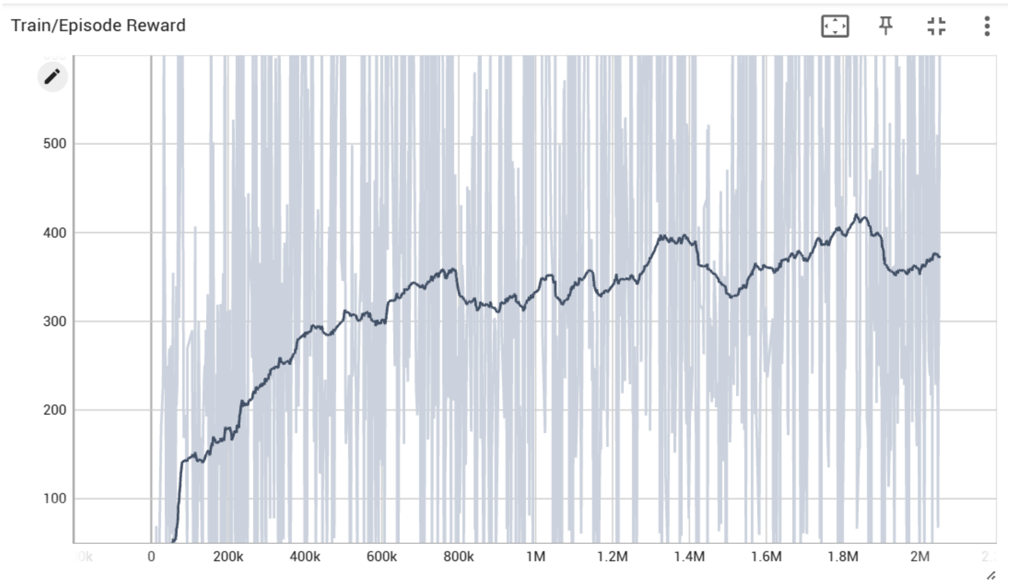
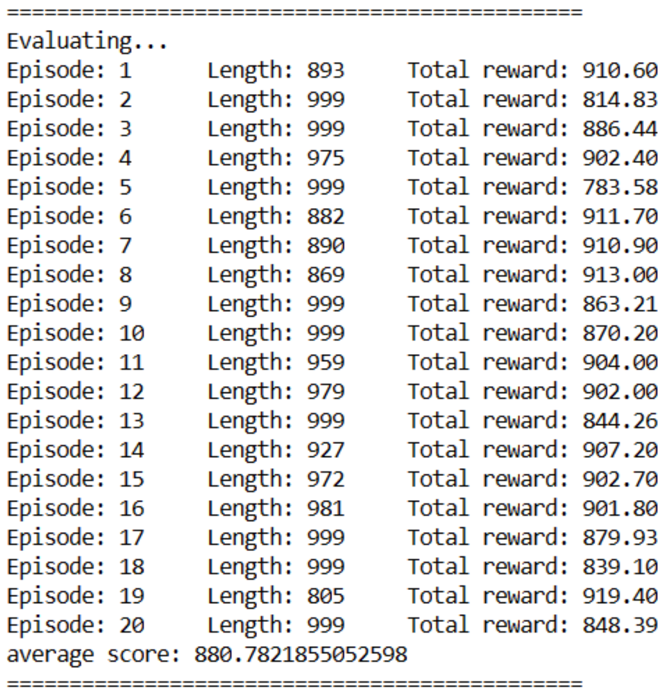
RL Lab4 TD3

Experimental Result

Training curve



Testing result (model trained on new reward)



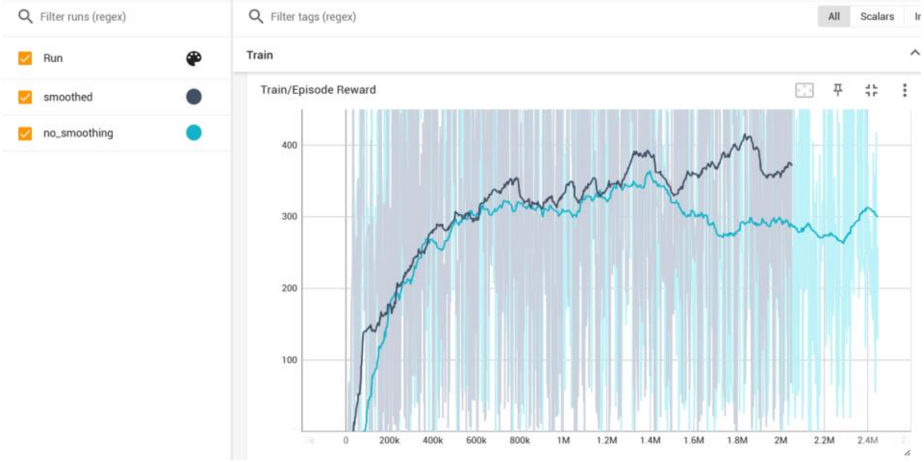
```
==============================================
Evaluating...
Episode: 1      Length: 893     Total reward: 910.60
Episode: 2      Length: 999     Total reward: 814.83
Episode: 3      Length: 999     Total reward: 886.44
Episode: 4      Length: 975     Total reward: 902.40
Episode: 5      Length: 999     Total reward: 783.58
Episode: 6      Length: 882     Total reward: 911.70
Episode: 7      Length: 890     Total reward: 910.90
Episode: 8      Length: 869     Total reward: 913.00
Episode: 9      Length: 999     Total reward: 863.21
Episode: 10     Length: 999     Total reward: 870.20
Episode: 11     Length: 959     Total reward: 904.00
Episode: 12     Length: 979     Total reward: 902.00
Episode: 13     Length: 999     Total reward: 844.26
Episode: 14     Length: 927     Total reward: 907.20
Episode: 15     Length: 972     Total reward: 902.70
Episode: 16     Length: 981     Total reward: 901.80
Episode: 17     Length: 999     Total reward: 879.93
Episode: 18     Length: 999     Total reward: 839.10
Episode: 19     Length: 805     Total reward: 919.40
Episode: 20     Length: 999     Total reward: 848.39
average score: 880.7821855052598
==============================================
```

