
# 3D Gaussian Splatting

## Implicit vs. Explicit 3D Scene Reconstructions

**IT University Copenhagen**

Spring Semester 2025

# Why is this cool?

**vSLAM** visual Simultaneous Localisation and Mapping

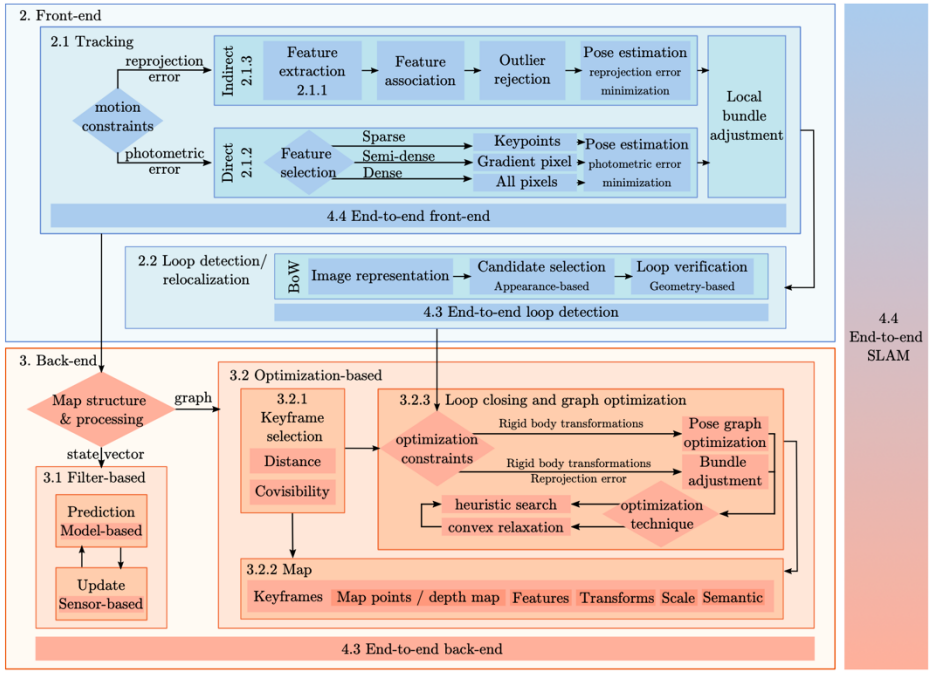
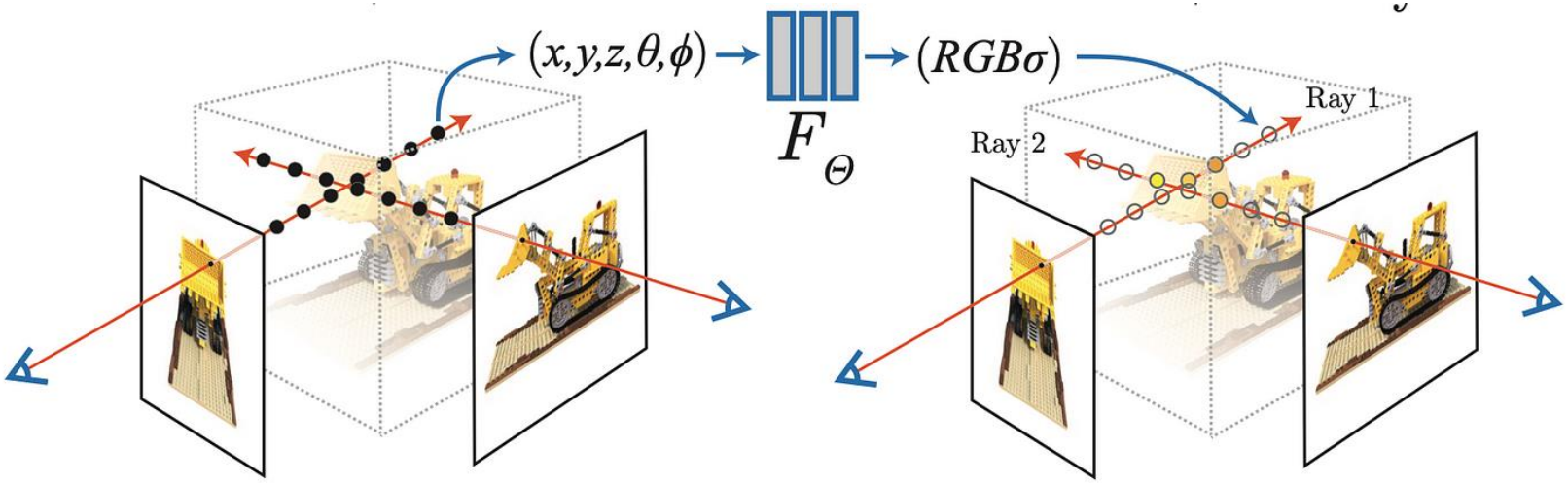



