Assignment 2: Policy Gradient

Andrew ID: `nikitama`
Collaborators: -
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

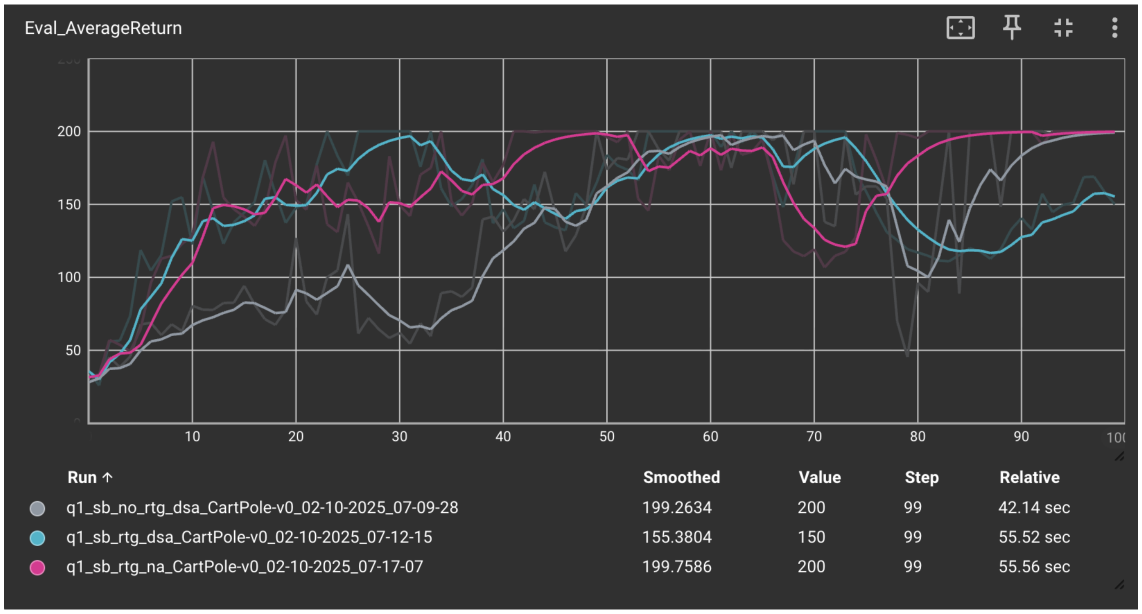
Q5.1.1

```
python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
    -dsa --exp_name q1_sb_no_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
    -rtg -dsa --exp_name q1_sb_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
    -rtg --exp_name q1_sb_rtg_na

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
    -dsa --exp_name q1_lb_no_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
    -rtg -dsa --exp_name q1_lb_rtg_dsa

python rob831/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
    -rtg --exp_name q1_lb_rtg_na
```

