Assignment 4: Model-Based RL and Exploration

Andrew ID: `vtambe`
Collaborators: `pvenkat2`
NOTE: Please do **NOT** change the sizes of the answer blocks or plots.

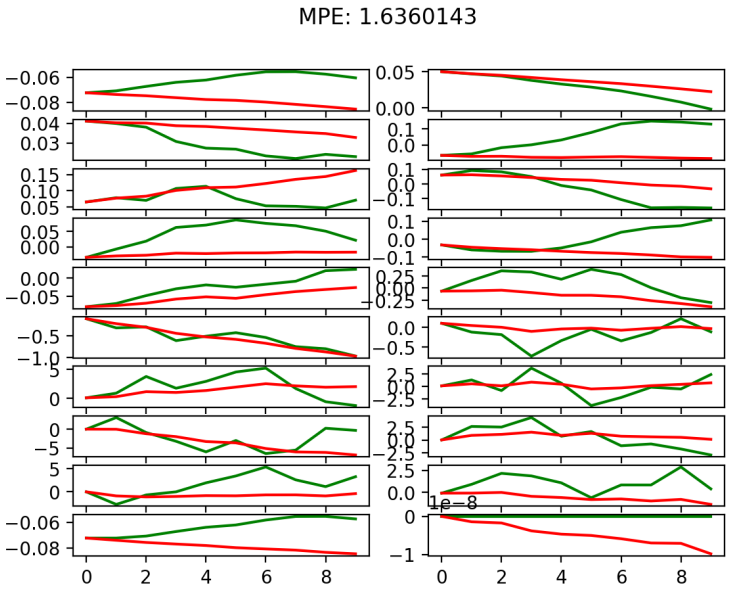
1 Problem 1: Dynamics Model Training

Answer to theory questions.

Comments on model performance:

The model that performs the best is the $q1\_cheetah\_n500\_arch2x250$. It has a mean squared error of 0.07804488 as can be seen by plot3 of Problem 1. It performs the best out of the 3 runs because the agent is trained for '500' per iteration and along with the size of the hidden layer used in the policy being 250 which allows the model to learn more information.

