CSE276F-Final Project

Zhenyu Wu | PID: A69030822

June 12, 2025

1 Environment Description

1.1 Overview

In this project, I picked the task: "Dump And Place". In the task, the Panda Robot need to first dump the cube placed in the bin to the table, and then move only the bin to the goal position. In general, it has 3 stages:

1. Grasp a predefined grasp point on the bin (one the wall of bin)
2. Try to dump the cube out of the bin while keep grasping the bin
3. If cube is successfully dumped, move the bin to the goal position (while keep grasping).

1.2 Environment Setup

This is a table-top task, the main objects include: a small cube, a bin and a goal position. In the environment, the bin only consists of 3 walls to balance the difficulties, since bin with certain mass consisting 4 walls with certain wall height would be hard for robot to dump out the cube stably.

Environment Initialization When env is initialized, the robot position is fixed. The bin and goal positions are randomized with certain distance from robot base with the work space of the robot (reachable). The cube position would be spawned at the center of the bin. Below are the demonstration of the environment at initialization.

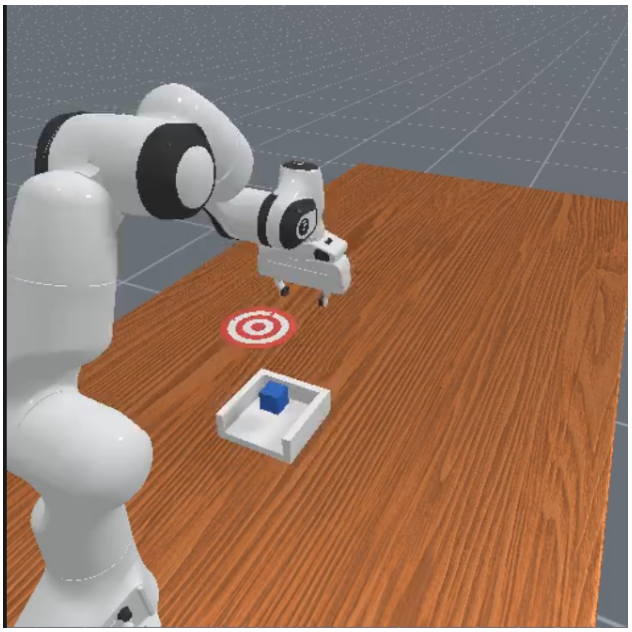

Figure 1: Initialized Env(without back wall)

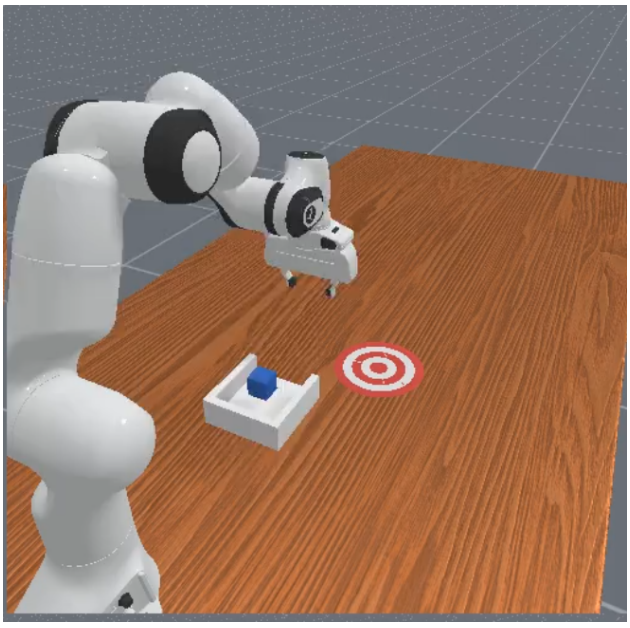

Figure 2: Initialized Env(without front wall)

Like I mentioned, one of the wall of the bin would be removed. I tried two different setups: 1. one removes the front wall, 2. one removes the back wall.

