Intelligent Systems and Robotics 2024: Homework 3

In this assignment, you will learn and practice using the Action Trunking Transformers (ACT) to perform imitation learning on robot demonstrations.

Your main task is to implement ACT **Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware** in the Lego pick-place tasks. You have already implemented the low-level Inverse-kinematic algorithm in homework 1, in this homework you learn a neural network to output the target position from vision input and leverage the IK controller to reach the target position.

To begin, follow the step-by-step instructions for installing and setting up the development environments, as well as for configuring ACT. Then you need to finish the codes marked with **#TODO**.

Please note:

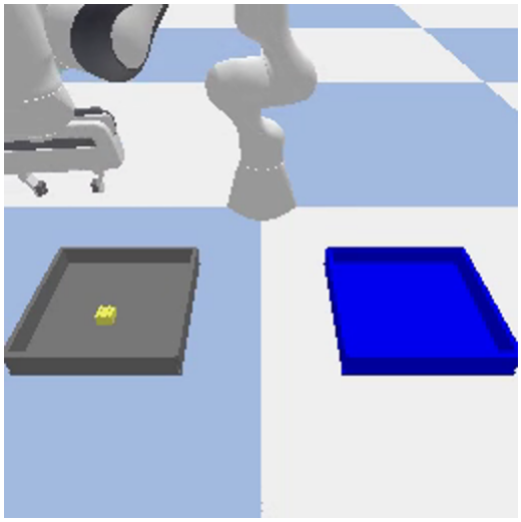
- Before beginning this task, please be aware that you may need to prepare **GPU resources** to complete this assignment. A minimum of 6GB GPU memory is required.
- You can complete this assignment as long as you follow the hints we give, however, we still encourage you to read the ACT paper.
- During implementation, it is beneficial to refer to the corresponding functions in the base class, as most implementations involve making only minor modifications to these functions.
- **Collaboration policy.** Discussions with others are encouraged. However, you should implement the idea on your own.
- You will run at least 1 learning program, which may take a total of **30-40 min on 3060, or 20 min on 3090**. Please take care of the ddl.
- If you have any questions about this assignment, please fell free to contact TA (Chengming Shi, Yanjiang Guo) via WeChat or email scm23@mails.tsinghua.edu.cn, guo-yj22@mails.tsinghua.edu.cn.

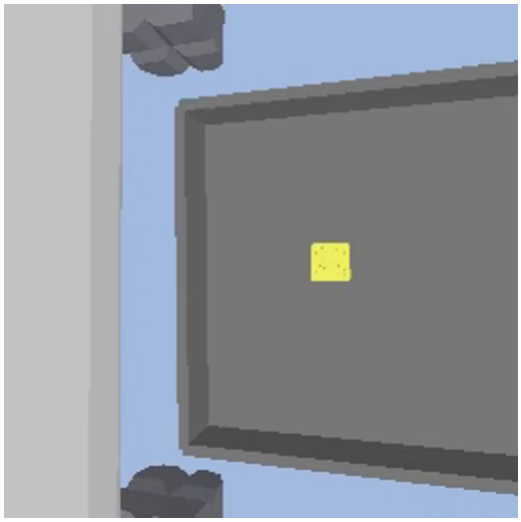
Exercise 1: Implement ACT (70%)

0. Before experiments, run the following line to install the necessary packages:

```
pip install -r requirements.txt
```

Since ACT is an imitation learning algorithm, demonstrations are required. We have collected 50 demonstrations for the Lego pick-place tasks in the folder demo. You can find the videos of demonstrations inside this folder, which contains two camera views.





1. **Implement the Dataset**

Datasets are a necessary part of supervised learning algorithms. We inherit datasets from the pytorch dataset **torch.utils.data.Dataset**. You need to correctly implement the **getitem** function and **len** function.

Please implement all the #TODO blocks in the following files. Some hints are provided along with #TODO blocks.

training/utils.py

2. **Implement part of the ACT backbone**

   The model backbone is the core of supervised learning algorithm. You should finish # TODO block in the following file:

   detr/models/detr_vae.py

3. Train the ACT

   After you finish the datasets and backbone, you can run imitation learning:

```
python train.py --task Lego_0
```

The checkpoints and loss are saved in checkpoints every 200 epochs. You need to run 400 or 600 epochs to converge which takes about 20 min.

Troubleshooting

The training code requires downloading the pre-trained Resnet-18 weight. We include a Resnet-18 weight in the folder. If you find the download process stuck or very slow, you can directly move the provided weight to the download path.

Exercise 2: Implement ACT Rollouts (30%)

1. **Implement rollout function:**

Please implement all the #TODO blocks in the following file. You need to implement the action chunking part which is a weighted average on all actions.

- rollout.py

2. **set the eval_ckpt_name in line 71 of config/config.py and than run:**

```
python rollout.py
```

Then you can see the rollouts in the test folder. If everything is correctly implemented, the yellow Lego can be transformed from the gray box to the blue box.

Submit Your Homework

You only need to include 4 files and two videos to in your submission. The submission should include:

- Rollout videos of two camera views in the test folder.
- rollout.py
- training/utils.py
- detr/models/detr_vae.py
- config/config.py

Please name your zip file with your student ID and Chinese name, e.g. `2023311763-施铖铭.zip`, and submit it through Web Learning.