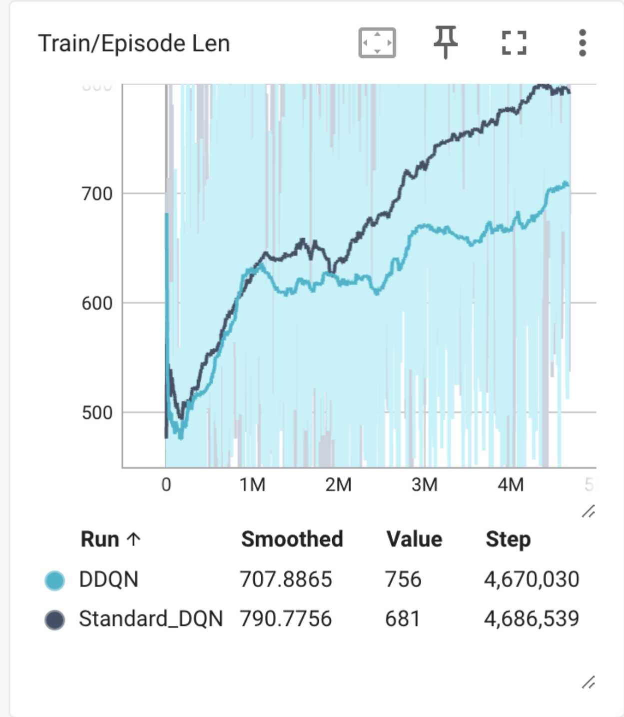
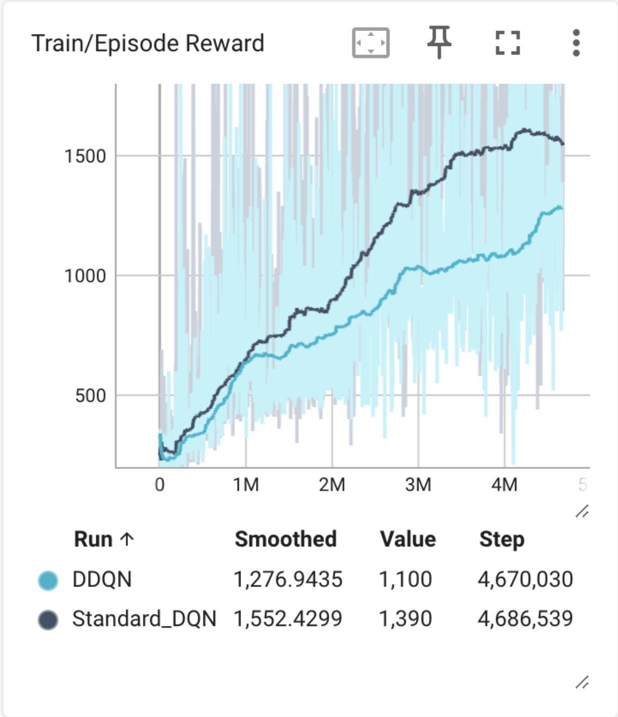
1. Screenshot of Tensorboard training curve and testing results on DQN.(Enduro)


```
A.L.E: Arcade Learning Environment (version 0.11.2+ecc1138)
[Powered by Stella]
Loading specified model: log/DQN/Enduro/model_4505963_543.pth...
==============================================
Evaluating...
episode 1 reward: 970.0
episode 2 reward: 776.0
episode 3 reward: 733.0
episode 4 reward: 490.0
episode 5 reward: 794.0
average score: 752.6
==============================================
```

2. Screenshot of Tensorboard training curve and testing results on DDQN, and discuss the difference between DQN and DDQN.






```
Loading specified model: log/DDQN/Pacman/model_5187918_1414.pth...
==============================================
Evaluating...
episode 1 reward: 2130.0
episode 2 reward: 1740.0
episode 3 reward: 1430.0
episode 4 reward: 920.0
episode 5 reward: 1190.0
average score: 1482.0
==============================================
```

$$Y_t^{Q} = r_{t+1} + \gamma \max_a Q(S_{t+1}, \boxed{a} | \theta^-)$$

$$Y_t^{DoubleQ} = r_{t+1} + \gamma Q\left(S_{t+1}, \boxed{\underset{a}{\operatorname{argmax}}\, Q(S_{t+1}, a|\theta)} | \theta^-\right)$$

