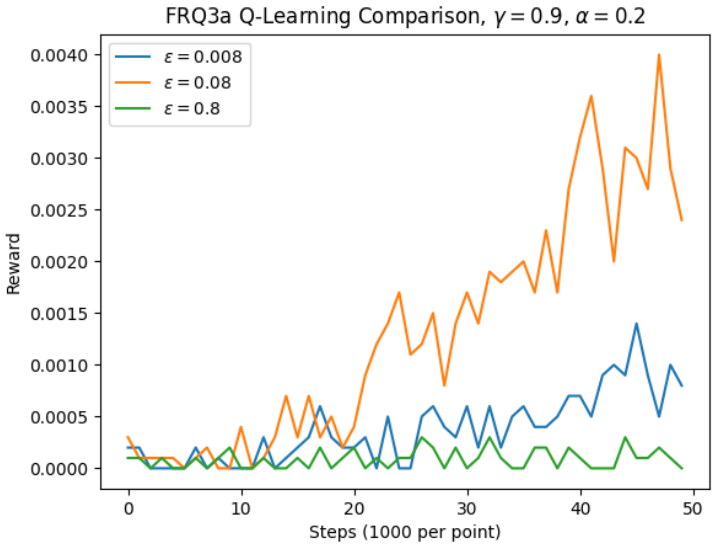
3a.

Plot:



3b.

The best value of epsilon seems to be 0.08, with 0.8 being the worst. Epsilon=0.008 is in the middle. This may be explained by the fact that epsilon=0.8 explores too much and does not exploit enough so it gets very low rewards, while $\varepsilon = 0.08$ is able to explore faster than $\varepsilon = 0.008$ and thus able to get better rewards.

3c.

I would expect the reward for the $\varepsilon = 0.8$ case to remain the same since it has likely learned all it is going to learn but will continue to be weighed down by needless exploration. I would expect the reward for the $\varepsilon = 0.008$ case to keep growing and eventually take the lead as it will take a long time to explore, but once it does it will very consistently choose to exploit the best path. I would expect the reward for the $\varepsilon = 0.08$ case to keep growing but eventually fall behind the $\varepsilon = 0.008$ case since it will eventually be choosing to exploit the optimal path less often.

3d.

The danger of choosing epsilon this way is that with so many epsilons so close to one another, noise in your training processes may provide confusion when training and cause you to “overfit” epsilon to the particular domain you trained on. This may result in worse performance when training on a new domain.