Reinforcement Learning Project

Lufan Zhou - 100690462
Peng Zhang - 100515945
ELEC-E8125 - Reinforcement Learning

December 13, 2022

1 Part I

1.1 Question 1

In the first environment, we trained lunarlander using two methods: DDPG and DQN. In this case, DDPG has the best performance with respect to DQN because it uses the actor-critic model, the former to update the action and the latter to determine the Q value. Therefore, we can say that DDPG tries to obtain the policy directly, while DQN has to determine the Q value to update the policy. According to the training results, DDPG is more sample-efficient; we can see from the graphs that they require fewer episodes to achieve the goal, but in clock-time, DDPG is larger than DQN, which means that they require more calculations because the number of parameters to be treated is much larger and requires more computation than only DQN.

In the second environment, we have trained Bipadelwalker using DDPG and PG_AC. DDPG is a better method than PG_AC because it needs to adjust parameters that are not as strict as PG_AC. On the other hand, with the same hyperparameters, we have not been able to reach a positive number in PG_AC, and it is observed that it is not so powerful for this environment. In DDPG, we have obtained very decent results, reaching the target. So DDPG is more sample-efficient, but clock time is much higher than PG_AC.

1.2 Question 2

- DDPG was considered in determining the actor's learning rate and critic. If your actor changes faster than your critic, your estimated Q-value will not really represent the value of your action because that value is based on past policies. We also adjusted the buff size to get enough data to train the environment to achieve the target.

