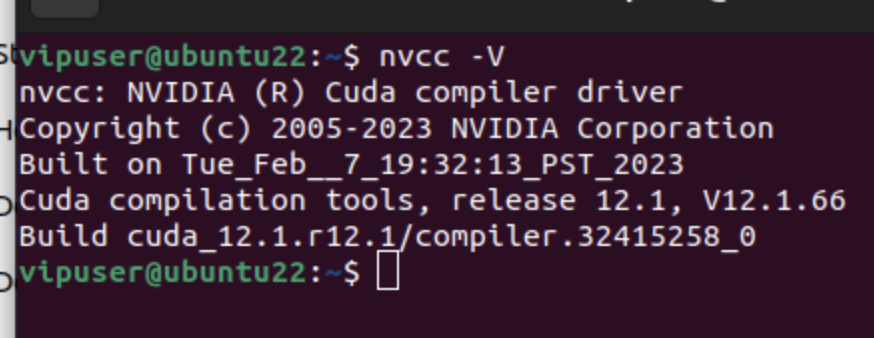
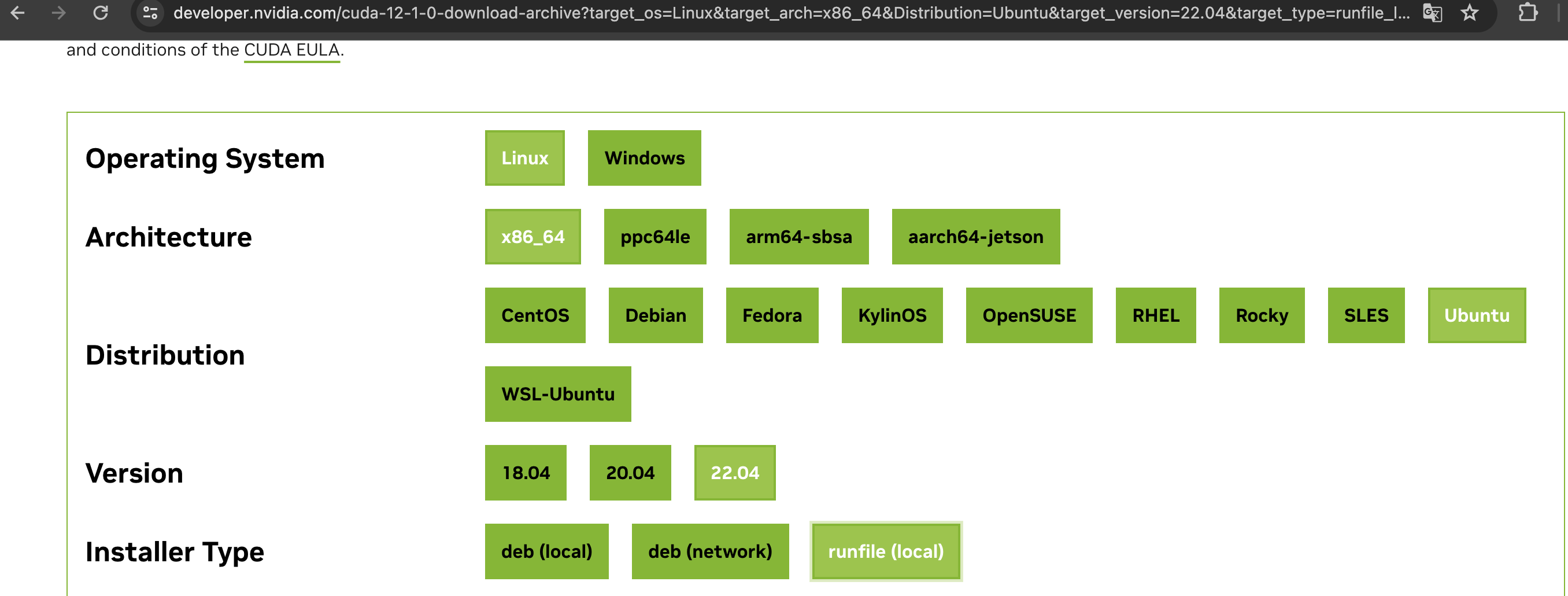
写一下自己搭建open-sora遇到的坑，搞了我三天时间才搭建好

首先这个项目要注意自己的CUDA版本和Torch版本要一致，我之前用的是Ubuntu20.04版本，发现这个版本根本就没有对应的apex，所以换了Ubuntu22.04，最好用这个版本

