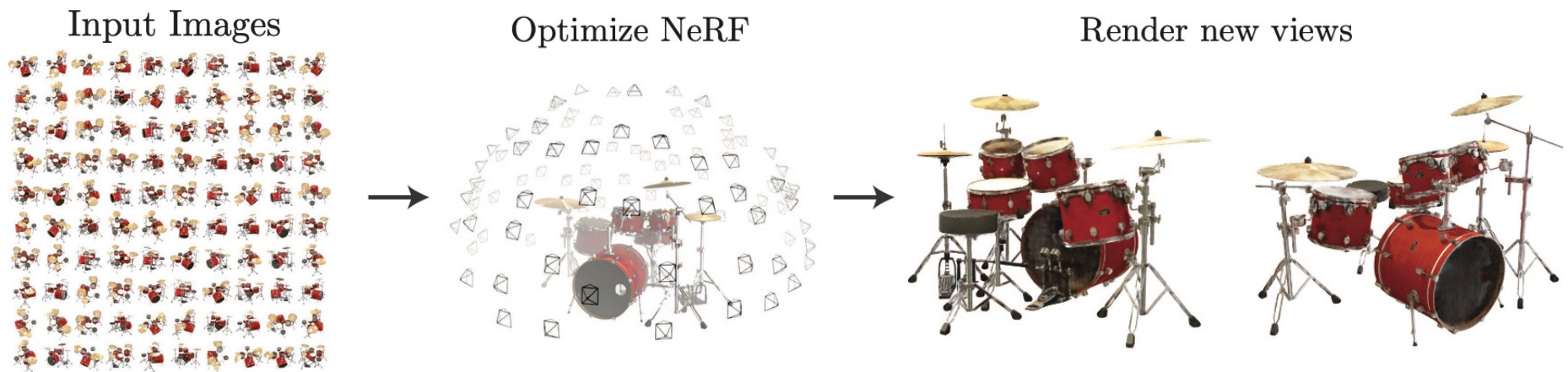


Neural Radiance Fields (NeRFs)



Many slides from Mildenhall et al. and Berkeley CS194

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

ECCV 2020 (best paper honorable mention)

Over 9700 Google Scholar citations as of 12/5/2024

Ben Mildenhall*



UC Berkeley



Pratul Srinivasan*



UC Berkeley



Matt Tancik*



UC Berkeley



Jon Barron



Google Research



Ravi Ramamoorthi



UC San Diego



Ren Ng



UC Berkeley

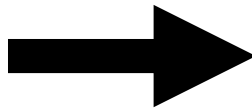


<https://www.matthewtancik.com/nerf>

Train a neural network to represent the plenoptic function

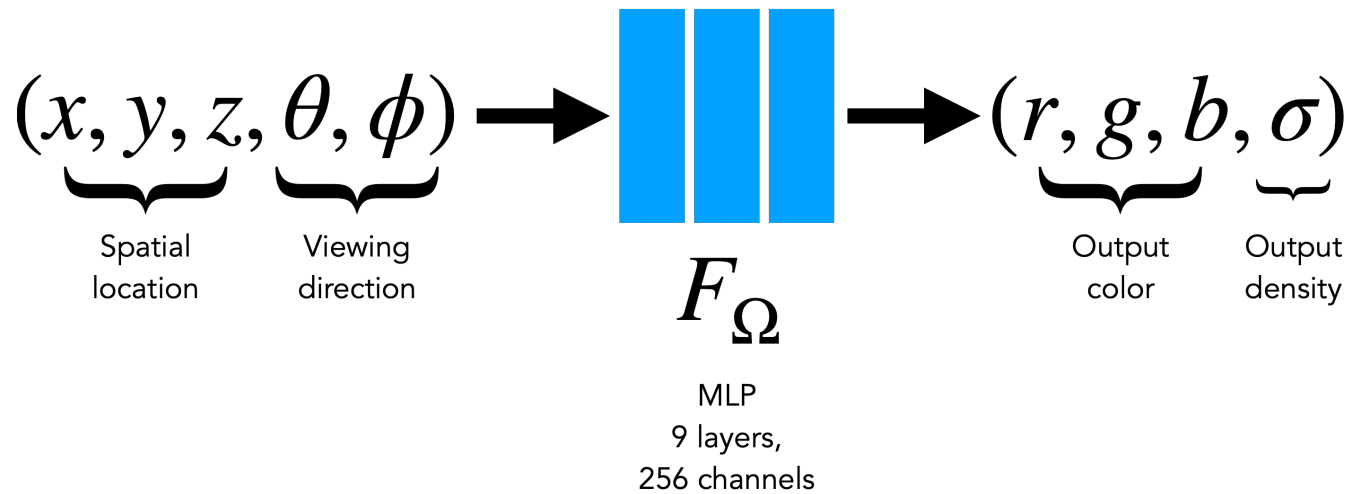


Inputs: sparsely sampled images of scene



Outputs: *new* views of same scene

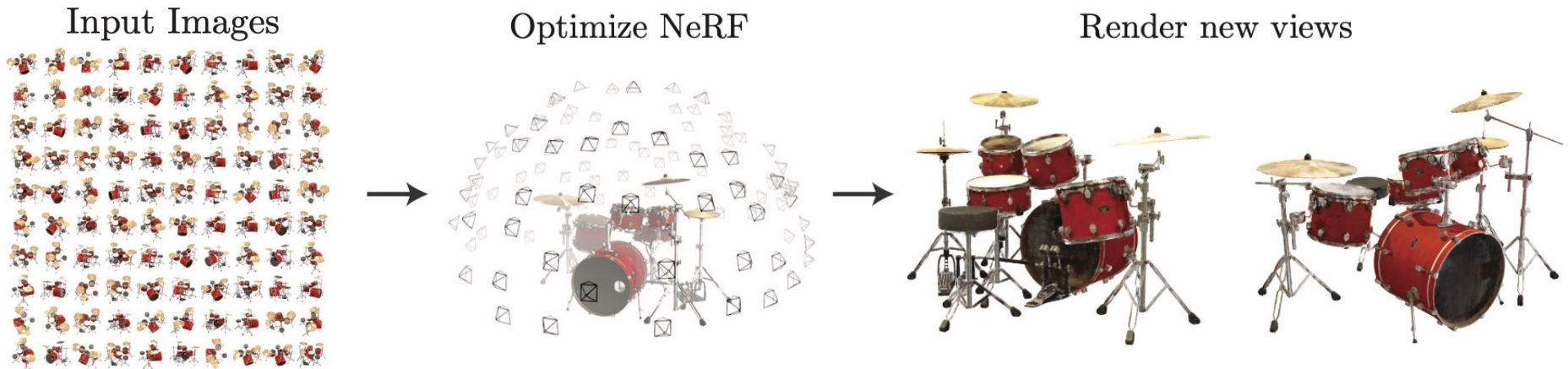
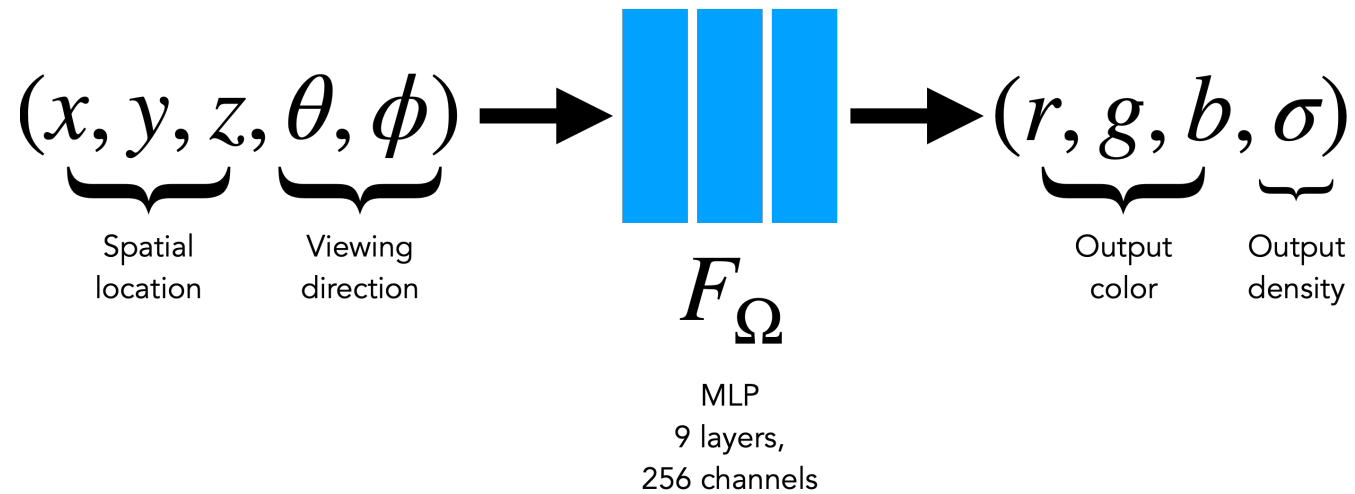
Train a neural network to represent the plenoptic function



Volumetric “fog” model: Every 5D input gets mapped to **Color** and **Density**

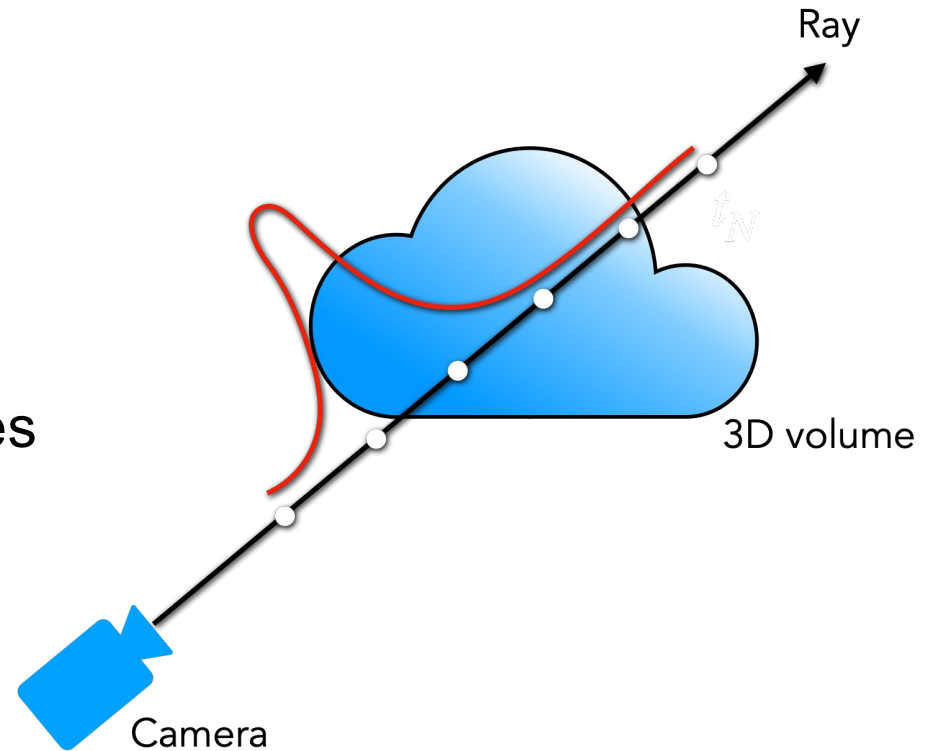


Train a neural network to represent the plenoptic function



NeRF rendering

- At every point (and direction) in space NeRF gives you the color and density: (c_i, σ_i)
- Need to integrate these values along a viewing ray to render a pixel
- Idea: sum how much light reaches each point * visibility * color



