Recent results

Progress has been made since the internship report was written. In this brief document, I explain this work, the results and my analysis of it.

Interesting results have been obtained with OpenAI Fetch environments using HER[1]. Among these environments, I particularly worked on the one that consists in grasping and moving a rigid cube : `FetchPickAndPlace-v1`. After a few hours of training, the model successfully grasped and moved a rigid cubic object in the majority of cases. Nevertheless, it is very likely that HER is highly dependent on various hyperparameters. In addition, to use HER, it is necessary to have an observation area that indicates the *achieved goal*: in most cases, the state of the robot can correspond to an achieved goal (which does not necessarily correspond to the one initially set). In `FetchPickAndPlace-v1`, this achieved goal is the position of the rigid cube at the end of the episode. Note that this information will not be directly available if the observation space corresponds to the raw data from cameras or sensors. The position of the object will have to be deduced from these data. Deducing the position of an object based on raw data from cameras and sensors is not easy and is a comprehensive subject of study.

In our study, the objective is to capture a soft object, in an environment where the observation space comes from cameras and different sensors. The simulator used eventually tries to reproduce realistic variations of the environment. Applying HER directly on a completely different environment, which does not use the same physics engine, not the same action space, not the same robot... will probably not give good results. This is why I first change only the physics engine and the robot used. This intermediate environment should look as close as possible as the OpenAI Fetch environment, while using our robot (Erika's Panda robot) and a free and open source physics engine (PyBullet[2]).

Panda Environnements

New environments have therefore been developed. Figure 1 shows an illustration of these environments. The similar appearance is deliberate. It allows to visually compare the differences in behavior between the two robots.

Source code and documentation for these environments are available on GitHub at: `https://github.com/quenting44/panda-gym`.

The robot has changed but the action space has remained the same. It is the three coordinates of the target position and the gripper opening (a total of four coordinates). The inverse dynamics that allows to deduce the torques to be applied is done by the PyBullet simulation engine.

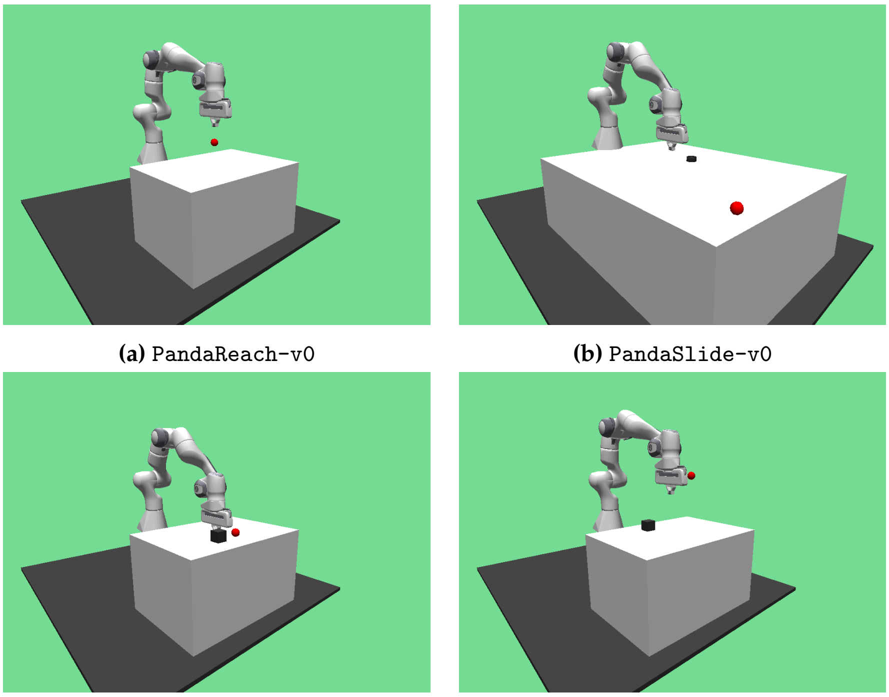
Learning PickAndPlace on our Panda environment

The same learning algorithm is applyied on these new environments. The learning curves are presented on Figure 2.

