HT_Pi Reinforcement Learning Baseline Technical Document

This document introduces a reinforcement learning environment based on NVIDIA Isaac Gym. Pi_rl_baseline is a sim2sim framework for High Torque Electromechanical's bipedal robot Mini Pi, supporting policy transfer from Isaac Gym to Mujoco. This framework enables users to verify the trained policies in different physics engines, thereby evaluating their robustness and generalization ability.

It is recommended to learn alongside the README of High Torque's open-source project (https://github.com/HighTorque-Robotics/livelybot_pi_rl_baseline) for a more comprehensive understanding.

The following is the introductory tutorial for reinforcement learning by High Torque Electromechanical, suitable for demonstrating to beginners how to train and transfer policies based on a simulation environment:

1. Install Miniconda or Anaconda independently;

2. Create a virtual environment:

```
代码块
1   conda create -n pi_env python=3.8
```

3. Check the NVIDIA graphics card driver:

Use the command `nvidia-smi` in the command line to check the CUDA version of the driver. You can see the CUDA version is 12.8 and the driver version is 570.

```
sunteng@zixiang:~$ nvidia-smi
Tue Jun  3 13:04:55 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 570.133.20             Driver Version: 570.133.20     CUDA Version: 12.8     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce RTX 4060 Ti     Off |   00000000:01:00.0  On |                  N/A |
|  0%   42C    P8             12W /  165W |    1067MiB /  16380MiB |     23%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                      GPU Memory |
|        ID   ID                                                             Usage      |
|=========================================================================================|
|    0   N/A  N/A            1380      G   /usr/lib/xorg/Xorg                      59MiB |
|    0   N/A  N/A            2194      G   /usr/lib/xorg/Xorg                     522MiB |
|    0   N/A  N/A            2330      G   /usr/bin/gnome-shell                    86MiB |
|    0   N/A  N/A            2563      G   .../sunloginclient --cmd=autorun        12MiB |
|    0   N/A  N/A            2707      G   ...me/58.0.3029.81 Safari/537.36         2MiB |
|    0   N/A  N/A            2733      G   ...D4C2B51A7D651DD79862114E7657D         6MiB |
|    0   N/A  N/A            3852      G   /usr/lib/firefox/firefox               193MiB |
|    0   N/A  N/A          346176      G   ...OTP --variations-seed-version        31MiB |
+-----------------------------------------------------------------------------------------+
```

4. Activate the created conda environment:

`conda install pi_env`

```
sunteng@zixiang:~$ conda activate pi_env
(pi_env) sunteng@zixiang:~$
```

5. Install PyTorch:

代码块

```
conda install pytorch torchvision torchaudio pytorch-cuda=12.4 -c pytorch -c nvidia
```

(The selected CUDA version must be less than or equal to the version installed on the computer; CUDA 12.4 is selected here.)

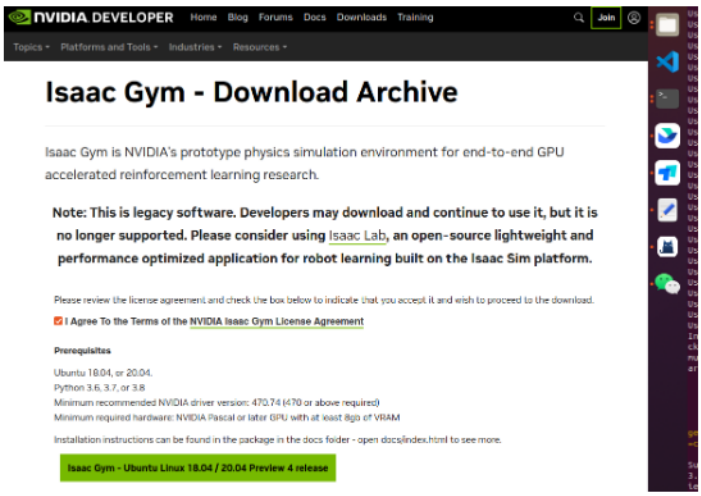
6. Install numpy using conda:

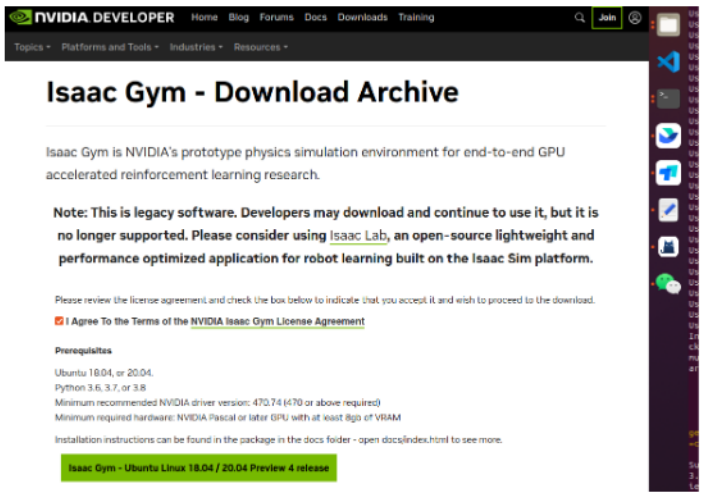
代码块

```
conda install numpy=1.23
```

7. Install Isaac Gym:

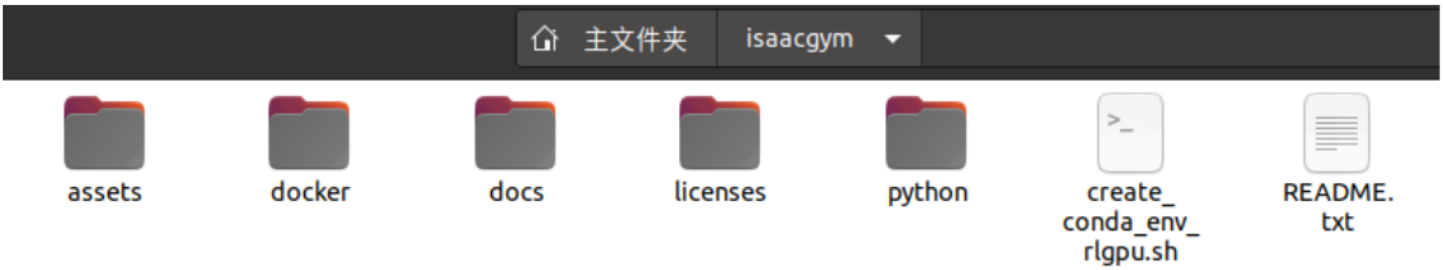
Download and install Isaac Gym Preview 4 from the NVIDIA official website (Note: Only supported on Ubuntu 18.04 or 20.04).





8. Extract to the home directory:

Extract the downloaded Isaac_Gym_Preview_4_package to the home directory, and cut the `isaacgym` package under the subfolder to the home directory.



9. Enter the isaacgym package for installation:

First enter:

```
代码块
1   cd isaacgym/python
```

Then enter:

```
代码块
1   pip install -e .
```

```
(pi_env) sunteng@zixiang:~$ cd isaacgym/
(pi_env) sunteng@zixiang:~/isaacgym$ cd python/
(pi_env) sunteng@zixiang:~/isaacgym/python$ pip install -e .
```

10. Test if the environment installation is successful:

First enter:

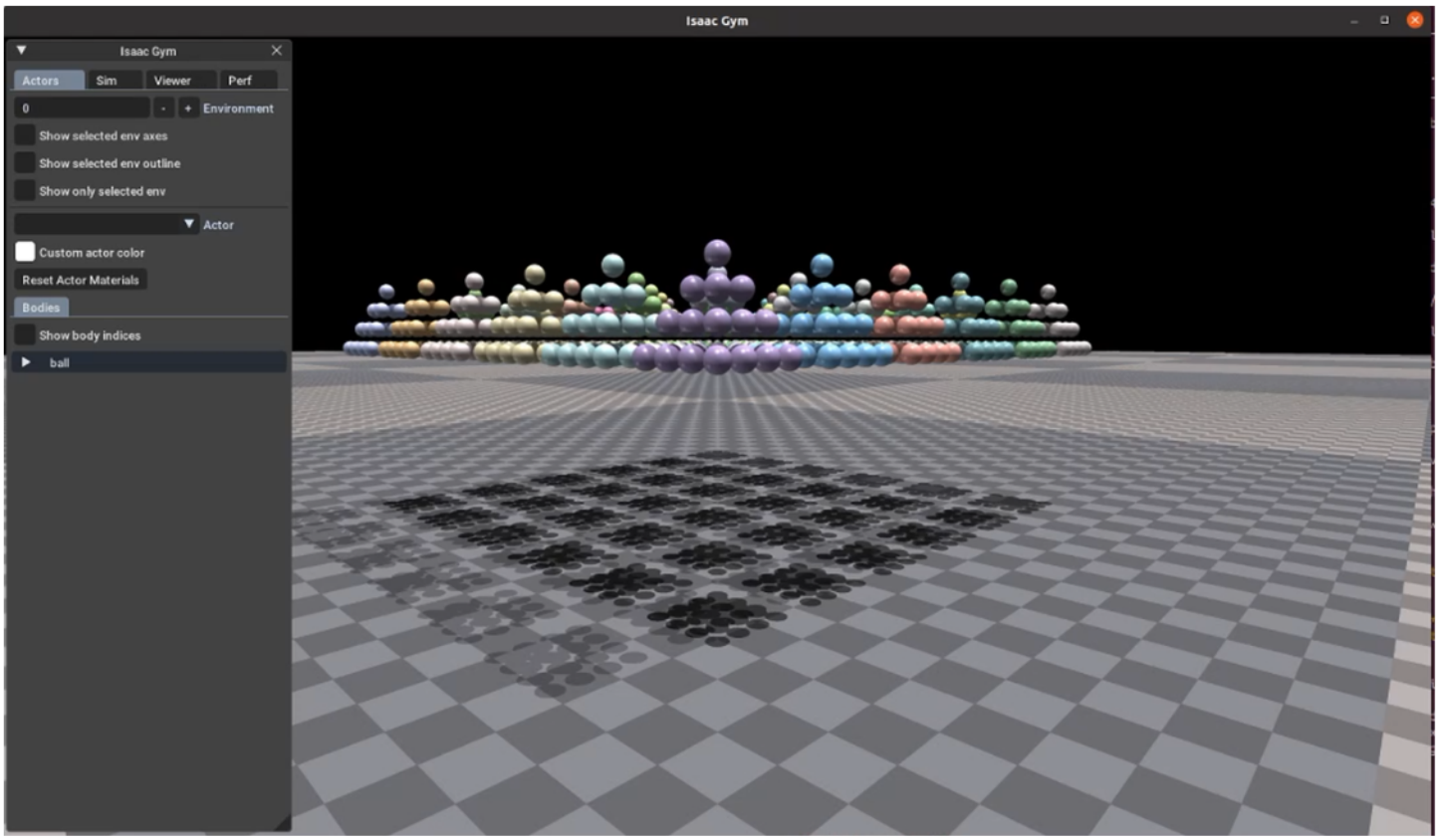
```
代码块
1   cd examples
```

Then enter:

```
代码块
1   python 1080_balls_of_solitude.py
```

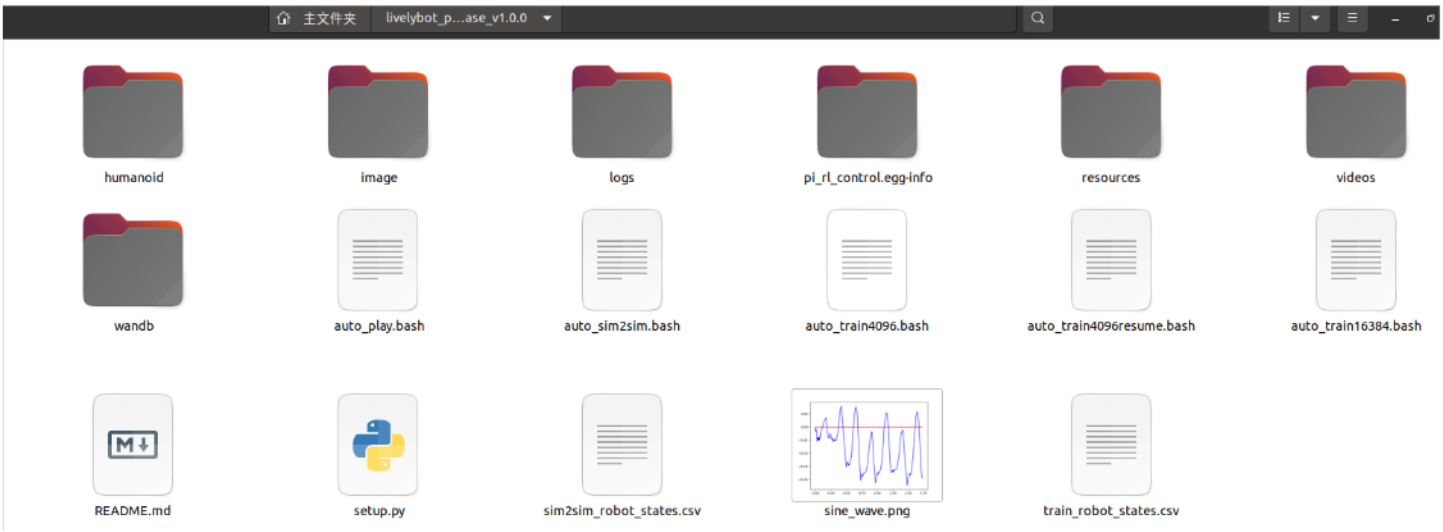
```
(pi_env) sunteng@zixiang:~/isaacgym/python$ cd examples/
(pi_env) sunteng@zixiang:~/isaacgym/python/examples$ python 1080_balls_of_solitude.py
```

If the page shown in the figure below appears, the environment installation is successful.



11. Download the zip package of pi_rl_baseline:

After successful download, extract it to the home directory. Enter `livelybot_pi_rl_baseline-release_v1.0.0(1)` and cut `livelybot_pi_rl_baseline-release_v1.0.0` to the home directory.



12. Install the current project in development mode:

· First enter:

```
代码块
1   cd livelybot_pi_rl_baseline-release_v1.0.0
```

· Then enter:

```
代码块
1   pip install -e .
```

(Note: The '.' must be included here.)

```
(pi_env) sunteng@zixiang:~$ cd livelybot_pi_rl_baseline-release_v1.0.0/
(pi_env) sunteng@zixiang:~/livelybot_pi_rl_baseline-release_v1.0.0$ pip install -e .
```

Finally, a prompt of "successfully installed" for a series of dependencies will appear, indicating successful configuration.

13. Train the PPO policy with 4096 environments and "v1" as the training version:

· First enter:

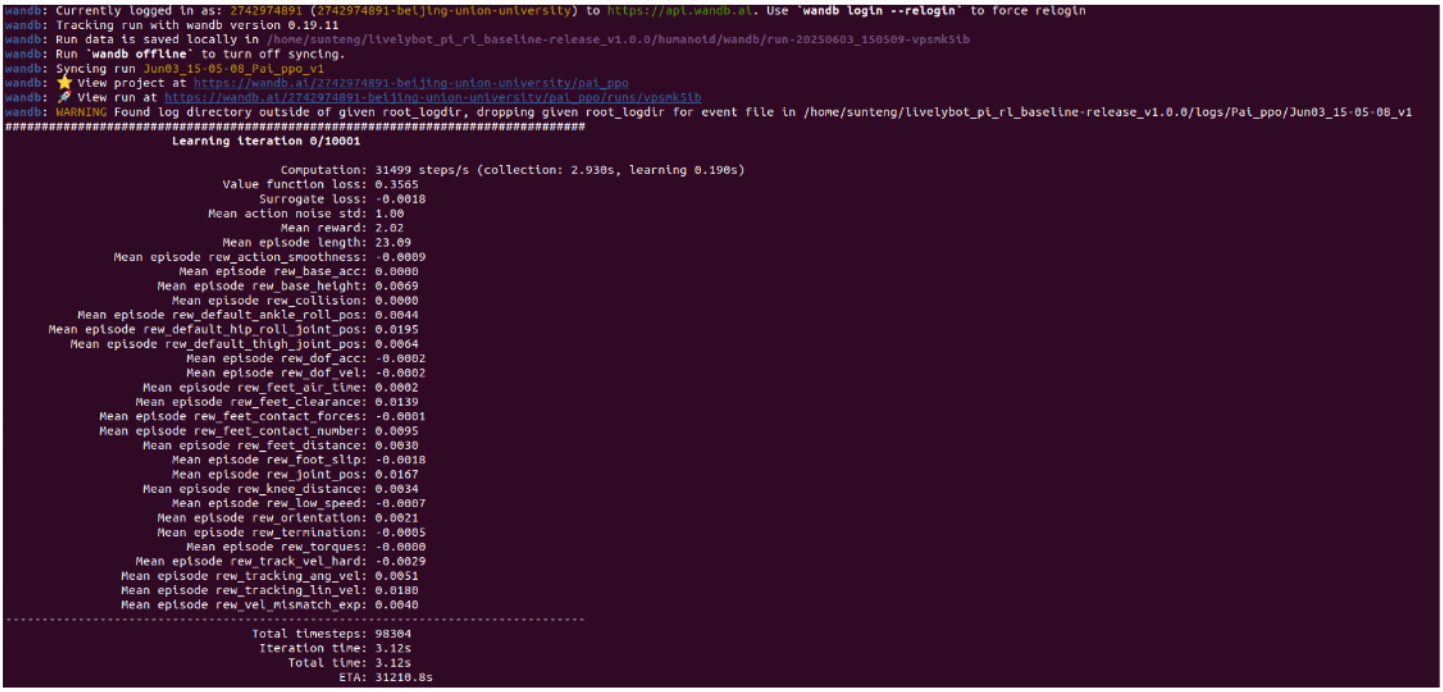
```
代码块
1   cd humanoid
```

· Then enter:

代码块

```
python scripts/train.py --task=pai_ppo --run_name v1 --headless --num_envs 4096
```

If the interface shown in the figure below appears, it means the PPO policy training with "v1" as the training version is in progress.



Note: A wandb registration website will pop up for account registration (wandb is used in Isaac Gym for experiment tracking, hyperparameter recording, and visualization of reward curves for training metrics).

14. Evaluate the trained policy and automatically export a JIT model suitable for deployment:

· First install the onnx package:

代码块

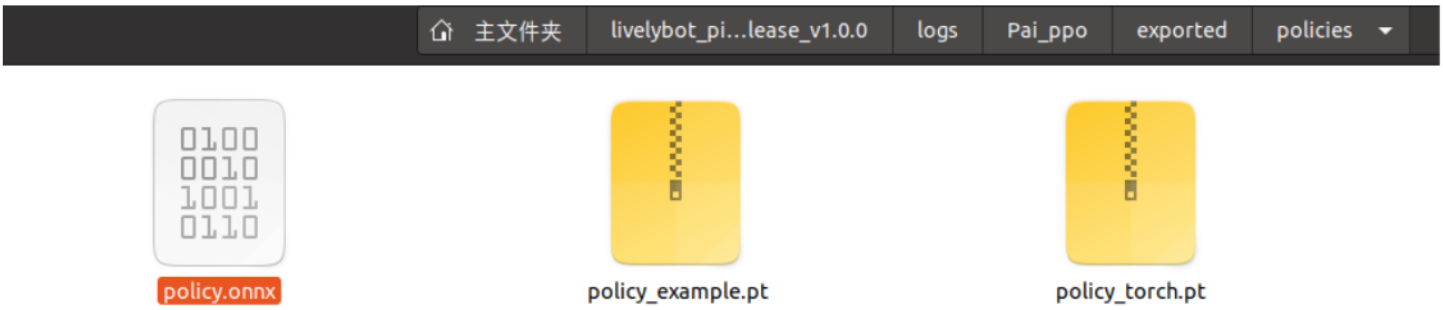
```
pip install onnx
```

· Then enter:

代码块

```
python scripts/play.py --task=pai_ppo --run_name v1
```

You can see `policy.onnx` under `/home/sunteng/livelybot_pi_rl_baseline-release_v1.0.0/logs/Pai_ppo/exported/policies`.