先用nvcc -V看一下自己cuda版本



如果没有的话，直接去官网安装cuda，<https://developer.nvidia.com/cuda-12-1-0-download-archive?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=22.04>



然后按照官网的步骤进行安装

wget <https://developer.download.nvidia.com/compute/cuda/12.1.0/local_installers/cuda_12.1.0_530.30.02_linux.run>

sudo sh cuda\_12.1.0\_530.30.02\_linux.run

然后再修改变量vim ~/.bashrc

export PATH=/usr/local/cuda-12.1/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/local/cuda-12.1/lib64:$LD\_LIBRARY\_PATH

export CUDA\_HOME=/usr/local/cuda-12.1:$CUDA\_HOME

然后保存后再source ~/.bashrc

然后再安装Torch

pip install torch==2.2.1 torchvision torchaudio==2.2.1 --index-url <https://download.pytorch.org/whl/cu121>

这里的cu121表示CUDA是12.1，目前CUDA12.2还没有对应的Torch安装包，先用到121也是可以的，如果你的CUDA是11.8，那你需要修改成cu118，否则后面会报错，这里是一个坑（推荐最好用最新的）

然后安装ninja

pip install packaging ninja

再然后安装flash-attn

这里我们需要选择对应相近python、cuda和torch版本的包安装

比如我是torch2.2.1,cuda12.1,python3.10那么我们在这里<https://github.com/Dao-AILab/flash-attention/releases>

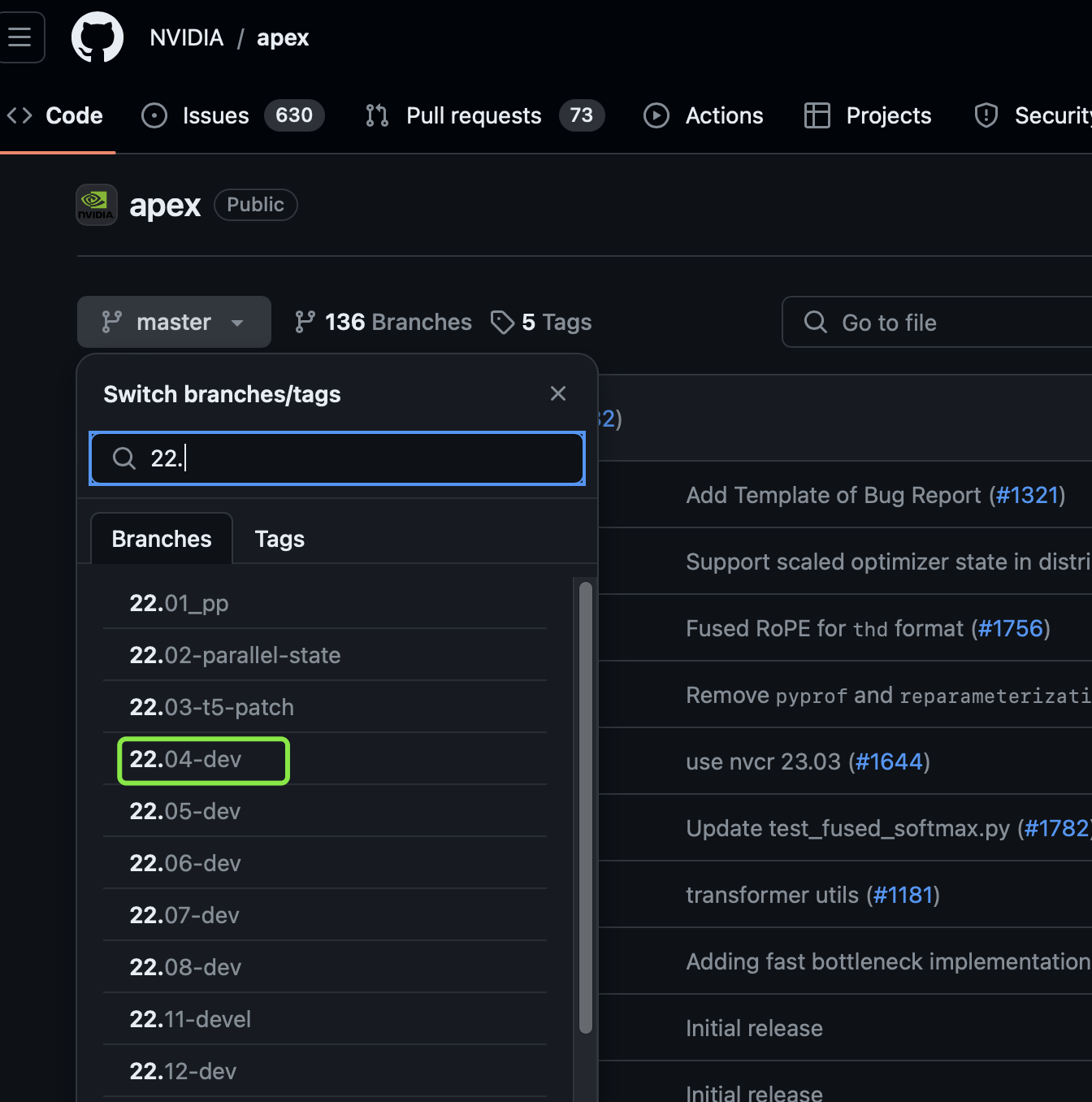
选择[flash\_attn-2.5.7+cu122torch2.2cxx11abiFALSE-cp310-cp310-linux\_x86\_64.whl](https://github.com/Dao-AILab/flash-attention/releases/download/v2.5.7/flash_attn-2.5.7+cu122torch2.2cxx11abiFALSE-cp310-cp310-linux_x86_64.whl)

