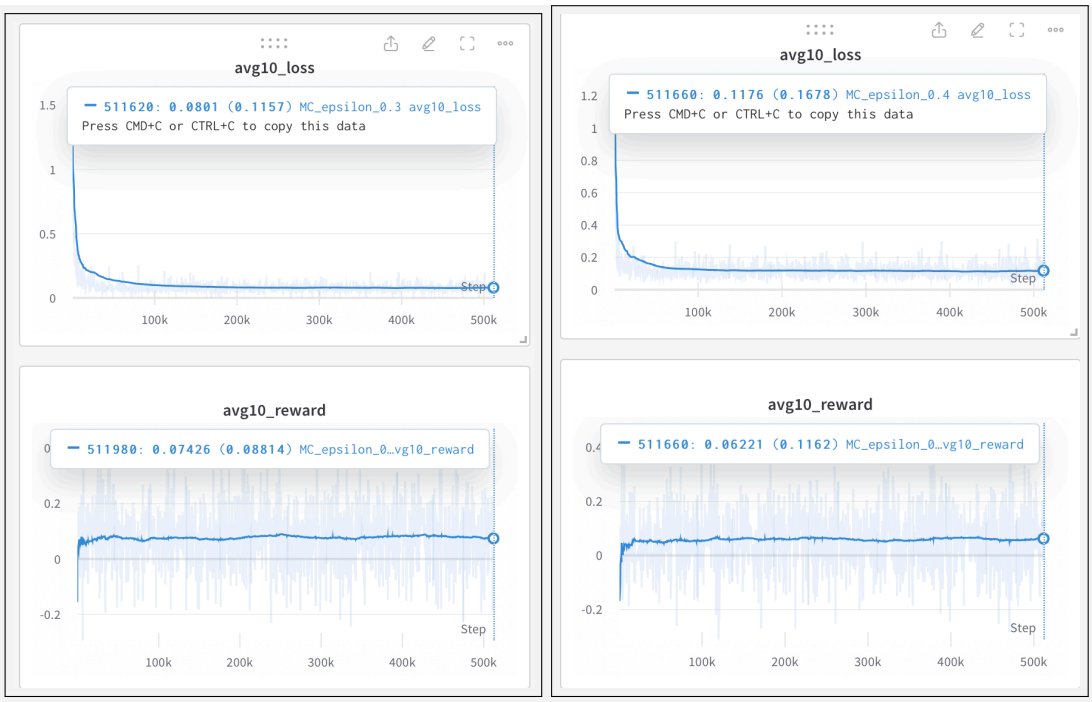
Q1 & Q2. Discuss and plot learning curves under ϵ values of $(0.1, 0.2, 0.3, 0.4)$ on MC, SARSA, and Q-Learning

I'll discuss Q1 and Q2 together, and consider each model separately first.

