Assignment 2: Policy Gradient

Andrew ID: `kkabeer`
Collaborators: `asenathi`
NOTE: Please do **NOT** change the sizes of the answer blocks or plots.

5 Small-Scale Experiments

5.1 Experiment 1 (Cartpole) – [25 points total]

5.1.1 Configurations

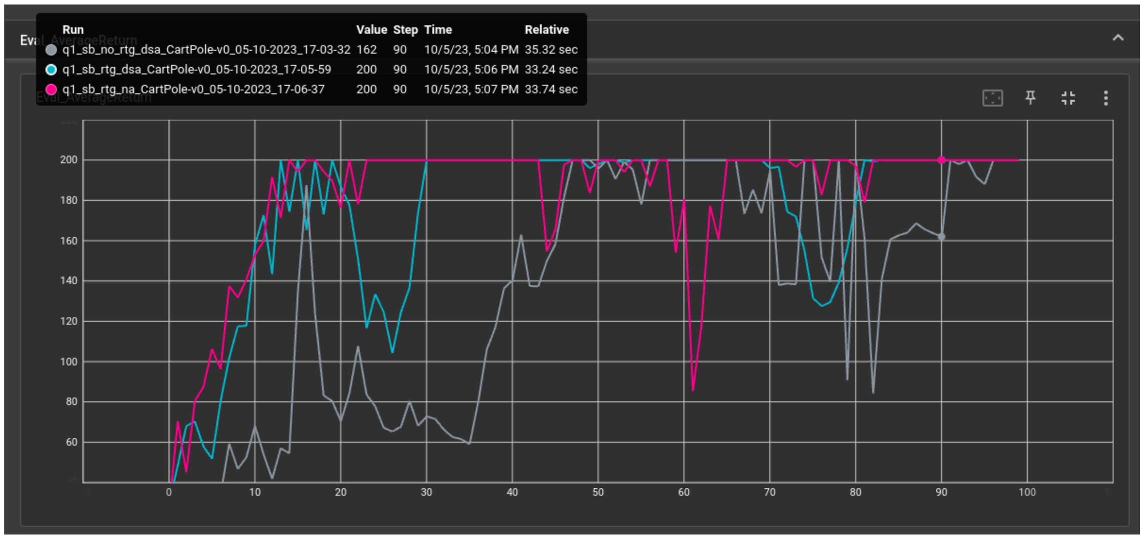
Q5.1.1

```
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 – [5 points]

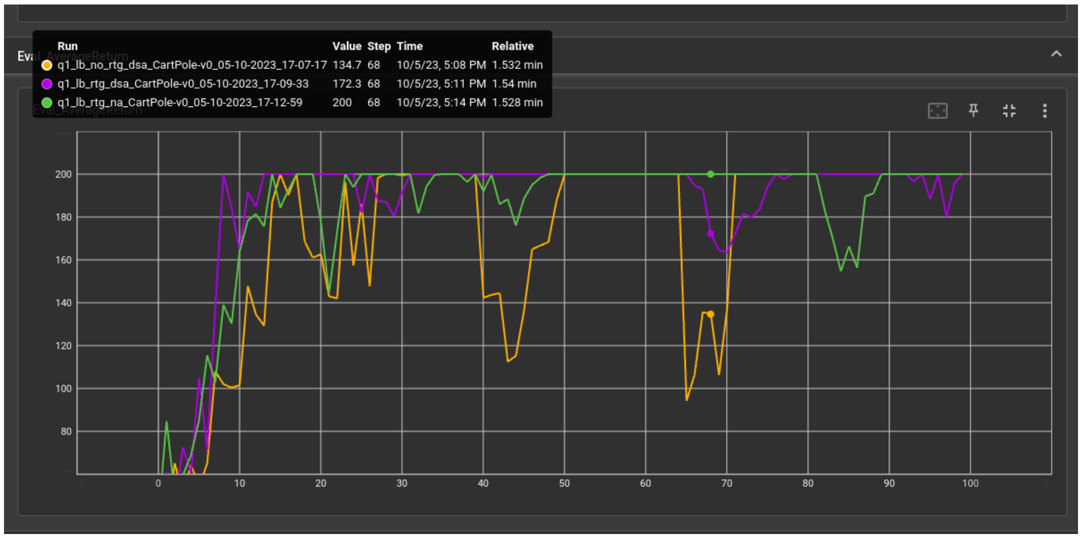
Q5.1.2.1



	Run	Value	Step	Time	Relative
●	q1_sb_no_rtg_dsa_CartPole-v0_05-10-2023_17-03-32	162	90	10/5/23, 5:04 PM	35.32 sec
●	q1_sb_rtg_dsa_CartPole-v0_05-10-2023_17-05-59	200	90	10/5/23, 5:06 PM	33.24 sec
●	q1_sb_rtg_na_CartPole-v0_05-10-2023_17-06-37	200	90	10/5/23, 5:07 PM	33.74 sec

5.1.2.2 Large batch – [5 points]

Q5.1.2.2



	Run	Value	Step	Time	Relative
🟧	q1_lb_no_rtg_dsa_CartPole-v0_05-10-2023_17-07-17	134.7	68	10/5/23, 5:08 PM	1.532 min
🟪	q1_lb_rtg_dsa_CartPole-v0_05-10-2023_17-09-33	172.3	68	10/5/23, 5:11 PM	1.54 min
🟩	q1_lb_rtg_na_CartPole-v0_05-10-2023_17-12-59	200	68	10/5/23, 5:14 PM	1.528 min

5.1.3 Analysis

5.1.3.1 Value estimator – [5 points]

Q5.1.3.1

According to the graphs above, the value estimator using reward-to-go (the aqua line in the small batch graph and the purple line in the large batch graph) had better performance than the value estimator using the entire trajectory (the grey line in the small batch graph and the yellow line in the large batch size graph).

5.1.3.2 Advantage standardization – [5 points]

Q5.1.3.2

Yes, advantage standardization did help, making the convergence much more sustained.

5.1.3.3 Batch size – [5 points]

Q5.1.3.1

Yes, the batch size had a huge effect, and the policy reached convergence more quickly for a larger batch size as compared to a smaller batch size.

5.2 Experiment 2 (InvertedPendulum) – [15 points total]

5.2.1 Configurations – [5 points]

Q5.2.1

