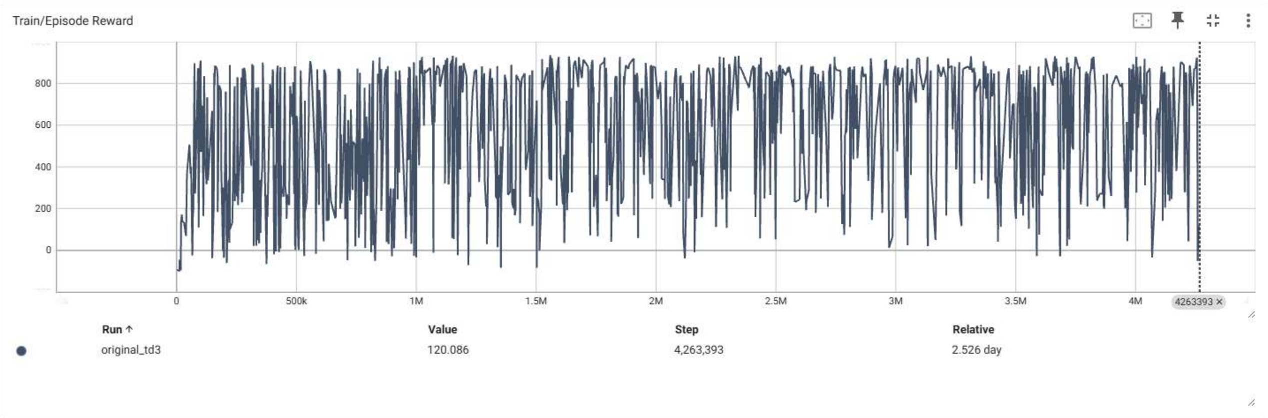
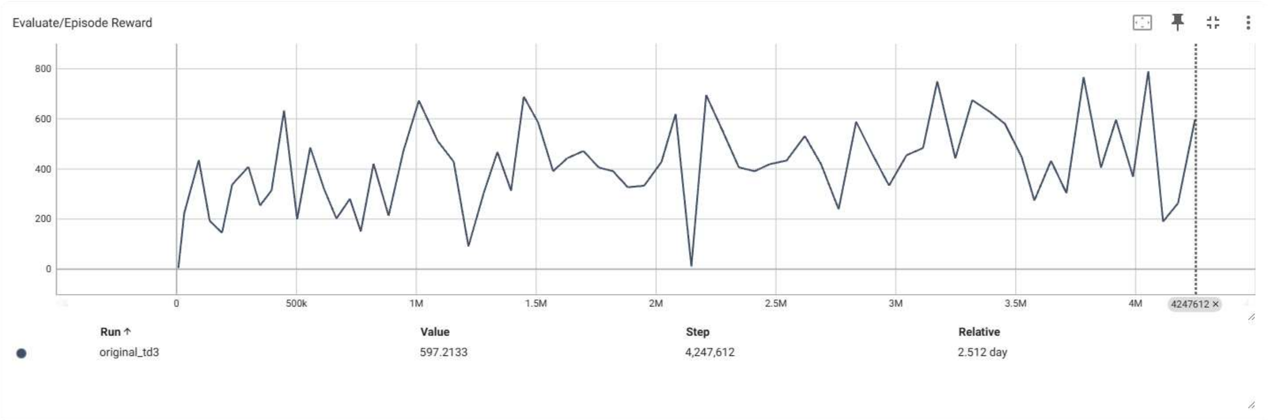
LAB4

Experimental Results

Training curve



Evaluate curve



Testing result

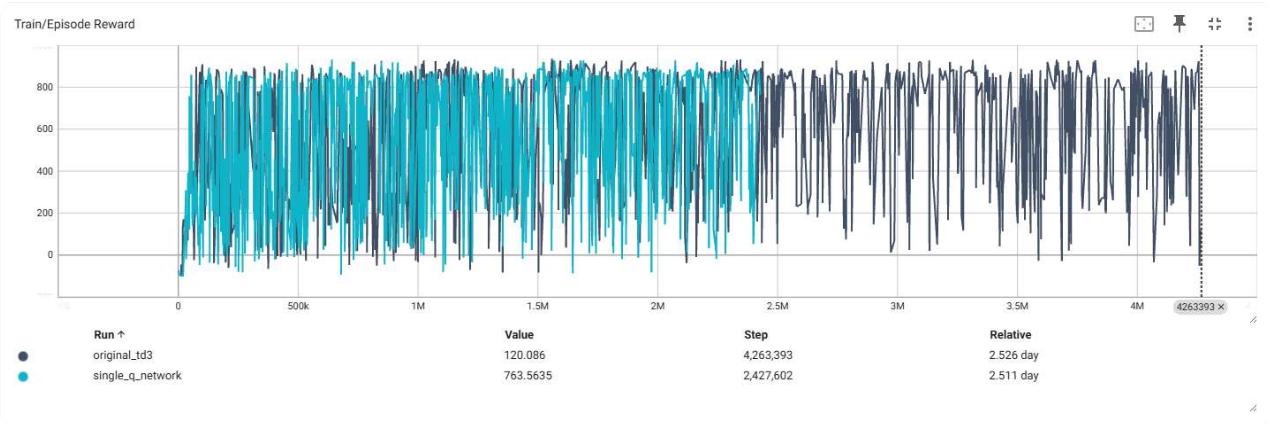
我最好的成果是 changing reward function。

```
==============================================
Evaluating...
C:\Users\a2320\miniconda3\envs\pytorch_env\lib\site-packages\gym\utils\passive_env_checker.py:233: Deprecat
ionWarning: `np.bool8` is a deprecated alias for `np.bool_`.  (Deprecated NumPy 1.24)
  if not isinstance(terminated, (bool, np.bool8)):
Episode: 1      Length: 969     Total reward: 883.58
Episode: 2      Length: 759     Total reward: 917.53
Episode: 3      Length: 657     Total reward: 926.54
Episode: 4      Length: 969     Total reward: 855.94
Episode: 5      Length: 748     Total reward: 918.30
Episode: 6      Length: 840     Total reward: 909.89
Episode: 7      Length: 743     Total reward: 918.68
Episode: 8      Length: 716     Total reward: 921.18
Episode: 9      Length: 722     Total reward: 920.73
Episode: 10     Length: 711     Total reward: 921.78
average score: 909.4155608567492
==============================================
```

