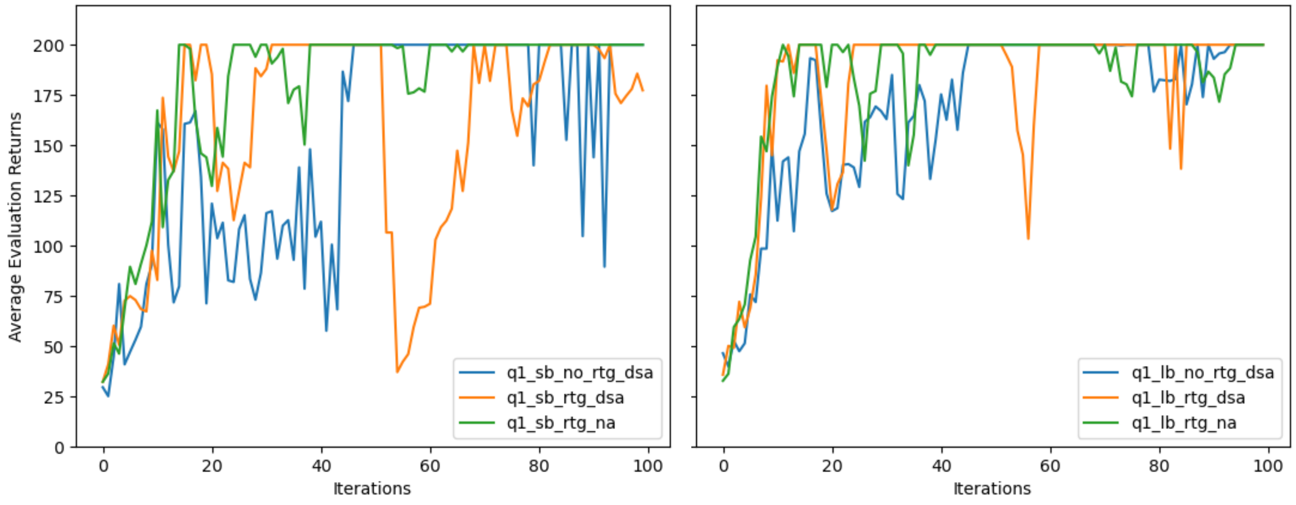
CS 285 HW 2

Exp 1



1. Which value estimator has better performance without advantage-standardization: the trajectory-centric one, or the one using reward-to-go?
   - Answer: The reward-to-go one has better performance, as most obviously shown in the right plot with a larger batch.

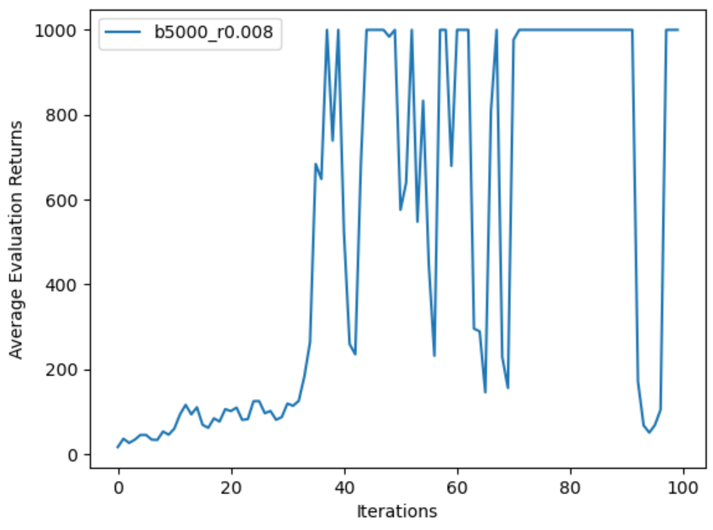
2. Did advantage standardization help?
   - Answer: No. It makes the training worse.

3. Did the batch size make an impact?
   - Answer: Yes. Large batch size gives more stable training and faster convergence.

Command line for Exp. 1:

```
python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
-dsa --exp_name q1_sb_no_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
-rtg -dsa --exp_name q1_sb_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 1000 \
-rtg --exp_name q1_sb_rtg_na

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
-dsa --exp_name q1_lb_no_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
-rtg -dsa --exp_name q1_lb_rtg_dsa

python cs285/scripts/run_hw2.py --env_name CartPole-v0 -n 100 -b 5000 \
-rtg --exp_name q1_lb_rtg_na
```

