Assignment 2: Policy Gradient

Andrew ID: `kevinh3`
Collaborators: `Write the Andrew IDs of your collaborators here (if any).`
NOTE: Please do **NOT** change the sizes of the answer blocks or plots.

5 Small-Scale Experiments

5.1 Experiment 1 (Cartpole) – [5 points total]

5.1.1 Configurations

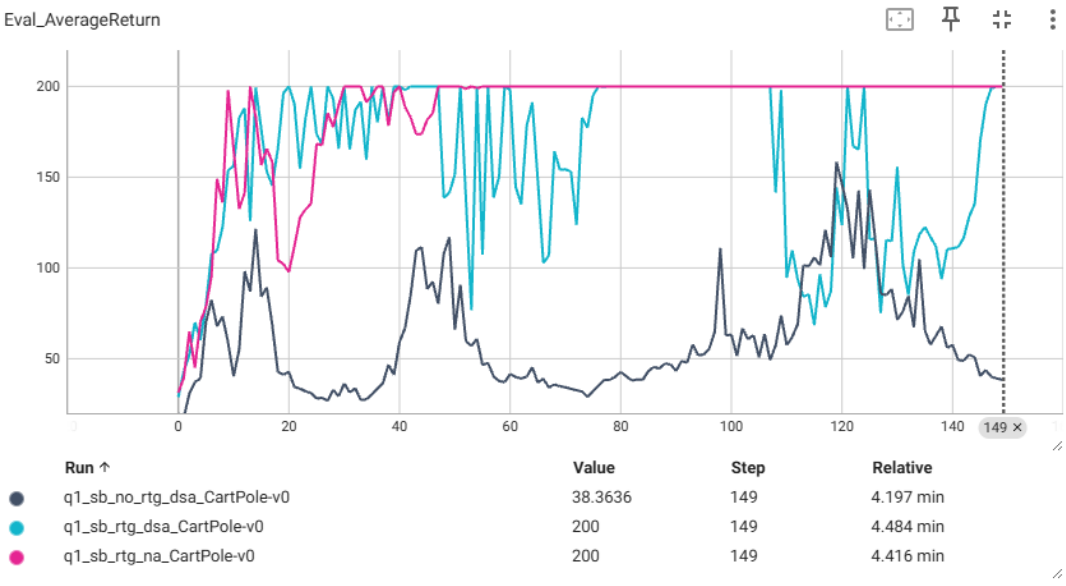
Q5.1.1

```
python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 150 -b 1500 \
    -dsa --exp_name q1_sb_no_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 150 -b 1500 \
    -rtg -dsa --exp_name q1_sb_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 150 -b 1500 \
    -rtg --exp_name q1_sb_rtg_na

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 150 -b 6000 \
    -dsa --exp_name q1_lb_no_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 150 -b 6000 \
    -rtg -dsa --exp_name q1_lb_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 150 -b 6000 \
    -rtg --exp_name q1_lb_rtg_na
```

