Assignment 1: Imitation Learning

Andrew ID: `rlokosso`

1 Behavioral Cloning (65 pt)

1.1 Part 2 (10 pt)

Table 1: Behavior Cloning on every environment for 2 roll-outs

Metric/Env	Ant-v2	Humanoid-v2	Walker2d-v2	Hopper-v2	HalfCheetah-v2
Mean	758.129	262.568	602.374	839.326	3114.354
Std.	76.647	20.849	754.99	278.769	38.313

1.2 Part 3 (35 pt)

To reach 30% at least, I used the following parameters:

- eval batch size = 5000
- num-agent-train-steps-per-iter = 2000
- All the others parameters to their default values

And then for the Ant environment I got 89.32% as accuracy while getting for the environment Humanoid-v2 an accuracy of 2.7%

Table 2: Parameters for 30% on Ants and less on Humanoid: eval batch size = 5000, num-agent-train-steps-per-iter = 2000

Env	Ant-v2		[Another Env (Humanoid-v2)]	
Metric	Mean	Std.	Mean	Std.
Expert	4713.653	12.196	10344.517	20.981
BC	4210.094	111.963	279.567	30.714

1.3 Part 4 (20 pt)

For this experiment, I chose to vary the number of training steps per iteration (num-agent-train-steps-per-iter) for the Behavioral Cloning agent on a given task. This hyperparameter determines how many times the agent updates its policy based on sampled expert data within each training iteration. The graph shows the average returns and standard deviations for both the BC agent and the expert over a range of training steps per iteration, from 1,000 to 19,000 in increments of 1,000. Each data point represents the mean and standard deviation of returns over at least five rollouts, as required (–ep_len=1000 and –eval_batch_size=5000 and all the others parameters with their default values).

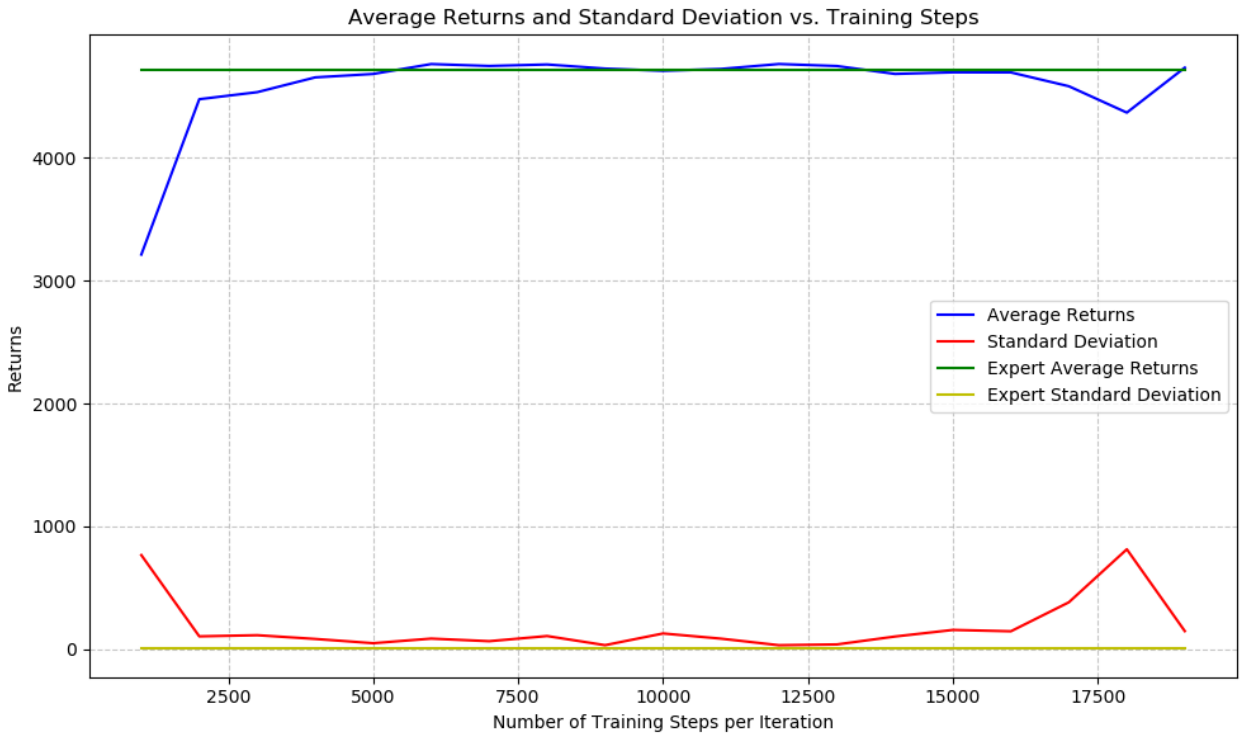


Figure 1: BC agent's performance varies with the value of num-agent-train-steps-per-iter parameter in Ant-v2 environment.