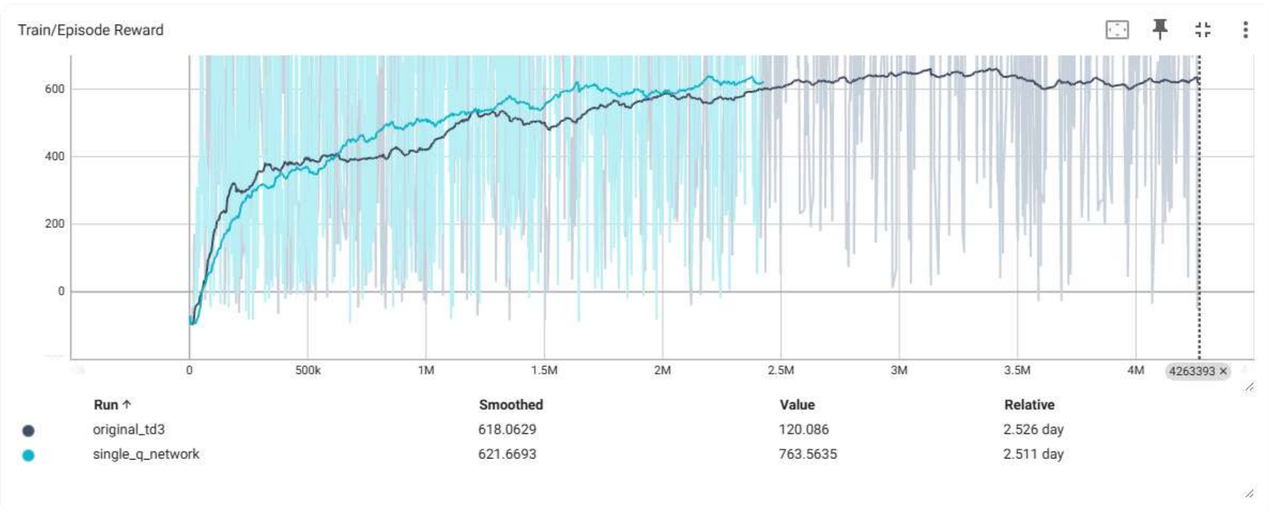
Bonus

single Q-networks

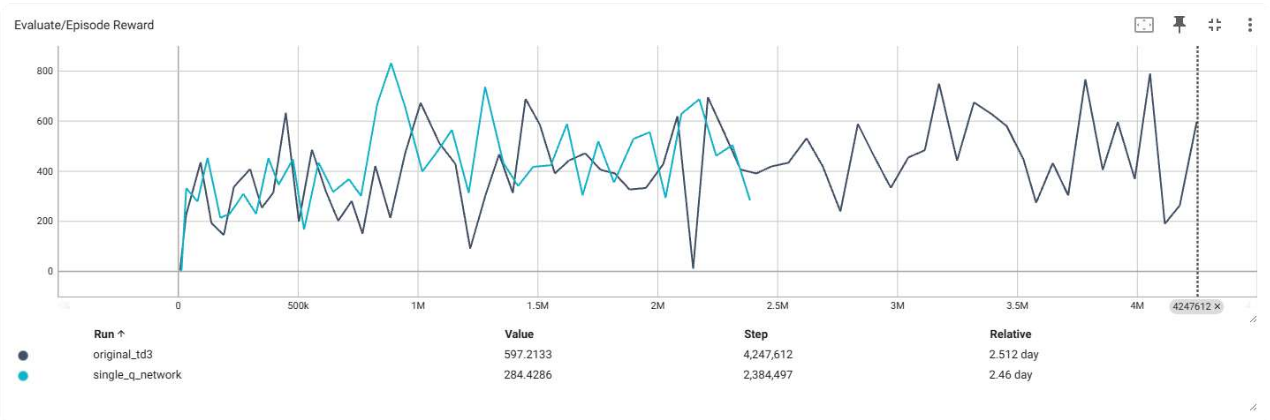
training curve



training curve (smoothed)



