

Training HIL-SERL using JAKA Arm in LeRobot.

Introduction

LeRobot is a library that aims to provide state-of-the-art machine learning for real-world robotics in PyTorch. It is designed to be a central place for robotics at Hugging Face. The goal is to make it easy for researchers and developers to train and share models for robotics tasks.

This library provides:

- A collection of policies for robotic control.
- Tools for data collection, processing, and visualization.
- Integrations with popular robotics hardware and simulation environments.
- A framework for training and evaluating robotics models.

JAKA S12 Robotic Arm for HIL-SERL

This repository is specifically adapted for integrating the JAKA S12 robotic arm within the LeRobot framework, with a particular focus on Human-in-the-Loop Shared Autonomy Reinforcement Learning (HIL-SERL) applications.

LeRobot provides robust support for the JAKA S12 robotic arm through dedicated modules:

- `lerobot.robots.jakaS12`: This module enables direct control and data acquisition from the JAKA S12 arm. It handles sensor data such as joint and Cartesian positions, integrates camera feeds, manages gripper state, and allows for the execution of precise Cartesian movements.
- `lerobot.teleoperators.jakaS12_leader`: This module facilitates advanced teleoperation by allowing the JAKA S12 arm to serve as a compliant input device. By leveraging admittance control and torque sensing, it generates control signals for a follower robot. This capability is crucial for intuitive human-in-the-loop control, enabling collaborative and complex robotics tasks, especially in the context of HIL-SERL where human guidance is combined with autonomous learning.

Installation

Install Lerobot

From Source

First, clone the repository and navigate into the directory:

```
git clone https://github.com/Dav1nGen/lerobot_jaka.git  
cd lerobot
```

Then, install the library in editable mode. This is useful if you plan to contribute to the code.

```
pip install -e .
```

Extra Features: To install additional functionality(HIL-SERL), use the following:

```
pip install 'lerobot[hilserl]'
```

Troubleshooting

If you encounter build errors, you may need to install additional dependencies: cmake, build-essential, and ffmpeg libs. To install these for linux run:

```
sudo apt-get install cmake build-essential python3-dev pkg-config  
libavformat-dev libavcodec-dev libavdevice-dev libavutil-dev libswscale-dev  
libswresample-dev libavfilter-dev
```

At this point, the installation of the LeRobot library is complete.

Usage

Training a robot with HIL-SERL consists of three steps:

1. Collect data for binary classifier from the robot and train a binary classifier model.
2. Collect human demonstration data.
3. Train a policy with the binary classifier model and human demonstration data.

How To Do It

Attention   : Before running the subsequent scripts, please understand the purpose of the script parameters and set them appropriately.

Step 1: Collect data for binary classifier

Before run collect data scripts, modify `RECORD_MODE = 0` in `./src/lerobot/rl/gym_manipulator.py` to collect **discrete data** for binary classifier.

To collect a dataset, run the following command:

```
python -m lerobot.rl.gym_manipulator --config_path  
./config/collect_reward_classifier_task.json
```

In the terminal where the script is running, you can press '**S**' to indicate a positive sample and '**F**' to indicate a negative sample.



Step 2: Train a binary classifier model with collected data

To train a binary classifier model, run the following command:

```
lerobot-train --config_path ./config/reward_classifier_train_config.json
```

After running lerobot-train script, you can test the trained model with the following command:

```
python -m lerobot.rl.gym_manipulator --config_path  
./config/test_reward_classifier_model.json
```

Test binary classifier model

Step 3: Collect human demonstration data

Before run collect data scripts, modify `RECORD_MODE = 1` in `./src/lerobot/rl/gym_manipulator.py` to collect **continuous human demonstration data** for human demonstration data

To collect human demonstration data, run the following command:

```
python -m lerobot.rl.gym_manipulator --config_path  
./config/record_env_config_jaka.json
```

Similarly, in the terminal where the script is running, you can press '**S**' to indicate a successful human demonstration and '**F**' to indicate a failed human demonstration.

Step 4: Train a policy with the binary classifier model and human demonstration data

1. Starting The Learner

```
python -m lerobot.rl.learner --config_path  
./config/train_config_hil_jaka.json
```

The learner:

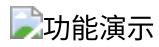
- Initializes the policy network
- Prepares replay buffers
- Opens a gRPC server to communicate with actors
- Processes transitions and updates the policy

2. Starting the Actor

```
python -m lerobot.rl.actor --config_path  
./config/train_config_hil_jaka.json
```

The actor:

- Connects to the learner via gRPC
- Initializes the environment
- Execute rollouts of the policy to collect experience
- Sends transitions to the learner
- Receives updated policy parameters



功能演示