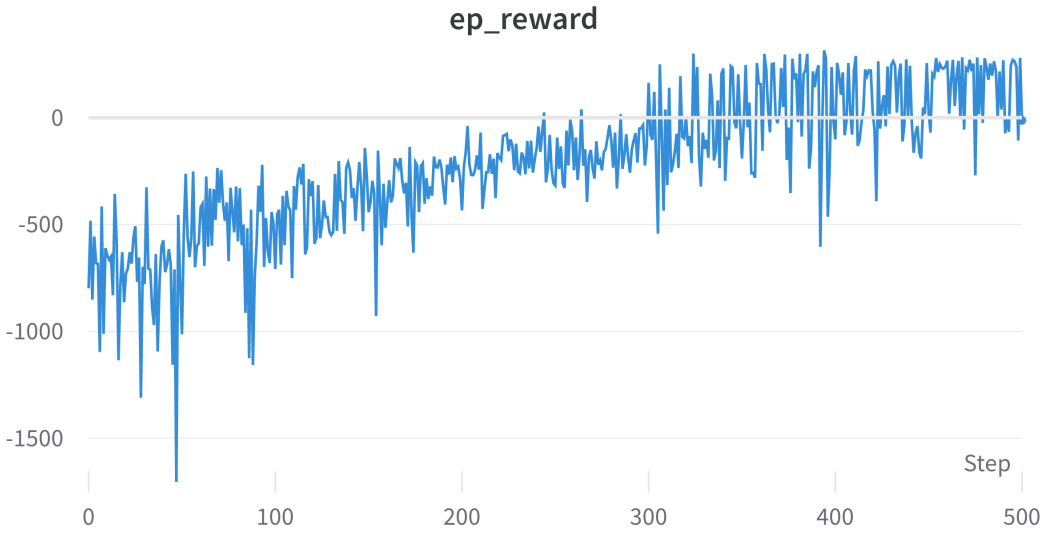

Figure 1: Plotting 1 of average reward by DDPG in Lunarlander

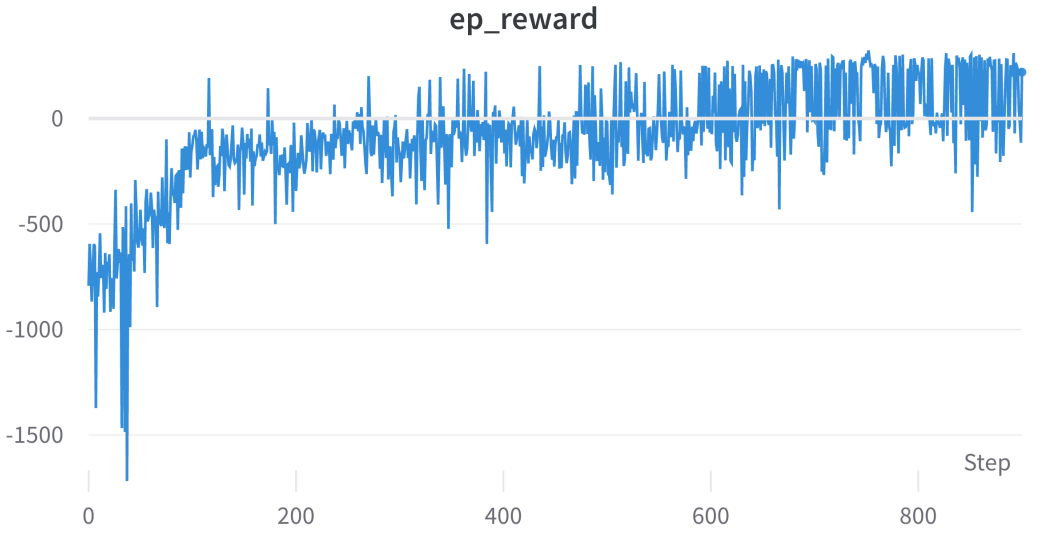

Figure 2: Plotting 2 of average reward by DDPG in Lunarlander

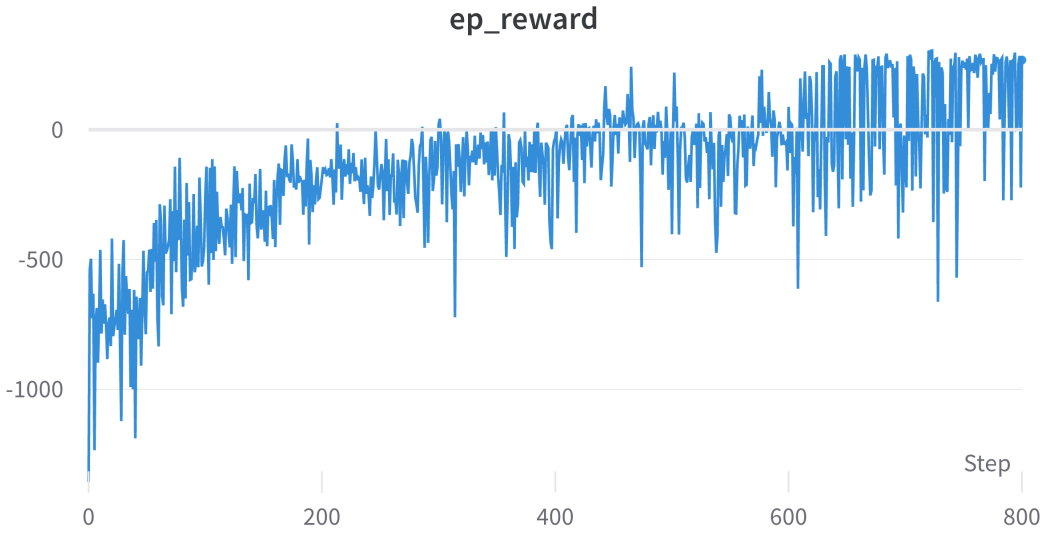

Figure 3: Plotting 3 of average reward by DDPG in Lunarlander

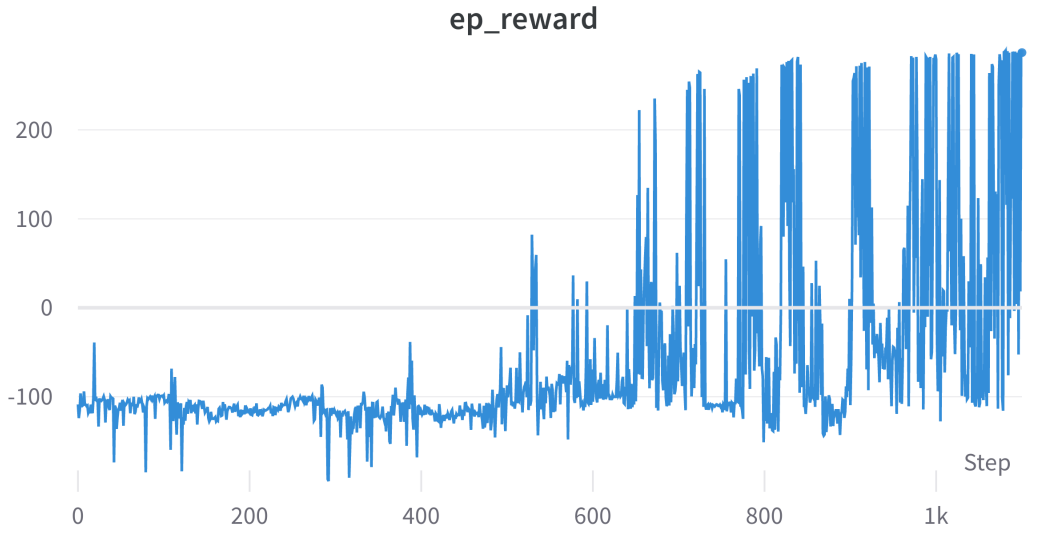

Figure 4: Plotting 1 of average reward by DDPG in BipedalWalker

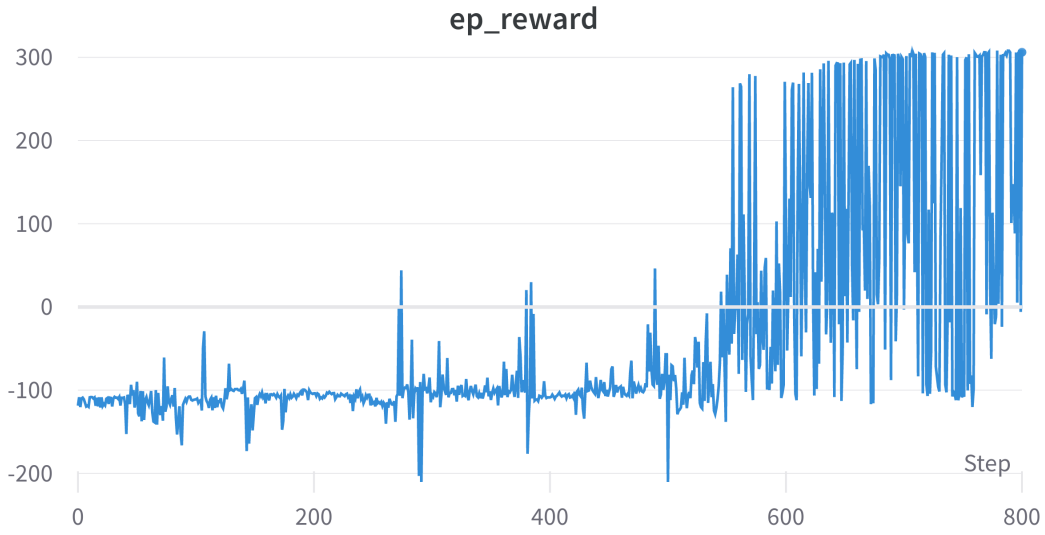

