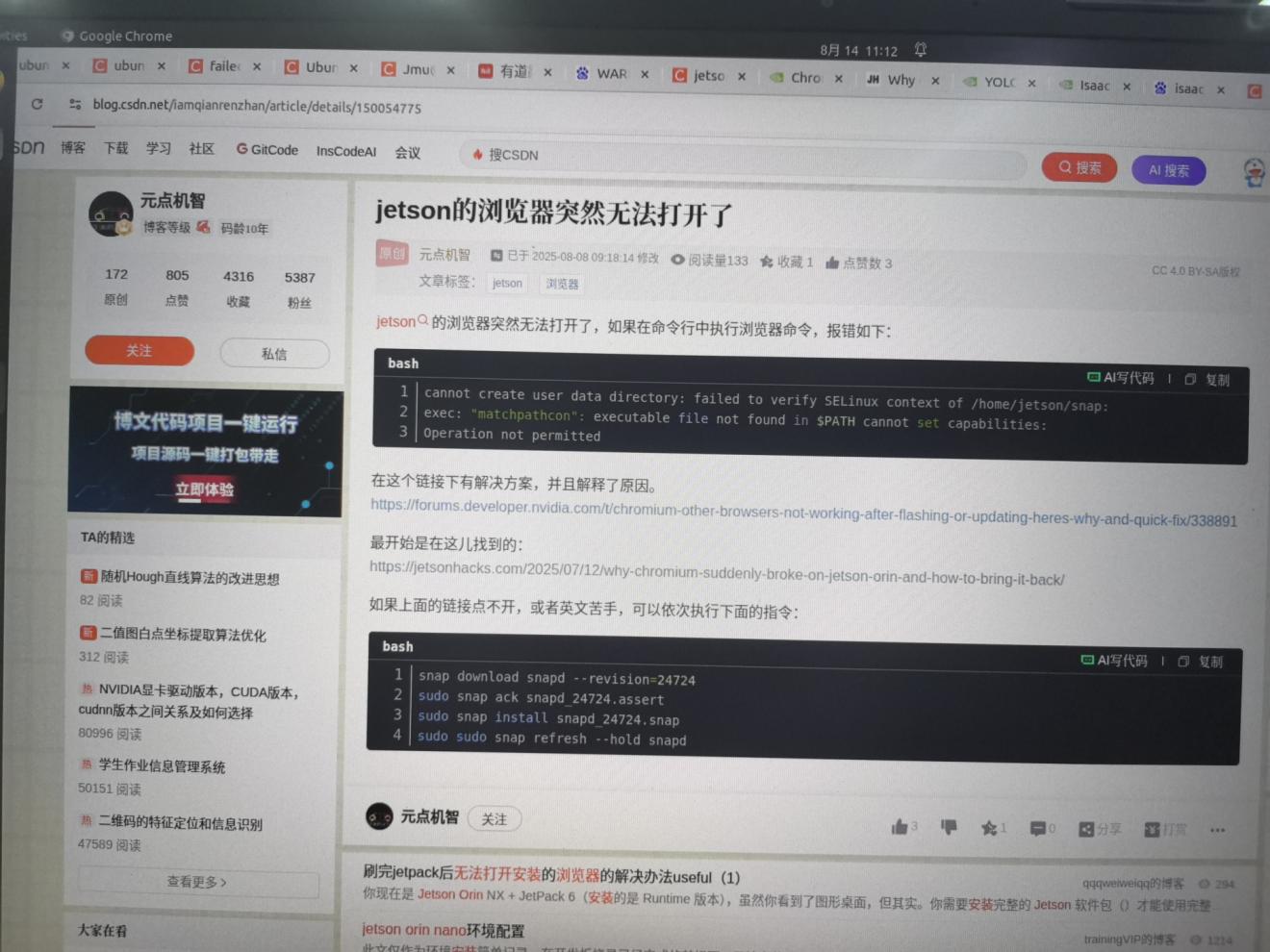
浏览器问题解决：



浏览器snap问题解决教程

任务一、RT-DetR、姿态估计、点云三维重建算法的部署

官方代码逻辑不完全适用于机器人特定场景，需要使用自制数据集进行训练，通过模型的表现找出代码逻辑的修改点，发现在原有代码上需要添加归一化逻辑以提升检测精度（原代码不含有归一化逻辑）。

threshold＝0.7即可检测-->但是检测不出来？需要更高的阈值

命令行加上更高的threshold参数，但是还是无法检测-->发现缺失了传递阈值的代码逻辑，于是添加此逻辑，最终实现成功检测。

机械臂执行可视化平台：使用代码支持的Rviz软件

机器人操作系统：采用ROS2

**完整部署流程：按照教程配置环境、训练并调试代码、连接传感器并通过命令行执行算法，机械臂运动的同时在Rviz中实现可视化。ROS2自带运动规划功能，摄像头识别出目标物体与障碍物体后自动规划运动轨迹并沿此轨迹到达目标物体处，执行抓取操作。**

