IFT6163 Assignment 2: Model-Based RL

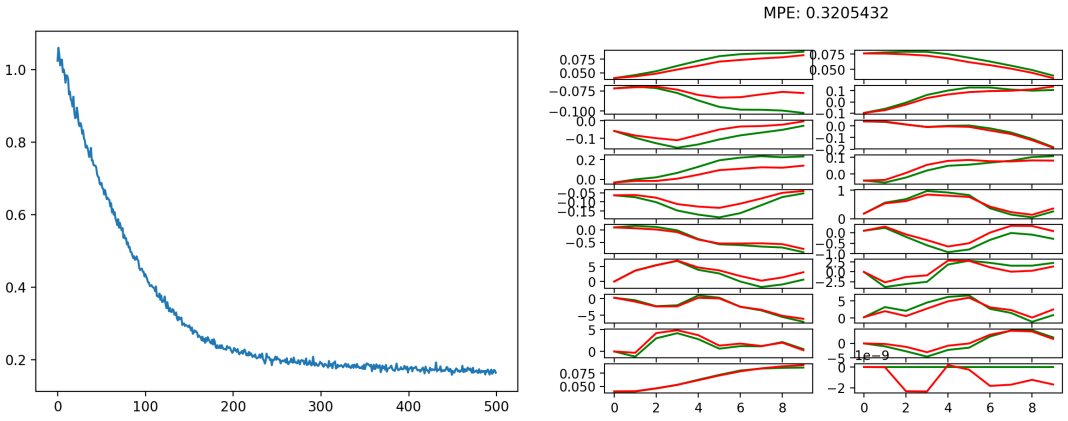
Simon Chamorro

February 14, 2022

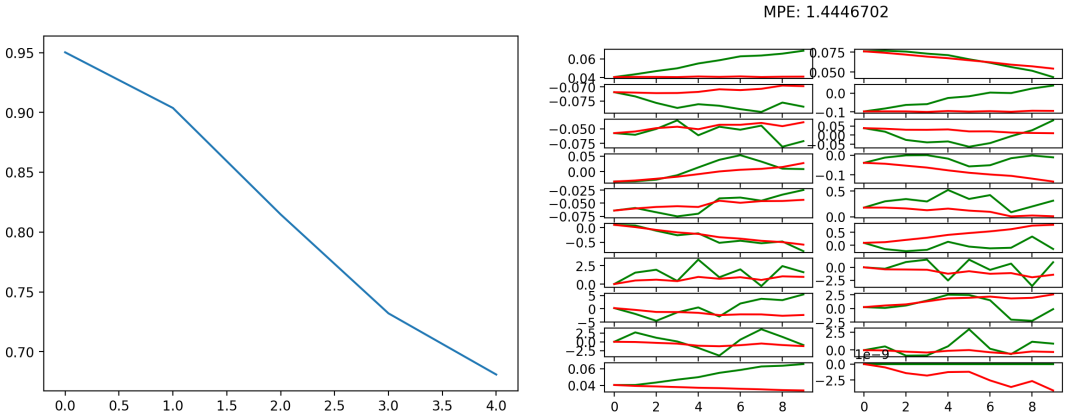
1 Dynamics Model

We evaluate three different model configurations for a dynamics model in the Cheetah environment. Figure 1 shows both the learning curves for the training of these models as well as some samples to visualise their predictions. First, model a) has only one hidden layer of size 32 and is trained for 500 steps. We can see that training does converge and it reaches a loss of 0.2. By looking at the predictions however, we see that they are somewhat accurate, but start diverging after a few steps. The second model b) has two hidden layers of size 250 and is trained for only 5 steps. We can clearly see that this amount of training is insufficient. The loss does decrease, but is far from attaining a plateau. Finally, model c) has two hidden layers of size 250 and is trained for 500 steps. We can see that the training loss reaches a slightly smaller value than, model a), but the model predictions seem more accurate and have a lower error. Overall, model c) is the best, since it has more capacity than model a) so it captures the dynamics better, and is trained for a fair amount of steps, reaching a loss plateau.