Fig. 1. Standard modules for SLAM pipeline implementations as outlined in the paper. The front-end is where the raw data is extracted from the sensors and abstracted into a model. The back-end infers from the front-end data and optimizes the estimates. The main modules in the front-end are tracking, loop detection, and relocalization. Tracking can be either direct or indirect. The back-end can be classified into filter and optimization-based techniques.

[Álvarez-Tuñón, O., Brodskiy, Y. and Kayacan, E., 2023. Monocular visual simultaneous localization and mapping: (r) evolution from geometry to deep learning-based pipelines. *IEEE Transactions on Artificial Intelligence*, *5*(5), pp.1990-2010.]

# Radiance Fields


- mathematical function that describes how light radiates from every point in a 3D scene
- used to model **view-dependent appearance**, meaning that the color of a point in space can change based on the viewing direction.



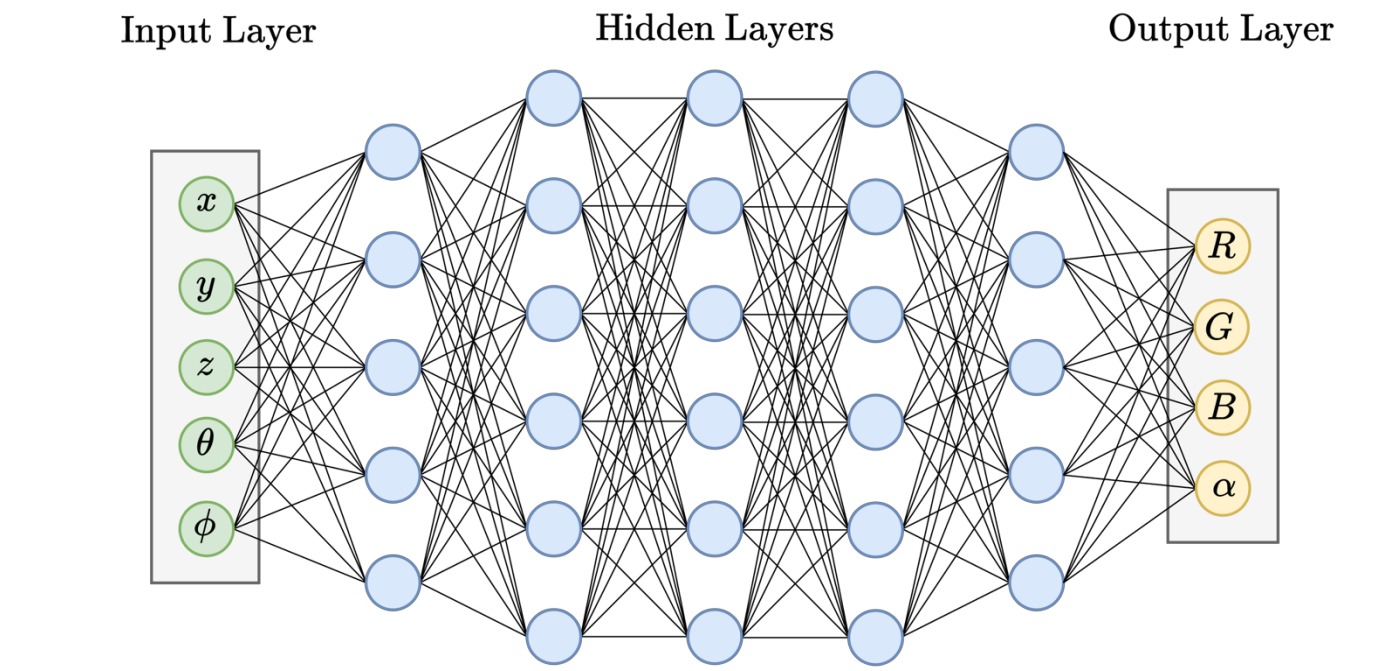

[https://blog.marvik.ai/wp-content/uploads/2023/05/nerf_3.png]

