

# ARX方舟无限aloha数据采集系统X7s使用手册

## 1 安全概要

⚠本产品为实验性产品，仅供科研实验场景使用，操作前请严格按照用户手册进行，如不按用户手册使用，若导致安全事故，后果自负。

⚠使用本产品具有一定的安全风险，建议用户首先观看本章节内容，再阅读后面的内容了解具体的使用过程。

⚠使用前请先阅读《X7s数据采集系统使用说明书.pdf》

## 运行环境

1. 工作温度为 $0^{\circ}\text{C}\sim 40^{\circ}\text{C}$ ，请勿在温度低于 $0^{\circ}\text{C}$ 、高于 $40^{\circ}\text{C}$ 环境中使用
2. 请勿在存在腐蚀性、易燃性气体的环境或者靠近可燃性物质的环境中使用
3. 请不要在加热器或者大型卷线电阻等发热体周围使用
4. 建议使用环境海拔高度不超过1000M
5. 建议使用环境昼夜温差不超过 $25^{\circ}\text{C}$

## 运行前检查

1. 确保与机器人共同工作的所有人员都接受过充分的培训，能够遵守操作规范，

安全操控机械臂

2. 确保机械臂正常供电
3. 确保机械臂周围无异物
4. 确保机械臂初始化完成
5. 确保机械臂安装牢固
6. 确保相机以及机械臂各关节电机的清洁

## 机械臂运行时操作

1. 运行时所有人员请远离机械臂工作空间
2. 机械臂停止空中时，断电前请手扶着机械臂，然后断电，手托机械臂归零位
3. 切勿在机械臂正常运转时断开电源

4. 切勿在机械臂正常运转时断开机械臂的can线
5. 模型推理时，请远离无关设备，防止机械臂掉落损坏设备
6. 机械臂模型推理时请远离机械臂工作空间
7. 如果机械臂推理发生错误导致机械臂运动紊乱，请立即中断推理程序
8. 如果推理程序无法中断导致机械臂运动紊乱，请立即关闭机械臂电源

## 注意事项

### 警告：电气危害

1. 如果不依照规范安全使用机械臂上的电子组件，可能导致设备损坏或人身伤害。
2. 请勿在机械臂上电时对其内部件执行任何操作；

### 小心：坠落危害

1. 机械臂突然断电，由于重力作用会导致机械臂掉落，从而导致机械臂损坏
2. 模型推理时，请远离无关设备，防止机械臂掉落损坏设备

## 操作注意事项

1. 首次启动系统和设备前，必须检查设备和系统是否完整，操作是否完全，检查机械臂和其他设备系统是否遭到损坏
2. 在使用机械臂前，需要对机器人及周边防护系统进行初步测试和检查
3. 在使用机械臂前，操作人员必须经过相应培训，必须确保输入的参数和操作是正确的
4. 机器人使用时，请无关人员远离机械臂的工作空间
5. 机械臂在发生意外时运行不正常的情况下，可以立即断开机械臂电源，但是断开机械臂电源可能会导致机械臂在掉落过程中损坏。
6. 请勿随意断开机械臂电源和机械臂 can线
7. 在机械臂作业时，切勿有人或其他设备出现在作业范围内
8. 切勿将手指伸到末端执行器以及线性模组连接处
9. 确保机械臂的工作空间没有障碍物
10. 切勿擅自对机械臂内部系统包括软件系统和硬件系统进行改动
11. 在机械臂运输或搬运过程中，做好防撞防水措施

12. 在机械臂运输或搬运过程中，做好保护好相应的传感器，确保传感器的安装位置以及姿态没有被调整
13. 当机械臂与其他机械协作时，应对整个协助系统做全面的安全评估
14. **切勿随意执行机械臂的模型推理过程**
15. 机械臂不使用情况下，请将机械臂归零位
16. 机械臂为易损件，请勿暴力使用

