1. Experiment 1 (CartPole, discrete control):

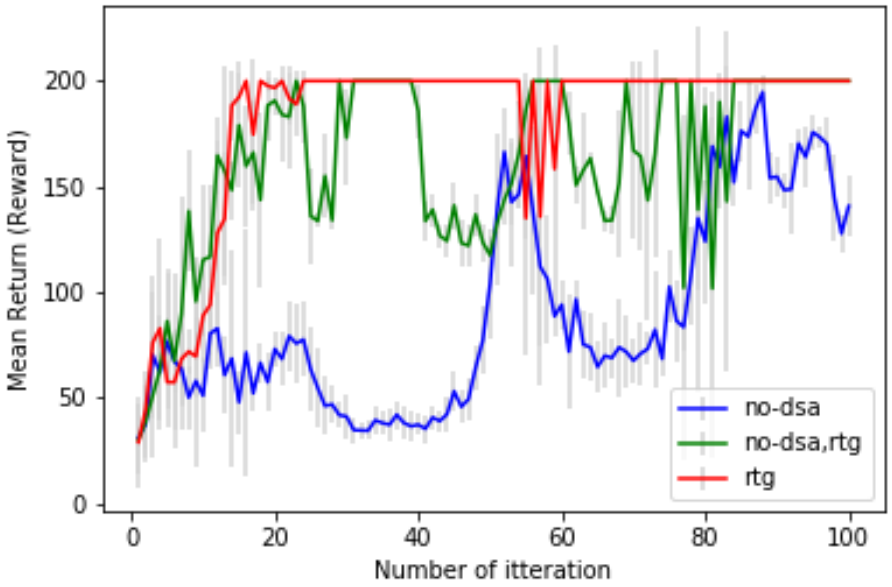

Fig 1. The learning curve for small batch size (1000), comparing advantages of having standardize_advantages and reward-to-go

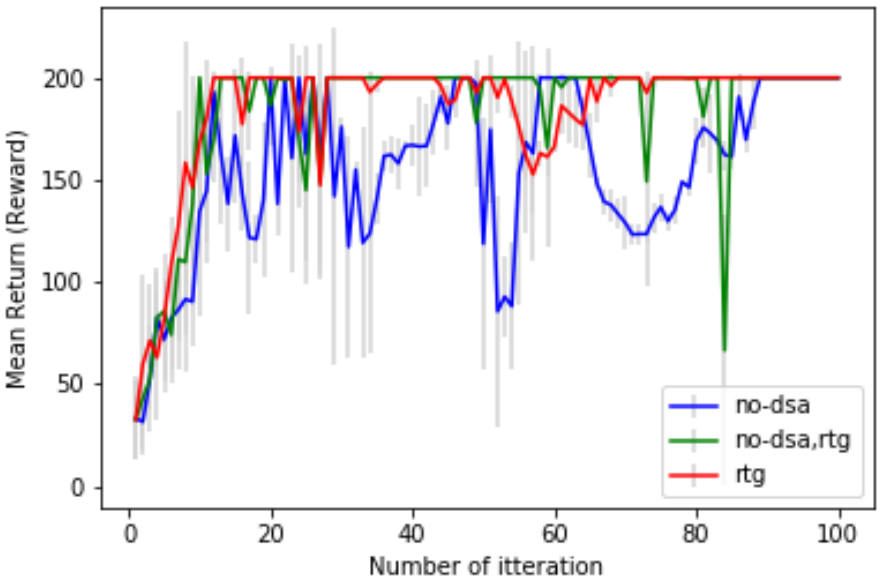

Fig 2. The learning curve for large batch size (5000), comparing advantages of having standardize_advantages and reward-to-go

Questions:

- Without advantage-standardization, reward-to-go has better performance
- The advantage-standardization has important impact on small-batch but little on large-batch
- The larger batch size reaches the max reward faster than the smaller batch size