evaluate curve

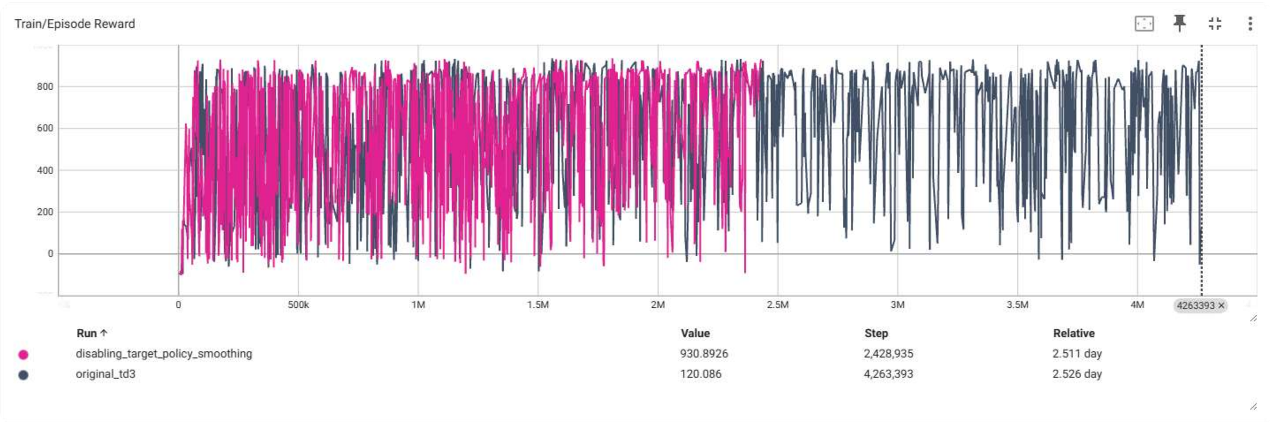


dicussion

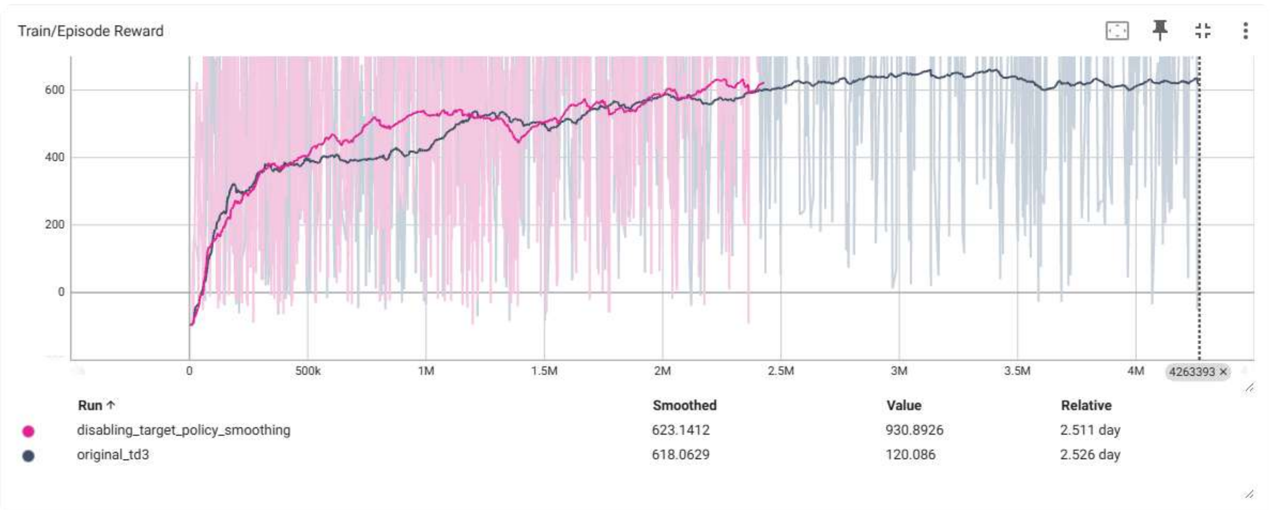
理論上 original td3 有 twin-q-networks 會更不容易 over-estimate，使得更新的 target 值更加準確，但我這邊訓練的結果看起來 single-q-network 的表現其實沒有差異很大，整體的訓練曲線沒有差到太多，只是在我每 100 個 training epoch 進行 evaluate 時，可能剛好抽到比較好的 model，或是 evaluete 時的賽道剛好是 model 有學習到如何應對的，所以 single q network 最好的結果比較好。

disabling target policy smoothing

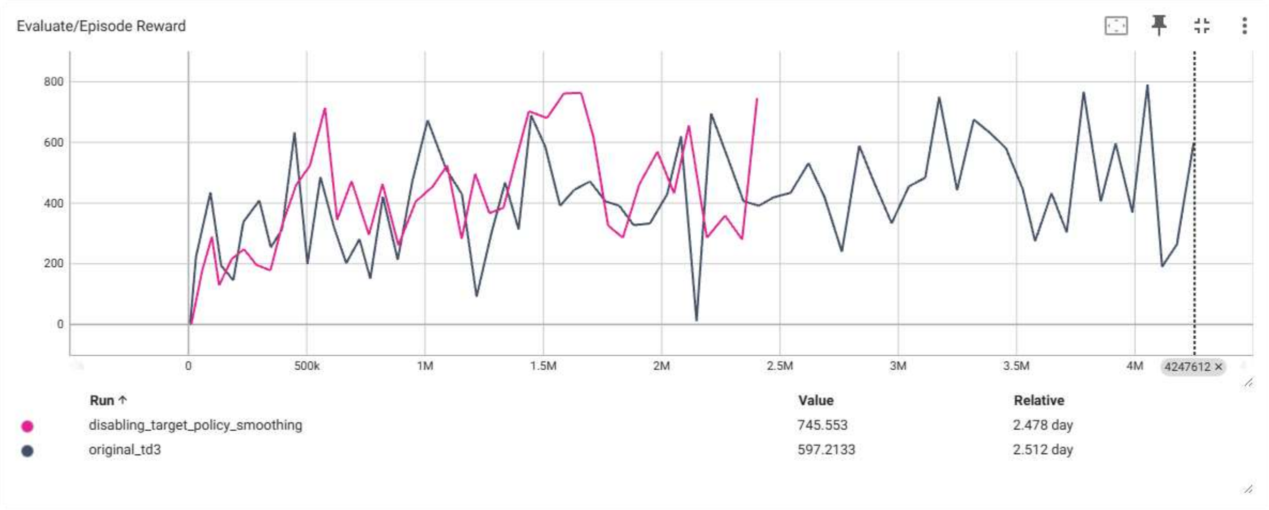
training curve



training curve (smoothed)



evaluate curve



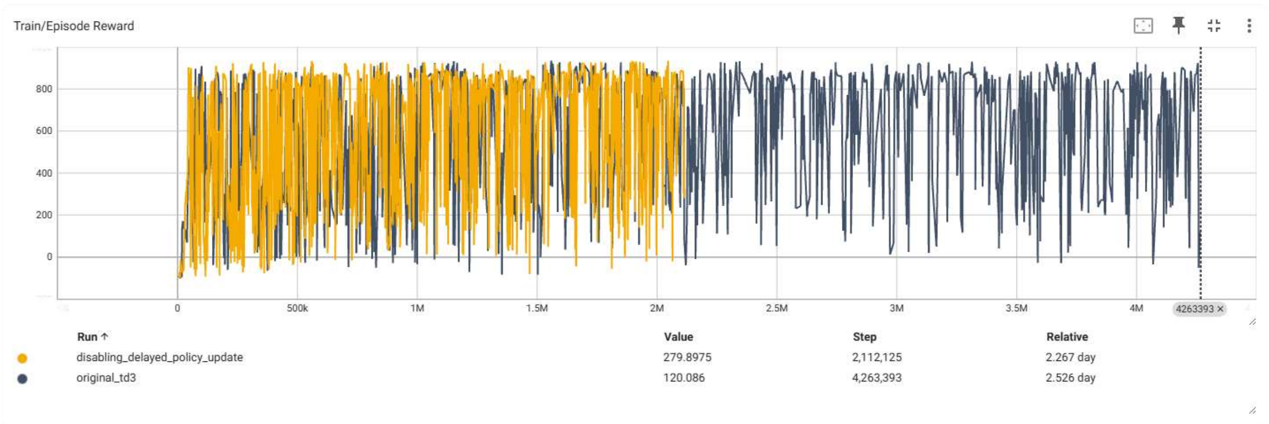
dicussion

理論上有 target smoothing 做 regularization 的 original td3 應該會更抗干擾，加上 noise 才進行更新，某種程度上確保了選擇的 best action 受到一點擾動還是會有不錯的表現，但其實在我這邊訓練曲線也沒有顯著差異，雖然 evaluate 的成果有比較好，但感覺還是誤差範圍內。

