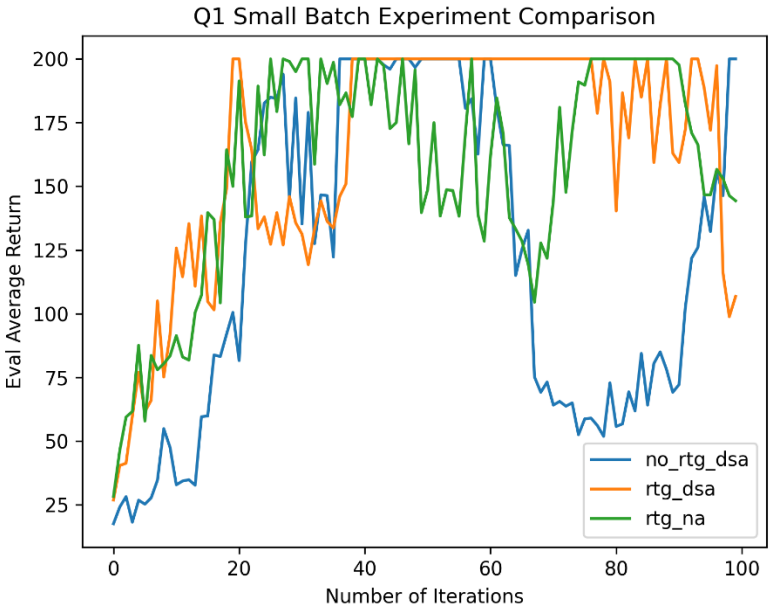
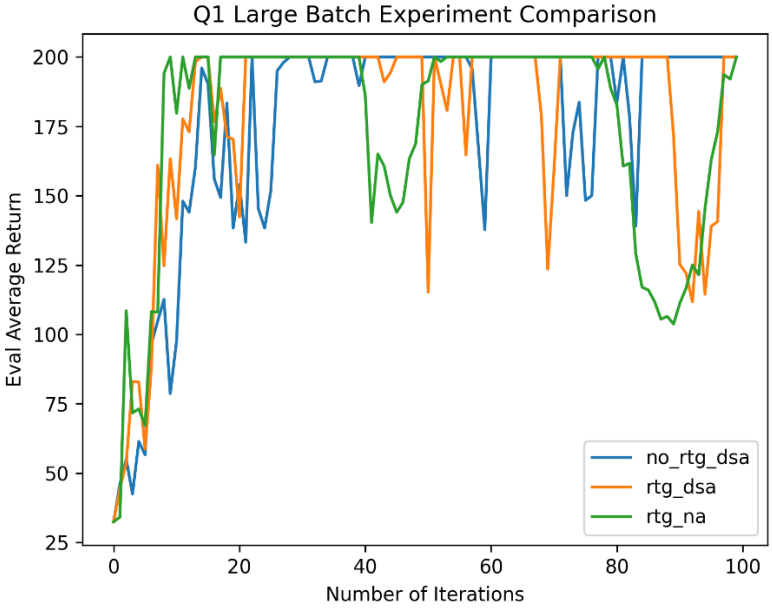
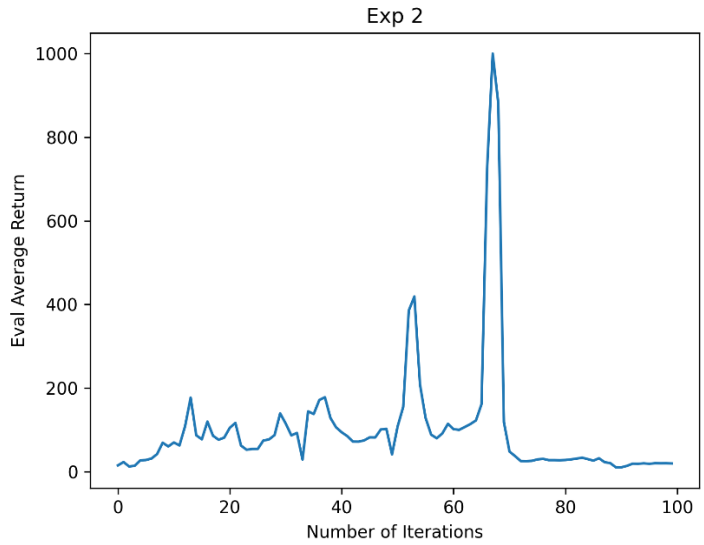
Experiment 1.