- MC

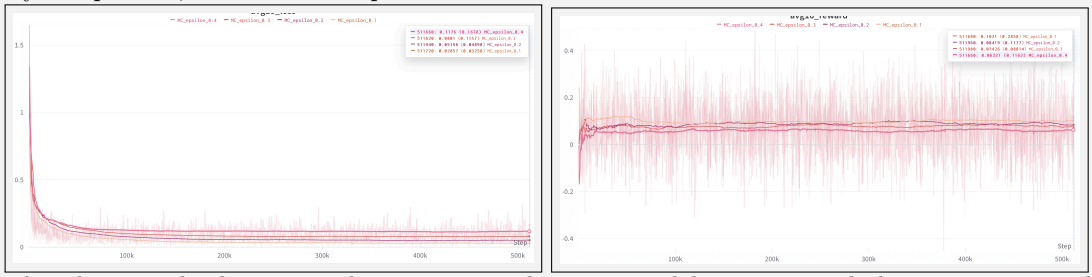


The left one's $\epsilon = 0.1$, and the right one's $\epsilon = 0.2$

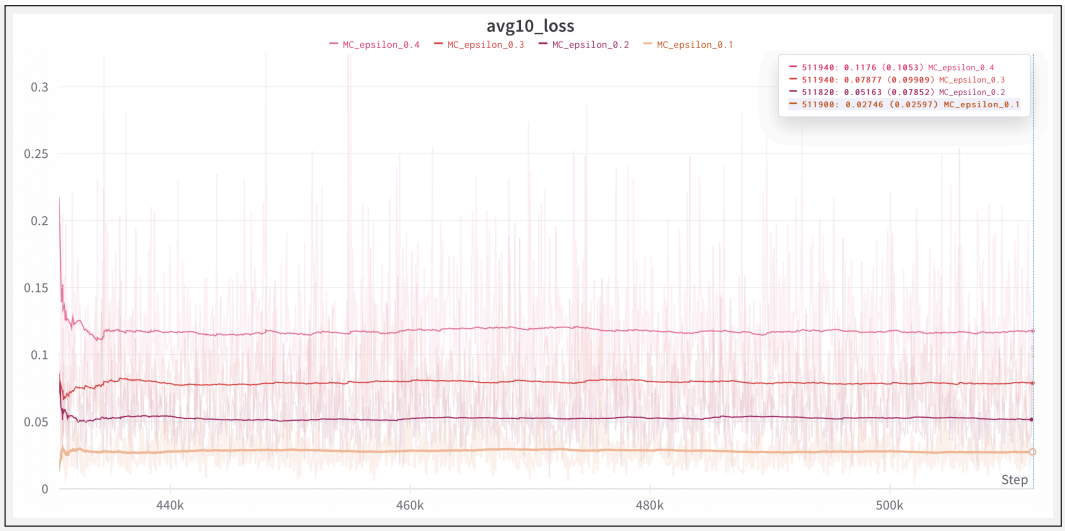


The left one's $\epsilon = 0.3$, and the right one's $\epsilon = 0.4$

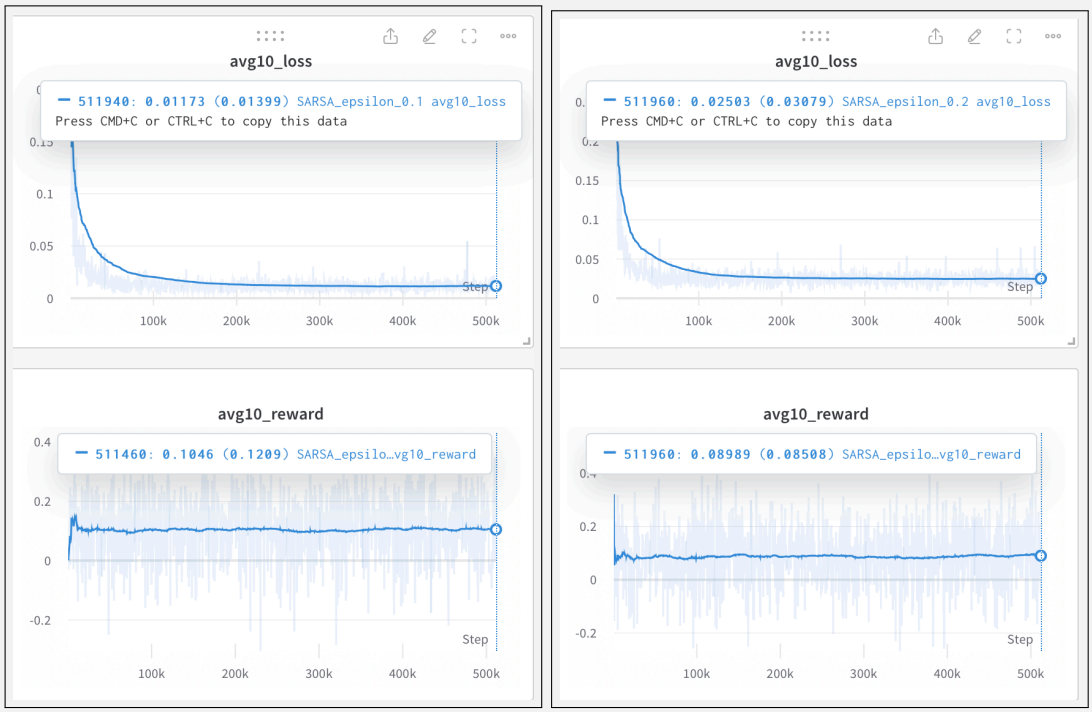
By comparison, we can see the plot below:



