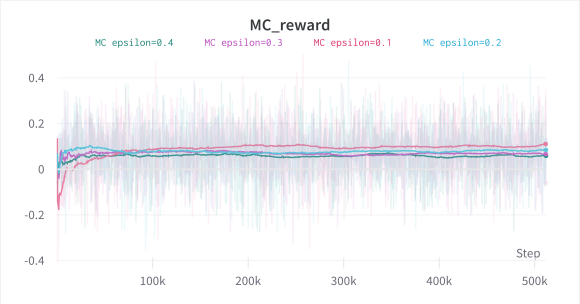
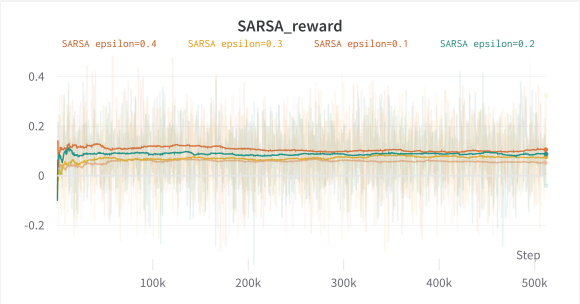
National Taiwan University
Department of Electrical Engineering
Name: An-Che, Liang
ID: B11705009

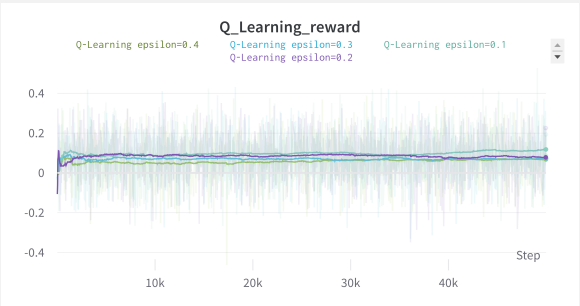
112-1 (Fall 2023)
Reinforcement Learning

Assignment 2 Report

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

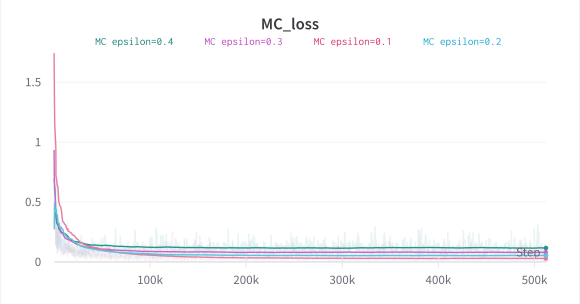


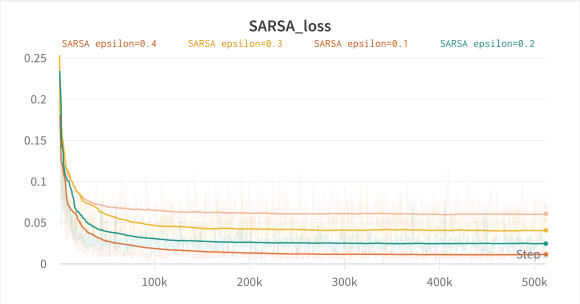


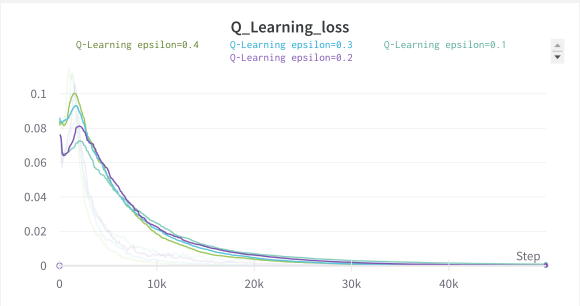


Above are the learning curve of **MC**, **SARSA** and **Q-Learning** models with different ϵ settings. We can see since our testing dataset is relatively small, thus all the model will converge to the optimal solution pretty quickly. The higher ϵ value generally cause the model to converge quicker, but since the ϵ value means to try different actions rather than acting greedy, the models with higher ϵ value will perform worse than those with lower ϵ value.

