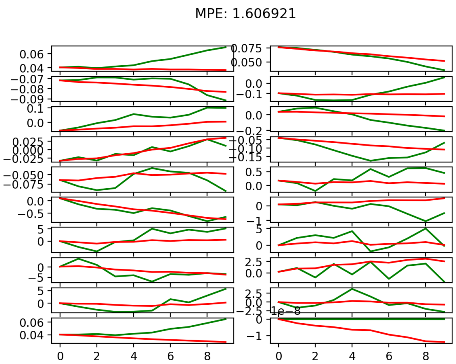
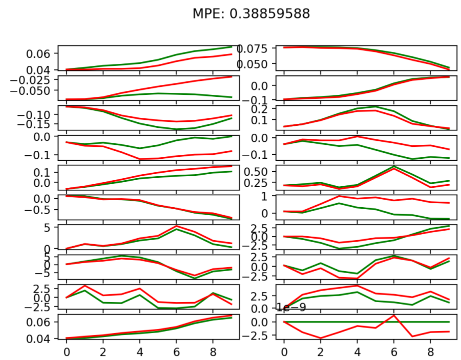
CS 285 Homework 4: Model-based RL
November 4, 2021
Gautam Mittal
gbm@berkeley.edu

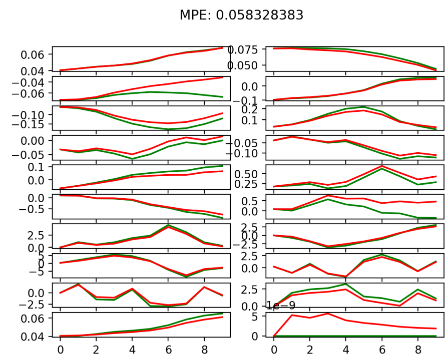
1. Problem 1



N=5, arch2x250



N=500, arch1x32



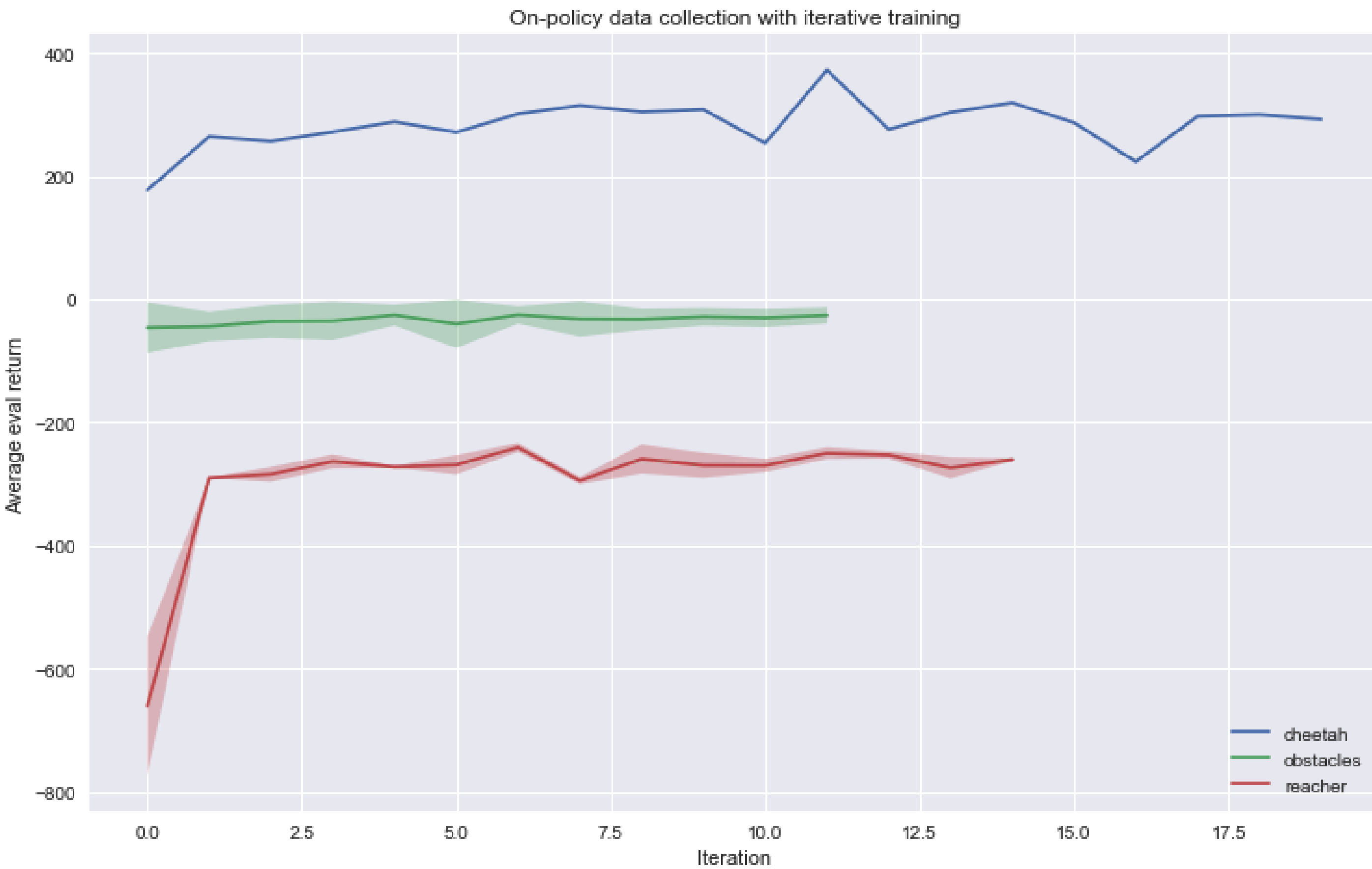
N=500, arch2x250

As shown in the above plots, the architecture with 2 layers each with 250 hidden units and trained for 500 iterations performs the best. This is likely because it has much larger capacity and