下载好

然后用pip install [flash\_attn-2.5.7+cu122torch2.2cxx11abiFALSE-cp310-cp310-linux\_x86\_64.whl](https://github.com/Dao-AILab/flash-attention/releases/download/v2.5.7/flash_attn-2.5.7+cu122torch2.2cxx11abiFALSE-cp310-cp310-linux_x86_64.whl) --no-build-isolation来安装

最坑爹的地方来了，安装apex

首先不知道为啥最新版本的这个在我的Ubuntu为啥用不了，我们只能选择别人修改兼容后的apex分支进行安装



为啥是22.04-dev，是因为我的Ubuntu版本是22.04

下载这个包后cd apex

如果pip版本大于23.1，用这个命名安装

pip install -v --disable-pip-version-check --no-cache-dir --no-build-isolation --config-settings "--build-option=--cpp\_ext" --config-settings "--build-option=--cuda\_ext" ./

否则用下面这个命令

pip install -v --disable-pip-version-check --no-cache-dir --no-build-isolation --global-option**=**"--cpp\_ext" --global-option**=**"--cuda\_ext" ./

这个编译安装过程时间有点长

然后是安装xformers

pip install -U xformers==0.0.25 --index-url <https://download.pytorch.org/whl/cu121>

为什么是0.0.25，因为最新的0.0.25.post1版本已经不支持Torch2.2.1版本了

然后下载Open-Sora

git clone https://github.com/hpcaitech/Open-Sora

cd Open-Sora

安装git-lfs，用来下载大文件，这个安装命令大家网上可以自己找

然后下载模型文件

cd opensora/models *#不是OpenSora，是OpenSora代码仓目录下的Opensora里面有一个models目录*

*# 下载VAE模型*

git clone https://www.modelscope.cn/AI-ModelScope/sd-vae-ft-ema.git

*# 下载ST-dit模型*

git clone https://www.modelscope.cn/AI-ModelScope/Open-Sora.git

*# 下载text-encoder模型*

cd text-encoder

git clone https://www.modelscope.cn/AI-ModelScope/t5-v1\_1-xxl.git

也可以用python脚本进行下载

**import** torch

**from** modelscope **import** snapshot\_download, AutoModel, AutoTokenizer

**import** os

snapshot\_download('AI-ModelScope/sd-vae-ft-ema', cache\_dir**=**'./Open-Sora/opensora/models/', revision**=**'master')

snapshot\_download('AI-ModelScope/Open-Sora', cache\_dir**=**'./Open-Sora/opensora/models/', revision**=**'master')

snapshot\_download('AI-ModelScope/t5-v1\_1-xxl', cache\_dir**=**'./Open-Sora/opensora/models/text\_encoder', revision**=**'master')

然后修改文件（Open-Sora/configs/opensora/inference/16x256x256.py）

num\_frames **=** 16

fps **=** 24 **//** 3

image\_size **=** (256, 256)

