Effects of Pre-training in Non-noisy Environment for Behavior Cloning using Fetch Robot Pick and Place Task

Batuhan Altundas, Yuuna Hoshi, Reagan Kan and Marcus McGuire

CS7648 Interactive Robot Learning - Group 13 Presentation

April 30, 2021

Motivation

- Training on physical robots can be hard
  - Expensive
  - Limited Access
  - Interferes with production line
- Real Life problems often have interference that is challenging to simulate accurately
  - Inconsistent motor power
  - Physical damage

Current Approach

Current Approach to training robots is to use Reinforcement Learning in the simulation environment, followed by an LfD model in the real environment. This is done because it is cheap to train in the Simulated Environment.

This method requires access to an expert to train the Reinforcement Learning Policy in the Simulated World, making it hard for non-experts to safely add new tasks to the Robots.

Our Approach

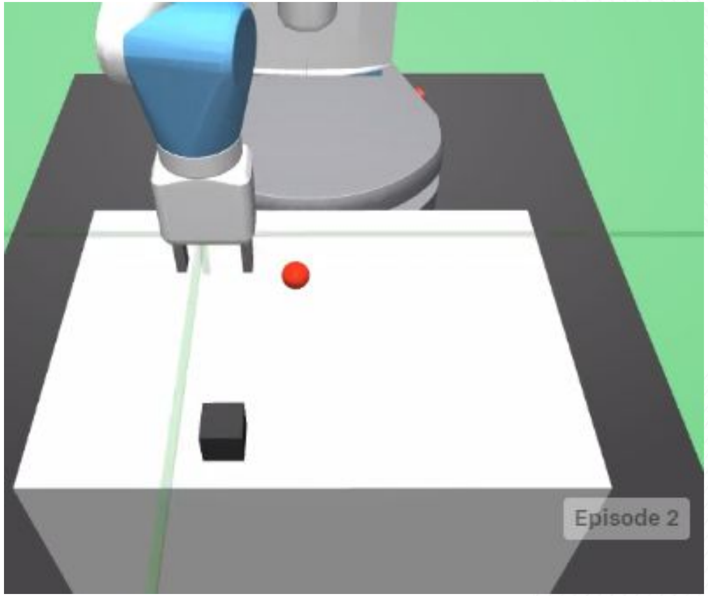
LfD is easier to use by a broader range of expertise than Reinforcement Learning.

Therefore, using an LfD Algorithm in both stages of simulation and real world would allow any user to train complex tasks in a safe, inexpensive and efficient way. Having access to a simulation environment to speed up training also increases the opportunities to train new tasks for the robots compared to simply using real world to train them.

Georgia Tech
CREATING THE NEXT

Project Overview

- **Goal:** Observe if using simulation to pretrain a model reduces the training time in a more complex environment
- **Task:** Pick and Place task using the Fetch robot (OpenAI Gym Module)



Why Fetch? Why Pick and Place?

- Fetch Robot is designed for commercial use, therefore accessibility to the robot is greater than any other robot.
- OpenAI Gym already has a simulation environment that can be used to access Fetch Robot.
- Pick and Place was chosen due to being a common task that has sparse reward space that make Reinforcement Learning inefficient.

Behavior Cloning

- IRL was considered but not used because of the continuous state space. While Keyboard Interface was discrete, the step size was so small that the resulting matrix would be computationally expensive for common use.

Limitations

Unfortunately, we do not have access to an actual Fetch Robot to test.

We were however able to modify the existing Fetch Simulation Environment into creating a more Realistic Environment.

Testing on a Real World Environment with a Fetch Robot is a potential future direction for this research.

