Homework 4

Nicklas Hansen (s153077) s153077@student.dtu.dk
Peter Ebert Christensen (s153758) pebch@dtu.dk

June 14, 2019

Attribution table

Part	Responsible
Implementation: Q1	s153077 (40%), s153758 (60%)
Implementation: Q2 & Q3	s153077 (25%), s153758 (75%)
Figure 1	s153077 (45%), s153758 (55%)
Table 1	s153077
Table 2	s153077
Figure 2	s153758
Figure 3	s153077
Figure 4	s153758

Problem 1

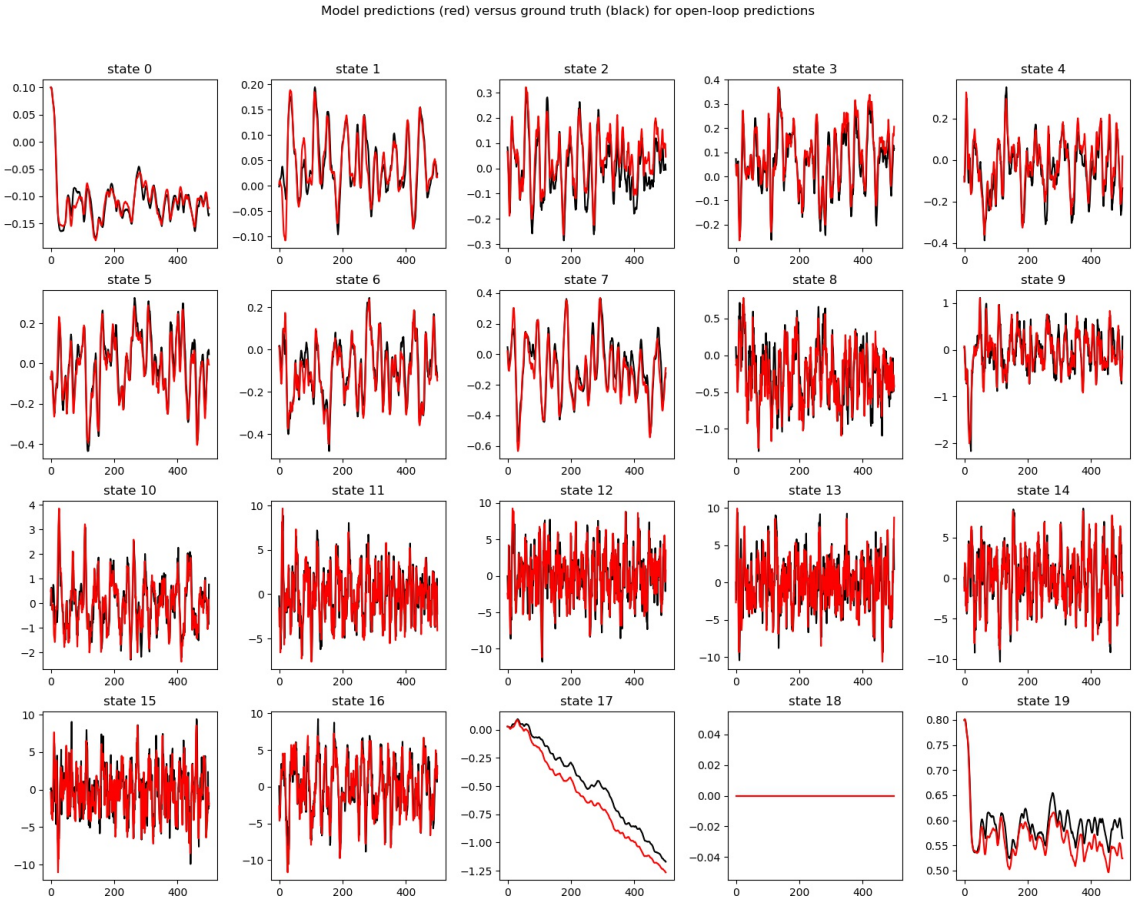

Figure 1: *Dynamics model predictions:* Plot of model prediction (red) and ground truth (black) for open-loop predictions on 512 states. Each subplot corresponds to one dimension in the state space. Generally, the learned dynamics model predicts states very well, except for dimension 17 (*ffoot*, the forward foot of the half cheetah). A reason for this might be that the state dimension is non-stationary.