RDK ×5系统烧录：

参考网址：<https://developer.d-robotics.cc/rdk_doc/Quick_start/>



烧录完成

ModelZoo快速上手，依次执行命令如下：

**pip install bpu\_infer\_lib\_x5 -i http://sdk.d-robotics.cc:8080/simple/ --trusted-host sdk.d-robotics.cc**

**pip install jupyterlab -i https://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com**

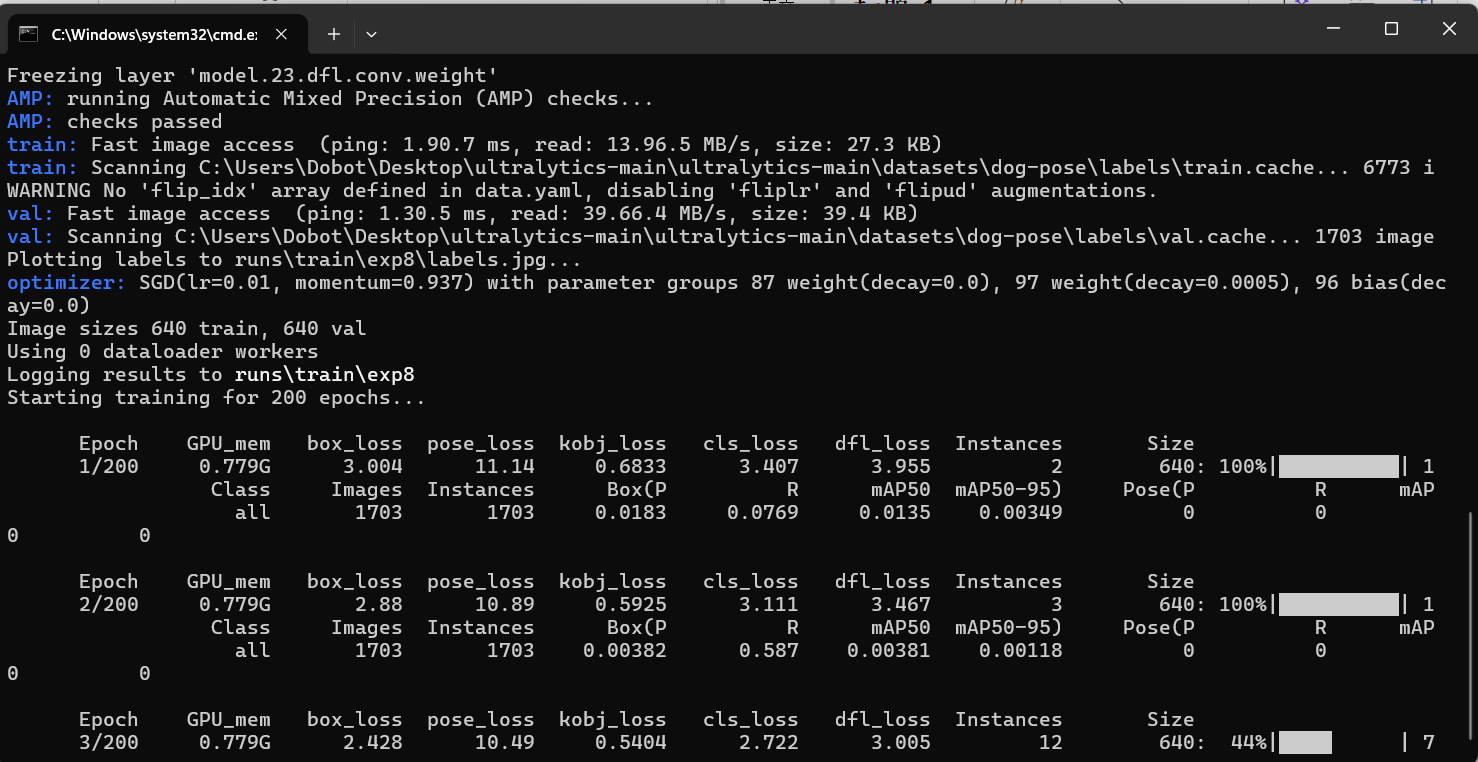
**git clone <https://github.com/D-Robotics/rdk_model_zoo>**