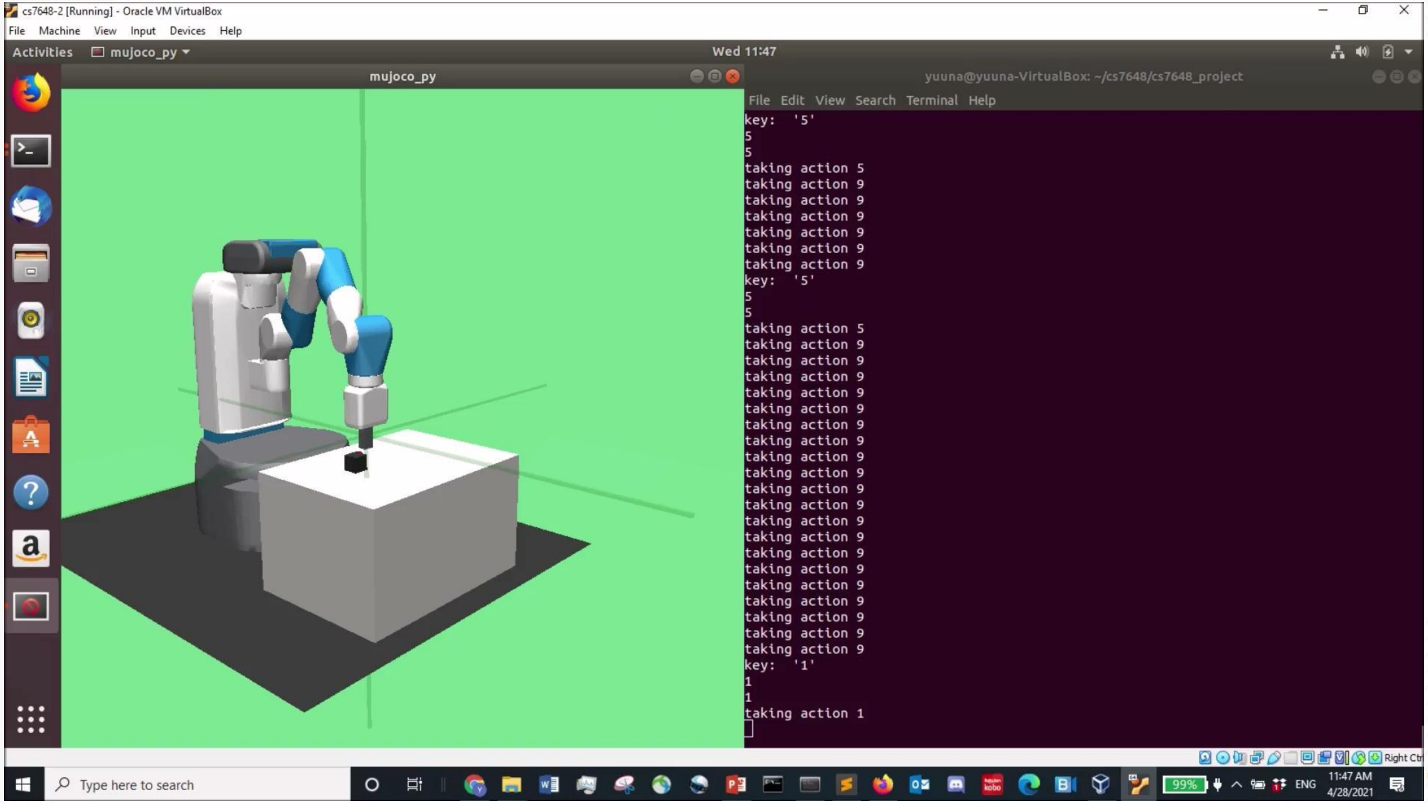
Experiment Environment Setup

- Simulation environment
  - Direct implementation of the Fetch Module
  - 3 linear actuators and 6 joints with actions defined as the location of the end-effector in the form of a gripper.
- Complex Simulation environment
  - Builds on top of the Simulation Environment with the intent of being more realistic
  - Contains multiple stackable Interference Layers that simulate different effects of the real world on the actions.
  - Simulated real-world factors: actuator noise and sensor noise, etc.
    - The end result was a robot that was less reliable than the Model in the Simulation Environment. (It even dropped the blocks!)
    - We were unable to add joint limiters for the arm as joint control was part of the Mujoco Environment that we could not access.

Data Collection

- Demonstration trajectory data is collected by using user keyboard input to control end effector of the Fetch robot, allowing for the training of pick and place task
  - This provided some challenge and had a learning curve. In the future, we are considering something more efficient and intuitive like a 3D Mouse for control.
- Training Data is collected twice, first in the simulation environment and next in the Complex Simulation Environment.
  - Complex Simulation Environment builds on the pretrained Simulation Model to speed up performance

