# Guide to install/use NeRFStudio & Gaussian official repository (GraphDECO)

**What this guide contains:**

- How to install everything from scratch
- How to start it once everything is set up
- How to run it on Nerfstudio + splatfacto
- How to run it on official 3DGS (not so harsh on hardware)

**Hardware Requirements from the official repo**

CUDA-ready GPU with Compute Capability 7.0+
24 GB VRAM (to train to paper evaluation quality)

I did it on a Compute Capability of 6.1 with 8GB VRAM (GTX 1070)
THIS was run on Ubuntu Linux

**Quick summary**

-NVIDIA drivers
-CUDA 12.4
-Conda env
-PyTorch (CUDA 12.4)
-ffmpeg
-colmap
-Nerfstudio+splatfacto OR official 3DGS

## 1. NVIDIA DRIVERS

```
sudo apt update

sudo apt install -y build-essential dkms linux-headers-$(uname -r) \
  curl wget ca-certificates software-properties-common

sudo ubuntu-drivers autoinstall


sudo reboot
```

---------- to check everything is ok run

nvidia-smi

output in my case:

...

0  NVIDIA GeForce GTX 1070  ...

## 2. CUDA TOOLKIT

source /etc/os-release

CUDA_REPO_PATH="ubuntu2404"

wget -O /tmp/cuda-keyring.deb \

"https://developer.download.nvidia.com/compute/cuda/repos/${CUDA_REPO_PATH}/x86_64/cuda-keyring_1.1-1_all.deb"

sudo dpkg -i /tmp/cuda-keyring.deb

sudo apt-get update

# Install CUDA 12.5 toolkit

sudo apt-get install -y cuda-toolkit-12-4

# Verify nvcc

/usr/local/cuda-12.5/bin/nvcc –version

### 2.1 ADD CUDA TO YOUR TOOLPATH

echo 'export PATH=/usr/local/cuda-12.5/bin:$PATH' >> ~/.bashrc

echo 'export LD_LIBRARY_PATH=/usr/local/cuda-12.5/lib64:${LD_LIBRARY_PATH}' >> ~/.bashrc

source ~/.bashrc

nvcc --version  # should work globally now

## 3. INSTALL CONDA (MINIFORGE)

wget https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh -O /tmp/miniforge.sh

bash /tmp/miniforge.sh -b -p $HOME/miniforge

$HOME/miniforge/bin/conda init bash

source ~/.bashrc

### 3.1 CREATE ENVIRONMENT FOR GSPLAT (GAUSSIAN SPLATTING)

conda create -y -n gsplat python=3.10

conda activate gsplat

## 4. INSTALL PyTorch with CUDA

IN the gsplat env:

conda install -y -c pytorch -c nvidia pytorch torchvision pytorch-cuda=12.4

#Quick test

python -c "import torch; print('PyTorch:', torch.__version__); print('CUDA compiled:', torch.version.cuda); print('CUDA available:', torch.cuda.is_available()); print('GPU:', torch.cuda.get_device_name(0) if torch.cuda.is_available() else 'N/A')"

It should print something like:

PyTorch: 2.6.0+cu124

CUDA compiled: 12.4

CUDA available: True

GPU: NVIDIA GeForce GTX 1070

---------- After this step you should choose between option A or B
In our case A means Nerfstudio + splatfacto (didn't work for me due to low hardware it kept crashing)
B: Official 3D Gaussian Splatting (GrappDECO repo)

## 5. I WILL SHOWCASE OPTION A:

# Make sure build deps are available

sudo apt install -y git cmake

# In the conda env:

pip install --upgrade pip

# Install nerfstudio + gsplat backend

pip install "nerfstudio>=1.0.0" "gsplat>=1.3.0"

# Install nerfstudio CLI wrapper (downloads helpers)

ns-install-cli

# Verify CLI works

ns-train --help

-start with preparing the pictures in a folder for my case "Home/IRONFILMS/ChurchCenter/JPG"

conda activate gsplat

ns-process-data images --data /home/fane/IRONFILMS/ChurchCenter/JPG --output-dir ./outputs/church_center

-here probably you will run into the first problem:

Could not find ffmpeg. Please install ffmpeg.