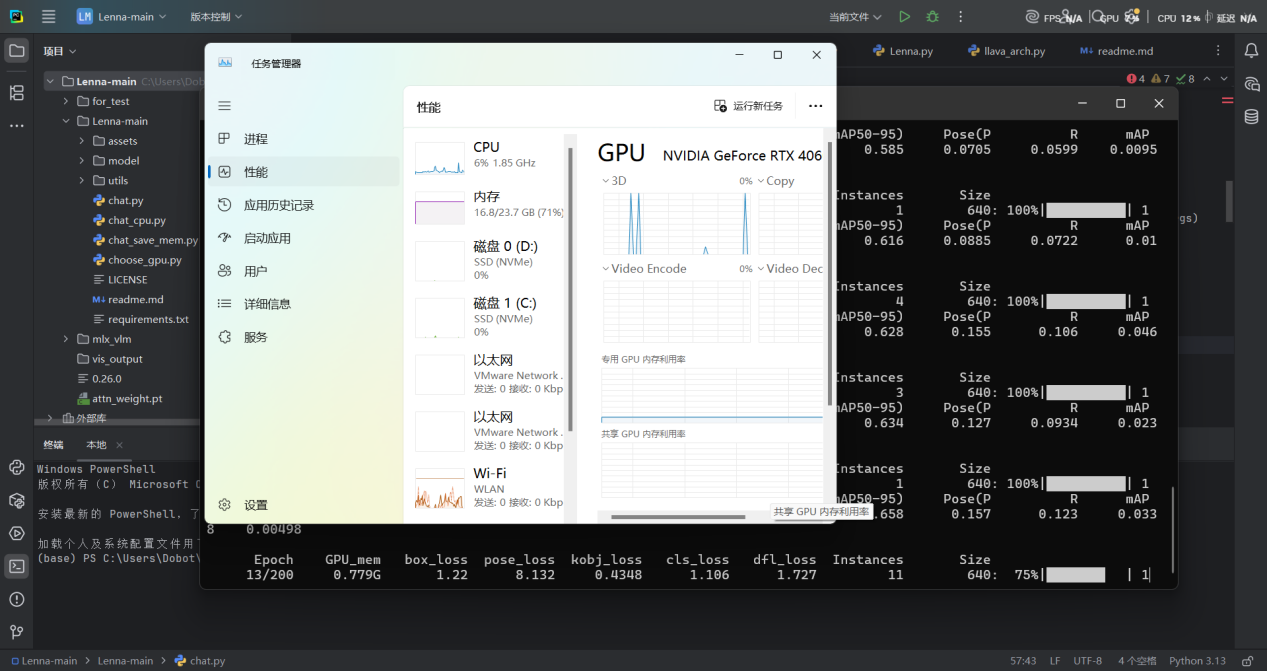
**cd rdk\_model\_zoo**

**sudo chmod -R 777 /rdk\_model\_zoo #放开权限，允许所有用户读写**

**python -m jupyterlab --ip=0.0.0.0 --port=8888**

**YOLOv11-pose在dog-pose数据集上进行训练：**



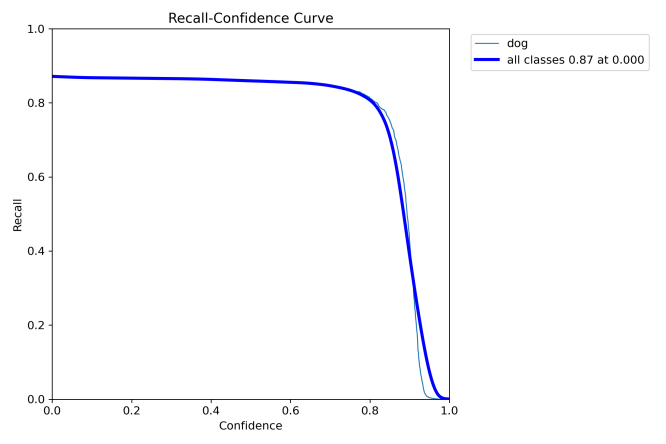
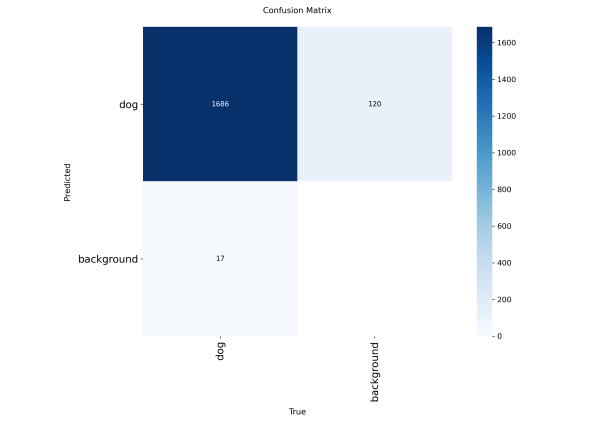