How to render a pixel: Volume rendering

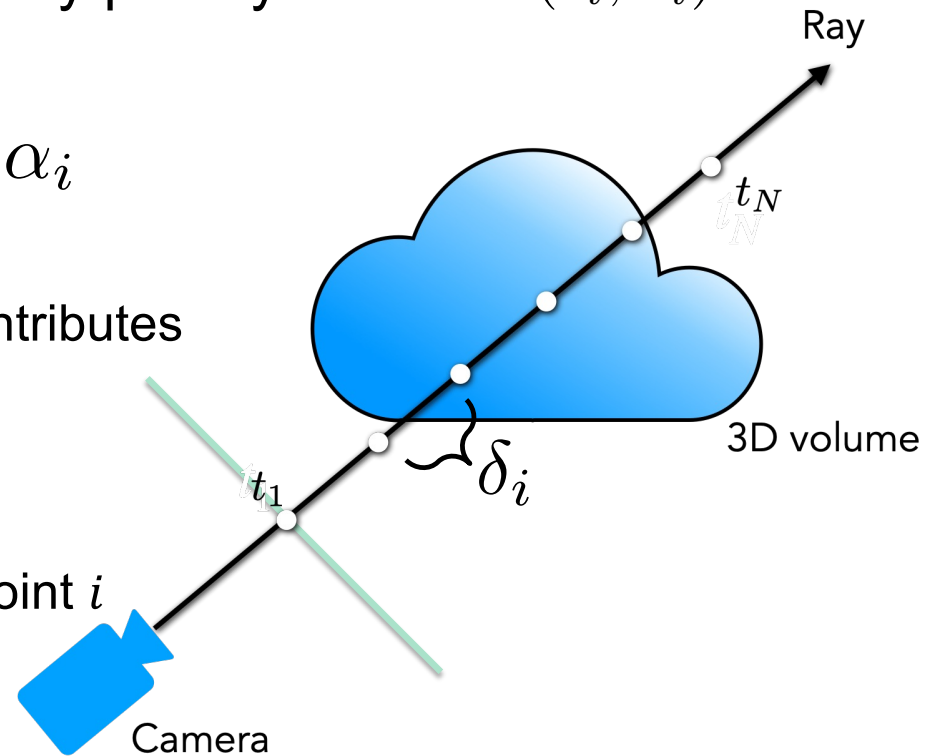
Given: a ray $\mathbf{r}(i) = \vec{o} + i\vec{d}$

At every point you know: (c_i, σ_i)

$$C(\mathbf{r}) \approx \sum_i^N w_i c_i \quad w_i = T_i \alpha_i$$

Alpha (α_i): How much light a ray segment contributes

Transmittance (T_i): how much light reaches point i



How to render a pixel: Volume rendering

Given: a ray $\mathbf{r}(i) = \vec{o} + i\vec{d}$

At every point you know: (c_i, σ_i)

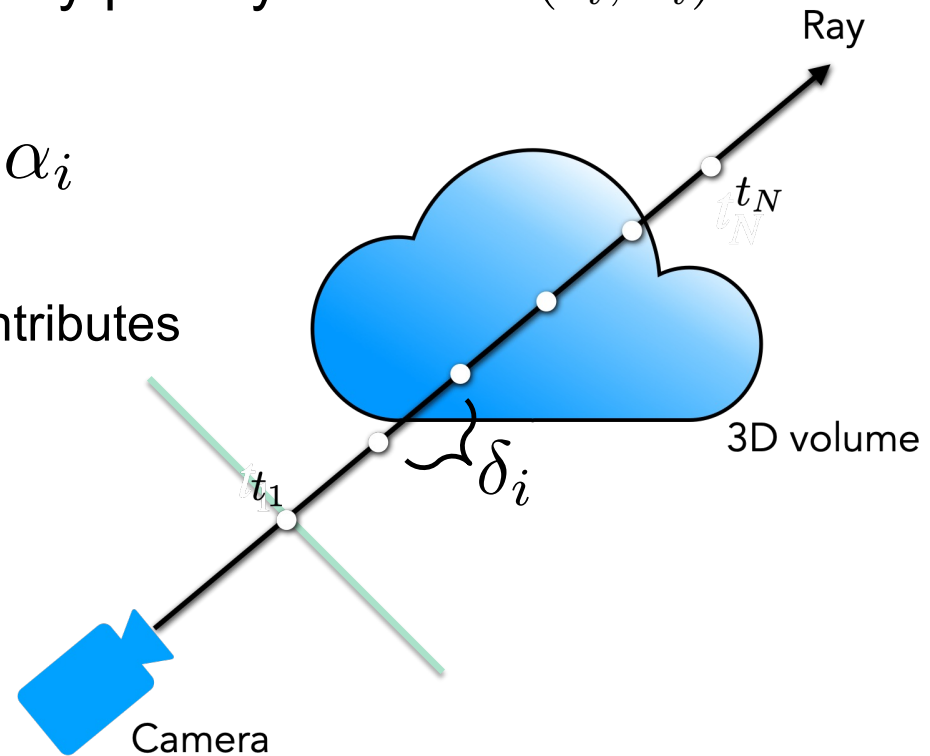
$$C(\mathbf{r}) \approx \sum_i^N w_i c_i \quad w_i = T_i \alpha_i$$

Alpha (α_i): How much light a ray segment contributes

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$

Alpha ranges from 0 (lowest density) to 1 (highest density)

- If $\sigma_i = 0$, then $\exp(-\sigma_i) = 1$ and $\alpha_i = 0$ (assuming $\delta_i = 1$)
- If $\sigma_i = \infty$, then $\exp(-\sigma_i) = 0$ and $\alpha_i = 1$



How to render a pixel: Volume rendering

Given: a ray $\mathbf{r}(i) = \vec{o} + i\vec{d}$

At every point you know: (c_i, σ_i)

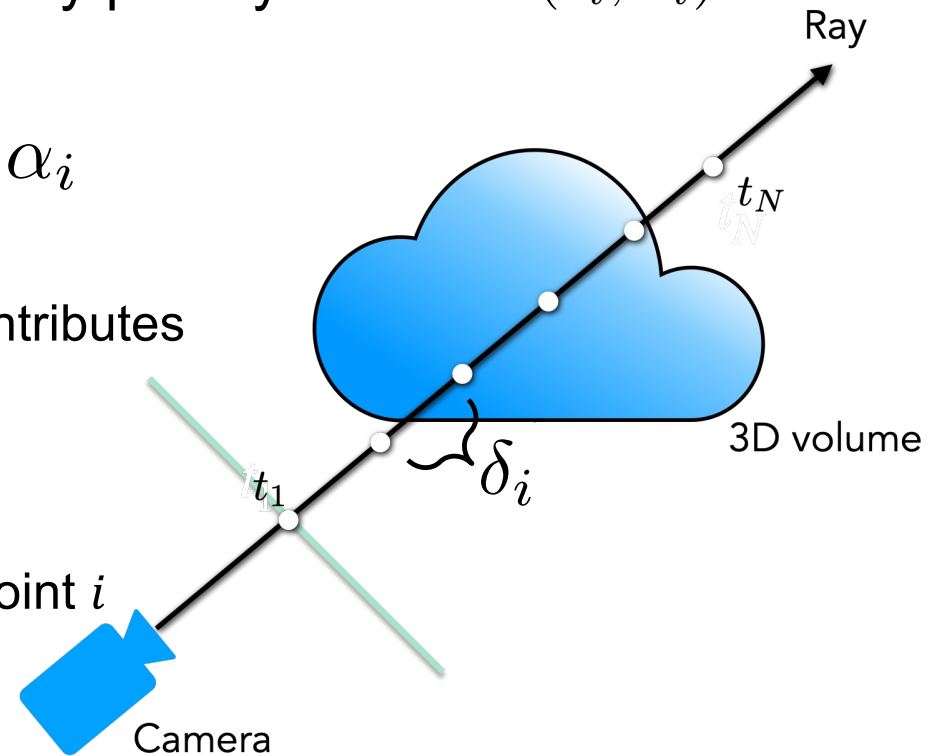
$$C(\mathbf{r}) \approx \sum_i^N w_i c_i \quad w_i = T_i \alpha_i$$

Alpha (α_i): How much light a ray segment contributes

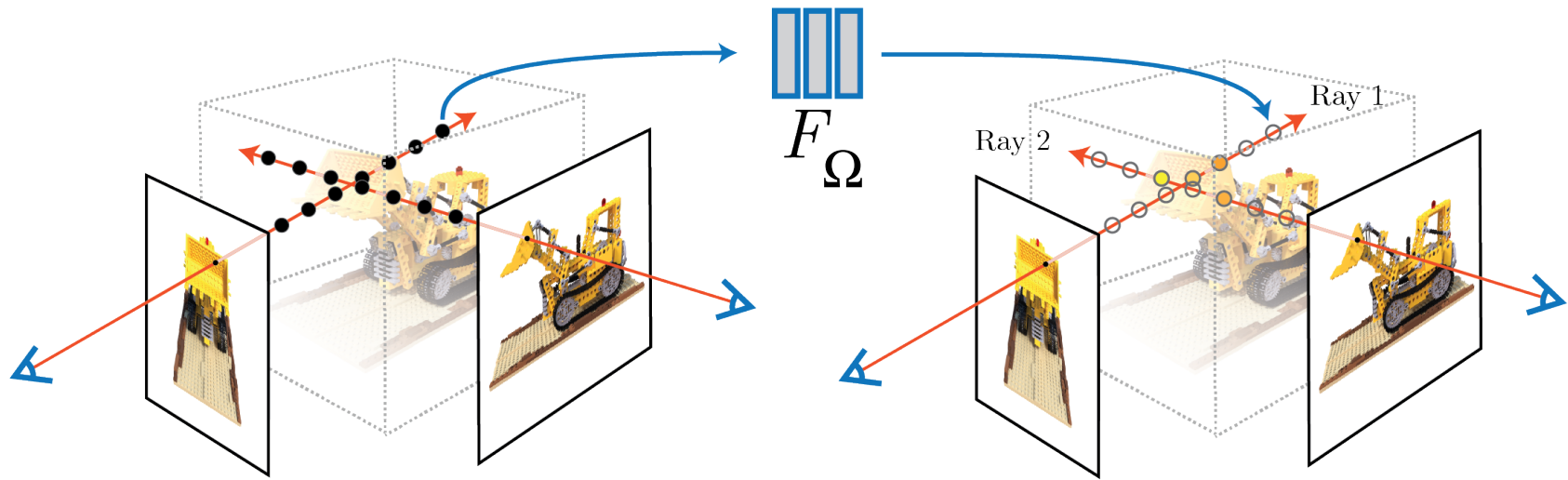
$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$

