## Basic Performance (6%)

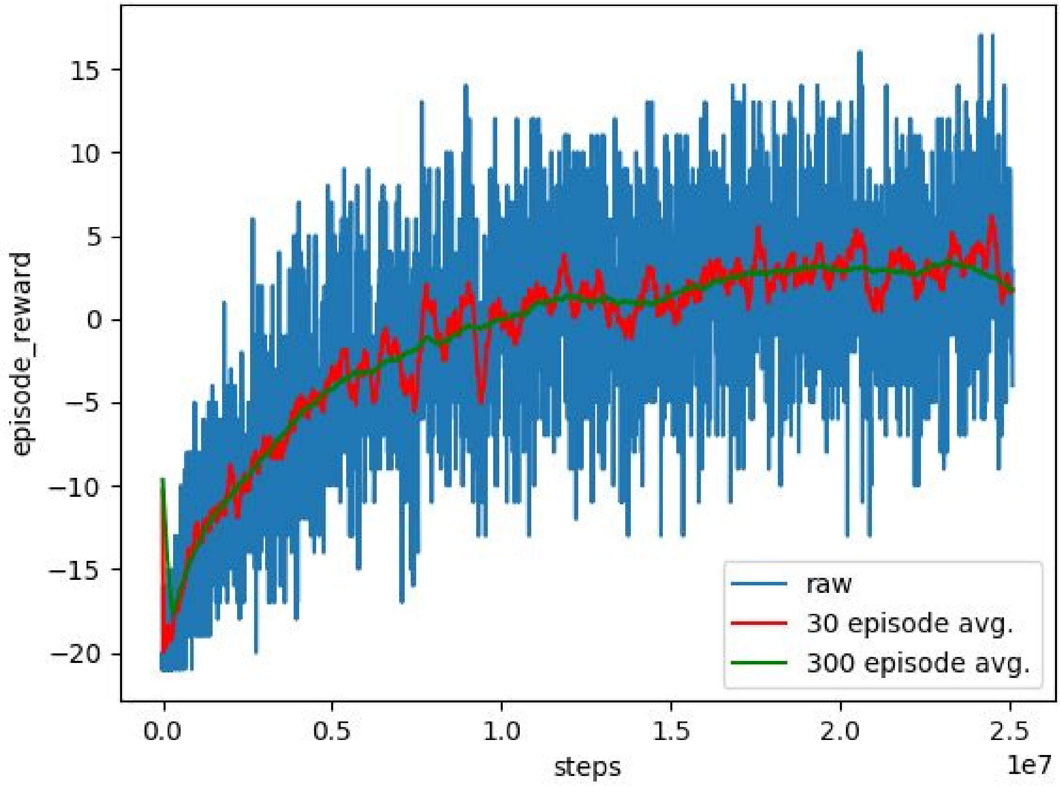
1. Model description (2%)：
   a. Policy Gradient model (1%)：
      主結構：
         Conv2D: 16, (8, 8), strides=(4, 4), relu
         Conv2D: 32, (4, 4), strides=(2, 2), relu
         Flatten
         Dense: 64, relu
         Dense: 2, softmax
      ( all kernel_initializer = 'lecun_normal' )

      input data：
         preporcessing: to gray scale, resize to (80, 80).
         state difference: obsevation - last_observation
      output action:
         only two classes, ( action=[2, 3])
      loss function:
         loss = sum(- log_action_prob * discount_reward)
      optimizer:
         Adam(lr=1e-4)

