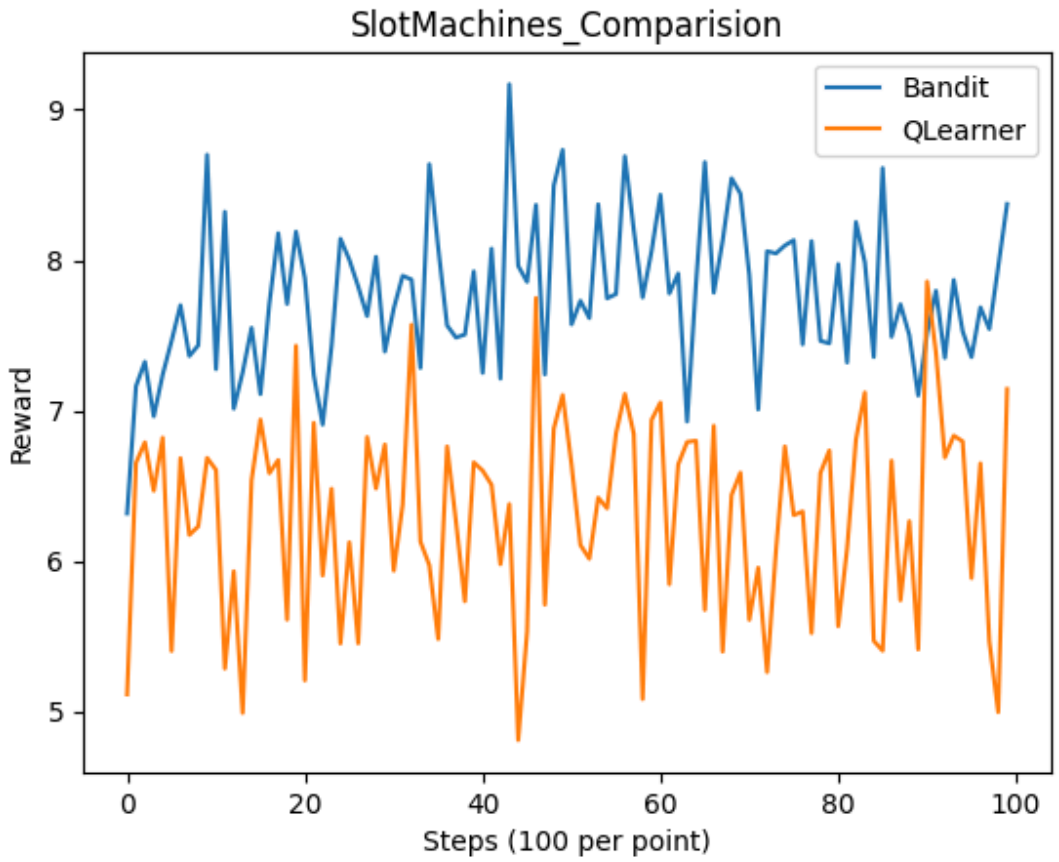
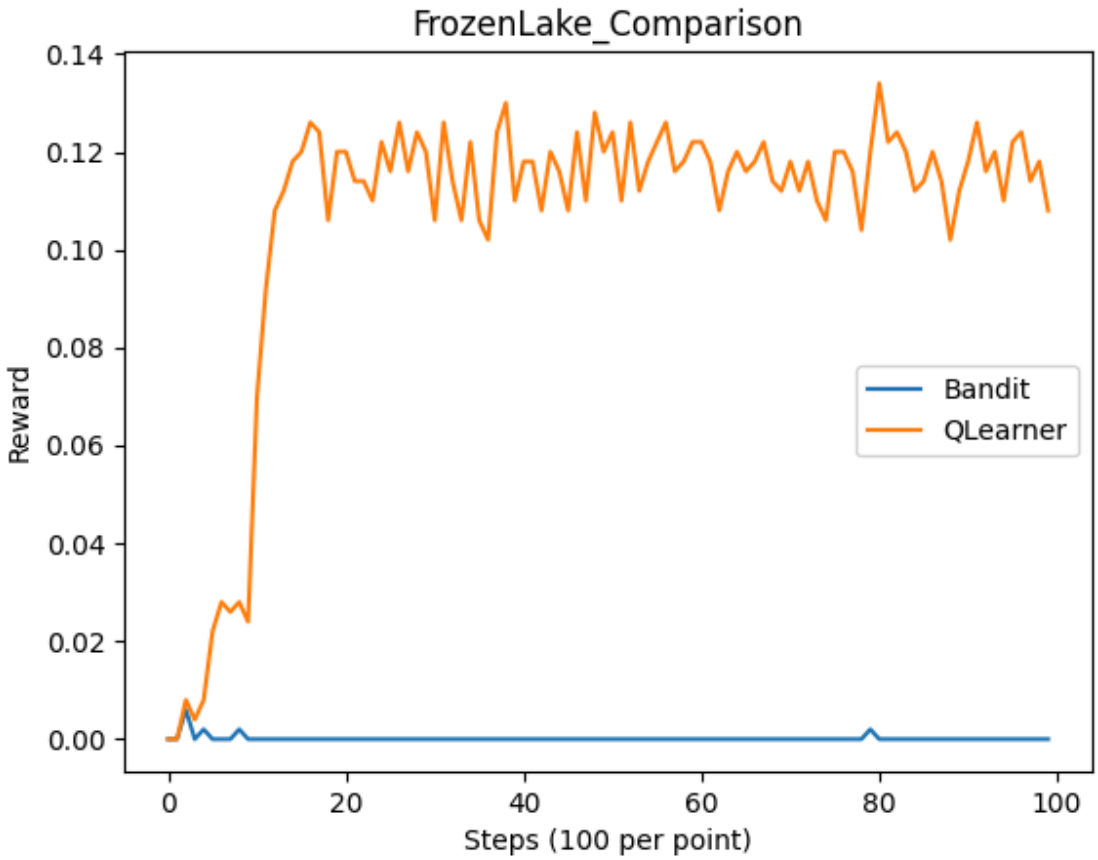
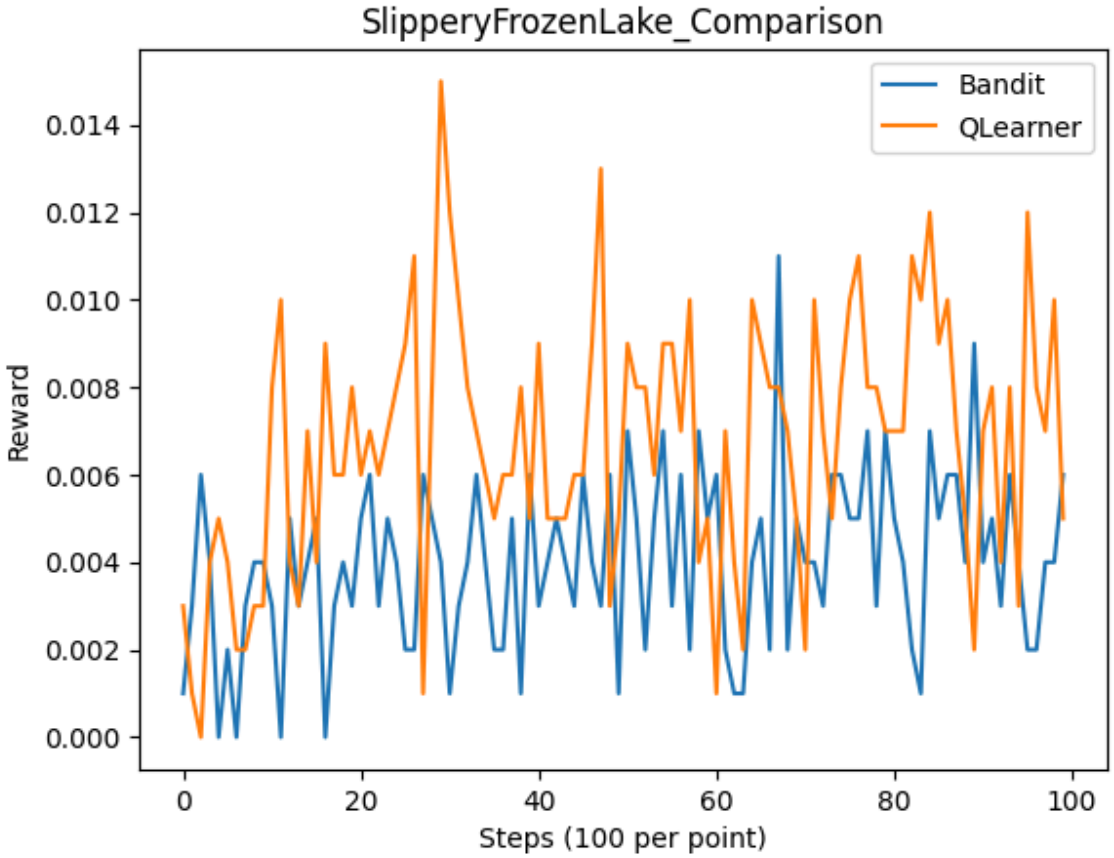