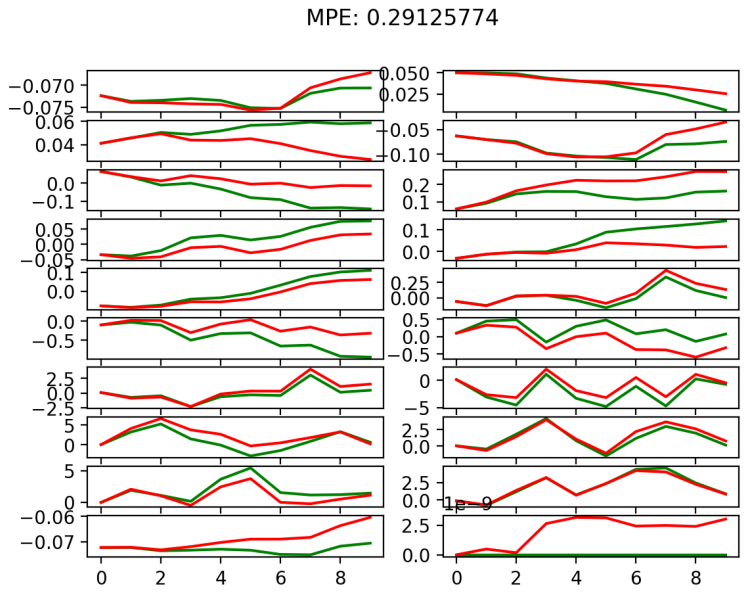
Plot 1



q1_cheetah_n500_arch1x32

1 Problem 1: Dynamics Model Training

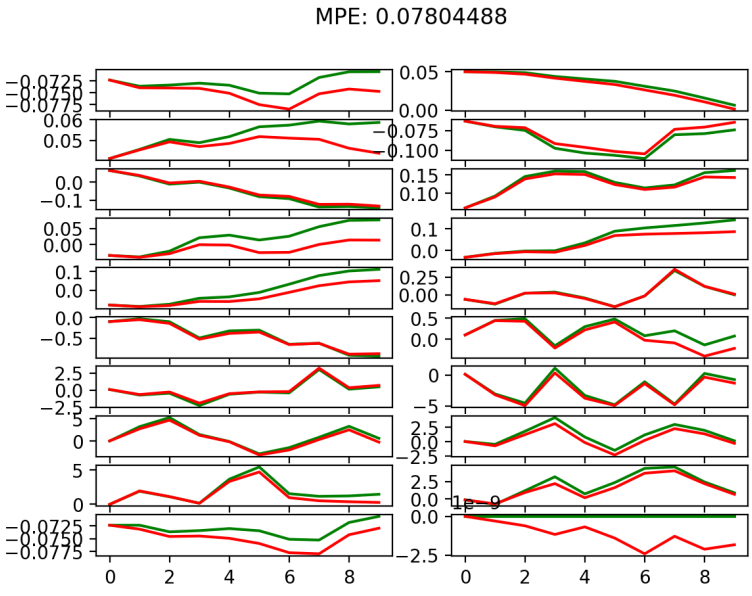
Plot 2



q1_cheetah_n5_arch2x250

1 Problem 1: Dynamics Model Training

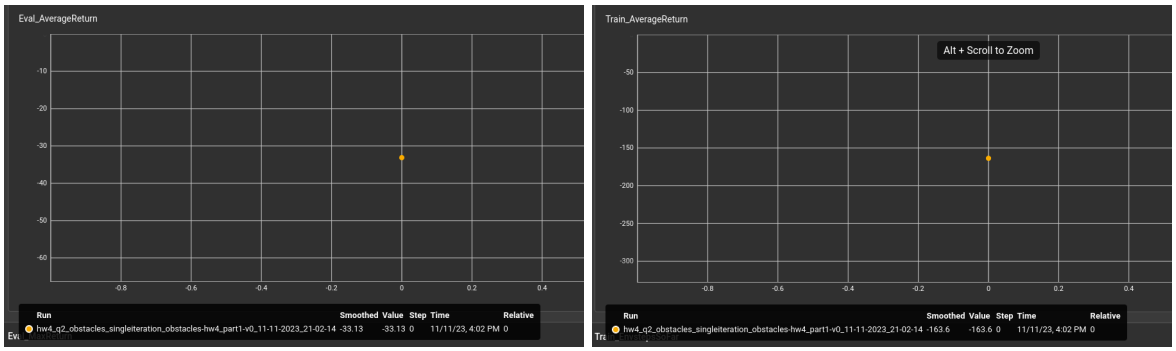
Plot 3



q1_cheetah_n500_arch2x250

2 Problem 2: Action Selection

Plot 1

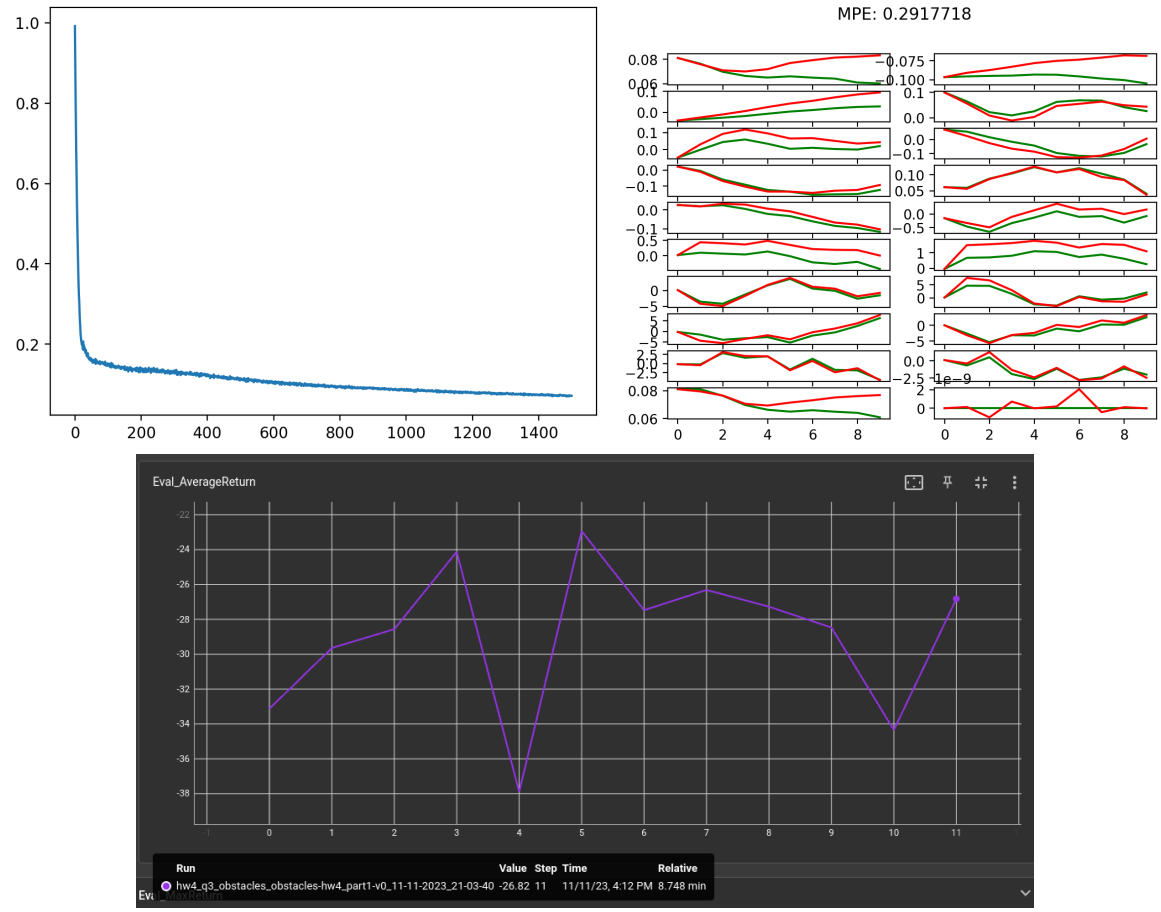