5.1.2 Plots

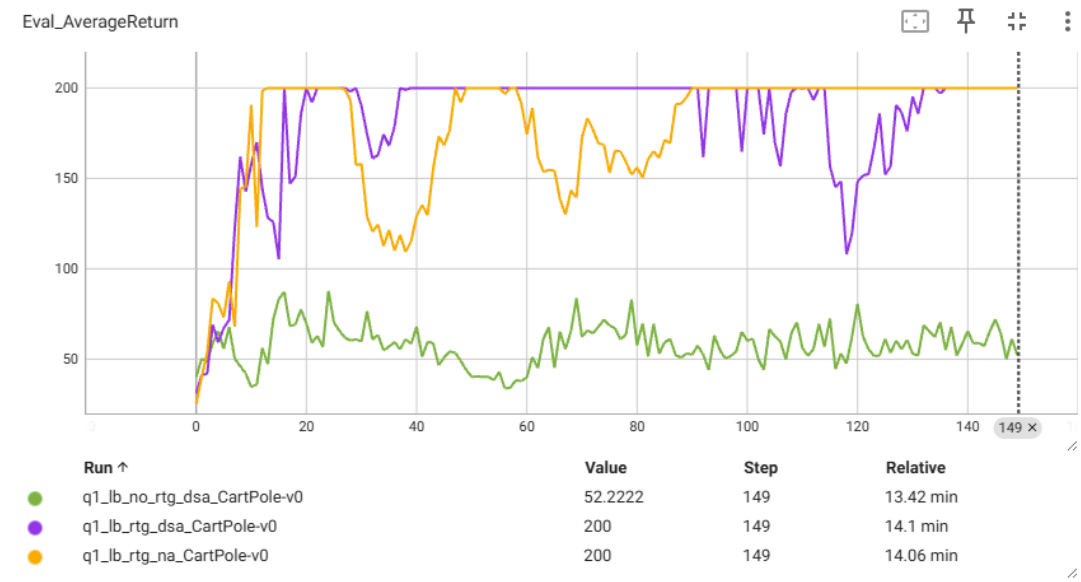
5.1.2.1 Small batch – [1 points]

Q5.1.2.1



5.1.2.2 Large batch – [1 points]

Q5.1.2.2



5.1.3 Analysis

5.1.3.1 Value estimator – [1 points]

Q5.1.3.1

The value estimator using reward-to-go has a better performance than that using trajectory-centric when the advantage standardization is disabled. This is because the reward-to-go reduces the noise and variance by only providing each action with its own future reward. Thus, that reward signal is more aligned and less noisy compared to the one provided by the trajectory-centric estimator. From the graph, it can be noticed that those without using reward-to-go tend to have a much lower average return value and fluctuate a lot throughout the entire learning.

5.1.3.2 Advantage standardization – [1 points]

Q5.1.3.2

The advantage standardization helps a lot for the estimator. It helps to normalize the advantages to have a zero mean and unit variance. This can help to stabilize the learning by making sure that the advantages won't have a huge impact and make the learning fluctuate a lot. It can be observed in the graph that when the advantage standardization is enabled, the learning tends to fluctuate less compared with those not using the advantage standardization.

5.1.3.3 Batch size – [1 points]

Q5.1.3.3

The batch size does make an impact on the estimator. With a larger batch size, the learning tends to have a lower variance and be more stable, especially when the advantage standardization is disabled. As shown in the graph, with a larger batch size, the learning tends to be much stable with a lower variance when the advantage standardization is disabled.

5.2 Experiment 2 (InvertedPendulum) – [4 points total]

5.2.1 Configurations – [1.5 points]

Q5.2.1

```
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 6000 -lr 1e-3 -rtg \
    --exp_name q2_b<6000>_r<1e-3>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 6000 -lr 1e-2 -rtg \
    --exp_name q2_b<6000>_r<1e-2>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 3000 -lr 1e-2 -rtg \
    --exp_name q2_b<3000>_r<1e-2>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 3000 -lr 2e-2 -rtg \
    --exp_name q2_b<3000>_r<2e-2>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 2000 -lr 2e-2 -rtg \
    --exp_name q2_b<2000>_r<2e-2>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 2000 -lr 3e-2 -rtg \
    --exp_name q2_b<2000>_r<3e-2>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 1500 -lr 1e-3 -rtg \
    --exp_name q2_b<1500>_r<1e-3>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 1000 -lr 1e-2 -rtg \
    --exp_name q2_b<1000>_r<1e-2>
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
    --ep_len 1000 --discount 0.92 -n 100 -l 2 -s 64 -b 2000 -lr 1e-2 -rtg \
    --exp_name q2_b<2000>_r<1e-2>
```

