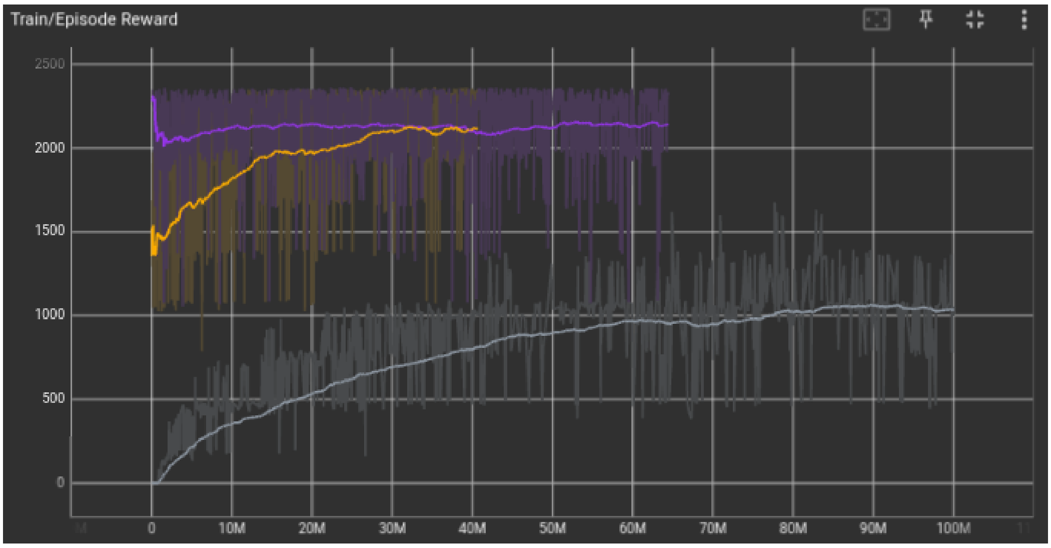
Lab 3 - PPO

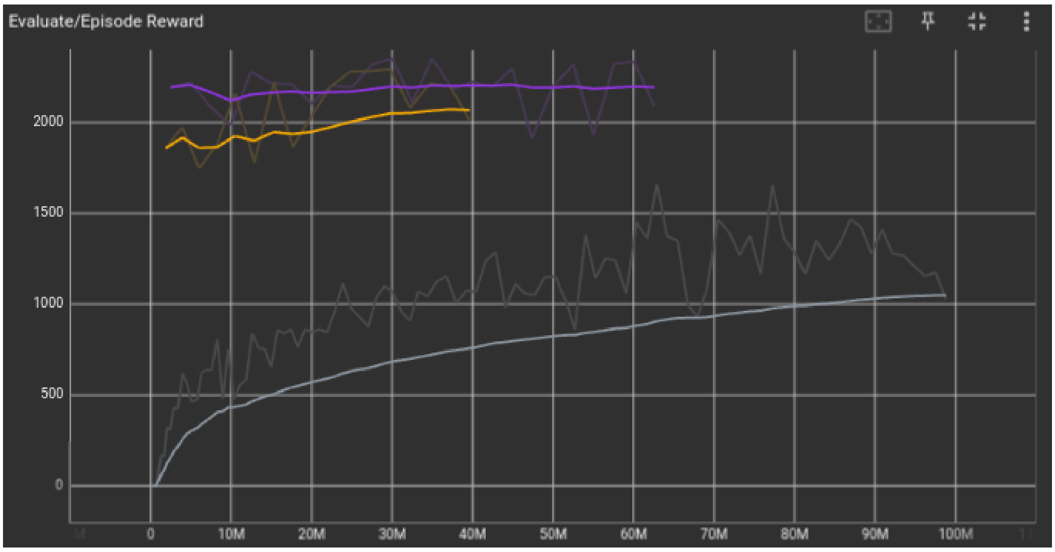
312553024 江尚軒

Experimental Results (30%)

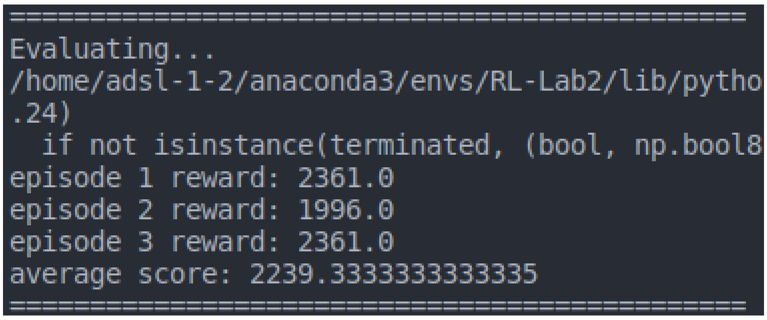
(1) Screenshot of Tensorboard training curve and testing results on PPO.