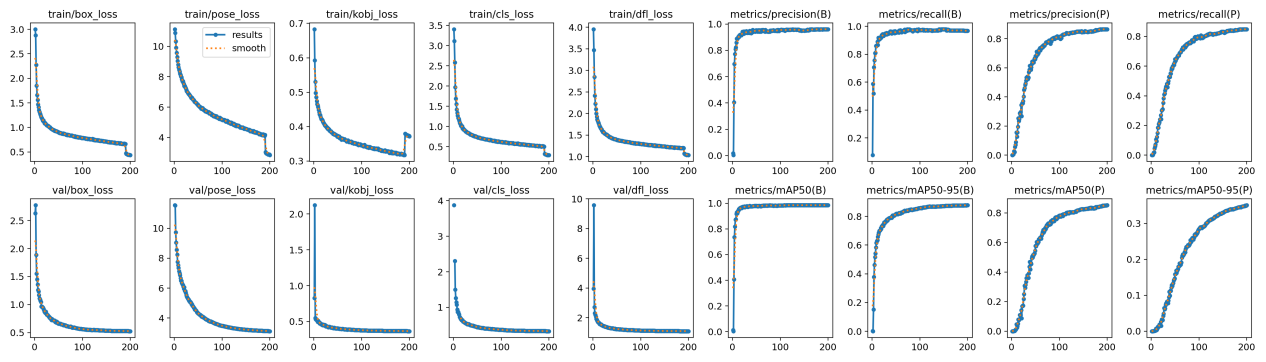
1694张图片用于训练，epochs=200，关键点个数=24

RTX4060，训练67.715h

模型训练、验证日志：



混淆矩阵 召回率-置信度曲线

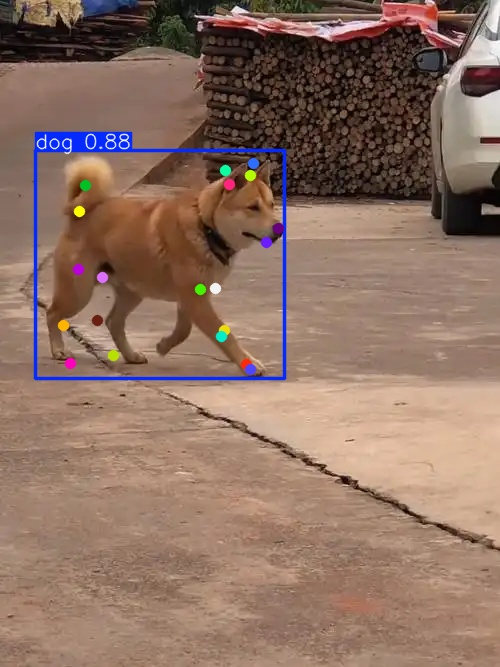


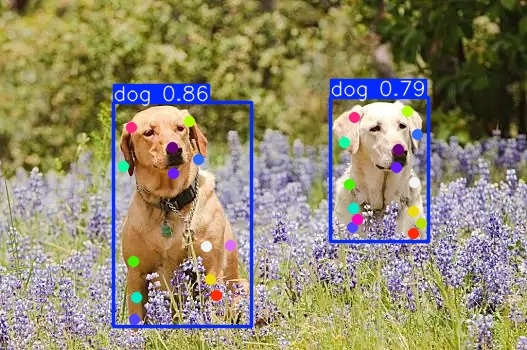
损失函数、mAP曲线、框&&点召回率与精度曲线



验证集表现

模型测试：





模型量化并部署在边缘计算节点RDK X5的教程：

<https://developer.d-robotics.cc/rdk_doc/Advanced_development/toolchain_development/intermediate/ptq_process>

# **第一步：模型准备**

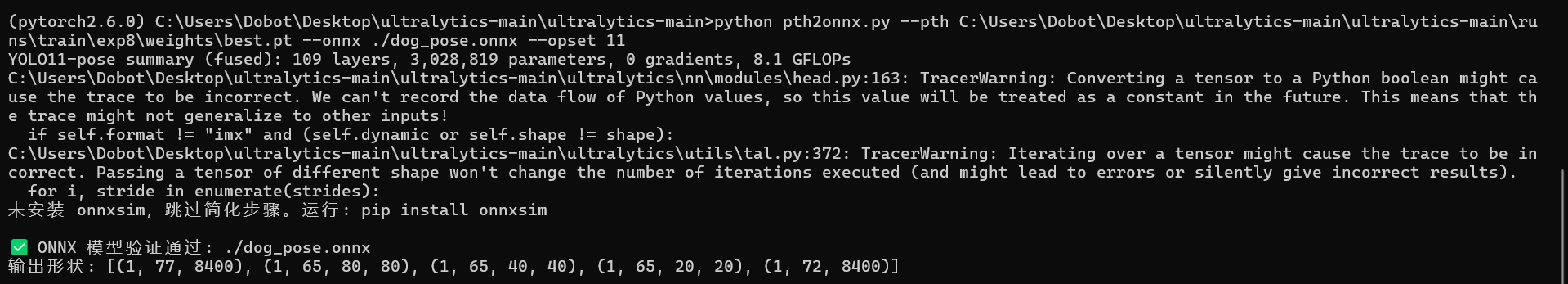
<https://developer.d-robotics.cc/rdk_doc/Advanced_development/toolchain_development/intermediate/ptq_process#model_preparation>

# **第二步：模型验证**

<https://developer.d-robotics.cc/rdk_doc/Advanced_development/toolchain_development/intermediate/ptq_process#model_check>

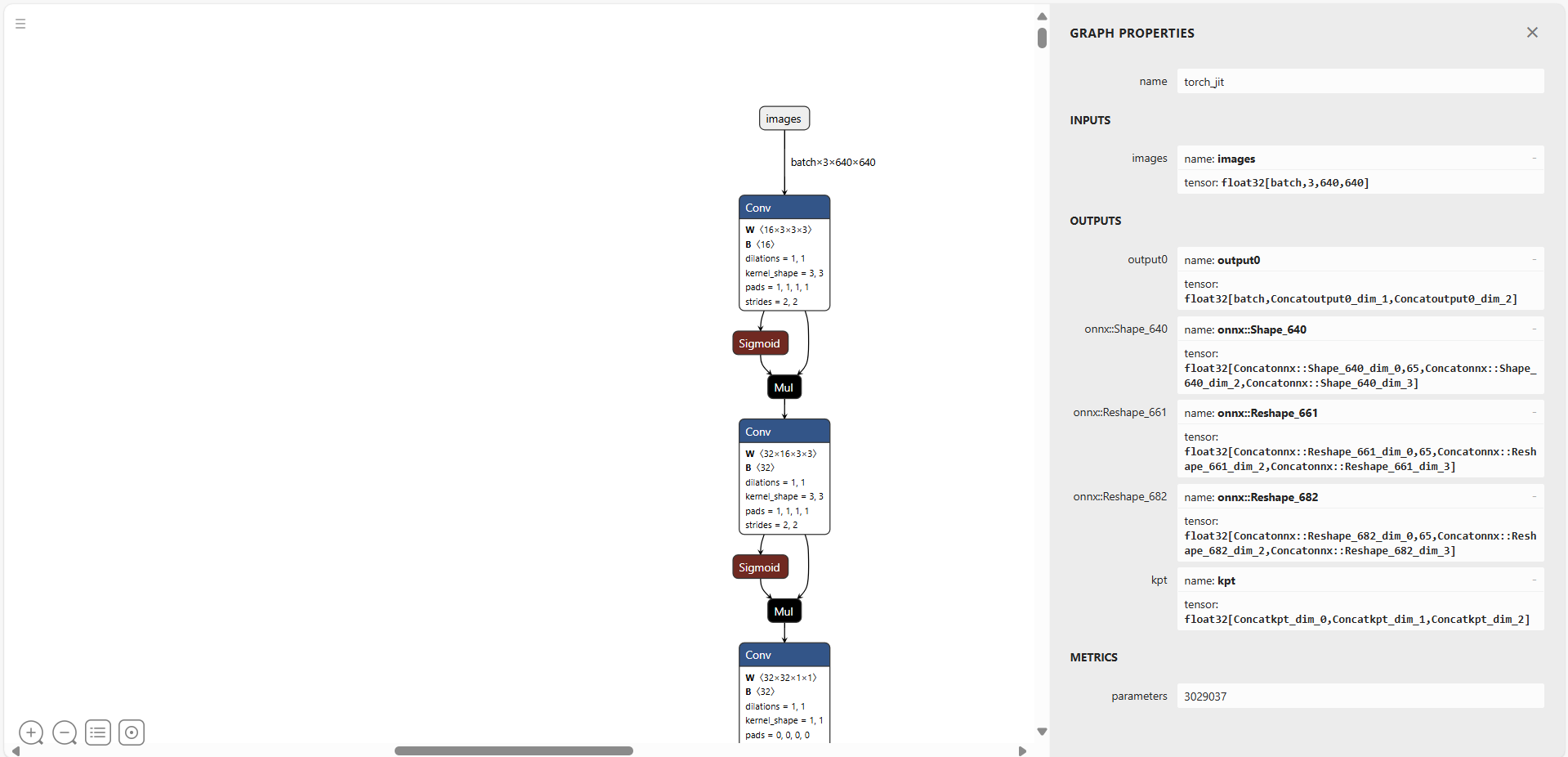
# **第三步：[Pytorch导出ONNX及模型可视化教程](https://forum.d-robotics.cc/t/topic/26296)**