Experimental Results and Discussion of bonus

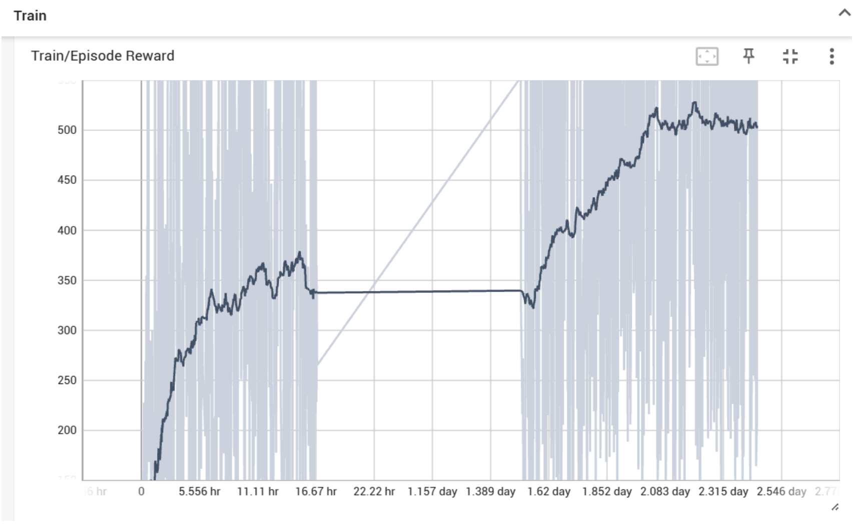
1. Twin network X

2. Target policy smoothing

Training curve

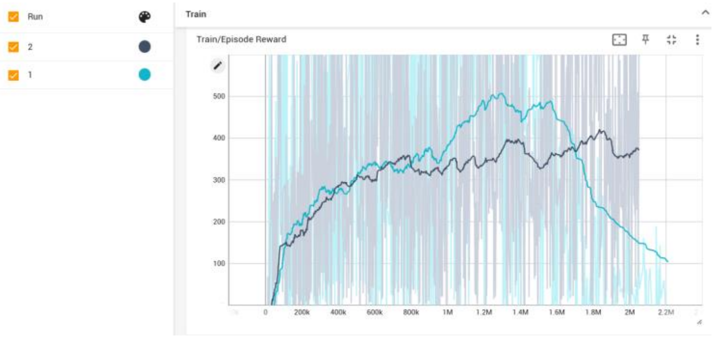


I set sigma = 0 to disable target policy smoothing.
At the beginning of training, there exist no huge difference of the 2 models. However, the model without smoothing seems to converge to 300 rewards. I trained further 2M steps for smoothed model with the model of the prior training result and sigma = 0.1, and the result showed that it achieves an average of 500 rewards, which is 200 rewards higher than that of the model without smoothing. This shows the result of target policy smoothing improves the performance. (I used Relative in tensorboard so there's a gap between the 2 training.)



3. update frequency

Training curve

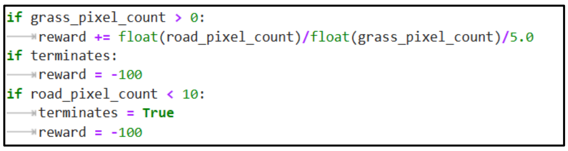


The result shows that with update frequency = 1(disable delayed update), the model learns faster but is relatively unstable (the reward dropped significantly from 1.5M to 2.2M step).

4. Action noise injection X

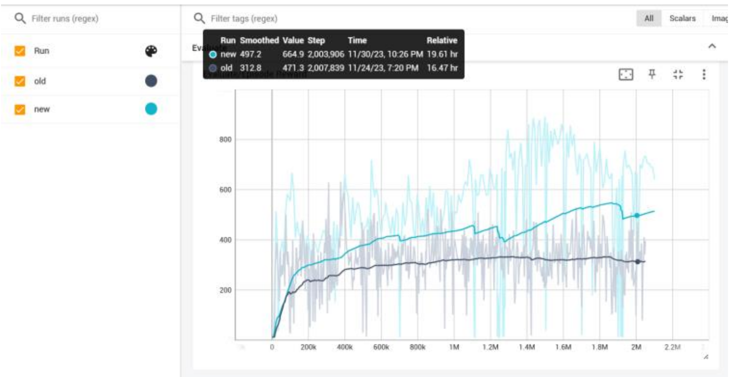
5. New reward

New reward:



```
if grass_pixel_count > 0:
    reward += float(road_pixel_count)/float(grass_pixel_count)/5.0
if terminates:
    reward = -100
if road_pixel_count < 10:
    terminates = True
    reward = -100
```

Evaluation curve



	Run	Smoothed	Value	Step	Time	Relative
●	new	497.2	664.9	2,003,906	11/30/23, 10:26 PM	19.61 hr
●	old	312.8	471.3	2,007,839	11/24/23, 7:20 PM	16.47 hr

The original reward -100 when road_pixel_count < 10. The third and fourth line actually have little effect (I thought the road_pixel_count is for the whole image), but I put it there because that is the original reward I used when training. In addition, I add a reward to encourage the agent to walk in the road with an additional reward of road_pixel_count / grass_pixel_count / 5.