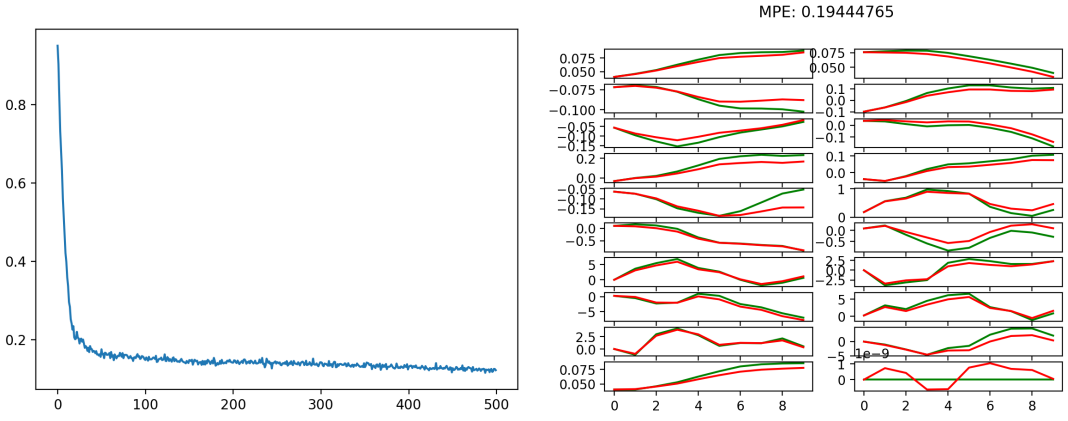
Name: Simon Chamorro
Date: February 14, 2022



(a) Model 1: 1x32, train steps = 500



(b) Model 2: 2x250, train steps = 5



(c) Model 3: 2x250, train steps = 500

Figure 1: **Dynamics Model Training Loss and Sample Predictions.**

Name: Simon Chamorro

2 Model-Based Agent

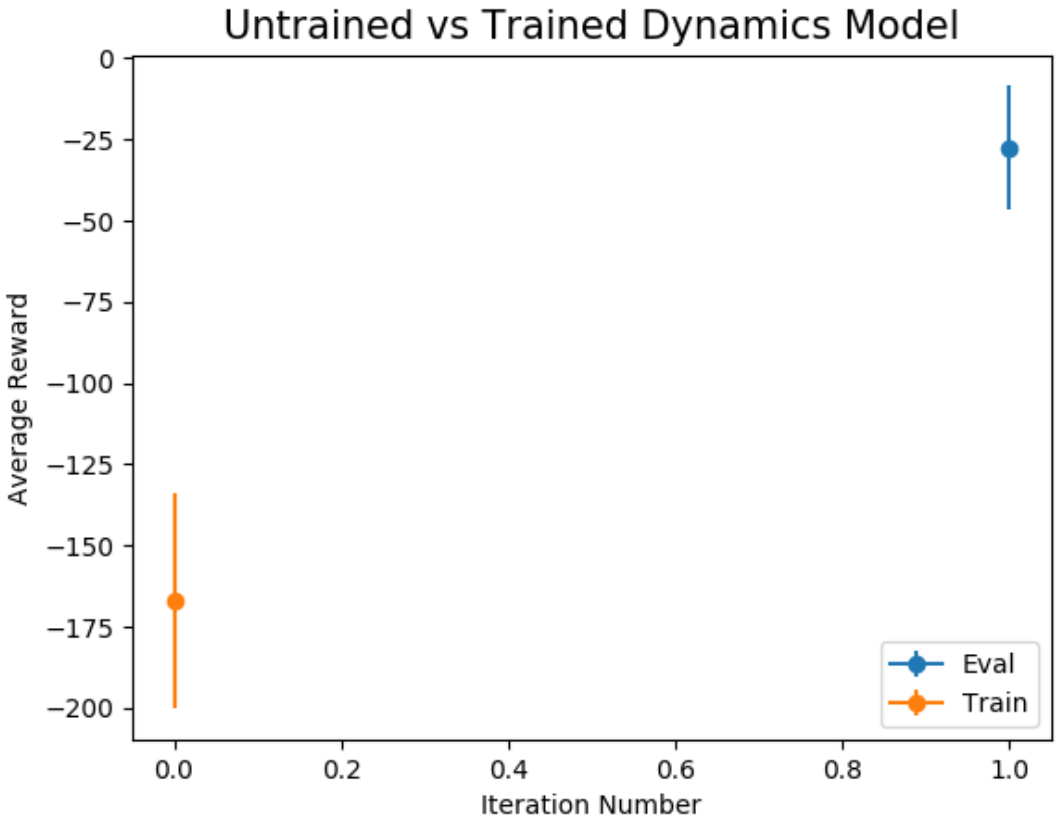
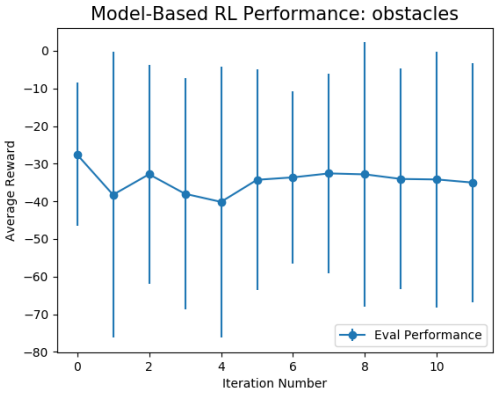


