Improving Hindsight Experience Replay (HER) on the Fetch Robotics Environments

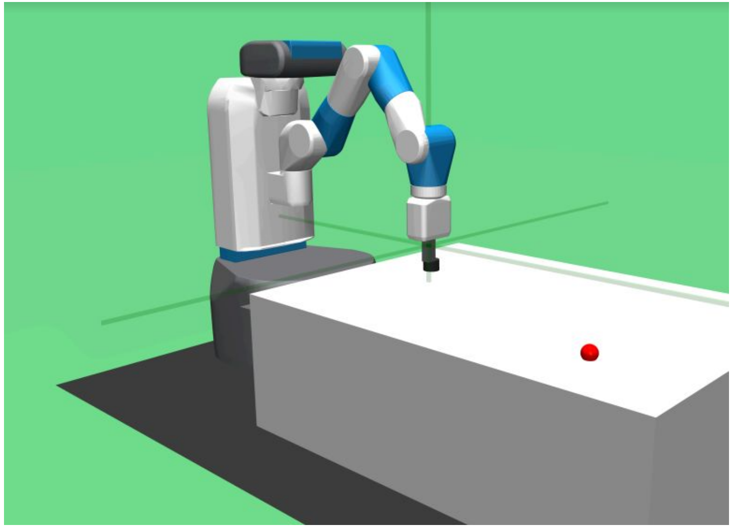
Final Presentation - Dec. 10, 2021

Hanlu Li, Allen Zeng

Fetch Robotics environments

OpenAI Gym & Mujoco physics simulator

- In **FetchSlide-v1,** the robot needs to hit a puck to a randomized target position on a tabletop.
- For **FetchPickAndPlace-v1,** the robot has to pick up a block and move it the target position above the table.



Potential Challenges of the Fetch Environments

- **FetchSlide-v1**
    - Very unlikely for gripper to hit puck, and for puck to stop exactly at the target position
    - Actions after hitting the puck do not matter while the robot waits for the puck to slide to the goal
- **FetchPickAndPlace-v1**
    - The robot needs to move the gripper to the block in the open position, then close the gripper and move the block to the goal

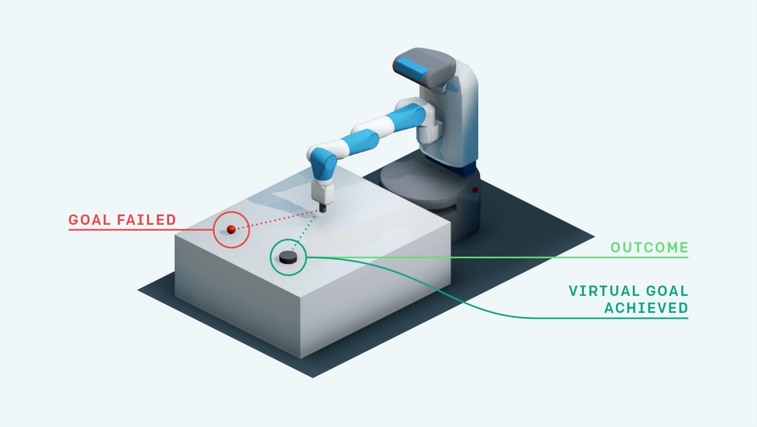
Background

- **Deep Q-Networks (DQN)** - Model-free RL algorithm that is used for discrete action spaces
- **Deep Deterministic Policy Gradients (DDPG)** - Model-free RL algorithm that learns Q functions and a policy at the same time
- **Twin Delayed DDPG (TD3)** - Improves on DDPG by using two Q-functions, delayed policy updates, and target policy smoothing
- **Hindsight Experience Replay (HER)** - Improves on the standard replay buffer by substituting goals in failed episodes so that there is still a learning signal; can be combined with any off-policy RL algorithm

HER & Research Direction

Main idea of Hindsight Experience Replay (HER) is to pretend we reached a goal even if we failed the original task.

Using this substitution, we can still get a learning signal rather than simply -1 at every step.



https://openai.com/blog/ingredients-for-robotics-research/

HER can be combined with any off-policy RL algorithm, such as DDPG.

We train a baseline method with DDPG+HER, then modify it to TD3+HER.

We also modify HER to incorporate demonstrations. This short-circuits the random exploration phase in early epochs of training.