q2_obstacles_singleiteration

Eval Average Return = -33.13 Train Average Return = -163.6

3 Problem 3: Iterative Model Training

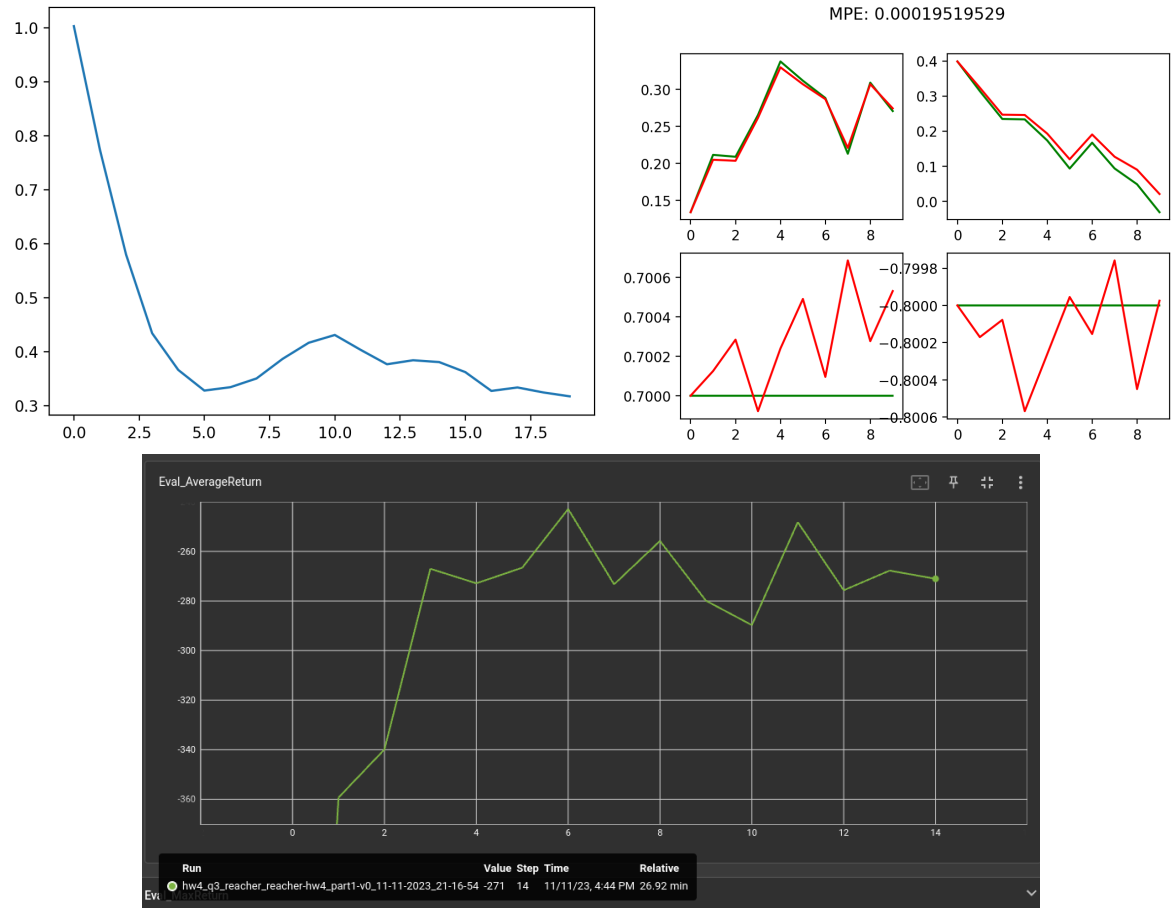
Plot 1



Obstacle env

4 Problem 3: Iterative Model Training

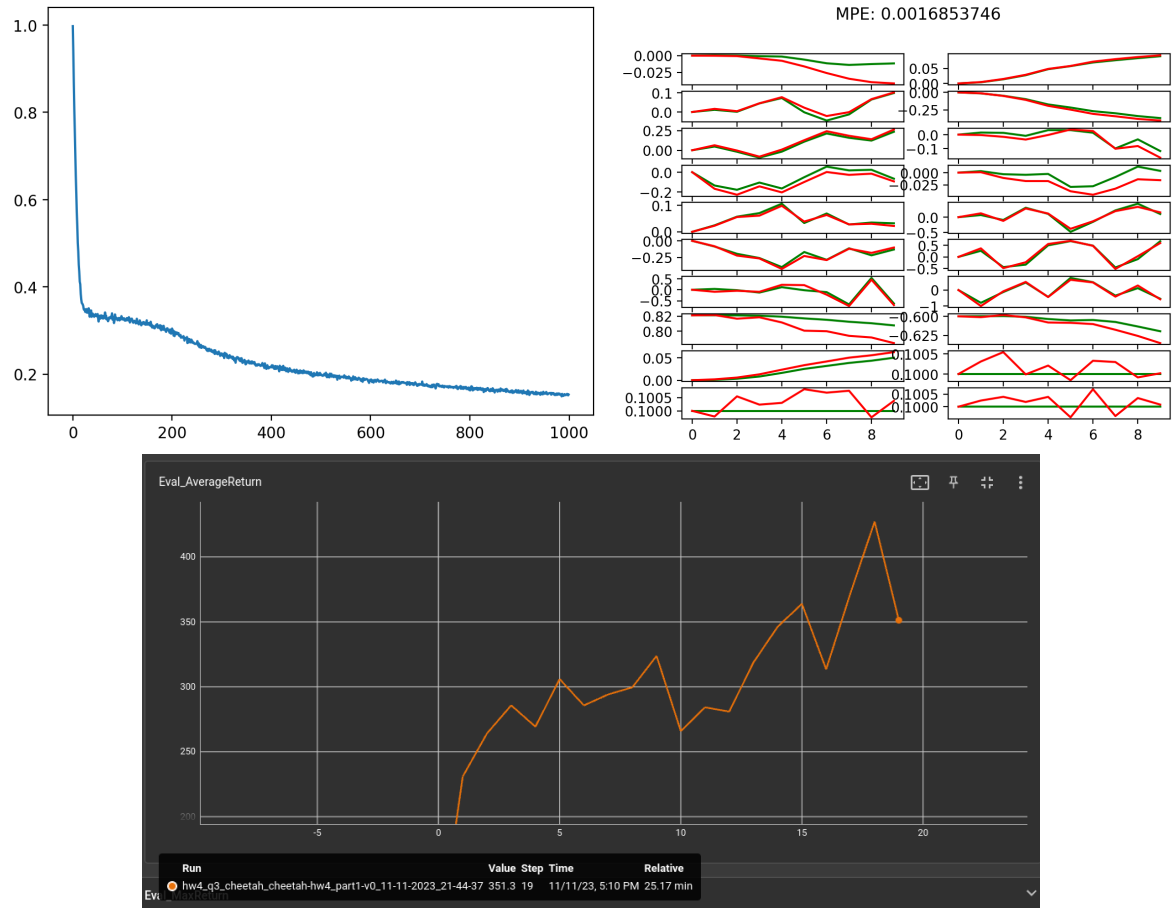
Plot 2



Reacher env

5 Problem 3: Iterative Model Training

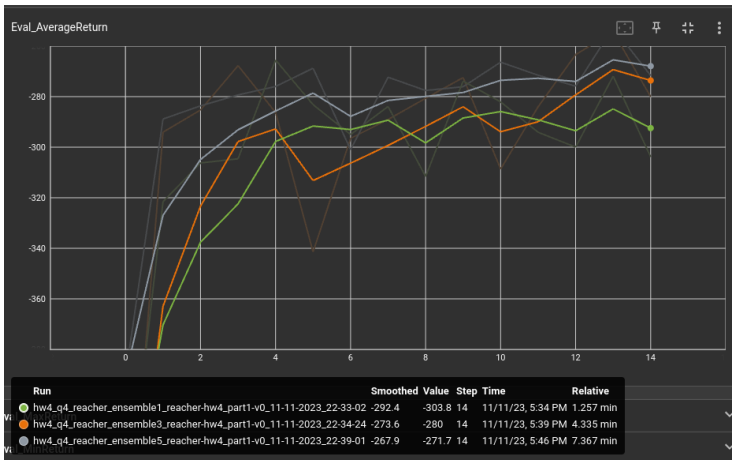
Plot 3



Cheetah env