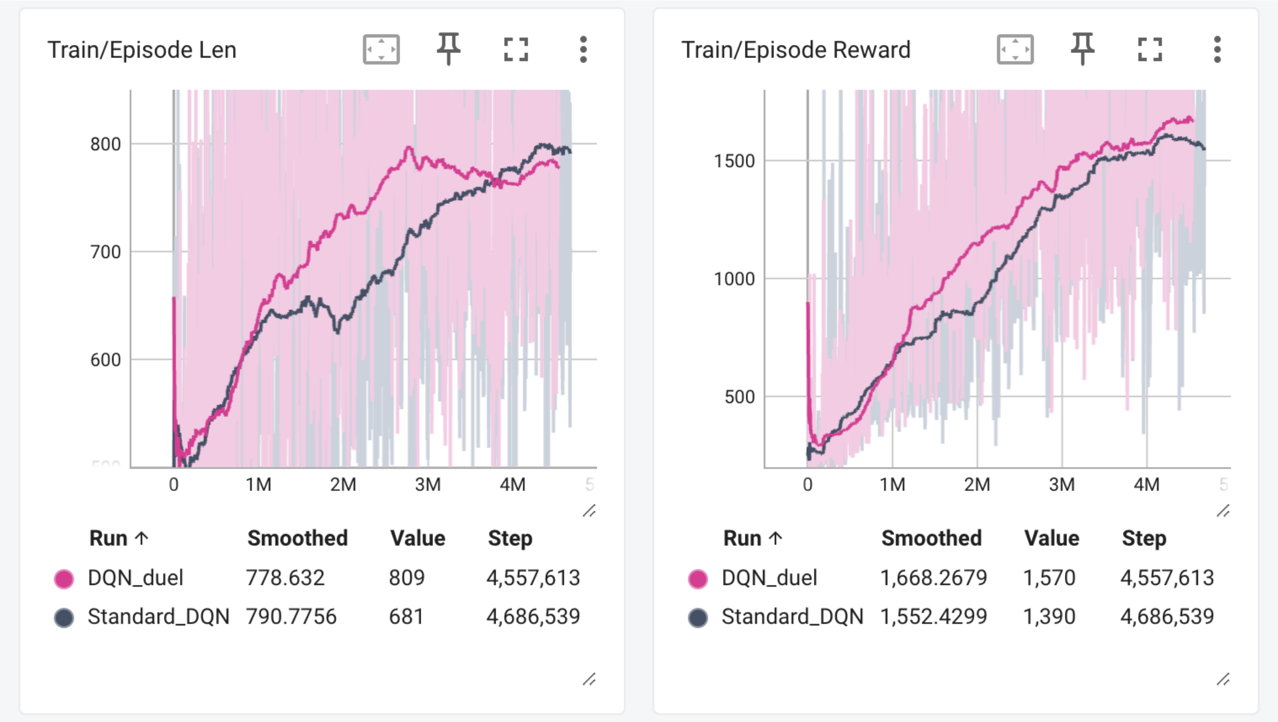
- 主要差別是DDQN用 behavior_net來選action，用target_net來評估該action的Q值，DQN只使用同一個target_net來選action與評估Q值。

在理論上，DDQN透過action selection（behavior network）與 evaluation（target network）分開，可以有效減少DQN中常見的Q-value overestimation的問題，因此通常能得到更好的結果。

但我目前DQN的表現反而優於DDQN。可能原因如下：

- DDQN的更新較保守，在訓練初期收斂速度較慢，後期可能DDQN會較好也說不定。
- Preprocessing 或 hyperparameter 設定差異（例如 frame skip、normalization 或 epsilon decay）造成輸入資料分佈與網路更新步調不一致。

3. Screenshot of Tensorboard training curve and testing results on Dueling DQN, and discuss the difference between DQN and Dueling DQN.




```
Loading specified model: log/DQN_duel/Pacman/model_5153273_1980.pth...
==============================================
Evaluating...
episode 1 reward: 1760.0
episode 2 reward: 2240.0
episode 3 reward: 2110.0
episode 4 reward: 1450.0
episode 5 reward: 1690.0
average score: 1850.0
==============================================
```