# Neural Radiance Fields (NeRF)

## Implicit 3D Scene Reconstructions

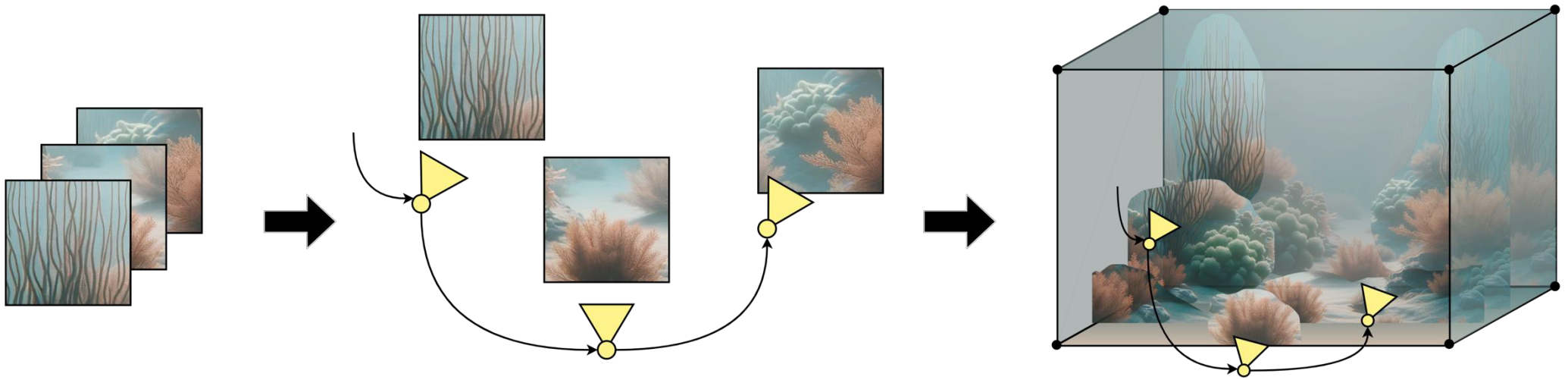


- Train an MLP to map camera pose to colour



# Neural Radiance Fields – Training

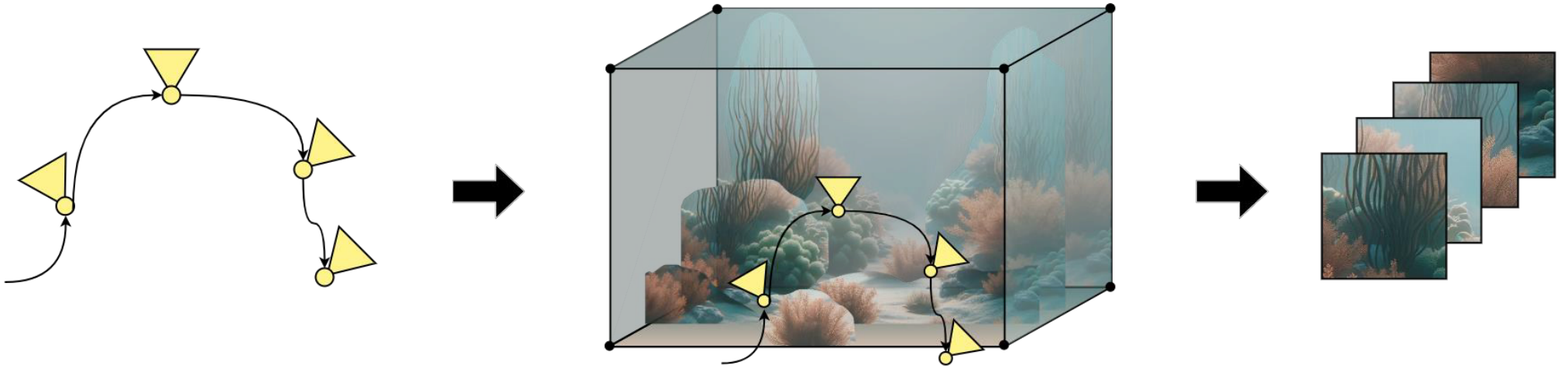
## Implicit 3D Scene Reconstructions



Video $v_{real}$ Estimated Path $(\mathbf{x}, \mathbf{d})$ Model $\theta(\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \alpha)$