Data Collection



Georgia Tech
CREATING THE NEXT

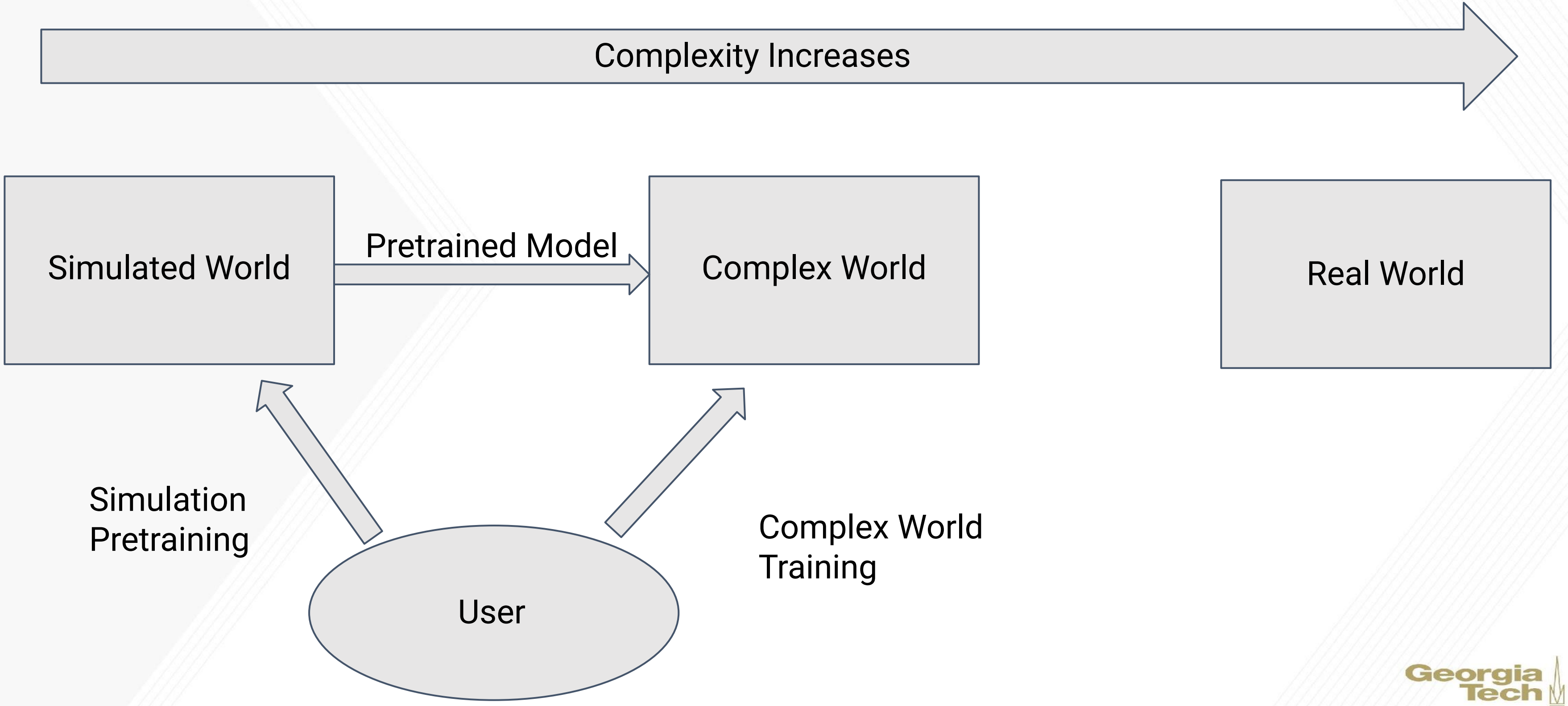
Tasks

Because Behavior Cloning is not good at complicated tasks, we separated the PickAndPlace Task to its components.