Key observations from the graph:

- The BC agent's performance (blue line) rapidly improves in the early stages of training, stabilizing around 2,500 training steps per iteration.
- The agent's average returns closely approach the expert's average returns (green line) as the number of training steps increases, indicating successful learning.
- The standard deviation of the BC agent's returns (red line) generally decreases as training steps increase, suggesting more consistent performance.
- The expert's performance (green and yellow lines) remains stable across different training step values, as expected.

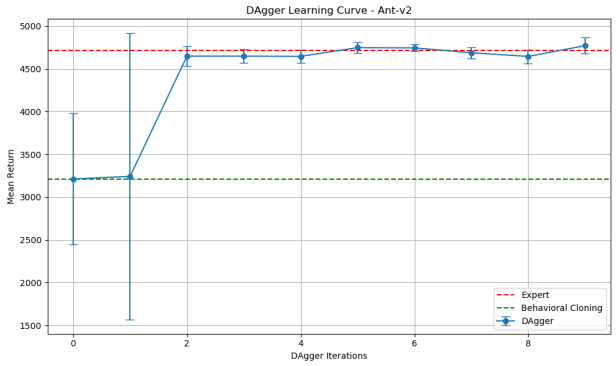
Rationale for choosing this hyperparameter:
I chose to experiment with the number of training steps per iteration because it directly affects how much the agent learns from the expert data in each training cycle. Too few steps might result in undertraining, while too many could lead to overfitting or unnecessary computation. This experiment helps identify the optimal range of training steps needed for the agent to effectively mimic the expert's behavior while maintaining computational efficiency. The results suggest that for this task, around 2,500 to 5,000 training steps per iteration might be optimal, as it achieves performance close to the expert with relatively low variability. Beyond this range, we see diminishing returns in performance improvement.

2 DAgger (35 pt)

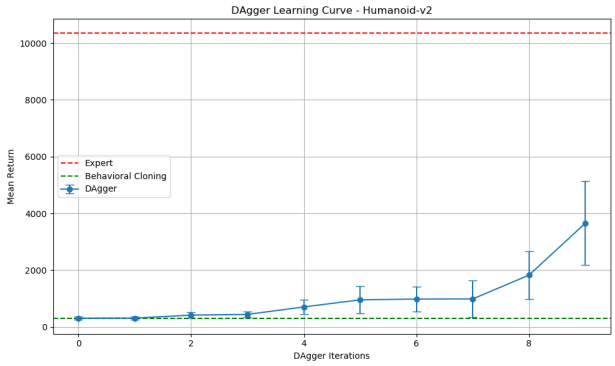
2.1 Part 2 (35 pt)

For this experiment, I implemented and tested the DAgger (Dataset Aggregation) algorithm on two environments: Humanoid-v2 and Ant-v2. I compared DAgger's performance against both the expert policy and a behavioral

cloning (BC) agent. Here are the learning curves and my analysis:



(a) Ant-v2



(b) Humanoid-v2

Figure 2: DAgger Learning Curves for Ant-v2 and Humanoid-v2 environments. The graphs show the mean return over DAgger iterations, comparing DAgger performance with expert policy and behavioral cloning.

In both environments, I used –ep_len=1000 and –eval_batch_size=5000, –n_iter=10 and all the others parameters have their default values.

- **Ant-v2 Results:**
  - The expert policy achieves a mean return of about 4,750.
  - Behavioral cloning performs reasonably well, with a mean return around 3,250.
  - DAgger's performance is impressive:
    - It starts at the BC level but quickly improves.
    - By iteration 3, it reaches expert-level performance.
    - It maintains or slightly exceeds expert performance for the remaining iterations.
    - The small error bars indicate consistent performance across runs.
- **Humanoid-v2 Results:**
  - The expert policy consistently achieves a high mean return around 10,500.
  - Behavioral cloning performs poorly, with a mean return of only about 500.
  - DAgger shows steady improvement over iterations:
    - It starts close to BC performance but gradually improves.
    - There's a significant jump in performance between iterations 6 and 8.
    - By iteration 9, DAgger reaches a mean return of about 3,700.
  - The increasing error bars in later iterations suggest more variability as performance improves.
- **Key Observations:**
  1. DAgger is more effective in Ant-v2, quickly matching the expert. In Humanoid-v2, it shows continuous improvement but doesn't reach expert-level within 10 iterations.
  2. The initial BC performance is much better for Ant-v2 than Humanoid-v2, suggesting Humanoid-v2 is more challenging for imitation learning.
  3. DAgger successfully improves upon BC in both environments, demonstrating its effectiveness as an iterative imitation learning algorithm.

4. The complexity of the environment significantly affects the performance of these algorithms. DAgger excels in Ant-v2 while still providing substantial improvements in the more challenging Humanoid-v2 task.

In conclusion, my results show that DAgger is a powerful algorithm capable of improving upon basic behavioral cloning, especially in less complex environments like Ant-v2. For more challenging tasks like Humanoid-v2, DAgger shows promise but might require more iterations or further tuning to match expert performance.