## Introduction

This homework is designed to follow up on the lecture about Deep Imitation Learning. For this assignment, you will need to know about the imitation learning algorithms we talked about in the class, in particular DAgger., If you have not already, we propose you brush up on the lecture notes.

**You are allowed to discuss this homework with your classmates. However, any work that you submit must be your own -- which means you cannot share your code, and any work that you submit with your write-up must be written by only you.**

## Code folder

Find the folder with the provided code in the following google drive folder: <https://drive.google.com/drive/folders/1T8B3gSNWjQU-JpifHkEDm9FfoxJA6wB_?usp=sharing>. Download all the files in the same directory, and then follow the instructions in the env\_installation.md file, and then dagger\_template.py file.

## Submission

Please submit your homework using this google form link: <https://forms.gle/c4DSHR8LYzfMPT1SA>

Deadline for submission**: September 17, 2021, 11:59 PM Easter Time.**

## Points

* Questions 1 is 10 points.
* Question 2 and 3 are 5 points each.
* Bonus question: 5 points.
* Total: 20 points (max 25 with bonus).

## 

## DAgger

In the class, we learned about the DAgger (dataset aggregation) algorithm, which is used to clone an expert policy. This method is quite useful especially when querying the expert is expensive, and thus we want to learn a policy that is almost as good as the expert without the high number of queries to it.

In this homework, we have provided you with an environment that is hard to learn directly. Thankfully, we have access to an expert in this environment. **In this homework, your task will be to utilize *DAgger* to learn a deep neural network policy that performs well on this task.**

### Environment

The environment we will use in this homework is built upon the Reacher environment from OpenAI gym (<https://gym.openai.com/envs/Reacher-v2/>). We have provided our environment in the reacher\_env.py file in our code directory. It follows the OpenAI gym API, which you can learn more about at <https://github.com/openai/gym#api>. For this homework, an agent in this environment is considered successful if it can achieve a mean reward of at least 15.0.

In this homework, we will attempt to learn this agent from image observations. Unfortunately, learning this agent directly from images without any priors is incredibly difficult, since images can be from a very high dimensional space. Thankfully, we have access to an expert prediction for any state the environment is currently on, which can be retrieved by the get\_expert\_action() function call. **Note: get\_expert\_action() does not take any arguments, thus you must be careful to call it right after you have called .reset() or .step() on the environment to get the associated expert action.**

### Question 1

Download the code folder, with every file associated, from here <https://drive.google.com/drive/folders/1T8B3gSNWjQU-JpifHkEDm9FfoxJA6wB_?usp=sharing> Complete the code template provided in dagger\_template.py, with the right code in every TODO section, to implement DAgger. Attach the completed file in your submission.

### Question 2

Create a plot with the number of expert queries on the X-axis, and the performance of the imitation model on the Y-axis. Elaborate if you see any clear trends here. (Hint: in the env, the variable expert\_calls counts the number of expert queries.

### Question 3

Could you potentially improve on the number of queries to the expert made by the DAgger algorithm? Think about when querying the expert may be redundant.

**Bonus points**: Try implementing your answer from question 3, and generate a query-vs-reward plot similar to question 2 for this implementation. Compare this plot with your answer from Q2. Is there a clear improvement?

## Python environment installation instructions

1. Make sure you have conda installed in your system. [Instructions link here.](https://conda.io/projects/conda/en/latest/user-guide/install/index.html#regular-installation)
2. Then, get the conda\_env.yml file, and from the same directory, run conda env create -f conda\_env.yml. If you don't have a GPU, you can remove the line saying - nvidia::cudatoolkit=11.1.
3. Activate the environment, conda activate hw1\_dagger.
4. Then, install pybullet gym using the following instructions: <https://github.com/benelot/pybullet-gym#installing-pybullet-gym>  
   (New: alternately, just install pybullet-gym from here: <https://github.com/shubhamjha97/pybullet-gym> thanks Shubham!)
5. If you installed it from the official repo, go to the pybullet-gym directory, find this file: pybullet-gym\pybulletgym\envs\roboschool\envs\env\_bases.py and change L29-L33 to the following:   
   self.\_cam\_dist = 0.75

self.\_cam\_yaw = 0

self.\_cam\_pitch = -90

self.\_render\_width = 320

self.\_render\_height = 240

1. If you are still having trouble with training, up the image resize from (60, 80) to something higher.
2. Finally, run the code with python dagger\_template.py once you have completed all the to-do steps in the code itself.