6 Problem 4: Hyper-parameter Comparison

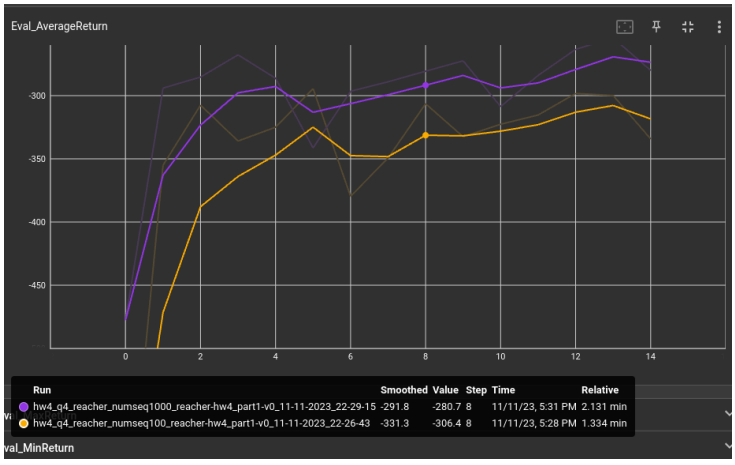
Plot 1



The above plot is for eval_averageReturns for an ensemble size of 1, 2 & 3. As can be seen from the plot the average return increases as we increase the number of models used in the ensemble. As we increase the number of ensembles the overall variance in the predictions is reduced and we get more stable results.