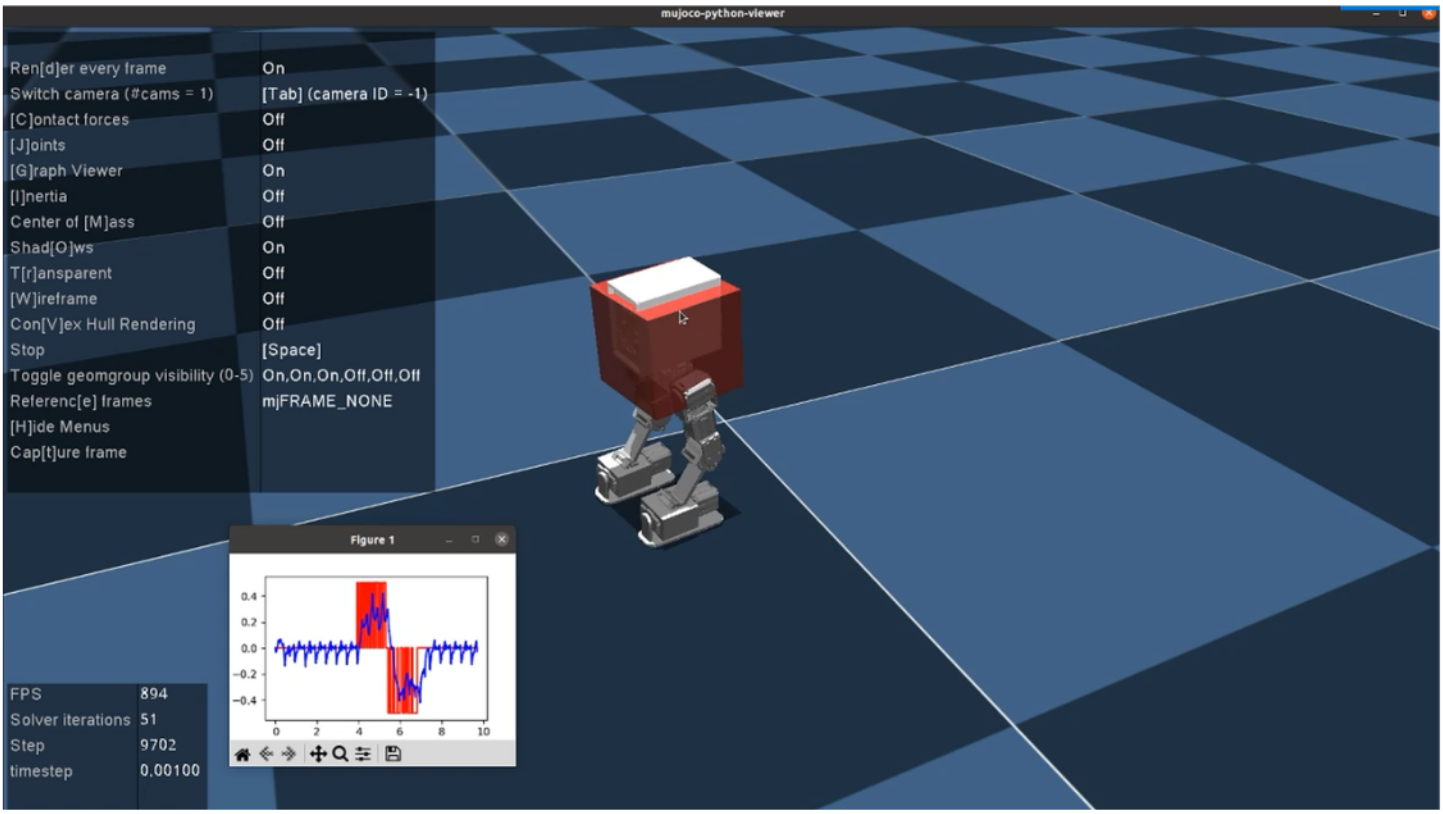
15. Implement sim2sim using Mujoco:

代码块

```
python scripts/sim2sim.py --load_model
/path/to/logs/Pai_ppo/exported/policies/policy_torch.pt
```

(Note: Replace `/path/to` with the path of the parent folder of `logs`.)