Task 1: Pick Up Cube

Task 2: Bring Cube to Target

Training Pipeline

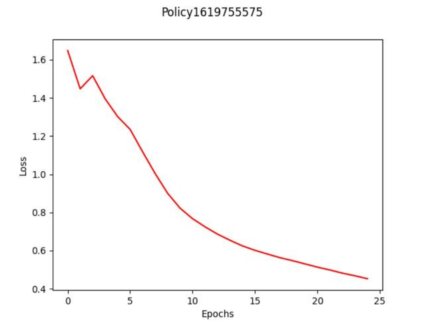


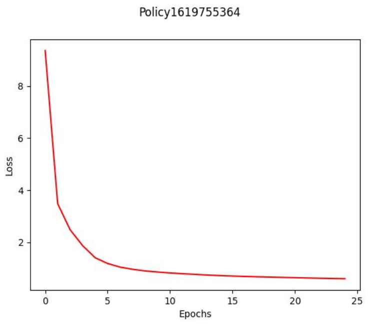
Experiment Design

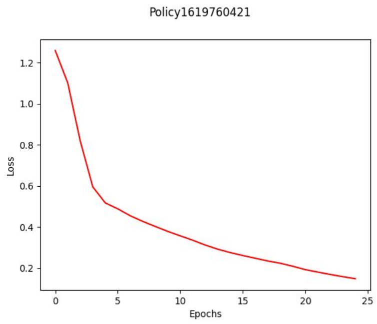
- Collect demonstration trajectories from both Simulated and Noisy environments
- Pre-train using demonstrations from Simulated environment
  - 50 demonstration trajectories
  - Network loss: Cross entropy
- Train using demonstrations from Noisy environment
  - Less demonstration trajectories (10 vs 25)
  - Same network structure

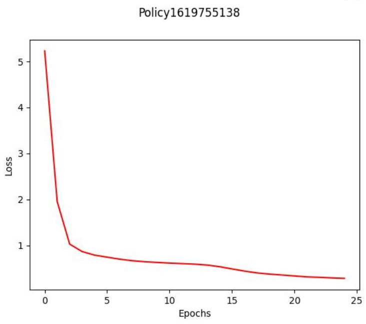
Georgia Tech
CREATING THE NEXT

Sensor Noise Interference





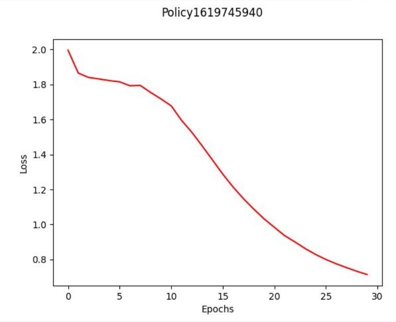


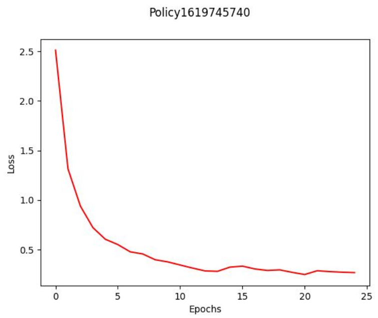


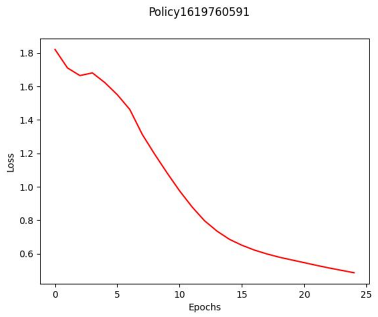
No Pretraining + 10 Demonstrations	Pretraining + 10 Demonstrations	No Pretraining + 25 Demonstration	Pretraining + 25 Demonstrations

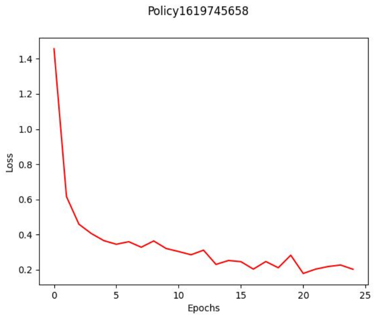
Pretraining: 30 epochs, 50 demonstrations from simulated environment

Actuator Noise Interference (Random Drift)





