CS 285 Homework 4

Jeffrey Cheng

November 2, 2022

Commands used to run each question can be found in the README.md of the submission folder.

Question 1

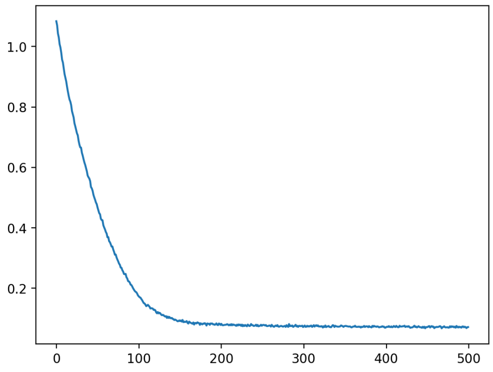

Figure 1: Dynamic Model Losses, 1 Network Layer, 32 Nodes per Layer, 500 Training Steps

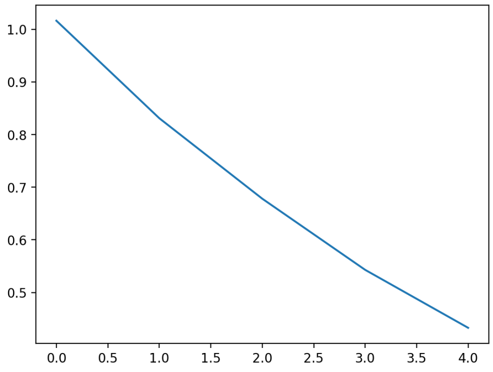

Figure 2: Dynamic Model Losses, 2 Network Layers, 250 Nodes per Layer, 5 Training Steps

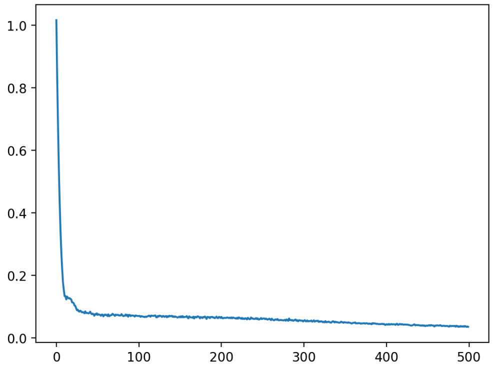

Figure 3: Dynamic Model Losses, 2 Network Layers, 250 Nodes per Layer, 500 Training Steps

The models with more numerous training steps taken perform better, as the networks are updated more. The network with more layers and nodes reaches low losses more quickly, perhaps due to the ability for the more complex network to more quickly learn complex dynamics.