Transmittance (T_i): how much light reaches point i

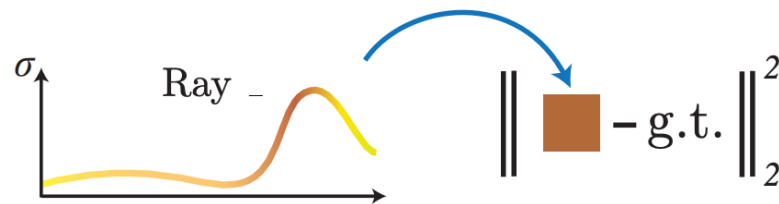
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$



Training: Optimization with reconstruction loss



$$\min_{\Omega} \sum_i \|\text{render}^{(i)}(F_{\Omega}) - I_{\text{gt}}^{(i)}\|^2$$



Example results





Viewpoint-dependent effects

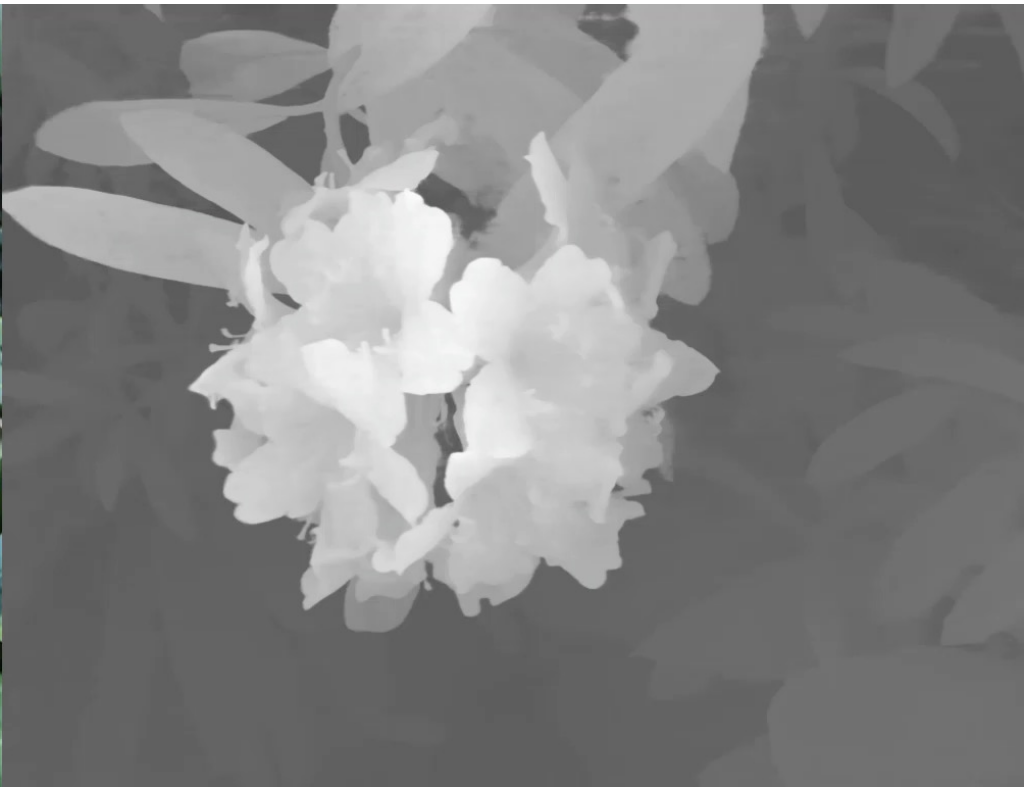


Viewpoint-dependent effects



Rendering depth

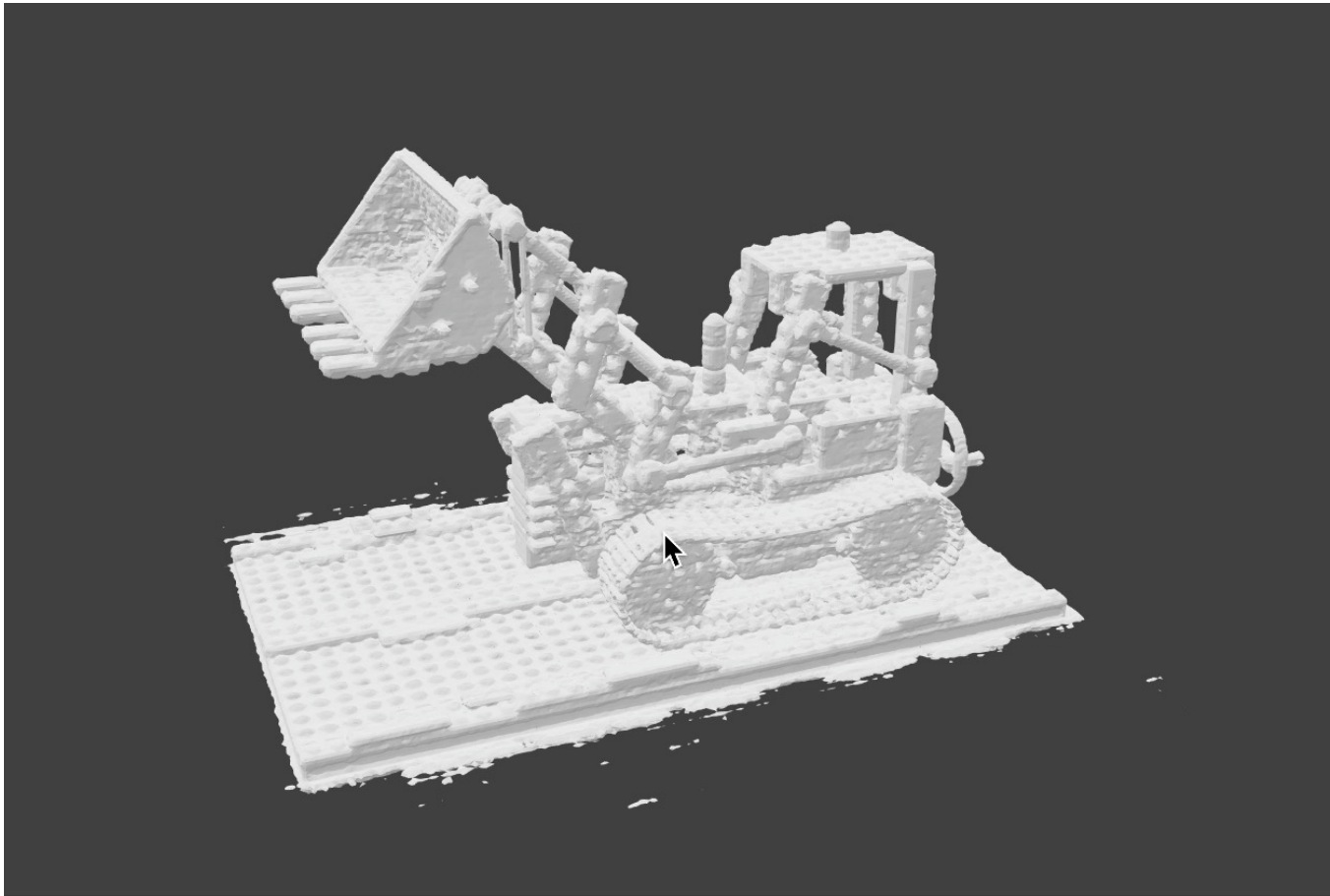
- Depth is given by expected ray termination



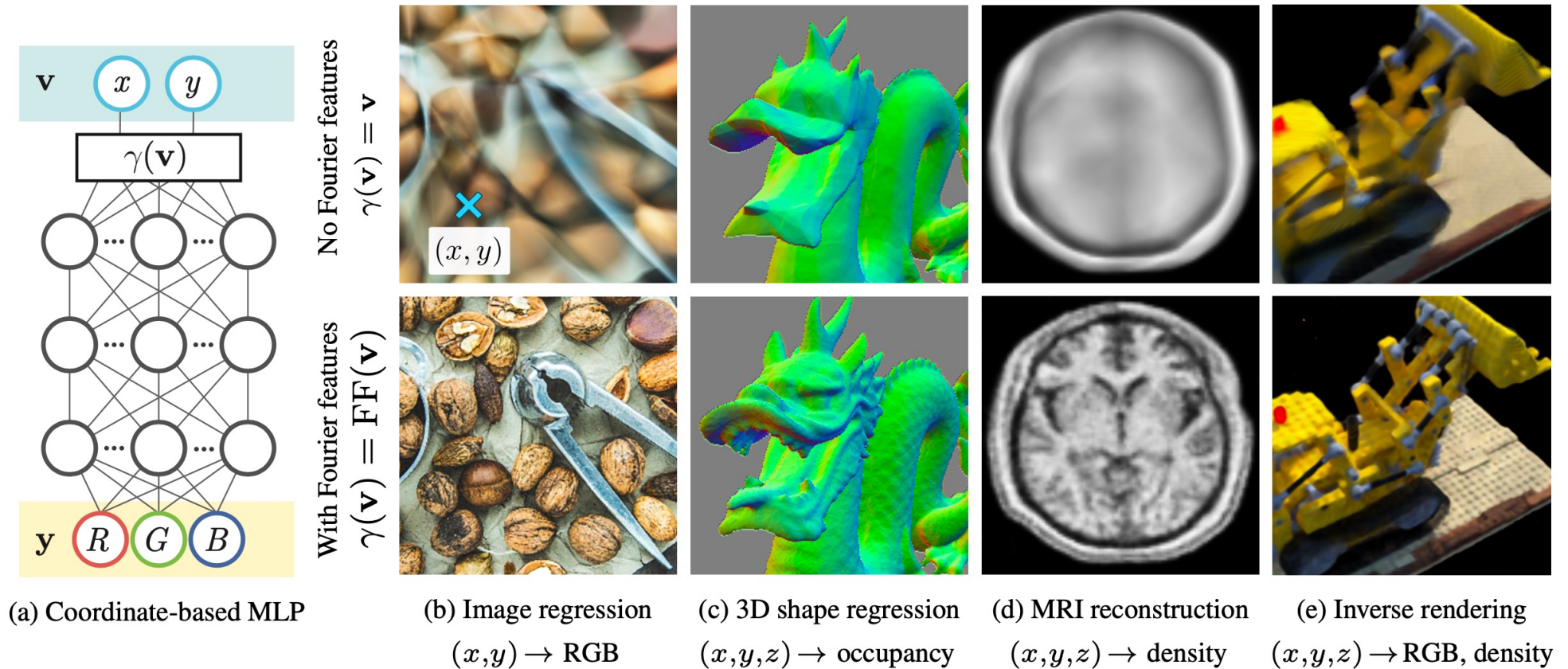
Because it models the entire plenoptic function you can insert objects with proper occlusion effects (in contrast to lightfields)



