Part 1

Analysis

Consider the problem of imitation learning within a discrete Markov Decision Process (MDP) with horizon T and an expert policy π^*. We gather expert demonstrations from π^* and fit an imitation policy π_θ to these trajectories such that:

$$E_{p_{\pi^*}(s)}\big[\pi_\theta(a \neq \pi^*(s) \mid s)\big] = \frac{1}{T}\sum_{t=1}^{T} E_{p_{\pi^*}(s_t)}\big[\pi_\theta(a_t \neq \pi^*(s_t) \mid s_t)\big] \leq \epsilon,$$

i.e., the expected likelihood that the learned policy π_θ disagrees with the expert π^* within the training distribution p_{π^*} of states drawn from random expert trajectories is at most ϵ.

We are trying the bound the max difference between probability distributions of π^* and π_θ at a given timestep.

$$\sum_{s_t} |p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)|$$

This can be represented as the probability we enter the mistake distribution (policy makes a mistake before time t) times the max difference in probability that states occur at t.

$$\sum_{s_t} |p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)| \leq \bigcup_t (\bigcup_{s_t} (s_T \in \pi^* \wedge \pi_\theta(a \neq pi^*(s_t|s_t))) * \sum_{s_t} |p_{mistake}(s_t) - p_{train}(s_t)|$$

By the Union Bound Inequality, and definition of Expectation

$$\begin{aligned}
&\bigcup_t \left(\bigcup_{s_t} \big(s_T \in \pi^* \wedge \pi_\theta(a \neq \pi^*(s_t \mid s_t))\big) \right) \leq \bigcup_t \sum_{s_t} \big(s_T \in \pi^* \wedge \pi_\theta(a \neq \pi^*(s_t \mid s_t))\big) \\
&\leq \sum_{t=1}^{T} \sum_{s_t} \big(s_T \in \pi^* \wedge \pi_\theta(a \neq \pi^*(s_t \mid s_t))\big) \\
&\quad \leq \sum_{t=1}^{T} E_{p_{\pi^*}(s_t)}\big[\pi_\theta(a_t \neq \pi^*(s_t) \mid s_t)\big] \\
&\qquad \leq T * \epsilon
\end{aligned}$$

We know that $\sum_{s_t} |p_{mistake}(s_t) - p_{train}(s_t)| \leq 2$, since in the worst case the training distribution reaches s_t 100% of the time and the mistake distribution reaches a different state 100% of the time.

$$\begin{aligned}
\sum_{s_t} |p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)| &\leq \bigcup_t \left(\bigcup_{s_t} (s_T \in \pi^* \wedge \pi_\theta(a \neq \pi^*(s_t|s_t))) * \sum_{s_t} |p_{mistake}(s_t) - p_{train}(s_t)| \right) \\
&\leq T \cdot \epsilon * \sum_{s_t} |p_{mistake}(s_t) - p_{train}(s_t)| \\
&\leq 2 \cdot T \cdot \epsilon
\end{aligned}$$

0.1 Problem 2

Consider the expected return of the learned policy π_θ for a state-dependent reward $r(s_t)$, where we assume the reward is bounded with $|r(s_t)| \leq R_{\max}$:

$$J(\pi) = \sum_{t=1}^{T} E_{p^{\pi}(s_t)}[r(s_t)].$$

(a) Show that $J(\pi^*) - J(\pi_\theta) = \mathcal{O}(T\epsilon)$ when the reward only depends on the last state, i.e., $r(s_t) = 0$ for all $t < T$. When $r(s_t) = 0$ for $t < T$, we have:

$$
\begin{aligned}
J(\pi^*) - J(\pi_\theta) &= \sum_{t=1}^{T} E_{p^{\pi^*}(s_t)}[r(s_t)] - \sum_{t=1}^{T} E_{p^{\pi_\theta}(s_t)}[r(s_t)] \\
&= \sum_{t=1}^{T-1} 0 + E_{p^{\pi^*}(s_T)}[r(s_T)] - \sum_{t=1}^{T-1} 0 + E_{p^{\pi_\theta}(s_T)}[r(s_T)] \\
&= E_{p_{\pi^*}(s_T)}[r(s_T)] - E_{p_{\pi_\theta}(s_T)}[r(s_T)] \\
&= \sum_{s_T} p_{\pi^*}(s_T) r(s_T) - \sum_{s_T} p_{\pi_\theta}(s_T) r(s_T) \\
&= \sum_{s_T} \left(p_{\pi^*}(s_T) - p_{\pi_\theta}(s_T)\right) r(s_T) \\
&\leq 2T\epsilon \cdot R_{\max} \quad \text{Using question 1.}
\end{aligned}
$$

(b) Show that $J(\pi^*) - J(\pi_\theta) = \mathcal{O}(T^2\epsilon)$ for an arbitrary reward.