In the slot machines environment Bandit ends up performing better, because the future rewards are non-deterministic.



In the frozen lake environment QLearner performs better, because it can evaluate how close the best step from the next step(as a result of taking an action) can get the navigator.



In the Slippery Frozen lake environment, there is not a clear winner. That's simply because the information of choosing the best step from the next step is nondeterministic. However, overall QLearner performs slightly better, because there is still a ⅓ percent chance that max_a' Q(s', a') will provide the correct information(meaning that the navigator will not slip and move towards the intended direction).

b. Q Learning definitely performs better on the Frozen Lake and very slightly better on the Slippery Frozen Lake. That's simply because in Frozen Lake the information gathered towards the goal state is deterministic. That's not the case for slot machines.

For the Slippery Frozen Lake there is ⅓ percent chance that the information gathered towards the goal state will be correct. That ends up leading the QLearning to perform slightly better than MAB.

c. MAB is definitely worse on Frozen Lake and slightly worse on Slipper Frozen lake. There is no way to change the hyperparameters to make it perform as well as QLearning, because evaluating future consequences of current actions is not a part of MAB.

d. MAB performs better on slot machines because there is no notion of current action leading to a better possible action. So the notion of the QLearning algorithm is not fit for slot machines problem. The best way to exploit gained information in slot machines is to choose the action with the highest expected reward. Which is what the MAB is designed to do.

e. Yes, setting gamma to 0, would lead the algorithm to only consider current rewards. Furthermore, the two algorithms MAB and QLearning can be made equivalent with the correct alpha value.