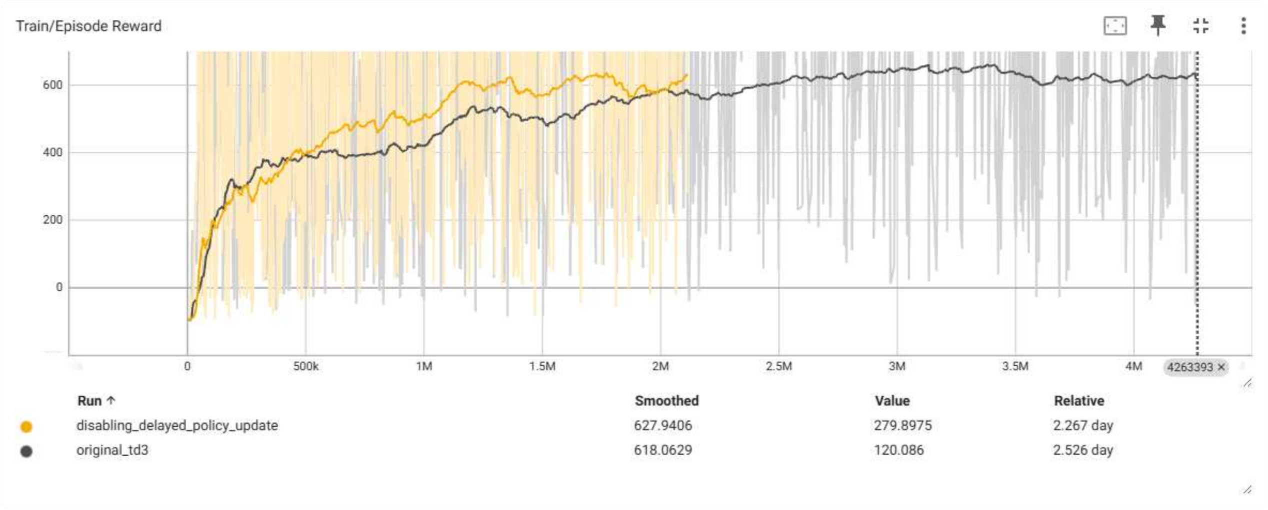
```
noise = torch.tensor(self.policy_noise * self.noise.generate(), dtype=torch.float32).to(self.device).clamp(-self.noise_clip, self.noise_clip)
a_next = (a_next + noise).clamp(self.min_action, self.max_action)
```

disabling delayed update steps

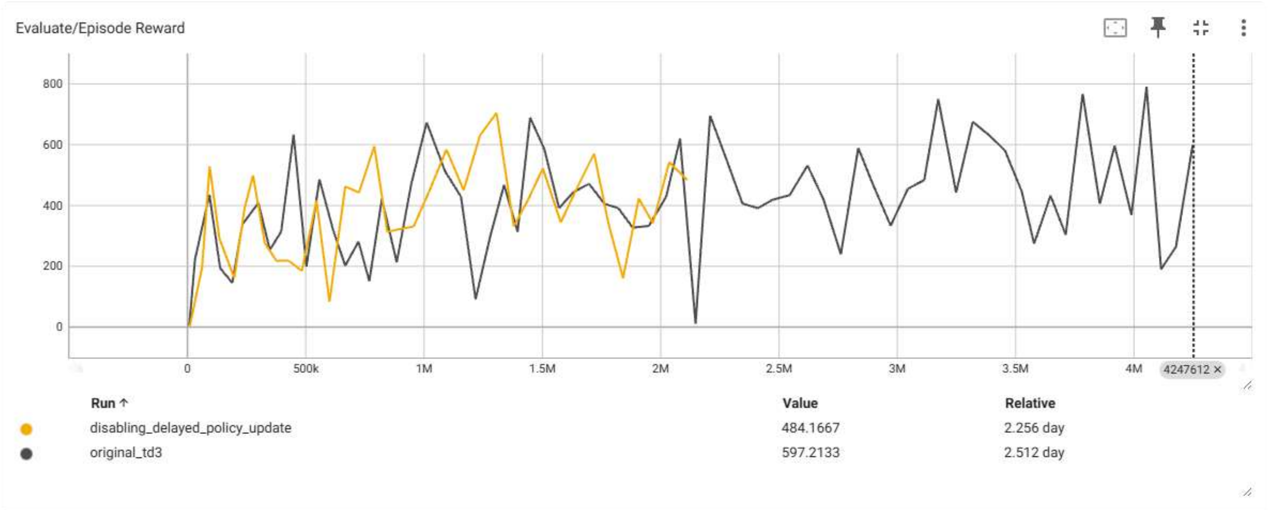
training curve



training curve (smoothed)