Figure 5: Plotting 2 of average reward by DDPG in BipedalWalker

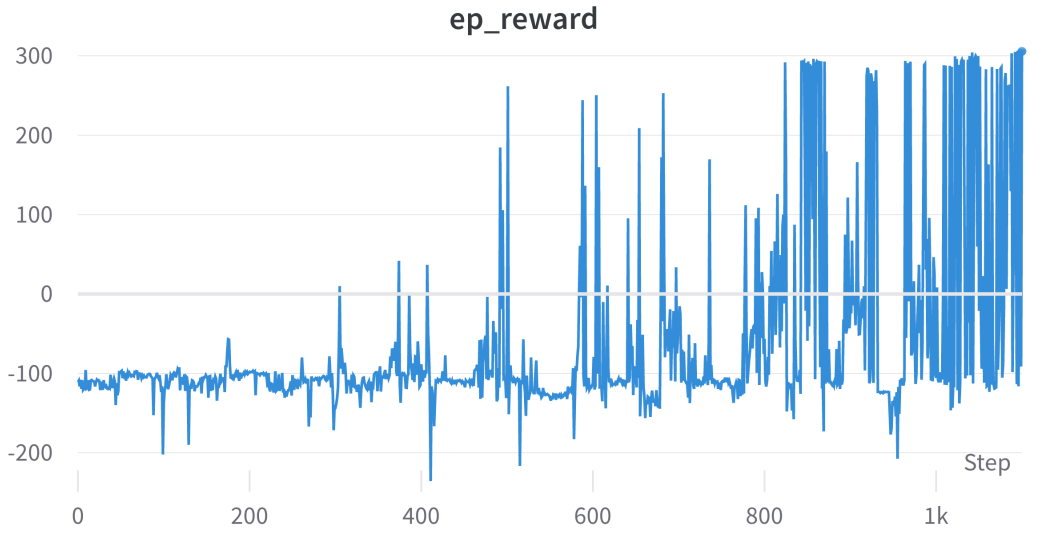

Figure 6: Plotting 3 of average reward by DDPG in BipedalWalker

- PG_AC had to be observed in the hyperparameters such as learning rate, buffer size and entropy coefficient, which serves as a regulator to achieve the objective. We were able to train with the hyperparameters, although we were not able to get good results.