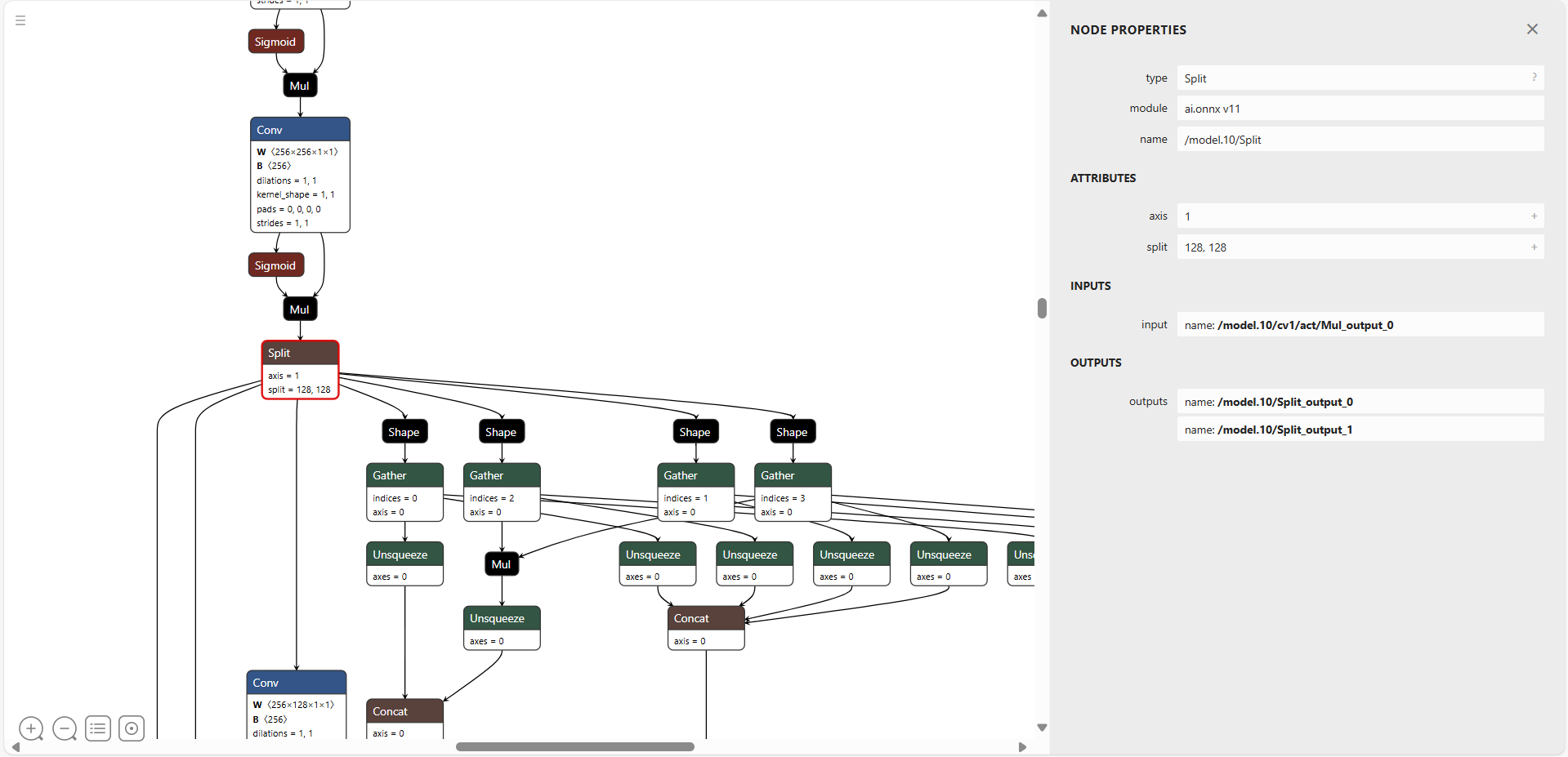
<https://forum.d-robotics.cc/t/topic/26296>



.pth-->.onnx与模型验证

使用netron.app打开onnx：





第四步:工具链生产环境

使用RDK X5内置的hb\_mapper验证onnx:

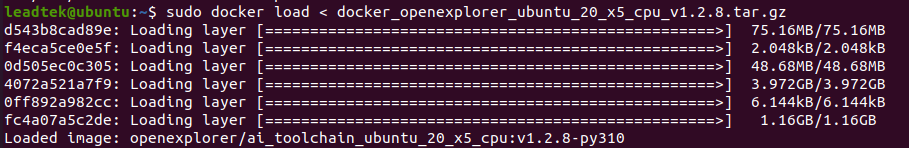
民间教程：

<https://blog.csdn.net/xiongqi123123/article/details/145265525>

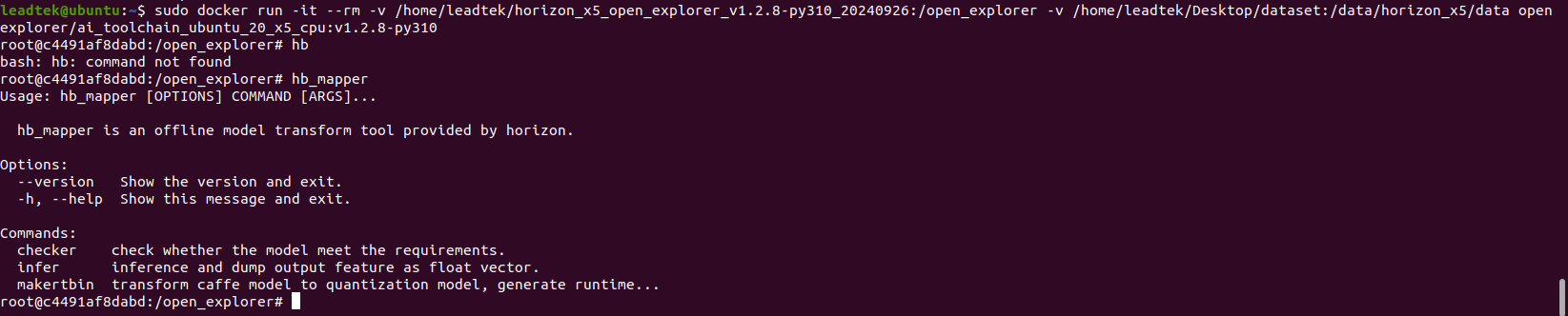
官网教程：

<https://developer.d-robotics.cc/rdk_doc/Advanced_development/toolchain_development/intermediate/environment_config#%E4%BA%A4%E4%BB%98%E7%89%A9%E4%BD%BF%E7%94%A8%E8%AF%B4%E6%98%8E>

首先下载安装Docker、OE交付包、CPU Docker镜像，然后：



导入镜像



安装docker-->配置镜像源文件-->docker容器成功启动

***小插曲：***一开始在RDK X5板子上部署docker容器，后报错如下：

sunrise@ubuntu:/mnt/usb$ docker run -it --rm -v

/horizon\_x5\_open\_explorer\_v1.2.6-py310\_20240724:/open\_explorer -v

/media/8D67-FE63/dataset:/data openexplorer/ai\_toolchain\_ubuntu\_20\_x5\_cpu:v1.2.6-py10

WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested

exec /bin/bash: exec format error

经过分析讨论，发现是平台不兼容的问题，于是转移到PC端Linux虚拟机进行部署



已经解决的报错

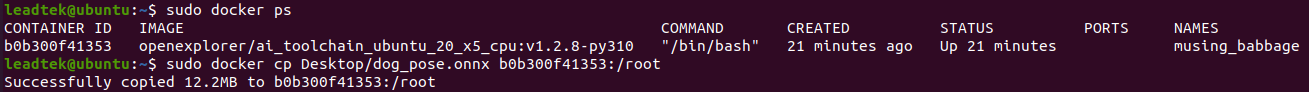
第五步：PTQ（训练后量化）

官方教程：

<https://developer.d-robotics.cc/api/v1/fileData/x5_doc-v126cn/oe_mapper/source/ptq/ptq_workflow.html>