Extract surface on high density regions



Improving results quality with Fourier features



M. Tancik et al. [Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains](#). NeurIPS 2020

NeRF limitations

- Expensive / slow to train and render
- Sensitive to sampling strategy
- Does not generalize between scenes
- Sensitive to pose accuracy
- Assumes static scene
- Assumes static lighting and camera focus
- Not a mesh

NeRF explosion

Awesome Neural Radiance Fields

A curated list of awesome neural radiance fields papers, inspired by [awesome-computer-vision](#).

How to Pull Request?

If you are interested in adding papers, feel free to submit a pull request following the instruction [here](#).

Table of Contents

- [Survey](#)
- [Papers](#)
- [Talks](#)

Survey

- [Neural Volume Rendering: NeRF And Beyond](#), Dellaert and Yen-Chen, Arxiv 2020 | [blog](#) | [github](#)

Papers

- [NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis](#), Mildenhall et al., ECCV 2020 | [github](#) | [bibtex](#)
- [NeRF++: Analyzing and Improving Neural Radiance Fields](#), Zhang et al., Arxiv 2020 | [github](#) | [bibtex](#)
- [DeRF: Decomposed Radiance Fields](#), Rebain et al. Arxiv 2020 | [bibtex](#)
- [NeRD: Neural Reflectance Decomposition from Image Collections](#), Boss et al., Arxiv 2020 | [github](#) | [bibtex](#)
- [NeRF--: Neural Radiance Fields Without Known Camera Parameters](#), Wang et al., Arxiv 2021 | [github](#) | [bibtex](#)

Faster Inference

- [Neural Sparse Voxel Fields](#), Liu et al., NeurIPS 2020 | [github](#) | [bibtex](#)
- [Autoint: Automatic Integration for Fast Neural Volume Rendering](#), Lindell et al., Arxiv 2020 | [bibtex](#)

Unconstrained Images

- [NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections](#), Martin-Brualla et al., Arxiv 2020 | [bibtex](#)

Deformable

- [Deformable Neural Radiance Fields](#), Park et al., Arxiv 2020 | [github](#) | [bibtex](#)
- [D-NeRF: Neural Radiance Fields for Dynamic Scenes](#), Pumarola et al., Arxiv 2020 | [bibtex](#)
- [Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction](#), Gafni et al., Arxiv 2020 | [bibtex](#)
- [Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Deforming Scene from Monocular Video](#), Tretschk et al., Arxiv 2020 | [github](#) | [bibtex](#)

Video

- [Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes](#), Li et al., Arxiv 2020 | [bibtex](#)
- [Space-time Neural Irradiance Fields for Free-Viewpoint Video](#), Xian et al., Arxiv 2020 | [bibtex](#)
- [Neural Radiance Flow for 4D View Synthesis and Video Processing](#), Du et al., Arxiv 2020 | [bibtex](#)
- [Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans](#), Peng et al., Arxiv 2020 | [bibtex](#)

Generalization

- [GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis](#), Schwarz et al., NeurIPS 2020 | [github](#) | [bibtex](#)
- [GRF: Learning a General Radiance Field for 3D Scene Representation and Rendering](#), Trevithick and

<https://github.com/yenchenlin/awesome-NeRF>

NeRF in the wild



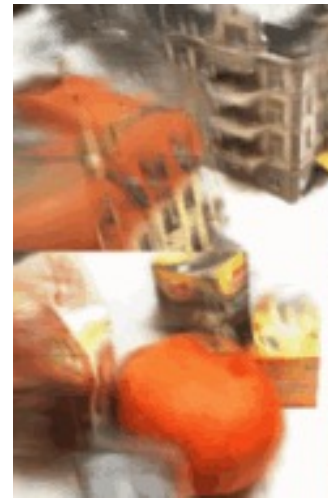
R. Martin-Brualla et al. [NeRF in the Wild](#). CVPR 2021

NeRF from one or few images

Three input views



PixelNeRF

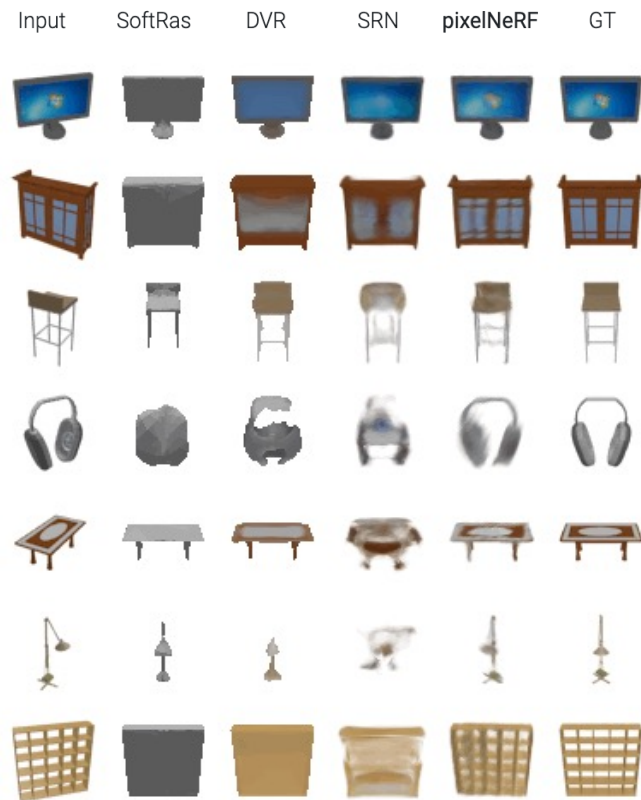


3-view NeRF



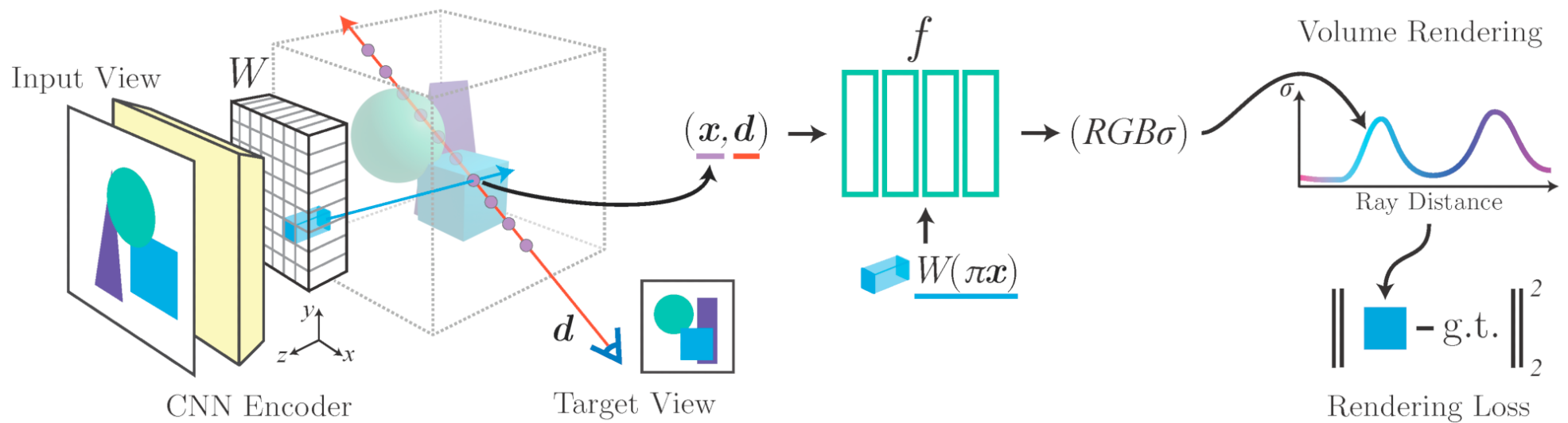
A. Yu et al. [PixelNeRF: Neural Radiance Fields from One or Few Images](#). CVPR 2021