*# Define model*

model **=** dict(

type**=**"STDiT-XL/2",

space\_scale**=**0.5,

time\_scale**=**1.0,

enable\_flashattn**=**True,

enable\_layernorm\_kernel**=**True,

from\_pretrained**=**"放OpenSora-v1-HQ-16x256x256.pth的路径",

)

vae **=** dict(

type**=**"VideoAutoencoderKL",

from\_pretrained**=**"放sd-vae-ft-ema文件夹的路径",

)

text\_encoder **=** dict(

type**=**"t5",

from\_pretrained**=**"放t5-v1\_1-xxl文件夹的路径",

model\_max\_length**=**120,

)

scheduler **=** dict(

type**=**"iddpm",

num\_sampling\_steps**=**100,

cfg\_scale**=**7.0,

)

dtype **=** "fp16"

*# Others*

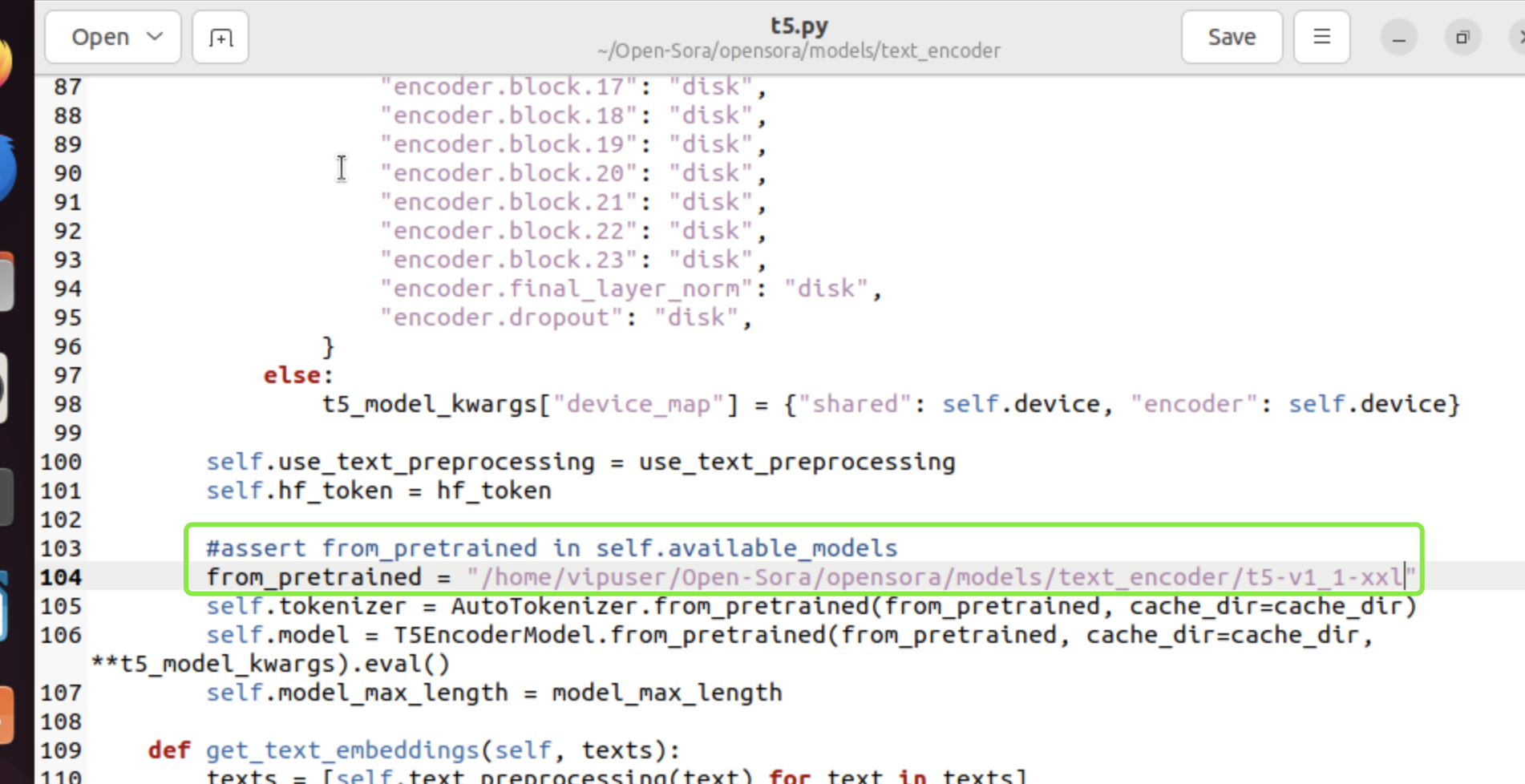
batch\_size **=** 2

seed **=** 42

prompt\_path **=** "./assets/texts/t2v\_samples.txt"

save\_dir **=** "./outputs/samples/"