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



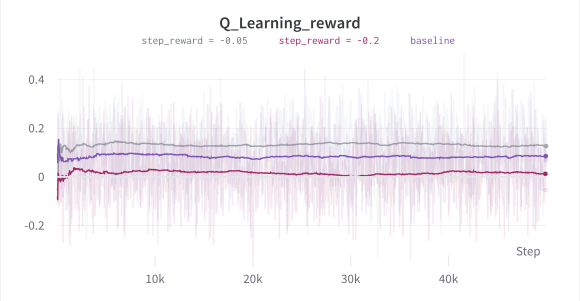


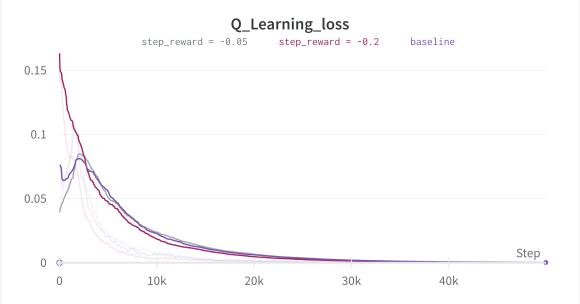


Just like the result we get in the previous section, all the models with different settings will converge pretty quickly. The model with higher ϵ will converge quicker in the beginning, but perform worse in the long run. To address this problem, one can make the ϵ changes with regard of the loss value, thus encourage exploration in the beginning and greediness in the long run.

Q3. Discuss and plot ...

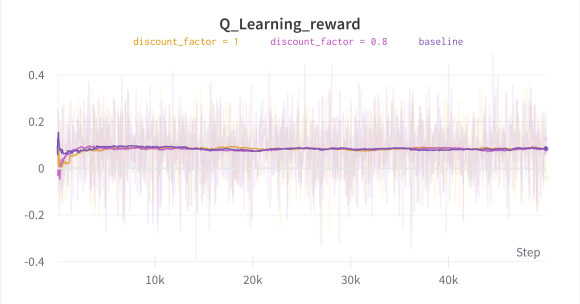
Step Reward

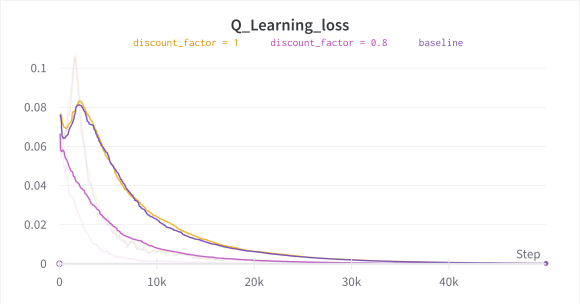




We can see that with `step_reward` = -0.2, we can eliminate the weird hill-like structure which appeared in the baseline model. I think thus it is better suited for this dataset. Change the step reward reward with smaller results in negligible changes.

Discount Factor

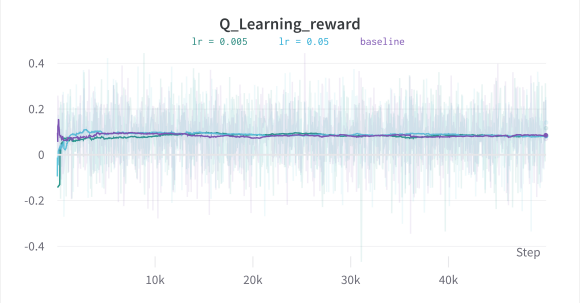




We can see that with `discount_factor` = 0.8, the model performs better, I thinks that's because

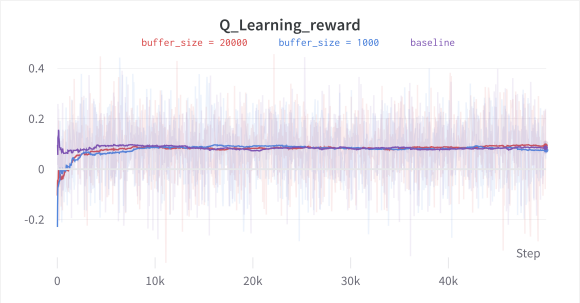
our dataset is relatively small, therefore the model will reach the terminal state with just a few steps, making high discount factor relatively pointless since the model will reach the terminal state eventually.