The plot reveals that as epsilon increases, the converged loss rises, and the average reward decreases. This observation aligns with the understanding that the epsilon-greedy approach isn't ideal for an environment with a well-trained Q value. Consequently, higher epsilon values result in greater converged losses. In terms of rewards, environments with smaller epsilon values tend to yield higher rewards due to a greater likelihood of choosing the optimal action. Also, the smaller epsilon values needs more iteration steps to converge, compared to the bigger one.

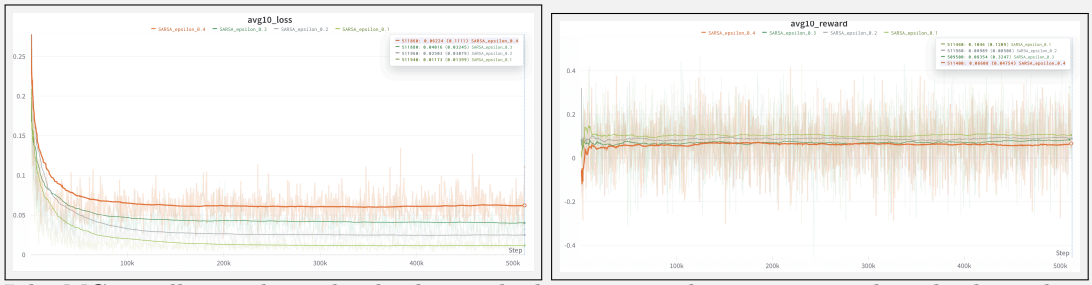


- SARSA



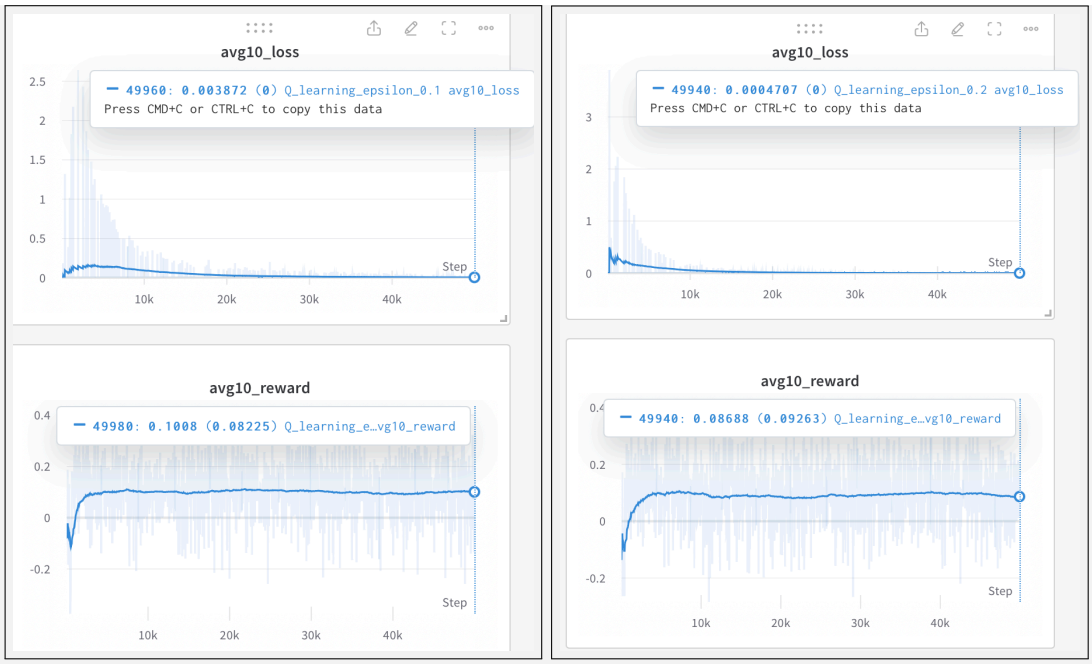
The left one's $\epsilon = 0.1$, and the right one's $\epsilon = 0.2$

