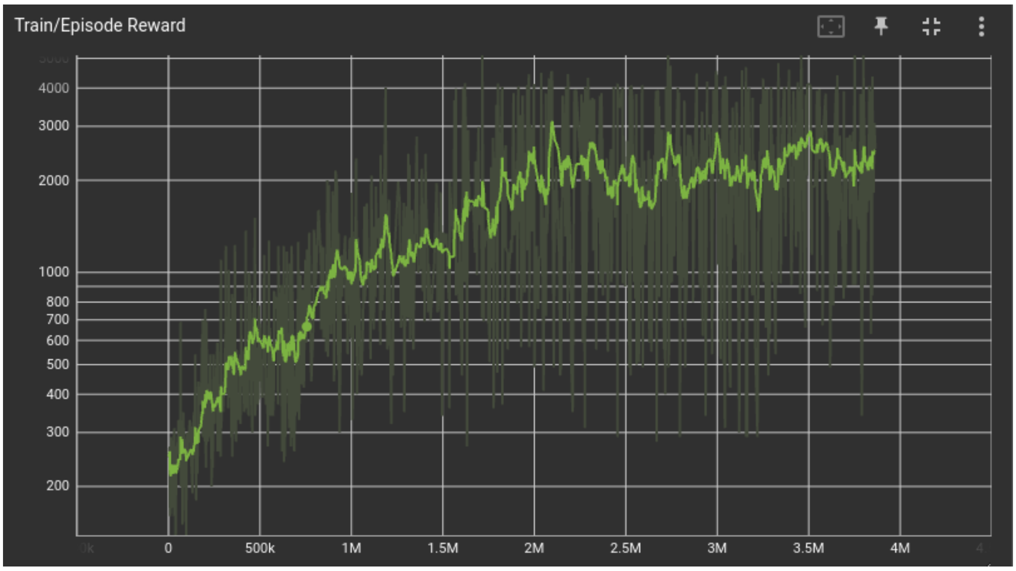
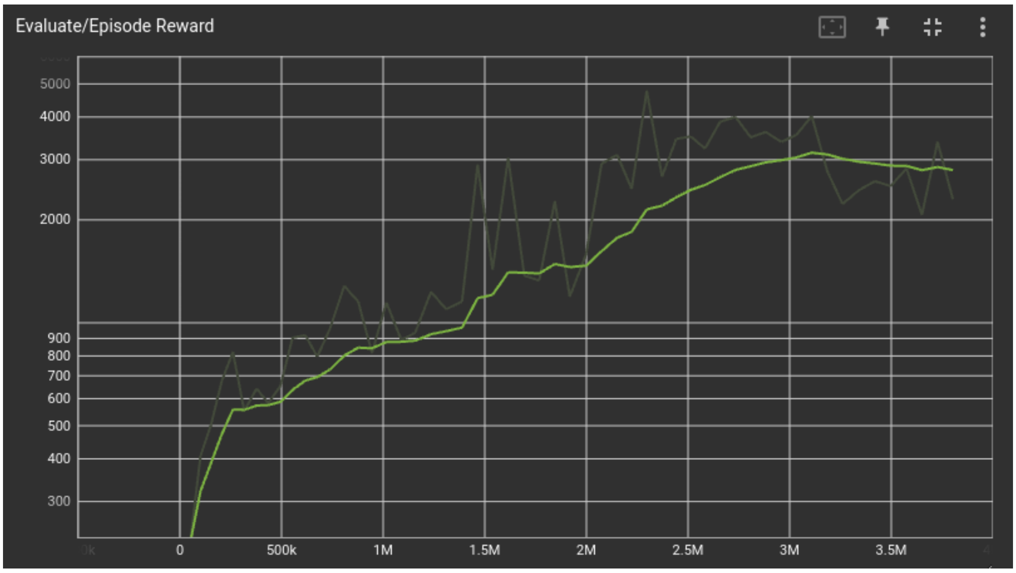
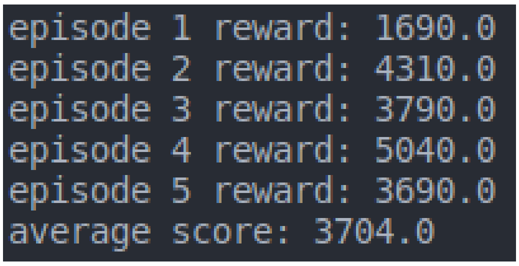
Lab 2 - DQN