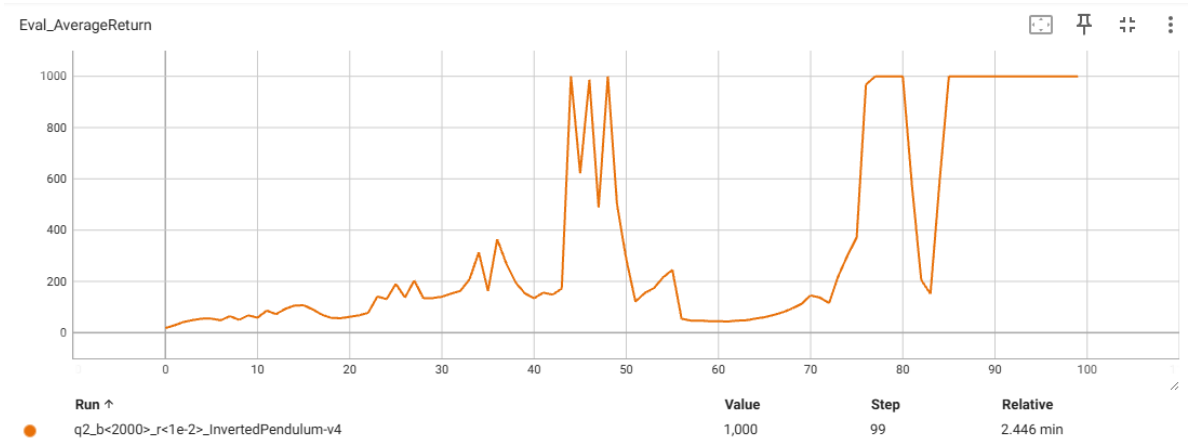
5.2.2 smallest b* and largest r* (same run) – [1.5 points]

Q5.2.2

The smallest batch size b: 2000
The largest r: 1e-2
I found these values by starting with a large batch size and a low learning rate. Then, I increased the learning rate to see its limit, where the curve diverges. After knowing the learning rate limit, I decreased the batch size. I kept repeating these steps until I found the optimal values.

5.2.3 Plot – [1 points]

Q5.2.3



7 More Complex Experiments

7.1 Experiment 3 (LunarLander) – [1 points total]

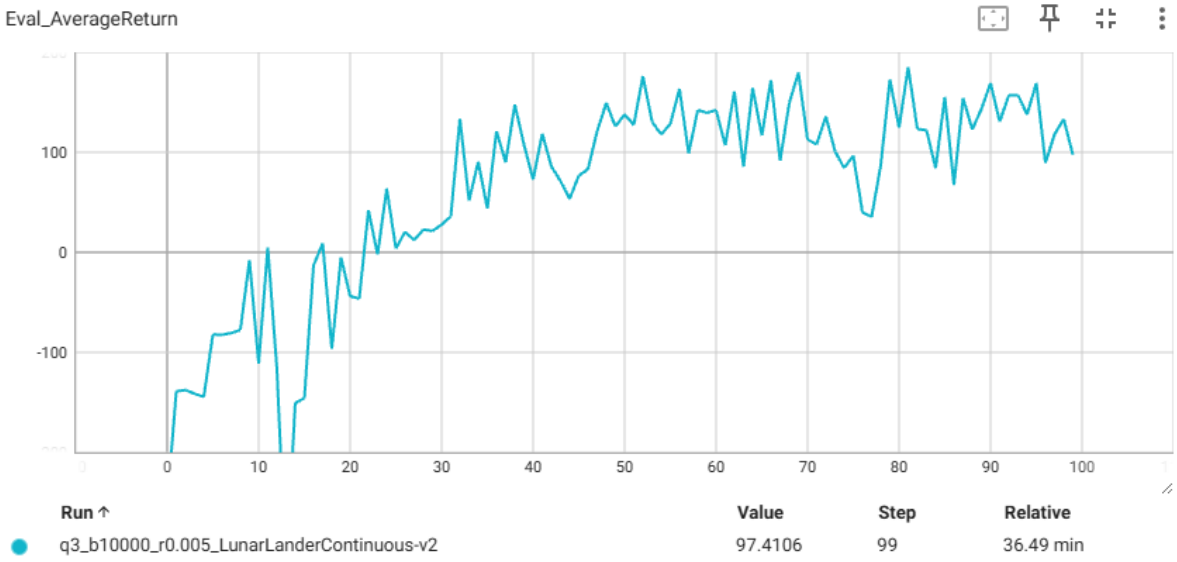
7.1.1 Configurations

Q7.1.1

```
python rob831/scripts/run_hw2.py \
    --env_name LunarLanderContinuous-v4 --ep_len 1000
    --discount 0.99 -n 100 -l 2 -s 64 -b 10000 -lr 0.005 \
    --reward_to_go --nn_baseline --exp_name q3_b10000_r0.005
```