7 Problem 4: Hyper-parameter Comparison

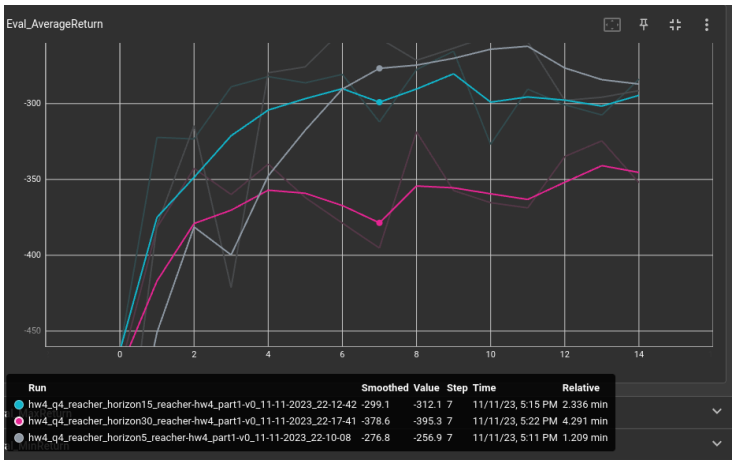
Plot 2



The above plot is for eval_averageReturns for random action sequences of length 100 & 1000. As can be seen from the plot the average return increases as we increase the number of random actions sequence. As we randomly sample a higher number of action sequences the probability of coming across a candidate action