项目地址：

运行环境：ubuntu20.04, python3.8.10, ros1-noetic, torch-2.2.0+cu121

## 2 快速开始

### 2.1 创建虚拟环境

进入mobile\_aloha文件夹，运行 `venv.sh` 脚本

```
1 cd mobile_aloha
2 bash venv.sh
```

后续所有程序都需要在虚拟环境中执行

### 2.2 编译相机驱动

- 依赖项安装

```
1 sudo apt-get install -y libudev-dev pkg-config libgtk-3-dev
2 sudo apt-get install -y libusb-1.0-0-dev pkg-config
3 sudo apt-get install -y libglfw3-dev
4 sudo apt-get install -y libssl-dev
```

- realsense库安装

```
1 git clone https://github.com/IntelRealSense/librealsense.git
2
3 cd librealsense
4 mkdir build
5 cd build
```

```
6  cmake ../ -DBUILD_EXAMPLES=true
7  make
8  sudo make install
```

- 添加udev规则

```
1  sudo cp config/99-realsense-libusb.rules /etc/udev/rules.d/
2  sudo udevadm control --reload-rules && udevadm trigger
```

- 编译相机工作空间

```
1  cd realsense_camera
2  sudo apt install -y ros-noetic-ddynamic-reconfigure
3  catkin_make
```

在电脑上连接realsense d405的usb

```
1  source ./devel/setup.bash
2  rosrun realsense2_camera list_devices_node
```

将获取到的Serial number分别填入src/ros\_realsense2\_camera/launch中的  
**rs\_multiple\_devices.launch**文件里的serial\_no\_camera中

```
<launch>
  <arg name="serial_no_camera1"          default="230322275966"/>
  <arg name="serial_no_camera2"          default="230322272285"/>
  <arg name="serial_no_camera3"          default="230322273049"/>
```

## 2.4 编译机械臂

在X7s\_PLAY目录下，执行 `make.sh`，其相当于执行X7s/00-sh/ROS/中的 `01_make.sh` 和 `02_make.sh`

详细信息请参考《X7s数据采集系统使用说明书.pdf》

## 3 数据采集

### 3.1 准备工作

- 运行X7s\_PLAY/realSense\_camera中的 `realsense.sh` 脚本
- 运行X7s/00-sh/ROS/remote\_X7s.sh中的 `remote_X7s.sh` 脚本

查看topic消息

```
1 source X7s_PLAY/x7s/ROS/x7s_ws/devel/setup.sh
2
3 # 机械臂控制信息
4 rostopic echo /joint_control
5 rostopic echo /joint_control2
```

## 3.2 采集数据

切记所有人员请远离机械臂工作空间

切勿暴力操作机械臂

如机械臂发生紧急情况, 请立即断电

在mobile\_aloha中 `source venv/bin/activate` 后执行 `python collect_data.py` 即可操作机械臂采集数据

在默认参数下, 会产生数据集 `episode_0.hdf5` 保存到mobile\_aloha/datasets文件夹中, 该工程目录如下

```
mobile_aloha
├── collect_data.py
├── datasets # --datasets 数据集保存路径
│   ├── episode_0.hdf5 # 产生的数据集文件
│   ├── episode_idx.hdf5 # idx由--episode_idx参数决定
│   └── ...
├── readme.md
└── requirements.txt
```

`collect_data.py` 主要命令行参数:

- `--datasets`: 采集数据集存放路径
- `--episode_idx`: 数据集索引号, 当设置为-1时, 将循环采集直到程序退出

- --frame\_rate: 采集帧率
- --max\_timesteps: 采集时长
- --use\_depth\_image: 是否采集深度图（默认为false）
- --use\_base: 是否启用底盘（默认为false）
- --record: 底盘采集模式（Distance或Speed，默认为Distance）
- --is\_compress: 是否使用压缩图像（默认为false）

注:

1. 操作人员操作VR，机械臂就会随动
2. 切记所有人员请远离机械臂工作空间，切勿暴力操作机械臂
3. 推荐采集压缩图像
4. 若底盘采集模式选择为 `Distance`，需要外接VIO或者激光雷达提供TF信息，若选择 `Speed` 模式，则采集底盘轮速信息，此时不需要外接里程计

```
1 source ./venv/bin/activate
2 python collect_data.py --datasets datasets --episode_idx -1 --max_timesteps
  800 --is_compress
```