```
python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 10000  -lr 0.01 -rtg --exp_name q2_b10k_r0.01

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 10000 -lr 0.05 -rtg --exp_name q2_b10k_r0.05

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 10000 -lr 0.1 -rtg --exp_name q2_b10k_r0.1

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 15000 -lr 0.01 -rtg --exp_name q2_b15k_r0.01

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 15000 -lr 0.05 -rtg --exp_name q2_b15k_r0.05

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 15000 -lr 0.1 -rtg --exp_name q2_b15k_r0.1

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 20000 -lr 0.01 -rtg --exp_name q2_b20k_r0.01

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 20000 -lr 0.05 -rtg --exp_name q2_b20k_r0.05

python rob831/scripts/run_hw2.py --env_name InvertedPendulum-v4 --ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 \
-b 20000 -lr 0.1 -rtg --exp_name q2_b20k_r0.1
```

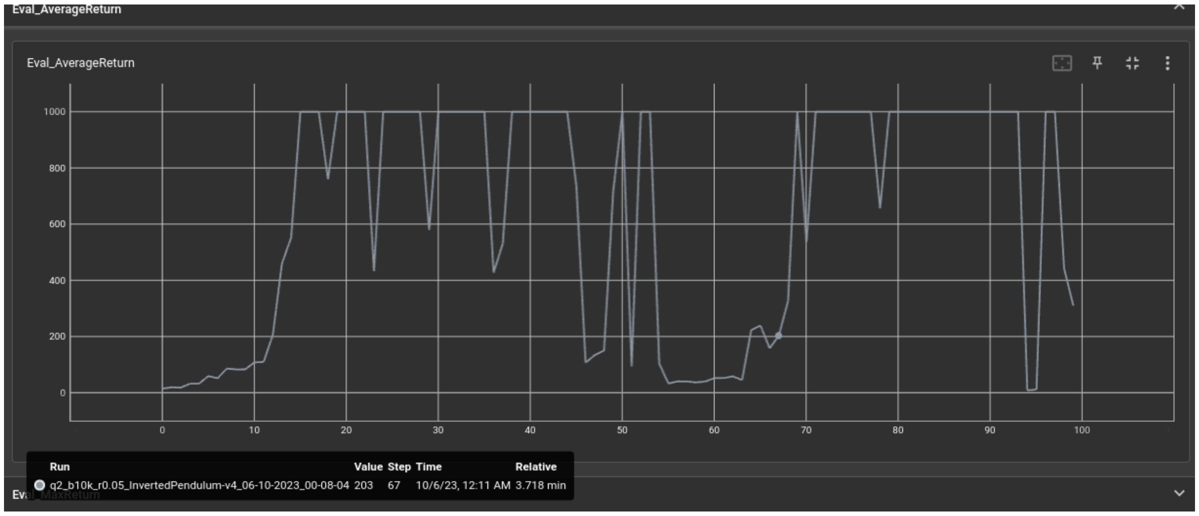
5.2.2 smallest b* and largest r* (same run) – [5 points]

Q5.2.2

The smallest batch size and largest learning rate which allowed the agent to get a reward of at least 1000 within 100 iterations was 10000 and 0.05 respectively. (All commands except the first one in the configurations was able to achieve this).

5.2.3 Plot – [5 points]

Q5.2.3



Run	Value	Step	Time	Relative
q2_b10k_r0.05_InvertedPendulum-v4_06-10-2023_00-08-04	203	67	10/6/23, 12:11 AM	3.718 min

7 More Complex Experiments

7.1 Experiment 3 (LunarLander) – [10 points total]

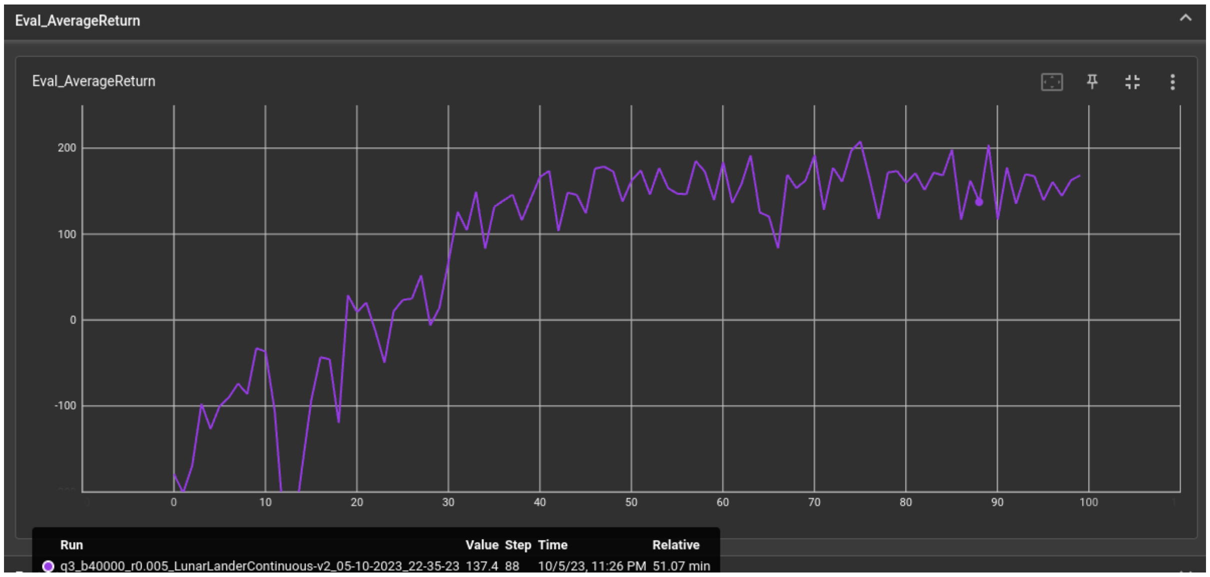
7.1.1 Configurations

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 40000 -lr 0.005 \
    --reward_to_go --nn_baseline --exp_name q3_b40000_r0.005
```