evaluate curve

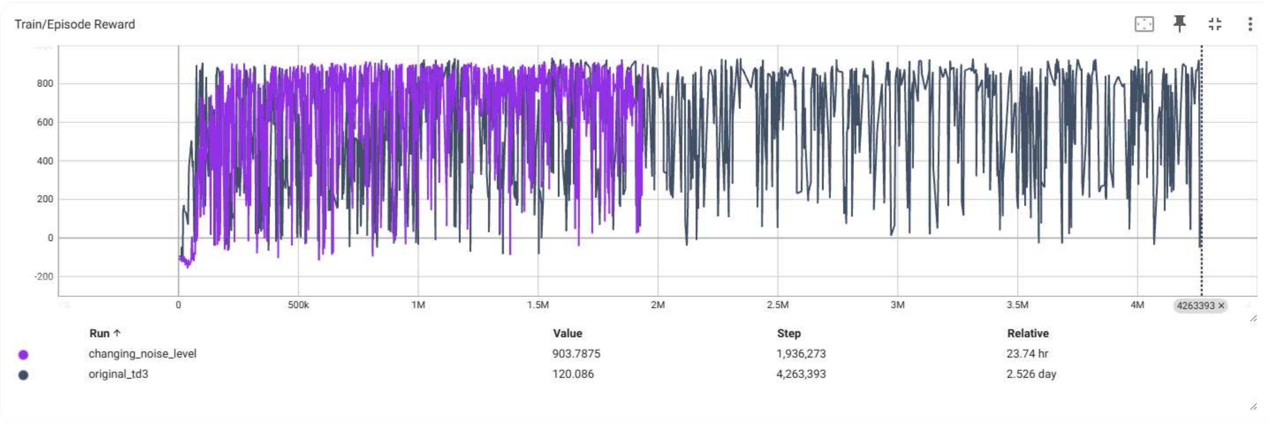


dicussion

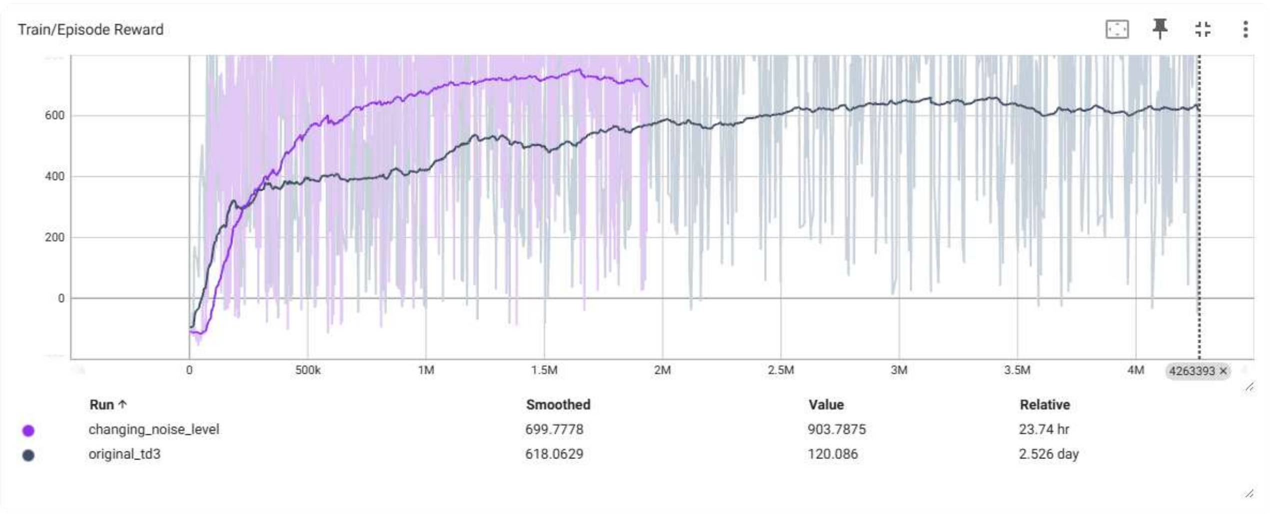
理論上讓 Actor 更新頻率小於 Critic 可以讓 Actor 不容易因為 Critic 震盪而練歪，但是我這邊 disable 好像也沒有練得比較震盪，甚至在 500 k 後的表現好像更穩定一點。

chaning different exploration noise level

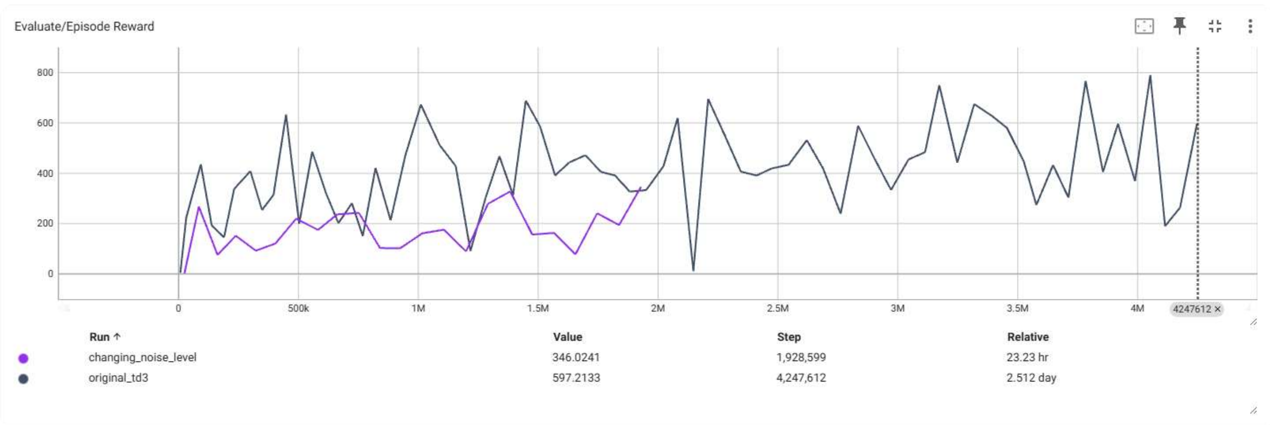
training curve



training curve (smoothed)



evaluate curve



dicussion

我這邊簡單的把 exploration noise 乘以三倍，他在 training 的表現看起來顯著比較好，但在 evaluate 的時候表現卻不太好，其實挺不合理的。我猜測 training 時 exploration noise 過大會讓他學習到抵抗很強的 noise 的能力，但同時 evaluete 時這個 noise 又不存在，導致他反而不會在沒有 noise 的環境下跑出好成績。

```
# exploration degree
sigma = max(0.1*(1-episode/self.total_episode), 0.01)
# sigma = 3 * max(0.1*(1-episode/self.total_episode), 0.01)
```