7.1.2 Plot – [1 points]

Q7.1.2



7.2 Experiment 4 (HalfCheetah) – [1 points]

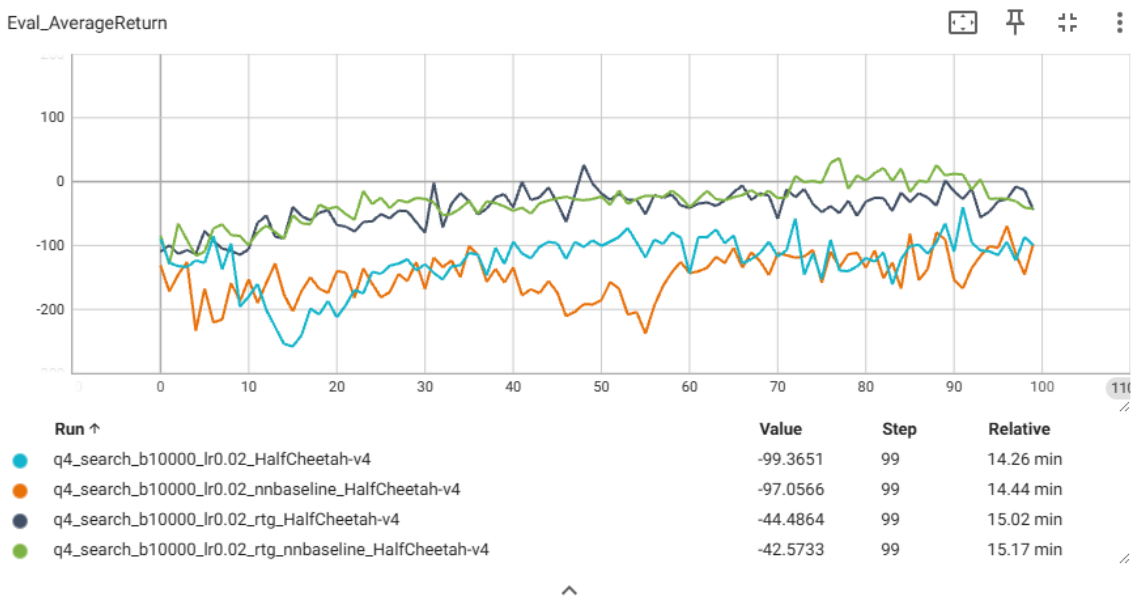
7.2.1 Configurations

Q7.2.1

```
python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.02 \
    --exp_name q4_search_b10000_lr0.02
python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.02 -rtg \
    --exp_name q4_search_b10000_lr0.02_rtg
python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.02 --nn_baseline \
    --exp_name q4_search_b10000_lr0.02_nnbaseline
python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.02 -rtg --nn_baseline \
    --exp_name q4_search_b10000_lr0.02_rtg_nnbaseline
```