5.1.2 Plots

5.1.2.1 Small batch – [1 points]

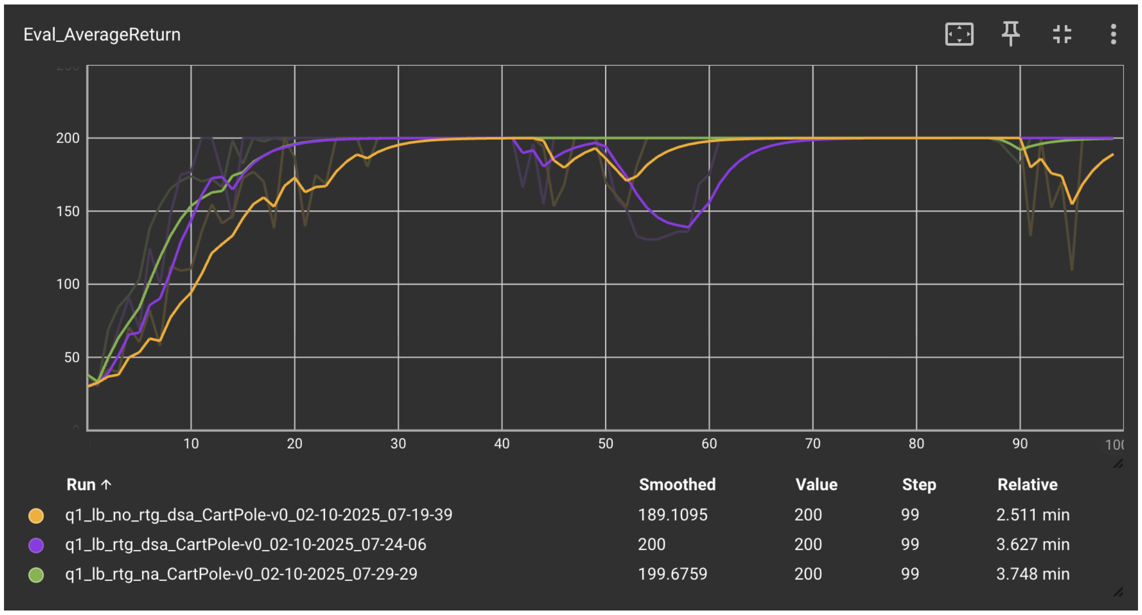
Q5.1.2.1



Run ↑	Smoothed	Value	Step	Relative
● q1_sb_no_rtg_dsa_CartPole-v0_02-10-2025_07-09-28	199.2634	200	99	42.14 sec
● q1_sb_rtg_dsa_CartPole-v0_02-10-2025_07-12-15	155.3804	150	99	55.52 sec
● q1_sb_rtg_na_CartPole-v0_02-10-2025_07-17-07	199.7586	200	99	55.56 sec

5.1.2.2 Large batch – [1 points]

Q5.1.2.2



	Run ↑	Smoothed	Value	Step	Relative
●	q1_lb_no_rtg_dsa_CartPole-v0_02-10-2025_07-19-39	189.1095	200	99	2.511 min
●	q1_lb_rtg_dsa_CartPole-v0_02-10-2025_07-24-06	200	200	99	3.627 min
●	q1_lb_rtg_na_CartPole-v0_02-10-2025_07-29-29	199.6759	200	99	3.748 min

5.1.3 Analysis

5.1.3.1 Value estimator – [1 points]

Q5.1.3.1

Value estimator using reward-to-go has better performance and more stable training as compared to trajectory centric one. RTG ensures that each time step's rewards only impact the future cumulated rewards, thus reducing variance.

5.1.3.2 Advantage standardization – [1 points]

Q5.1.3.2

Advantage standarization is helpful in case of smaller batch sizes as it helps in early convergence. However, in case of large batch size it does not have any significant improvement.

5.1.3.3 Batch size – [1 points]

Q5.1.3.3

Larger batch size helps in stable and early convergence since the gradients are much smoother in case of larger batch sizes as compared to smaller batch sizes.

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 1000 -lr 0.005 -rtg --exp_name q2_b_1000_r_0.005

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 0.01 -rtg --exp_name q2_b_1000_r_0.01

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 0.02 -rtg --exp_name q2_b_1000_r_0.02

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 0.05 -rtg --exp_name q2_b_1000_r_0.05