2. Experiment 2: (Inverted Pendulum, continuous control)

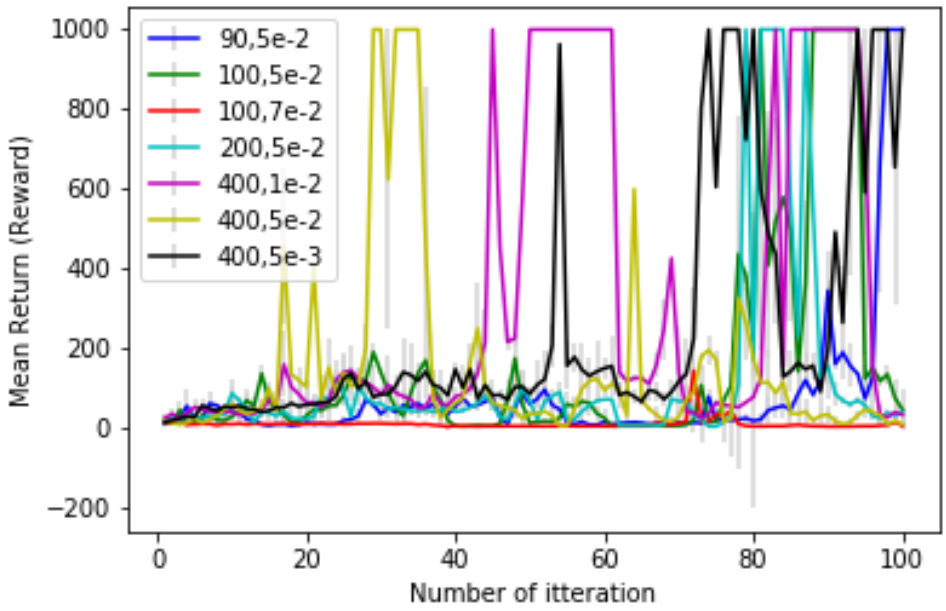

Fig. 3: Reward vs iteration for different sets of batch-size and learning rate. The smallest batch-size and largest learning rate to reach the goal in less than 100 iterations are 90 and 5e-2, respectively.

3. Experiment 3 (LunarLander): Policy gradient with baseline

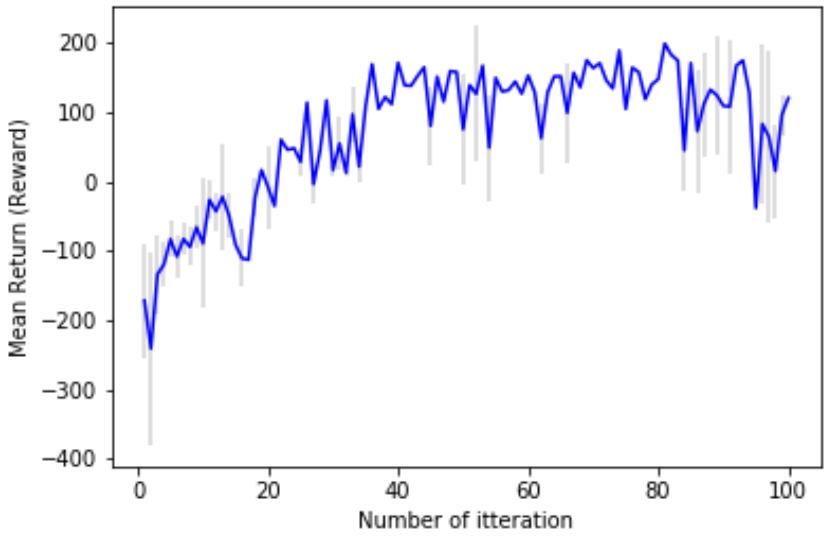

Fig. 4: Mean return vs number of iterations of the Lunar Lander environment

4. Experiment 4 (HalfCheetah)

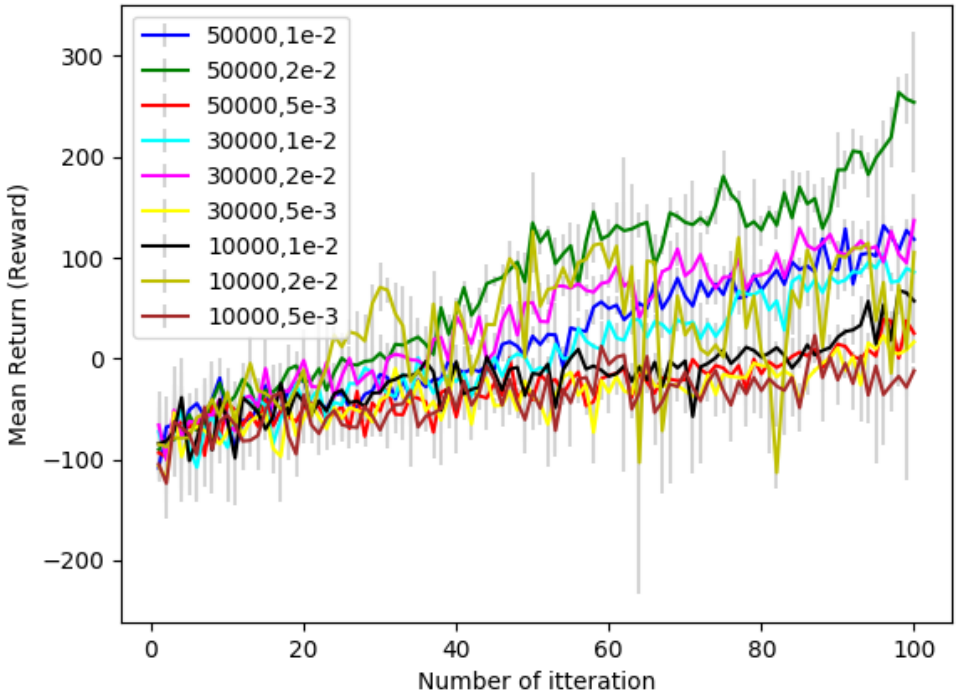

Fig. 5: Reward value vs number of iterations for the HalfCheetah environment. Increasing both the batch size and learning rate helped to reach the max reward in 100 iterations. The best set for batch size and the learning rate is 50000 and 0.02