312553024 江尚軒

- **Experimental Results (30%)**
  - ○ **Screenshot of Tensorboard training curve and testing results on DQN.**



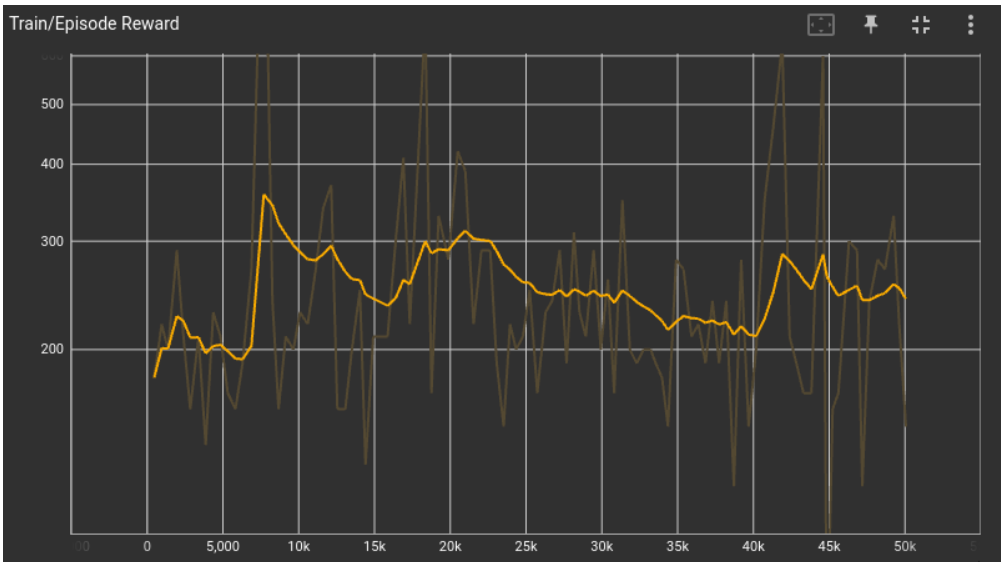


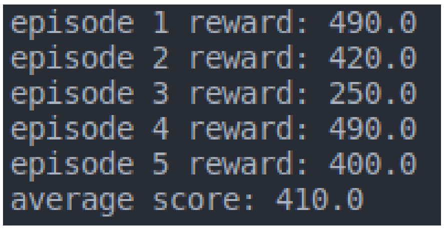


```
episode 1 reward: 1690.0
episode 2 reward: 4310.0
episode 3 reward: 3790.0
episode 4 reward: 5040.0
episode 5 reward: 3690.0
average score: 3704.0
```

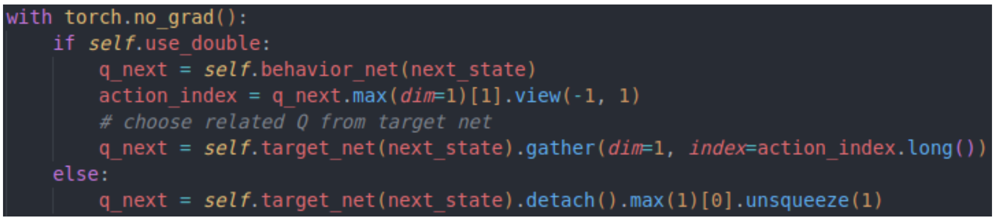
- **Experimental Results and Discussion of bonus parts (bonus) (20%)**
    - **Screenshot of Tensorboard training curve and testing results on DDQN, and discuss the difference between DQN and DDQN (3%).**

      (I am sorry that I don't have enough time to train the DDQN model, so I only train 100 episodes.)





```
episode 1 reward: 490.0
episode 2 reward: 420.0
episode 3 reward: 250.0
episode 4 reward: 490.0
episode 5 reward: 400.0
average score: 410.0
```