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 \
--ep_len 1000 --discount 0.92 \
-n 100 -l 2 -s 64 -b 5000 -lr 0.01 -rtg --exp_name q2_b_5000_r_0.01
```

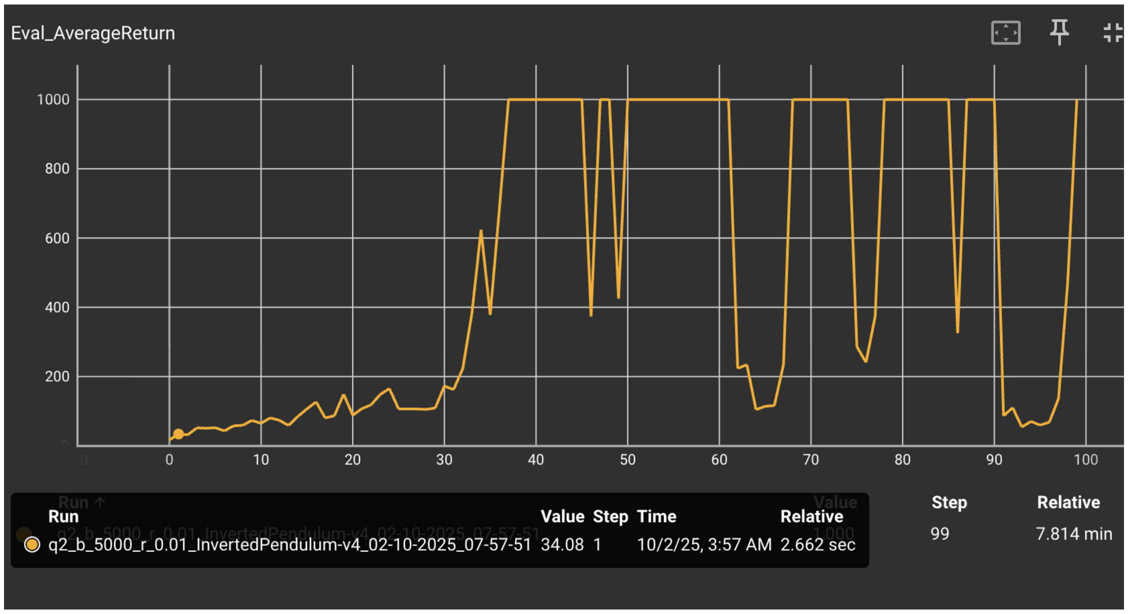
5.2.2 smallest b* and largest r* (same run) – [1.5 points]

Q5.2.2

Smallest batch size is 1000 and largest learning rate is 0.01

5.2.3 Plot – [1 points]

Q5.2.3



Run	Value	Step	Time	Relative
● q2_b_5000_r_0.01_InvertedPendulum-v4_02-10-2025_07-57-51	34.08	1	10/2/25, 3:57 AM	2.662 sec

Step	Relative
99	7.814 min

7 More Complex Experiments

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

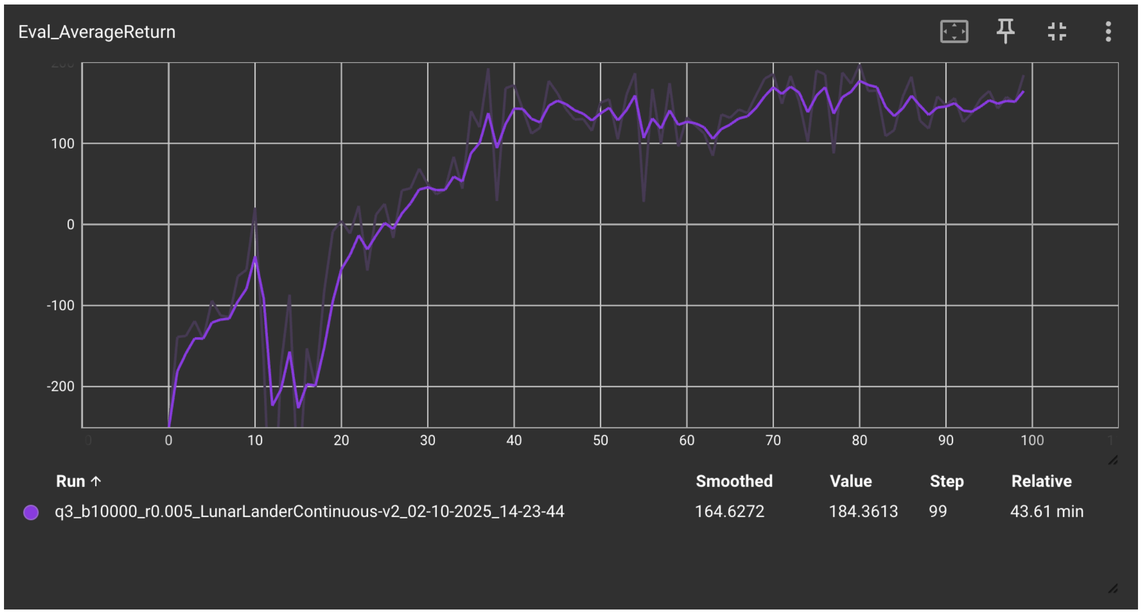
7.1.1 Configurations

Q7.1.1