See https://ffmpeg.org/download.html for installation instructions.

FIX: sudo apt install -y ffmpeg

-now probably you will run into the second problem

Could not find COLMAP. Please install COLMAP.

See https://colmap.github.io/install.html for installation instructions.

FIX: sudo apt install -y colmap

If sudo apt install -y colmap didn’t work you can try

conda install -y -c conda-forge colmap

-now you should be able to run again:

ns-process-data images --data /home/fane/IRONFILMS/ChurchCenter/JPG --output-dir ./outputs/church_center

-start training on your dataset (images)

ns-train splatfacto --data ./outputs/church_center --max-num-iterations 30000

-first tried fix

export TORCH_COMPILE_DISABLE=1

ns-train splatfacto --data ./outputs/church_center --max-num-iterations 30000

-ended up with a black screen and a restart

-second tried fix

export TORCH_COMPILE_DISABLE=1

export TORCHDYNAMO_DISABLE=1

ns-train splatfacto --data ./outputs/church_center --max-num-iterations 30000

## 6. Time for OPTION B Official gaussian splat repo GraphDECO

-new terminal:

conda activate gsplat

cd ~

git clone https://github.com/graphdeco-inria/gaussian-splatting.git

cd gaussian-splatting

pip install plyfile tqdm

cd submodules/diff-gaussian-rasterization

mkdir -p build

cd build

cmake .. -DCMAKE_BUILD_TYPE=Release

make -j$(nproc)

-error because missing <cstdint>
-this file needs modification: submodules/diff-gaussian-rasterization/cuda_rasterizer/rasterizer_impl.h

#pragma once

#include <iostream>

#include <vector>

#include <cstdint>  // <-- this is the key one

#include "rasterizer.h"

#include <cuda_runtime_api.h>

-rebuild and it should work

cd ~/gaussian-splatting/submodules/diff-gaussian-rasterization/build

make -j$(nproc)

-training

cd ~/gaussian-splatting

export PYTHONPATH="${PYTHONPATH}:$(pwd)/submodules/diff-gaussian-rasterization:$(pwd)/submodules/simple-knn"

python train.py -s ~/outputs/church_center/colmap -m ~/outputs/church_center/3dgs_model

-possible problem

ModuleNotFoundError: No module named 'simple_knn'

### 6.1. Build the simple_knn extension

conda activate gsplat

cd ~/gaussian-splatting/submodules/simple-knn

python setup.py build_ext --inplace

### 6.2. Make Python see the submodules (PYTHONPATH)

cd ~/gaussian-splatting

export PYTHONPATH="${PYTHONPATH}:$(pwd)/submodules/diff-gaussian-rasterization:$(pwd)/submodules/simple-knn"

### 6.3. Run the training command

cd ~/gaussian-splatting

export PYTHONPATH="${PYTHONPATH}:$(pwd)/submodules/diff-gaussian-rasterization:$(pwd)/submodules/simple-knn"

python train.py -s ~/outputs/church_center/colmap -m ~/outputs/church_center/3dgs_model

## 7. HOW TO WORK WITH A NEW SET

### 7.1. Prepare your new dataset folder

cd ~

mkdir -p new_scene/input

# Example: if your photos are in ~/Pictures/new_photos/

cp ~/Pictures/new_photos/*.JPG ~/new_scene/input/

# or whatever path your photos are at

## 7.2. Run the official converter

cd ~/gaussian-splatting

python convert.py -s ~/new_scene --colmap_executable /home/fane/miniforge/envs/gsplat/bin/colmap

## 7.3. Train 3DGS

cd ~/gaussian-splatting

export PYTHONPATH="${PYTHONPATH}:$(pwd)/submodules/diff-gaussian-rasterization:$(pwd)/submodules/simple-knn"

python train.py -s ~/new_scene -m ~/new_scene/3dgs_model

## 7.4. Render views

python render.py -m ~/new_scene/3dgs_model

WHAT TO FOLDERS TO KEEP:

- 3dgs_model
- images
- sparse