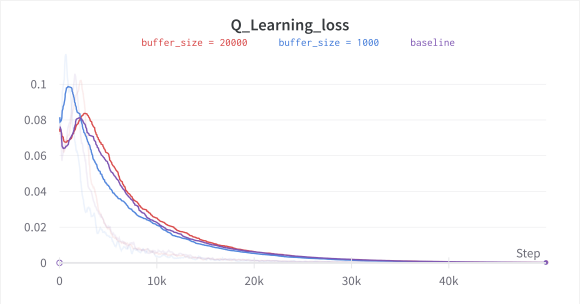
Learning Rate



We can see that the change in learning rate functions just like the learning rate in supervised deep learning, raising the value value will cause the model to learn faster in the risk of missing the optimal solution, but since this dataset is really small, I didn't observe our model been stuck in some local optimal solution.

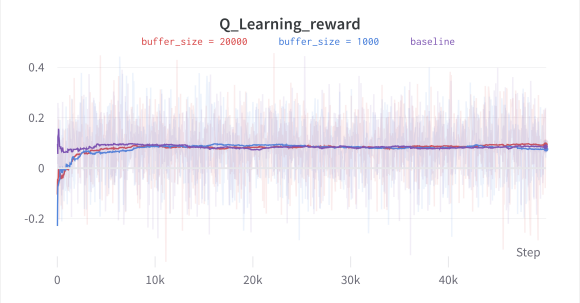
Buffer Size

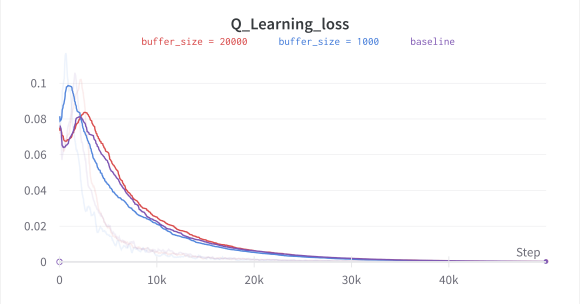




Intuitively, larger the buffer, the records stored in the buffer is more generalized and less prone to randomized results, thus the model with larger buffer will learn slower but less violate (with the loss value).

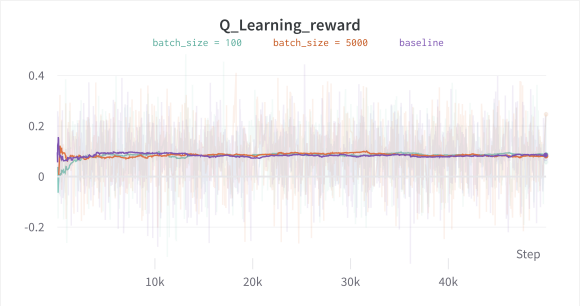
Update Frequency

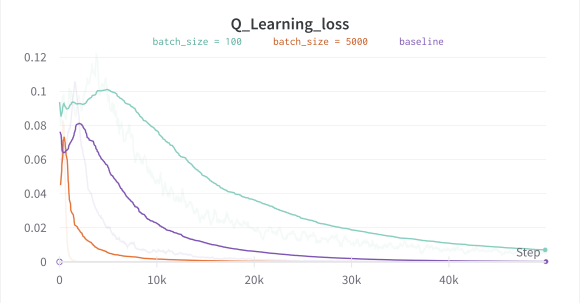




The change in update frequency functions somewhat just like buffer size does, more updates means the model will change gradually, and vice-versa.

Sample Batch Size





Though with fewer sample in each batch will cause the model to perform poorly, but since the process of updating value function is linearly related to batch size, by lowering the sample batch size, one can reduce the training time of the model by a large margin.