Dueling DQN 將 Q-value拆解為「狀態價值 (Value)」與「動作優勢 (Advantage)」：

- Value：估計當前狀態的整體價值V(s)，，代表該狀態本身的好壞。
- Advantage：估計在該狀態下各動作的相對優勢A(s,a)。

- $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$

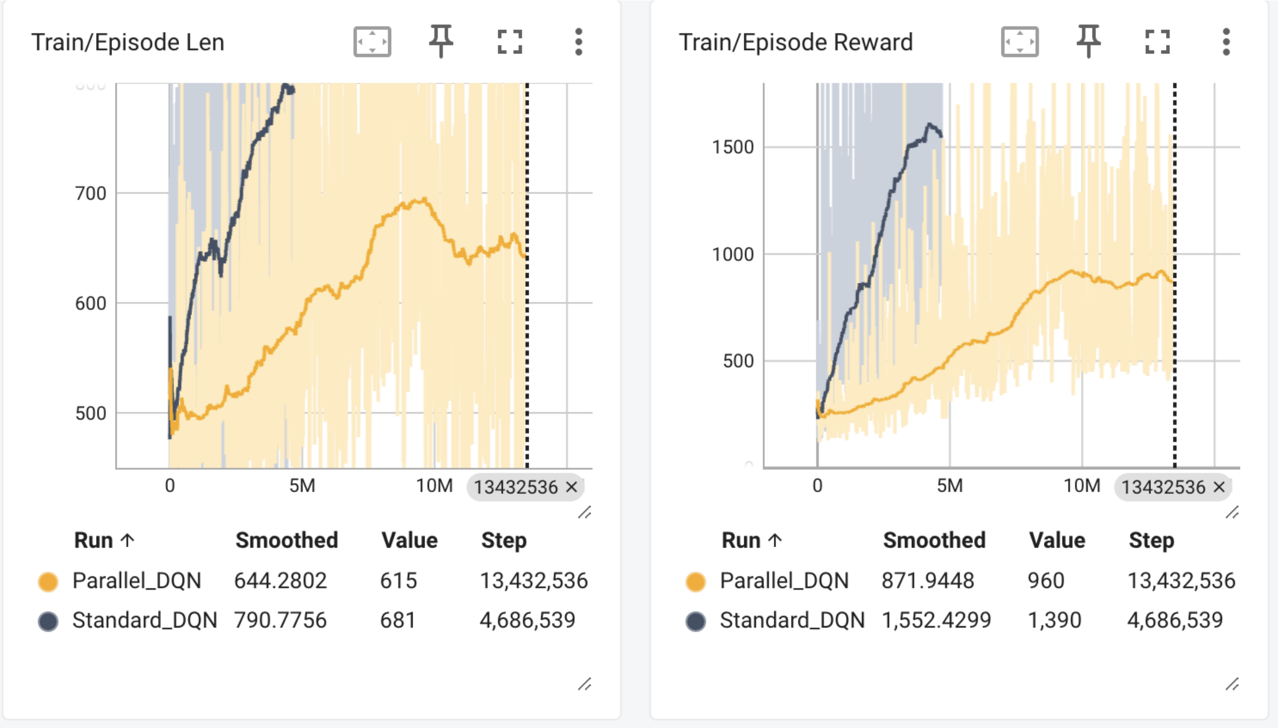
  $\rightarrow Q(s_t, a_t) = A(s_t, a_t) + V(s)$

- Constrain the value of A:

$$Q(s_t, a_t) = V(s) + (A(s_t, a_t) - \frac{1}{|A|} \sum_{a_t'} A(s_t, a_t'))$$

我的結果看來Dueling DQN也略好。

4. Screenshot of Tensorboard training curve and testing results on DQN with parallelized rollout, and discuss the difference between DQN and DQN with parallelized rollout.




```
[Powered by Stella]
Loading specified model: log/DQN_par/Pacman/model_15171504_1138.pth...
==============================================
Evaluating...
episode 1 reward: 770.0
episode 2 reward: 870.0
episode 3 reward: 1000.0
episode 4 reward: 1020.0
episode 5 reward: 810.0
average score: 894.0
==============================================
```

DQN with Parallelized Rollout 同時啟動多個環境（例如 4 或 8 個環境）並行收集經驗。每個環境獨立與 agent 互動，產生各自的 (state, action, reward, next_state) transition，再一起存入 replay buffer。