7.2.2 Plot – [1 points]

Q7.2.2



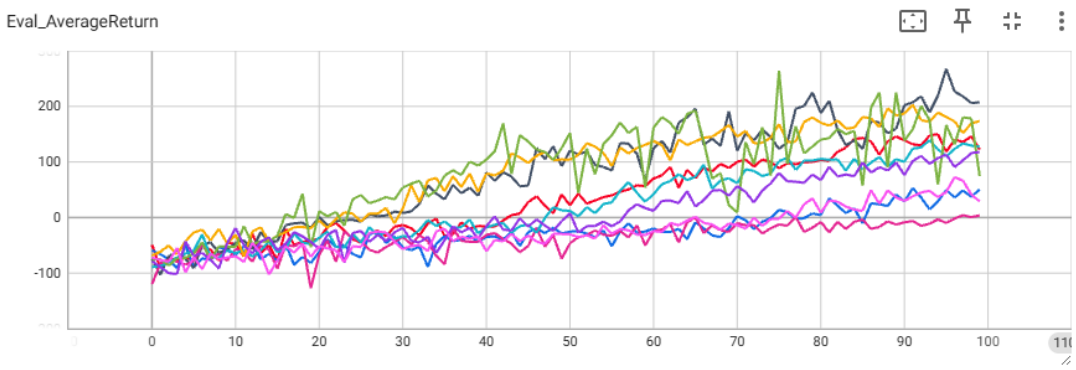
7.2.3 Optimal b* and r* – [0.5 points]

Q7.2.3

Optimal b: 35000
Optimal r: 0.02
I found these optimal values by simply running all 9 combinations for different batch sizes, which include 15000, 35000, and 55000, and distinct learning rates, which include 5e-3, 1e-2, and 2e-2.

7.2.4 Plot – [0.5 points]

Q7.2.4



Run	Value	Step	Relative
q4_search_b<15000>_lr<1e-2>_rtg_nnbaseline_HalfCheetah-v4	118.033	99	16.98 min
q4_search_b<15000>_lr<2e-2>_rtg_nnbaseline_HalfCheetah-v4	74.5276	99	14.9 min
q4_search_b<15000>_lr<5e-3>_rtg_nnbaseline_HalfCheetah-v4	3.9106	99	15.67 min
q4_search_b<35000>_lr<1e-2>_rtg_nnbaseline_HalfCheetah-v4	127.1817	99	1.144 hr
q4_search_b<35000>_lr<2e-2>_rtg_nnbaseline_HalfCheetah-v4	207.488	99	33.65 min
q4_search_b<35000>_lr<5e-3>_rtg_nnbaseline_HalfCheetah-v4	50.8424	99	38.03 min
q4_search_b<55000>_lr<1e-2>_rtg_nnbaseline_HalfCheetah-v4	122.366	99	1.22 hr
q4_search_b<55000>_lr<2e-2>_rtg_nnbaseline_HalfCheetah-v4	173.6909	99	1.24 hr
q4_search_b<55000>_lr<5e-3>_rtg_nnbaseline_HalfCheetah-v4	29.0014	99	1.228 hr

7.2.5 Describe how b* and r* affect task performance – [0.5 points]

Q7.2.5

When the batch size is small, high gradient noise can lead to unstable updates; however, these unstable updates might help exploration. A larger batch size, on the other hand, lowers the gradient variance and leads to a smoother convergence. However, if the batch size is too large, the performance might not be able to improve because of the diminishing returns as the gradient is almost the same as the true value. It can be observed from the graph that small batch size learning tends to be less stable, and very large batch size tends to stabilize the curve too much, which also doesn't lead to high values.

Learning rates also affect the performance. With a low learning rate, the learning is more stable but leads to slower training. On the other hand, high learning rates lead to faster progress per update but have a risk of divergence. Therefore, it is better to have a high learning rate when the batch size is large, as a large batch size leads to lower variance.

