DISCOVERSE data、training、inference SOP

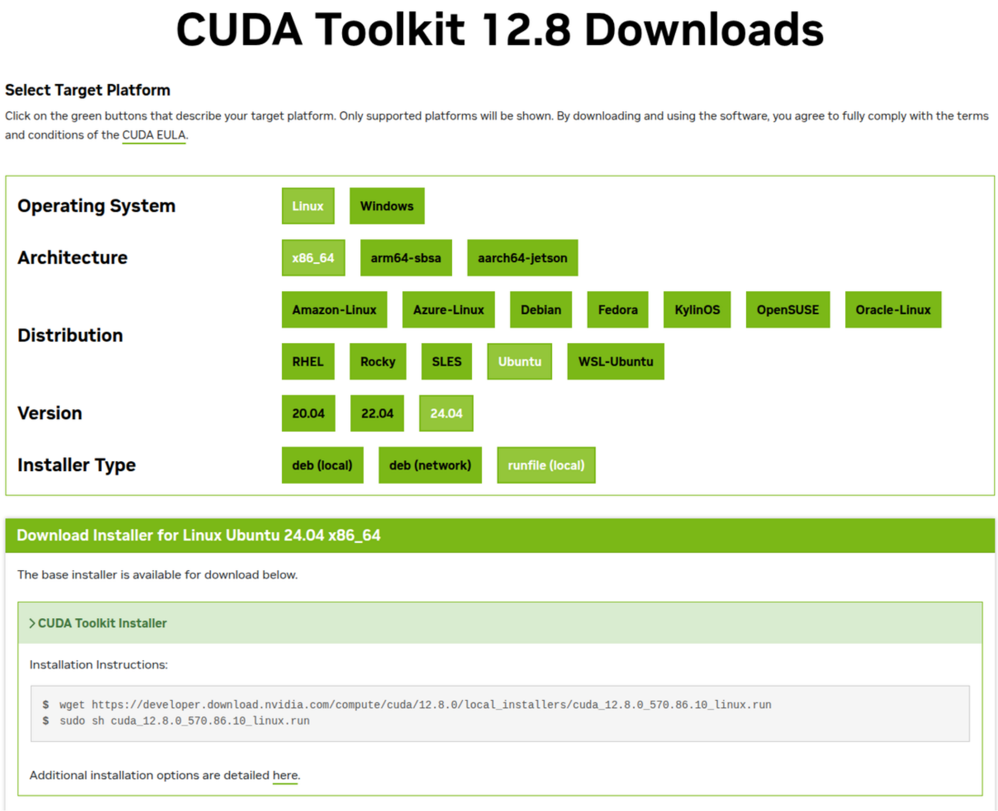
按照DISCOVERSE README完成环境搭建后，还需要安装CUDA12.8和PyTorch2.7**(建议使用终端代理)**

显卡配置：以使用GEFORCE RTX5060 8GB为例：

1. CUDA 12.8

CUDA 安装

地址:CUDA Toolkit 12.8 Downloads



CUDA Toolkit 12.8 Downloads

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the CUDA EULA.

Operating System: Linux | Windows

Architecture: x86_64 | arm64-sbsa | aarch64-jetson

Distribution: Amazon-Linux | Azure-Linux | Debian | Fedora | KylinOS | OpenSUSE | Oracle-Linux | RHEL | Rocky | SLES | Ubuntu | WSL-Ubuntu

Version: 20.04 | 22.04 | 24.04

Installer Type: deb (local) | deb (network) | runfile (local)

Download Installer for Linux Ubuntu 24.04 x86_64

The base installer is available for download below.

> CUDA Toolkit Installer

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/12.8.0/local_installers/cuda_12.8.0_570.86.10_linux.run
$ sudo sh cuda_12.8.0_570.86.10_linux.run
```

Additional installation options are detailed here.

代码块

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64/cuda-ubuntu2404.pin
sudo mv cuda-ubuntu2404.pin /etc/apt/preferences.d/cuda-repository-pin-600
wget https://developer.download.nvidia.com/compute/cuda/12.8.0/local_installers/cuda-repo-ubuntu2404-12-8-local_12.8.0-570.86.10-1_amd64.deb
sudo dpkg -i cuda-repo-ubuntu2404-12-8-local_12.8.0-570.86.10-1_amd64.deb
sudo cp /var/cuda-repo-ubuntu2404-12-8-local/cuda-*-keyring.gpg /usr/share/keyrings/
sudo apt-get update
sudo apt-get -y install cuda-toolkit-12-8
```