NeRF from one or few images



A. Yu et al. [PixelNeRF: Neural Radiance Fields from One or Few Images](#). CVPR 2021

NeRF from one or few images



A. Yu et al. [PixelNeRF: Neural Radiance Fields from One or Few Images](#). CVPR 2021

NeRF without calibration

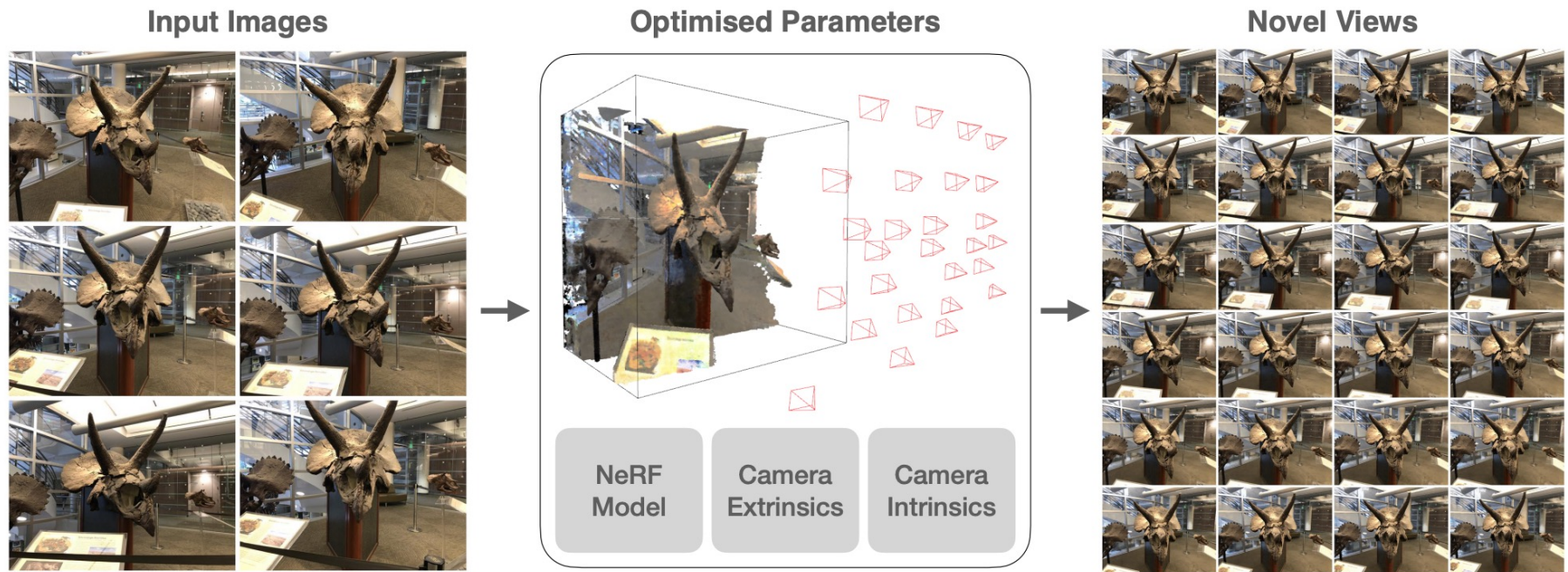


Figure 1: We propose *NeRF--*, a NeRF-based framework for novel view synthesis without pose supervision. Specifically, our method jointly optimises camera poses and intrinsics of a set of forward-facing input images while training a NeRF model.

NeRF without calibration

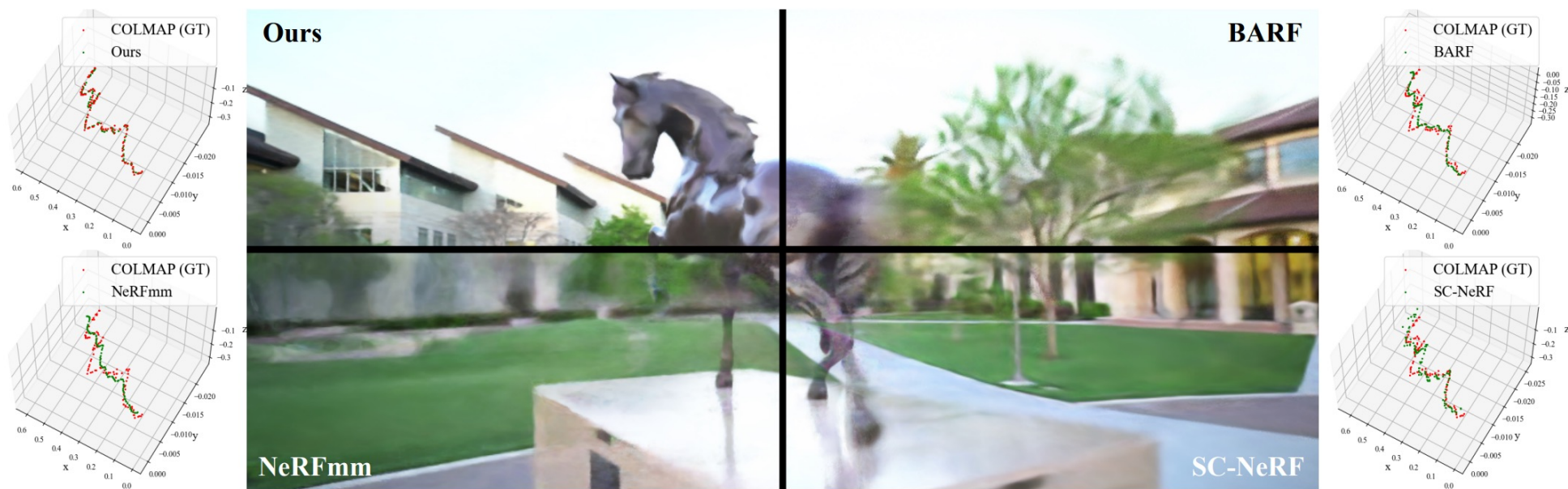


Figure 1. **Novel view synthesis comparison.** We propose NoPe-NeRF for joint pose estimation and novel view synthesis. Our method enables more robust pose estimation and renders better novel view synthesis than previous state-of-the-art methods.

W. Bian et al. [NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior](#). CVPR 2023

NeRF without calibration

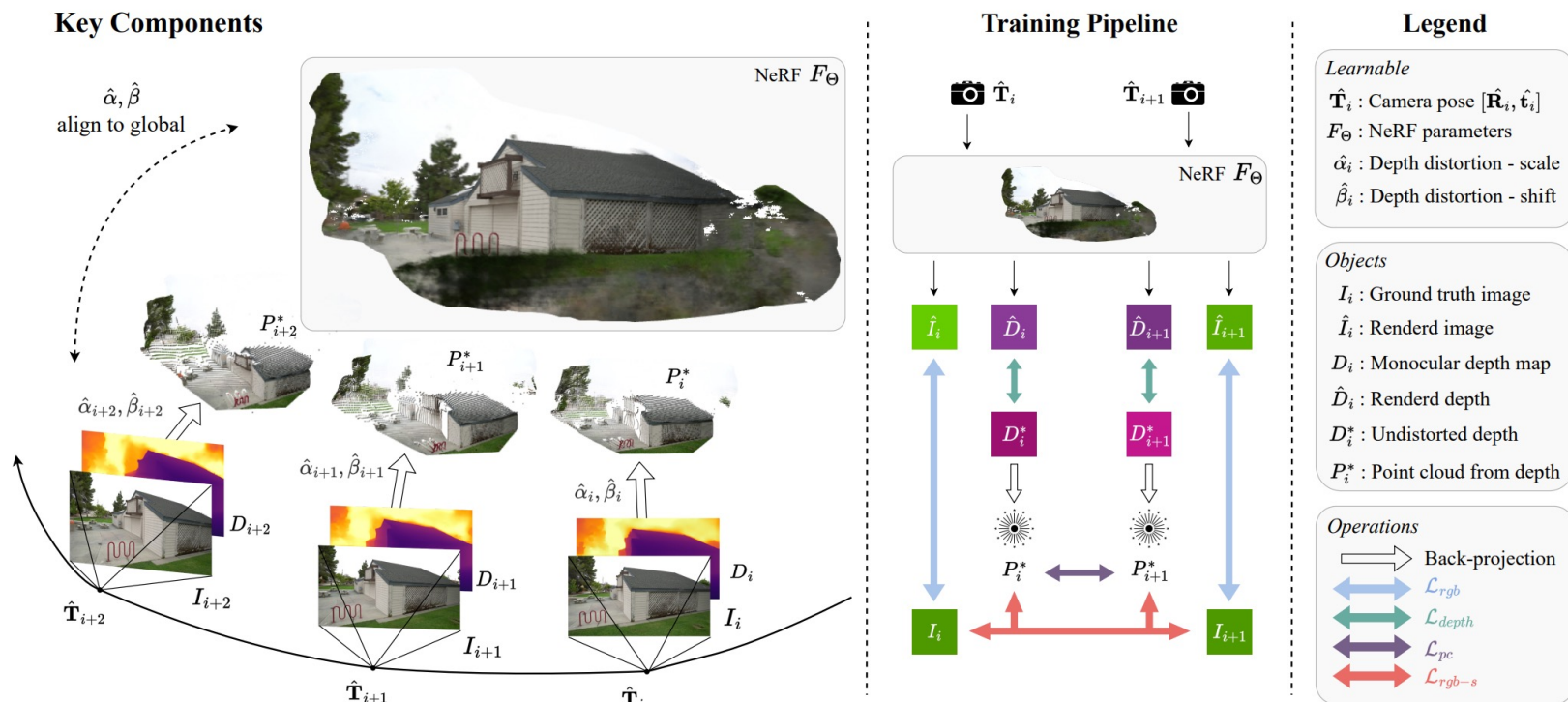
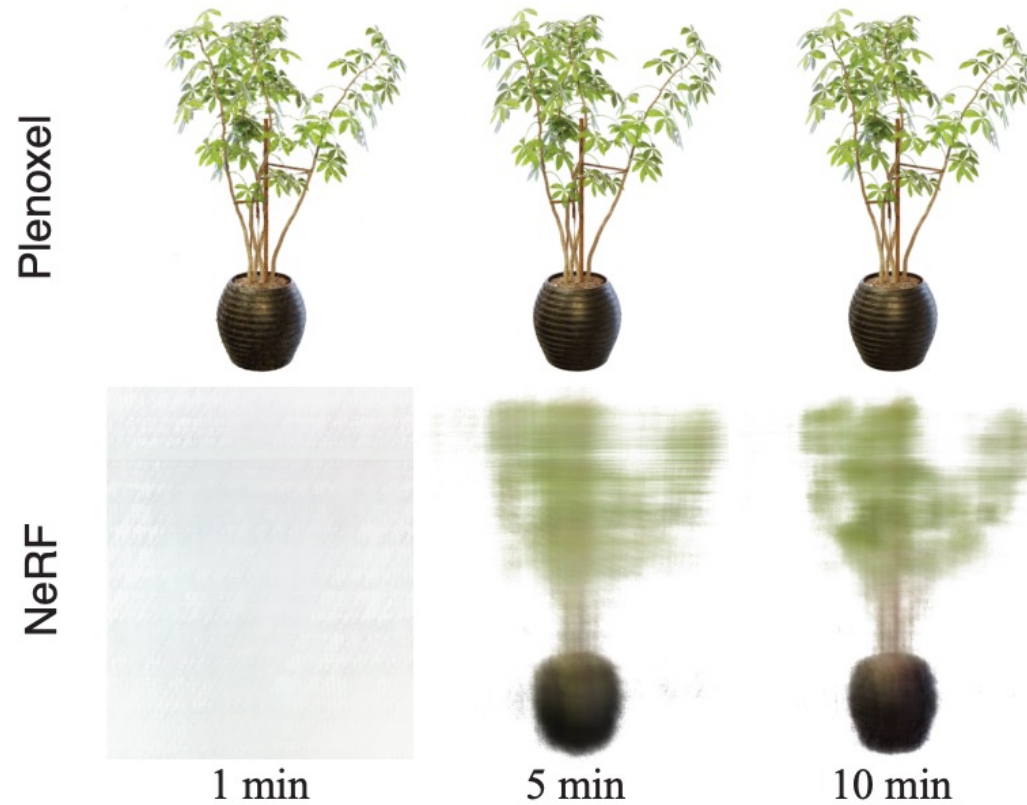


Figure 2. **Method Overview.** Our method takes a sequence of images as input to reconstruct NeRF and jointly estimates the camera poses of the frames. We first generate monocular depth maps from a mono-depth estimation network and reconstruct the point clouds. We then optimise NeRF, camera poses, and depth distortion parameters jointly with inter-frame and NeRF losses.