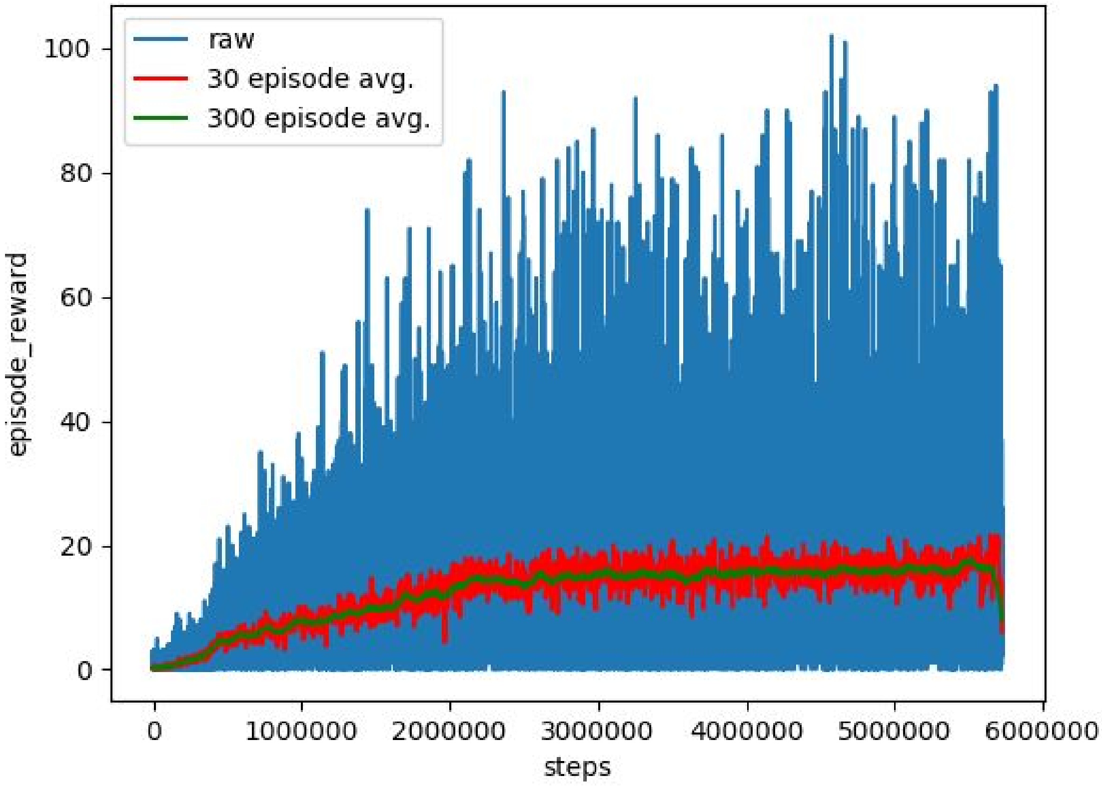
   b.
      主結構：
         Conv2D: 32, (8, 8), strides=(4, 4), relu
         Conv2D: 64, (4, 4), strides=(2, 2), relu
         Conv2D: 64, (3, 3), strides=(1, 1), relu
         Flatten
         Dense: 512, relu
         LeakyReLU: 0.1
         Dense: 4, linear

      input data：
         preporcessing: no further processing, size=(80, 80, 4).
      output action:
         four classes, ( action=[0, 1, 2, 3])
      loss function:
         target_q = reward +gamma * next_q * (1 - done)
         losses =max(square(target_q - current_q))
      optimizer:
         RMSprop(lr=1e-4, rho=0.99)