Learning is faster at the beginning but quickly reaches a maximum value. Other longer training sessions have been ran, but the success rate does not

[1] `https://openai.com/blog/ingredients-for-robotics-research/`

[2] `https://github.com/bulletphysics/bullet3`



(a) `PandaReach-v0`

(b) `PandaSlide-v0`

(c) `PandaPush-v0`

(d) `PandaPickAndPlace-v0`

Figure 1: Panda environments

approach 100%.

How can these differences in learning performance be explained? Several explanations seem credible. First, since the simulation engine has changed (MujoCo to PyBullet), it is likely that some default parameters make the task more complex: the friction model, the default constants, etc. Future work will have to remove this unknown. Second, the robot arm has also changed and therefore the contact surface of the fingers. It is important to note that on the Panda robot simulator, the visually represented contact surface is not the collision surface (the one useful for grasping). For the Fetch robot, the gripping surface is 5.19 cm^2 and for the Panda robot it is 5.56 cm^2. So the Panda robot has a larger gripping surface (always in simulation) than the Fetch robot. Nevertheless, I recently noticed that the collision surfaces corresponding to the panda robot's fingers were not strictly parallel. This could reduce the contact area to two contact segments and greatly complicate the gripping task. A figure of the gripper is shown in the Figure 3.

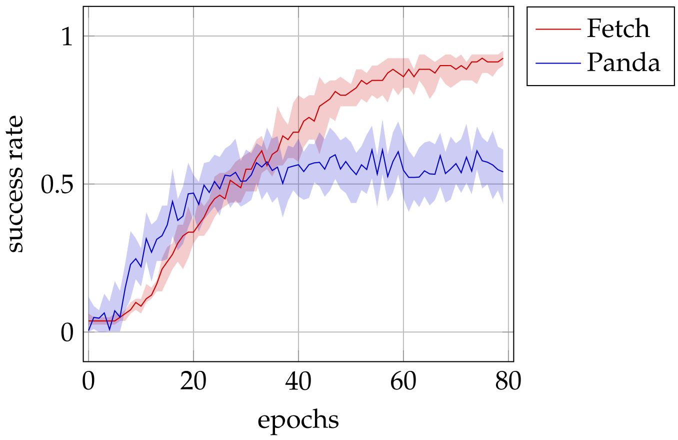


Figure 2: Comparing learning curve for PickAndPlace task between OpenAI Fetch environnement and our Panda environnement

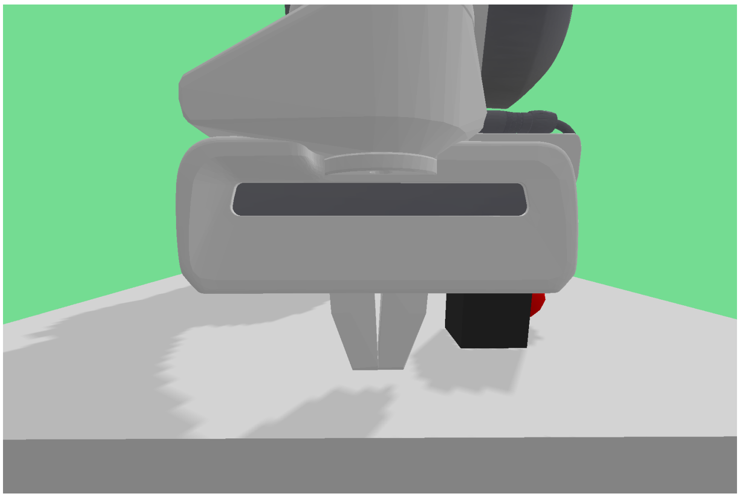


Figure 3: The collision contours of the panda robot's fingers are not parallel in simulation.