## 一键启动

在X7s\_PLAY目录下，执行 `collect.sh`

注意：机械臂can id和realsense相机serial需要提前配置

根据需求修改其中的 `datasets`

数据集采集要求

1. 目标物体始终在相机范围内
2. 机械臂运动轨迹有一定分布
3. 环境保持一致

### 3.2.1 采集教程

当程序初始化完成后，程序会语音播报初始化信息，当操控双臂同时闭合并张开夹爪后，程序将会语音播报 `go`，此时程序将会采集图像和机械臂的相关信息，直到达到设定采集时间或者使用VR手柄上的按键让双臂回归初始位置

## 3.3 可视化数据集

在mobile\_aloha中 `source venv/bin/activate` 后执行 `python visualize_episodes.py --datasets ./datasets --episode_idx 0` 即可可视化数据集

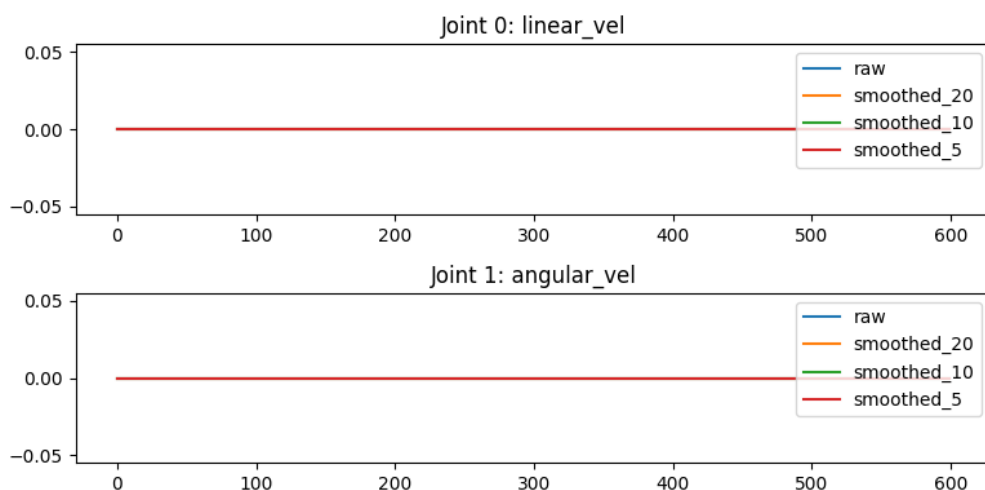
将3.2采集的数据集进行可视化运行上述代码，其中 `--datasets` 和 `--episode_idx` 需要和3.2采集数据时相同

运行代码时，终端会打印action，并显示彩色窗口图像

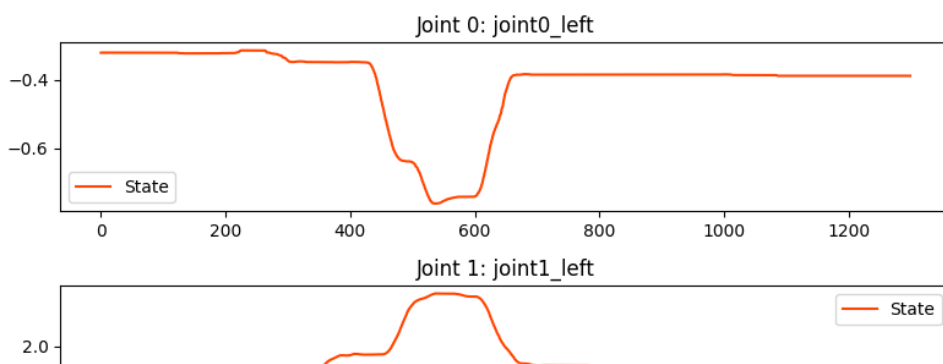
运行完成后，会在`${datasets}`下产生 `episode_${idx}_qpos.png`、`episode_${idx}_base_action.png` 与 `episode_${idx}_video.mp4` 文件，目录结构如下：