Question 2

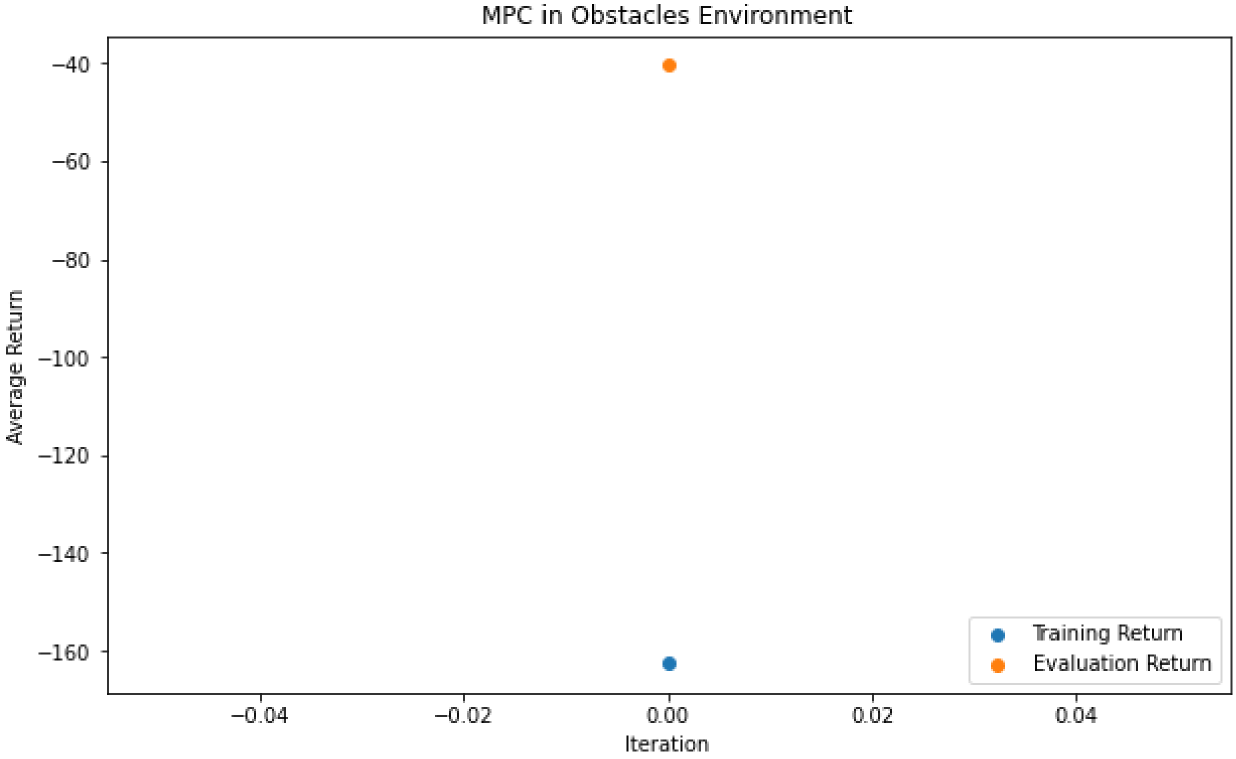

Figure 4: Single iteration returns of model predictive control in the obstacles environment.

Question 3

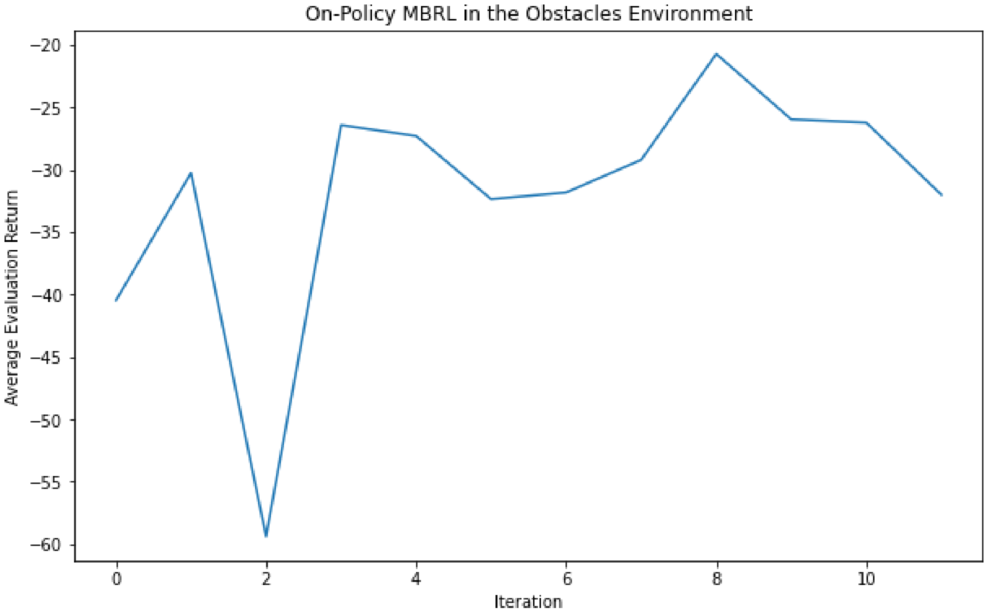

Figure 5: Model predictive control learning curve in the obstacles environment.

