# Deformable

目标动！相机动！

## D-NeRF/hook子数据集中：

### 生成质量（是否跳帧、模糊）

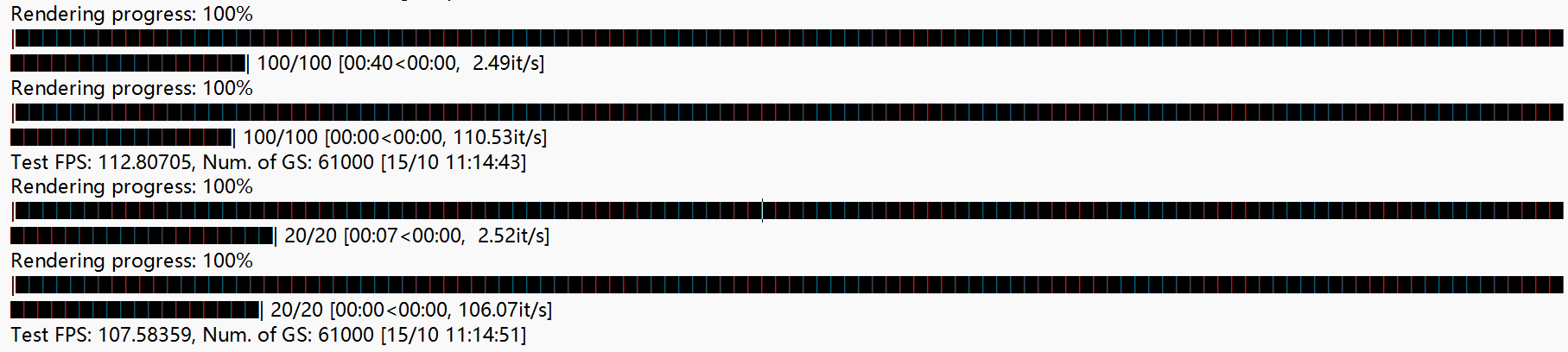
质量指标

SSIM : 0.9857901

PSNR : 37.0039139

LPIPS: 0.0161794

### 速度



训练集渲染进度：

模型渲染训练集时的迭代速度（2.49 it/s），表示每秒可以生成 2.49 帧图像。

训练集 FPS 测试：

在训练集上测试模型推理速度的 FPS 输出，110.53 it/s 是渲染的迭代速度，112.80705 FPS 则是最终推理速度，表示每秒生成 112 帧图像。

测试集渲染进度：

在 测试集 上的渲染进度。渲染的迭代速度为 2.52 it/s，表示每秒渲染 2.52 帧图像。

测试集 FPS 测试：

在 测试集 上推理时的 FPS 测试结果。106.07 it/s 是渲染速度，而 107.58359 FPS 是模型在推理测试集时的帧率。

### 训练和推理时的GPU要求

实验室的服务器可以满足

## NeRF/standup子数据集中：

### 尝试GUI训练

python train\_gui.py -s ./data/D-NeRF/standup -m output/D-NeRF-standup-exp --eval --is\_blender

问题：

(base) root@localhost:~/SHARE/Deformable-3D-Gaussians# python train\_gui.py -s ./data/D-NeRF/standup -m output/D-NeRF-standup-exp --eval --is\_blender Optimizing output/D-NeRF-standup-exp

Output folder: output/D-NeRF-standup-exp [04/11 03:52:03]

Tensorboard not available: not logging progress [04/11 03:52:03]

Found transforms\_train.json file, assuming Blender data set! [04/11 03:52:12]

Reading Training Transforms [04/11 03:52:12]

Reading Test Transforms [04/11 03:52:48]

Generating random point cloud (100000)... [04/11 03:52:54]

Loading Training Cameras [04/11 03:52:55]

Loading Test Cameras [04/11 03:53:09]

Number of points at initialisation : 100000 [04/11 03:53:11]

Training progress: 0%| | 0/40000 [00:00<?, ?it/s]

