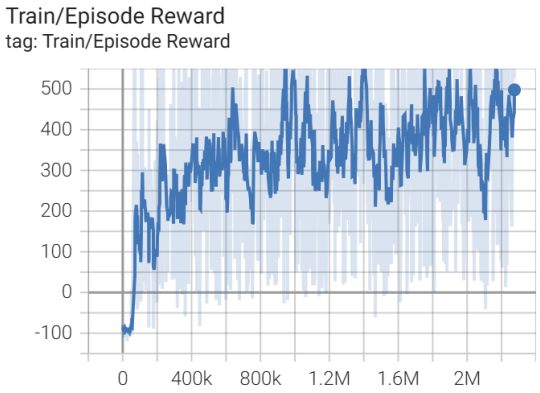
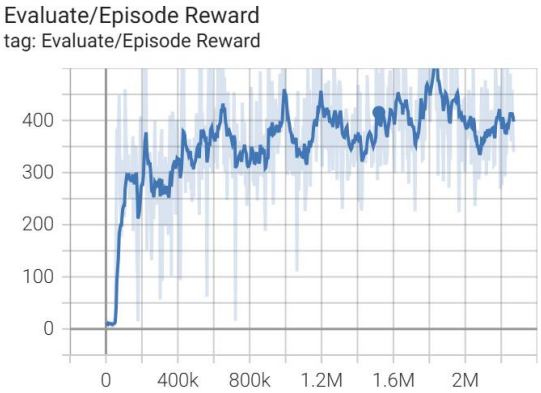
RL Lab 4 Report

資科工碩 陳冠廷 313551058

- **Basic**
  - **Screenshot of Tensorboard training curve and testing results on TD3**

    Smoothing 是 0.85 的結果





    Testing 結果 (原 TD3)

```
~/Documents/RL-Lab4
python main.py
=============================================
Evaluating...
/home/larrychen1120/anaconda3/envs/RL/lib/python3.9/site-packag
deprecated alias for `np.bool_`.  (Deprecated NumPy 1.24)
  if not isinstance(terminated, (bool, np.bool8)):
Episode: 1      Length: 999     Total reward: 819.19
Episode: 2      Length: 473     Total reward: 411.67
Episode: 3      Length: 203     Total reward: 186.16
Episode: 4      Length: 368     Total reward: 400.36
Episode: 5      Length: 259     Total reward: 304.68
Episode: 6      Length: 999     Total reward: 818.03
Episode: 7      Length: 270     Total reward: 254.05
Episode: 8      Length: 429     Total reward: 484.78
Episode: 9      Length: 999     Total reward: 791.47
Episode: 10     Length: 257     Total reward: 267.06
average score: 473.7454413069865
=============================================
```

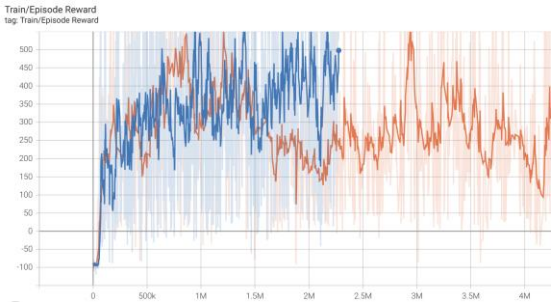
    Testing 結果 (改 reward)

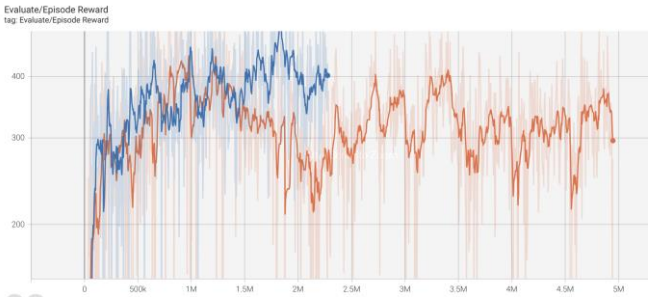
```
(RL) larrychen1120@adsl-3090:~/hdd0/HW4$ python main.py
============================================
Evaluating...
/home/larrychen1120/hdd0/miniconda3/envs/RL/lib/python3.9/
: `np.bool8` is a deprecated alias for `np.bool_`.  (Depr
  if not isinstance(terminated, (bool, np.bool8)):
Episode: 1      Length: 999     Total reward: 893.77
Episode: 2      Length: 779     Total reward: 922.00
Episode: 3      Length: 999     Total reward: 889.62
Episode: 4      Length: 999     Total reward: 886.25
Episode: 5      Length: 730     Total reward: 926.90
Episode: 6      Length: 693     Total reward: 930.60
Episode: 7      Length: 999     Total reward: 889.32
Episode: 8      Length: 999     Total reward: 868.55
Episode: 9      Length: 688     Total reward: 931.10
Episode: 10     Length: 714     Total reward: 928.50
average score: 906.6620445635979
============================================
```

- **Bonus**
  - **Screenshot of Tensorboard training curve and compare the performance of using twin Q-networks and single Q-networks in TD3, and explain**

訓練一個 Q network 容易有估計過高的問題，因為資料上的一些誤差而訓練上會產生偏差。而 TD3 則是使用兩個 Q-networks 來分別計算 Q 值，並取兩個 Q 值的最小值當作學習的目標，使用兩者的最小值可以有效地避免高估 value 的問題，從下方的 curve 也可以看出使用兩個 Q-networks 的表現較好也較穩定。

因為 TD3 一開始跑錯了所以 t 重跑的長度沒有單個 network 長，但還是可以看出使用兩個 Q-networks 效果好而且穩定。(藍色是 TD3 with twin-Q)