補充：我發現我的 total_episode 使用預設值沒有改動為 100000，所以 sigma 應該幾乎會是 3 * 0.1 = 0.3 上下(我大概都跑幾千個 episode 而已)，我在 evaluate 時也加上 0.3 的 noise 後，其實就跟 training 時跑得差不多好了，雖然看起來有點像是在強風中前進的車子，緩慢穩定的前進，有稍微異常就煞車重開的感覺。

```
action = self.decide_agent_actions(state, sigma=0.3)
```

```
=============================================
Evaluating...
C:\Users\a2320\miniconda3\envs\pytorch_env\lib\site-packages\gym\utils\passive_env_checker.py:233: DeprecationWarning: `np.bool8` is a deprecated alias for `np.bool_`.  (Deprecated NumPy 1.24)
  if not isinstance(terminated, (bool, np.bool8)):
Episode: 1      Length: 969     Total reward: 716.56
Episode: 2      Length: 969     Total reward: 778.42
Episode: 3      Length: 969     Total reward: 774.19
Episode: 4      Length: 969     Total reward: 658.10
Episode: 5      Length: 969     Total reward: 862.56
Episode: 6      Length: 969     Total reward: 637.22
Episode: 7      Length: 969     Total reward: 772.57
Episode: 8      Length: 969     Total reward: 776.33
Episode: 9      Length: 968     Total reward: 895.55
Episode: 10     Length: 969     Total reward: 627.59
average score: 749.908831642909
=============================================
```

changing reward funtion

這邊我設計了兩個版本，兩者差異在係數和有無到草地上即 terminate，首先我有先讓 part_image 裁的更置中一點，sample code 裁的有點偏左，然後根據路寬和車寬選擇了裁減的寬度。

```
part_image = obs[66-5:77+5, 47-6:49+6, :]
```

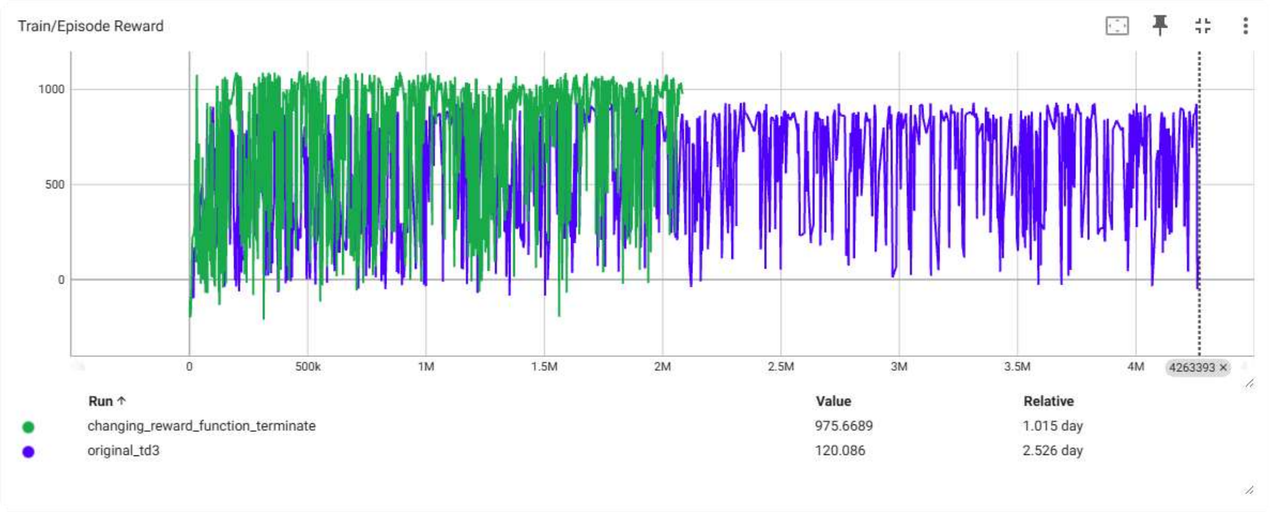
大致上的目標有兩點：

1. 我希望他開在盡量靠路中間，不要靠到路邊，所以我裁得更小一點，並且懲罰他開在路邊看到草會扣與 grass pixel 成正比的 reward。
2. 有時候他會漏掉一些格子，導致他跑第二圈時，在第一圈把大部分的格子都吃完後，他就沒有開在路上的 reward，只會隨著時間慢慢扣分，所以我也獎勵了與 road pixel 成正比的 reward，讓他開在路上可以抵銷時間的扣分，並且有一點點微微的加分。

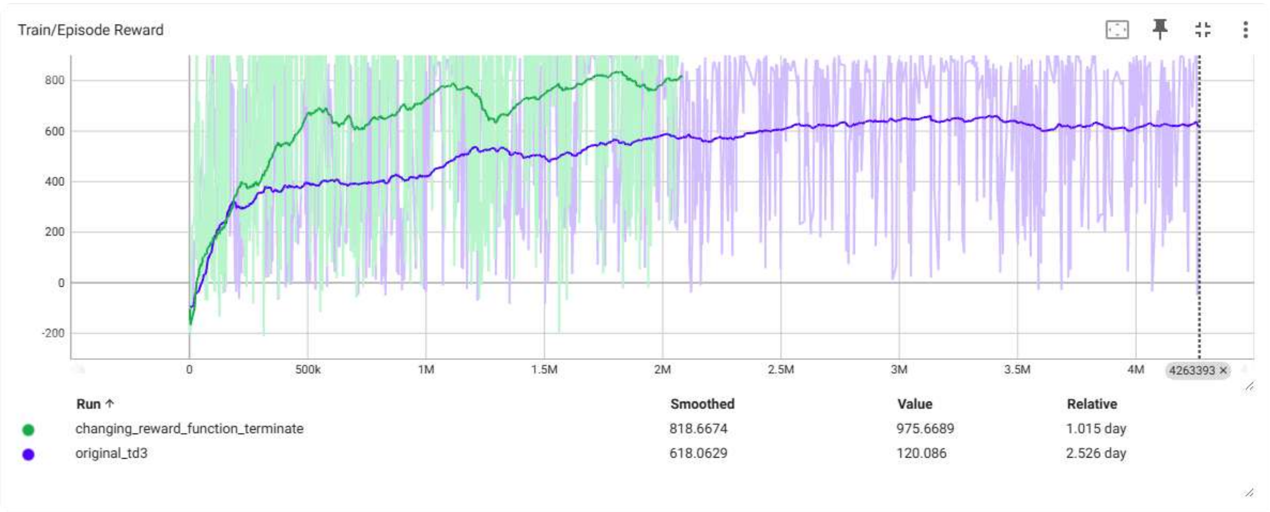
我發現他完全在路上大概會有 250 個 road pixel，完全在草地上大概會有 250 個 grass pixel，我希望他看到草地的懲罰是大的，並且比時間的自然扣分還大，所以初步就設了 0.02，讓他每個 frame 會扣 0~5 分。同時我不希望待在路面上給的獎勵高過前進吃格子給的獎勵，所以我設 0.001，讓他每個 frame 會加 0~0.2 分，

```
reward = reward - 0.02 * grass_pixel_count
reward = reward + 0.001 * road_pixel_count
```