2. Policy Gradient on Pong (2%) :



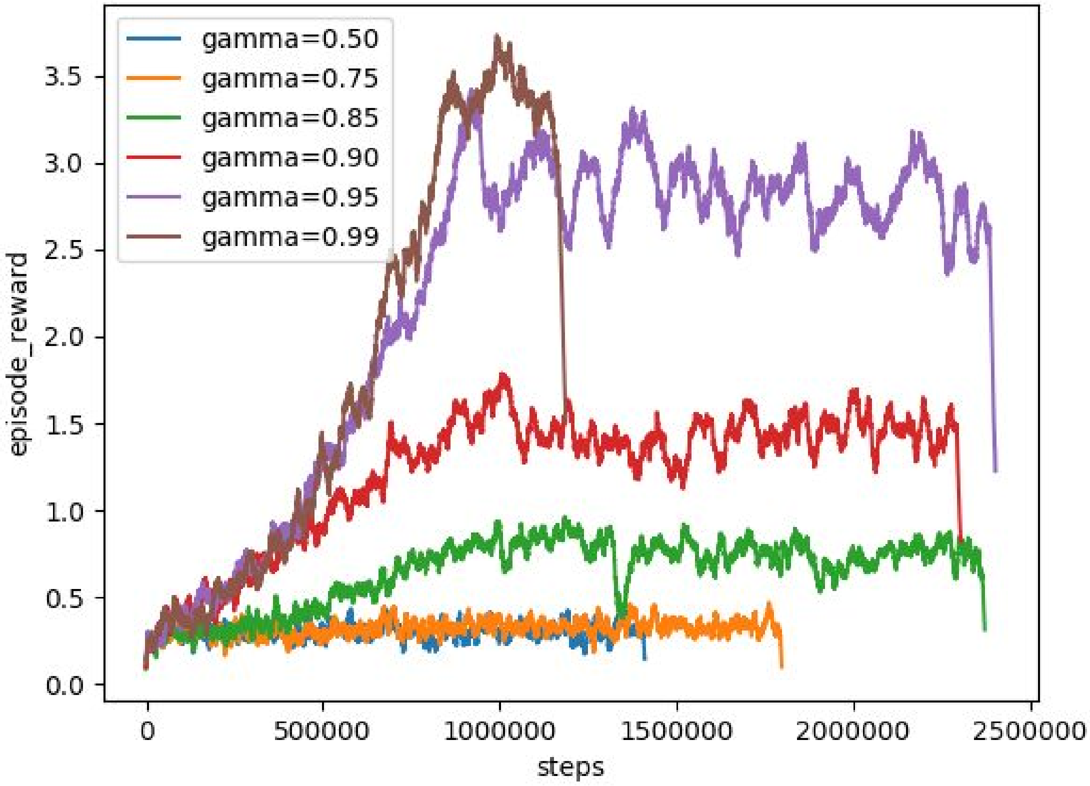

3. DQN on Breakout (2%) :



**Experimenting with DQN hyperparameters (4%)**

Hyperparameter: Gamma

1. Plot: ( moving average=300)




2. Why choosing this hyperparameter?
   Gamma代表對於未來reward的視野範圍，設定對的gamma值可以幫助model看到適當範圍的future reward，也可幫助Q function的數值收斂至接近實際值。
3. How it affects the results?
   以此遊戲來說，gamma高於0.9始得學習，gamma愈接近1，model可以計劃較長遠的動作。例如在方塊區鑿洞以將球傳至上方重複反彈得分；低於0.75時，撞到方塊的reward難以傳遞至擊球板準備接球的action，因此幾乎無法習得如何打球得分。

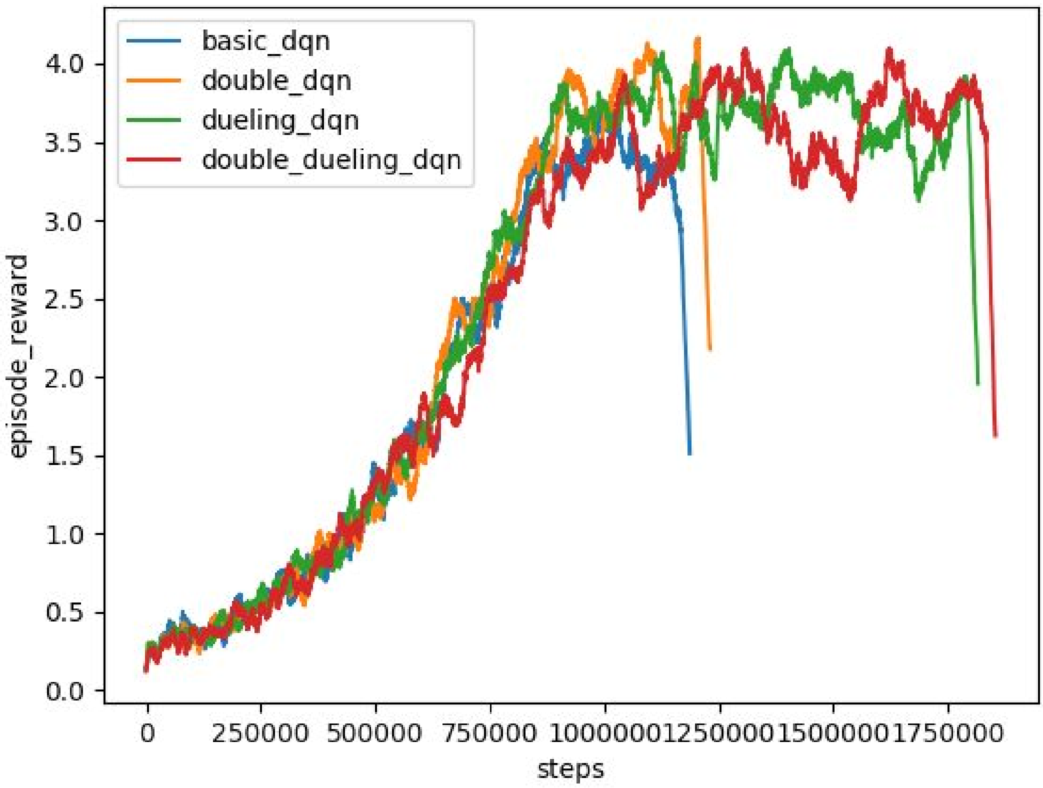
**Bonus (4%)**

Improvements to Policy Gradient (2%)

