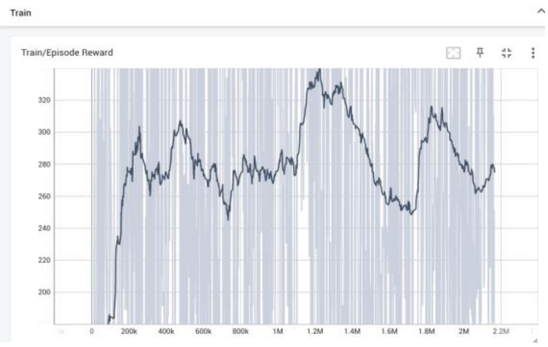
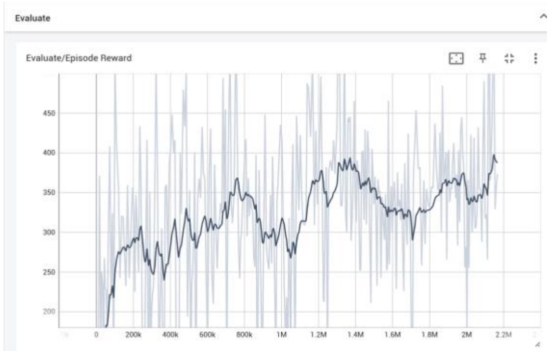
RL lab4

Name: 陳以瑄 Student ID: 109705001

- **Experimental Results**

  Screenshot of Tensorboard training curve





  Screenshot of testing results on TD3 (with the old reward)

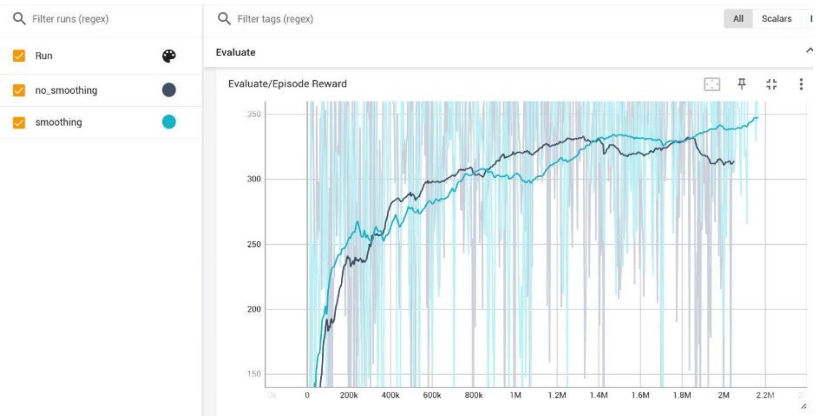
```
Evaluating...
Episode: 1      Length: 154     Total reward: 152.47
Episode: 2      Length: 331     Total reward: 385.72
Episode: 3      Length: 999     Total reward: 885.56
Episode: 4      Length: 292     Total reward: 325.77
Episode: 5      Length: 299     Total reward: 323.15
Episode: 6      Length: 144     Total reward: 117.64
Episode: 7      Length: 493     Total reward: 591.45
Episode: 8      Length: 342     Total reward: 386.23
Episode: 9      Length: 160     Total reward: 131.44
Episode: 10     Length: 999     Total reward: 871.33
average score: 417.0751429991862
```

  Screenshot of testing results on TD3 (with the new reward)

```
Episode: 1      Length: 999     Total reward: 890.07
Episode: 2      Length: 984     Total reward: 901.50
Episode: 3      Length: 827     Total reward: 917.20
Episode: 4      Length: 999     Total reward: 893.44
Episode: 5      Length: 945     Total reward: 905.40
average score: 901.5217696232643
```

- **Bonus2. Target Policy Smoothing**

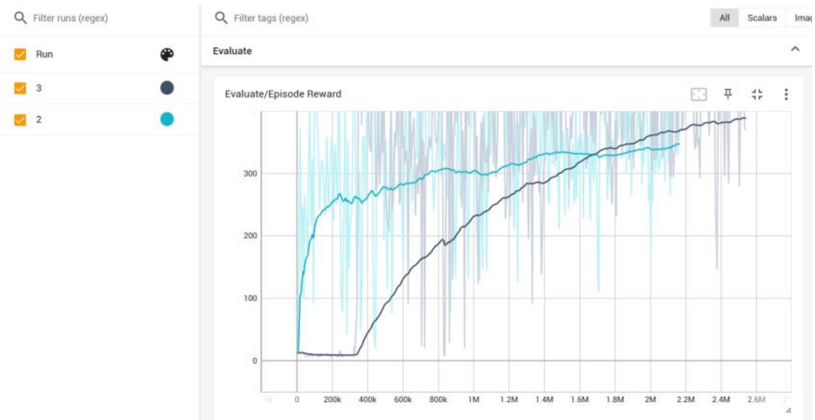
  Screenshot of Tensorboard training curve



  The plot above compares the results of the model with smoothing and without smoothing. As the figure shows, although they have only slight differences, we can still observe that the model with smoothing performs better in the end. This is because smoothing makes the learning process less sensitive to small changes in the Q-values, resulting in greater stability.

- **Bonus3. Delayed Policy Update Mechanism**

  Screenshot of Tensorboard training curve



I compared the model with a 'update_freq' hyperparameter set to 3 to the one with a 'update_freq' of 2. The results show that the model with a higher 'update_freq,' which updates less frequently, learns more slowly in the beginning but performs better after 1.8 million updates. This is because delay update mechanism can stabilize the learning, so that it can get a better performance.

- **Bonus5. Reward Function Design**

The following is my reward function:

```
        #if terminates:
        #   reward = -150
        # more road better
        reward += (float(road_pixel_count) - float(grass_pixel_count))/20.0

        if road_pixel_count < 10:
            terminates = True
            reward = -100
```

The idea is that if the proportion of the road surface in the visual scene is higher, it indicates that the race car is less deviated from the track. Therefore, it should receive a higher reward.

Screenshot of Tensorboard training curve

As the figure shows, the model with the new reward obtains higher rewards more quickly. Although its later improvement rate shows some stagnation, the overall performance still surpasses that of the old reward function. This means that the idea of encouraging the car to stay on the road can indeed help the model perform better.