If the operation is successful, the Mujoco scene shown in the figure below will be displayed:

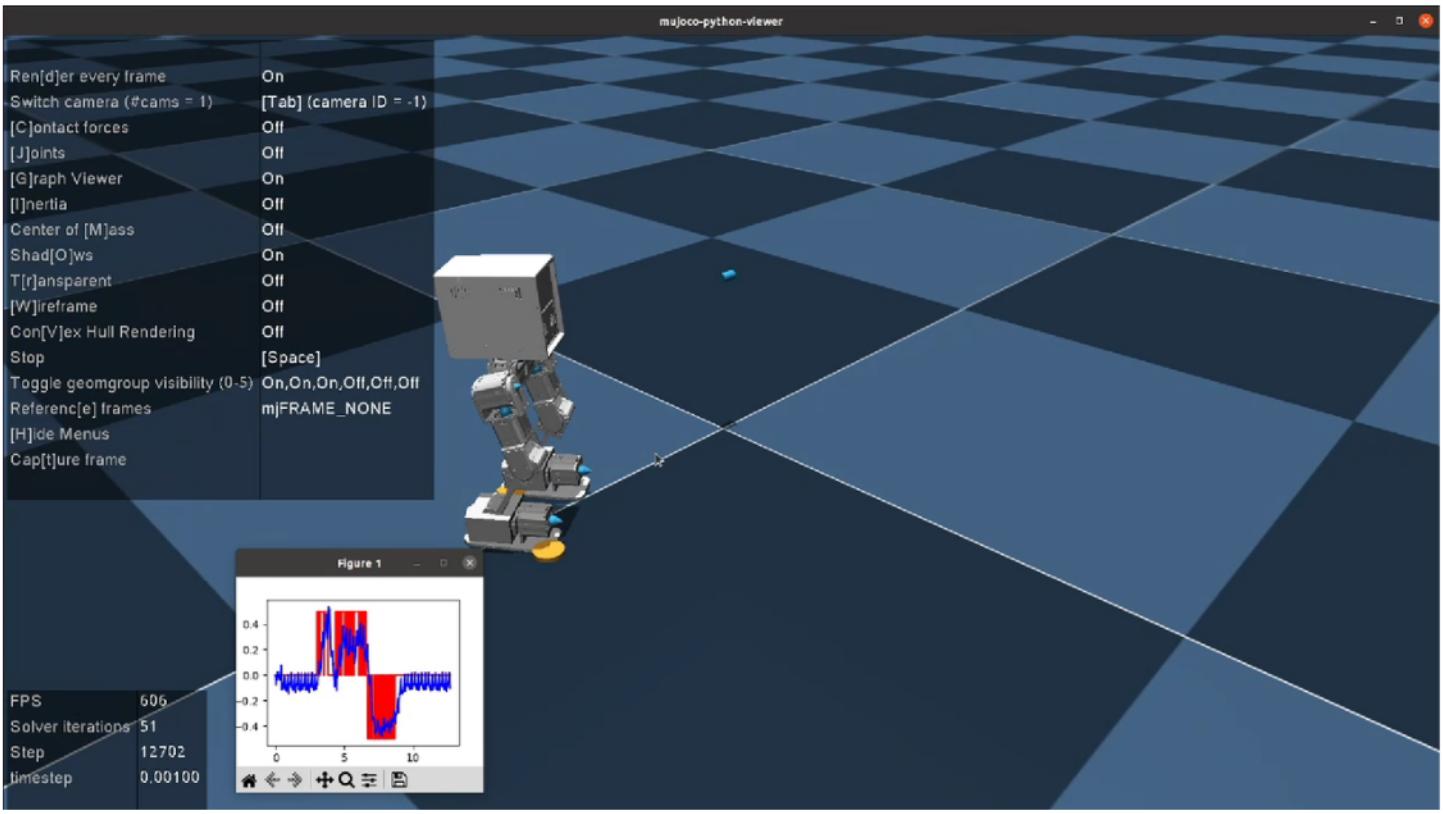


16. Run the pre-trained policy_example provided by High Torque:

代码块

```
python scripts/sim2sim.py --load_model
/path/to/logs/Pai_ppo/exported/policies/policy_example.pt
```

(Note: Replace `/path/to` with the path of the parent folder of `logs` .)



This concludes the basic reinforcement learning baseline tutorial. You can debug your own robot in the simulation environment. For more technical support, please call: +86-18144897433 or add WeChat: GQJD2022.