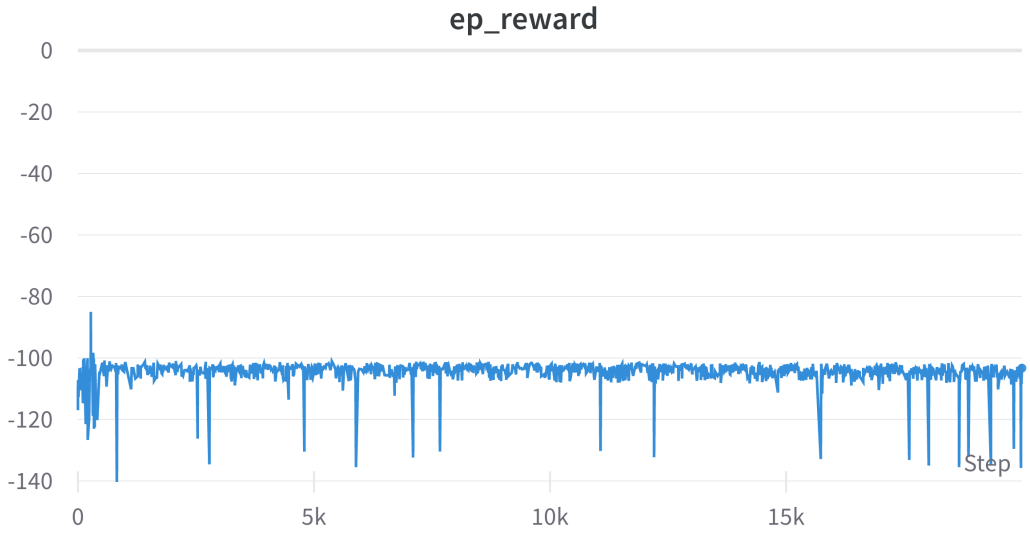

Figure 7: Plotting 1 of average reward by PG_AC in BipedalWalker

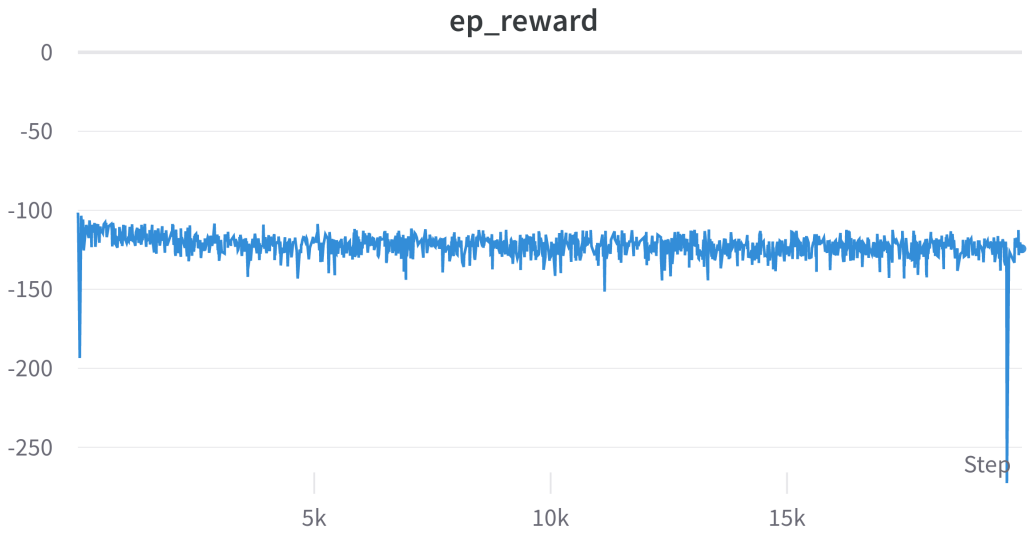

Figure 8: Plotting 2 of average reward by PG_AC in BipedalWalker

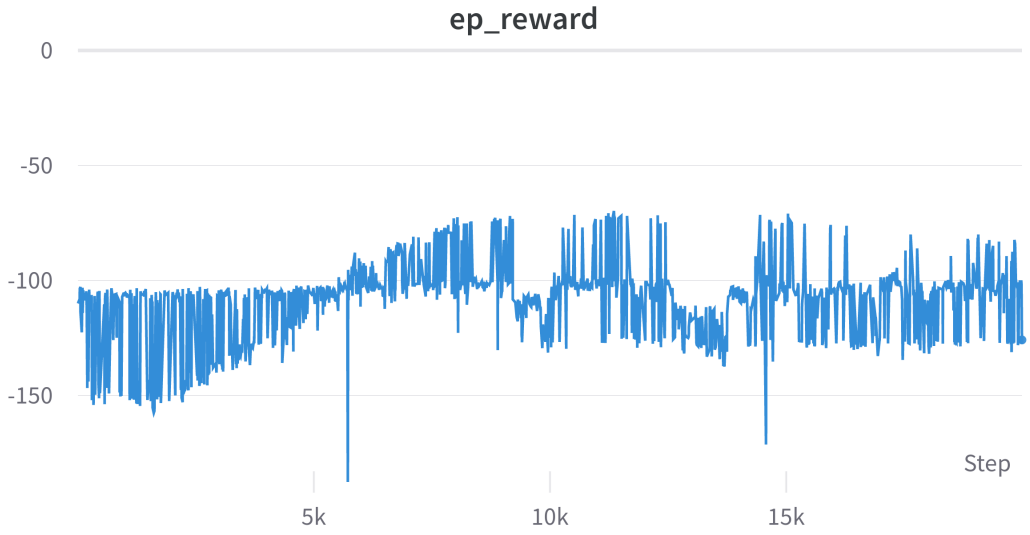

Figure 9: Plotting 3 of average reward by PG_AC in BipedalWalker

- DQN had to adjust the buff size as DDPG did to find sufficient data, and on the other hand, the learning rate of the critical method is important to know which is the best way to optimize the algorithm. For the hidden networks has been increased so that more networks can be used to reach the target.