W. Bian et al. [NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior](#). CVPR 2023

Fast inference: Plenoxels



S. Fridovich-Keil et al. [Plenoxels: Radiance Fields without Neural Networks](#). CVPR 2022

Fast inference: Plenoxels

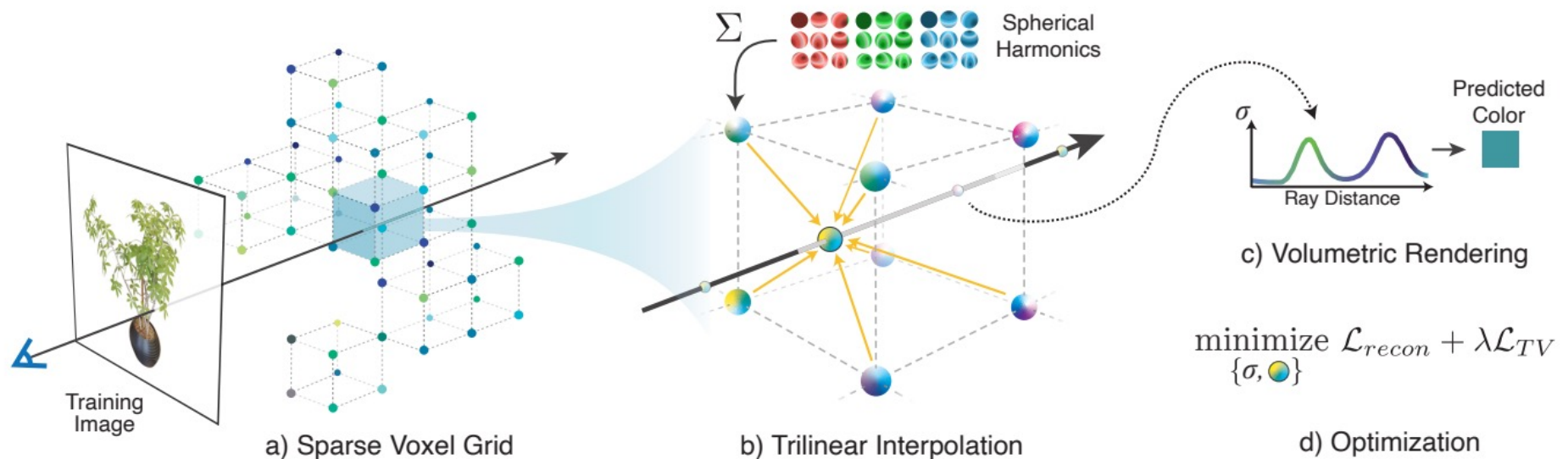
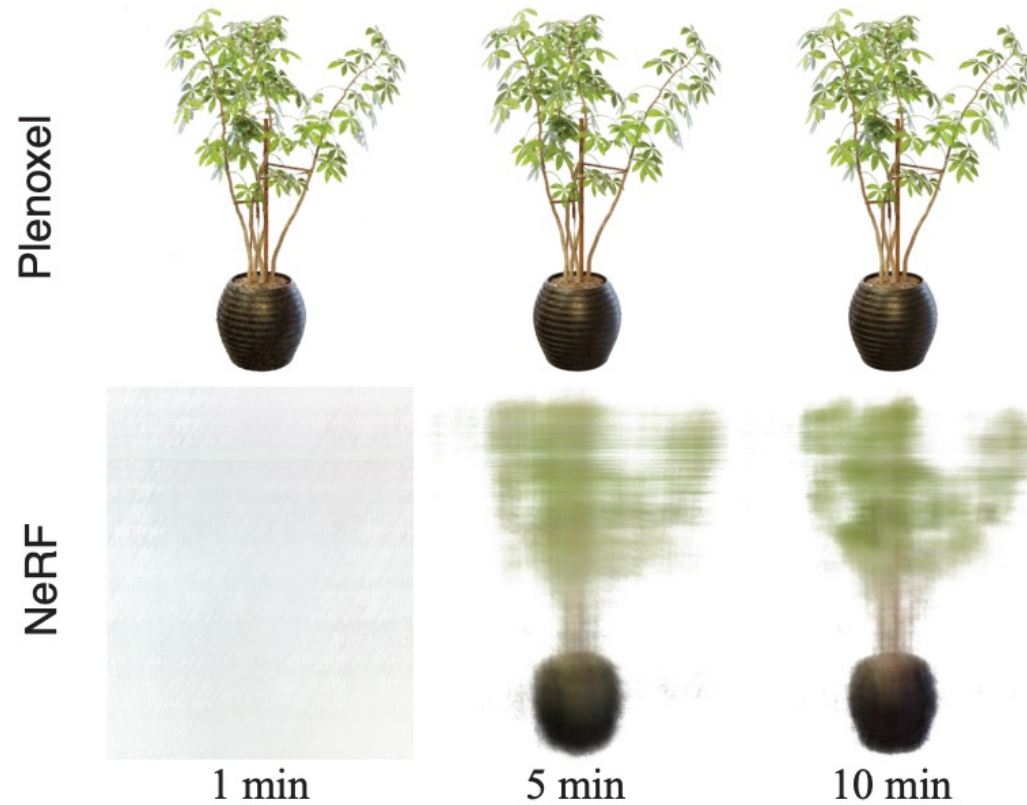


Figure 2. **Overview of our sparse Plenoxel model.** Given a set of images of an object or scene, we reconstruct a (a) sparse voxel (“Plenoxel”) grid with density and spherical harmonic coefficients at each voxel. To render a ray, we (b) compute the color and opacity of each sample point via trilinear interpolation of the neighboring voxel coefficients. We integrate the color and opacity of these samples using (c) differentiable volume rendering, following the recent success of NeRF [26]. The voxel coefficients can then be (d) optimized using the standard MSE reconstruction loss relative to the training images, along with a total variation regularizer.

S. Fridovich-Keil et al. [Plenoxels: Radiance Fields without Neural Networks](#). CVPR 2022

Fast inference: Plenoxels



S. Fridovich-Keil et al. [Plenoxels: Radiance Fields without Neural Networks](#). CVPR 2022

Fast inference: Plenoxels

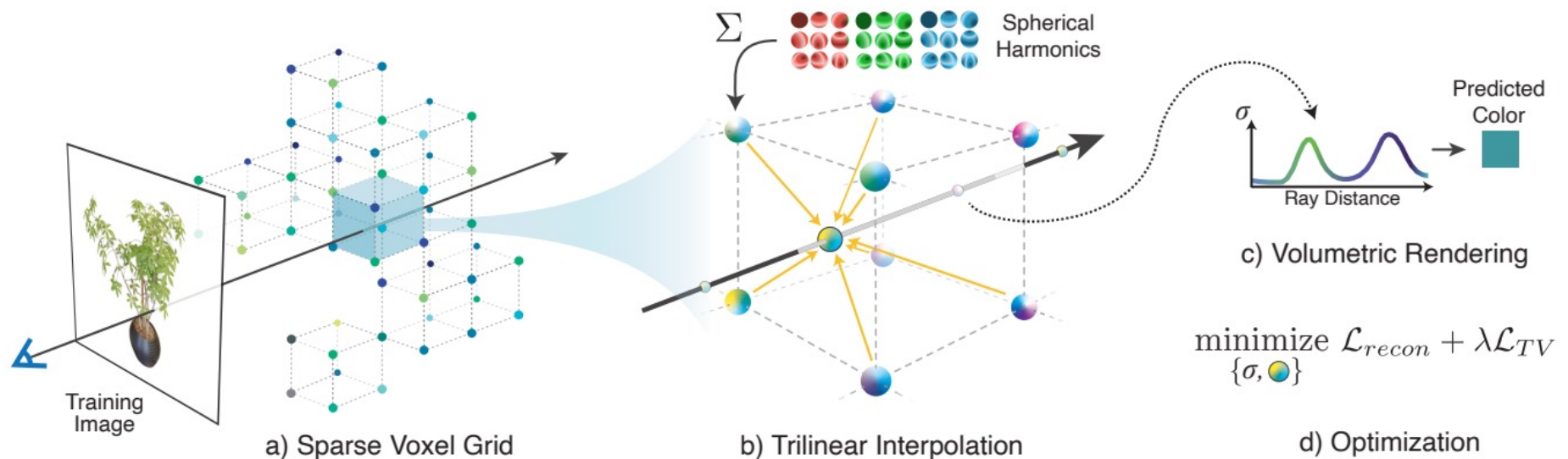


Figure 2. **Overview of our sparse Plenoxel model.** Given a set of images of an object or scene, we reconstruct a (a) sparse voxel (“Plenoxel”) grid with density and spherical harmonic coefficients at each voxel. To render a ray, we (b) compute the color and opacity of each sample point via trilinear interpolation of the neighboring voxel coefficients. We integrate the color and opacity of these samples using (c) differentiable volume rendering, following the recent success of NeRF [26]. The voxel coefficients can then be (d) optimized using the standard MSE reconstruction loss relative to the training images, along with a total variation regularizer.