1.3 Enviornment at Complete

Given the initial env setup and task description, if the task is successfully completed, we would observe following scene:

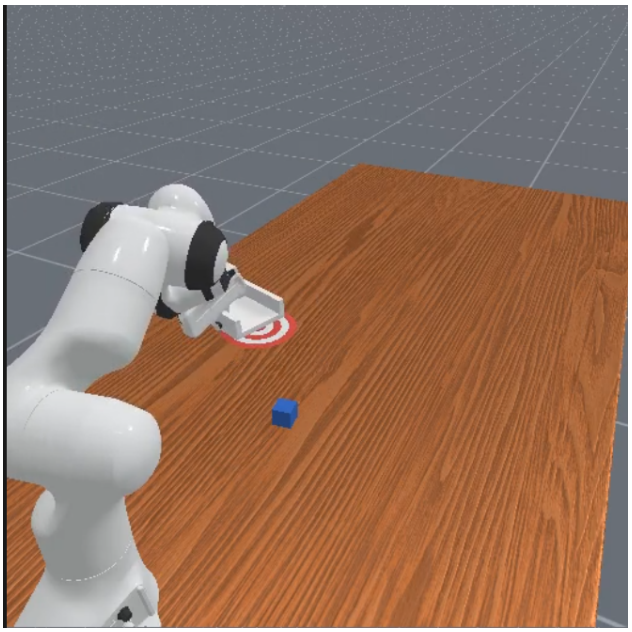

Figure 3: Complete Env (with out back wall)

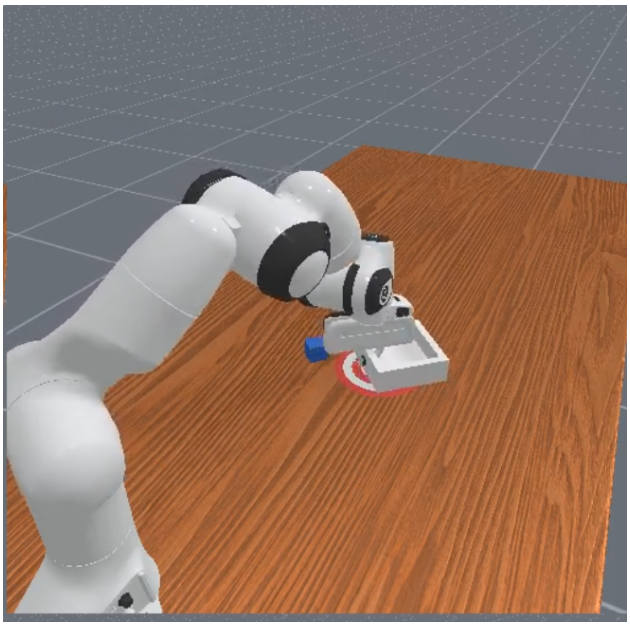

Figure 4: Complete Env (without front wall)

In both cases, the robot would grasp the bin to reach the goal with cube really dumped on the table.

1.4 Success and Fail Condition

Success Condition For the success condition of the task, it is defined as

$$\text{Success} = \text{grasp bin} \wedge \text{cube not in bin} \wedge \text{cube on table} \wedge \text{bin in goal region}$$

Fail Condition In this task, it is difficult to explicitly define a fail condition and assign penalties accordingly, since the task involves multiple stages. Initially, the cube is placed inside the bin, and even after the bin is grasped and lifted—but before the dumping action—the cube may still remain inside the bin. Therefore, judging failure based on intermediate states is nontrivial and potentially misleading.

Although no explicit fail condition was used during training, we can heuristically define a potential failure as:

$$\text{Fail} = \text{grasp bin} \wedge \text{cube in bin} \wedge \text{bin in goal region}$$

This configuration suggests that the agent successfully moved the bin to the goal region but failed to dump the cube, which defeats the task objective.

1.5 Observation

In the observation, I defined the Observation as

$$O = \{q, \dot{q}, p_c, p_b, p_g, p_{grasp}, p_{tcp2grasp}, p_{tcp2bin}, p_{bin2goal}, p_{cube2bin}\}$$

, where q is the robot joint position, $\dot{q}$ is the robot joint velocity, $p_c \in R^3$ is the cube position, p_b is the bin position, p_g is the goal position, p_{grasp} is the grasping position, $p_{tcp2grasp} = p_{grasp} - p_{tcp}$ is the relative position between grasping point and tcp, $p_{bin2goal} = p_g - p_b$ is the relative position between bin and goal, $p_{cube2bin} = p_b - p_c$ is the relative position between bin and cube.

