

# CSE 599B: Homework 3 Submission

Alexey Yermakov

May 31, 2024

## 1 Random MPC with Shooting

- The reward during training with default hyperparameters for Random MPC with Shooting is plotted below:

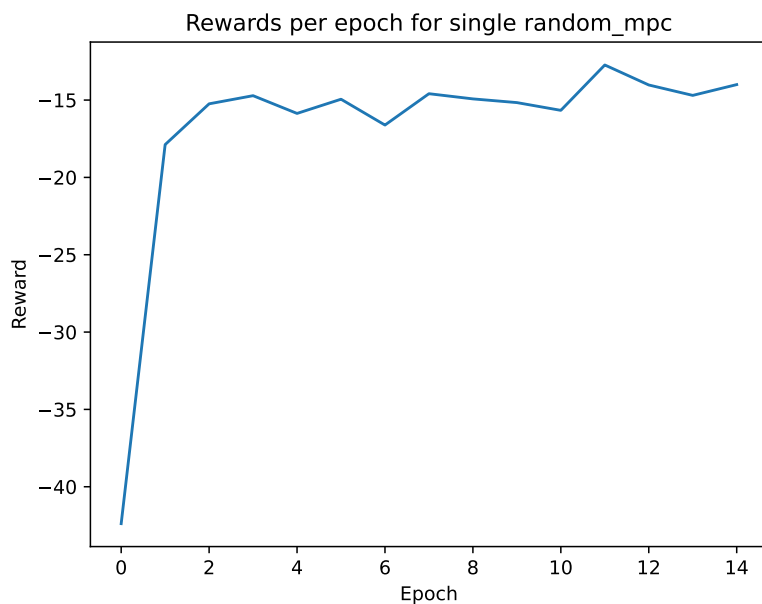


Figure 1: Reward per epoch with default hyperparameters for Random MPC with Shooting.

- The success rate and average reward for Random MPC with Shooting with default hyperparameters is:

Success rate: 0.41

Average reward (success only): -13.303889559240533

Average reward (all): -14.45317813467756

## 2 Model Predictive Path Integral (MPPI)

- The reward during training with default hyperparameters for MPPI is plotted below:

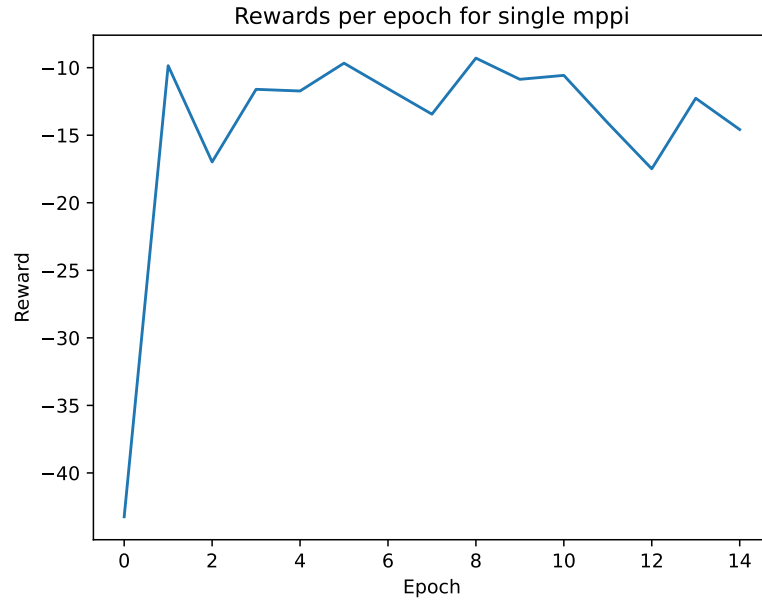


Figure 2: Reward per epoch with default hyperparameters for MPPI.

- The success rate and average reward for MPPI with default hyperparameters is:

Success rate: 0.37

Average reward (success only): -9.265817016106594

Average reward (all): -12.27599838847684

### 3 Ensemble MPPI

- The reward during training with default hyperparameters for ensemble MPPI is plotted below:

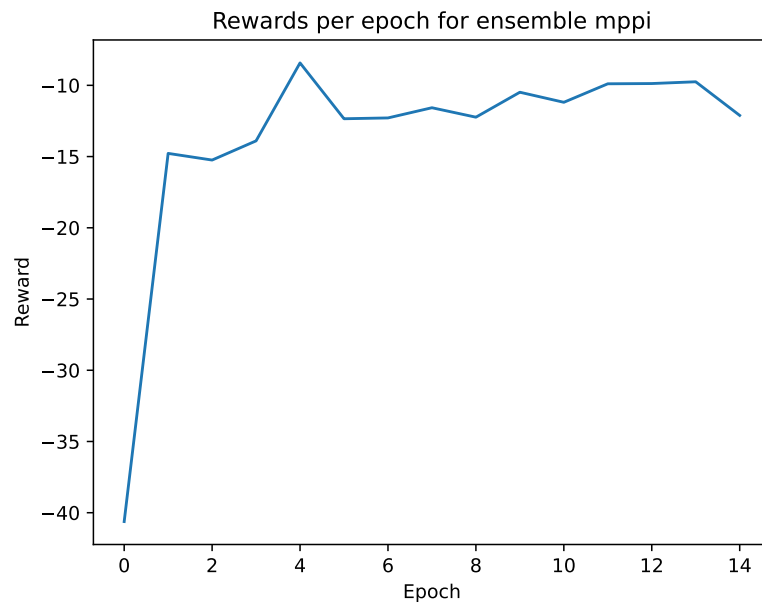


Figure 3: Reward per epoch with default hyperparameters for ensemble MPPI.

- The success rate and average reward for ensemble MPPI with default hyperparameters is:

Success rate: 0.41

Average reward (success only): -9.288529749275632

Average reward (all): -11.397747670089936

## 4 Discussion

1. Observing the performance of Random MPC with shooting, MPPI, and Ensemble MPPI leads to several key conclusions. First of all, the success rate for the reacher task is approximately the same for all three tasks from when I ran the code, at about 40%. However, the average reward of the MPPI based methods are much higher than the random shooting. This makes sense since the random shooting is a random search and MPPI is much more of an informed. However, this informed search comes at the cost of runtime and code complexity. Random shooting ran in 1 minute and 3 seconds, single model MPPI ran in 6 minutes and 52 seconds, while ensemble MPPI ran in 29 minutes and 17 seconds. So, the improvement in reward for using MPPI came at the cost of runtime, which makes sense. However, the difference between single MPPI and ensemble MPPI was marginal in my results: an approximately the same average reward and a 4% success rate improvement for the ensemble. This marginal improvement, however, came at the cost of running the code 5 times longer (the number of ensemble models used)!
2. Diversity is important in an ensemble model because it helps promote searching diverse paths when doing path planning. If all the models were the same, the ensemble would do nothing to improve performance - the planning phase of the reinforcement learning loop would just take longer to run. Thus, in this homework I implemented varying batch sizes and learning rates used during model training for each model in the ensemble.