Problem 2

Attribute	Ours	Random
Mean Return	-11.78	-155.3
Return std	26.99	30.91
Max Return	47.57	-107.9
Min Return	-48.13	-218.2

Table 1: *Off-policy training using learned dynamics model:* Results from experiment where a policy is trained (using its learned dynamics model) for 60 epochs on 512 uniformly sampled transitions (sampled at every epoch) generated by a random agent. It can be concluded that – despite not achieving a positive mean return – our agent does in fact learn something useful solely from its learned dynamics and transitions generated by a random agent.

Problem 3a

Attribute	Iteration 1	Iteration 3	Iteration 5	Iteration 10
Mean Return	66.07	192.9	260.3	296.3
Return std	29.24	20.89	32.99	25.33
Max Return	126.9	233.5	304.8	296.3
Min Return	30.63	162.1	181.4	254.4

Table 2: *On-policy training using learned dynamics model:* Results from experiment where a policy is trained initially on a set of random transitions like in Problem 2, and then on a mixture of transitions sampled from the random set and from rollouts generated using the trained policy, much like in the DAgger algorithm of Imitation Learning. We do this for 10 iterations and during each iteration, 10 rollouts are generated and added to the set that we sample from. It is observed that on-policy training drastically increases an agent's performance compared to just off-policy training with a random agent.

Problem 3b

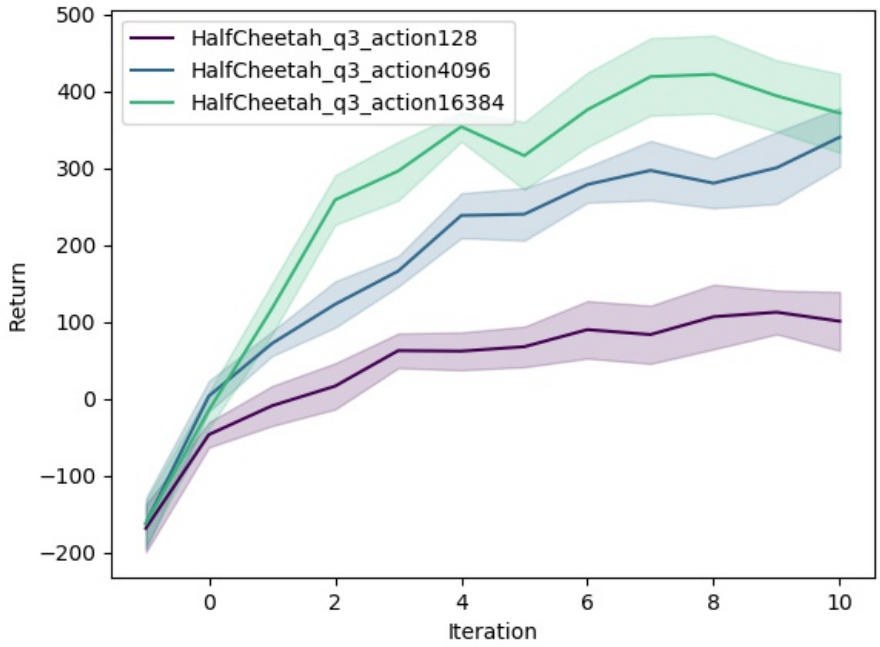

Figure 2: *Mean return by number of actions:* Plot of mean reward for the HalfCheetah environment with different lengths of the action sequences. It can be seen that longer action sequences generally appear to converge to a better solution and also learns faster. The difference in mean return for the 10th iteration of the 4096 and 16384 action sequences is however not substantial.