然后再修改opensora/models/text\_encoder/t5.py文件里面这一行

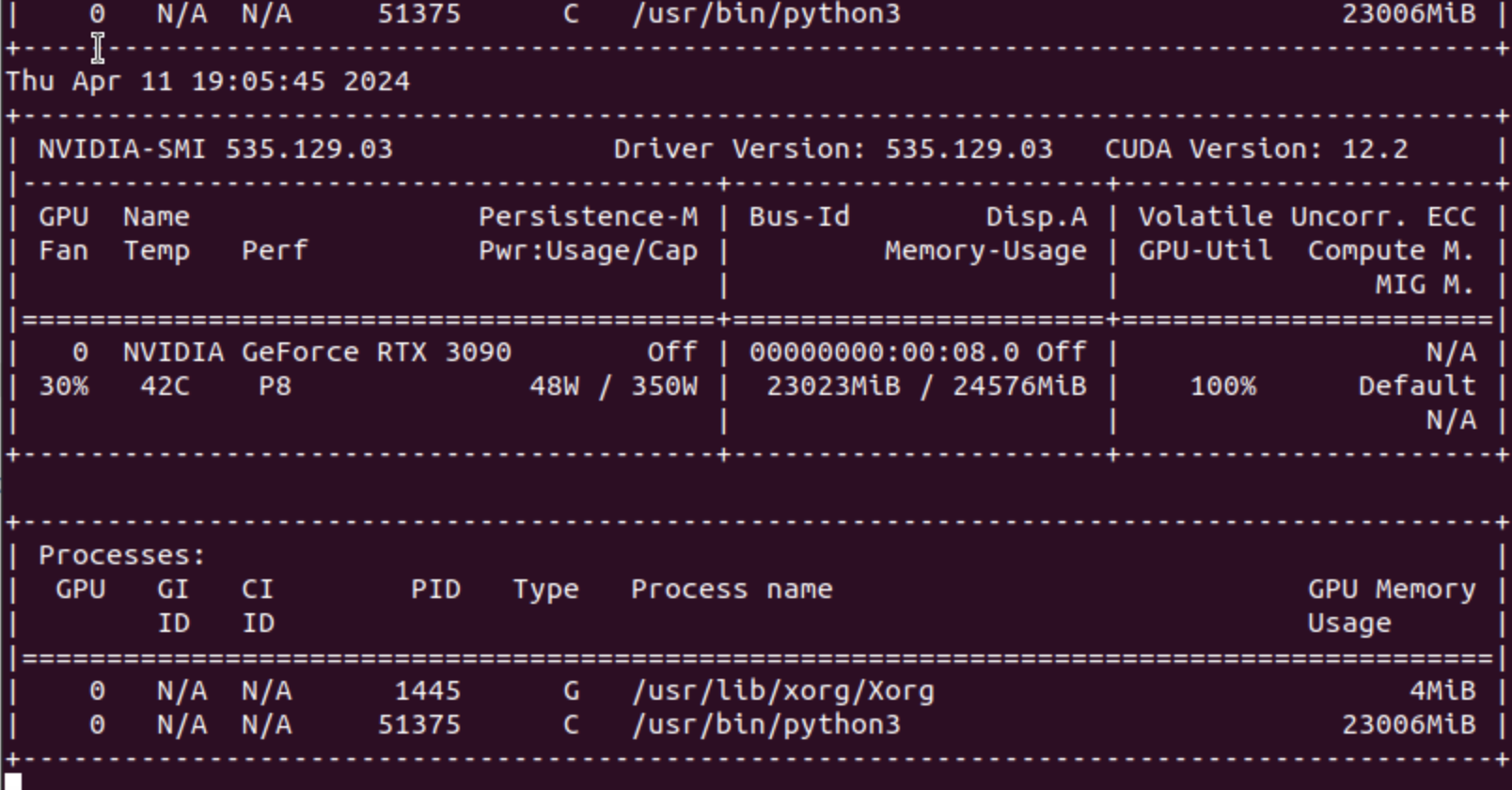


最后安装pip install -v .