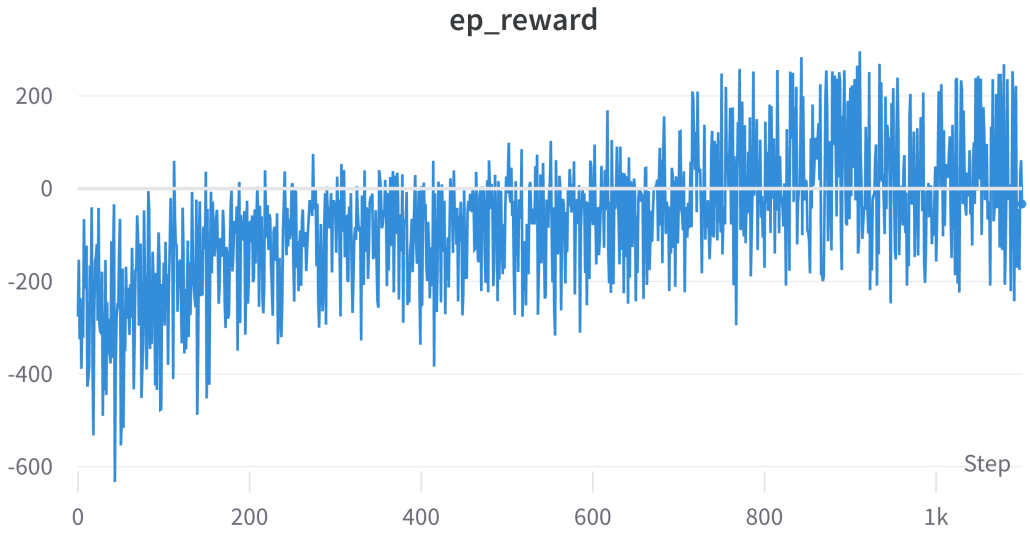

Figure 10: Plotting 1 of average reward by DQN in Lunarlander

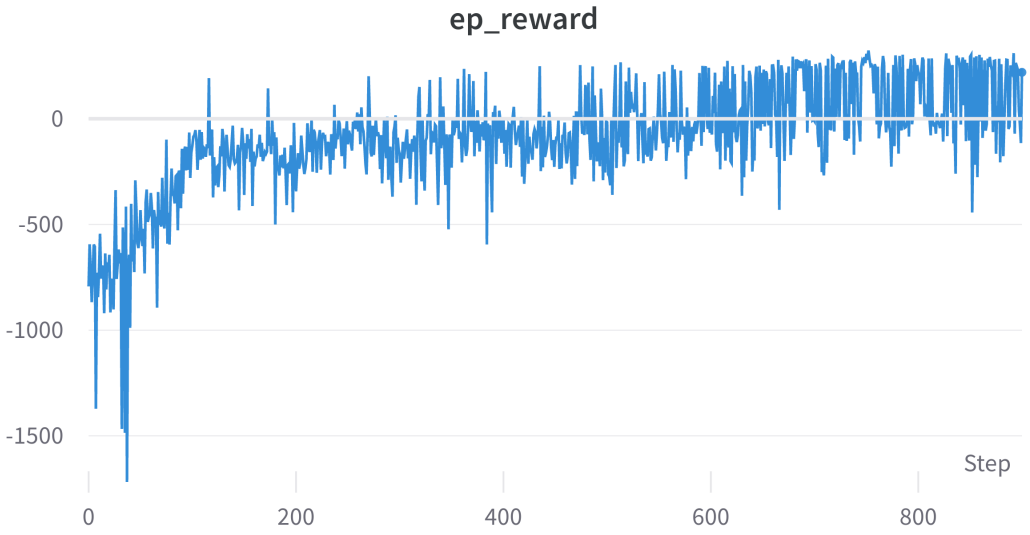

Figure 11: Plotting 2 of average reward by DQN in Lunarlander

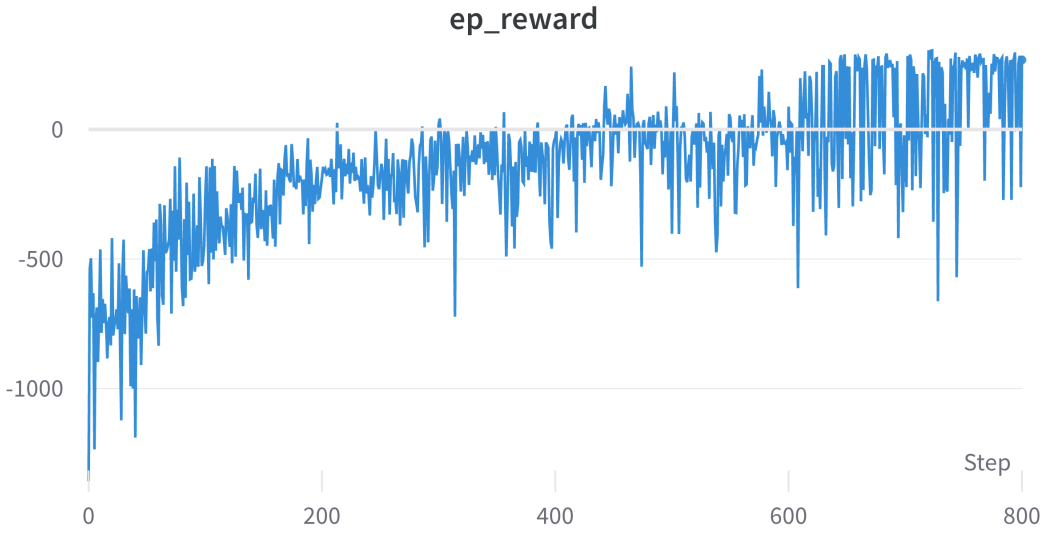

Figure 12: Plotting 3 of average reward by DQN in Lunarlander

Lunarlander average rewards

Lunarlander (Medium)	DDPG	DQN
Reward of the seed 1	134.4519	140.3754619787104
Reward of the seed 2	175.529519	119.36688657457094
Reward of the seed 3	202.443451	105.80019373197605
Mean	170.80829	121.8475141
Standard deviation	34.2407689	17.42060356

Table 1: Mean and standard deviation of 3 seeds in lunarlander

Bipadelwalker average rewards

Bipadel walker (Easy)	DDPG	PG_AC
Reward of the seed 1	264.3615690793662	-91.5575826
Reward of the seed 2	259.0836351631904	-65.26842556
Reward of the seed 3	256.5236541552157	-81.82624622
Mean	259.9896195	-79.55075146
Standard deviation	3.996727819	13.29147666

Table 2: Mean and standard deviation of 3 seeds in Bipadelwalker

2 Part II

2.1 Question 1

We used TD3 (Twin Delayed Deep Deterministic Policy Gradient) algorithm to improve our DDPG (Deep Deterministic Policy Gradient). TD3 is an extension of the DDPG algorithm that uses two separate networks, known as the actor and the critic, to learn the optimal policy for a given task.[1]

The main idea behind TD3 is to improve the original DDPG algorithm by addressing some of its known limitations. One of these limitations is that DDPG can have a high variance in the estimated action values, which can lead to unstable learning.

Instead of just one critic network, TD3 employs two. This helps to reduce the variance in the estimated action values, which in turn leads to more stable learning.

2.2 Question 2

Yes, from the plot we can see that by using TD3, the agent is more stable, and it needs less time to reach a certain reward.