# Neural Radiance Fields – Rendering

## Implicit 3D Scene Reconstructions

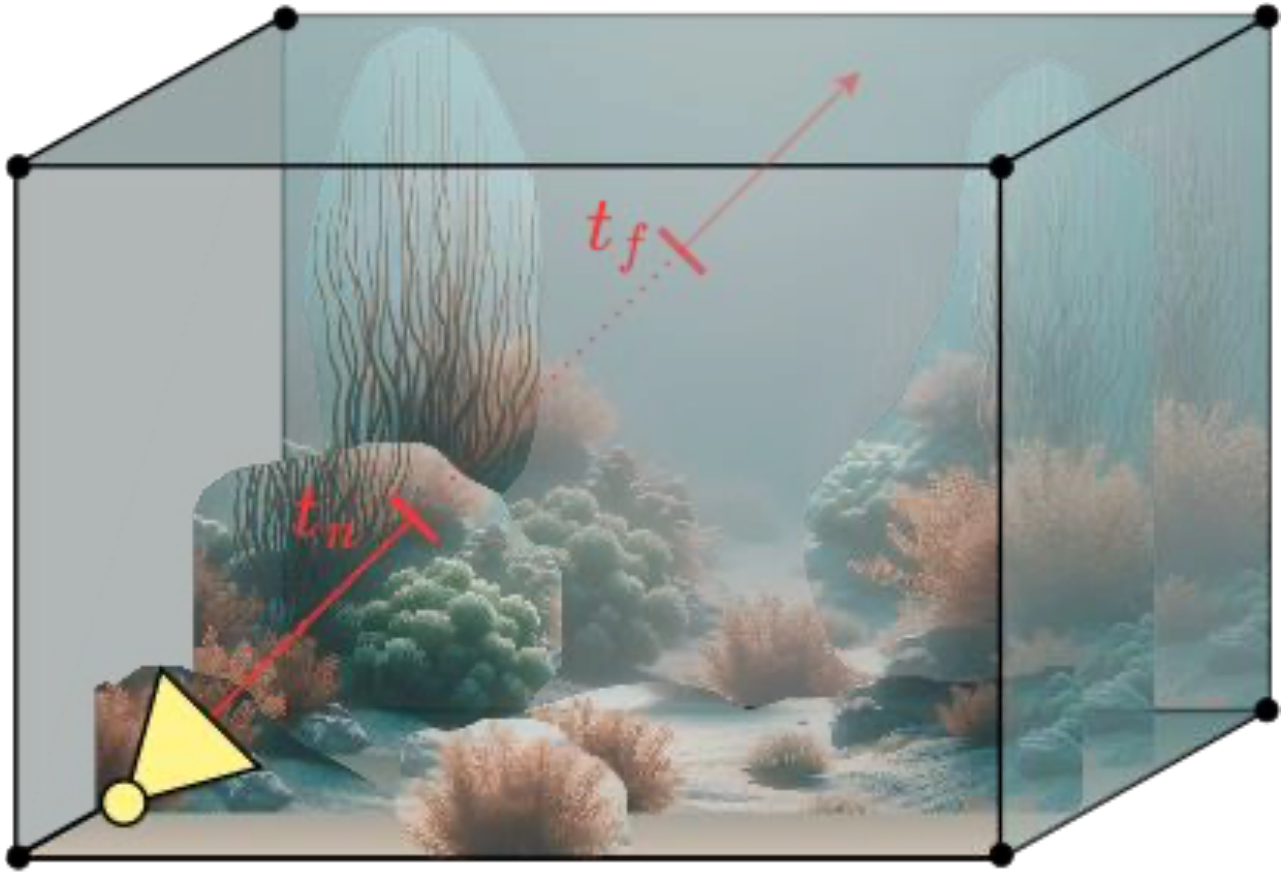


New Path $(\mathbf{x}, \mathbf{d})$

Model $\theta(\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \alpha)$