2 Reward and Training

2.1 Reward Design

To facilitate learning, I dived the task into 3 stages as

1. Grasp Bin
2. Dump Cube
3. Bin to Goal

Stage 1: Grasp Bin In this stage, the goal to learn to approach the grasp point of the bin and grasp it. The reward for this stage is defined as:

$$r_{stage1} = r_{approach} + r_{grasp}$$

where $r_{approach}$ is the reward to encourging approaching grasp point which defined as $r_{approach} = 1 - \tanh(5\|p_{\text{grasp}} - p_{\text{tcp}}\|_2)$ which increases as the TCP gets closer to the grasp point, and saturates near 1 when the distance is small, and r_{grap} is a given when robot is grasping the bin.

Stage 2: Dump Cube In this stage, the goal is to dump the cube out from the bin while keep grasping the bin. The reward is defined as

$$r_{stage2} = r_{dump}$$

where

$$r_{\text{dump}} = \mathbb{1}_{\text{cube\_on\_table}} \cdot (1 - \mathbb{1}_{\text{cube\_in\_bin}}) \cdot \mathbb{1}_{\text{is\_grasping}}$$

Here, $\mathbb{1}_{\text{cube\_on\_table}}$, $\mathbb{1}_{\text{cube\_in\_bin}}$, and $\mathbb{1}_{\text{is\_grasping}}$ are indicator functions that equal 1 if the corresponding condition is true, and 0 otherwise. This reward encourages the agent to successfully dump the cube onto the table while still holding the bin.

Stage 3: Bin to Goal In this stage, the goal is to move bin to the goal. THe reward is defined as

$$r_{stage3} = r_{goal}$$

where

$$r_{\text{goal}} = (1 - \tanh(5\|p_{\text{goal}} - p_{\text{bin}}\|_2)) \cdot 8.0 \cdot \mathbb{1}_{\text{is\_grasping}} \cdot (1 - \mathbb{1}_{\text{cube\_in\_bin}}) \cdot \mathbb{1}_{\text{cube\_on\_table}}$$

The indicator functions ensure that the reward is only granted when: the agent is still grasping the bin, the cube has been successfully dumped (i.e., no longer in the bin), the cube is on the table. This encourages completing the final part of the task only after the dumping is done.

Total Reward The Total Reward for this task is defined as:

$$r = r_{stage1} + r_{stage2} + r_{stage3} + r_{success}$$

where $r_{success}$ is the one-time bonus if task is completed.

2.2 PPO On Policy Training

I applied the provided PPO on-policy runner to train the policy. I applied the default hyperparameter come with the PPO runner script (eg. gamma, learning rate...). Below is the visualization of the episode length, return, reward and success rate (**Success Once**) during training.