7.1.2 Plot – [10 points]

Q7.1.2



Run	Value	Step	Time	Relative
● q3_b40000_r0.005_LunarLanderContinuous-v2_05-10-2023_22-35-23	137.4	88	10/5/23, 11:26 PM	51.07 min

7.2 Experiment 4 (HalfCheetah) – [30 points]

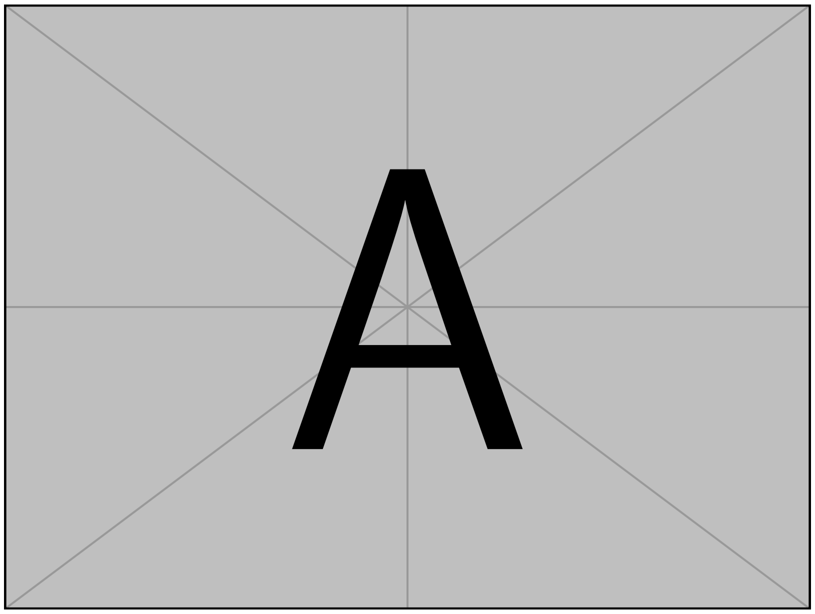
7.2.1 Configurations

Q7.2.1

```
# $b \in [10000, 30000, 50000], r \in [0.005, 0.01, 0.02]$
python rob831/scripts/run_hw2.py --env_name HalfCheetah-v4 --ep_len 150 \
    --discount 0.95 -n 100 -l 2 -s 32 -b <b> -lr <r> -rtg --nn_baseline \
    --exp_name q4_search_b<b>_lr<r>_rtg_nnbaseline
```