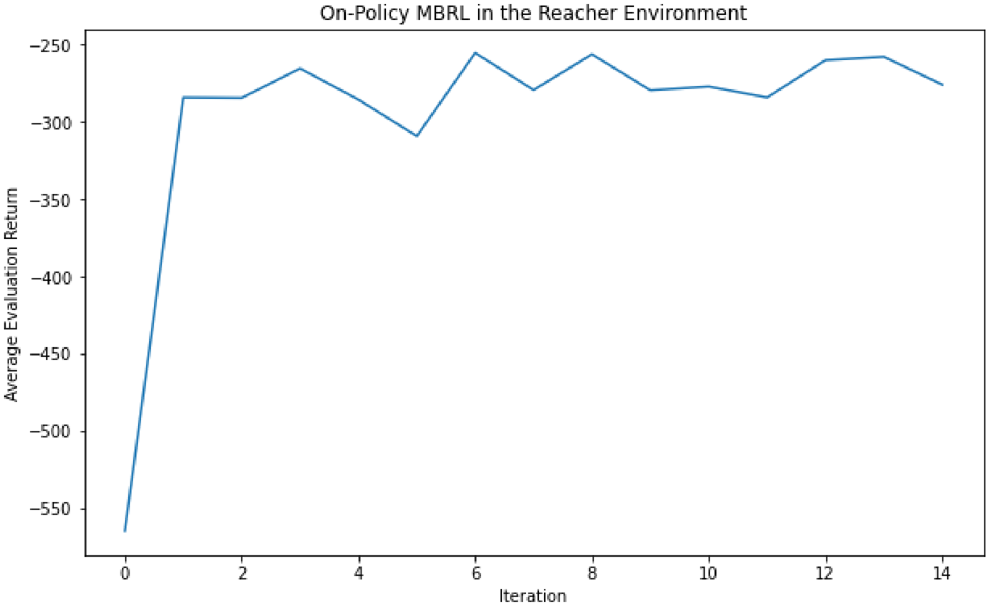

Figure 6: Model predictive control learning curve in the reacher environment.

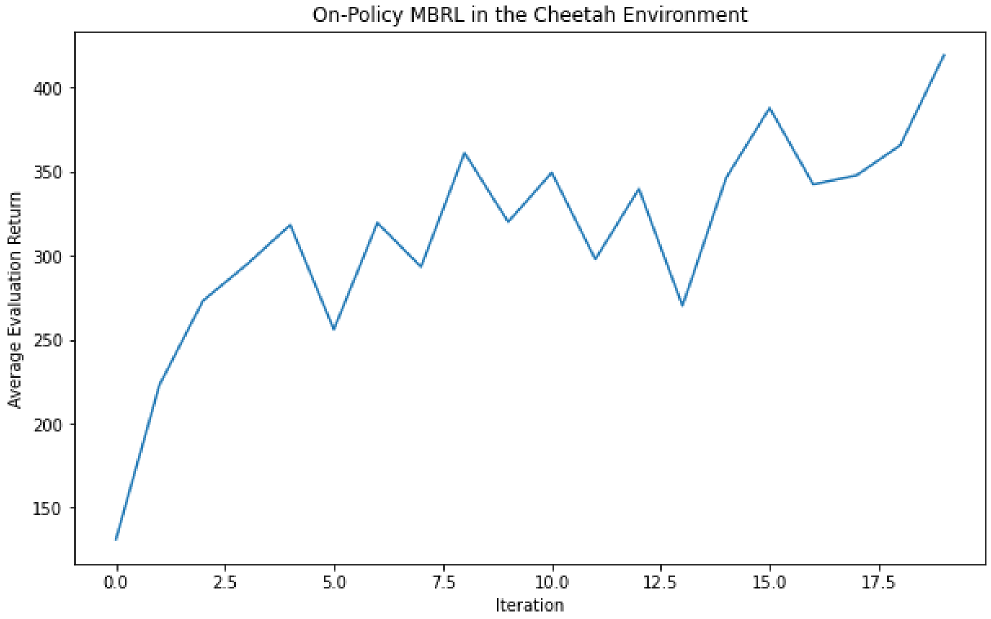

Figure 7: Model predictive control learning curve in the cheetah environment.