Glfw Error 65544: X11: Failed to open display localhost:12.0

Glfw Error 65537: The GLFW library is not initialized

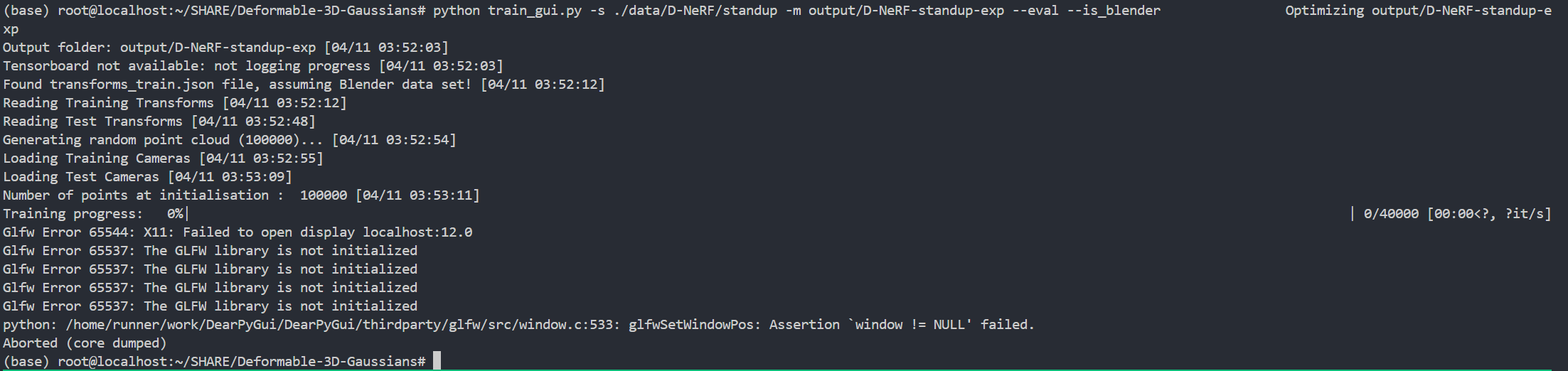
Glfw Error 65537: The GLFW library is not initialized

Glfw Error 65537: The GLFW library is not initialized

Glfw Error 65537: The GLFW library is not initialized

python: /home/runner/work/DearPyGui/DearPyGui/thirdparty/glfw/src/window.c:533: glfwSetWindowPos: Assertion `window != NULL' failed.

Aborted (core dumped)





原因：容器里面没有安装x11相关库，下面进行安装：

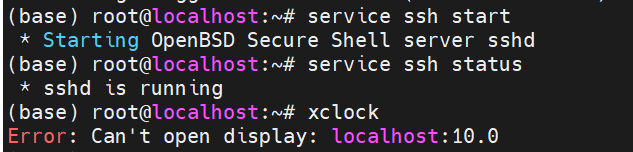
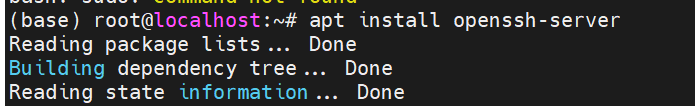
(base) root@localhost:~# apt-get install -y x11-apps libx11-6 libxext6



安装成功啦！

接下来add xauth还是不行

又在容器里安装了openssh-server



启动，还是不行

不启动GUI训练：

python train.py -s ./data/D-NeRF/standup -m output/D-NeRF-standup-exp --eval --is\_blender

### 生成质量

Best PSNR = 43.844539642333984 in Iteration 38000

### 速度

## NeRF-DS/as子数据集中：

### (1)生成质量（是否跳帧、模糊）

质量指标

SSIM : 0.8844547

PSNR : 26.1240101

LPIPS: 0.1805718

### (2)速度



## NeRF-DS/basin子数据集中：

### (1)生成质量（是否跳帧、模糊）

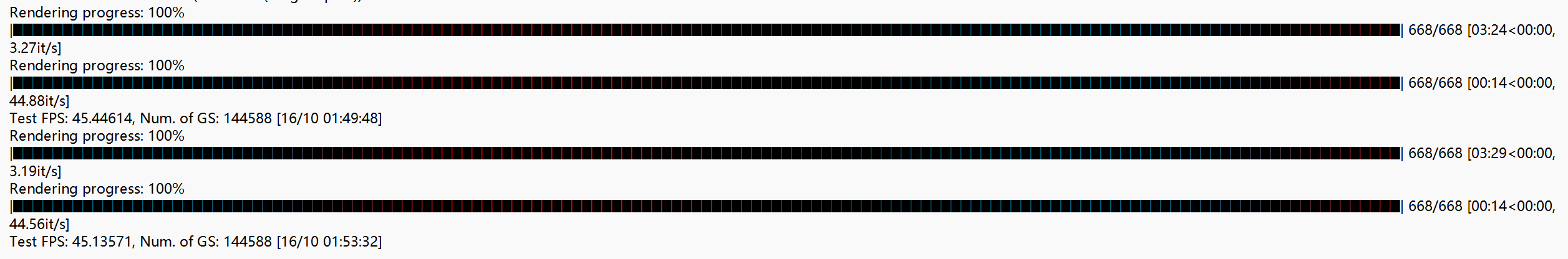
质量指标

SSIM : 0.7941079

PSNR : 19.6068096

LPIPS: 0.1915355

### (2)速度



# MVSplat

MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-View Images[ECCV-2024]

