

13. ACT 模型

13.1 介绍

确保计算机配备 NVIDIA 显卡（显存小于 16G 可能无法训练大多数模型），并且 `nvidia-smi` 命令可用（驱动已安装）。

推荐使用 **anaconda** 管理 Python 环境。可运行以下命令来下载安装（如果下载速度过慢，可以[点击这里](#)手动下载）：

```
wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Miniconda3-py38_4.9.2-Linux-x86_64.sh
chmod u+x Miniconda3-py38_4.9.2-Linux-x86_64.sh && ./Miniconda3-py38_4.9.2-Linux-x86_64.sh
```

重新启动终端，即可使用 conda：

```
conda config --set auto_activate_base false && conda deactivate
```

13.2 安装

推荐使用 conda Python 环境。如果还未安装，可以使用以下命令创建并激活：

```
conda create -n act python=3.10 && conda activate act
```

克隆仓库，或联系售后服务以获取安装包：

```
git clone --depth 1 https://github.com/OpenGHZ/Imitate-All.git && cd Imitate-All
```

运行以下命令安装所需的包：

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

此外，针对模型训练，**请确保**已搭建好软件、硬件的机器人控制环境，例如 AIRBOT Play、TOK2、MMK2 等。

评估还需要使用 `airbot-data-collection` 包，该包可通过我们的售后服务获得。

13.3 预训练权重

首次运行训练或评估命令时，会自动下载预训练权重文件。

若您的终端没有代理，下载可能会失败。可以使用以下链接手动下载该文件（复制并粘贴到浏览器中）：<https://download.pytorch.org/models/resnet18-f37072fd.pth>。

或者，可以联系售后服务获取该文件，然后通过执行以下命令将其移动到指定目录：

```
mkdir -p ~/.cache/torch/hub/checkpoints/ && mv resnet18-f37072fd.pth  
~/.cache/torch/hub/checkpoints/
```

13.4 参数配置

训练或推理前准备，参数配置是必要的。请在 `./configurations/task_configs` 目录下创建一个与任务同名的 Python 文件（不建议直接修改或重命名 `example_task.py` 文件）来配置任务。

该配置主要涉及修改各种路径（使用 `replace_task_name` 函数来使用默认路径或手动指定路径）、摄像头名称（`camera_names`）、`state_dim`、`action_dim` 等参数。下面是来自 `example_task.py` 的示例，展示了如何基于 `template.py` 中的默认配置进行修改，而无需重写全部内容（如需更灵活的配置，请参考 `./configurations/task_configs/template.py`）：

```

chunk_size = 25
camera_names = ["env_camera"]
# camera_names = ["env_camera", "follow_camera"]
TASK_CONFIG_DEFAULT["common"]["camera_names"] = camera_names
TASK_CONFIG_DEFAULT["common"]["state_dim"] = 7
TASK_CONFIG_DEFAULT["common"]["action_dim"] = 7
TASK_CONFIG_DEFAULT["common"]["policy_config"]["temporal_agg"] = True
TASK_CONFIG_DEFAULT["common"]["policy_config"]["chunk_size"] = chunk_size
TASK_CONFIG_DEFAULT["common"]["policy_config"]["num_queries"] = chunk_size
TASK_CONFIG_DEFAULT["common"]["policy_config"]["kl_weight"] = 10
TASK_CONFIG_DEFAULT["common"]["policy_config"]["policy_maker"] = policy_maker

TASK_CONFIG_DEFAULT["train"]["data_type"] = "mcaps"
TASK_CONFIG_DEFAULT["train"]["load_data"]["num_episodes"] = "ALL"
TASK_CONFIG_DEFAULT["train"]["load_data"]["batch_size_train"] = 4
TASK_CONFIG_DEFAULT["train"]["load_data"]["batch_size_validate"] = 4
TASK_CONFIG_DEFAULT["train"]["load_data"]["observation_slice"] = None
TASK_CONFIG_DEFAULT["train"]["load_data"]["action_slice"] = None
TASK_CONFIG_DEFAULT["train"]["load_data"]["mcaps_state_topics"] = [
    "/follow/arm/joint_state/position",
    "/follow/eef/joint_state/position",
]
TASK_CONFIG_DEFAULT["train"]["load_data"]["mcaps_action_topics"] = [
    "/lead/arm/joint_state/position",
    "/lead/eef/joint_state/position",
]
TASK_CONFIG_DEFAULT["train"]["load_data"]["mcaps_camera_topics"] = [
    f"/{cam_name}/color/image_raw" for cam_name in camera_names
]

TASK_CONFIG_DEFAULT["train"]["num_epochs"] = 500
TASK_CONFIG_DEFAULT["train"]["learning_rate"] = 2e-5
TASK_CONFIG_DEFAULT["train"]["pretrain_ckpt_path"] = ""
TASK_CONFIG_DEFAULT["train"]["pretrain_epoch_base"] = "AUTO"

TASK_CONFIG_DEFAULT["eval"]["max_timesteps"] = 300
TASK_CONFIG_DEFAULT["eval"]["start_joint"] = "AUTO"
TASK_CONFIG_DEFAULT["eval"]["ensemble"] = None
TASK_CONFIG_DEFAULT["eval"]["environments"]["environment_maker"] = environment_maker
TASK_CONFIG_DEFAULT["eval"]["ckpt_names"] = ["policy_best.ckpt"]

# final config
TASK_CONFIG = TASK_CONFIG_DEFAULT