Question 4

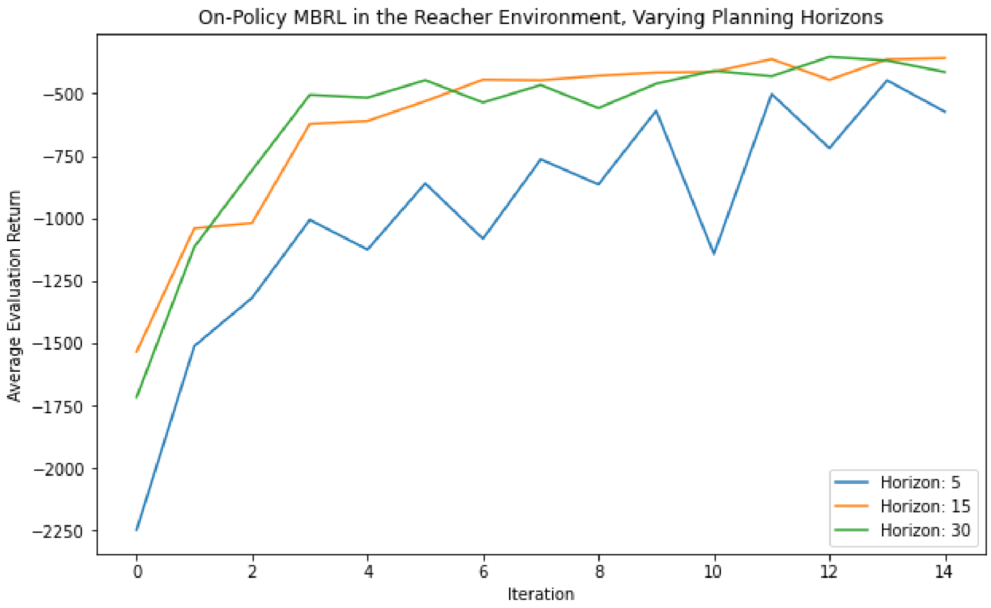

Figure 8: Model predictive control learning curves in the reacher environment with varying planning horizons. We see that longer planning horizons tend to be more stable and see better rewards at the expense of being more computationally costly.

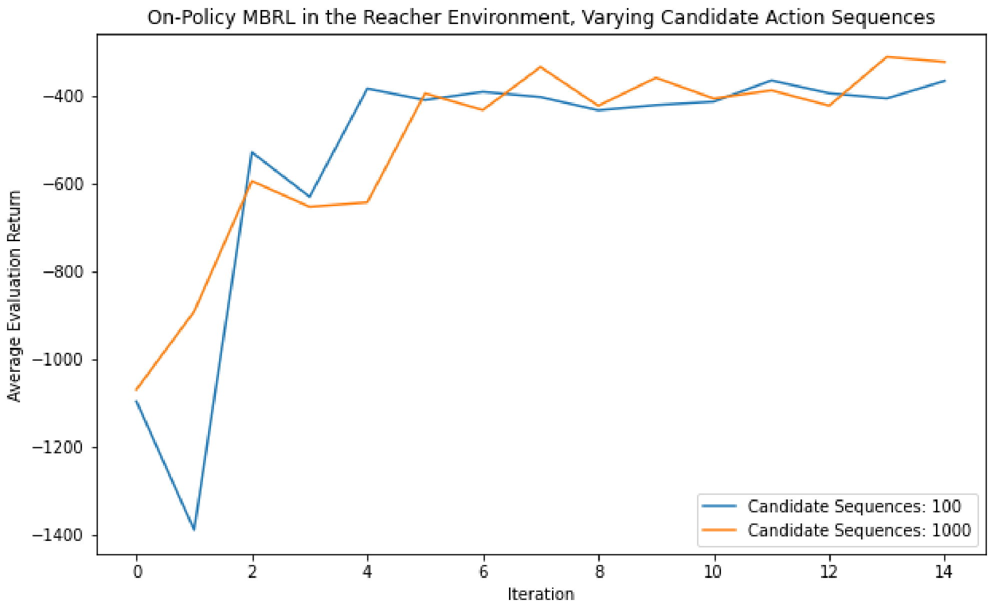

Figure 9: Model predictive control learning curves in the reacher environment with varying number of candidate action sequences. We see that the algorithm with larger candidate action sequences seems to have more stable initial learning, but higher overall variance.

