This handout includes space for every question that requires a written response. Please feel free to use it to handwrite your solutions (legibly, please). If you choose to typeset your solutions, the —README.md— for this assignment includes instructions to regenerate this handout with your typeset LaTeX solutions.

1.b

These are the results of Ant-v4 and HalfCheetah-v4 environments. The average return is the mean of the returns obtained from evaluating the trained policy on multiple episodes (total 5000 timesteps), while the standard deviation of return indicates the variability of returns across those episodes:

Environment	Eval_AverageReturn	Eval_StdReturn
Ant-v4	4601.469727	98.333435
HalfCheetah-v4	3833.641357	61.327606

These were the relevant hyperparameters for Ant-v4 (the only training data used was the 2000 timesteps of experience in the provided expert data):

Hyperparameter	Value
num_agent_train_steps_per_iter	10000
train_batch_size	100
eval_batch_size	5000
n_layers	2
size	64

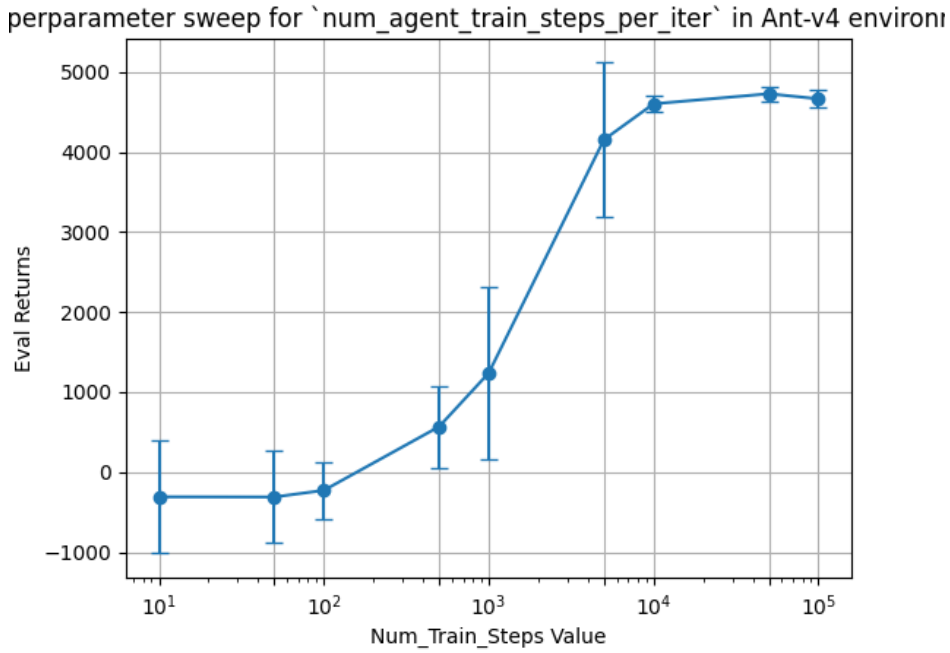
These were the relevant hyperparameters for HalfCheetah-v4 (the only training data used was the 2000 timesteps of experience in the provided expert data):

Hyperparameter	Value
num_agent_train_steps_per_iter	10000
train_batch_size	100
eval_batch_size	5000
n_layers	2
size	8

1.c

I did a hyperparameter sweep for the 'num_agent_train_steps_per_iter' hyperparameter in the Ant-v4 environment. My motivation was that in Behaviour Cloning, that was the only variable that really had any effect on the algorithm's performance. I wanted to see if spamming higher and higher values for this hyperparameter would lead to better and better performance, or if we would see diminishing returns at some point.

These were the results of varying the 'num_agent_train_steps_per_iter' hyperparameter for the Ant-v4 environment:

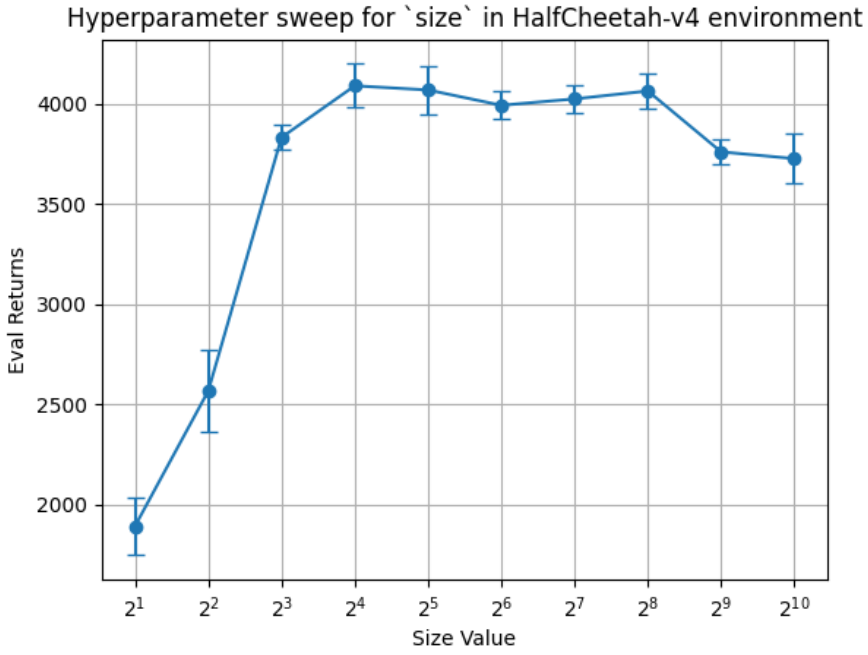


The rest of the hyperparameters were kept constant:

Hyperparameter	Value
train_batch_size	100
eval_batch_size	5000
n_layers	2
size	64

I also did a hyperparameter sweep for the 'size' hyperparameter in the HalfCheetah-v4 environment. I noticed that the observation size for the HalfCheetah-v4 environment was 17, which was much smaller than for Ant-v4 environment's 111. Because of this, I suspected that a smaller neural network would be able to learn a good enough policy for HalfCheetah-v4, and that we would see diminishing returns with larger networks. I suspected that very large networks would perform even more poorly because they would overfit to the 2000 timesteps observed in the expert data (assuming we would see sufficiently different data in the evaluation runs).

These were the results of varying the 'size' hyperparameter for the HalfCheetah-v4 environment:

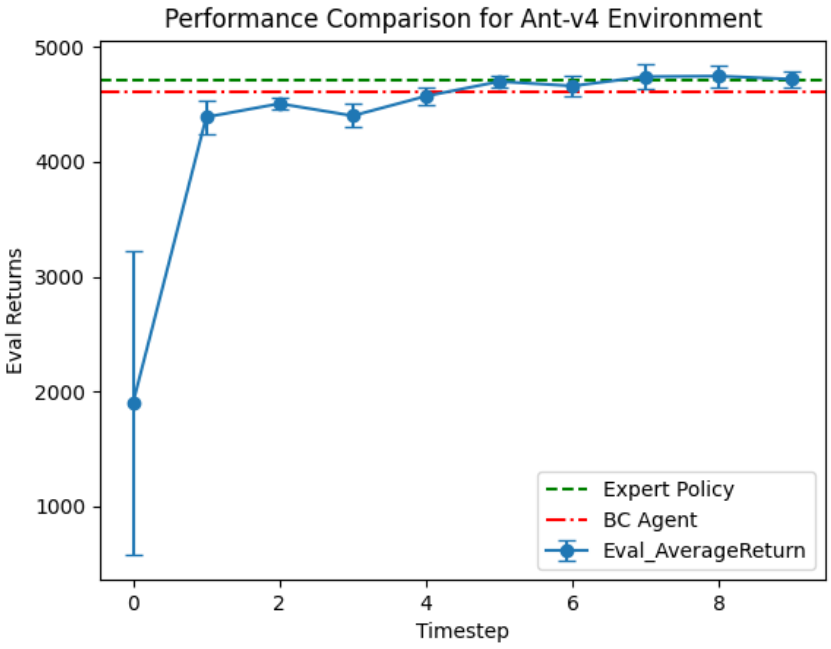


The rest of the hyperparameters were kept constant:

Hyperparameter	Value
num_agent_train_steps_per_iter	10000
train_batch_size	100
eval_batch_size	5000
n_layers	2

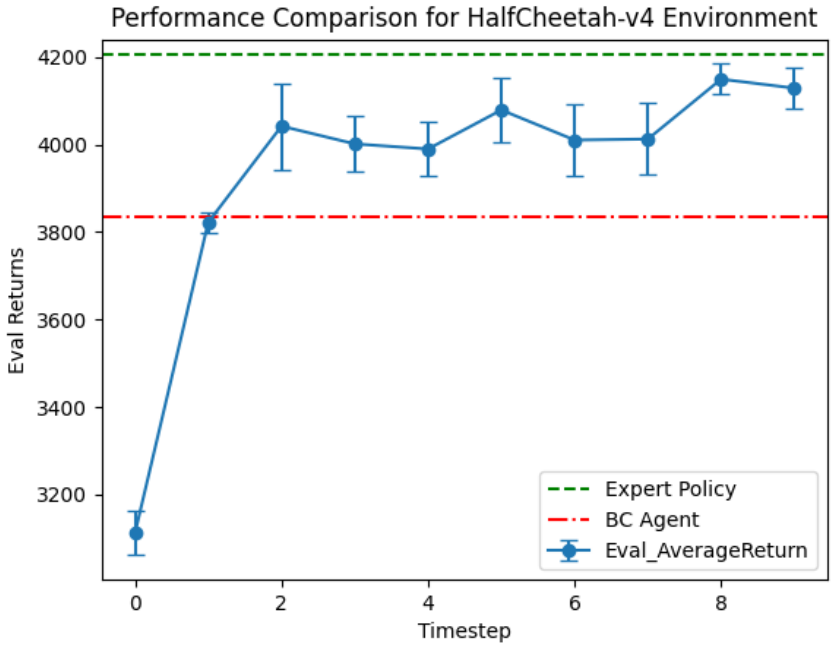
2.b

These are the results for Ant-v4:



Hyperparameter	Value
n_iter	10
num_agent_train_steps_per_iter	1000
batch_size	10000
train_batch_size	100
eval_batch_size	5000
n_layers	2
size	64

These are the results for HalfCheetah-v4:



Hyperparameter	Value
n_iter	10
num_agent_train_steps_per_iter	1000
batch_size	10000
train_batch_size	100
eval_batch_size	5000
n_layers	2
size	64