民间教程：

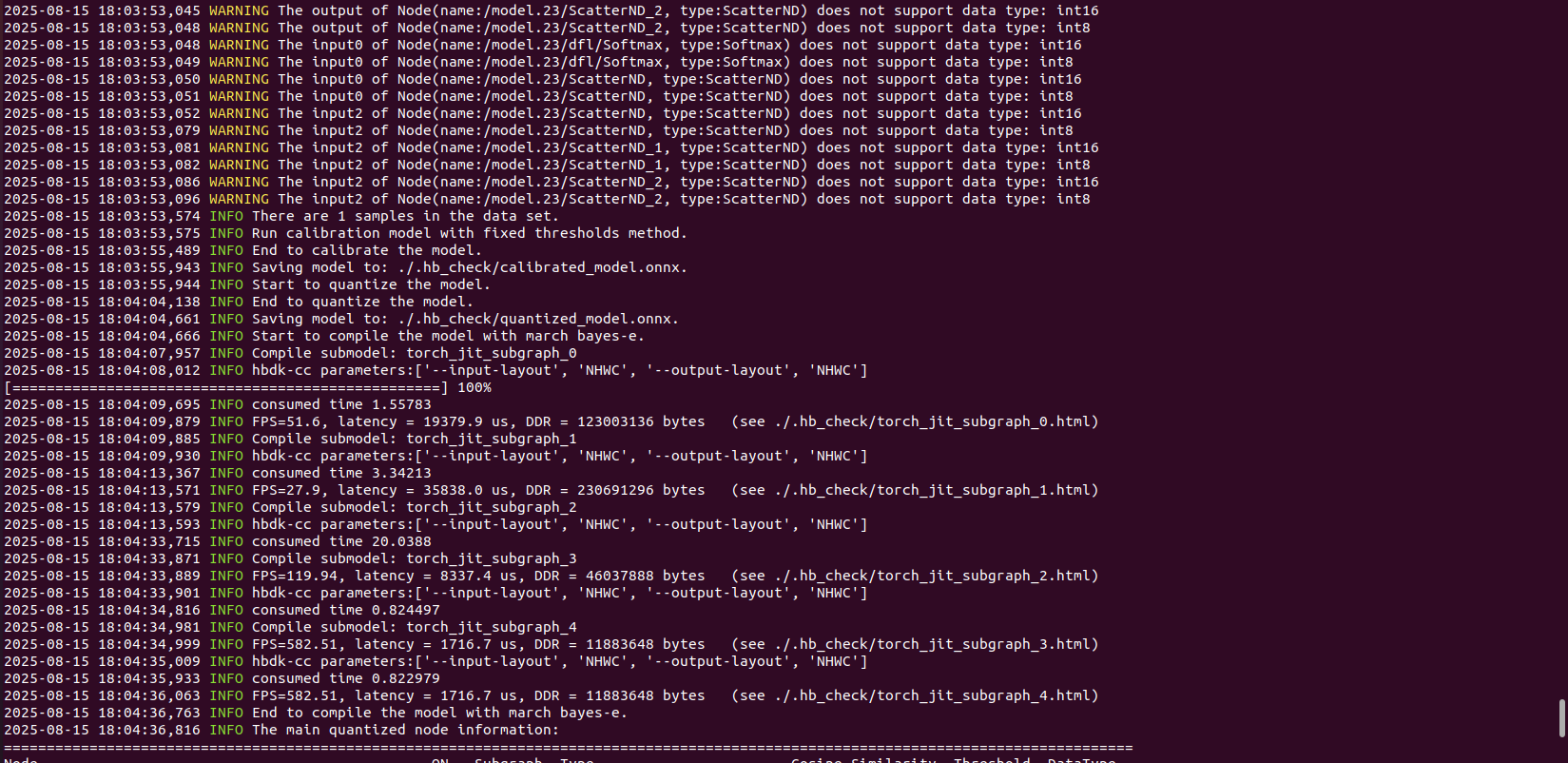
<https://blog.csdn.net/qq_34912332/article/details/147415926>

