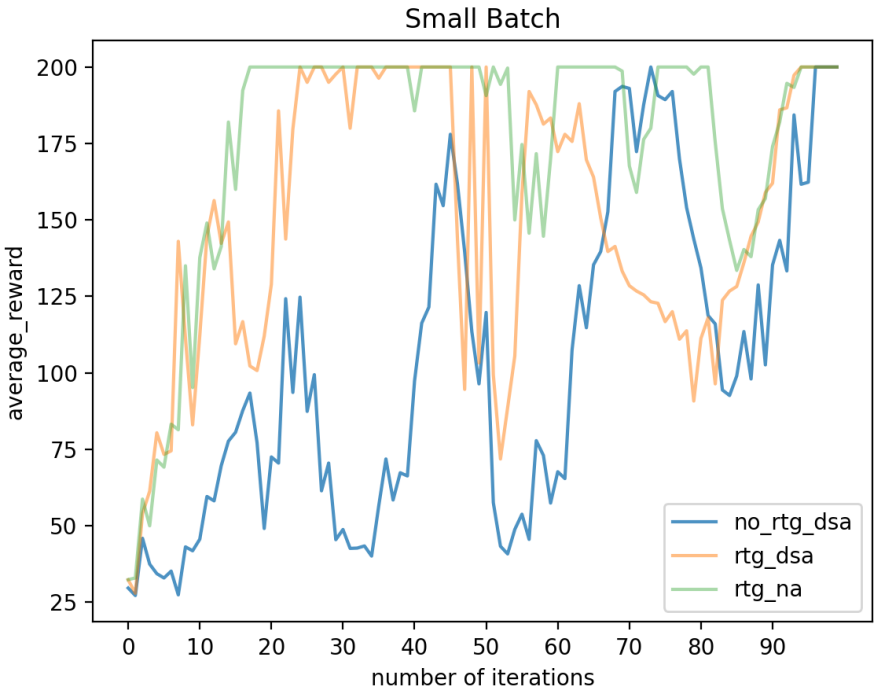
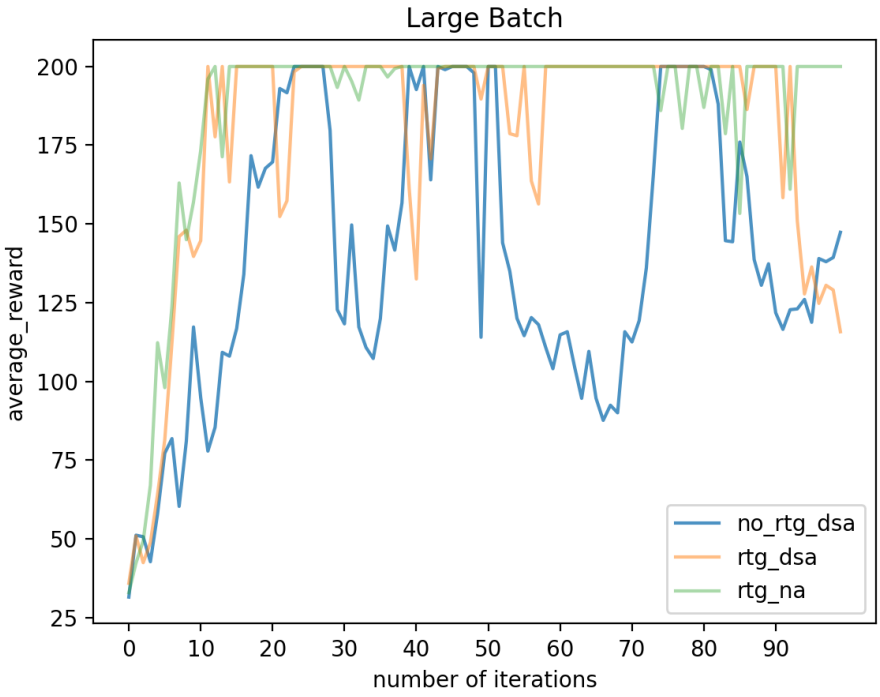
Exp1:
Small-scale:
Learning_rate_0.005, n_layers:2, size: 64



Small Batch Result



Large Batch Result

Questions:

1. Which value estimator has better performance without advantage-standardization: the trajectory-centric one, or the one using reward-to-go?

   Value estimator has better performance by the one using reward-to-go. Training procedure has less variance times and faster to converge.