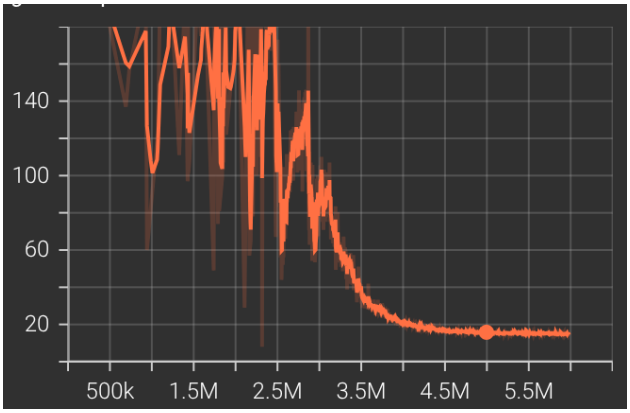

Figure 5: Training Episode Length

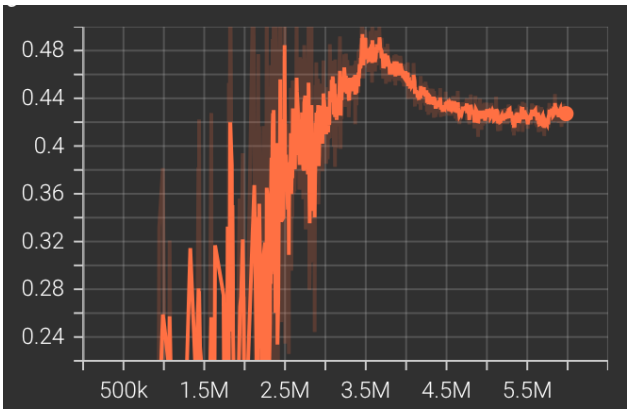

Figure 6: Training Reward

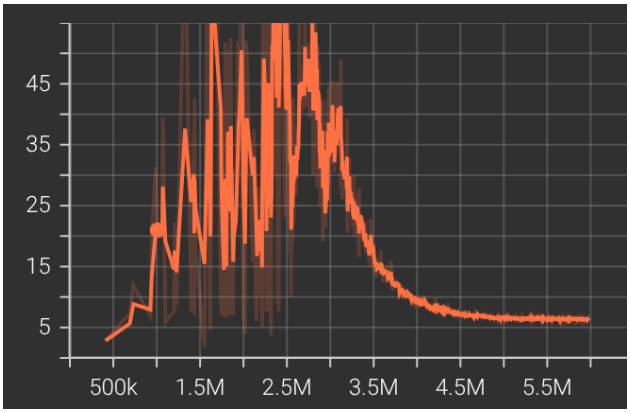

Figure 7: Training Average Return

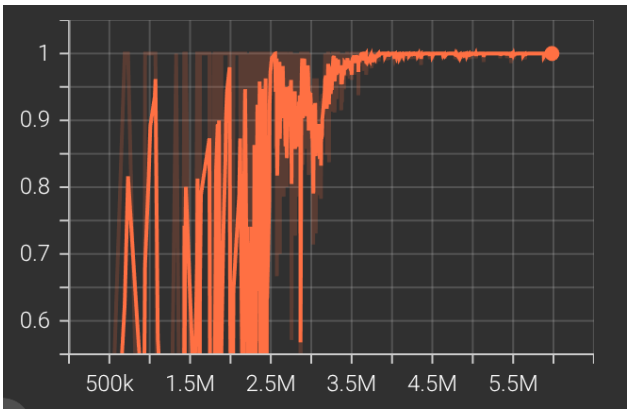

Figure 8: Training Success Rate (**Success Once**)

We can observe that during the early stage of training, when the agent is still exploring, the reward, return, and success rate exhibit noticeable fluctuations. After approximately 3.5 million steps, the policy converges and ultimately achieves a 100% success rate.

Additionally, as the agent learns to solve the task more efficiently, the episode length decreases significantly. This is expected, since successful trajectories terminate earlier. Consequently, the average return also decreases in the later stage of training. This decline is not due to degraded performance, but rather the result of shorter episodes providing fewer opportunities to accumulate step-wise rewards, despite the agent achieving success.

3 Ablation Study

3.1 Overview

In the ablation study, I changed the observation space where **the** $p_{grasp}, p_{tcp2grasp}$ **are removed**. The observation space now is defined as

$$O' = \{q, \dot{q}, p_c, p_b, p_g, p_{tcp2bin}, p_{bin2goal}, p_{cube2bin}\}$$

In addition, I also added noise to observation O'. I specifically, the observation is

$$O' = \{q, \dot{q}, p_c + cw, p_b + cw, p_g + cw, p_{tcp2bin} + cw, p_{bin2goal} + cw, p_{cube2bin} + cw\}$$

where a noise $w \sim N(0, 1)$ is added to all position-related obs, and c is the noise scale.