```
python rob831/scripts/run_hw2.py \
    --env_name LunarLanderContinuous-v2 --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



Run ↑	Smoothed	Value	Step	Relative
● q3_b10000_r0.005_LunarLanderContinuous-v2_02-10-2025_14-23-44	164.6272	184.3613	99	43.61 min

7.2 Experiment 4 (HalfCheetah) – [1 points]

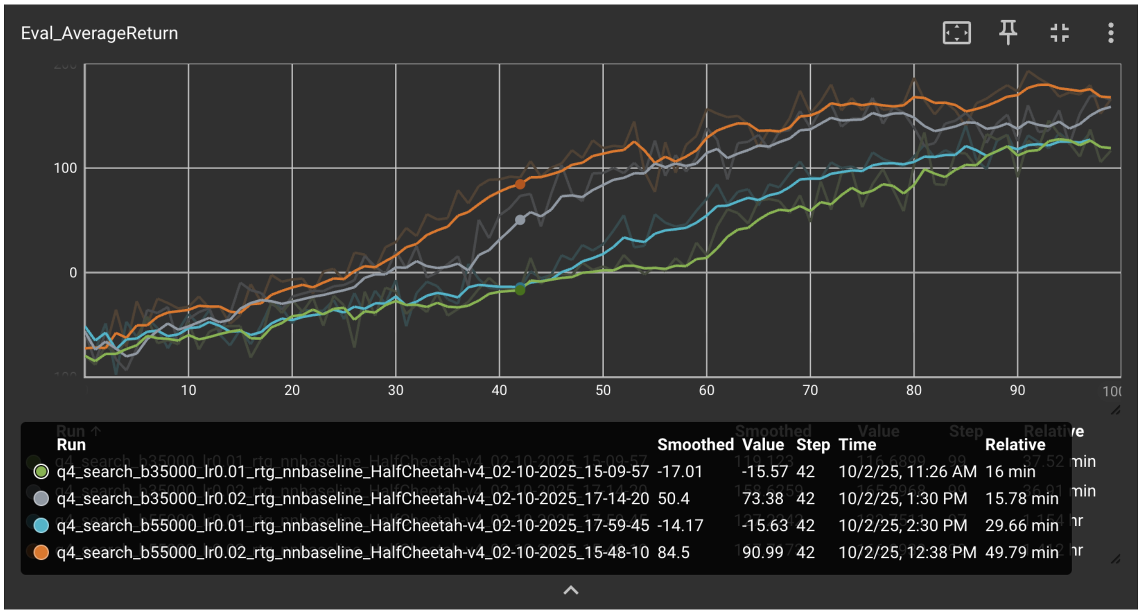
7.2.1 Configurations

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 35000 -lr 0.01 -rtg --nn_baseline \
--exp_name q4_search_b35000_lr0.01_rtg_nnbaseline

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 55000 -lr 0.02 -rtg --nn_baseline \
--exp_name q4_search_b55000_lr0.02_rtg_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 0.02 -rtg --nn_baseline \
--exp_name q4_search_b35000_lr0.02_rtg_nnbaseline