$$
\begin{aligned}
J(\pi^*) - J(\pi_\theta) &= \sum_{t=1}^{T} E_{p_{\pi^*}(s_t)}[r(s_t)] - \sum_{t=1}^{T} E_{p_{\pi_\theta}(s_t)}[r(s_t)] \\
&= \sum_{t=1}^{T} E_{p_{\pi^*}(s_T)}[r(s_T)] - E_{p_{\pi_\theta}(s_T)}[r(s_T)] \\
&= T \cdot (E_{p^{\pi^*}(s_T)}[r(s_T)] - E_{p_{\pi_\theta}(s_T)}[r(s_T)]) \\
&= T \cdot (\sum_{s_T} p^{\pi^*}(s_T) r(s_T) - \sum_{s_T} p^{\pi_\theta}(s_T) r(s_T)) \\
&= T \cdot \sum_{s_T} \left(p^{\pi^*}(s_T) - p^{\pi_\theta}(s_T)\right) r(s_T) \\
&\leq 2T^2\epsilon \cdot R_{\max} \quad \text{Using question 1.}
\end{aligned}
$$

Part 3

1. Report Mean and Standard Deviation of Policy's Return Over Multiple Rollouts in a Table

Table 1: Evaluation Results for Ant.pkl Across 5 Rollouts (Eval Batch Size = 5000)

Metric	**Value**
Eval_AverageReturn	1276.5576
Eval_StdReturn	280.2951
Eval_MaxReturn	1738.5537
Eval_MinReturn	982.6870
Eval_AverageEpLen	1000.0

2. Hyperparameter Chosen

Number of Agent Training Steps Per Iteration

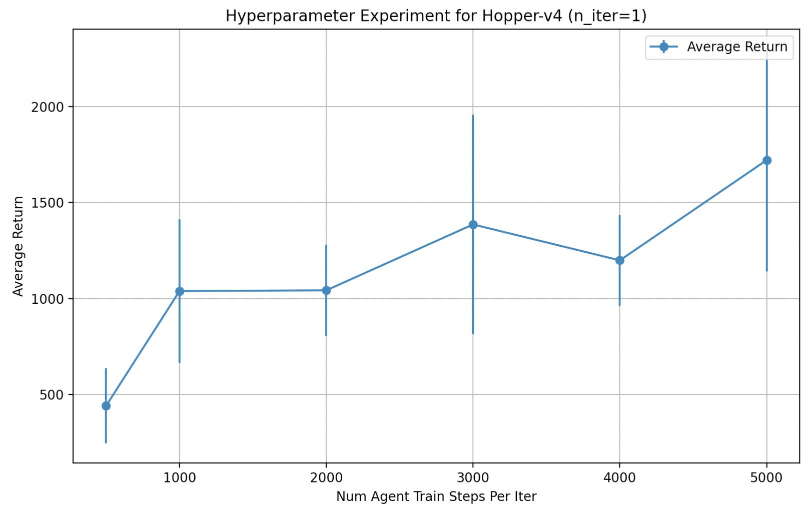

Figure 1: Enter Caption

I wanted to see if increasing the amount of training steps for the behavior cloning would cause overfitting or lead to even better performance.

Part 4: DAGGer

Using `Ant.pkl`, average return across DAGGer iterations.

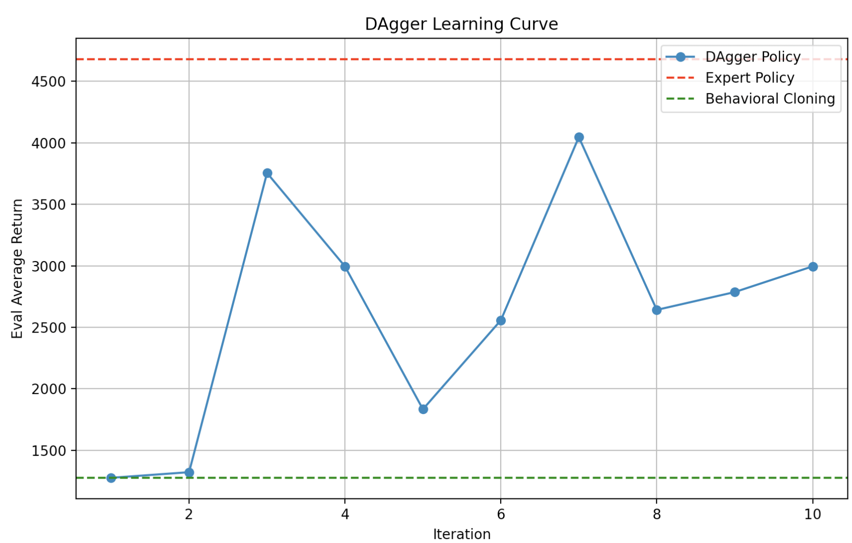

Figure 2: Enter Caption

- Green: Behavioral Cloning
- Red: Expert Policy

Command

The following command was used to run the DAGGer experiment:

```
python cs285/scripts/run_hw1.py \
    --expert_policy_file cs285/policies/experts/Ant.pkl \
    --env_name Ant-v4 \
    --exp_name dagger_ant \
    --n_iter 10 \
    --do_dagger \
    --expert_data cs285/expert_data/expert_data_Ant-v4.pkl \
    --video_log_freq -1
```