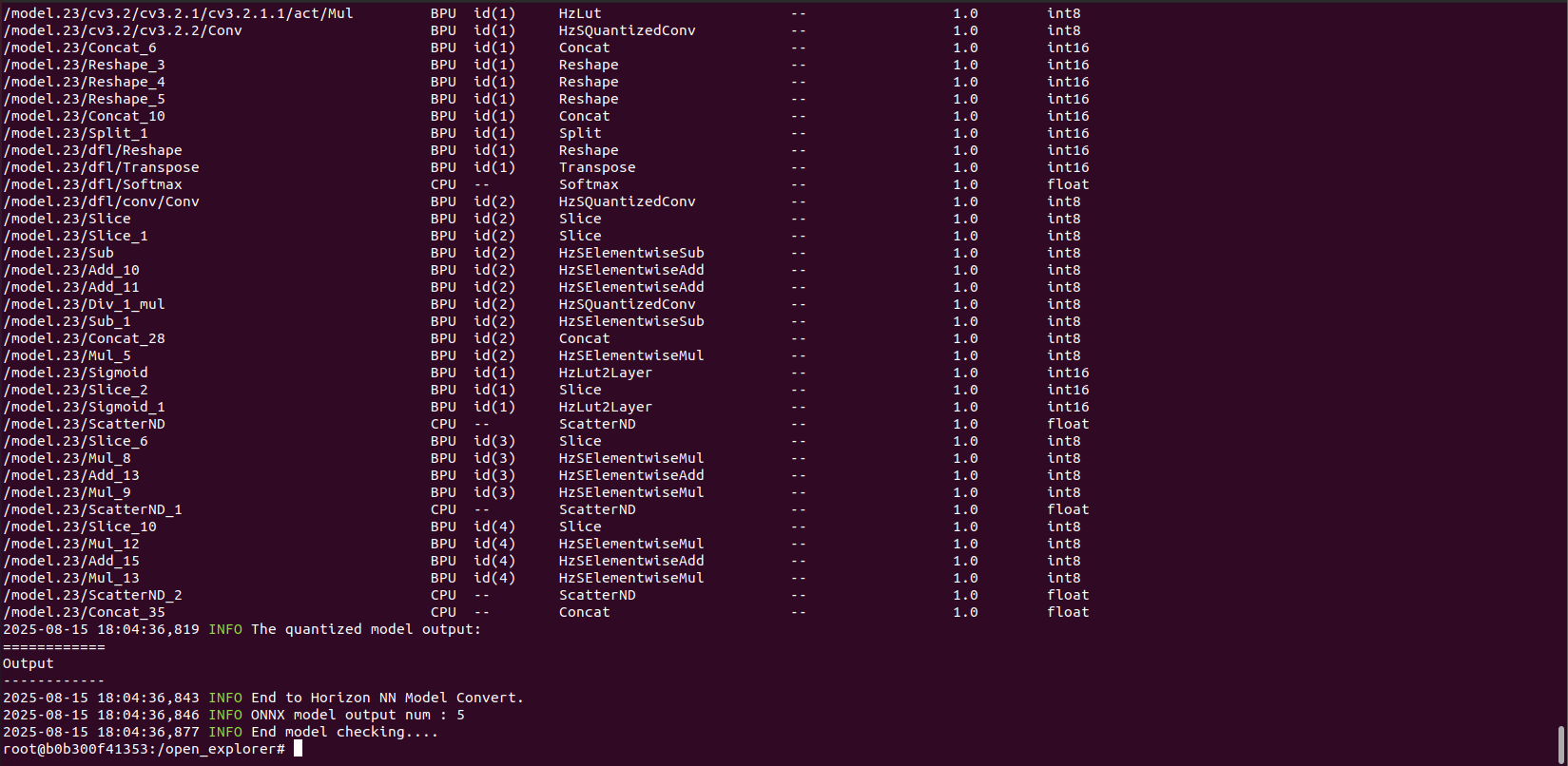


导入.onnx模型至Docker容器



使用 hb\_mapper checker 工具验证模型

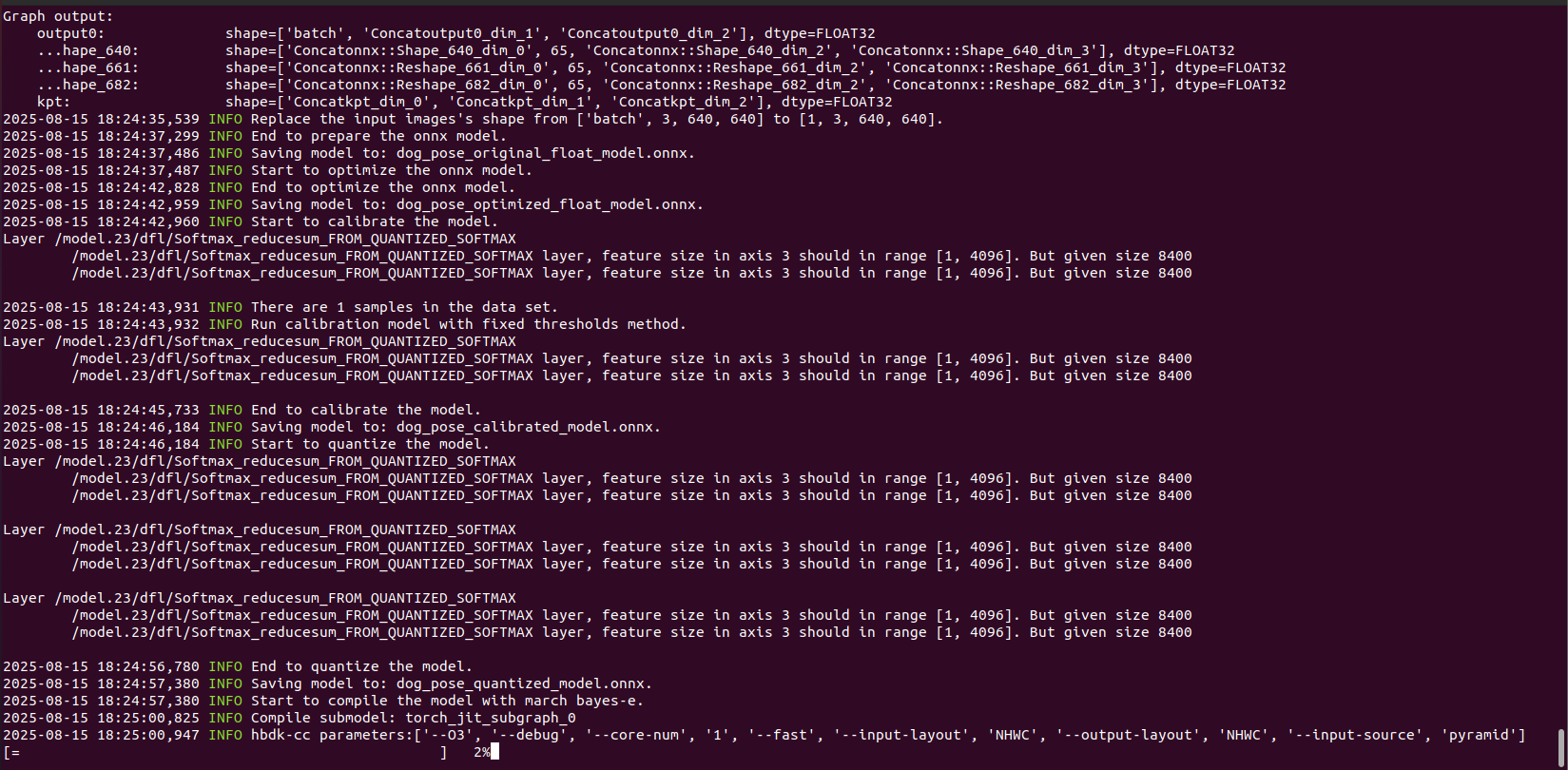




模型验证通过

**不开启fast-perf模式转换：发现FPS低，于是开启fast-perf模式进行转换**

使用hb\_mapper marketbin，同时开启 fast-perf 模式进行模型转换：



onnx-->bin模型转换中

下周计划：

根据RDK X5 ONNX模型量化yaml文件模板，填写参数，并使用yaml文件再次进行onnx-->bin的格式转换：



算法开发-->控制机器人运动的全流程总结：

主要有几点：三维重建，姿态检测，机器人控制是通过ros进行的，路径规划是英伟达的算法

要控制机械臂，需要先导入机械臂的模型