is able to model more complex dynamics as opposed to the single layer with 32 units. The N=5 model performs worse due to being undertrained (too few gradient updates).

2. Problem 2

The eval average return is in green and the train average return is in blue.

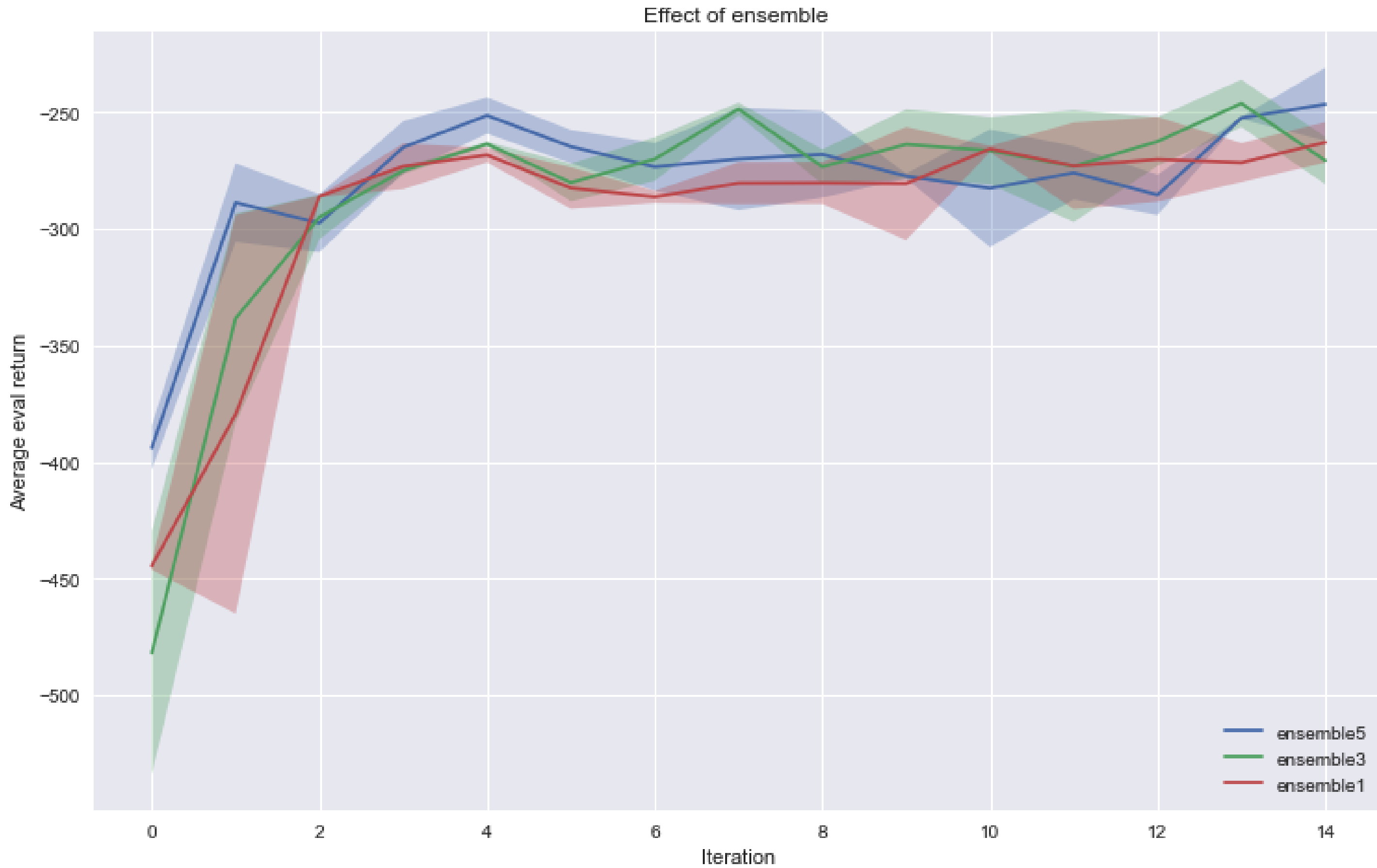
3. Problem 3


On-policy data collection with iterative training
Average eval return
Iteration
400
200
0
−200
−400
−600
−800
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
cheetah
obstacles
reacher

