BC

2.

Table 1

Task	Eval_AverageReturn	Eval_StdReturn	Train_AverageReturn	Percentage of expert
Ant	3160.151611328125	546.2410888671875	4713.6533203125	67%
Hopper	1012.318115234375	202.26747131347656	3772.67041015625	27%

For ant, the hyperparameters are: ep_len=1000, num_agent_train_steps_per_iter=1000, batch_size=1000, eval_batch_size=5000, train_batch_size=1000, max_reply_buffer_size=1000000, Network: 3 layers of size 64, learning_rate=5e-3.
For hopper, the hyperparameters are exactly the same because I thought the difficulty is roughly the same as ant because the ant needs to walk with four legs while hopper needs one, however, hopper needs to figure out how to jump which is slightly more complicated than walking.
Number of iterations is also 1 for both tasks because we are doing behavioral cloning.
Since my ep_len is 1000 and eval_batch_size is 5000, I am collecting approximately 5 trajectories. For Ant, logged eval_averagereturn is 8, logged eval_stdreturn is 6.3, these are the mean and standard deviation of my policy over these 5 rollouts. For hopper, logged eval_averagereturn is 6.9 and logged eval_stdreturn is 5.3, these are the mean and std of my policy over these 5 rollouts.
3. Ant:

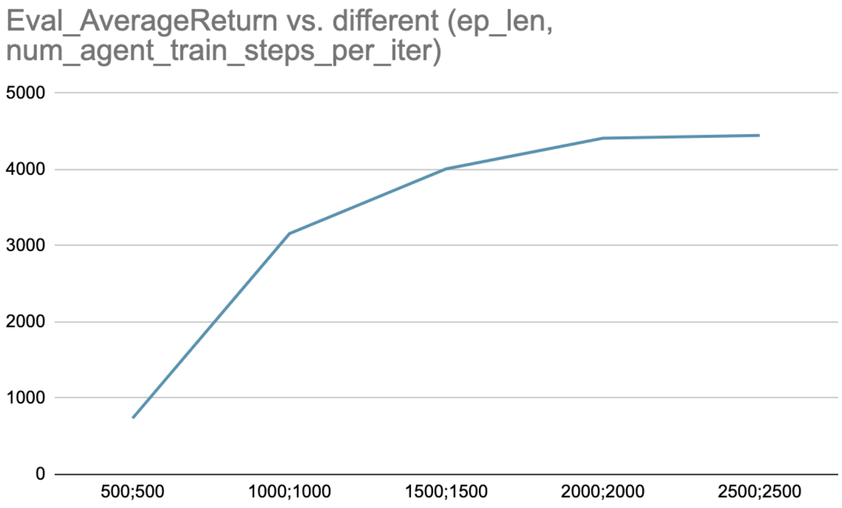

Figure 1

Set of hyperparameters that I’ve decided to try: ep_len and num_agent_train_steps_per_iter because I want to change around the path length and number of agents’s train steps provided as they will affect the learning behavior--the higher these numbers are, the more expert data we are providing, and the better the trainer should perform because the agent is essentially copying form the expert to do simple task of walking without any obstacle, so the more behavior it learns, the better it should perform. I experimented with the following set of hyperparameters:

{ (ep_len=500, num_agent_train_steps_per_iter = 500),
(ep_len=1000, num_agent_train_steps_per_iter = 1000),
(ep_len=1500, num_agent_train_steps_per_iter = 1500),
(ep_len=2000, num_agent_train_steps_per_iter = 2000),
(ep_len=2500, num_agent_train_steps_per_iter = 2500),}.
The performance of BC varies as follows:
[737.4010620117188, 3160.151611328125, 4008.434814453125, 4411.8232421875, 4447.6669921875]
As expected, the more expert data we provide, the better BC performs.

DAgger

2.