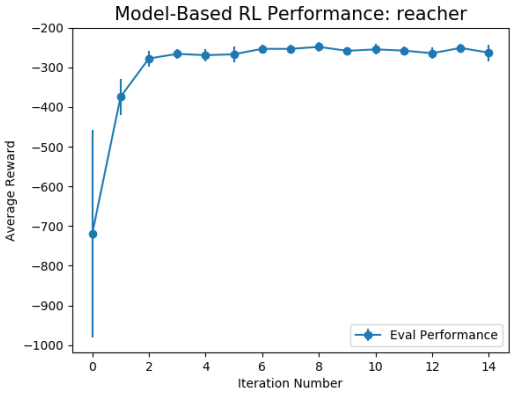
Figure 2: **Model-Based Agent Performance.** This figure shows the average reward obtained per episode before and after training the dynamics model. This illustrates that the MPC agent is doing a good action selection using the trained model. After one iteration of training, we see that the evaluation trajectories, which are selected via random-shooting using the model, achieve higher reward than completely random trajectories, which is expected behavior.

Name: Simon Chamorro

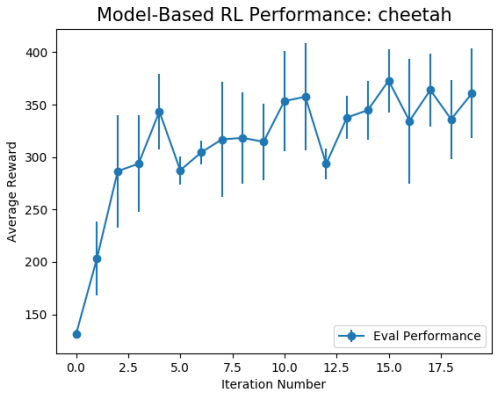
3 On-Policy Data Collection and Iterative Training



(a) Obstacles



(b) Reacher



(c) Cheetah

Figure 3: **MBRL Performance.** This figure shows the performance of a model-based RL agent using a dynamics model and random-shooting sampling. We can see that performance reaches -30 on the obstacles environment after only one training iteration and remains relatively constant afterward. On the reacher and cheetah environments, performance reaches the desired average rewards (-300 and 350) after only a few iterations.

Name: Simon Chamorro

4 Hyperparameter Analysis

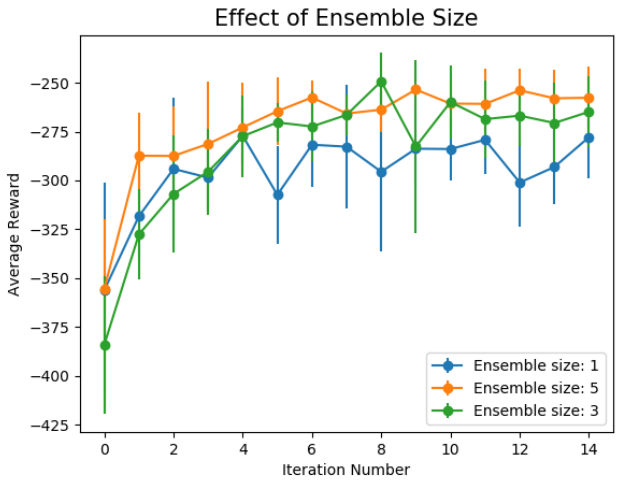


Figure 4: **Ensemble Size.** This plot shows performance for different ensemble sizes. We can see that higher ensemble sizes seem to perform better. By training several independent networks, we obtain a more unbiased prediction. We can see that the performance curve for an ensemble size of one is less stable. However, there is a trade off between ensemble size and training time, as increasing the number of model increases the training time significantly.