S. Fridovich-Keil et al. [Plenoxels: Radiance Fields without Neural Networks](#). CVPR 2022

3D Gaussian splatting



[Figure source](#)

B. Kerbl et al. [3D Gaussian Splatting for Real-Time Radiance Field Rendering](#). SIGGRAPH 2023

3D Gaussian splatting

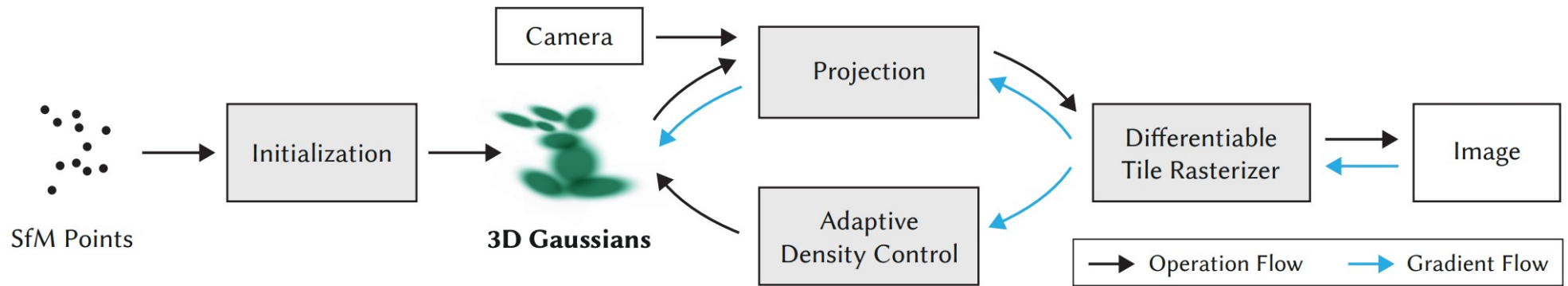


Fig. 2. Optimization starts with the sparse SfM point cloud and creates a set of 3D Gaussians. We then optimize and adaptively control the density of this set of Gaussians. During optimization we use our fast tile-based renderer, allowing competitive training times compared to SOTA fast radiance field methods. Once trained, our renderer allows real-time navigation for a wide variety of scenes.

3D Gaussian splatting

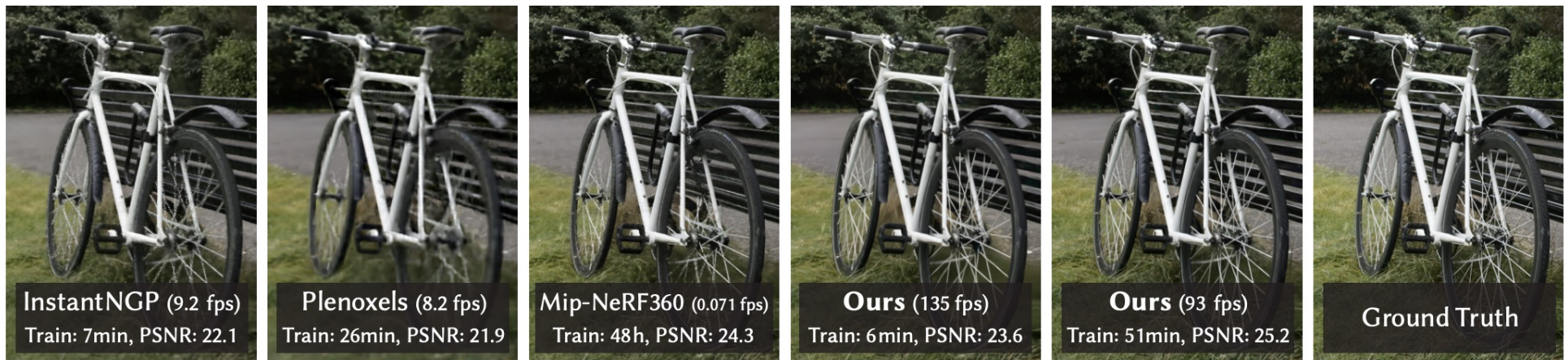


Fig. 1. Our method achieves real-time rendering of radiance fields with quality that equals the previous method with the best quality [Barron et al. 2022], while only requiring optimization times competitive with the fastest previous methods [Fridovich-Keil and Yu et al. 2022; Müller et al. 2022]. Key to this performance is a novel 3D Gaussian scene representation coupled with a real-time differentiable renderer, which offers significant speedup to both scene optimization and novel view synthesis. Note that for comparable training times to InstantNGP [Müller et al. 2022], we achieve similar quality to theirs; while this is the maximum quality they reach, by training for 51min we achieve state-of-the-art quality, even slightly better than Mip-NeRF360 [Barron et al. 2022].

B. Kerbl et al. [3D Gaussian Splatting for Real-Time Radiance Field Rendering](#). SIGGRAPH 2023

3D Gaussian splatting without calibration

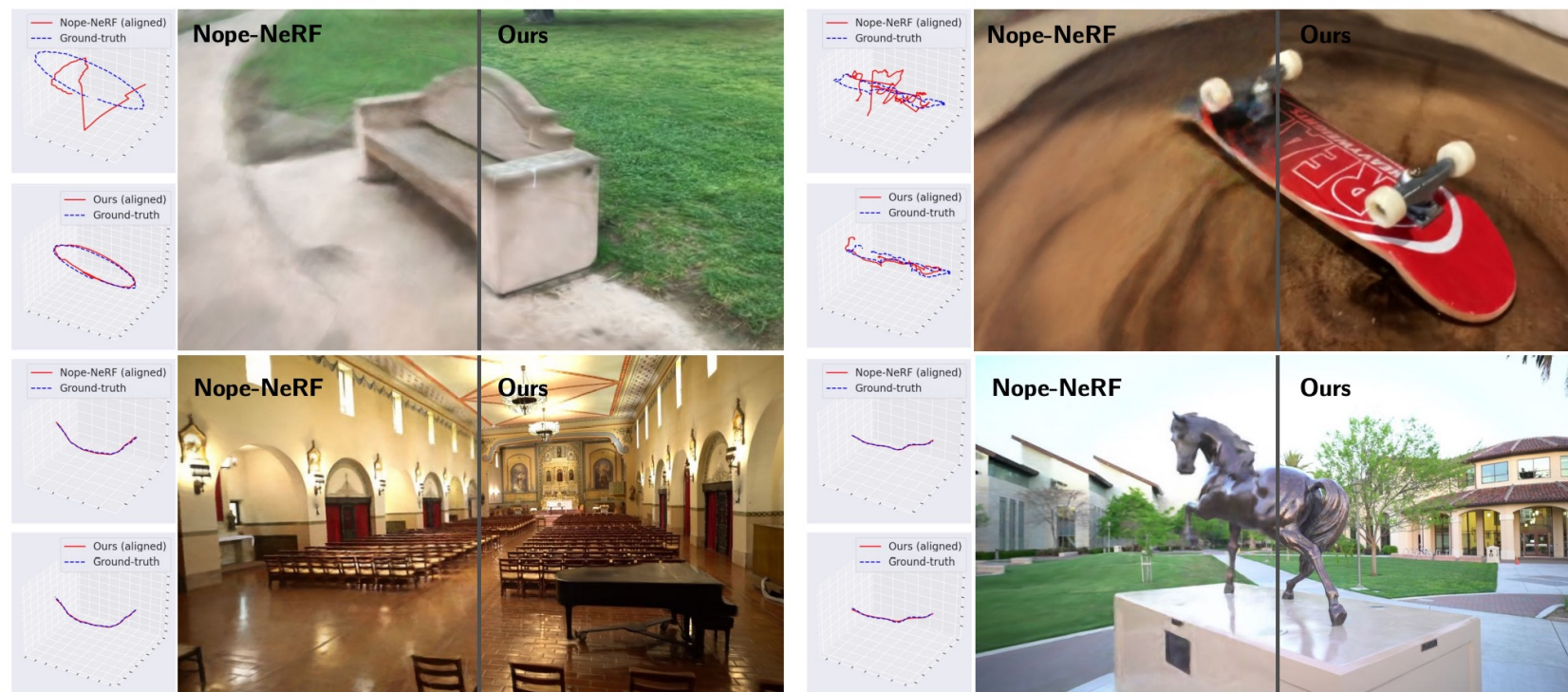


Figure 1. **Novel View Synthesis and Camera Pose Estimation Comparison.** We propose COLMAP-Free 3D Gaussian Splatting (CF-3DGS) for novel view synthesis without known camera parameters. Our method achieves more robustness in pose estimation and better quality in novel view synthesis than previous state-of-the-art methods.