The left one's $\epsilon = 0.3$, and the right one's $\epsilon = 0.4$

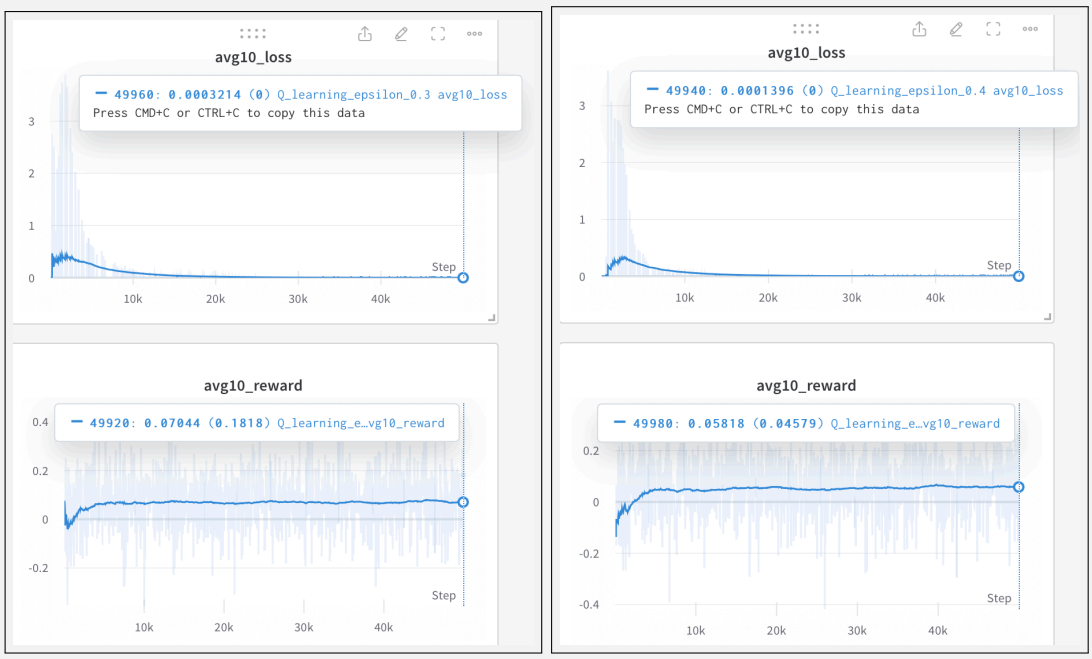


Like MC, smaller epsilon value leads to a higher converged average reward, and a lower loss. Overall, the attribute is quite similar to MC, in terms of a long term iteration.

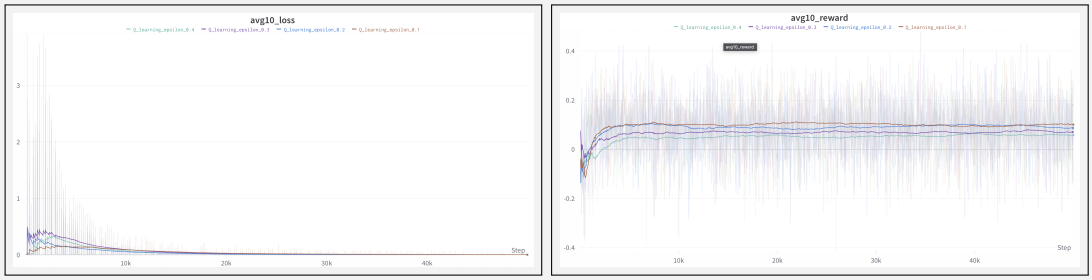
- Q-learning



The left one's $\epsilon = 0.1$, and the right one's $\epsilon = 0.2$



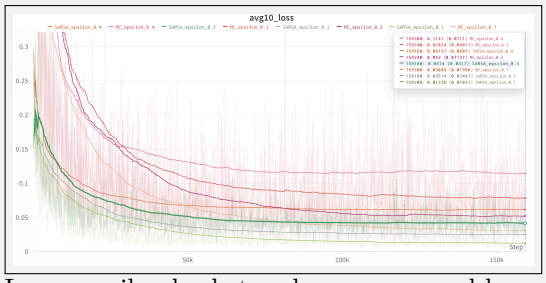
The left one's $\epsilon = 0.3$, and the right one's $\epsilon = 0.4$