## GPU需求

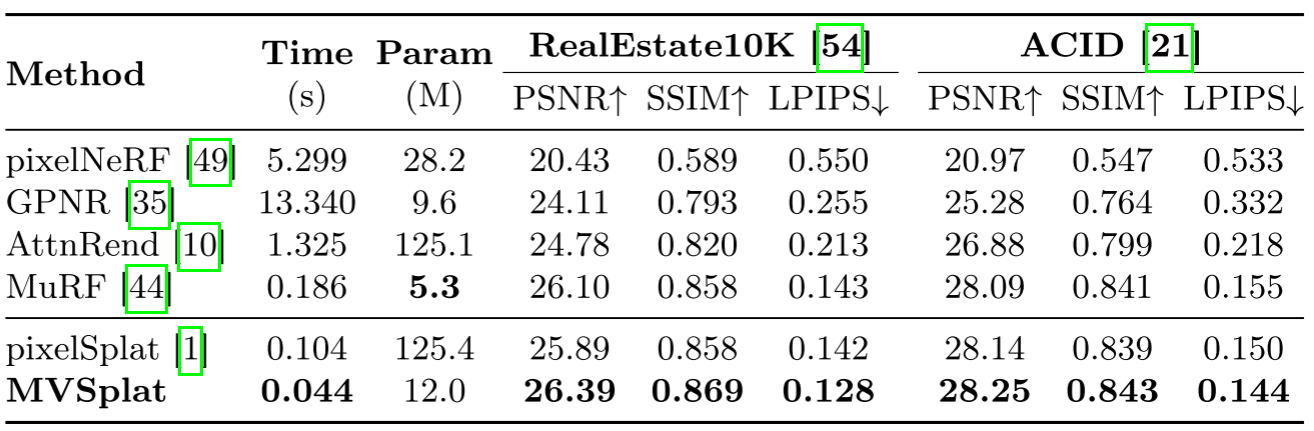
All models are trained on a single A100 GPU for 300,000 iterations with the Adam [20]

optimizer.（80GB）

on multiple GPUs with smaller RAM by setting a smaller data\_loader.train.batch\_size per GPU

复现使用4张GPU（24G）采用DDP分布式训练

原数据集：大型RealEstate10K和ACID数据集。 RealEstate10K包含从YouTube下载的房地产视频，拆分为67477个训练场景和7289个测试场景；ACID包含航拍无人机拍摄的自然场景，拆分为11075个训练场景和1972个测试场景。