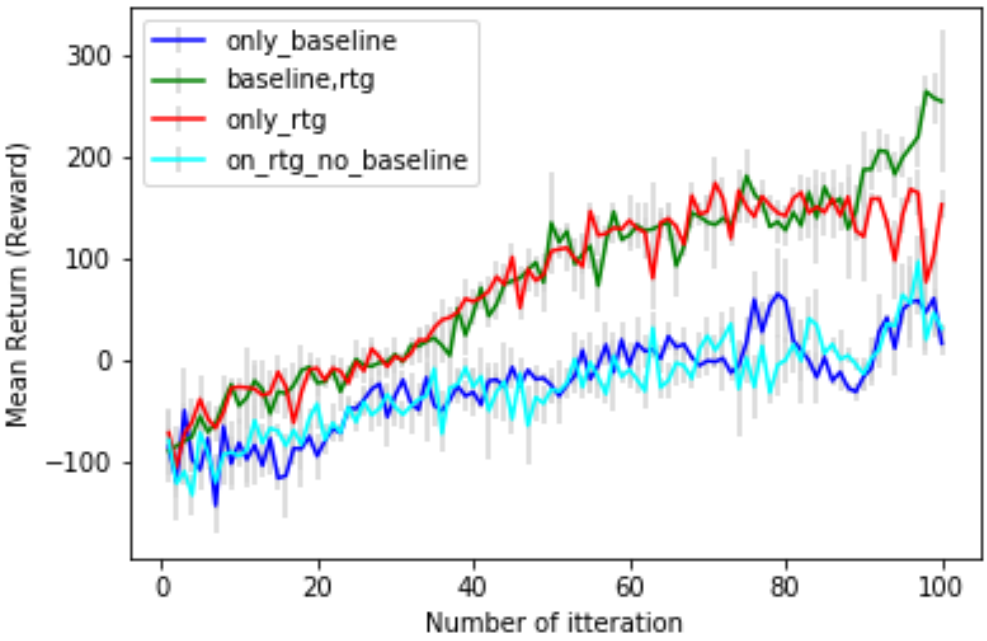

Fig. 6: Mean return vs number of iterations for HalfCheetah env. with batch-size=50000 and learning rate=0.02. The reward-to-go and advantage-standardization improve the performance.

5. Experiment 5: (Hopper V2, generalized advantage estimate)

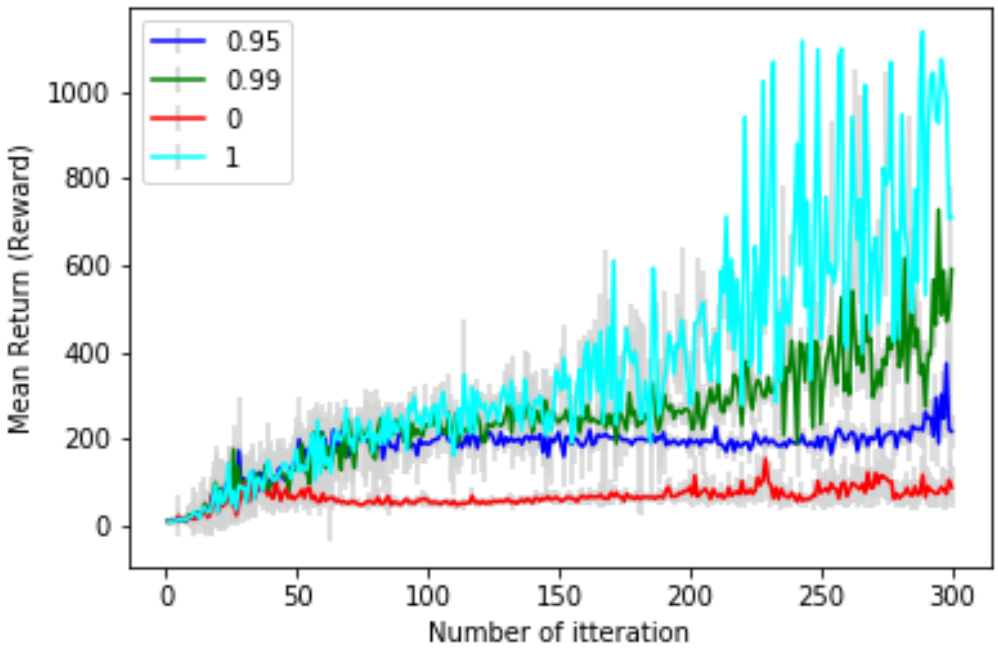

Fig. 7: Performance of the GAE with Lambda: [0, 0.95, 0.99, 1]. Larger lambda has a higher reward. However, lambda equal to 0.99 reached the goal reward with less fluctuation.