New video $v_{nerf}$

# Ray Marching

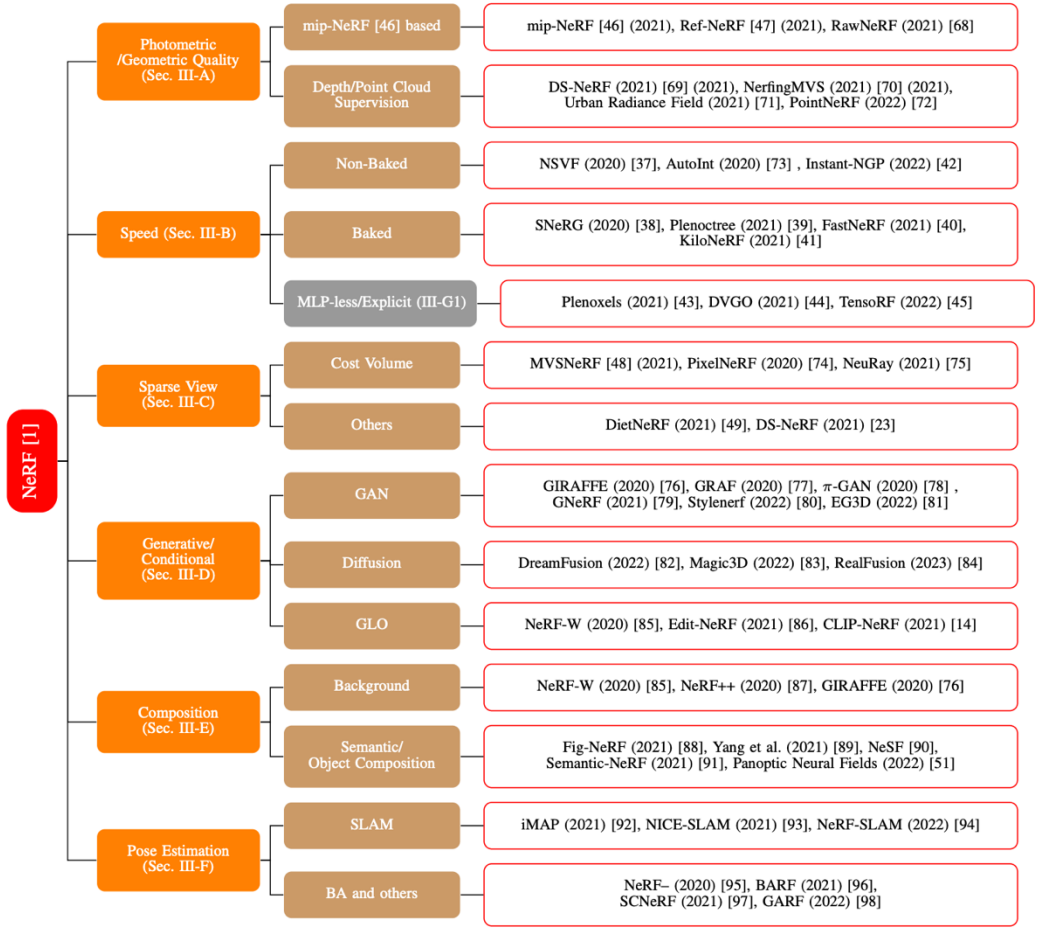



$$C(\mathbf{r}) = \int_{t_n}^{t_f} \mathrm{Tr}(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt.$$