## re10k\_subset数据集（从YouTube下载的房地产视频）：

psnr 26.021498153084202

ssim 0.8743962833755895

lpips 0.11940941283185232

encoder: 33 calls, avg. 0.12000023957454797 seconds per call

decoder: 99 calls, avg. 0.004638361208366625 seconds per call

psnr 26.021498153084202

ssim 0.8743962833755895

lpips 0.11940941283185232

encoder: 33 calls, avg. 0.12705410610545764 seconds per call

decoder: 99 calls, avg. 0.004810092425105547 seconds per call

psnr 26.021498153084202

ssim 0.8743962833755895

lpips 0.11940941283185232

encoder: 33 calls, avg. 0.12947073849764737 seconds per call

decoder: 99 calls, avg. 0.004894066338587289 seconds per call

psnr 26.021498153084202

ssim 0.8743962833755895

lpips 0.11940941283185232

encoder: 33 calls, avg. 0.13046956062316895 seconds per call

decoder: 99 calls, avg. 0.004929559399383236 seconds per call

## Acid\_subset数据集（无人机拍摄的自然场景图像）：

psnr 27.82588870708759

ssim 0.8714552934353168

lpips 0.12284397419828635

encoder: 8 calls, avg. 0.11508834362030029 seconds per call

decoder: 24 calls, avg. 0.004530260960261027 seconds per call

psnr 27.82588870708759

ssim 0.8714552934353168

lpips 0.12284397419828635

encoder: 8 calls, avg. 0.11397555470466614 seconds per call

decoder: 24 calls, avg. 0.004415333271026611 seconds per call

psnr 27.82588870708759

ssim 0.8714552934353168

lpips 0.12284397419828635

encoder: 8 calls, avg. 0.11869743466377258 seconds per call

decoder: 24 calls, avg. 0.004565447568893433 seconds per call

psnr 27.82588870708759

ssim 0.8714552934353168

lpips 0.12284397419828635

encoder: 8 calls, avg. 0.12178334593772888 seconds per call

decoder: 24 calls, avg. 0.0046697258949279785 seconds per call

## 

## DTU数据集：

psnr 13.942063219845295

ssim 0.47347488859668374

lpips 0.3857552495319396

encoder: 59 calls, avg. 0.12766714015249478 seconds per call

decoder: 59 calls, avg. 0.012667769092624471 seconds per call

psnr 13.942063219845295

ssim 0.47347488859668374

lpips 0.3857552495319396

encoder: 59 calls, avg. 0.1259312629699707 seconds per call

decoder: 59 calls, avg. 0.012464301060822051 seconds per call

psnr 13.942063219845295

ssim 0.47347488859668374

lpips 0.3857552495319396

encoder: 59 calls, avg. 0.12739642595840714 seconds per call

decoder: 59 calls, avg. 0.012497029062044822 seconds per call

psnr 13.942063219845295

ssim 0.47347488859668374

lpips 0.3857552495319396

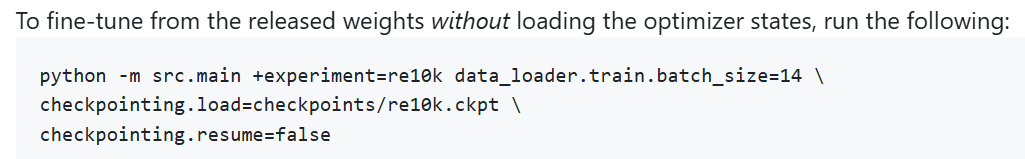
encoder: 59 calls, avg. 0.1279546325489626 seconds per call

decoder: 59 calls, avg. 0.01282871779748949 seconds per call

## 模型具有泛化能力：

使用re10k.ckpt和acid.ckpt预训练模型来评估DTU数据集，效果不错。

## 可以进一步微调：



利用预训练模型继续去训练新的数据集，比如针对特定教室场景的数据集。

修改小模块

## 存在问题：

Real Estate 10k和ACID原数据集分别为500GB和160GB，太庞大，难以复现。

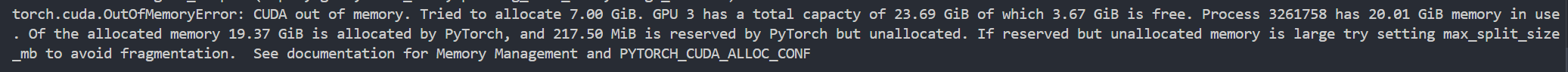
使用项目提供的子数据集re10k\_subset和acid\_subset进行推理，psnr等各项指标均低于论文给出的结果值。