1. Using reward-to-go is better than the trajectory centric one.
2. Based on the plots, using advantage standardization seems to help provide a more stable return.
3. The large batch size experiment had quicker convergence compared to the small batch ones.

Experiment 2.



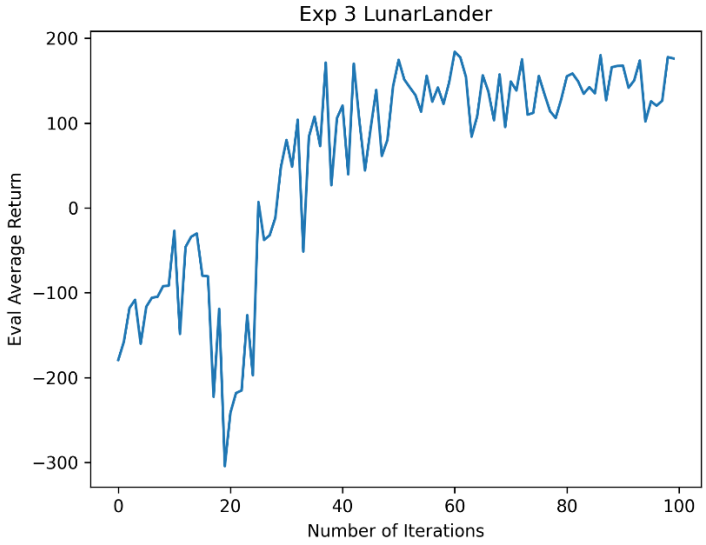
Command line: python cs285/scripts/run_hw2.py --env_name InvertedPendulum-v4 \

--ep_len 1000 --discount 0.9 \

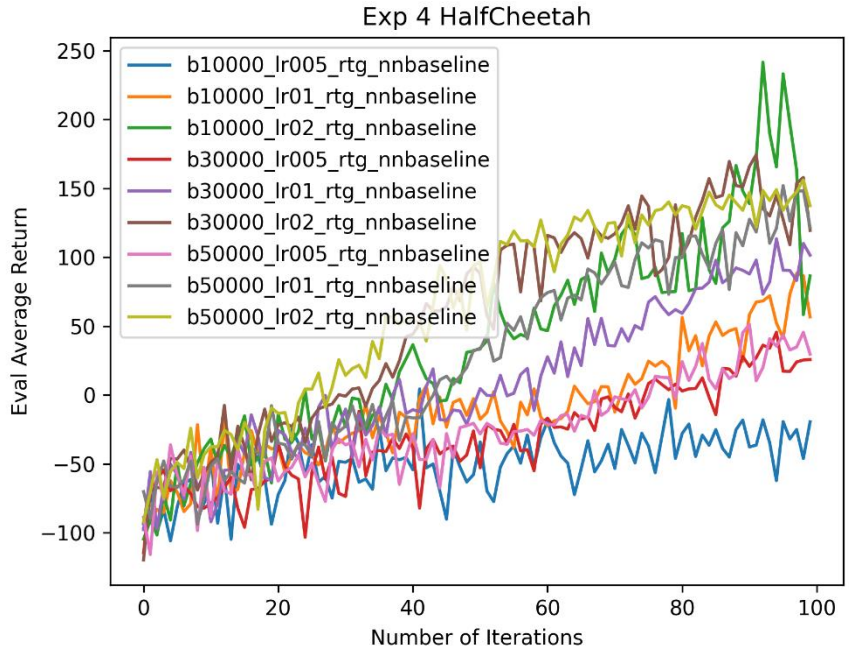
-n 100 -l 2 -s 64 -b 400 -lr 0.03 -rtg \

--exp_name q2_b400_r0.03

Experiment 3.