$$\mathrm{Tr}(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

Machine Learning
Paper Reading Club

# Neural Radiance Fields SOTA



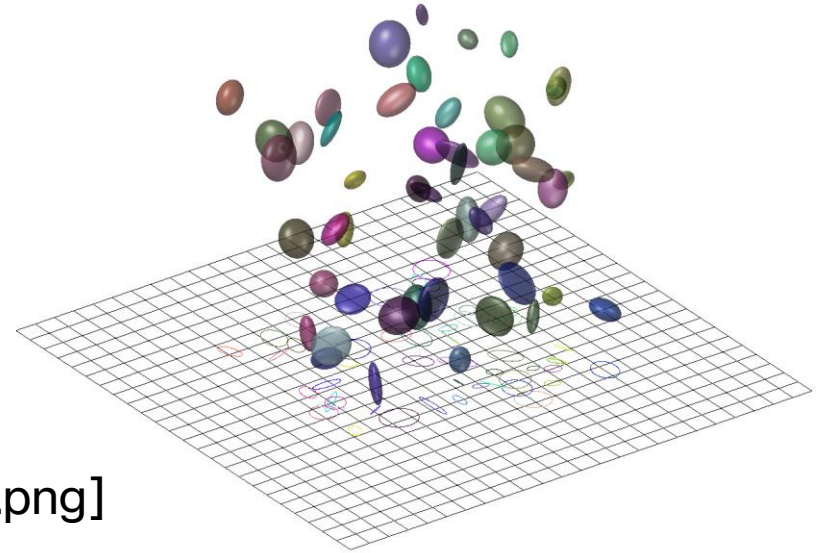

- solved:
  - small indoor scenes
  - static scenes
  - on-land environments

- challenges:
  - unbounded outdoor scenses
  - dynamic scenes
  - reflective surfaces
  - NeRF-based SLAM

# 3D Gaussian Splatting

## Explicit 3D Scene Reconstructions


- Instead of **MLP-based implicit representations** (like NeRF), Gaussian Splatting explicitly models 3D points as Gaussians.
- each Gaussian has 3D Position (X, Y, Z), Colour (R, G, B), opacity (α), covariance (Shape & Spread in 3D Space)



[https://zjwfufu.github.io/images/math_3dgs_3.png]

# 3D Gaussian Splatting - Training


1. **Initialize** a point cloud (often from COLMAP/SfM).
2. **Convert** points into **3D Gaussians** with position, color, opacity, and covariance.
3. **Render** by projecting Gaussians into a 2D image (differentiable rasterization).
4. **Optimize** Gaussian parameters to match training images.

alpha-blending:

$$\alpha A + (1 - \alpha) B$$

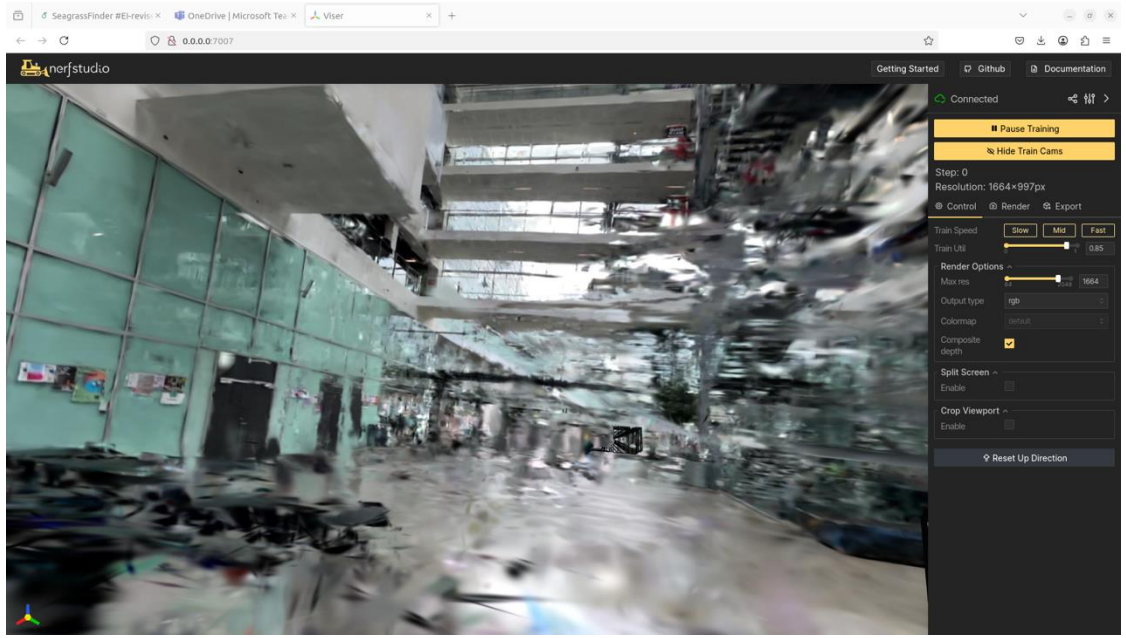
# 3DGS vs NeRF

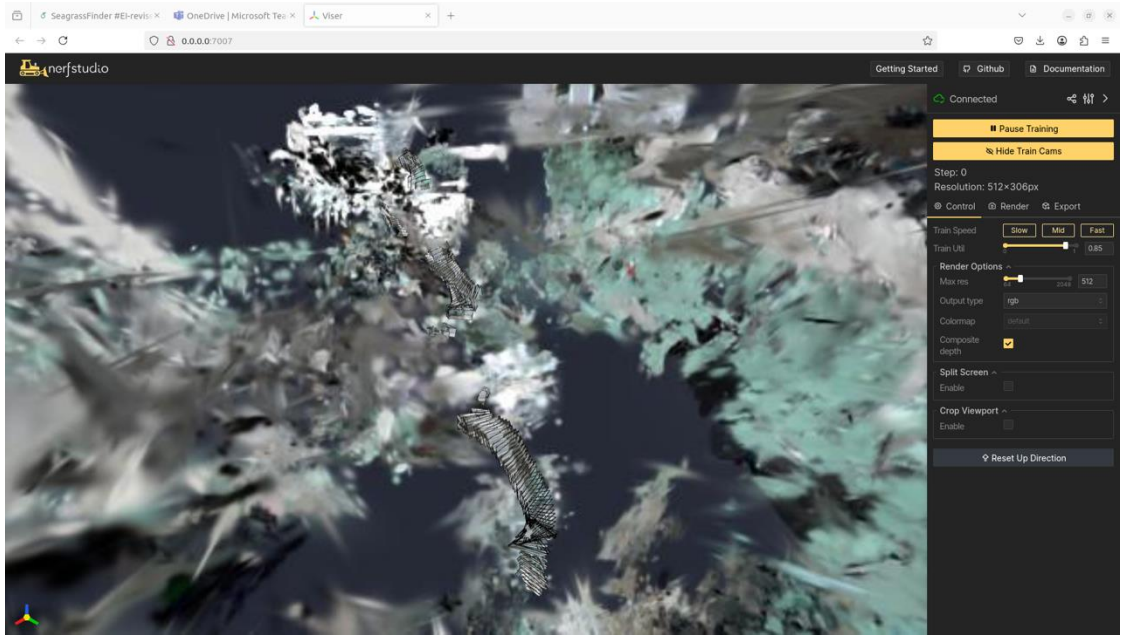

| Feature | Gaussian Splatting | Neural Radiance Fields |
|---|---|---|
| **Representation** | Explicit (3D Gaussians) | Implicit (MLP-based) |
| **Rendering** | Rasterisation & Alpha Compositing | Raymarching & Volume Rendering |
| **Training Speed** | Faster (~minutes to hours) | Slower |
| **Rendering Speed** | **Real-time** (~60 FPS) | Slow (seconds per frame) |
| **Memory Usage** | Higher (Stores all Gaussians) | Lower (Compact MLP) |

