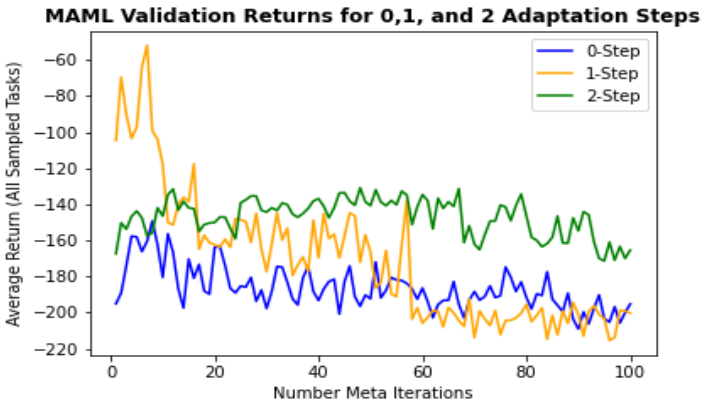
Douglas Post
COSC 689
Homework 2

Validation and Testing Results for MAML Implementation

To start, this was probably the most difficult programming task I have had in graduate school so far. It took me around four days to get my code running properly because the parameter passing and model graphs were really difficult for me to grasp in Tensorflow. Now, moving to my results, obviously they don't look like I expected them to, but I am going to go through the two or three reasons why I believe that is the case. I will also point out the little instances that make me think it is correct, just not tuned properly.

First, I could not get the environment to give a positive return, and I was not sure if this was supposed to be the case or not, but all of my average task returns in the inner and outer loops were always negative values. I don't know how much of an issue this is as that should not change the optimization and learning of the algorithm. The more important notes are the Vanilla Policy Gradient for my inner agent, and that I did not have enough time for tuning and rerunning the algorithm to test learning rates and network architectures. REINFORCE is by nature a high variance algorithm, so using that as the adaptation per task for my inner agent probably made it very difficult for the outer agent to learn efficiently. I had to shorten the number of meta iterations to only 100, which means the outer agent was trained less, and the inner agent has high variance in its updates. My learning rate was probably too high, only adding to the variance problem. With more time I would have trained with many more meta iterations, a lower learning rate and added a baseline to my inner REINFORCE agent.

However, there are some important things to note about my results. In 100 iterations the 2-step adaptation is the steadiest, and ends at the best return value. This makes sense because it was able to update the REINFORCE agent twice on each sample task per meta iteration, so the policy network would better be able to learn the state-action values. However, with more and more iterations REINFORCE having high variance, and updating twice could lead to some overfitting or certain iterations causing possibly noisy updates to the outer agent parameters. The 1-step method would see less of this. This algorithm with 0 adaptation steps is pretty much a PPO Agent trying to learn multiple tasks, and it makes sense that it would have a difficult time with this, especially with only 100 iterations. This also would explain 2-steps superior performance on the testing set.



	Return [All]	Return [Forward]	Return [Backward]
0 – Step	-160.52	-164.23	-156.81
1 – Step	-170.87	-166.35	-175.39
2 – Step	-141.64	-133.64	-145.95