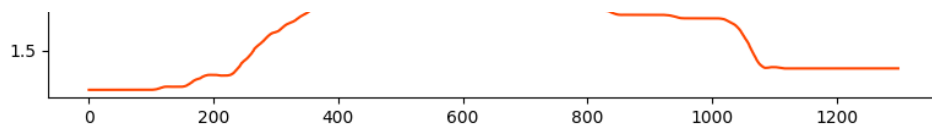
```
mobile_aloha
├── visualize_episodes.py
├── datasets
│   ├── episode_0.hdf5
│   ├── episode_0_base_action.png # base_action图
│   ├── episode_0_qpos.png # qpos图
│   └── episode_0_video.mp4 # 彩色图像视频流
```

- `episode_${idx}_base_action.png`

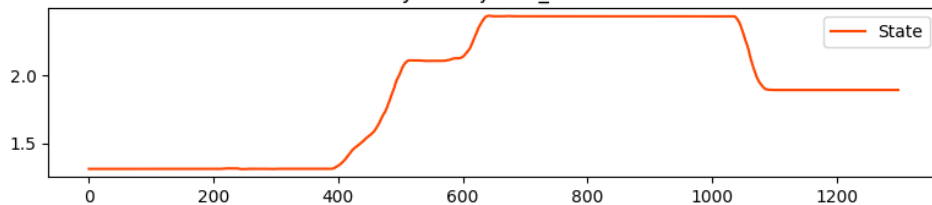


- `episode_${idx}_qpos.png`

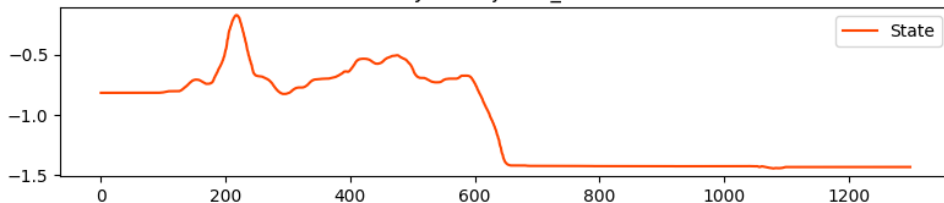




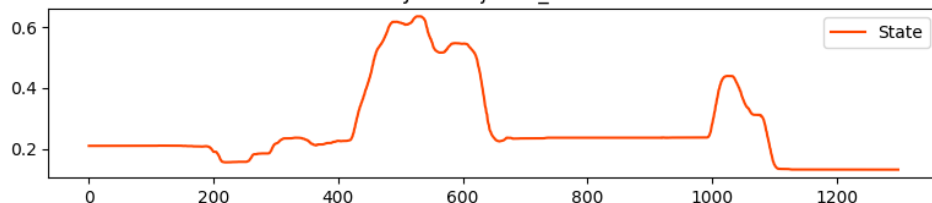
Joint 2: joint2\_left



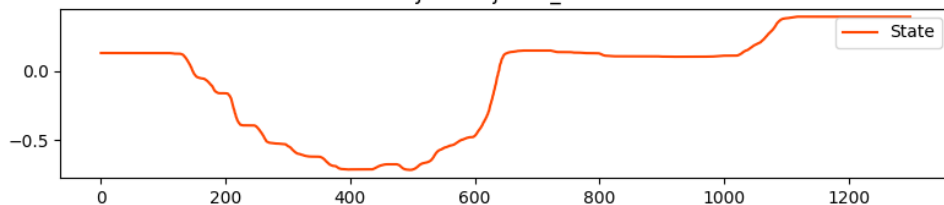
Joint 3: joint3\_left



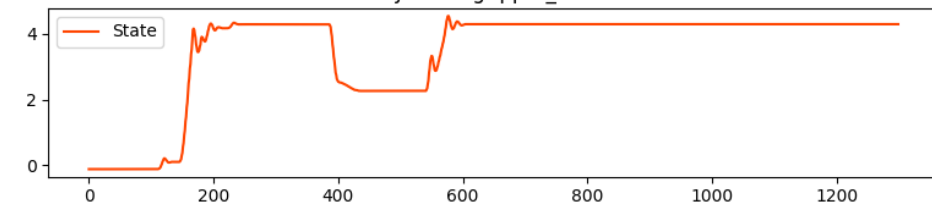
Joint 4: joint4\_left



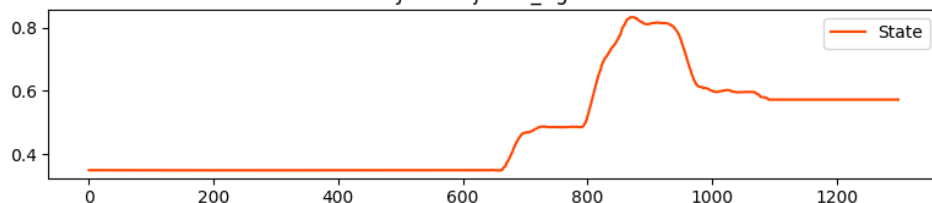
Joint 5: joint5\_left



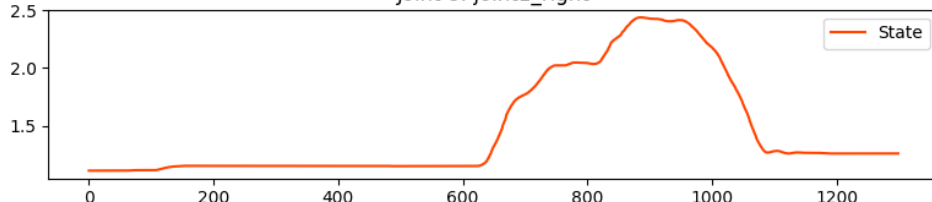
Joint 6: gripper\_left



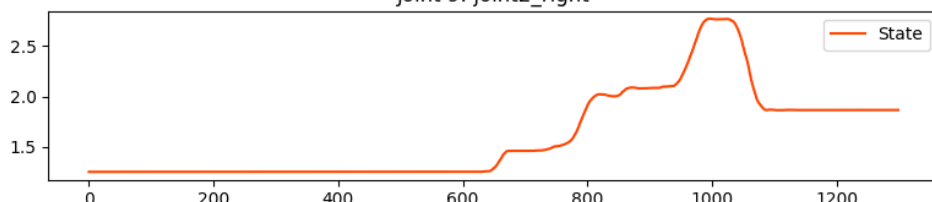
Joint 7: joint0\_right