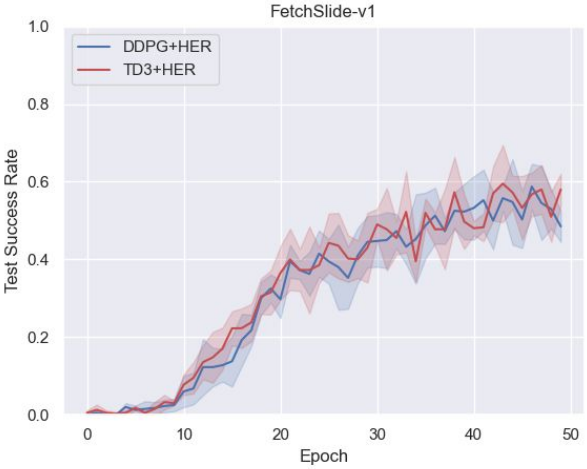
HER with Demonstrations

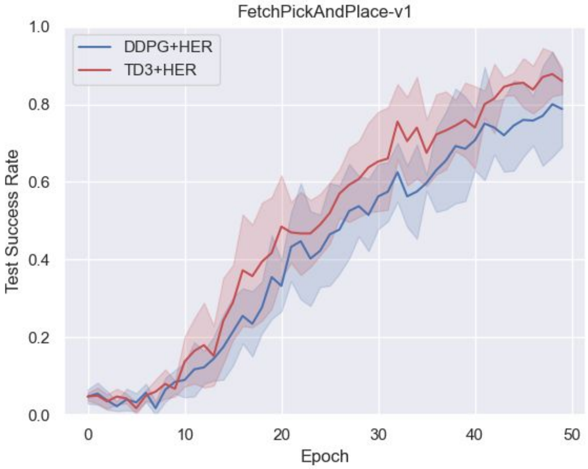
- Collect success test episodes from DDPG+HER and TD3+HER into a demonstration buffer
- Train new agents with demonstrations for the above two algorithms
- Balance RL losses with behavior cloning loss with Q-filter:

$$L_{BC} = \sum_{i=1}^{N_D} ||\pi(s_i|\theta_\pi) - a_i||^2$$

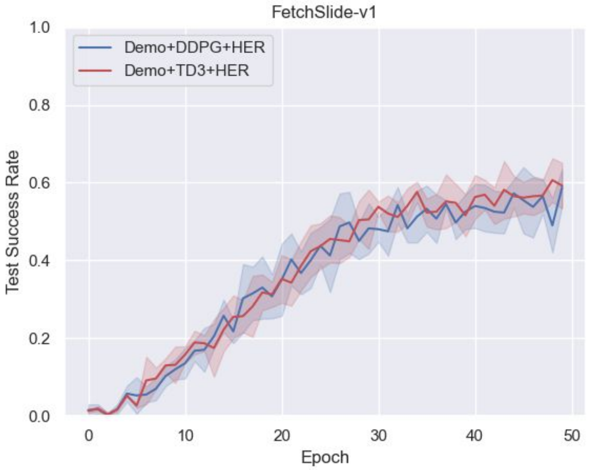
$$L_{BC,QF} = \sum_{i=1}^{N_D} ||\pi(s_i|\theta_\pi) - a_i||^2 \mathbb{1}_{Q(s_i,a_i)>Q(s_i,pi(s_i))}$$