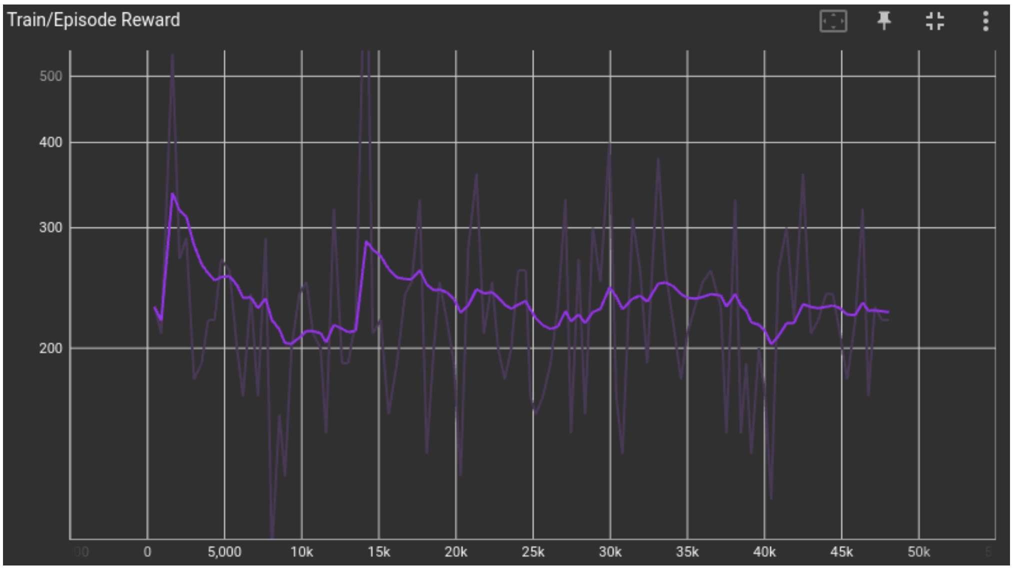


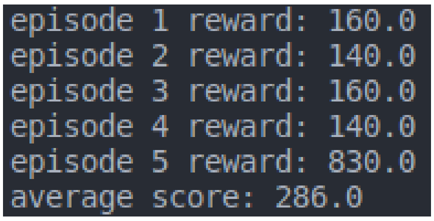
```
with torch.no_grad():
    if self.use_double:
        q_next = self.behavior_net(next_state)
        action_index = q_next.max(dim=1)[1].view(-1, 1)
        # choose related Q from target net
        q_next = self.target_net(next_state).gather(dim=1, index=action_index.long())
    else:
        q_next = self.target_net(next_state).detach().max(1)[0].unsqueeze(1)
```

Both DQN and DDQN are reinforcement learning algorithms based on Q-learning. DDQN improves upon DQN by addressing overestimation bias through a double Q-learning approach. In DDQN, two Q-networks are used to select and evaluate actions separately, resulting in more stable and accurate Q-value estimates. This helps in training more robust and effective decision-making agents.

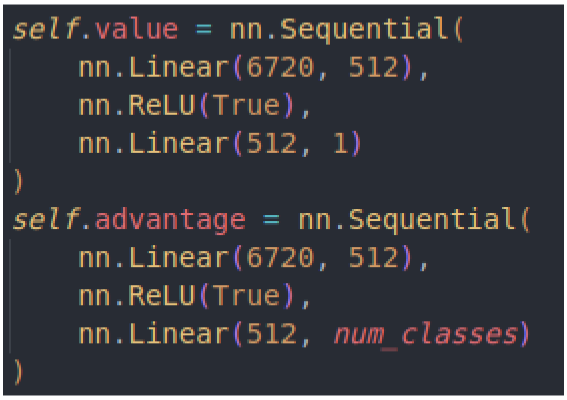
    - **Screenshot of Tensorboard training curve and testing results on Dueling DQN, and discuss the difference between DQN and Dueling DQN (3%).**

(I am sorry that I don’t have enough time to train the Dueling DQN model, so I only train 100 episodes.)