training curve:
The gray line is the first training, and learning rate = 2.5e-4
The yellow line is the second training, and learning rate = 1e-5
The purple line is the third training, and learning rate = 1e-6
I found that the reward was boosted when I decreased the learning rate and trained again on the pre-trained model.





testing results:



```
==============================================
Evaluating...
/home/adsl-1-2/anaconda3/envs/RL-Lab2/lib/pytho
.24)
  if not isinstance(terminated, (bool, np.bool8
episode 1 reward: 2361.0
episode 2 reward: 1996.0
episode 3 reward: 2361.0
average score: 2239.3333333333335
==============================================
```

Answer the questions of bonus parts (bonus) (20%)

(1) PPO is an on-policy or an off-policy algorithm? Why? (5%)

PPO is an on-policy reinforcement learning algorithm. This means that the agent uses its current policy to interact with the environment and generate data, and then uses that data to update its policy.

Off-policy algorithms, on the other hand, can use data generated by a different policy, such as a policy that was trained in the past or a policy that is being used by another agent.

PPO is on-policy because it uses the following update rule:

*θ_new = θ_old + α * (g(θ_old) - π_old(a|s)) * grad log π_old(a|s)*

where:
θ is the policy parameter
α is the learning rate
g(θ) is the expected return of the policy πθ
π_old(a|s) is the agent's current policy
grad log π_old(a|s) is the gradient of the log of the agent's current policy
This update rule ensures that the agent's policy always remains close to its current policy, which makes it more stable and easier to train.

(2) Explain how PPO ensures that policy updates at each step are not too large to avoid destabilization. (5%)

PPO uses a technique called clipping to ensure that policy updates at each step are not too large. Clipping works by limiting the amount that the policy can change at each step. This helps to prevent the policy from becoming unstable and oscillating wildly.

The clipping algorithm works as follows:

Calculate the ratio of the new policy to the old policy.
If the ratio is greater than a certain threshold, clip it to the threshold.
Update the policy using the clipped ratio.
The threshold value is typically set to a value between 0.1 and 0.5. This means that the policy can only change by up to 10% or 50% at each step.

Clipping helps to ensure that the policy updates are gradual and that the policy does not become too unstable. This makes PPO a more robust and stable reinforcement learning algorithm.

(3) Why is GAE-lambda used to estimate advantages in PPO instead of just one-step advantages? How does it contribute to improving the policy learning process? (5%)

GAE-lambda is used to estimate advantages in PPO instead of just one-step advantages because it provides a more accurate estimate of the value of taking an action in a given state. This is because GAE-lambda takes into account the long-term consequences of taking an action, not just the immediate reward.

Specifically, GAE-lambda estimates the advantage of taking an action in a given state by considering the discounted sum of future rewards that are likely to be obtained by taking that action. This sum is discounted by a factor of lambda, which controls how much weight is given to future rewards.

A higher value of lambda gives more weight to future rewards, which results in a more accurate estimate of the advantage of taking an action. However, a higher value of lambda can also make the algorithm more sensitive to noise.

(4) Please explain what the lambda parameter represents in GAE-lambda, and how adjusting the lambda parameter affects the training process and performance of PPO? (5%)

The lambda parameter in GAE-lambda represents the importance of future rewards when estimating the advantage of taking an action. A higher value of lambda gives more weight to future rewards, while a lower value of lambda gives more weight to immediate rewards.

Adjusting the lambda parameter can affect the training process and performance of PPO in a number of ways.

- **Accuracy of the advantage estimate:** A higher value of lambda will result in a more accurate estimate of the advantage of taking an action. This is because GAE-lambda takes into account the long-term consequences of taking an action, not just the immediate reward.
- **Stability of the policy update:** A higher value of lambda can make the policy update more stable. This is because it reduces the variance of the policy update.
- **Sample efficiency of the algorithm:** A higher value of lambda can make the algorithm more sample-efficient. This is because it allows the algorithm to learn from future rewards, which can help it avoid getting stuck in local optima.