sequence with high reward increases thus the model is trained on high reward action sequences which improves the performance of the model.

8 Problem 4: Hyper-parameter Comparison

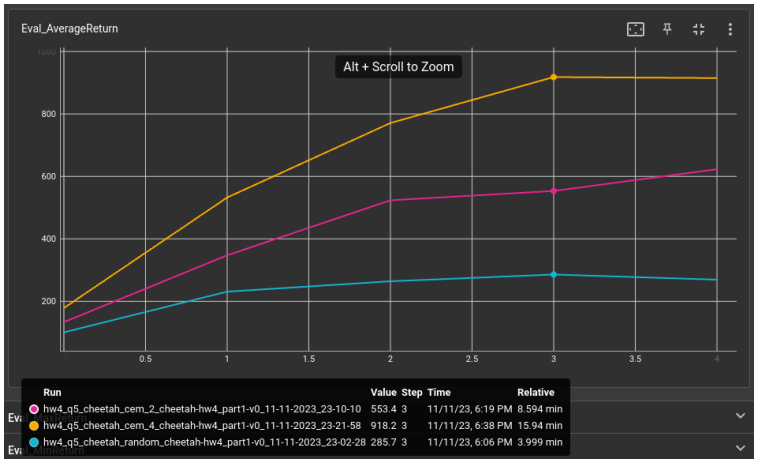
Plot 3



The above plot is for eval_averageReturns for a horizon of length 5, 15 & 30. As can be seen from the plot the average return decreases as we increase the horizon length.