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 55000 -lr 0.01 -rtg --nn_baseline \
--exp_name q4_search_b55000_lr0.01_rtg_nnbaseline
```

7.2.2 Plot – [1 points]

Q7.2.2



	Run	Smoothed	Value	Step	Time	Relative
🟩	q4_search_b35000_lr0.01_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_15-09-57	-17.01	-15.57	42	10/2/25, 11:26 AM	16 min
●	q4_search_b35000_lr0.02_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_17-14-20	50.4	73.38	42	10/2/25, 1:30 PM	15.78 min
🟦	q4_search_b55000_lr0.01_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_17-59-45	-14.17	-15.63	42	10/2/25, 2:30 PM	29.66 min
🟧	q4_search_b55000_lr0.02_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_15-48-10	84.5	90.99	42	10/2/25, 12:38 PM	49.79 min

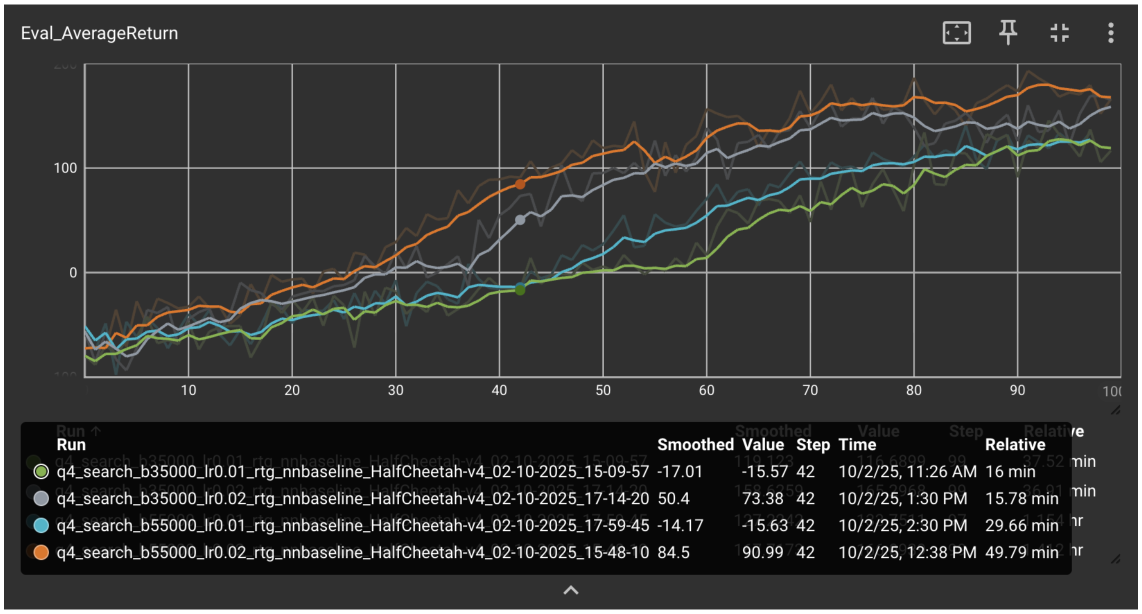
7.2.3 Optimal b* and r* – [0.5 points]

Q7.2.3

b = 55000 and r = 0.02

7.2.4 Plot – [0.5 points]

Q7.2.4



	Run	Smoothed	Value	Step	Time	Relative
🟢	q4_search_b35000_lr0.01_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_15-09-57	-17.01	-15.57	42	10/2/25, 11:26 AM	16 min
⚪	q4_search_b35000_lr0.02_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_17-14-20	50.4	73.38	42	10/2/25, 1:30 PM	15.78 min