4. Problem 4

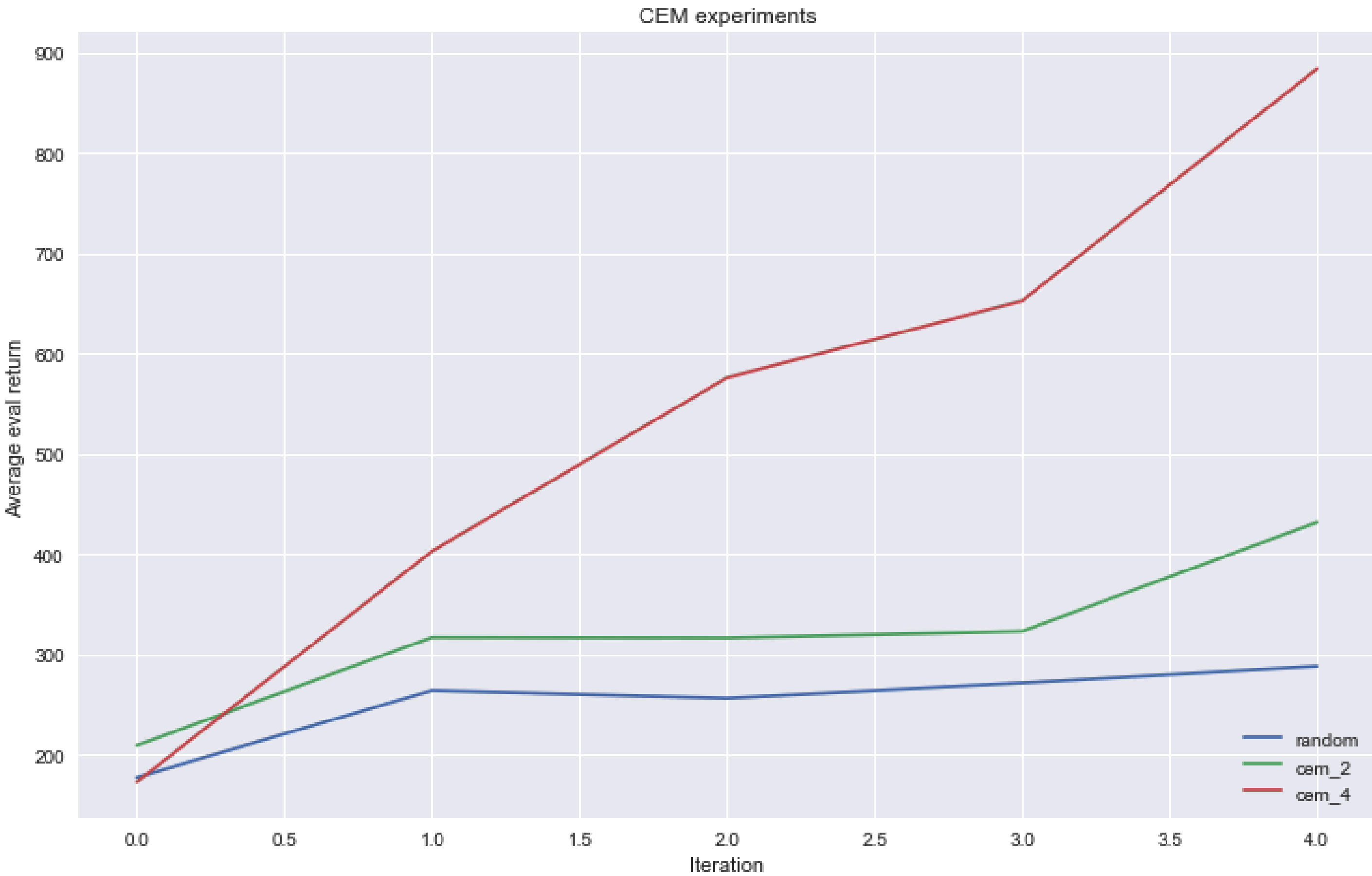
Planning horizon: Interestingly, a longer planning horizon does not necessarily lead to improved performance, with horizon=15 having most stable performance. This may be because the task cannot be effectively solved with 5 steps of planning and with 30 steps too much error from the model accumulates, preventing sufficient estimates.

Number of candidate sequences: Intuitively, more sequences means a much larger search space that can be evaluated and therefore a higher chance of finding performant candidates. This is consistent with the results shown above (better performance with larger number of candidate sequences).



Ensemble size: An increased ensemble size leads to slightly improved performance and lower variance in the overall average performance, which makes sense since the averaged estimates of many learners is more likely to produce a lower variance estimate when evaluating candidate sequences. This is consistent with the results above.

5. Problem 5



As shown, CEM with 2 iterations produces slightly improved results over random shooting and 4 iterations of CEM produces significantly improved results, outperforming both agents by nearly a factor of 2 in 2 iterations. While the 4 iteration agent is slower to train in terms of wall clock time, it is faster to improve in fewer training iterations.