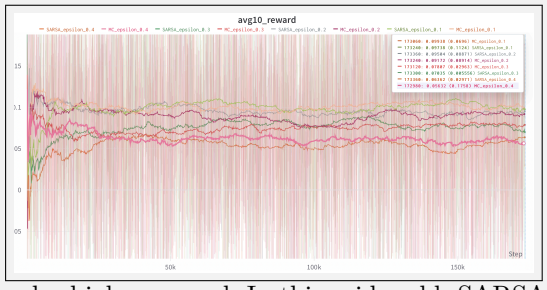


In Q learning, the loss converge at a early step, and smaller epsilon also leads to a higher

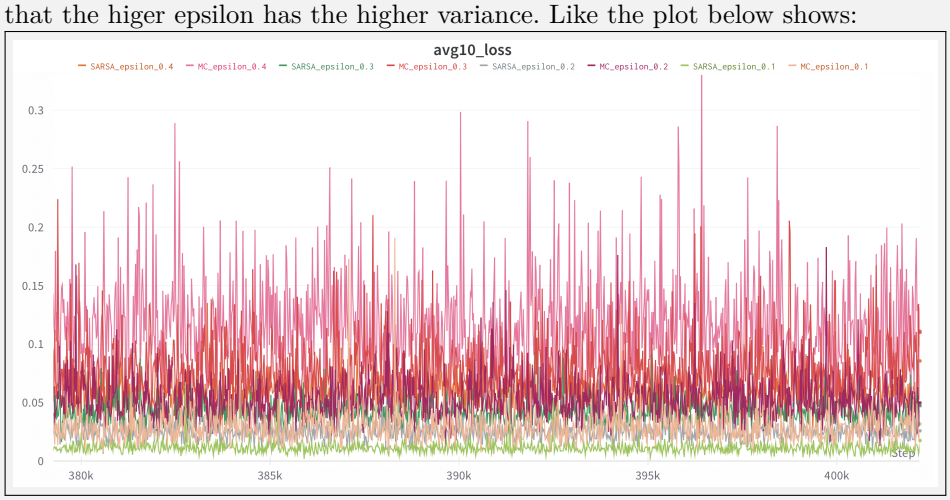
converged average reward. It's interesting that the smaller epsilon has a higher loss.

- Comparison

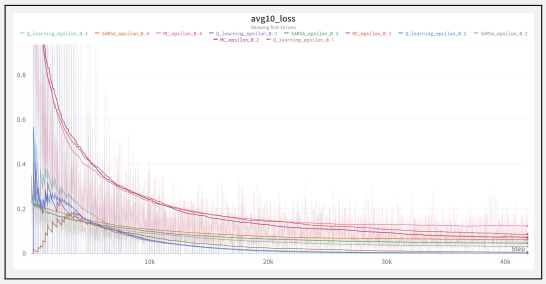


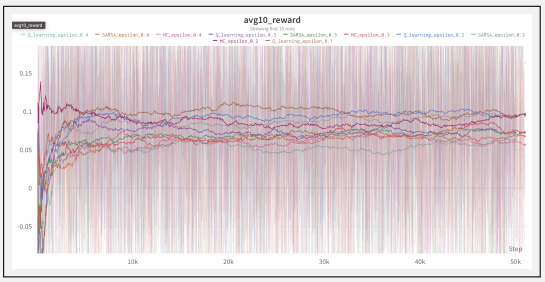


Lower epsilon leads to a lower converged loss, and a higher reward. In this gridworld, SARSA and MC seem to have similar performance in same epsilon. Apart from this, it's also visible that the higer epsilon has the higher variance. Like the plot below shows:



SARSA ($epsilon_{0.4}$) has the highest variance.