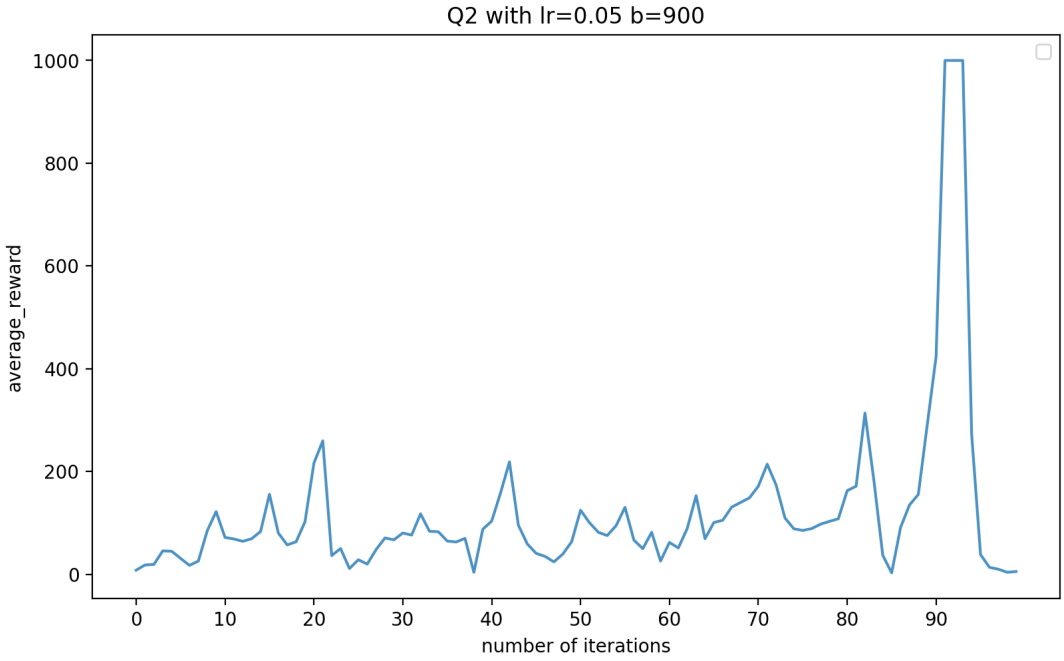
2. Did advantage standardization help?

   Yes. Advantage standardization helps the estimator being more stable. Once the estimator reaches to the maximum, has lower variance to go back and forth.

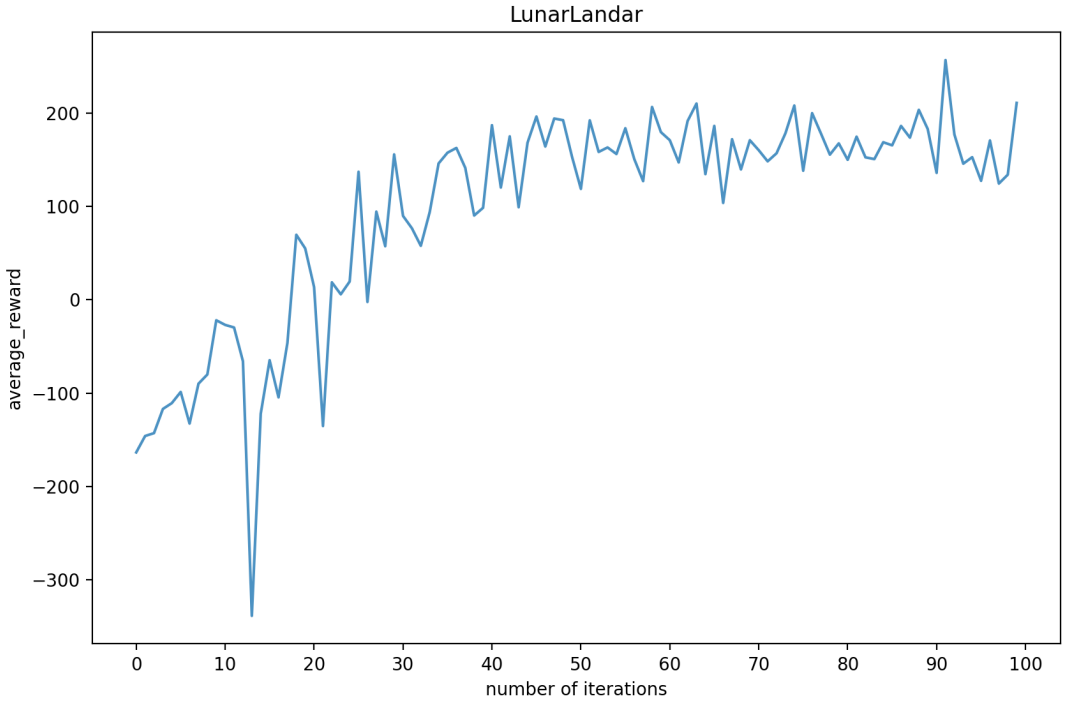
3. Did the batch size make an impact?