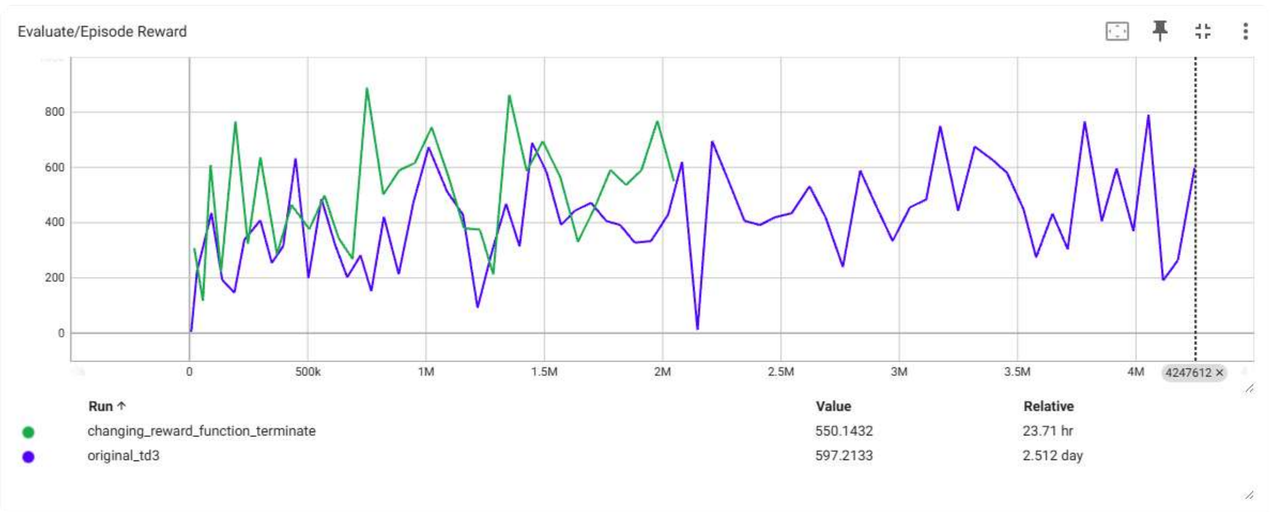
training curve



training curve (smoothed)



evaluate curve

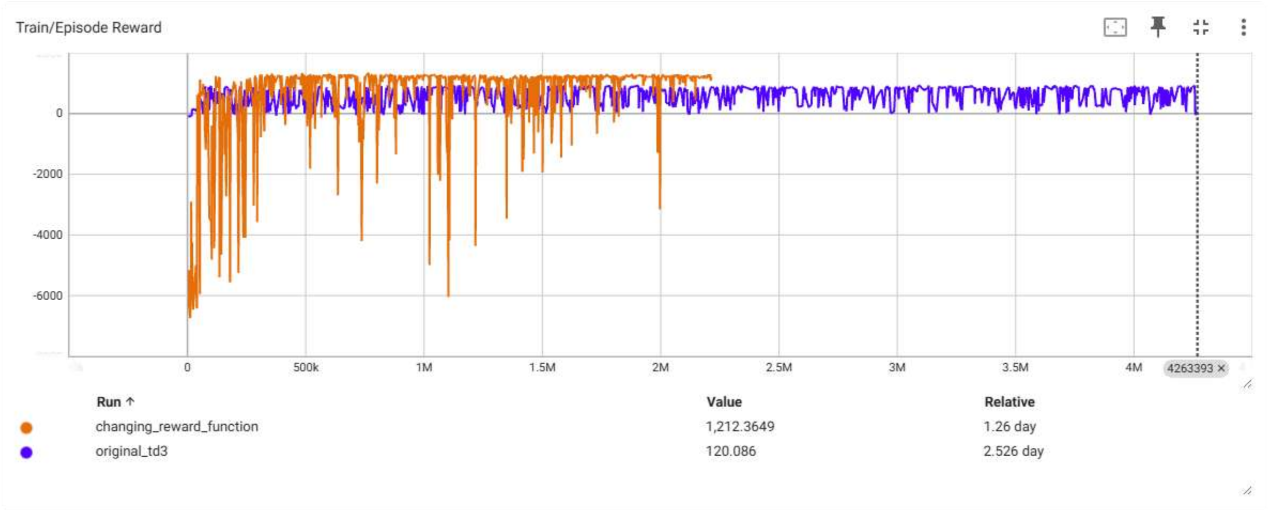


我觀察後覺得上個 reward function 練出來的結果還是有點波動，因為有時候他會不小心開到草地上，就直接 terminate 掉，他沒有學習到救車回路上的能力。