If comparing with Q-learning, MC and SARSA's performance are lower than Q-learning.

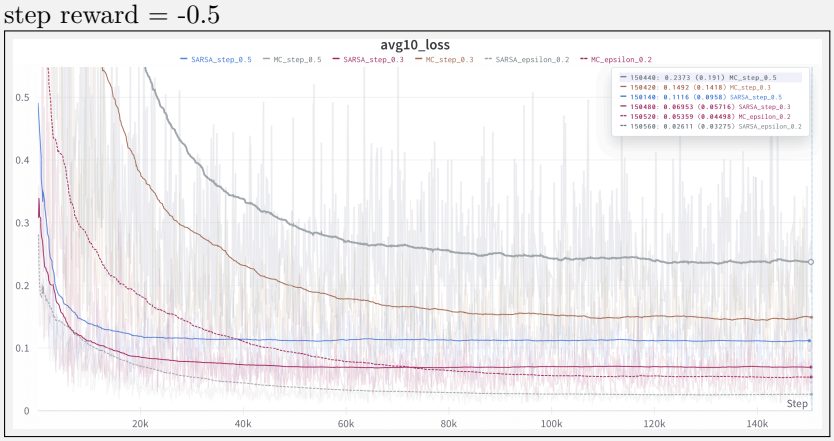
Q3. Discuss and plot ...

- step reward

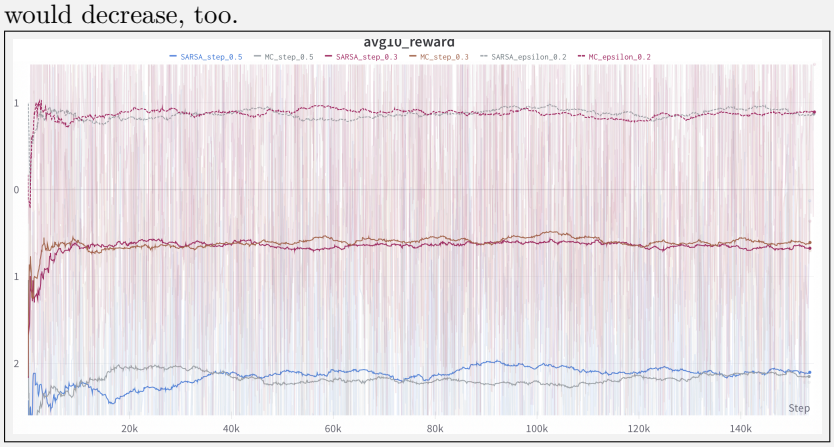
  - step reward = -0.3

This graph shows the difference speed of convergence in different reward(original step reward = -0.1). As the result shows, the smaller step reward can boost the convergence to a early step. The result is more evident when step reward is smaller.

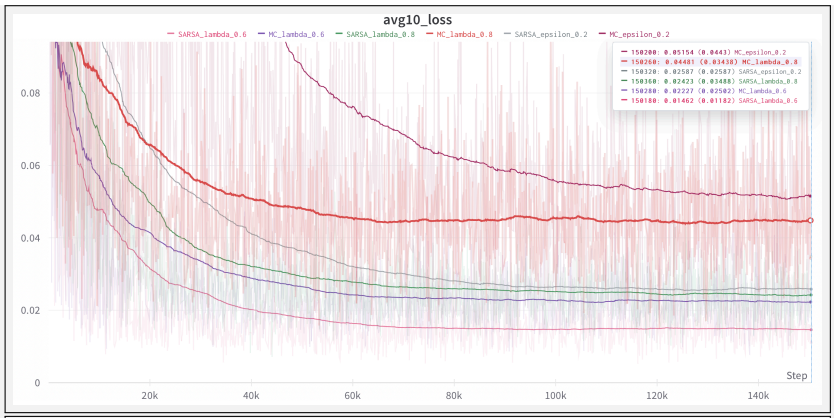
– step reward = -0.5



The reward plot is easy to imagine. As step rewards gets smaller, the average reward would decrease, too.