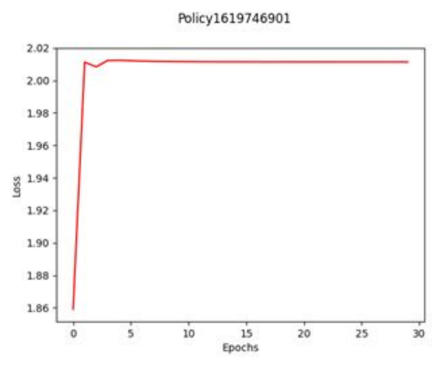


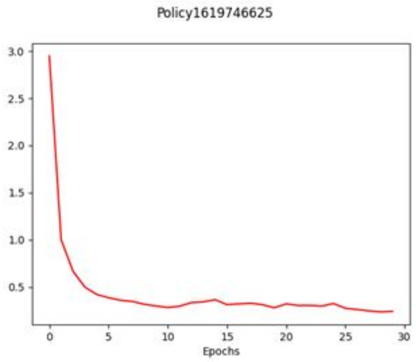


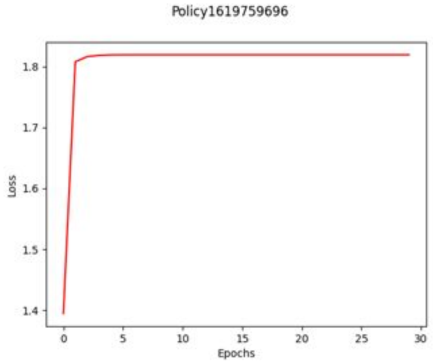
No Pretraining + 10 Demonstrations	Pretraining + 10 Demonstrations	No Pretraining + 25 Demonstration	Pretraining + 25 Demonstrations

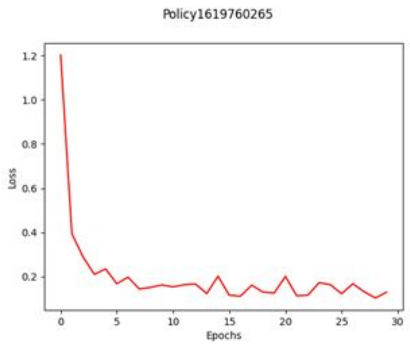
Pretraining: 30 epochs, 50 demonstrations from simulated environment

Speed Interference (Lack of Precise Control)