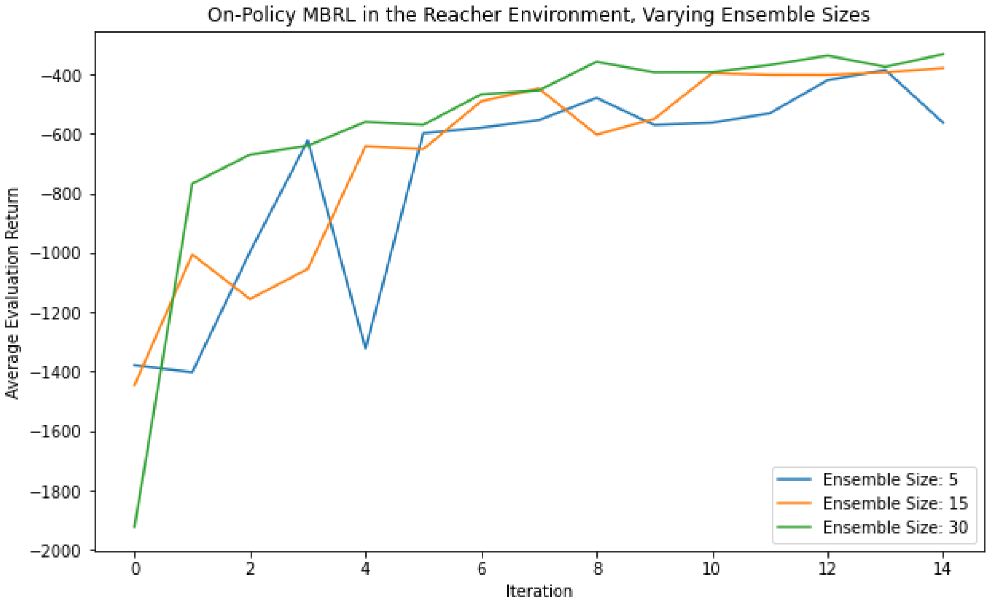

Figure 10: Model predictive control learning curves in the reacher environment with varying number of ensemble sizes. We see that larger ensemble sizes are more stable and have less variance.