# Why is this cool?

Original Video

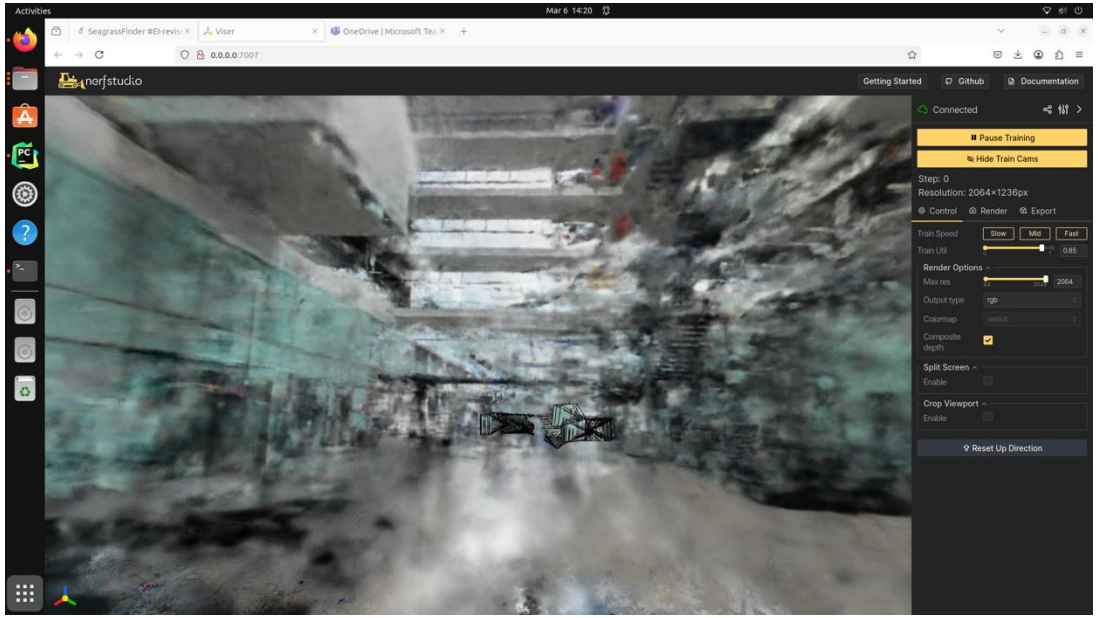
# Why is this cool?

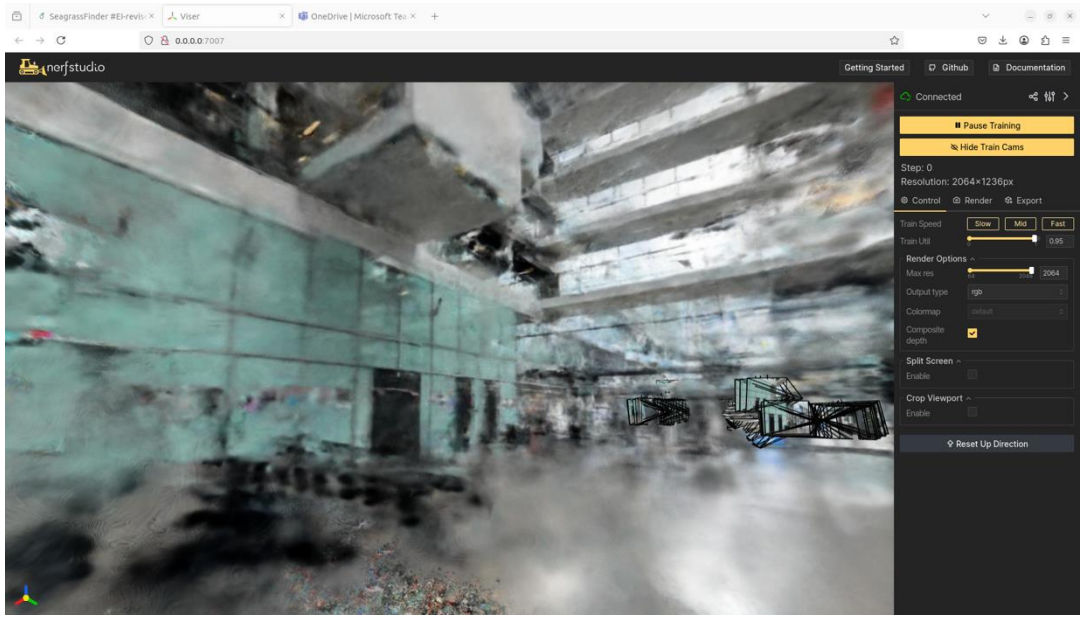



3DGS

# Why is this cool?




NeRF