7.2.2 Plot – [10 points]

Q7.2.2



7.2.3 Optimal b* and r* – [3 points]

Q7.2.3

7.2.4 Describe how b* and r* affect task performance – [7 points]

Q7.2.4

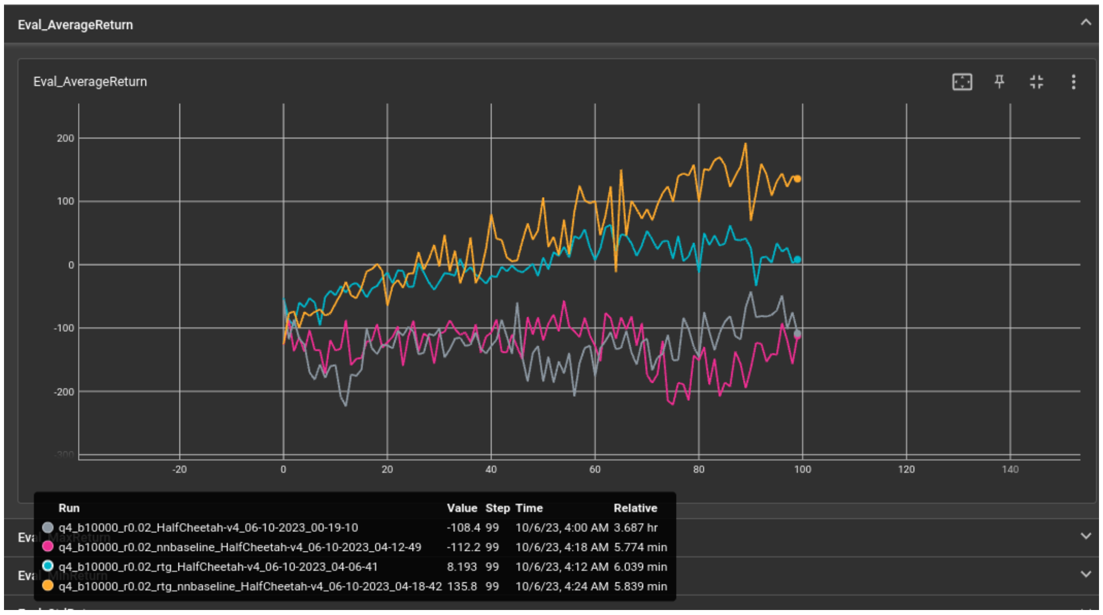
7.2.5 Configurations with optimal b* and r* – [3 points]

Q7.2.5

```
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_b10000_r0.02 --no_gpu

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_b10000_r0.02_rtg --no_gpu

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_b10000_r0.02_nnbaseline --no_gpu

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_b10000_r0.02_rtg_nnbaseline --no_gpu
```

7.2.6 Plot for four runs with optimal b* and r* – [7 points]

Q7.2.6



	Run	Value	Step	Time	Relative
●	q4_b10000_r0.02_HalfCheetah-v4_06-10-2023_00-19-10	-108.4	99	10/6/23, 4:00 AM	3.687 hr
●	q4_b10000_r0.02_nnbaseline_HalfCheetah-v4_06-10-2023_04-12-49	-112.2	99	10/6/23, 4:18 AM	5.774 min
●	q4_b10000_r0.02_rtg_HalfCheetah-v4_06-10-2023_04-06-41	8.193	99	10/6/23, 4:12 AM	6.039 min
●	q4_b10000_r0.02_rtg_nnbaseline_HalfCheetah-v4_06-10-2023_04-18-42	135.8	99	10/6/23, 4:24 AM	5.839 min