Question 5: Run actor-critic with more difficult tasks

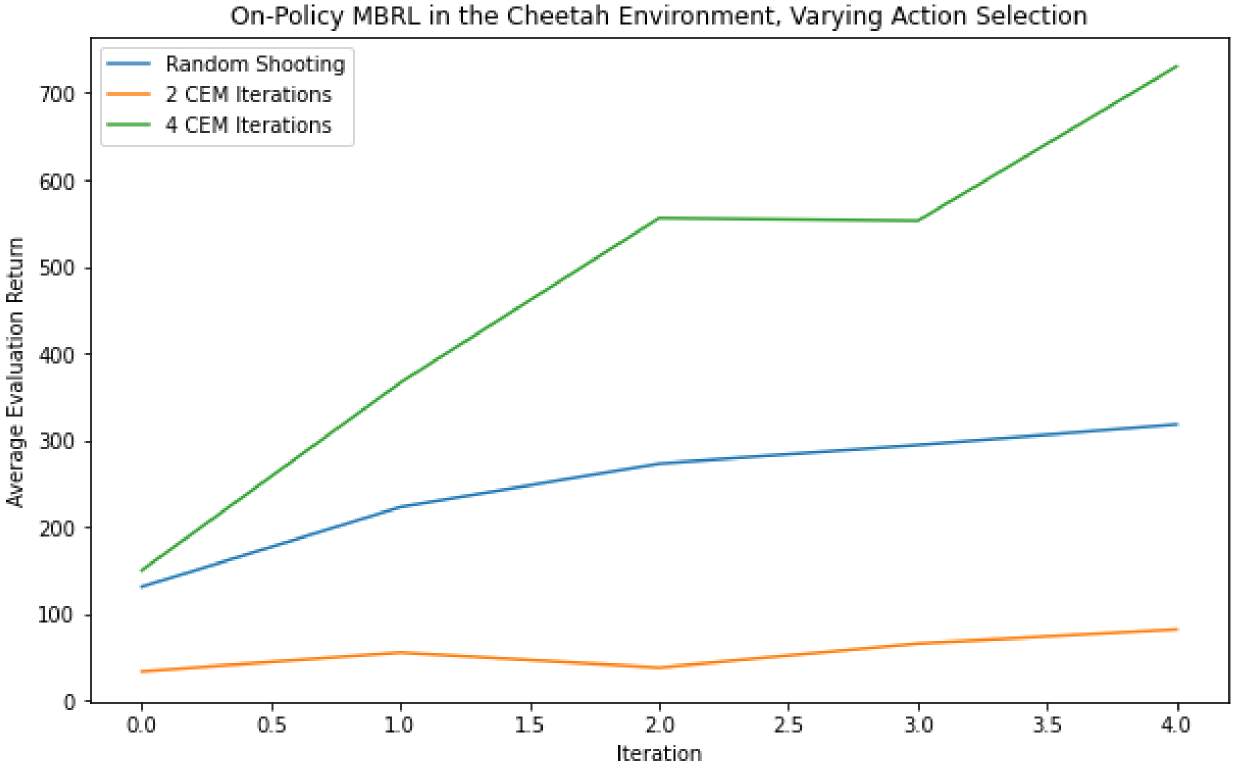

Figure 11: Model predictive control learning curves in the cheetah environment with varying action selection methods. We see that more cross entropy selection of actions leads to increased learning. Interestingly, 2 iterations of cross entropy action selection performs worse than random shooting. More iterations is more computationally expensive though.

Question 6

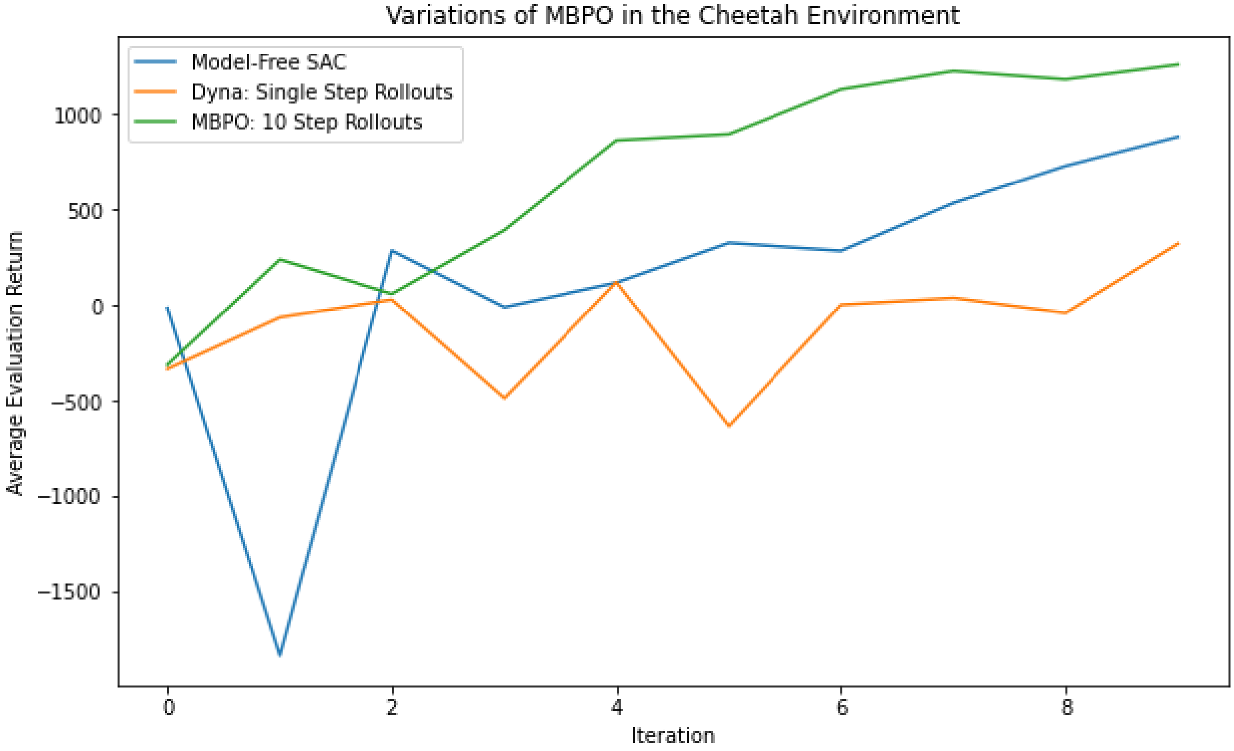

Figure 12: Variations of model based policy optimization learning curves in the cheetah environment. We see that larger rollout steps lead to increased learning. Model-free SAC has high initial variance as seen previously. Single step rollouts do not perform well, but are less computationally expensive.