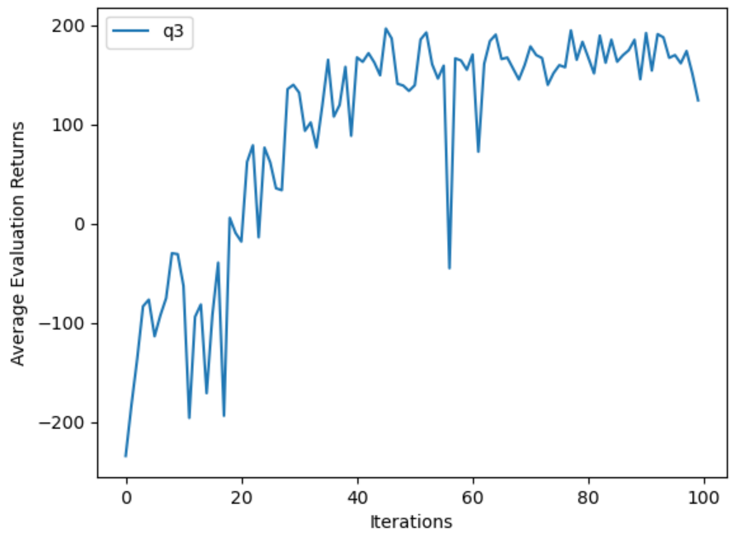
Exp 2



Command line for Exp. 2:

```
python cs285/scripts/run_hw2.py --env_name InvertedPendulum-v2 \
--ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b 5000 -lr 0.008 -rtg \
--exp_name q2_b5000_r0.008
```

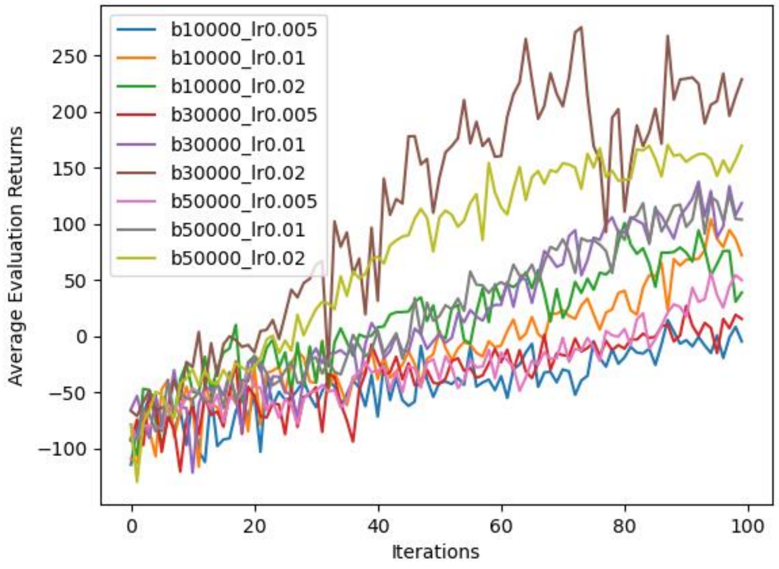
Exp 3



Command line for Exp. 3:

```
python cs285/scripts/run_hw2.py --env_name InvertedPendulum-v2 \
--ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b 5000 -lr 0.008 -rtg \
--exp_name q2_b5000_r0.008
```

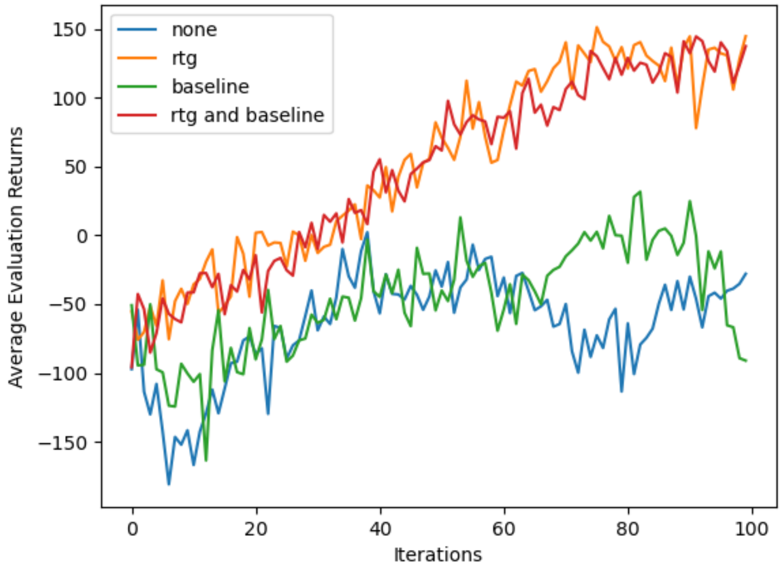
Exp 4



- Describe in words how the batch size and learning rate affected task performance.

  Answer: Larger batch size generally gives better return than smaller batch size, for example, both 30000 and 50000 batch outperform the 10000 batch size. A greater learning rate also gives better return, for example, with batch size 30000 batch, lr=0.02 outperforms lr=0.01 and etc.

As can be readily seen, the best batch size is 30000 and the best learning rate is 0.02. Plug them in for the remaining 4 runs:


none
rtg
baseline
rtg and baseline
Average Evaluation Returns
Iterations