9 Problem 5: Random Shooting vs CEM

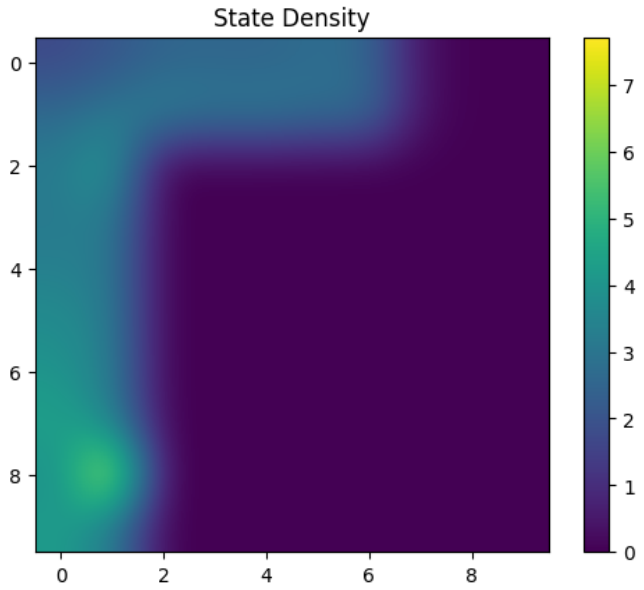
Plot 1



The above graph plots the *Eval_AverageReturn* for a policy used using random shooting and 2 policies trained using cross-entropy-method (CEM) for 2 iterations and 4 iterations respectively. The returns for CEM is more than random shooting and as we increase the CEM iterations we see the results improve. This is because CEM works by sampling K random action sequences similar to random shooting but uses only the 'J' highest reward action sequences for model training thus converging to a be better solution faster.

10 Problem 6: Exploration

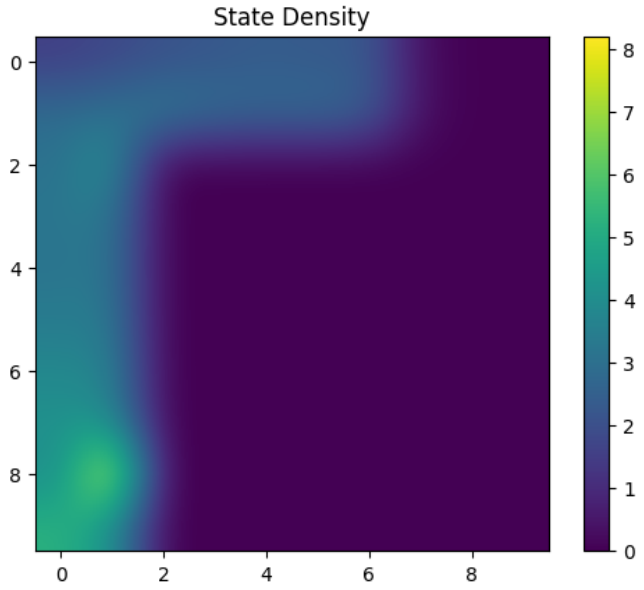
Plot 1



State Density Plot - exploration using random network distillation - PointMassEasy.

11 Problem 6: Exploration

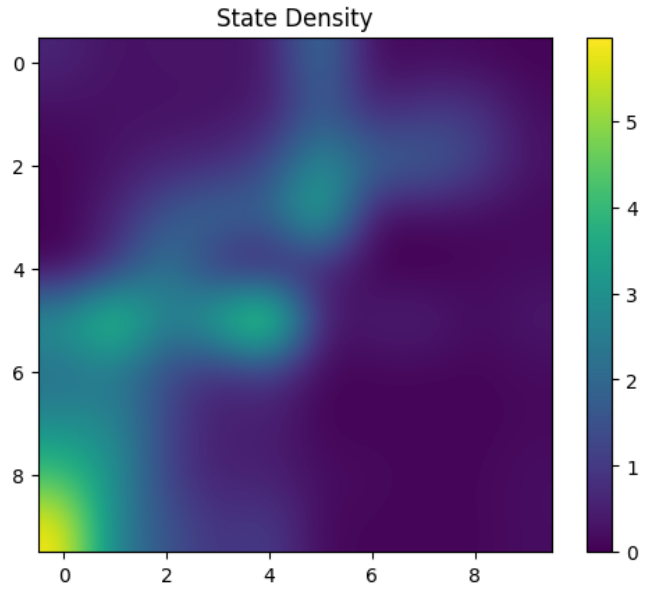
Plot 2



State Density Plot - random exploration - PointMassEasy.