   Observed from those two figures, seems small batches are more difficult to train to converge. For all three different conditions.

Exp2:
Min_batch_size: 900
Max_learning_rate: 0.05
Command: python cs285/scripts/run_hw2.py --env_name InvertedPendulum-v2 \
--ep_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b <900> -lr <0.05> -rtg \
--exp_name q2_b<900>_r<0.05>_50_iter



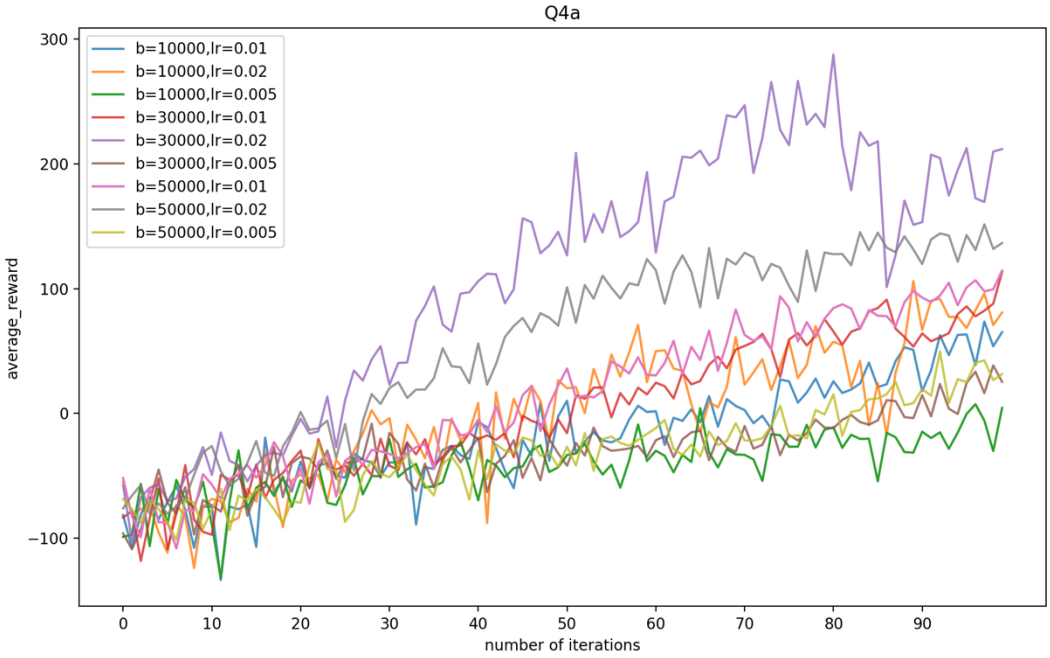
Exp3:
Command: python cs285/scripts/run_hw2.py \
--env_name LunarLanderContinuous-v2 --ep_len 1000
--discount 0.99 -n 100 -l 2 -s 64 -b 40000 -lr 0.005 \
--reward_to_go --nn_baseline --exp_name q3_b40000_r0.005



Exp4:
For 4a)
As the picture shows the best performance happened where b = 30000 and lr = 0.02, the purple line.

So for learning rate, small learning rate performs worse in this experiment, lr = 0.005 is the worst regardless of the batch size.