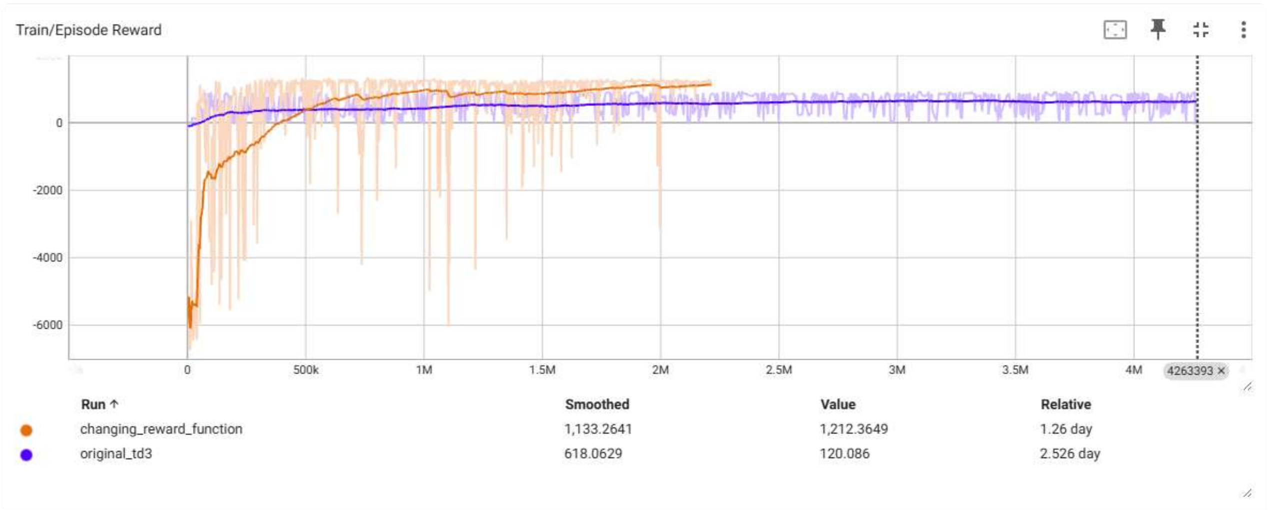
所以我接著嘗試了另一組參數如下，給予更多的草地懲罰以及路面獎勵，並且把到草地上會直接 terminate 這個條件拔掉，雖然訓練明顯花了更久，但是他在不小心衝出去道路外時，就會想辦法回到路上後繼續跑，相對穩定不少。

```
reward = reward - 0.03 * grass_pixel_count
reward = reward + 0.002 * road_pixel_count
```

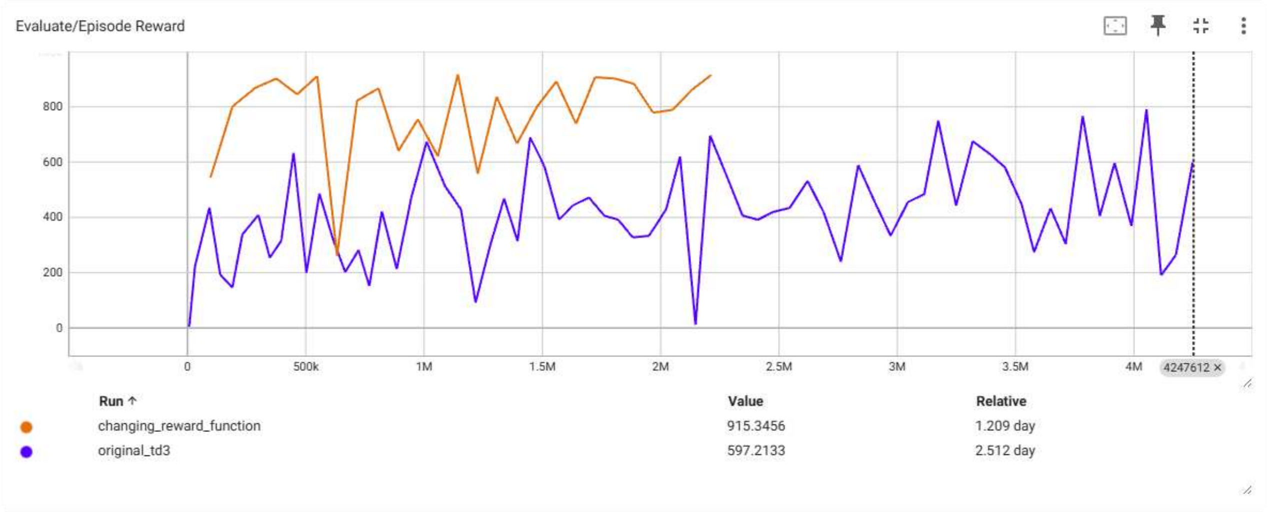
training curve



training curve (smoothed)



evaluate curve

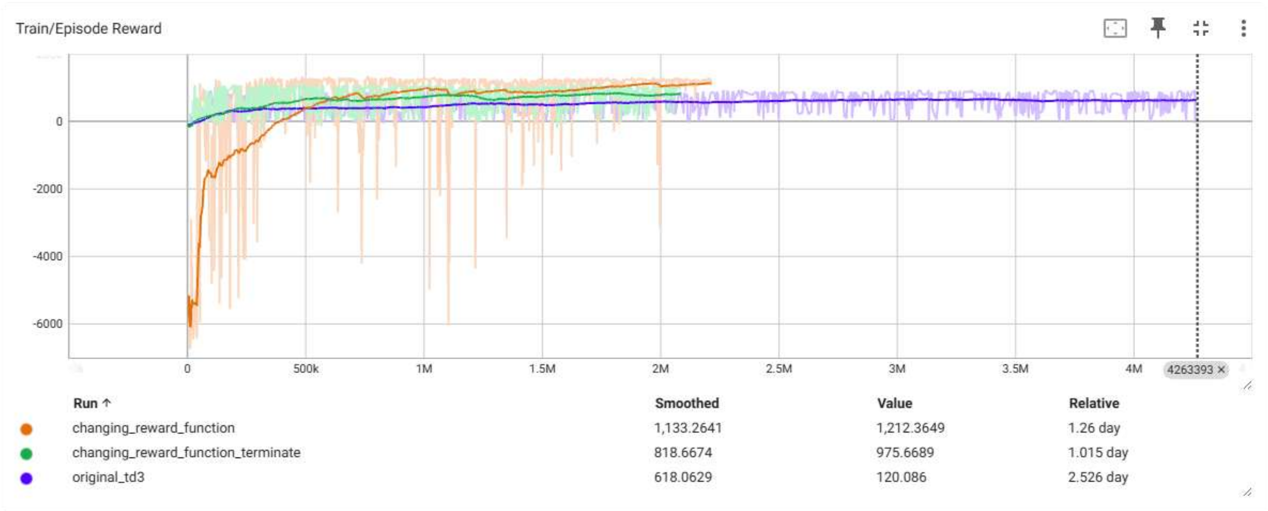


唯一缺點是他偶爾會在彎道處漏掉一兩格獎勵，導致第一圈結束後沒有直接 terminate 吃到完整的時間懲罰，接著第二圈又來不及吃到，所以最後分數就會在 880~900 之間。

compare

最後附上比較圖，可以在 evaluate 的圖上看出沒有 terminate 版本的明顯比其他兩個還要更厲害，但同時也練得比有 terminate 的版本更久一點。

Training curve (smoothed)



Evaluate curve (smoothed)