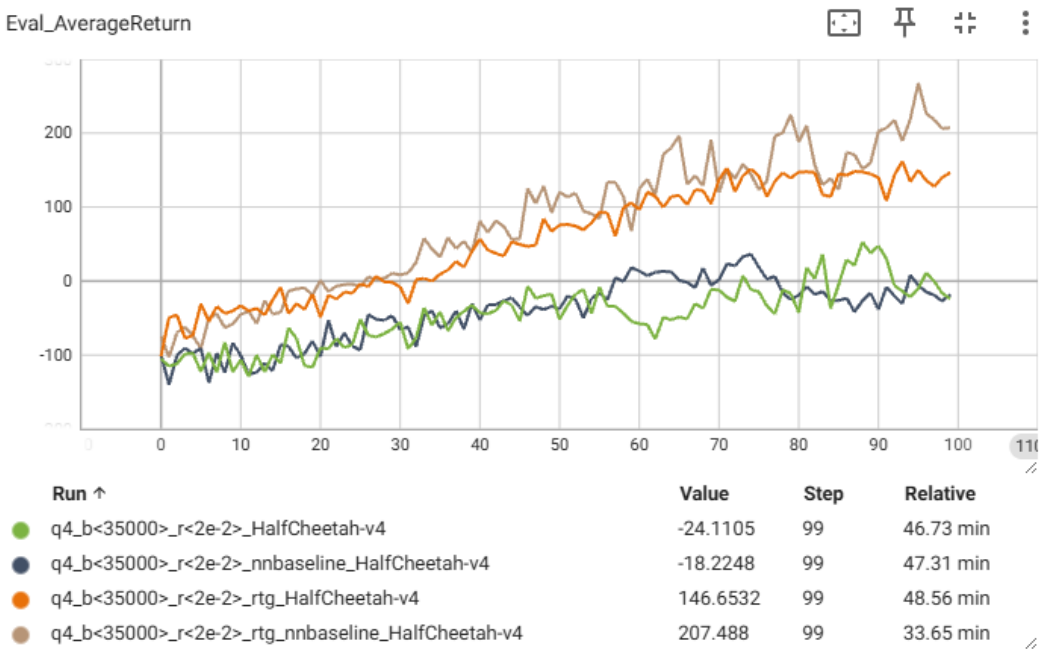
7.2.6 Configurations with optimal b* and r* – [0.5 points]

Q7.2.6

```
python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 35000 -lr 2e-2 \
    --exp_name q4_b<35000>_r<2e-2>

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 35000 -lr 2e-2 -rtg \
    --exp_name q4_b<35000>_r<2e-2>_rtg

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 35000 -lr 2e-2 --nn_baseline \
    --exp_name q4_b<35000>_r<2e-2>_nnbaseline

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 35000 -lr 2e-2 -rtg --nn_baseline \
    --exp_name q4_b<35000>_r<2e-2>_rtg_nnbaseline
```