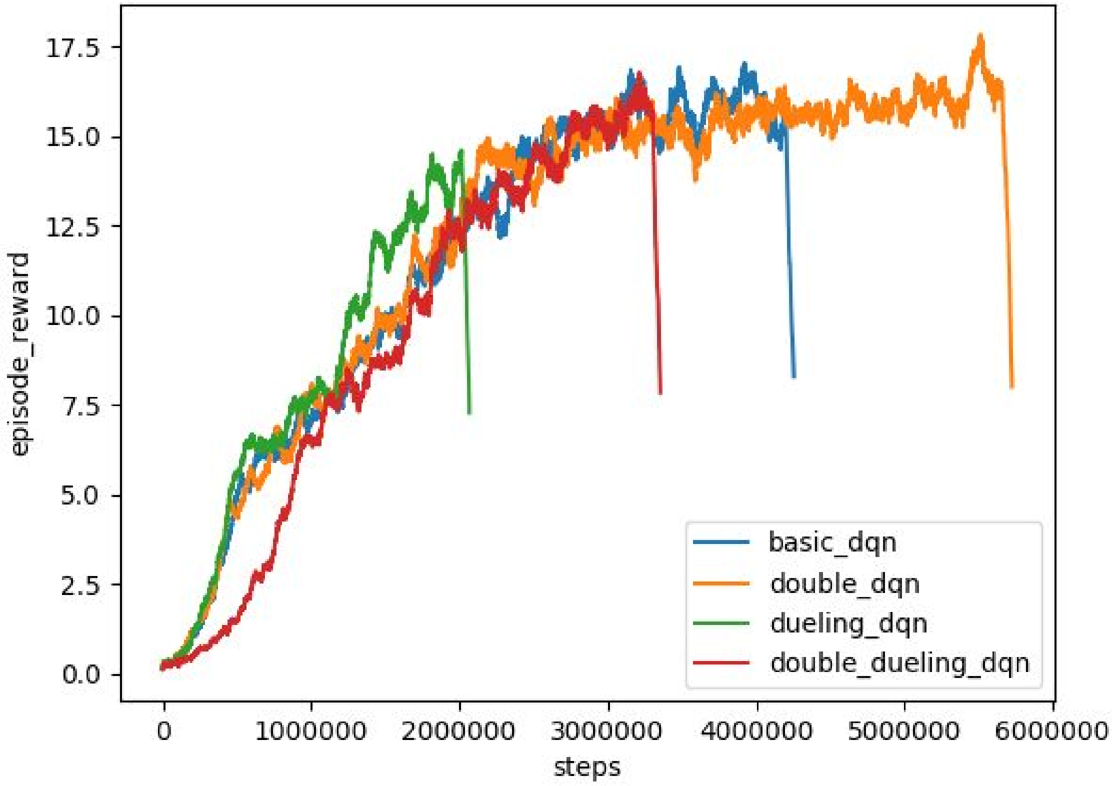
1. discount reward: 能夠解決action與reward的延遲關係，並減少reward的變異量、離散度。
   reward standardization: 讓reward平均正負平均，鼓勵較好的action，懲罰較差的action，而不影響其他的action。

Improvements to DQN (2%)

1. **Double DQN:** 強調利用online model做this state與next state的action決策，較慢更新的target model負責提供next Q value，減少online model高估Q(s, a) value，使學習更穩定。
   **Dueling DQN:** 將Q value拆分為Value scalar與Advantage vector，Value負責評估state平均優勢、Advantage評估每一個action對於state的優劣程度，理論上可以幫助學習穩定。
2. experiment: (a). laten_dim=128 (moving average=300)



experiment: (b). laten_dim=512 (moving average=300)




dueling可能幫助model再學習上比較平順、順利，但是double與dueling對於model最佳表現並無顯著差別。