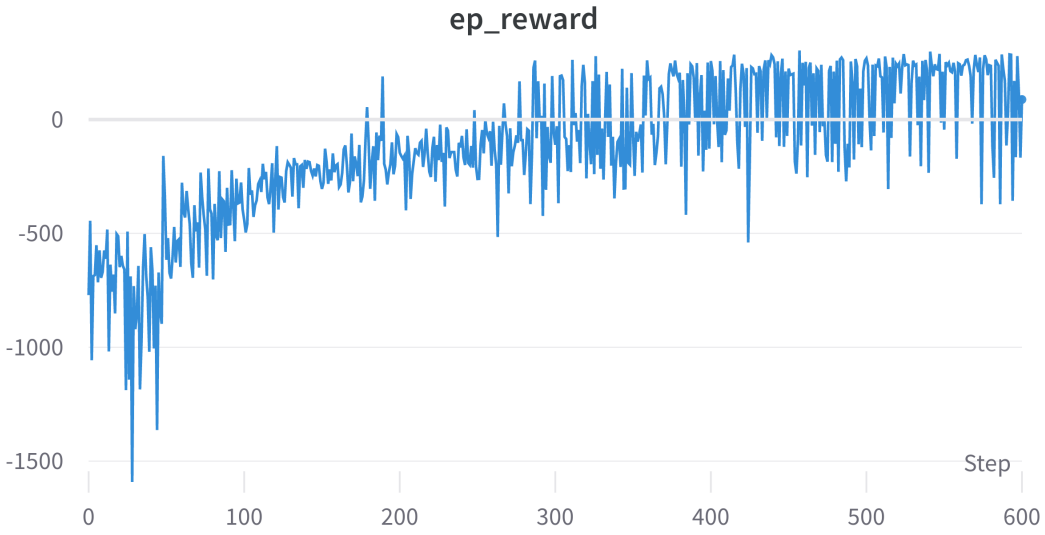

Figure 13: TD3's reward in Lunarlander 1

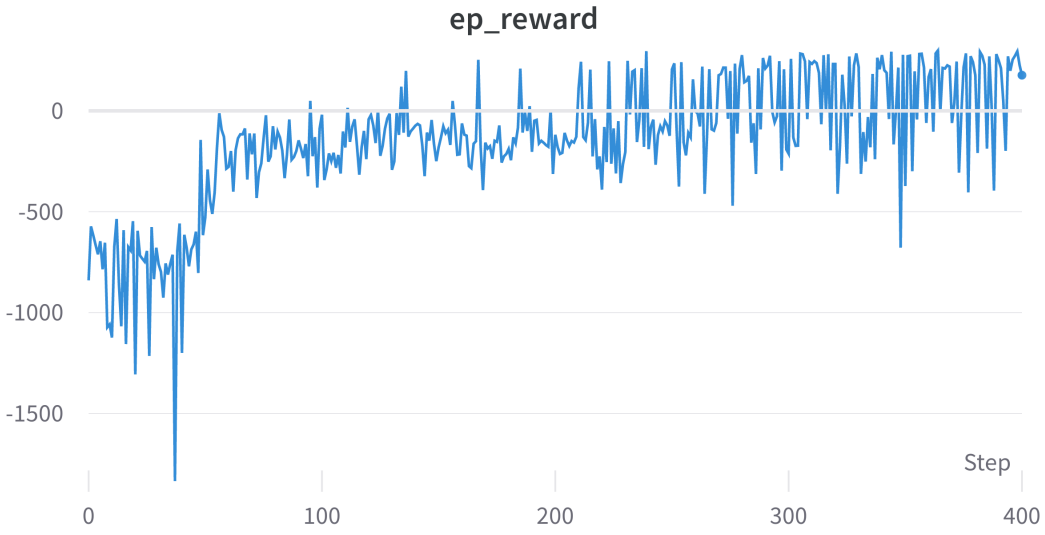

Figure 14: TD3's reward in Lunarlander 2

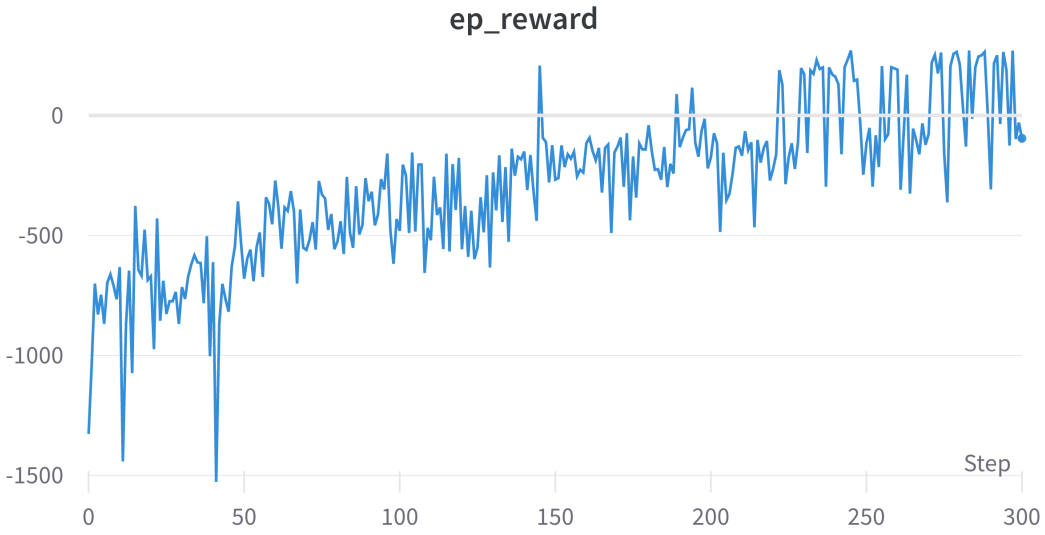

Figure 15: TD3's reward in Lunarlander 3

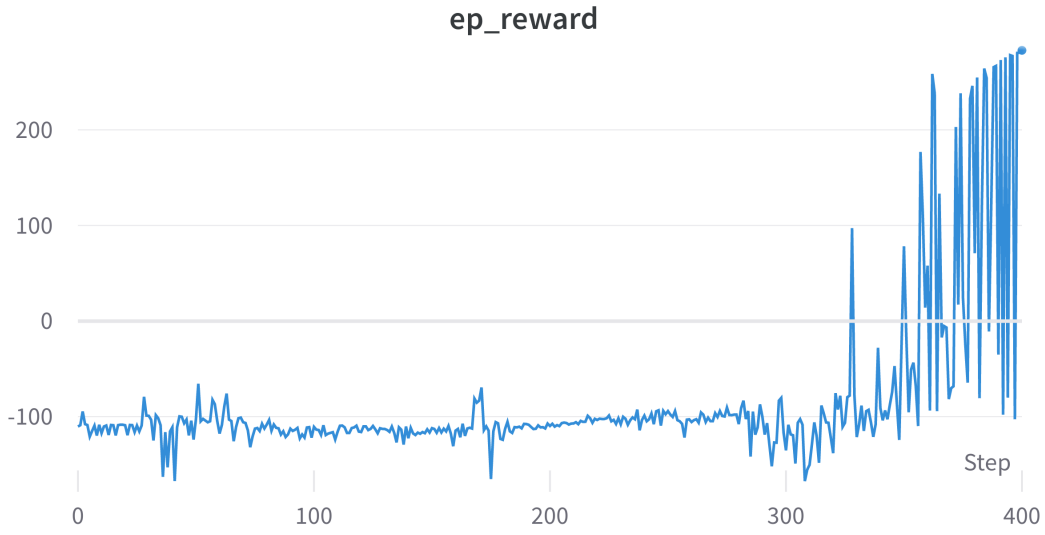

Figure 16: TD3's reward in Bipadel walker 1

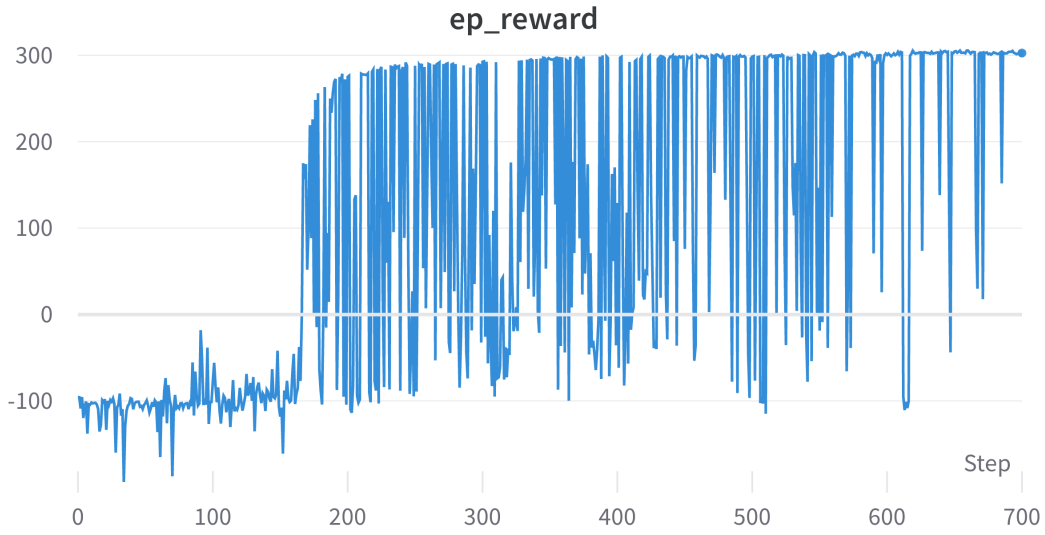

Figure 17: TD3's reward in Bipadel walker 2

Figure 18: TD3's reward in Bipadel walker 3

2.3 Question 3

Yes, the TD3 copes well with increased environment complexity because it is an extension of DDPG, which means it can work in any environment where DDPG is supported.

One of the key limitations of DDPG is that it can have a high variance in the estimated action values, which can lead to unstable learning. TD3 addresses this issue by using two critic networks instead of just one.

Lunarlander average rewards

Lunarlander (Medium)	TD3
Reward of the seed 1	154.170794
Reward of the seed 2	133.857745
Reward of the seed 3	112.879299
Mean	133.635946
Standard deviation	20.6466407

Table 3: Mean and standard deviation of 3 seeds in lunarlander

Bipadelwalker average rewards

Bipadel walker (Easy)	TD3
Reward of the seed 1	276.149131
Reward of the seed 2	290.6511994
Reward of the seed 3	259.4062672
Mean	275.4021992
Standard deviation	15.6358523

Table 4: Mean and standard deviation of 3 seeds in Bipadelwalker

3 Conclusion

According to the results obtained using the four methods, TD3 is the proposed method with the best performance. It takes less time to train and needs fewer episodes to reach the target environment. Although DDPG is a powerful enough method to train in the two environments chosen by us, we can say that we have made improvements to the method by using the double critic network. The other methods, such as DQN and PG_AC, take more time than DDPG; therefore, we discarded them. In the future, we can further improve the TD3 method to achieve results with more sample efficiency and shorter clock times.

References

[1] F. Zhang, J. Li, and Z. Li, "A td3-based multi-agent deep reinforcement learning method in mixed cooperation-competition environment," *Neurocomputing*, vol. 411, pp. 206–215, 2020.