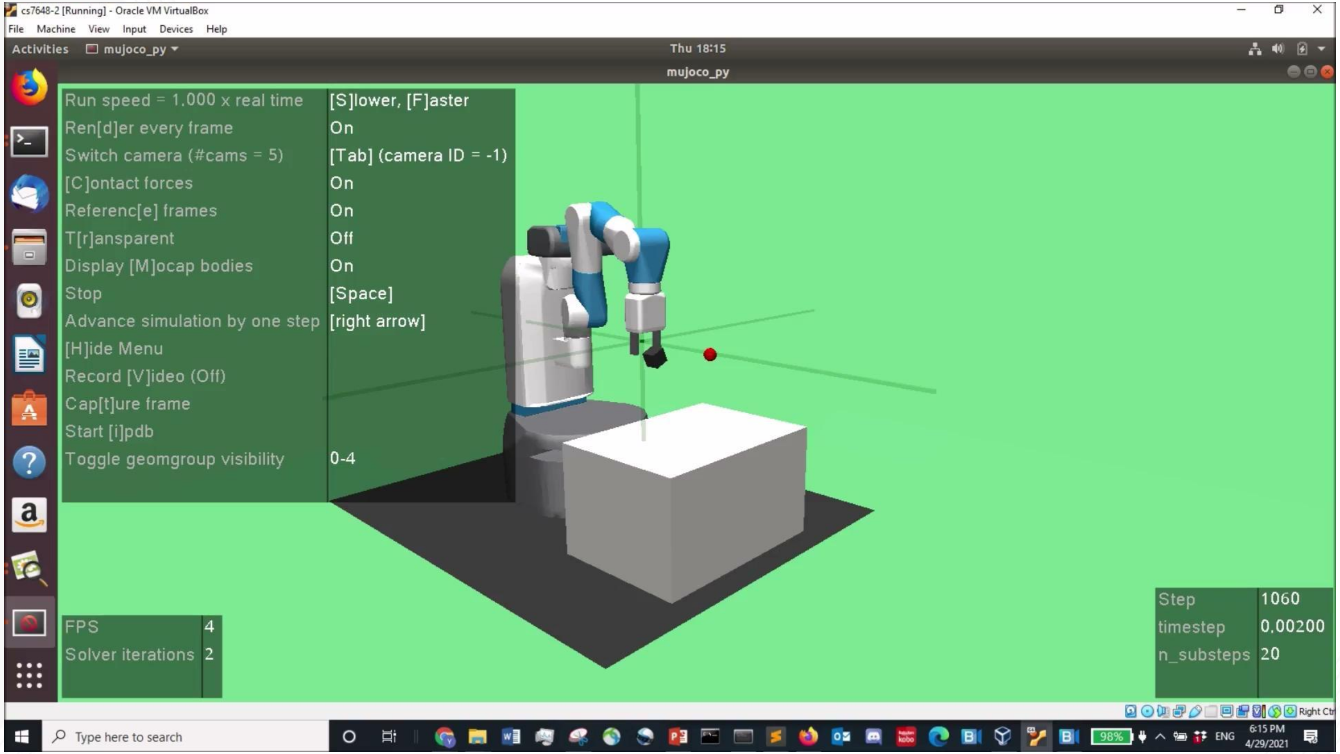




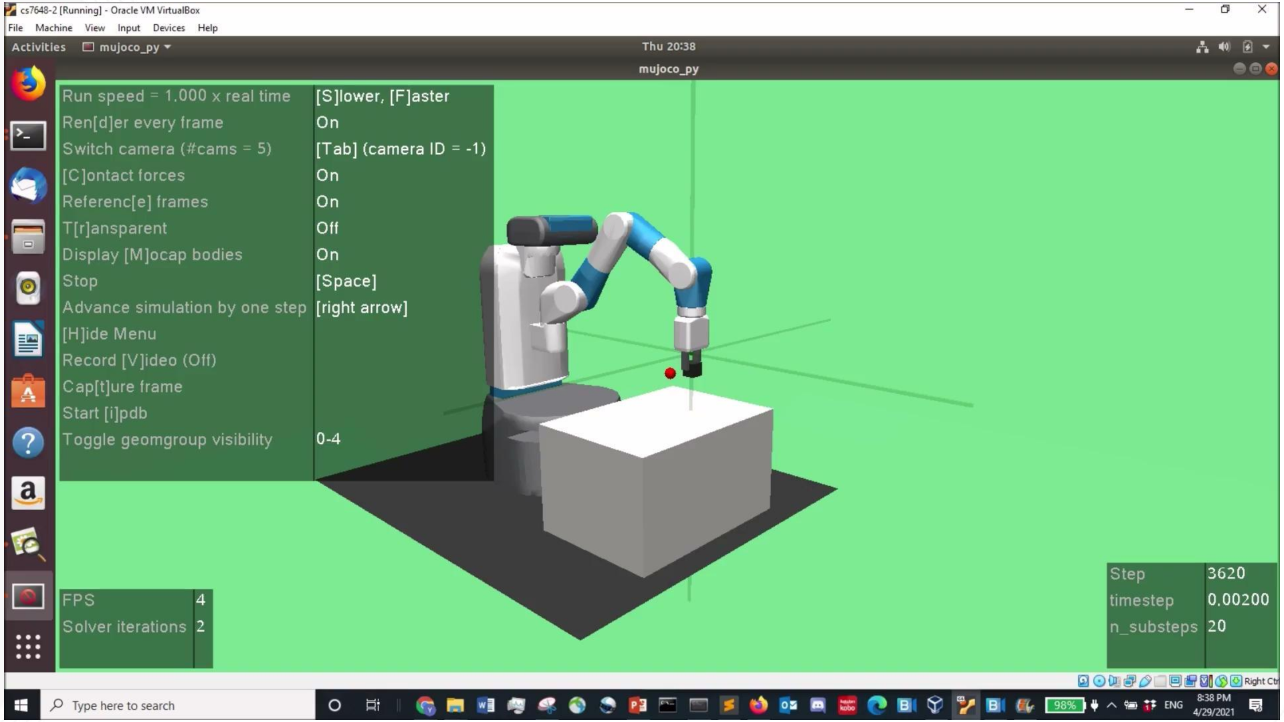
No Pretraining + 10 Demonstrations	Pretraining + 10 Demonstrations	No Pretraining + 25 Demonstration	Pretraining + 25 Demonstrations