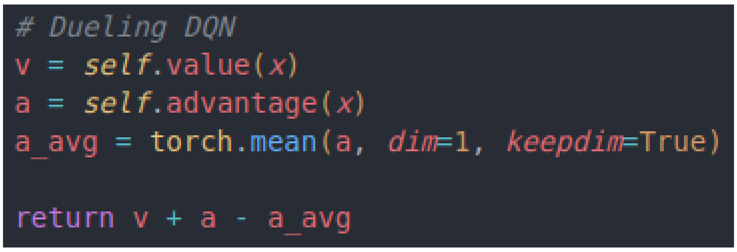




```
episode 1 reward: 160.0
episode 2 reward: 140.0
episode 3 reward: 160.0
episode 4 reward: 140.0
episode 5 reward: 830.0
average score: 286.0
```



```
self.value = nn.Sequential(
    nn.Linear(6720, 512),
    nn.ReLU(True),
    nn.Linear(512, 1)
)
self.advantage = nn.Sequential(
    nn.Linear(6720, 512),
    nn.ReLU(True),
    nn.Linear(512, num_classes)
)
```

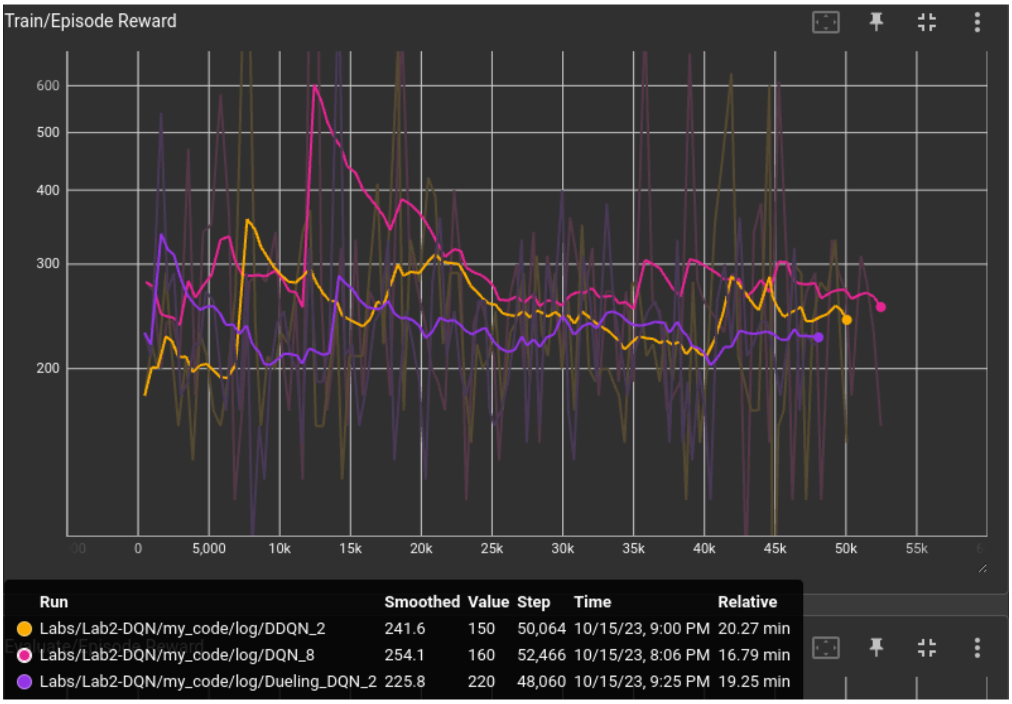


```
# Dueling DQN
v = self.value(x)
a = self.advantage(x)
a_avg = torch.mean(a, dim=1, keepdim=True)

return v + a - a_avg
```

DQN and Dueling DQN are both reinforcement learning algorithms, but Dueling DQN improves upon DQN by using a specific neural network architecture that separates the estimation of state values and action advantages, addressing issues like overestimation bias and enhancing learning efficiency.

- **Training curve comparison (DQN vs. DDQN vs. Dueling DQN)**



	Run	Smoothed	Value	Step	Time	Relative
●	Labs/Lab2-DQN/my_code/log/DDQN_2	241.6	150	50,064	10/15/23, 9:00 PM	20.27 min
●	Labs/Lab2-DQN/my_code/log/DQN_8	254.1	160	52,466	10/15/23, 8:06 PM	16.79 min
●	Labs/Lab2-DQN/my_code/log/Dueling_DQN_2	225.8	220	48,060	10/15/23, 9:25 PM	19.25 min