Ant

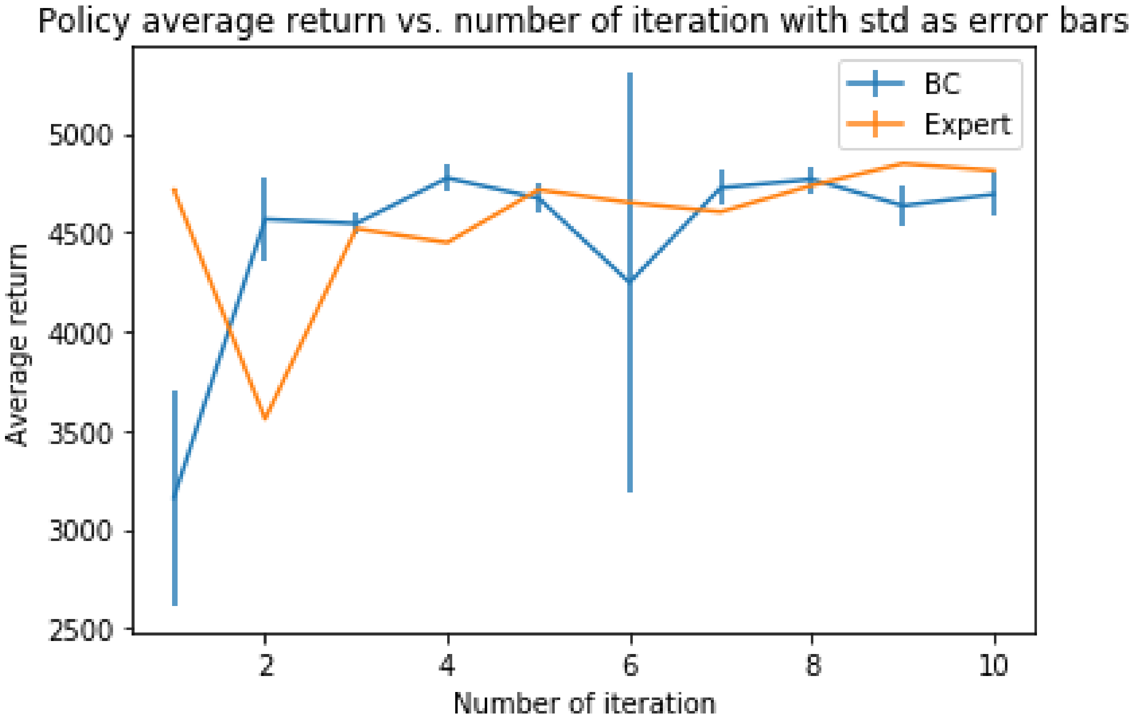

Figure 2.1

Parameters: ep_len = 1000, num_agent_train_steps_per_iter=1000, batch_size=1000, eval_batch_size=5000,train_batch_size=1000,max_replay_buffer_size=1000000, n_layers=3,size=64,learning_rate = 5e-3

HalfCheetah

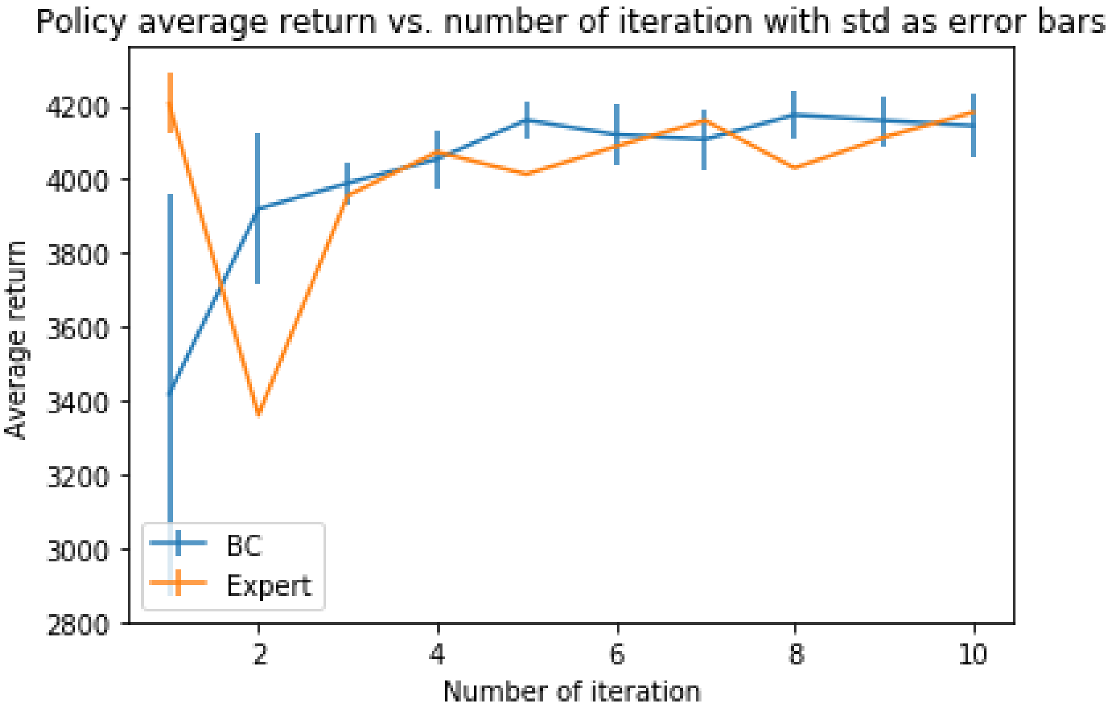

Figure 2.2

Parameters: ep_len = 1000, num_agent_train_steps_per_iter=1000, batch_size=1000, eval_batch_size=5000,train_batch_size=1000,max_replay_buffer_size=1000000, n_layers=3,size=64,learning_rate = 5e-3