Pretraining: 30 epochs, 50 demonstrations from simulated environment

No Pretraining - Drops the Box After Pick Up

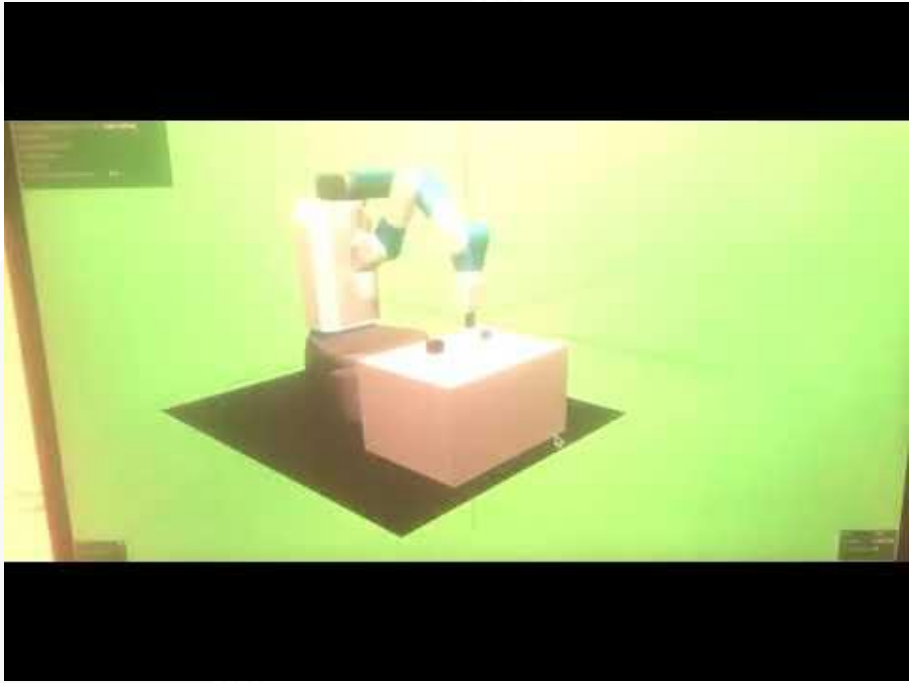


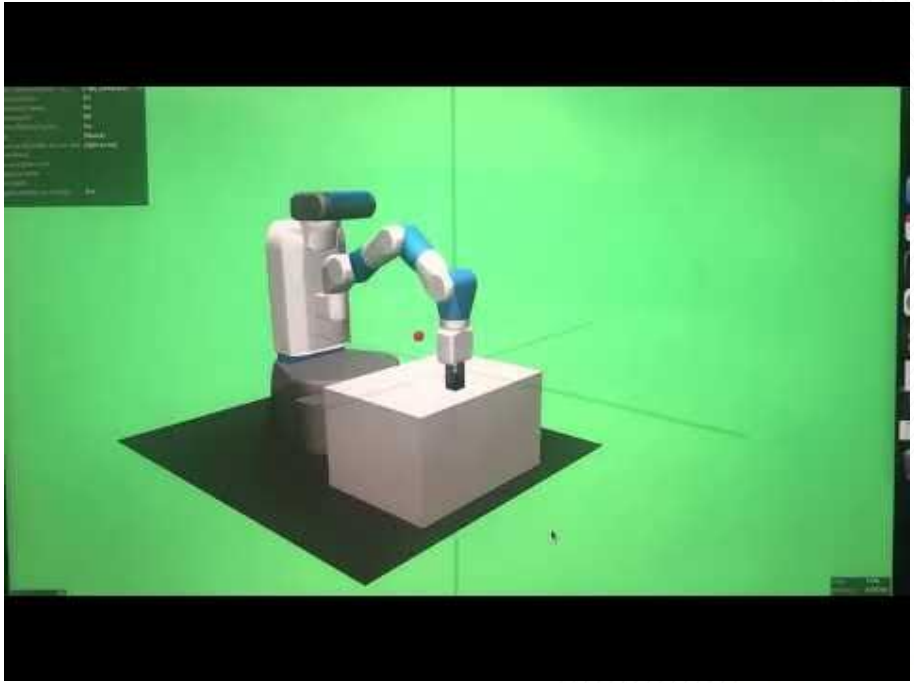
Pretraining only 5 Steps for PickUp Task



Actuator Noise

No Pretraining + **25** Demos vs **Pre**training + **10** Demos





Discussion

We have observed that pretraining can significantly reduce the need for demonstrators in the complex environment. Based on the performance comparison, models with pretraining that are trained with 10 demonstrations in the complex world outperform models without pretraining that are trained with 25 demonstrations in the complex world.

Conclusion

- Demonstration in a Simulated Environment is easier to give than Noisy Environment
- Training in Simulated Environment followed by training in the Noisy Environment provides better results than training just in Noisy Environment
- Learning is far more sensitive to Sensor Noise than Actuator Noise

Potential Future Work

- Use an actual Fetch Robot to test improvements of pretraining in Real World Environment
- Improve the User Interface for the Control Simulator
- Apply the reverse of the principles explored by adding a Noise Filter to a Noisy Environment to create a pre-training on Simplified Environments.

Thank You for Listening

Are there any questions?