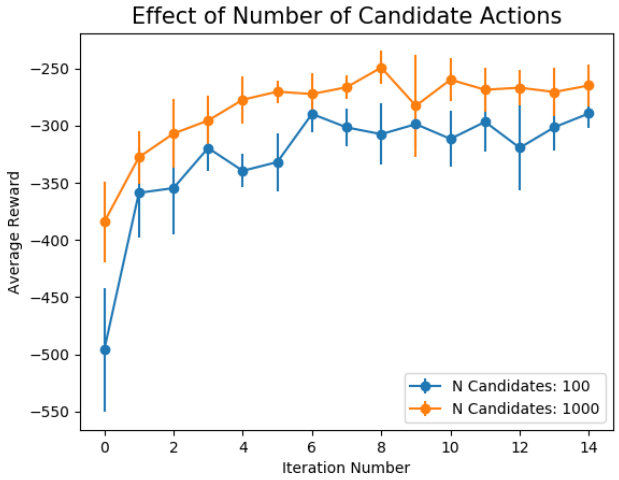


Figure 5: **Number of Candidate Sequences.** This plot shows performance for agents with different quantities of sampled candidate actions per step. We notice that a higher number of candidates yields better performance. In fact, the sampling is done via random-shooting, so more candidates means we explore a bigger space, so we have a higher probability of fining a better action. However, increasing this number increases computing time when evaluating actions.

Name: Simon Chamorro

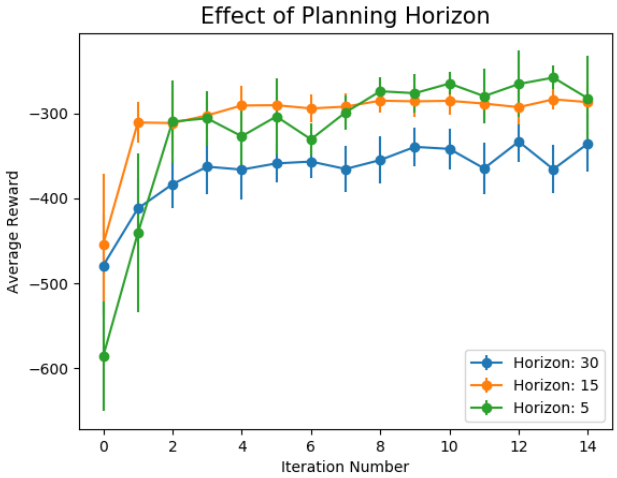


Figure 6: **Horizon.** This plot shows performance for different planning horizons. We notice that the performance in similar between the run with a horizon of 5 and the run with a horizon of 15, however, there seems to be a tendency for better performance for lower planning horizon. These results are tightly linked to the task. In fact, in the cheetah environment, the agent only needs to learn a good locomotion policy. This does not require any long-term planning, since the gait is repetitive and a cycle only spans over a few steps. I think these results would be very different in an environment that requires a long-term strategy.

Name: Simon Chamorro

5 Performance Comparison: Random-Shooting vs CEM

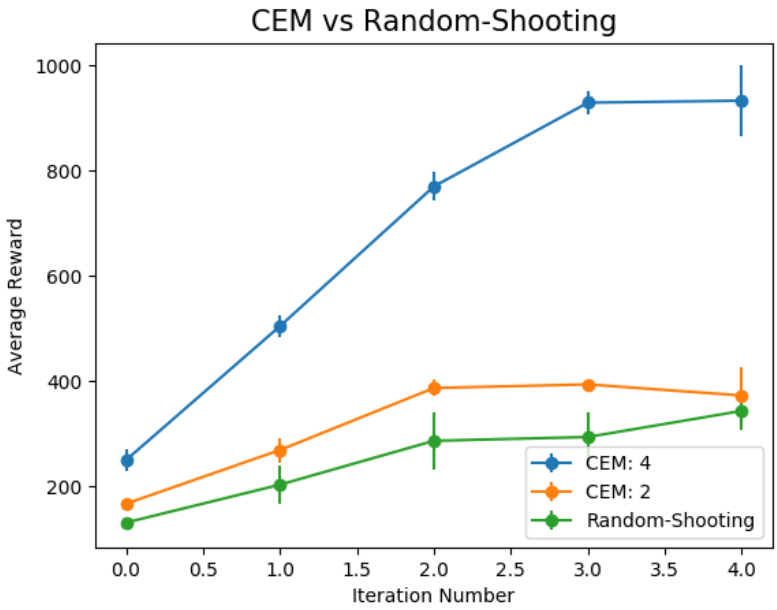


Figure 7: **CEM Performance.** This plot shows the performance of a MBRL algorithm using random-shooting for sampling versus using CEM with different numbers of iterations. We can see that CEM with 2 iterations slightly outperforms random-shooting, but there is a significant performance increase when using 4 CEM iterations. The more iterations we use, the more there is a bias to shift the distribution of actions towards an optimum, each CEM iteration helps steer the mean in the right direction all while continuing to explore the action space, since it still involves stochastic sampling from a distribution. The computational cost of CEM, however, is relatively high, so there is a trade off to be made when choosing a CEM iteration number.