```

以上文件可作为单臂任务训练和推理的参考示例。对于不同的 AIRBOT 机器人，请在 `configurations/task_configs/airbots` 目录下选择对应的默认配置文件。例如，对于单个 AIRBOT Play 任务，请参考 `airbot_play_example.py`，对于双臂 AIRBOT Play 任务，请参考 `airbot_ptk_example.py`。

`camera_names` 参数可以是常用的摄像头 ID 编号，或摄像头所连接的 USB 端口编号。关于如何获取摄像头 USB 端口编号，请参考数据采集文档。

13.5 数据预处理

当使用默认路径进行训练时，请将 `.mcap` 数据文件放在 `./data/mcap/<task_name>` 文件夹中。

可以使用以下命令创建该目录：

```
mkdir -p data/mcap
```

然后，可以手动复制数据，或使用类似下面的命令（记得修改命令中的路径）：

```
cp -r /path/to/your/task/mcap_folder data/mcap/<task_name>
```

13.6 模型训练

请先完成 [安装](#) 和 [参数配置](#)（训练时至少需要 2 条数据，否则由于无法划分训练集和验证集将会报错）。

进入仓库文件夹并激活 Conda 环境：

```
conda activate act
```

然后运行训练命令：

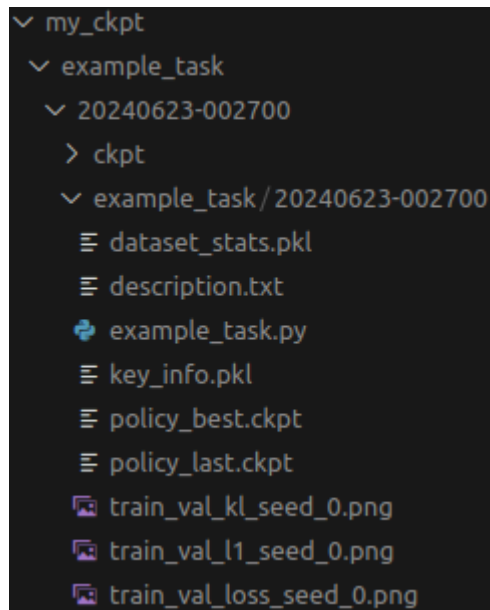
```
python3 policy_train.py -tn example_task
```

上述命令仅使用 `-tn` 参数时，会使用 `configurations/task_configs` 文件夹中对应任务名称的 `.py` 配置文件中的配置。

训练完成后，默认情况下，你可以在 `./my_ckpt/<task_name>/<time_stamp>` 目录中找到两个文件夹：

`ckpt` 文件夹包含所有权重文件（称为 **process folder**），而与 `<task_name>` 同名的文件夹（称为 **core folder**）包含以下文件：

- 最终权重和最优权重：分别为 `policy_best.ckpt` 和 `policy_last.ckpt`
- 统计数据：`dataset_stats.pkl`
- 关键信息（包括初始关节角度、训练参数配置等）：`key_info.pkl`
- 训练损失曲线：`train_val_kl_seed_0.png`、`train_val_l1_seed_0.png` 和 `train_val_loss_seed_0.png`
- 训练结果的简单描述：`description.txt`，例如 `Best ckpt: val loss 0.174929 @ epoch9499 with seed 0`



为了方便后续使用，建议将 **core folder** 存储在指定磁盘的 **Imitate-All/my_ckpt** 文件夹中。

13.7 模型评估

请确保已安装用于控制机器人（仿真或实物）的必要依赖。以下示例展示了如何使用 AIRBOT Play 机械臂来评估模型。

如果使用 realsense，请先安装：`pip install pyrealsense2 || pip install pyrealsense2-beta`。

13.7.1 环境配置

- 连接机械臂。
- 按照数据采集时的顺序或 USB 端口连接摄像头，如果自数据采集后未断开，则可跳过此步骤。
- 启动机械臂服务器：

```
airbot_server -i can_left -p 50051  
airbot_server -i can_right -p 50053
```