8 Implementing Generalized Advantage Estimation

8.1 Experiment 5 (Hopper) – [20 points]

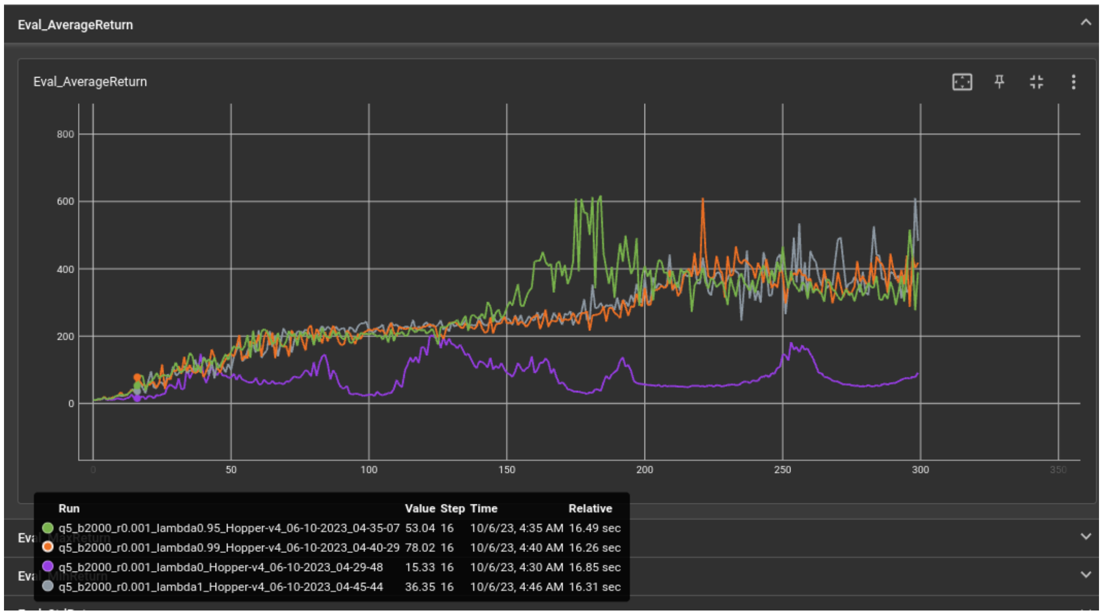
8.1.1 Configurations

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 – [13 points]

Q8.1.2



	Run	Value	Step	Time	Relative
🟢	q5_b2000_r0.001_lambda0.95_Hopper-v4_06-10-2023_04-35-07	53.04	16	10/6/23, 4:35 AM	16.49 sec
🟠	q5_b2000_r0.001_lambda0.99_Hopper-v4_06-10-2023_04-40-29	78.02	16	10/6/23, 4:40 AM	16.26 sec
🟣	q5_b2000_r0.001_lambda0_Hopper-v4_06-10-2023_04-29-48	15.33	16	10/6/23, 4:30 AM	16.85 sec
⚪	q5_b2000_r0.001_lambda1_Hopper-v4_06-10-2023_04-45-44	36.35	16	10/6/23, 4:46 AM	16.31 sec

8.1.3 Describe how λ affects task performance – [7 points]

Q8.1.3

As expected, $\lambda = 0$ gives the worst performance, since $A^{\pi}_{GAE} = A^{\pi}_{1}$ in this case. The best value of λ is 0.95 (the green line), which allows the network to reach rewards as high as 600 during the training. It is clear that the higher learning rates (0.99 and 1) skips over a really good minima at around the 150th iteration mark, whereas $\lambda = 0.95$ is able to get into that minima, allowing it to reach rewards of about 600.

9 Bonus! (optional)

9.1 Parallelization – [15 points]

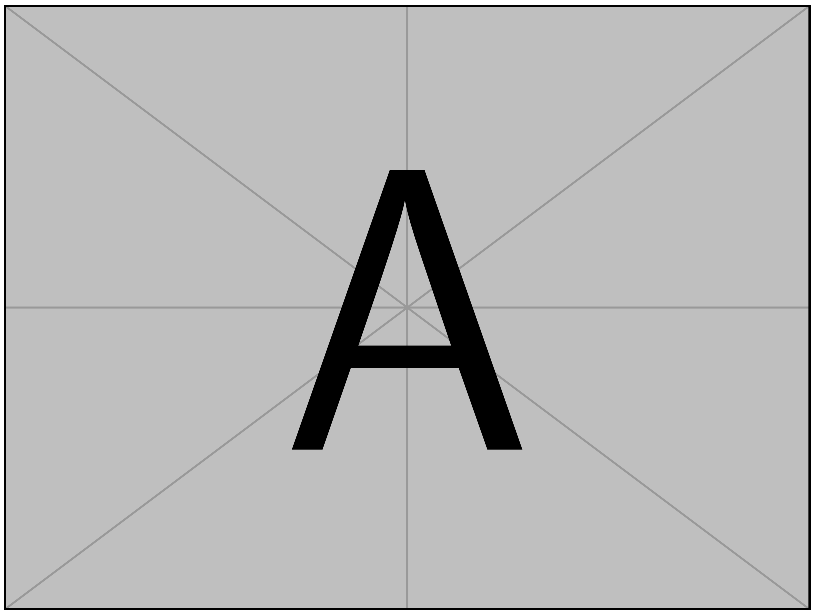
Q9.1

Difference in training time:

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

9.2 Multiple gradient steps – [5 points]

Q9.1



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