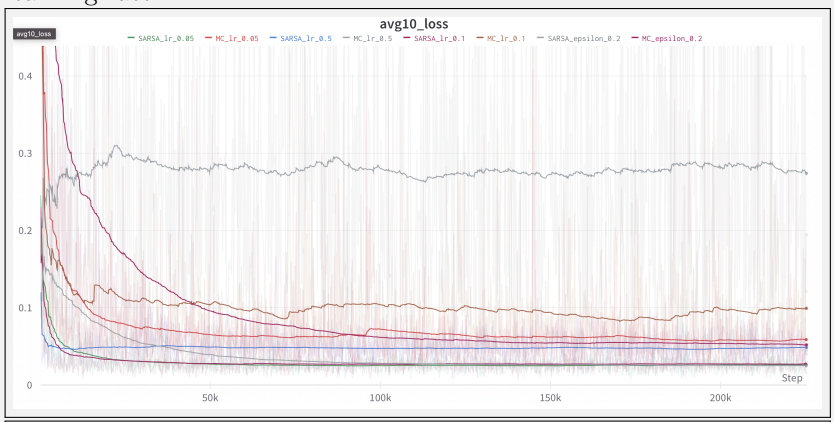


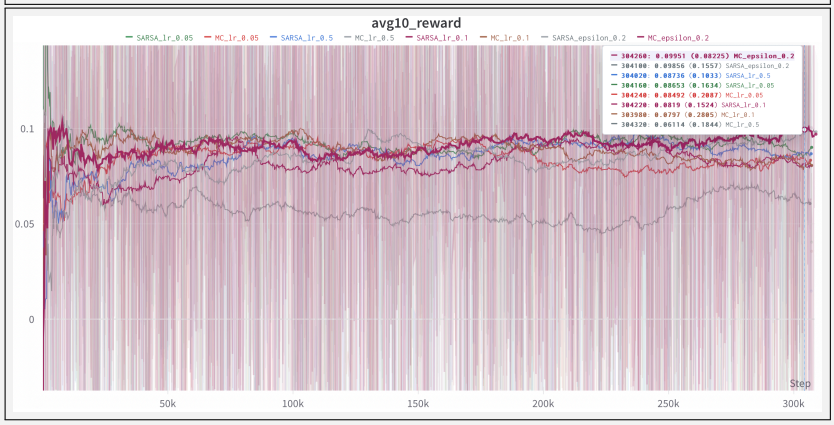
- discount factor



The smaller the discount factor, the lower the loss is. However, the smaller loss doesn't guarantee the bigger average reward.