7.2.7 Plot for four runs with optimal b* and r* – [0.5 points]

Q7.2.7



8 Implementing Generalized Advantage Estimation

8.1 Experiment 5 (Hopper) – [4 points]

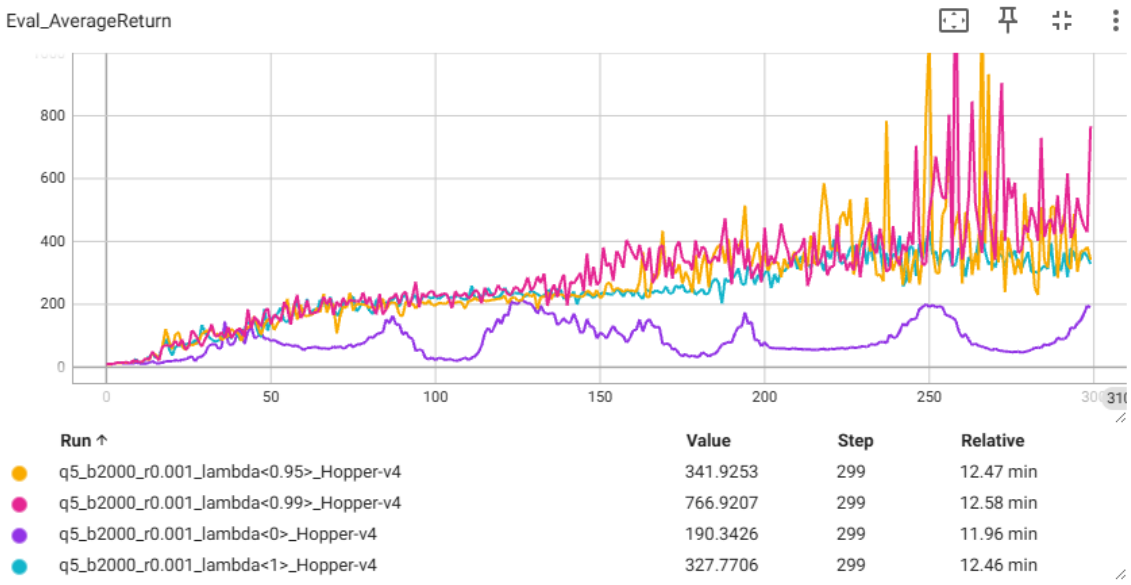
8.1.1 Configurations

Q8.1.1

```
# λ ∈ [0, 0.95, 0.99, 1]
python rob831/scripts/run_hw2.py \
    --env_name Hopper-v4 --ep_len 1000
    --discount 0.99 -n 300 -l 2 -s 32 -b 2000 -lr 0.001 \
    --reward_to_go --nn_baseline --action_noise_std 0.5 --gae_lambda <λ> \
    --exp_name q5_b2000_r0.001_lambda<λ>
```

8.1.2 Plot – [2 points]

Q8.1.2



Run	Value	Step	Relative
q5_b2000_r0.001_lambda<0.95>_Hopper-v4	341.9253	299	12.47 min
q5_b2000_r0.001_lambda<0.99>_Hopper-v4	766.9207	299	12.58 min
q5_b2000_r0.001_lambda<0>_Hopper-v4	190.3426	299	11.96 min
q5_b2000_r0.001_lambda<1>_Hopper-v4	327.7706	299	12.46 min

8.1.3 Describe how λ affects task performance – [2 points]

Q8.1.3

Lambda controls the balance between bias and variance in estimating advantages. With a lower lambda, the performance tends to be more stable but might be too biased. However, with a high lambda, the performance can be less stable and have a higher variance, but lead to a more accurate result. Therefore, it can be noticed that a lower lambda is great for task that has an immediate reward strongly correlated with long-term success, while a higher lambda benefits when rewards are sparse or tasks that need continuous control.

9 More Bonus!

9.1 Parallelization – [1.5 points]

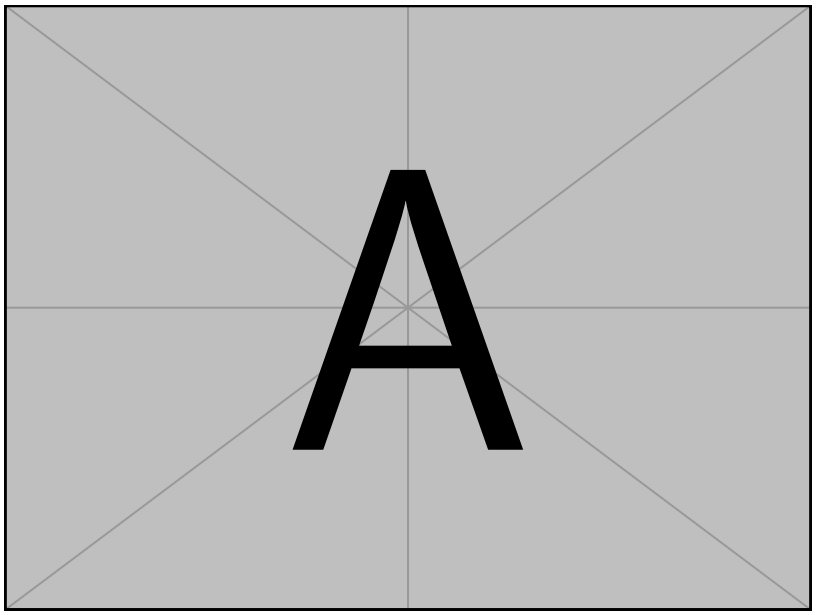
Q9.1

Difference in training time:

```
python rob831/scripts/run_hw2.py \
```

9.2 Multiple gradient steps – [1 points]

Q9.1




```
python rob831/scripts/run_hw2.py \
```