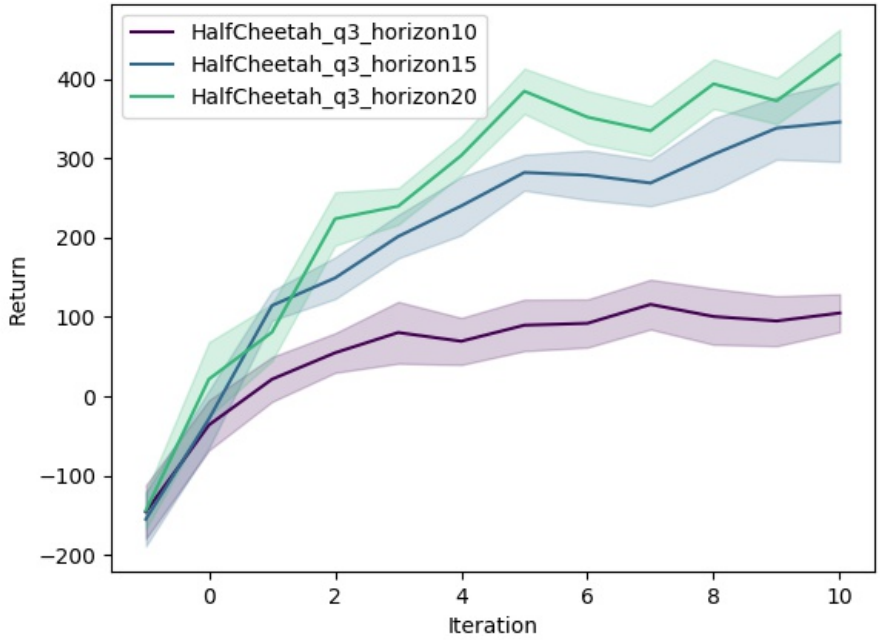

Figure 3: *Mean return by MPC horizon:* Plot of mean reward for the HalfCheetah environment with different MPC horizons. It can be seen that a longer horizon yields a better solution, but the biggest performance improvement is to be found when increasing the horizon from 10 to 15.

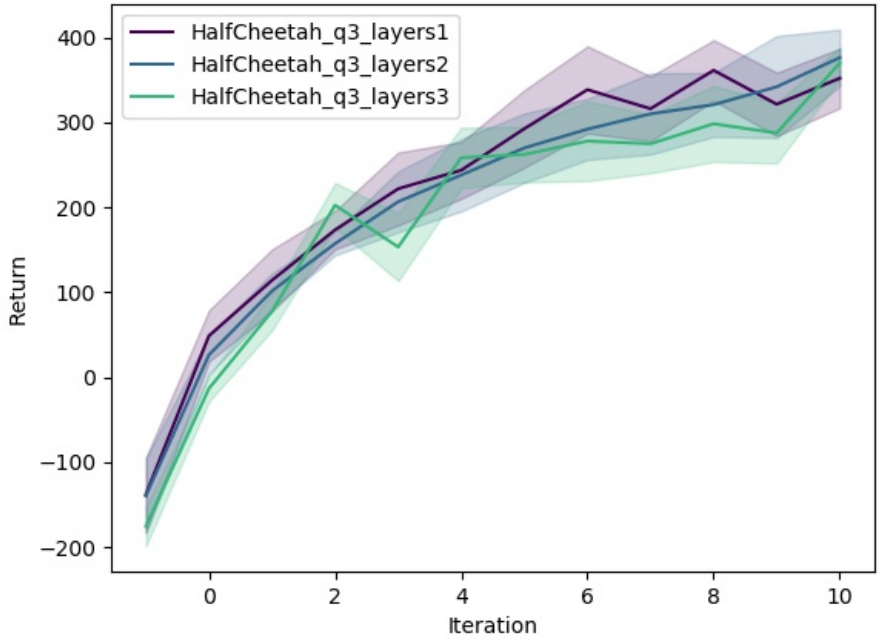

Figure 4: *Mean return by number of layers:* Plot of mean reward for the HalfCheetah environment with varying the number of neural network layers for the learned dynamics model. It can be seen that the number of layers doesn't really matter, probably because the number of non linearities are low in this game and also because we can complete the game with 1 layer with enough neurons (512).