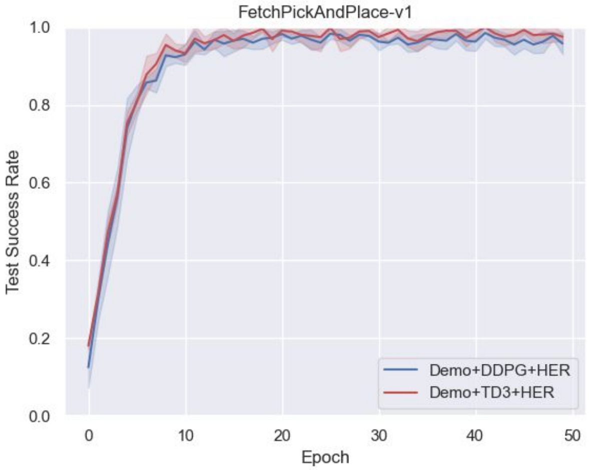
DDPG+HER vs. TD3+HER



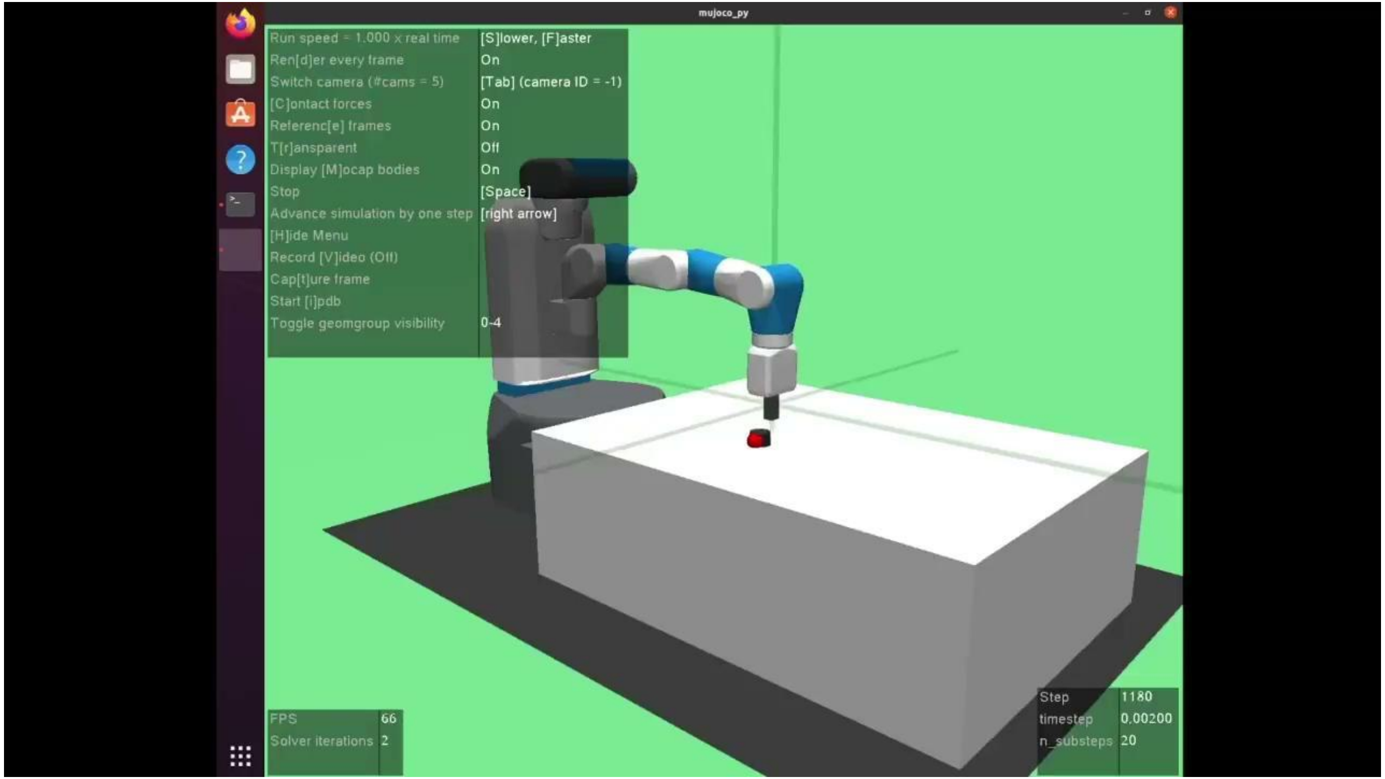


DDPG+HER vs. TD3+HER, with Demonstrations





FetchSlide-v1 Visualization

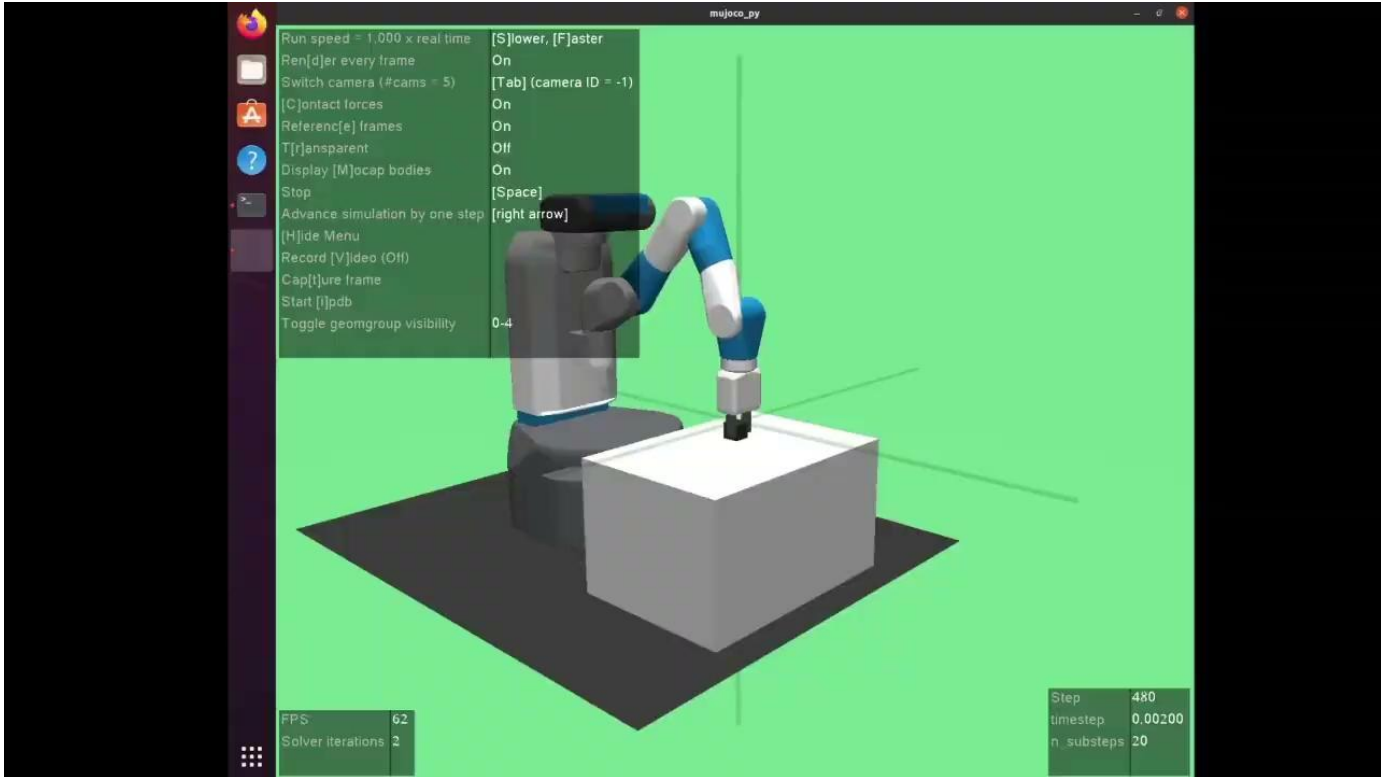


```
Run speed = 1.000 x real time      [S]lower, [F]aster
Ren[d]er every frame               On
Switch camera (#cams = 5)          [Tab] (camera ID = -1)
[C]ontact forces                   On
Referenc[e] frames                 On
T[r]ansparent                      Off
Display [M]ocap bodies             On
Stop                               [Space]
Advance simulation by one step     [right arrow]
[H]ide Menu
Record [V]ideo (Off)
Cap[t]ure frame
Start [i]pdb
Toggle geomgroup visibility        0-4
```

```
FPS                 66
Solver iterations   2
```

```
Step         1180
timestep     0.00200
n_substeps   20
```

FetchPickAndPlace-v1 Visualization



```
Run speed = 1,000 x real time      [S]lower, [F]aster
Ren[d]er every frame               On
Switch camera (#cams = 5)          [Tab] (camera ID = -1)
[C]ontact forces                   On
Referenc[e] frames                 On
T[r]ansparent                      Off
Display [M]ocap bodies             On
Stop                               [Space]
Advance simulation by one step     [right arrow]
[H]ide Menu
Record [V]ideo (Off)
Cap[t]ure frame
Start [i]pdb
Toggle geomgroup visibility        0-4
```

```
FPS                62
Solver iterations  2
```

```
Step         480
timestep     0.00200
n_substeps   20
```

Benefits & Limitations

Pros:

- HER can be paired with any off-policy RL algorithm, and improves with them
    - HER benefits from better RL algorithms that already get close to the goal
- Training with demonstrations, even from RL-trained agents, improve performance

Cons:

- Relies on demonstrating agent to be able to solve the environment on its own first
- Training with demonstrations essentially requires double the data
    - Need to train the base agent, and the agent with demonstrations

Implications & Application to Real World Tasks

This method can help learn many tasks, especially those with sparse and binary rewards, or other difficulties in exploration.

Many tasks in the real world have sparse and binary reward signals. I.e. simply, “has the goal been achieved?”

Thank you!