And for batch_size, there is no guarantee that for larger batch size the performance will be better, but generally speaking for this experiment, batch size with 30000 and 50000 will performs better than 10000



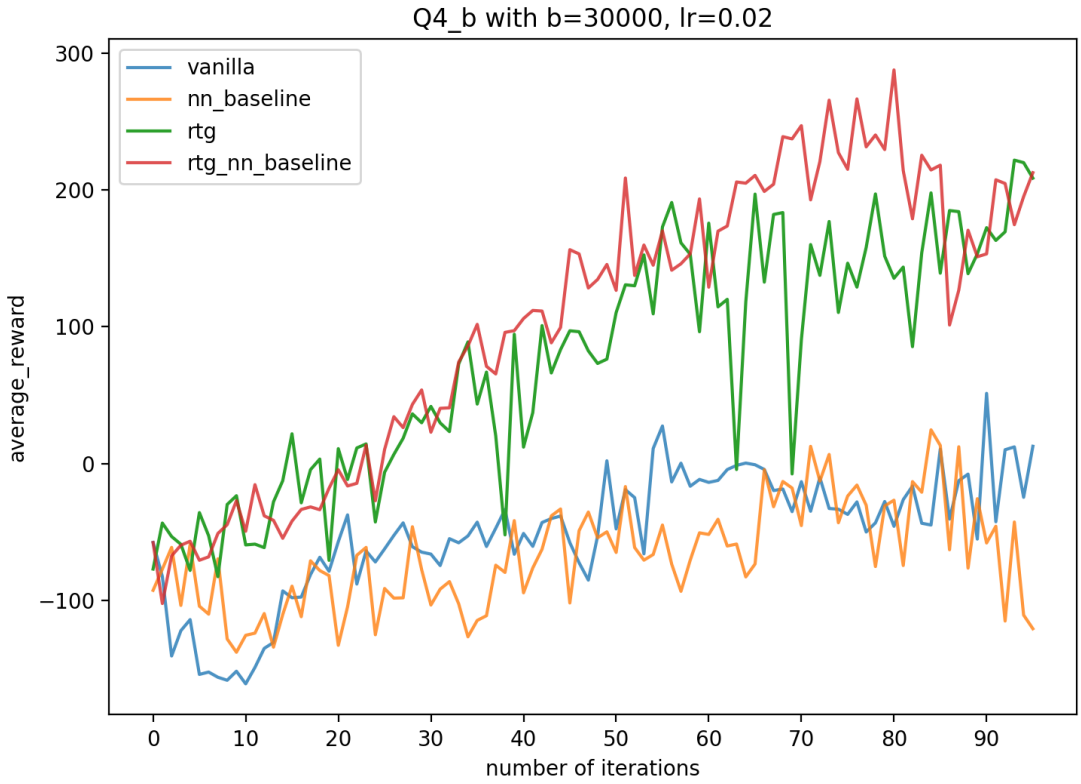


Figure for q4_b

Exp5:

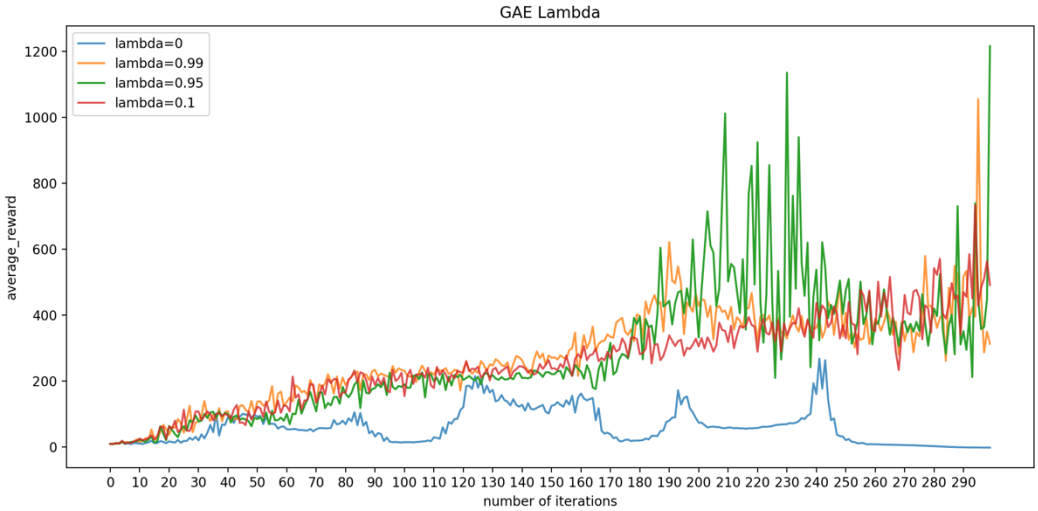


Figure with lambda setting

When lambda = 0.95, the network training process has more variance than using lambda=0.99/1. When lambda = 0, which leads the A_pi(st+1,at+1) times 0; so at that point the network are only using the current step of reward and estimator. And the result is not good.