Y. Fu et al. [COLMAP-Free 3D Gaussian Splatting](#). CVPR 2024

3D Gaussian splatting without calibration

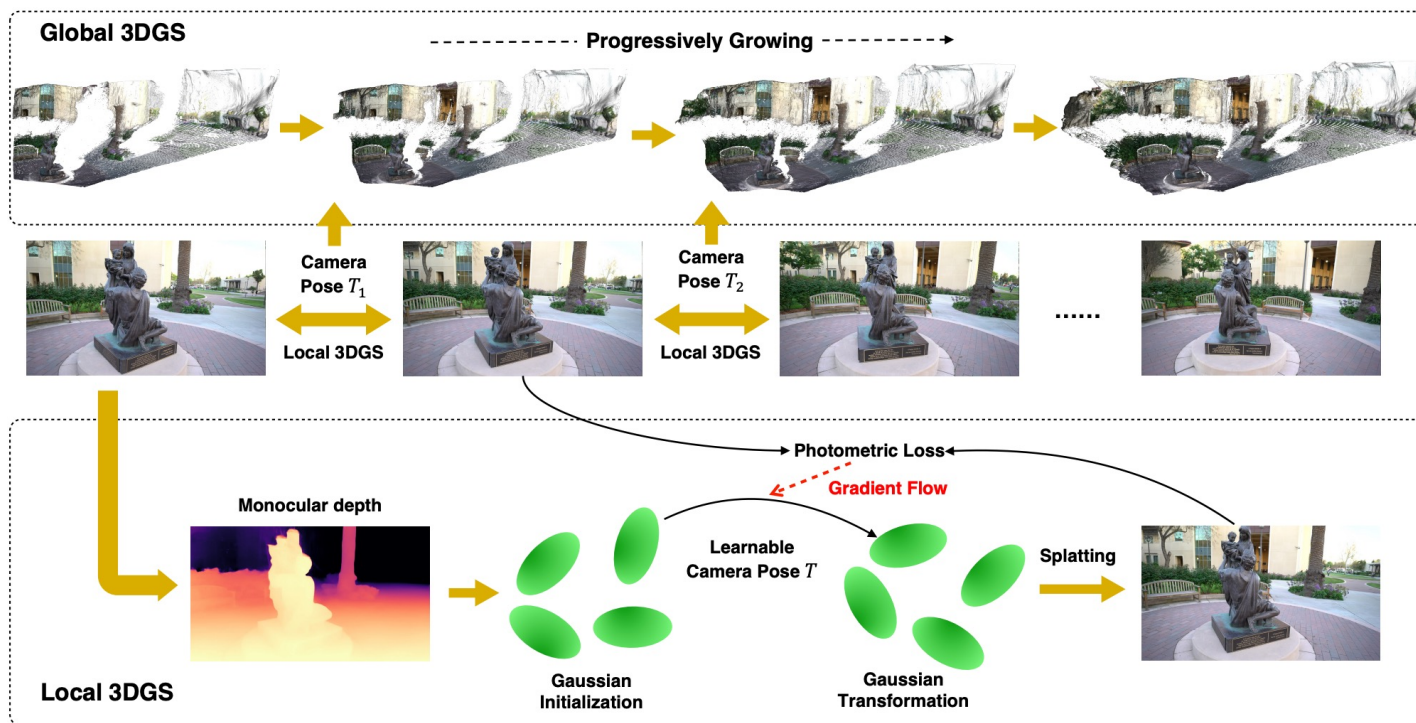
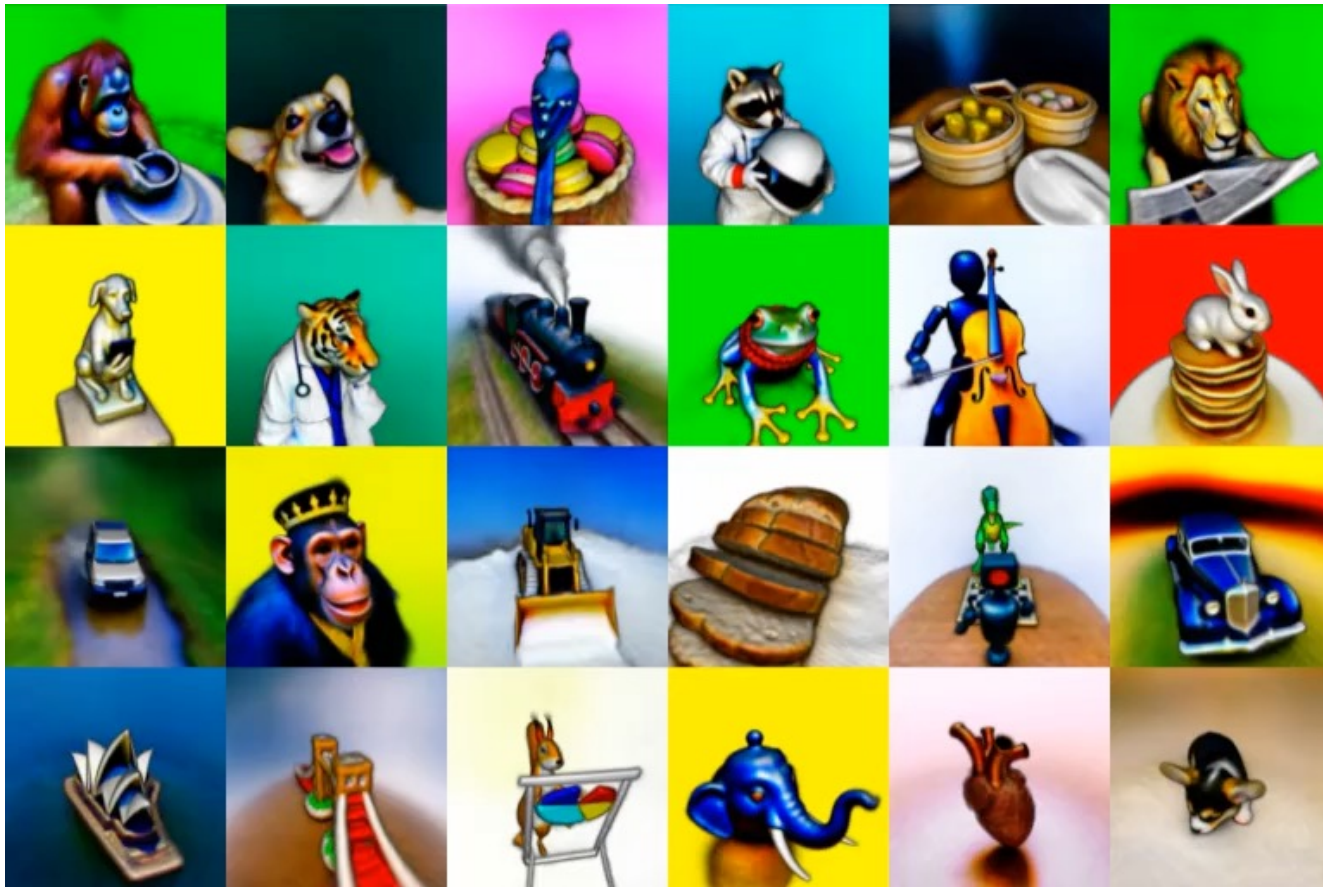


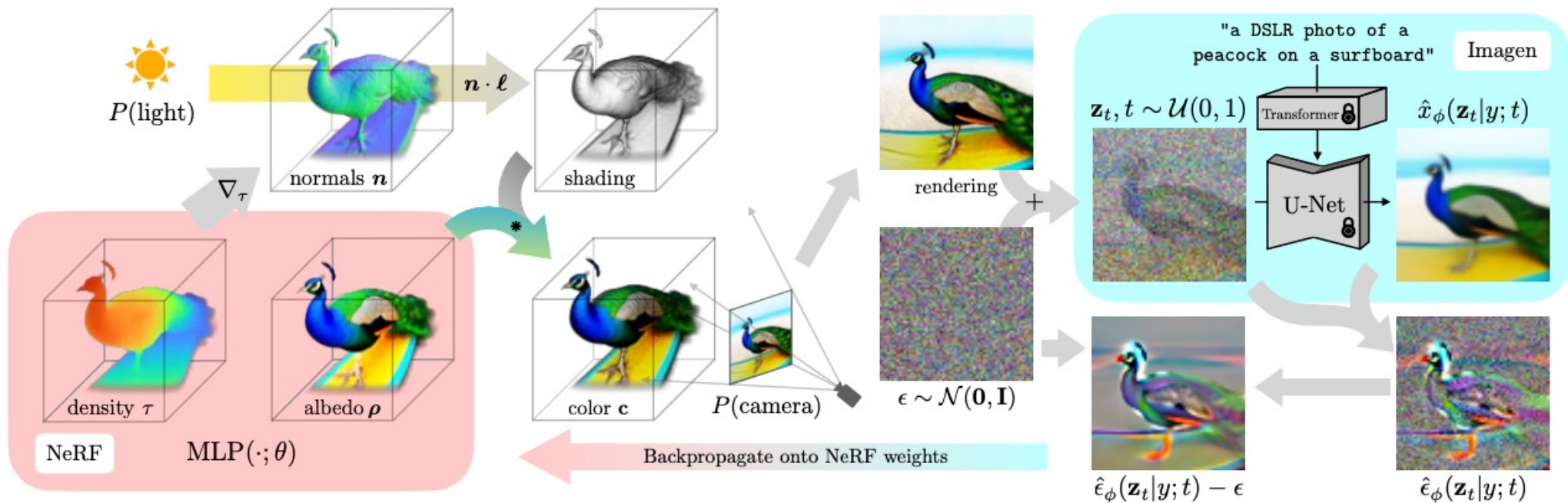
Figure 2. **Overview of proposed CF-3DGS.** Our method takes a sequence of images as input to learn a set of 3D Gaussian that presents the input scene and jointly estimates the camera poses of the frames. We first introduce a local 3DGS to estimate the relative pose of two nearby frames by approximating the Gaussian transformation. Then, a global 3DGS is utilized to model the scene by progressively growing the set of 3D Gaussian as the camera moves.

Text-to-NeRF generation



B. Poole, A. Jain, J. Barron, B. Mildenhall. [DreamFusion: Text-to-3D using 2D Diffusion](#). ICLR 2023

Text-to-NeRF generation



B. Poole, A. Jain, J. Barron, B. Mildenhall. [DreamFusion: Text-to-3D using 2D Diffusion](#). ICLR 2023

Othe NeRF-related topics

Awesome Neural Radiance Fields awesome

A curated list of awesome neural radiance fields papers, inspired by [awesome-computer-vision](#).

How to Pull Request?

If you are interested in adding papers, feel free to submit a pull request following the instruction [here](#).

Table of Contents

- [Survey](#)
- [Papers](#)
- [Talks](#)

Survey

- [Neural Volume Rendering: NeRF And Beyond](#), Dellaert and Yen-Chen, Arxiv 2020 | [blog](#) | [github](#)

Papers

- [NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis](#), Mildenhall et al., ECCV 2020 | [github](#) | [bibtex](#)
- [NeRF++: Analyzing and Improving Neural Radiance Fields](#), Zhang et al., Arxiv 2020 | [github](#) | [bibtex](#)
- [DeRF: Decomposed Radiance Fields](#), Rebain et al. Arxiv 2020 | [bibtex](#)
- [NeRD: Neural Reflectance Decomposition from Image Collections](#), Boss et al., Arxiv 2020 | [github](#) | [bibtex](#)
- [NeRF--: Neural Radiance Fields Without Known Camera Parameters](#), Wang et al., Arxiv 2021 | [github](#) | [bibtex](#)

Faster Inference

- [Neural Sparse Voxel Fields](#), Liu et al., NeurIPS 2020 | [github](#) | [bibtex](#)
- [Autoint: Automatic Integration for Fast Neural Volume Rendering](#), Lindell et al., Arxiv 2020 | [bibtex](#)

Unconstrained Images

- [NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections](#), Martin-Brualla et al., Arxiv 2020 | [bibtex](#)

Deformable

- [Deformable Neural Radiance Fields](#), Park et al., Arxiv 2020 | [github](#) | [bibtex](#)
- [D-NeRF: Neural Radiance Fields for Dynamic Scenes](#), Pumarola et al., Arxiv 2020 | [bibtex](#)
- [Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction](#), Gafni et al., Arxiv 2020 | [bibtex](#)
- [Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Deforming Scene from Monocular Video](#), Tretschk et al., Arxiv 2020 | [github](#) | [bibtex](#)

Video

- [Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes](#), Li et al., Arxiv 2020 | [bibtex](#)
- [Space-time Neural Irradiance Fields for Free-Viewpoint Video](#), Xian et al., Arxiv 2020 | [bibtex](#)
- [Neural Radiance Flow for 4D View Synthesis and Video Processing](#), Du et al., Arxiv 2020 | [bibtex](#)
- [Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans](#), Peng et al., Arxiv 2020 | [bibtex](#)

Generalization

- [GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis](#), Schwarz et al., NeurIPS 2020 | [github](#) | [bibtex](#)
- [GRF: Learning a General Radiance Field for 3D Scene Representation and Rendering](#), Trevithick and

<https://github.com/yenchenlin/awesome-NeRF>