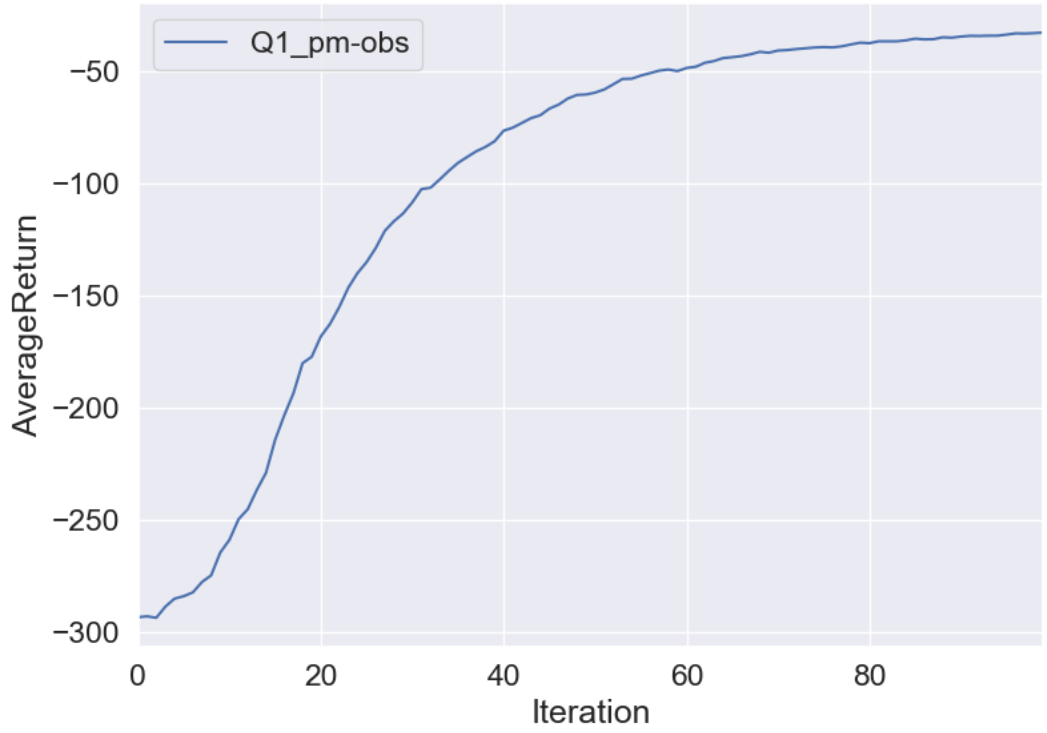
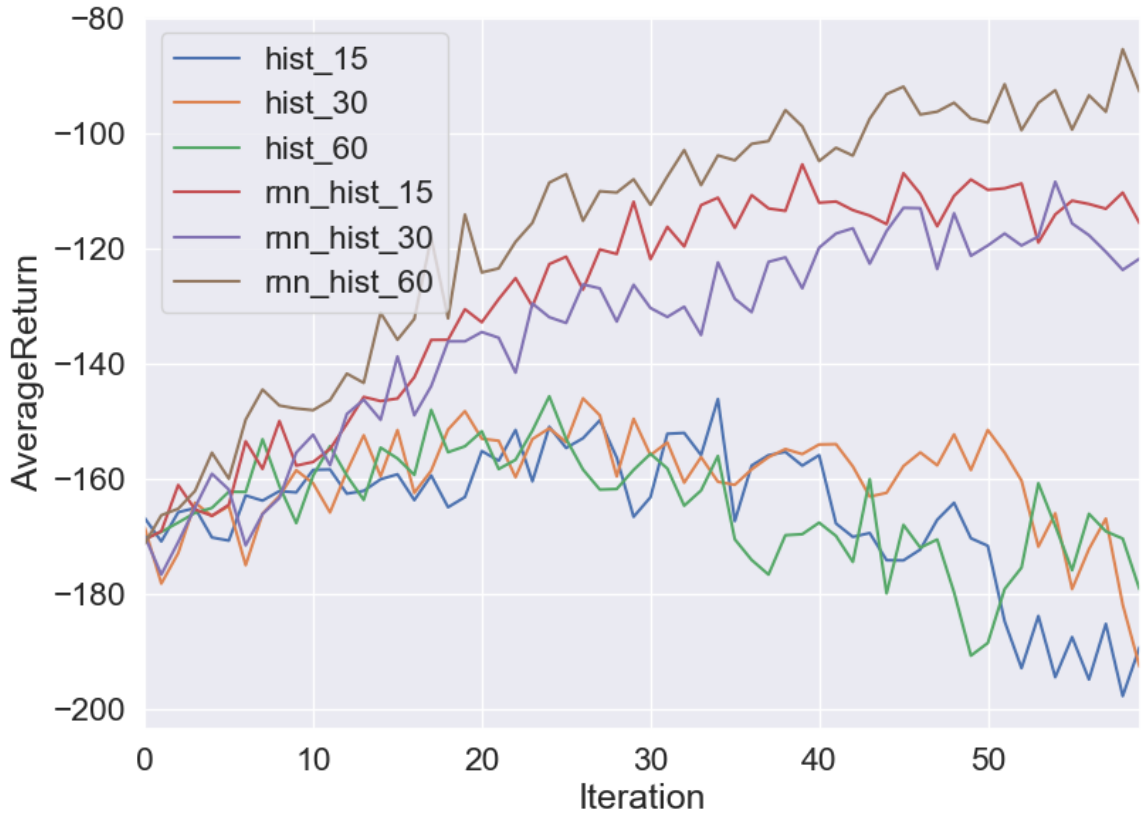
Ezbakhe Hatim

CS294-112 Deep Reinforcement Learning
HW5: Meta-Reinforcement Learning

Problem 1



Problem 2



Looking at the results, it is clear that the fully connected policy performs poorly on this environment, no matter the number of history kept. Storing the meta observations does not improve the performance as we hardly go beyond -150 before dropping again.