环境变量配置

代码块

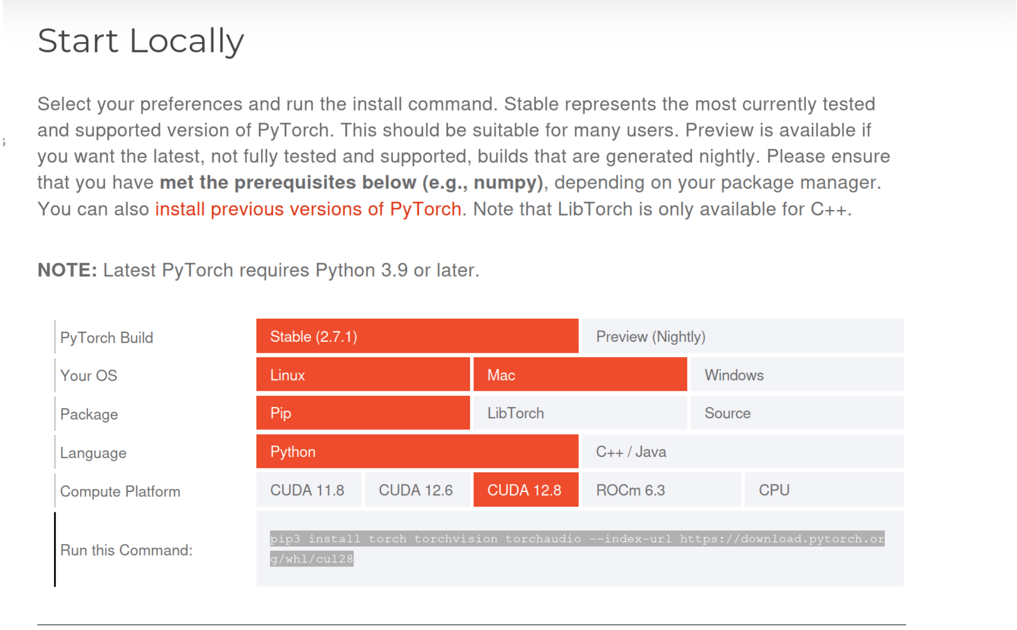
```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.8/lib64
export PATH=$PATH:/usr/local/cuda-12.8/bin
export CUDA_HOME=/usr/local/cuda-12.8
```

2. PyTorch 2.7

安装

地址：https://pytorch.org/get-started/locally/

CUDA12.8对应的PyTorch版本安装如下：



Start Locally

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. You can also install previous versions of PyTorch. Note that LibTorch is only available for C++.

NOTE: Latest PyTorch requires Python 3.9 or later.

PyTorch Build	[✓ Stable (2.7.1)]	Preview (Nightly)			
Your OS	[✓ Linux]	[✓ Mac]	Windows		
Package	[✓ Pip]	LibTorch	Source		
Language	[✓ Python]	C++ / Java			
Compute Platform	CUDA 11.8	CUDA 12.6	[✓ CUDA 12.8]	ROCm 6.3	CPU
Run this Command:	pip3 install torch torchvision torchaudio --index-url https://download.pytorch.or g/whl/cu128				

代码块

```
pip3 install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu128
```

3. 数据采集

✏️ 无括号特殊说明，命令均在 `DISCOVERSE` 根目录下执行

我们准备了若干个单臂、双臂的操作任务，分别位于 `discoverse/examples/tasks_airbot_play` 和 `discoverse/examples/tasks_mmk2` 中，要自动收集数据，请运行

代码块

```
e.g. python3 discoverse/examples/tasks_airbot_play/pick_jujube.py --data_set_size 5
# 表示使用airbot_play机械臂，任务为放置猕猴桃，总共生成5条任务轨迹。
```

仿真数据转换（airbot-data-collection）：

将仿真采集的原始数据格式转换为mcap格式，命令如下(注意路径):

```
python3 scripts/data_convert/discoverse2mcap.py --root /home/ghz/Work/airbot/DISCOVERSE/data --task-name pick_jujube
```

数据混合（按需）：

代码块

```
python3 policies/act/data_process/concatenate_datasets.py --datasets data/mcap/pick_jujube data/mcap/pick_jujube_copy --output-path data/mcap/concatenate
```

4. 模型训练

依赖安装

代码块

```
pip install -r policies/act/requirements/train_eval.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
```

数据集位置

仿真采集的数据默认位于DISCOVERSE仓库根目录的data文件夹中，而训练时默认从policies/act/data/mcap中寻找数据。因此，建议使用软连接的方式将前者链接到后者，命令如下（注意修改命令中的路径，并且需要绝对路径）：

代码块

```
ln -sf /absolute/path/to/discoverse/data
```

```
    /absolute/path/to/discoverse/policies/act/data
```

训练配置

参考的训练配置文件位于 `policies/act/configurations/airbots/airbot_discoverse_example.py` 中，其中主要参数解释如下：

- `camera_names`：训练数据中相机的序号
- `state_dim`：训练数据中观测向量的维度
- `action_dim`：训练数据中动作向量的维度
- `batch_size_train`：训练时的batch_size大小
- `batch_size_validate`：验证时的batch_size大小
- `chunk_size`：单次预测的动作数量
- `num_epochs`：训练的总步数
- `learning_rate`：学习率

训练特定任务时，需要复制一份配置文件并重命名为任务名，后续将通过任务名索引相关配置文件。

训练命令

代码块

```
python3 policies/train.py act -tn pick_jujube --batch_size 16 --learning_rate 2e-5 --num_epochs 500 -gth 50
```

5. 模型推理：

依赖安装

代码块

```
pip install -r policies/act/requirements/train_eval.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
```

推理配置

推理配置文件可基于训练配置文件修改，其中主要参数解释如下：

- `max_timesteps`：动作执行总步数，动作达到总步数后自动结束本次推理

推理命令

代码块

```
python3 policies/infer.py act -tn <task_name> -mts 100 -ts 20241125-110709 -rn discoverse/examples/<tasks_folder>/<task_script>
```

其中：

- `-tn` 任务名，程序会根据任务名分别在 `task_configs` 和 `data` 目录下寻找同名的配置文件和数据集
- `-mts` 动作执行总步数，该命令行参数会覆盖配置文件中的 `max_timesteps`
- `-ts` 时间戳，对应训练得到的模型文件所在的以时间戳命名的文件夹，程序会根据任务名和时间戳在policies/act/my_ckpt目录下寻找对应的模型文件
- `-rn` 数据采集时使用的脚本文件路径，例如 `discoverse/examples/tasks_airbot_play/pick_jujube.py`，程序会加载其中的 `SimNode` 类和 `AirbotPlayCfg` 的实例 `cfg` 来创建仿真环境