Joint 8: joint1\_right

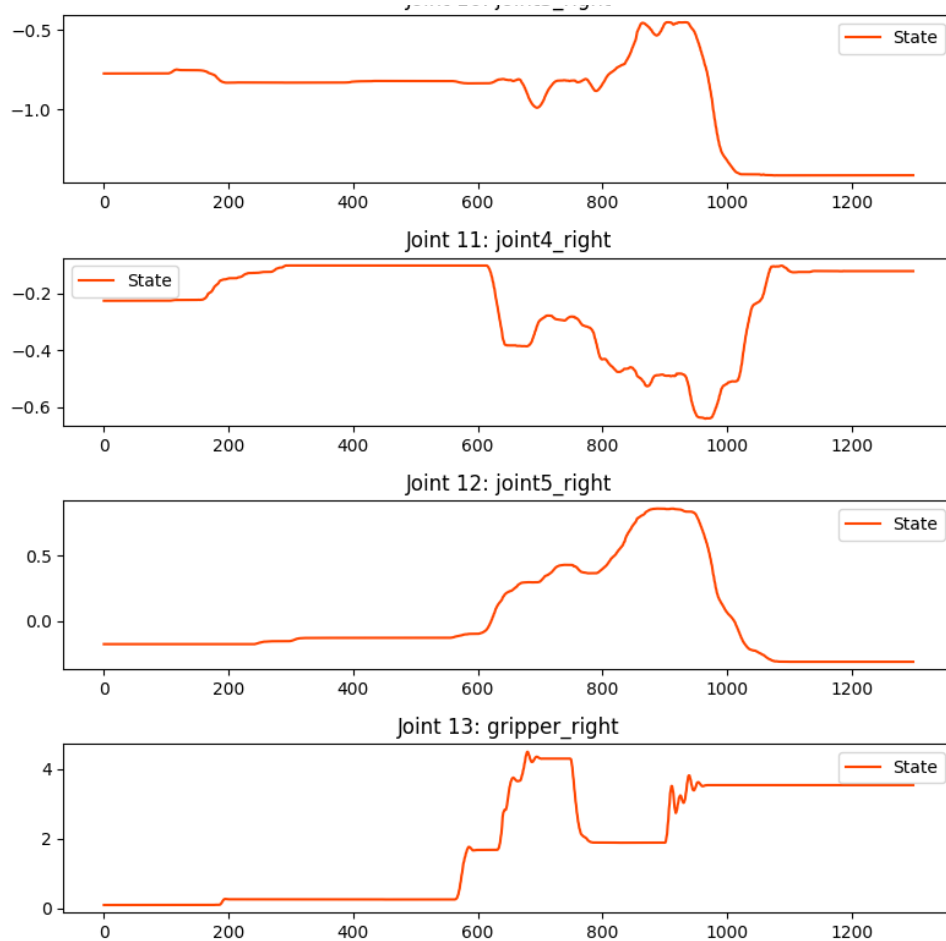


Joint 9: joint2\_right



Joint 10: joint3\_right





`visualize_episodes.py` 主要命令行参数:

- `--datasets`: 数据集存放路径
- `--episode_idx`: 数据集索引号

```
1 source ./venv/bin/activate
2 python visualize_episodes.py --datasets datasets --episode_idx 1
```

## 4 ACT 模型训练推理

### 4.1 模型训练

在`mobile_aloha`文件夹中 `source venv/bin/activate` 后执行 `python train.py` 即可训练模型

`train.py`主要命令行参数:

- `--datasets`: 采集数据集存放路径
- `--ckpt_dir`: 训练模型保存路径
- `--ckpt_name`: 最佳模型保存名称
- `--pretrain_ckpt`: 预训练模型

- `--num_episodes`: 数据集数量
- `--batch_size`: 批次大小
- `--epochs`: 迭代次数
- `--use_qvel`: 是否训练速度信息 (默认为false)
- `--use_effort`: 是否训练力矩信息 (默认为false)
- `--use_base`: 是否训练底盘 (默认为false)
- `--use_depth_image`: 是否训练深度图 (默认为false)

```
1 source ./venv/bin/activate