However, using a recurrent neural network improves the results considerably. We cannot really tell because the experiment stopped at 60 iterations, but it looks like for a history lower than 60, the returns stopped increasing and start to drop, which is not the case when history = 60.

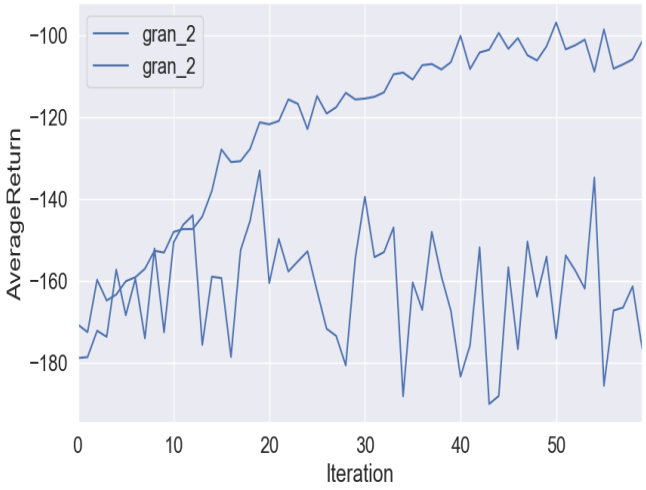
It looks like there is a minimum history length needed, that seems to be 60 (or between 30 and 60).

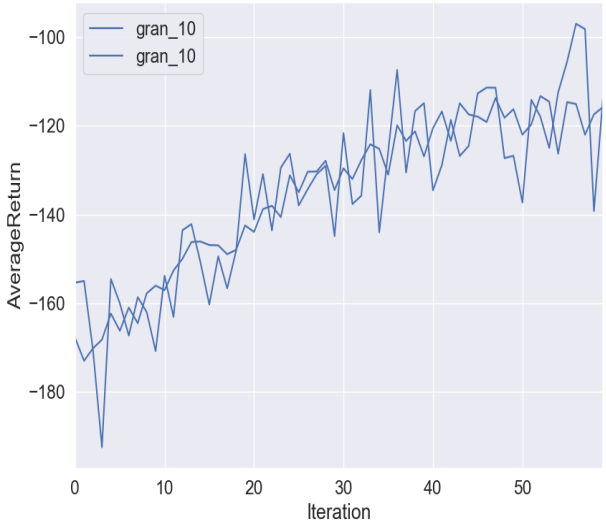
Problem 3

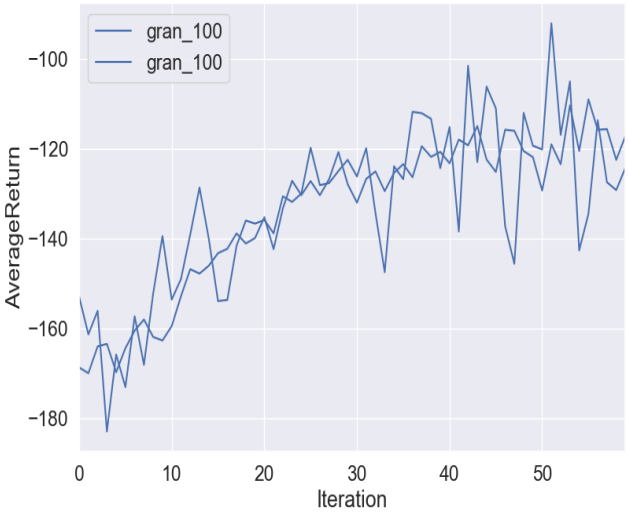
For this problem, to get more results, I ran the algorithm with history set to 30 so that it runs faster.

I defined the granularity as the number of sub squares of the chessboard on one axis. So for a granularity of 2, the (-10, -10) x (10, 10) square is divided into 4 squares.

Here are the results for different values of granularity:







The ValAverageReturn plot correspond to the one with higher variance.

What we can wee from the results shown above is that for a high granularity, the model performs as well on the training set than on the evaluation set. That seems pretty logical. Indeed, with a high granularity, the space is subdivided into such small squares, that the evaluation set and the training set are extremely close. Although being disjoints, the distance between a given point in the evaluation set and its closest point in the training set is really small. That means that when we set a goal, despite being never solved before, the model solved a high number of tasks near that point. The only notable difference is the higher variance of the results on the evaluation set.

For a low granularity, the model doesn't perform as well anymore, as we can see for a granularity of 2. The variance is high for goals in the evaluation set, and the returns extremely low (we see no improvement). This is due to the fact that a goal taken from the evaluation set would be (with a high probability) far away from any goals previously seen in the training set. That explains why the model performs poorly for those cases.
Note that we can see improvements on both the evaluation and training set for a granularity of 5. The variance issue still remains