3.2 Result

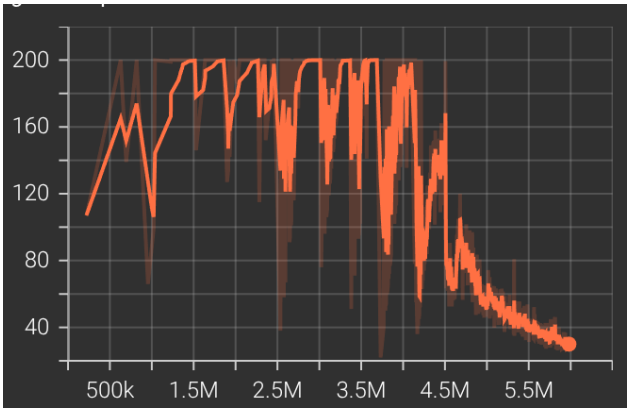

Figure 9: Ablation: Episode Length

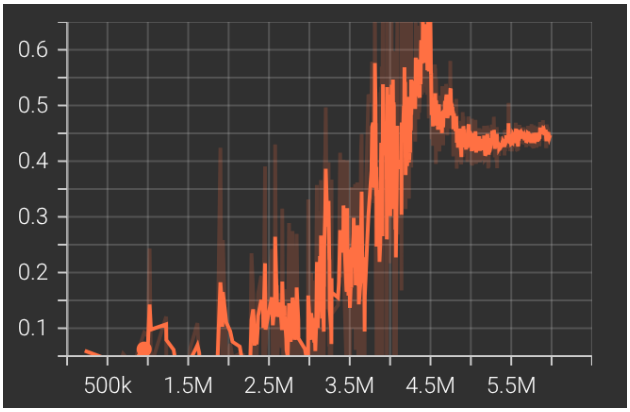

Figure 10: Ablation: Reward

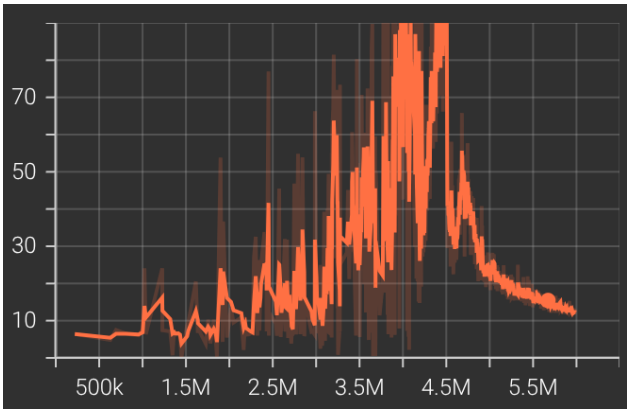

Figure 11: Ablation: Average Return

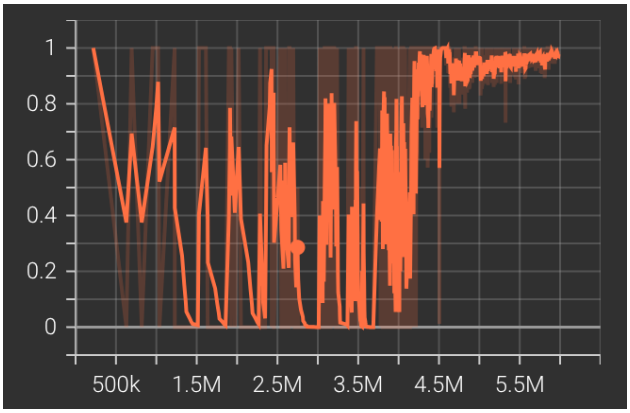

Figure 12: Ablation: Success Rate (**Success Once**)

From the results, we observe that the agent requires significantly more environment steps (up to 5.5M) to reach a stable and high success rate, compared to the original setting where convergence occurred earlier. This degradation in learning efficiency is likely due to the removal of explicit grasp cues, such as the grasp point position and the relative TCP-to-grasp vector. Without these informative features, the agent has to rely on more indirect signals to infer how to approach and manipulate the object, which increases the difficulty of early exploration, and the injection of observation noise also complicates learning by introducing uncertainty in position-related features.

Despite these changes, the policy eventually converges to a competent solution, demonstrating the task is solvable under degraded and noisy observations. However, the learning process becomes less sample-efficient and more unstable in early training as we can observe fluctuating episode length and return curves and agent requires significantly more steps (5.5M) to reach a reasonable success rate compared to the original setting., which suggest less consistent task completion.

4 Code

I included both video and scripts in my submission which contains. The video of my result can be found at /video folder. In addition, to run the scripts, please use following cmd:

Random Agent To run random agent, use

```
python random_agents.py
```

PPO Training To run training, use

```
python train.py --env_id="DumpPlace-v1" --num_envs=2048 --update_epochs=8
    --num_minibatches=32 --total_timesteps=4_000_000 --eval_freq=10 --num-
    steps=20 --num_eval-steps=100
```

On my end, Sometimes I would receive an error regarding to close environment after training complete, Please just ignore that

Policy Evaluation To run evaluation, use

```
python train.py --env_id="DumpPlace-v1" --evaluate --checkpoint=
    CHECKPOINT_PATH --num_eval_envs=1 --num_eval_steps=1000
```