2 python train.py --datasets datasets --ckpt_dir weights --num_episodes 50 --
  batch_size 32 --epochs 3000
```

注意：若使用额外电脑训练，需要确保训练电脑python环境和部署电脑相同

## 一键启动

在X7s\_PLAY目录下，执行 `train.sh`

根据需求修改其中的 `datasets` , `num_episodes` , `batch_size` , `epochs`

## 4.2 模型推理

### 4.2.1 准备工作

- 运行`realsense_camera`中的 `realsense.sh` 脚本
- 在 `source X7s/body/ROS/devel/setup.bash` 环境后，执行 `roslaunch arx_lift_controller x7s.launch`
- 在 `source X7s/x7s/ROS/x7s_ws/devel/setup.bash` 环境后，分别执行 `roslaunch arx_x7_controller left_arm_inference.launch` 和 `roslaunch arx_x7_controller right_arm_inference.launch`

- 
- 切记所有人员请远离机械臂工作空间
  - 切勿暴力操作机械臂
  - 如机械臂发生紧急情况, 请立即终止推理代码,如果不能终止,请立即断开机械臂电源

械臂电源

---

## 4.2.2 模型推理

在**mobile\_aloha**文件夹中 `source venv/bin/activate` 后执行 `python inference.py` 即可进行推理