12 Problem 6: Exploration

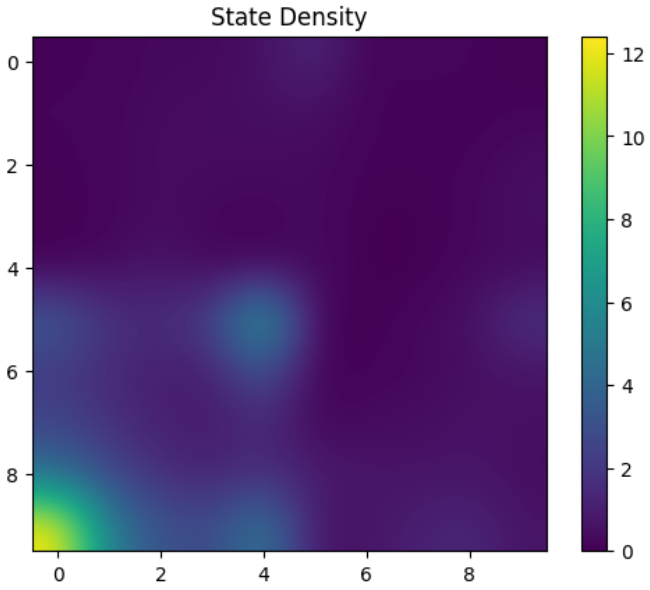
Plot 3



State Density Plot - exploration using random network distillation - PointMassHard.

13 Problem 6: Exploration

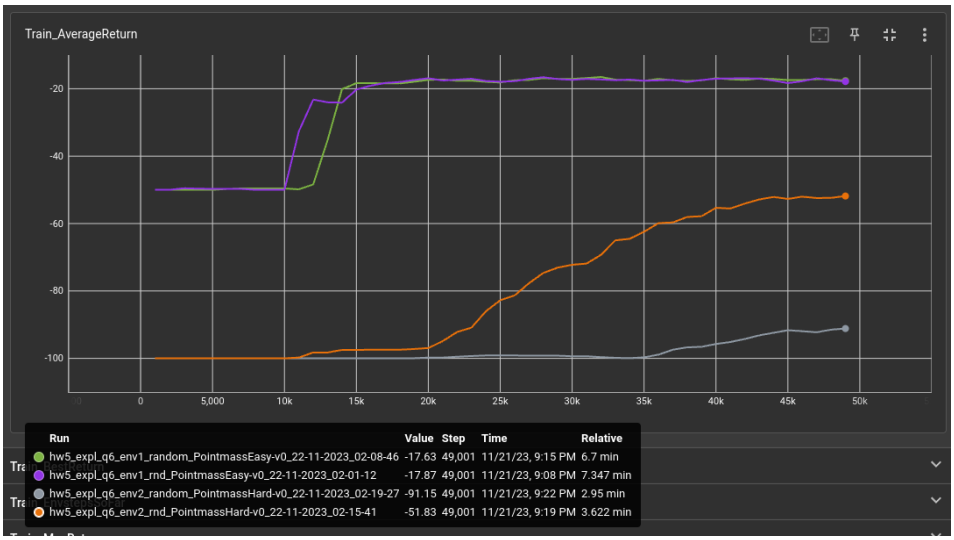
Plot 4



State Density Plot - random exploration - PointMassHard.

14 Problem 6: Exploration

Plot 5



As seen from the above plot the agent learns better policy using random network destination (RND) as it is able to explore regions of the state space where it has high uncertainty (using exploration-bonus). This is not possible wth random exploration.

15 Problem 6: Bonus

Plot 5