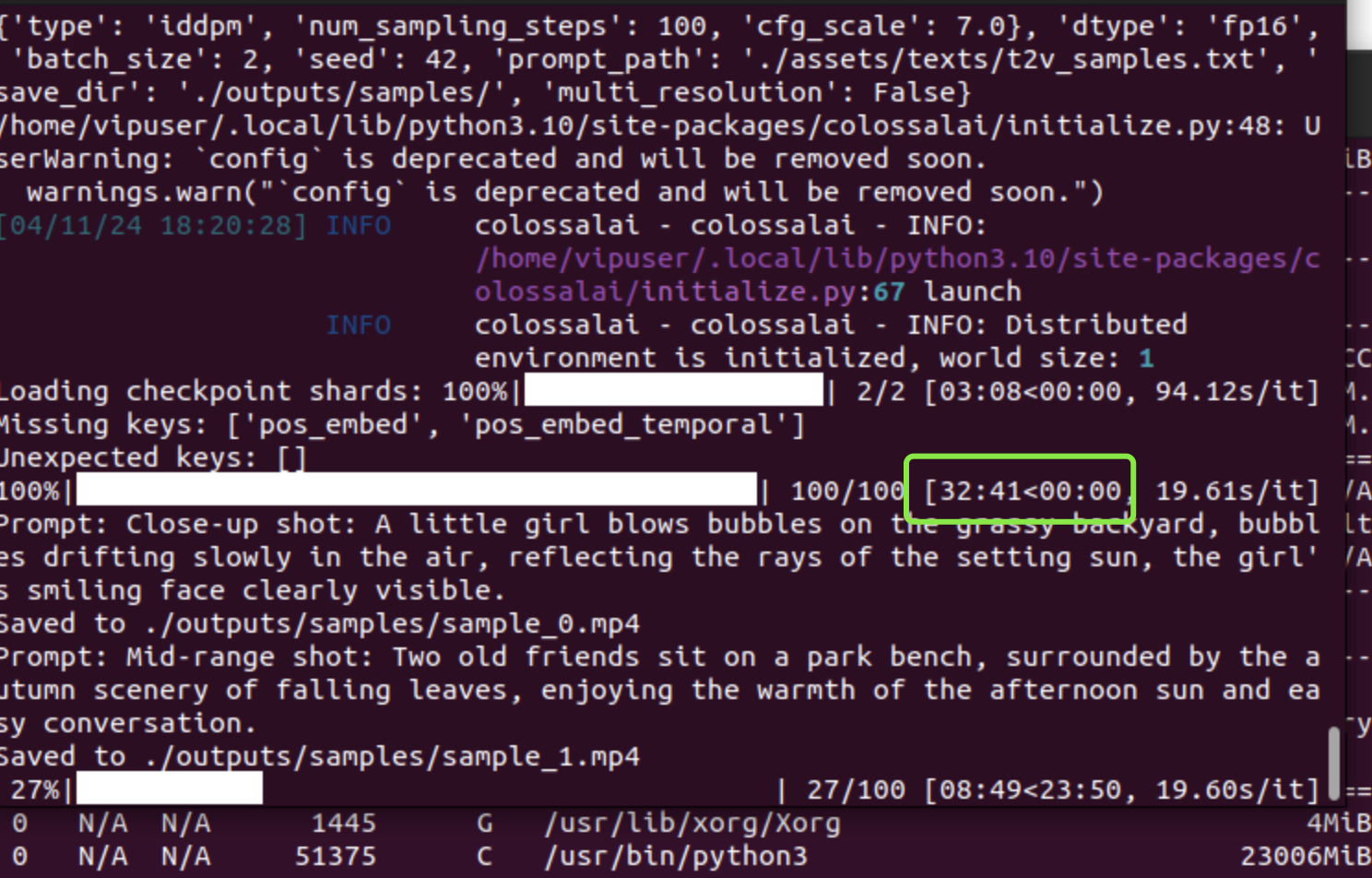
然后运行

torchrun --standalone --nproc\_per\_node 1 scripts/inference.py configs/opensora/inference/16x256x256.py

我这里用的是24G显存的显卡，运行一次差不多打满了



生成一个2秒视频耗时近半个小时



不得不说国产道理还是很漫长的