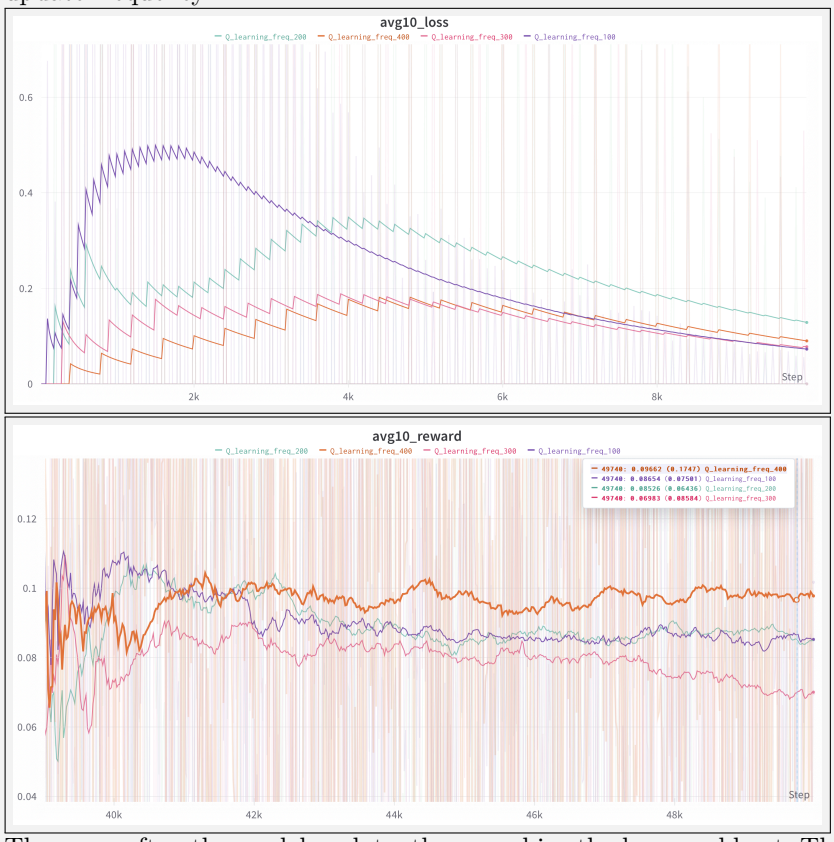
- learning rate





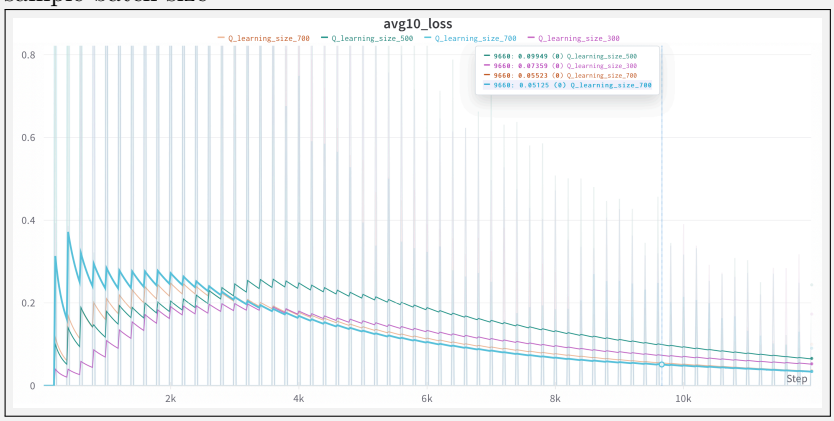
A bigger learning rate can converge in a early step, yet the average reward goes down, too. Also, MC using bigger learning rate might affect to its loss drastically. In spect of average reward, the one with the smallest learning rate(0.01), has the best reward.

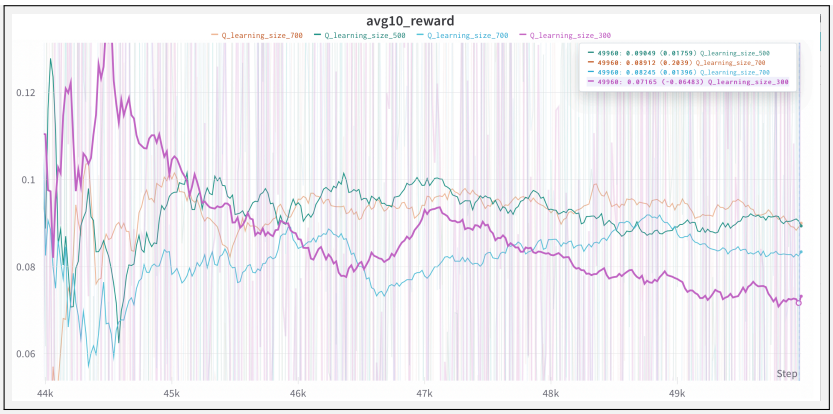
- update frequency



The more often the model update, the more bias the loss would get. Therefore, it's reasonable freq 100 has the highest loss at the beginning. As for the reward, since the loss converge well for four cases of frequencies, so it's the average value are similar to four different situations, yet the learning speed is different due to different frequencies.

- sample batch size





In loss plot, the initial loss is big if batch size is large, which might be the reason that too many mistakes need to be fix. Then, the loss decreases and the smallest lost is held by the biggest batch size, which is a sign of stability.
Also, the reward of these batch size all converges well, but it's still a good practice to check which batch size's average reward grows the fastest.