`inference.py` 主要命令行参数：

- `--max_publish_step`：最大执行步长
- `--ckpt_dir`：模型路径
- `--ckpt_name`：模型名称
- `--use_qvel`：是否使用速度信息推理（默认为false）
- `--use_effort`：是否使用力矩信息推理（默认为false）
- `--use_base`：是否推理底盘（默认为false）
- `--record`：记录底盘信息（Distance或Speed，默认为Distance）
- `--use_depth_image`：是否使用深度图推理（默认为false）

```
1 source ./venv/bin/activate
2 python inference.py --ckpt_dir weights --ckpt_name policy_best.ckpt
```

注：

1. 若推理底盘，record需要和数据集采集的方式相同，采集时使用Distance，推理时也需要使用Distance模式，Speed模式同理

## 一键启动

在**X7s\_PLAY**目录下，执行 `inference.sh`

注意：机械臂can id和realsense相机serial需要提前配置

## 5 异常处理

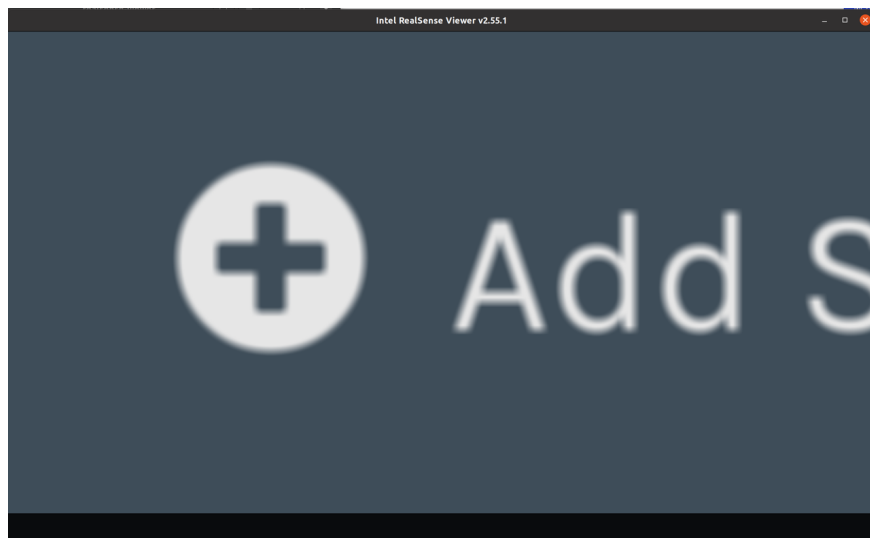
1. 在查找相机 `serial number` 时，出现如下报错

```
No device connected, please connect a RealSense device
[ERROR] [1736693438.047729003]: No device connected, please connect a RealSense device
list SN is empty
```

检查相机连接情况

此外，需要确保相机连接在电脑上的USB3.0口。一个USB3.0口只能同时读取两个realsense相机

2. 启动 `realsense-viewer` 时，界面出现如下情况



降低电脑屏幕刷新率

3. 训练时出现如下情况

```
torch.OutOfMemoryError: CUDA out of memory. Tried to allocate 48.00 MiB. GPU 0 has a total capacity of 7.73 GiB of which 108.12 MiB is free. Including non-PyTorch memory, this process has 5.87 GiB memory in use. Of the allocated memory 5.54 GiB is allocated by PyTorch, and 217.42 MiB is reserved by PyTorch but unallocated. If reserved but unallocated memory is large try setting PYTORCH_CUDA_ALLOC_CONF=expandable_segments:True to avoid fragmentation. See documentation for Memory Management (https://pytorch.org/docs/stable/notes/cuda.html#environment-variables)
```

降低 `batch_size` 大小

4. 推理时出现如下情况

```
numpy._core._exceptions._ArrayMemoryError: Unable to allocate 2.04 TiB for an array with shape (100000, 100030, 28) and data type float64
```

降低 `max_publish_step` 大小