

GR00T 的关节 index 核对:

一、在 GR00T 中主要存在如下几处关节 index

1. 数据集

1. 关节 index 表位于输出模型 checkpoints/experiment_cfg/metadata.json 或 数据集文件中 meta/modality.json

2. 输入:

1. 从仿真器读出来的 43 维数组(可从宇树官方 unitree_sim_isaac/robots/unitree.py 找到): left_leg, right_leg, waist, left_arm, right_arm, left_hand, right_hand

3. 模型推理端

1. 位于 Isaac-GR00T/gr00t/experiment/data_config.py

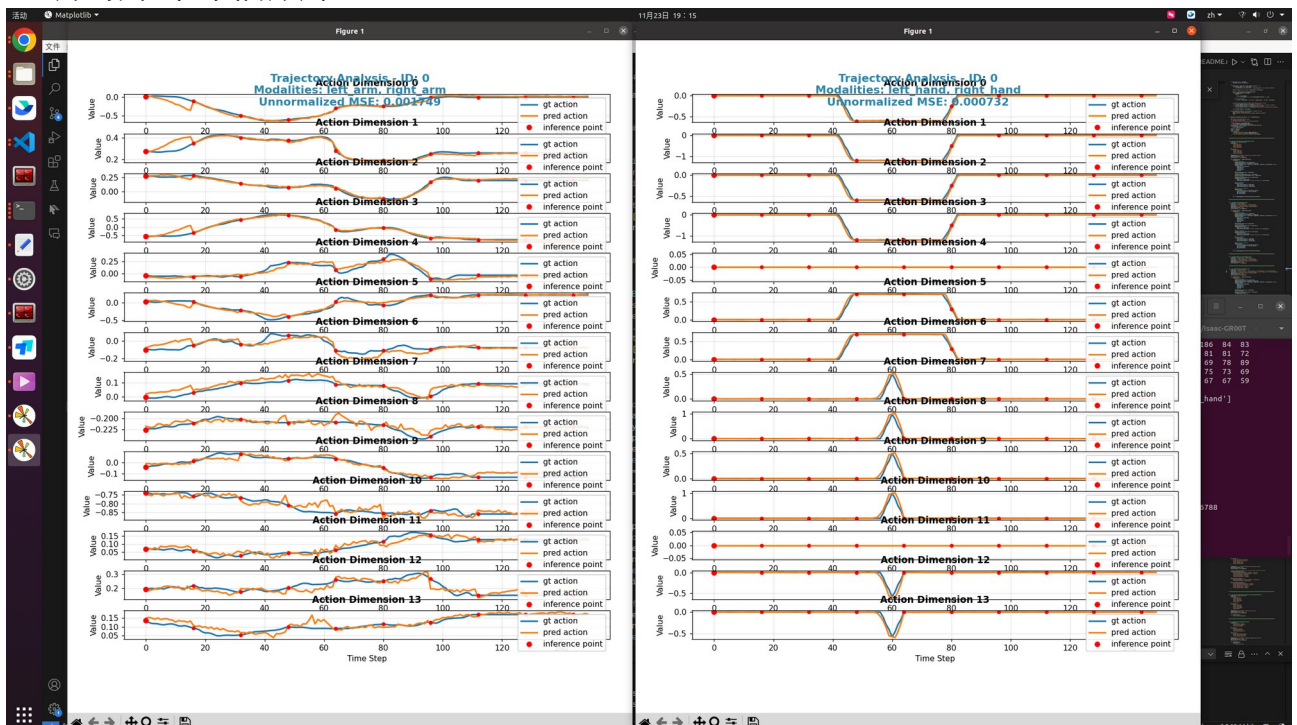
二、关节核对方法:

1. 预先核对

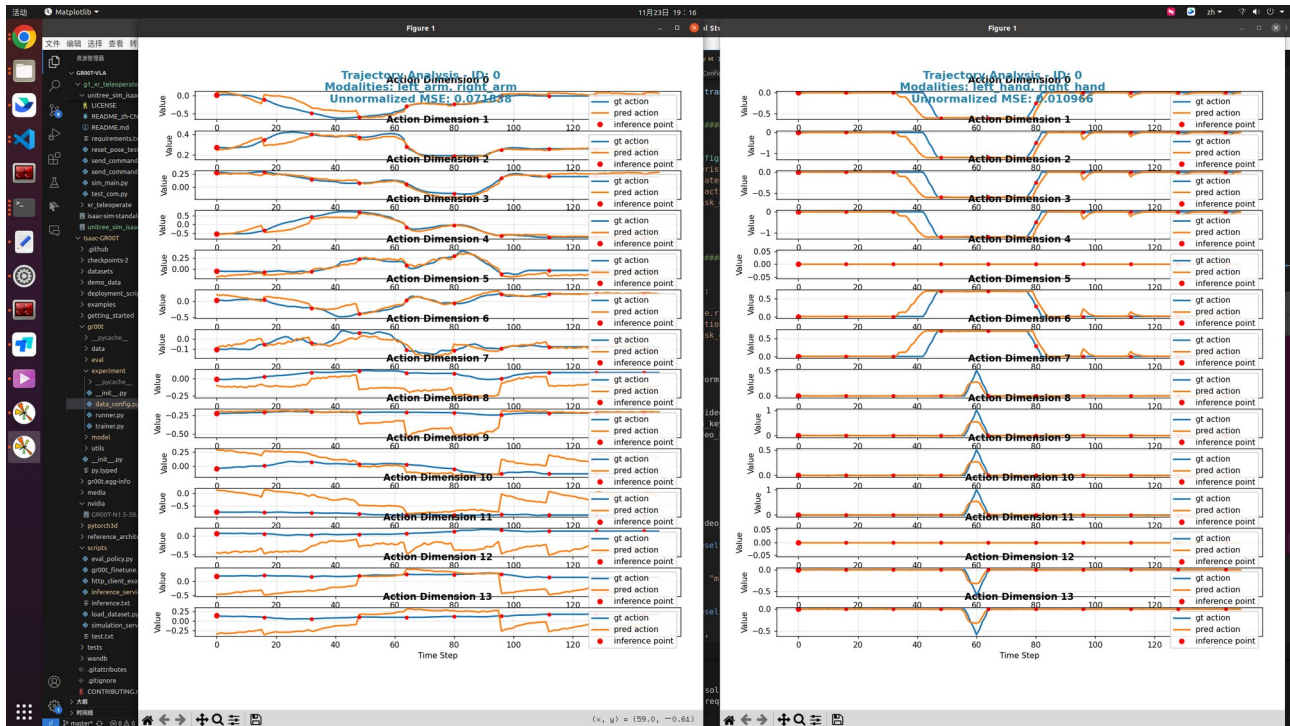
(1)首先可以确保的是从仿真器读出来的关节 index是没有问题的(因为宇树的 lab 中和官网都是一致的), 所以主要确保在输入和模型推理端的关节 index 的对齐

(2)其次需要保证开环评估 scripts/eval_policy.py 效果是比较好的(确保模型无问题, 即 Isaac-GR00T/gr00t/experiment/data_config.py):

正确的开环评估效果:



当模型输入 state 的 index 不对的效果：



2.实际修改

主要是按照宇树官方 G1 的关节顺序，直接手动赋值输入，条件赋值输出

(1).首先手动设置从仿真器读出的 43 维数组中上身部分关节 (unitree_sim_isaac/robot/action_provider/action_provider_gr00t.py 中 get_robot_state 函数)：

```
# joint_pos[15:22] = 1.0(left_arm)
# joint_pos[22:29] = 2.0(right_arm)
# joint_pos[29:36] = 3.0(left_hand)
# joint_pos[36:43] = 4.0(right_hand)
```

(2).其次修改模型端(Isaac-GR00T/gr00t/eval/http_server.py 中 predict_action 函数)

```
if 'state.left_arm' in obs:
if 'action.left_arm' in action:
action['action.left_arm'] = np.full_like(action['action.left_arm'], -1.0)
if 'state.right_arm' in obs:
if 'action.right_arm' in action:
```

```

action['action.right_arm'] = np.full_like(action['action.right_arm'], -2.0)
if 'state.left_hand' in obs:
    if 'action.left_hand' in action:
        action['action.left_hand'] = np.full_like(action['action.left_hand'], -3.0)
    if 'state.right_hand' in obs:
        if 'action.right_hand' in action:
            action['action.right_hand'] = np.full_like(action['action.right_hand'], -4.0)

```

(3).最终效果:

如果关节 index 对应上的话，理想情况是模型输入：

1. state.left_arm = 1
2. state.right_arm = 2
3. state.left_hand = 3
4. state.left_hand = 4

模型输出:

1. action.left_arm = -1
2. action.right_arm = -2
3. action.left_hand = -3
4. action.left_hand = -4

