

Install & Download

1. Dependencies

System Support:

We currently best support Linux based systems. There is limited support for windows and no support for MacOS at the moment. We are working on trying to support more features on other systems but this may take some time. Most constraints stem from what the [SAPIEN](#) package is capable of supporting.

System / GPU	CPU Sim	GPU Sim	Rendering
Linux / NVIDIA GPU	✓	✓	✓
Windows / NVIDIA GPU	✓	✗	✓
Windows / AMD GPU	✓	✗	✓
WSL / Anything	✓	✗	✗
MacOS / Anything	✓	✗	✓

Occasionally, data collection may get stuck when using A/H series GPUs. This issue may be related to [RoboTwin issue #83](#) and [SAPIEN issue #219](#).

Python versions:

- Python 3.10

CUDA version:

- 12.1 (Recommended)

Hardware:

- Rendering: NVIDIA or AMD GPU
- Ray tracing: NVIDIA RTX GPU or AMD equivalent
- Ray-tracing Denoising: NVIDIA GPU
- GPU Simulation: NVIDIA GPU

Software:

- Ray tracing: NVIDIA Driver >= 470
- Denoising (OIDN): NVIDIA Driver >= 520

1.1 Additional Requirements for Docker

When running in a Docker container, ensure that the following environment variable is set when starting the container:

```
1 | -e NVIDIA_DRIVER_CAPABILITIES=compute,utility,graphics
```

Important : The graphics capability is essential. Omitting it may result in segmentation faults due to missing Vulkan support.

For more information, see [HERE](#).

2. Install Vulkan (if not installed)

```
1 | sudo apt install libvulkan1 mesa-vulkan-drivers vulkan-tools
```

Check by running `vulkaninfo`

3. Basic Env

First, prepare a conda environment.

```
1 | conda create -n RoboTwin python=3.10 -y  
2 | conda activate RoboTwin
```

RoboTwin 2.0 Code Repo: <https://github.com/RoboTwin-Platform/RoboTwin>

```
1 | git clone https://github.com/RoboTwin-Platform/RoboTwin.git
```

Then, run `script/_install.sh` to install basic envs and CuRobo:

```
1 | bash script/_install.sh
```

If you meet curobo config path issue, try to run `python script/update_embodiment_config_path.py`

If you encounter any problems, please refer to the [manual installation](#) section. If you are not using 3D data, a failed installation of pytorch3d will not affect the functionality of the project.

If you haven't installed ffmpeg, please turn to <https://ffmpeg.org/>. Check it by running `ffmpeg -version`.

4. Download Assets (RoboTwin-OD, Texture Library and Embodiments)

To download the assets, run the following command. If you encounter any rate-limit issues, please log in to your Hugging Face account by running `huggingface-cli login`:

```
1 | bash script/_download_assets.sh
```

The structure of the `assets` folder should be like this:

```
1 | assets
2 |   └── background_texture
3 |   └── embodiments
4 |     └── embodiment_1
5 |       └── config.yml
6 |       ...
7 |     ...
8 |   └── objects
9 |     ...
```

5. Manual Installation (Only when step 3 failed)

1. Install requirements

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

2. Install pytorch3d

```
1 | pip install "git+https://github.com/facebookresearch/pytorch3d.git@stable"
```

3. Install CuRobo

```
1 cd envs
2 git clone https://github.com/NVlabs/curobo.git
3 cd curobo
4 pip install -e . --no-build-isolation
5 cd ../../
```

4. Adjust code in `mplib` (**Important**)

5. You can use `pip show mplib` to find where the `mplib` installed.
6. Remove `or collide`

```
1 # mplib.planner (mplib/planner.py) line 807
2 # remove `or collide`
3
4 if np.linalg.norm(delta_twist) < 1e-4 or collide or not within_joint_limit:
5         return {"status": "screw plan failed"}
6 =>
7 if np.linalg.norm(delta_twist) < 1e-4 or not within_joint_limit:
8         return {"status": "screw plan failed"}
```

⌚ August 5, 2025 🚩 TianxingChen, byml-c