🔵	q4_search_b55000_lr0.01_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_17-59-45	-14.17	-15.63	42	10/2/25, 2:30 PM	29.66 min
🟠	q4_search_b55000_lr0.02_rtg_nnbaseline_HalfCheetah-v4_02-10-2025_15-48-10	84.5	90.99	42	10/2/25, 12:38 PM	49.79 min

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

Q7.2.5

Increasing the batch size leads to more stable and early convergence because of smoother gradients accumulations over steps. Smaller batch sizes produce jittery and noisy results but faster updates. Increasing the learning rate leads to better results because smaller learning rate might get stuck in local neighborhood and lead to slow learning.

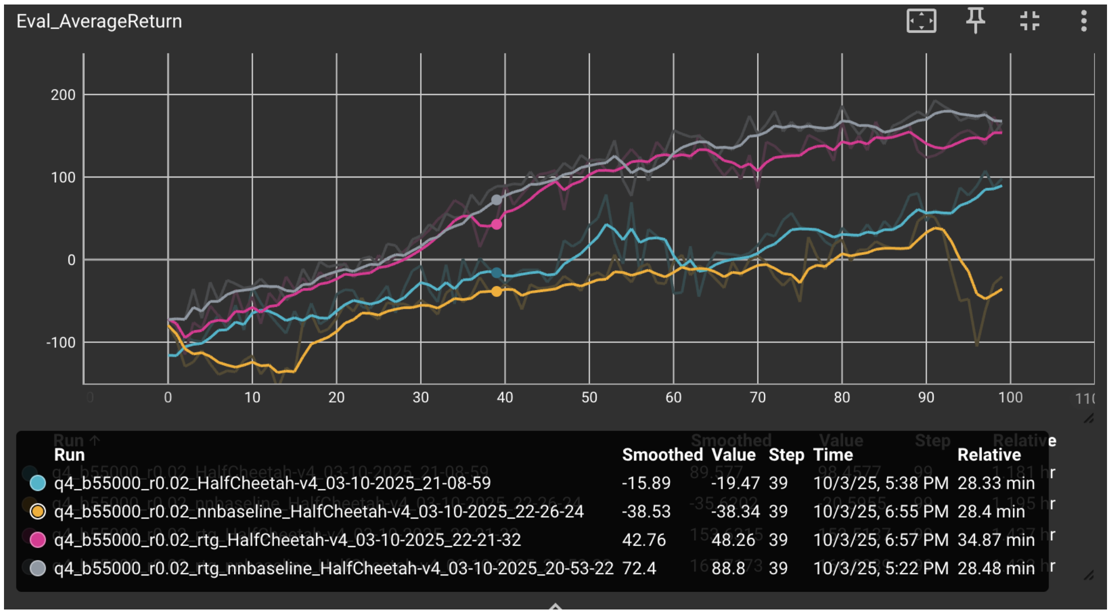
7.2.6 Configurations with optimal b* and r* – [0.5 points]

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 55000 -lr 0.02 \
    --exp_name q4_b55000_r0.02

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 55000 -lr 0.02 -rtg \
--exp_name q4_b55000_r0.02_rtg

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 55000 -lr 0.02 --nn_baseline \
    --exp_name q4_b55000_r0.02_nnbaseline

python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b 55000 -lr 0.02 -rtg --nn_baseline \
    --exp_name q4_b55000_r0.02_rtg_nnbaseline
```