Experiment 4 part 1.

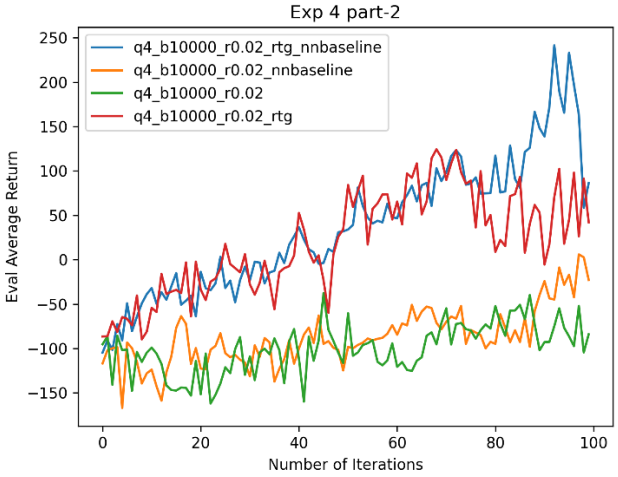


As learning rate increases, the task is also learned at a faster rate (learning curve slope is steeper).

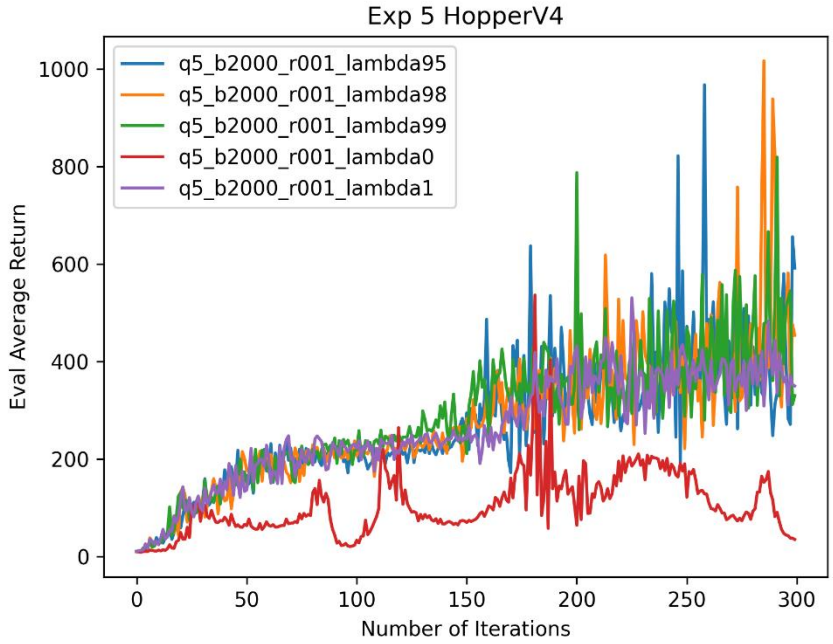
As the batch size increases, the performance of the task increases generally, but it is noticeable that batch size 10000 surprisingly outperforms in the end. Also, it could be that a smaller batch size allows the agent to learn a better policy without overfitting. Meanwhile, a smaller batch size is trained significantly faster than batch size of 50000.

The optimal pick of b and r: **b* = 10000, r* = 0.02**

Experiment 4 part 2.



Experiment 5.



As we know that increasing λ decreases bias and increases variance, when λ is 0, the network seems to have high bias and it's not properly predicting actions with high returns. On the other hand, when we increase λ to around 0.98, the network is performing well and achieves 400+ avg. return, but it's having high variances on its performance.