对于单臂任务，只需运行其中一条命令，并将 `-i` 参数修改为实际值，例如 `can_follow`。

13.7.2 检查任务配置

模型推理需要使用与训练相同的配置文件，并且需要调整推理部分的参数。

这主要包括修改 `environment_maker` 函数和 `TASK_CONFIG_DEFAULT["eval"]` 字典。

13.7.3 推理

进入仓库文件夹并激活 conda 环境：

```
conda activate act
```

评估命令及参数（请确保没有其他程序占用摄像头，如数据采集相关程序等）：

```
python3 policy_evaluate.py -tn example_task -cf
configurations/basic_configs/example/environment/airbot_ptk.yaml -ts
20250608-213703 -show -mts 250
```

- `-ts`：对应任务的时间戳（检查模型训练结果保存路径，例如 `my_ckpt/example_task/20240325-153007`）。
- `-show`：通过 OpenCV 窗口实时显示推理画面。
- `-cf`：配置文件路径。上述命令中的配置文件路径是双臂任务用的。单臂任务请参考同目录下的 `airbot_play_with_usbcams.yaml`。你应根据使用情况修改文件中的配置，例如摄像头索引、机械臂端口等。使用 Realsense 摄像头时，请参考 `airbot_play_with_realsenses.yaml`。
- `-mts`：评估每次 rollout 的最大时间步数，通常等于或略大于单次 episode 的长度。

机械臂启动并移动到任务定义的初始姿态后，你会在终端看到提示：

```
Press Enter to start evaluation or z and Enter to exit... （不使用 -show，请点击终端并按对应按键）
```

或

```
Press Enter to start evaluation or ESC to exit... （使用 -show，请点击任一图像窗口并按对应按键）。
```

每次评估完成后，你可以在当前目录的 `eval_results` 文件夹内对应时间戳的子文件夹中找到与评估相关的文件（包括过程视频）。

13.8 查看数据

模型训练完成后，关键信息和数据集统计将存储在 `key_info.pkl` 和 `dataset_stats.pkl` 文件中，可通过以下步骤查看。

进入仓库文件夹并激活 conda 环境：

```
conda activate act
```

然后，使用以下命令查看指定时间戳的信息：

```
python3 show_info.py -tn example_task -ts 20240420-214215 -in key_info
```

你将在终端看到该任务相关的关键信息，包括：

```
config_file_sys_path=/root/act/task_configs/test_task.py
init_info:
  init_joint: [-5.1308632e-02 -1.5125465e-01 2.0389843e-01 1.4547567e+00
-7.1011639e-01 -1.4394979e+00 2.8635000e-05]
  init_action: [-0.0516901 -0.15163612 0.2042799 1.4543753 -0.71049786 -1.439498
0.05392929]
all_config:
  config_file_sys_path: /home/ghz/Work/ALOHA/act/task_configs/example_task.py
  joint_num: 7
  robot_num: 1
  camera_names: ['0', '1', '2']
  task_type: static
  seed: 0
  policy_config:
    lr: 2e-05
    lr_backbone: 1e-05
    backbone: resnet18
    enc_layers: 4
    dec_layers: 7
    nheads: 8
    camera_names: ['0', '1', '2']
    policy_class: ACT
    kl_weight: 10
    chunk_size: 40
    hidden_dim: 512
    dim_feedforward: 3200
    temporal_agg: False
    num_queries: 40
    state_dim: 7
  ckpt_dir: /home/ghz/Work/ALOHA/act/my_ckpt/example_task/20240323-121312/ckpt
  stats_path: /home/ghz/Work/ALOHA/act/my_ckpt/example_task/20240323-121312/example_task/dataset_stats.pkl
  dataset_dir: /home/ghz/Work/ALOHA/act/data/hdf5/example_task
  batch_size: 16
  learning_rate: 2e-05
  num_epochs: 10
  pretrain_ckpt_path:
  pretrain_epoch_base: AUTO
  eval_every: 0
  validate_every: 2
  save_every: 2
  skip_mirrored_data: False
  sample_weights: None
  train_ratio: 0.9
  cotrain_dir:
  task_name: example_task
  show_train_info: False
  num_episodes: 2
  start_action: [-0.0516901 -0.15163612 0.2042799 1.4543753 -0.71049786 -1.439498
0.05392929]
  start_joint: [-5.1308632e-02 -1.5125465e-01 2.0389843e-01 1.4547567e+00
-7.1011639e-01 -1.4394979e+00 2.8635000e-05]
  policy_class: ACT
  state_dim: 7
```

这些信息包括训练过程中使用的 mcap 数据的绝对路径、训练参数配置、推理时第一条 episode 的初始关节值以及其他信息。

这些信息有助于保证实验的可复现性。如果摄像头刚性固定在机械臂上，复现机械臂动作相对容易。物体摆放则可以通过重新训练数据来确定。

对于数据集统计，只需在上述命令中设置 `-in stats` 即可。