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

Q7.2.7



	Run	Smoothed	Value	Step	Time	Relative
●	q4_b55000_r0.02_HalfCheetah-v4_03-10-2025_21-08-59	-15.89	-19.47	39	10/3/25, 5:38 PM	28.33 min
●	q4_b55000_r0.02_nnbaseline_HalfCheetah-v4_03-10-2025_22-26-24	-38.53	-38.34	39	10/3/25, 6:55 PM	28.4 min
●	q4_b55000_r0.02_rtg_HalfCheetah-v4_03-10-2025_22-21-32	42.76	48.26	39	10/3/25, 6:57 PM	34.87 min
●	q4_b55000_r0.02_rtg_nnbaseline_HalfCheetah-v4_03-10-2025_20-53-22	72.4	88.8	39	10/3/25, 5:22 PM	28.48 min

8 Implementing Generalized Advantage Estimation

8.1 Experiment 5 (Hopper) – [4 points]

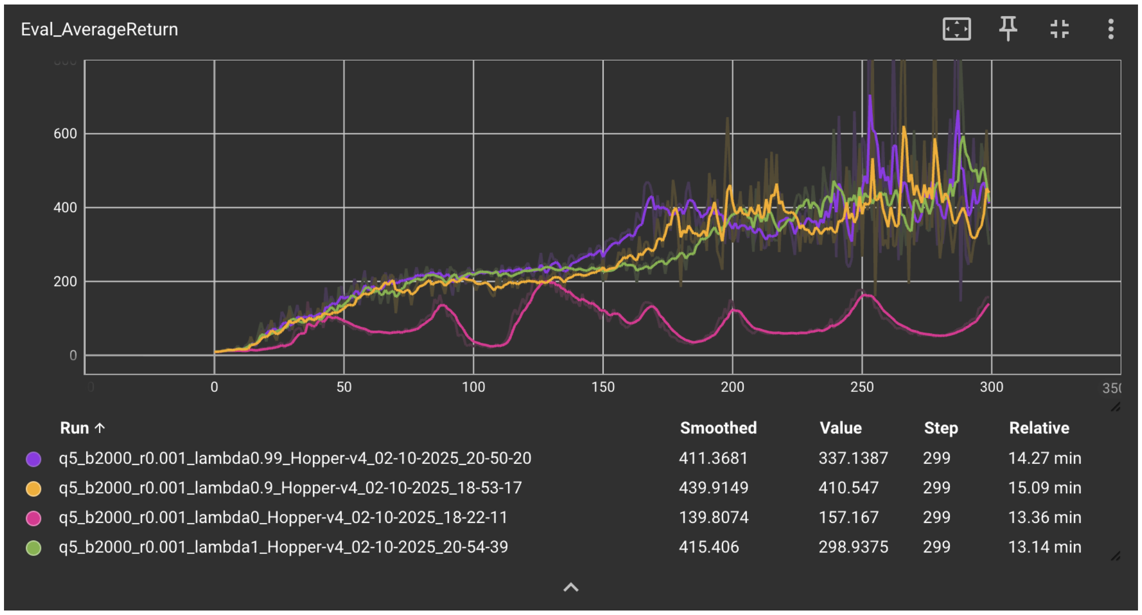
8.1.1 Configurations

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 0 --exp_name q5_b2000_r0.001_lambda0
```

8.1.2 Plot – [2 points]

Q8.1.2



Run ↑	Smoothed	Value	Step	Relative
● q5_b2000_r0.001_lambda0.99_Hopper-v4_02-10-2025_20-50-20	411.3681	337.1387	299	14.27 min
● q5_b2000_r0.001_lambda0.9_Hopper-v4_02-10-2025_18-53-17	439.9149	410.547	299	15.09 min
● q5_b2000_r0.001_lambda0_Hopper-v4_02-10-2025_18-22-11	139.8074	157.167	299	13.36 min
● q5_b2000_r0.001_lambda1_Hopper-v4_02-10-2025_20-54-39	415.406	298.9375	299	13.14 min

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

Q8.1.3

When the value of λ is 0, the model doesn't learn at all since we are not using the baseline. As the value of λ increases the model converges earlier and the performance is stable and better.

9 More Bonus!

9.1 Parallelization – [1.5 points]

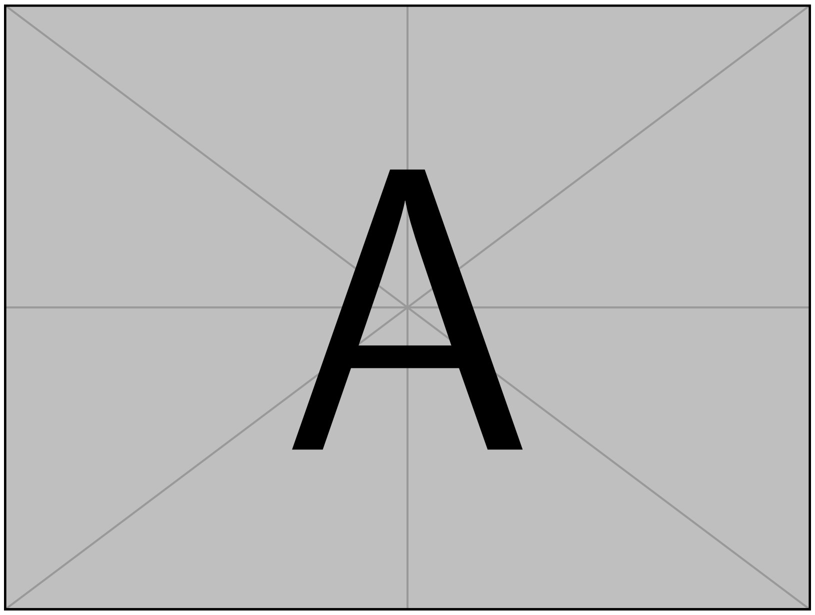
Q9.1

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 \
```