而DTU数据集的测试结果与论文结果一致。

## 自己训练：

## ddp训练，利用子数据集re10k\_subset

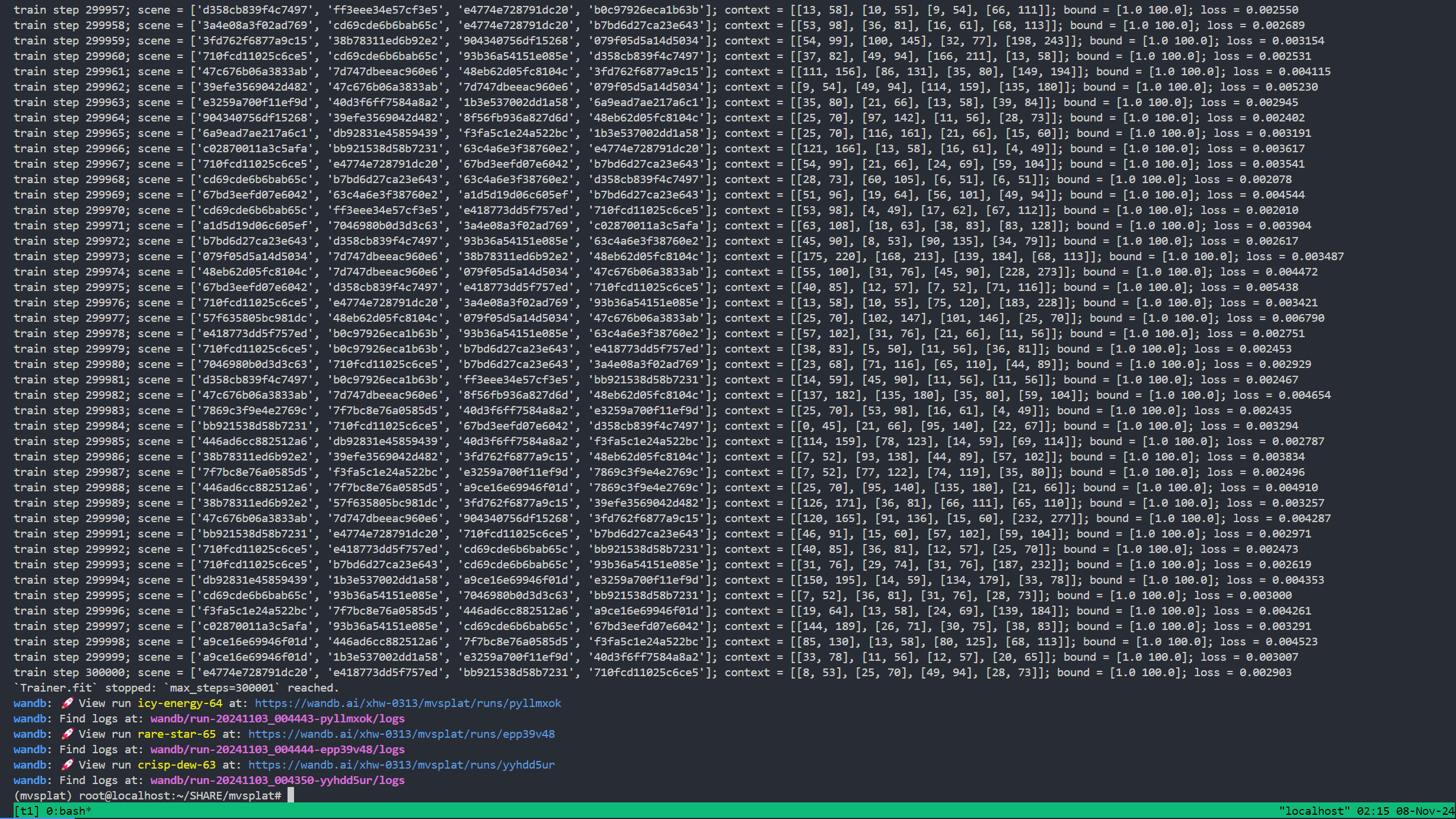
python -m src.main +experiment=re10k data\_loader.train.batch\_size=14



显存不足

python -m src.main +experiment=re10k data\_loader.train.batch\_size=4

3块3090GPU，训练30万steps，将近6天



**推理结果**

**在re10k子数据集上：**

**psnr 21.47158213665611**

**ssim 0.7745077359048944**

**lpips 0.22853690248570943**

encoder: 33 calls, avg. 0.12605281309647995 seconds per call

decoder: 99 calls, avg. 0.004771227788443517 seconds per call

psnr 21.47158213665611

ssim 0.7745077359048944

lpips 0.22853690248570943

encoder: 33 calls, avg. 0.1267075321891091 seconds per call

decoder: 99 calls, avg. 0.004877061554879852 seconds per call

psnr 21.47158213665611

ssim 0.7745077359048944

lpips 0.22853690248570943

encoder: 33 calls, avg. 0.1326753804177949 seconds per call

decoder: 99 calls, avg. 0.005132501775568182 seconds per call

**在DTU数据集上：**

**psnr 11.022906579077244**

**ssim 0.23344479239312932**

**lpips 0.6209165807813406**

encoder: 59 calls, avg. 0.12381026300333314 seconds per call

decoder: 59 calls, avg. 0.012311377767789162 seconds per call

psnr 11.022906579077244

ssim 0.23344479239312932

lpips 0.6209165807813406

encoder: 59 calls, avg. 0.13125287880331782 seconds per call

decoder: 59 calls, avg. 0.013426356396432651 seconds per call

psnr 11.022906579077244

ssim 0.23344479239312932

lpips 0.6209165807813406

encoder: 59 calls, avg. 0.13549692751997608 seconds per call

decoder: 59 calls, avg. 0.01347734160342459 seconds per call

# 横向对比

选择Deformable 3D Gaussians深入学习：

1. 考虑人形机器人室内场景重建任务特点：动态、小场景（准确度要求高）、速度

MVSplat 主要面向静态场景的3D重建

1. GPU需求（3090）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目  指标 | 应用场景 | 质量 | 速度 | GPU需求 | 泛化能力 |
| Deformable | 单目动态场景的高保真重建 | 对比基线提升超过10个点 | 超过30FPS | an NVIDIA RTX 3090 | —— |
| MVSplat | 稀疏多视图图像下的3D场景重建和新视角合成 |  | 22FPS | an A100 | 有 |