  - **Screenshot of Tensorboard training curve and compare the impact of enabling and disabling target policy smoothing in TD3, and explain**

Target policy smoothing 是在 policy 的輸出值上再加入一個小小的 noise 擾

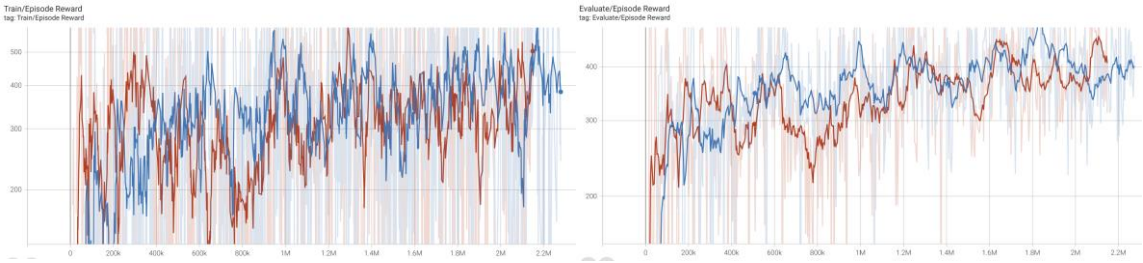
動，透過這樣的方式，action 的值會有些許的擾動，可以幫助模型有機會去探索到不一樣的組合，進而幫助模型的 Q-networks 可以學習到更多種的組合，幫助 Q-networks 的訓練更穩定。

由實驗結果可以看到 training 上兩者的差異不大，但是實際進行 evaluate 時會發現進行 policy smooth 的結果(橘色)比沒有 smooth 的表現較好。因為加入 noise 讓訓練的時候有一點擾動可以幫助模型去考慮到 action 擾動的影響，在實際遊玩的時候更加穩定。

- **Screenshot of Tensorboard training curve and compare the impact of delayed update steps and compare the results, and explain**

TD3 額外使用了 delayed update steps 來避免頻繁更新 policy，實驗中我是使用每 2 steps 才更新 policy network，而每個 step 都會更新 critic networks (Q-networks)。因為 policy 的學習需要由 Q-network 來引導，所以讓 Q-network 多學習可以保證 Q-network 先學到較好的 critic 來引導 policy network 的學習，可以增加學習的穩定性。

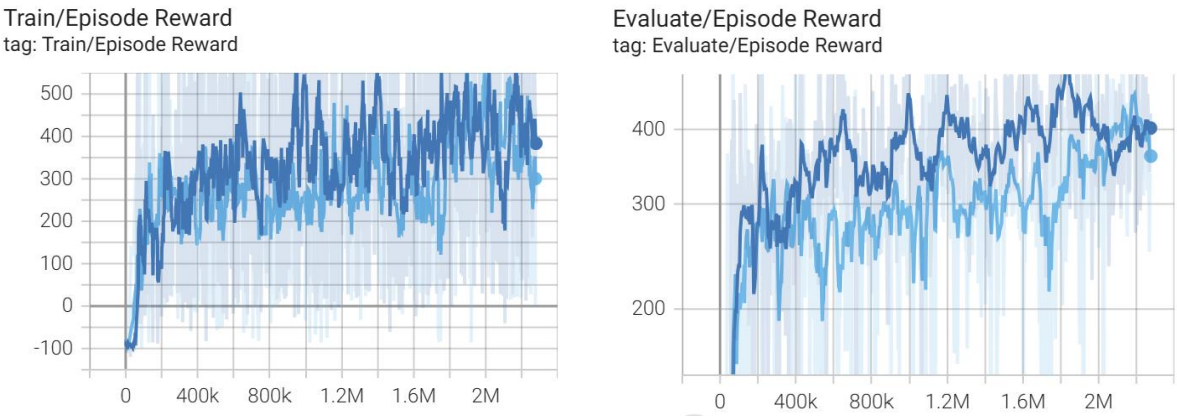
有沒有 delay 的影響，從實驗結果看不太出來，但是仍可以發現有 delay 的藍色線還是偏向在紅色線的上方，而且震盪的幅度相對小一點。



■ **Screenshot of Tensorboard training curve and compare the effects of adding different levels of action noise (exploration noise) in TD3, and explain**

這次實驗主要比較使用 Gaussian Noise 和 Ornstein-Uhlenbeck Noise，這兩個 Noise 的差異主要在 Gaussian Noise 每一次 sample 都是獨立的，所有它的變動性比較大，而 OU Noise 是跟時間相關的 Noise，會將擾動程度漸漸變小，進而讓模型前期可以多探索但後期漸漸穩定的 Noise 可以幫助模型後期的收斂，相較於 Gaussian Noise 訓練可以更穩定。

從實驗結果可發現，使用 Gaussian Noise (淡藍色)的 performance 明顯比原始的 TD3 差，原因就是 Gaussian Noise 浮動較大容易讓模型訓練不穩定。



■ **Screenshot of Tensorboard training curve and compare your reward function with the original one and explain why your reward function works better**

我針對車子開到草叢中設計了特別的 reward，用 reward 來懲罰車子開在草地上的狀況，透過這樣的方式車子可以更快學會開在道路上，而不用嘗試過多不必要的路徑。具體的 reward 設計為將**草地每個 pixel 都給-0.01** 的分

數，可以發現前期的 reward 會負到 4 千多分，也就是前期模型就是會開在草地上，但是當車子保持在賽道時會發現他的分數相對高，而鼓勵其保留在賽道上，由 curve 就可以看到：更改 reward(粉色)可以達到更高分而且非常快就可以達到很好的結果。