Plenoxels: Radiance Fields without Neural Networks

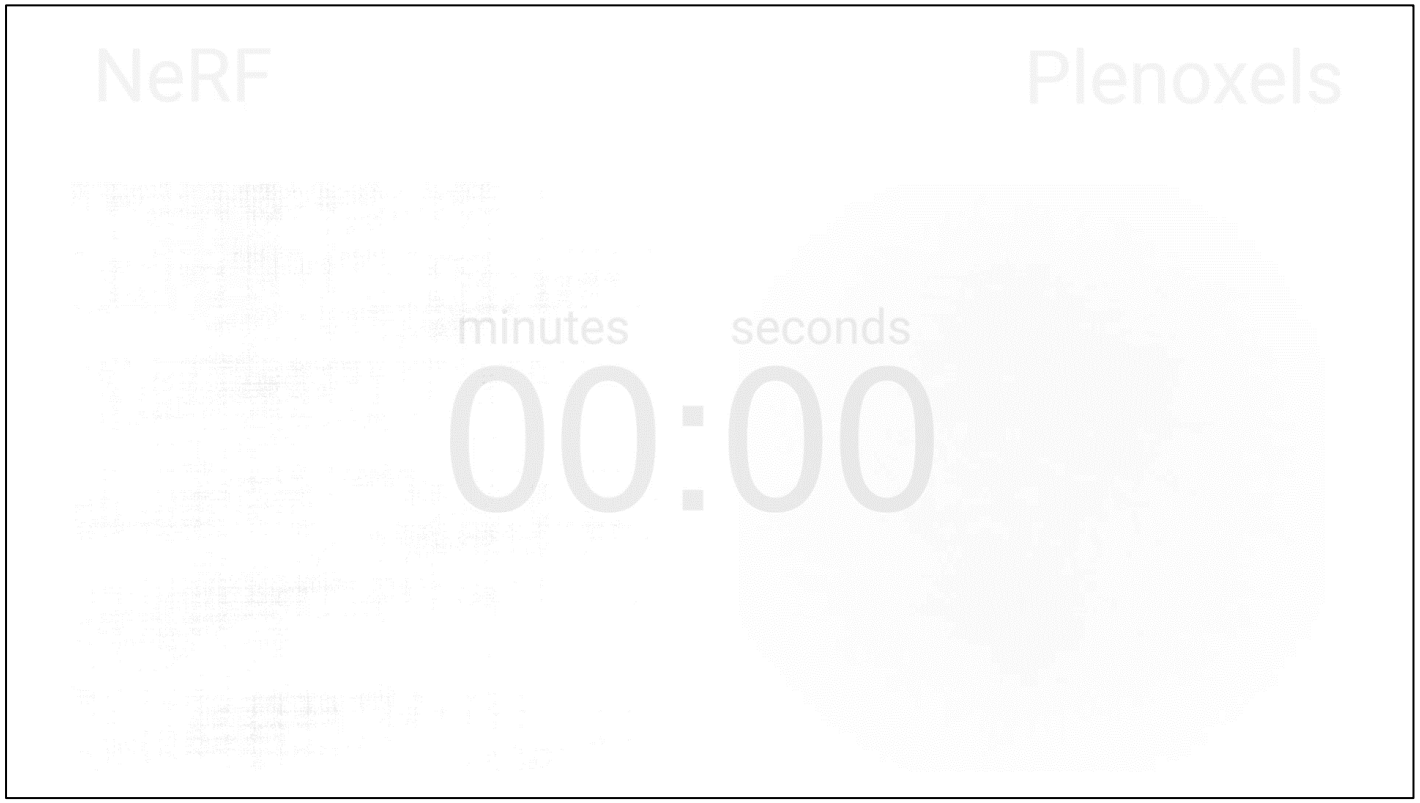
Alex Yu*, Sara Fridovich-Keil*, Matthew Tancik, Qinhong Chen,
Benjamin Recht, Angjoo Kanazawa

UC Berkeley

Presenter: Minho Park

Demos

- Fast optimization



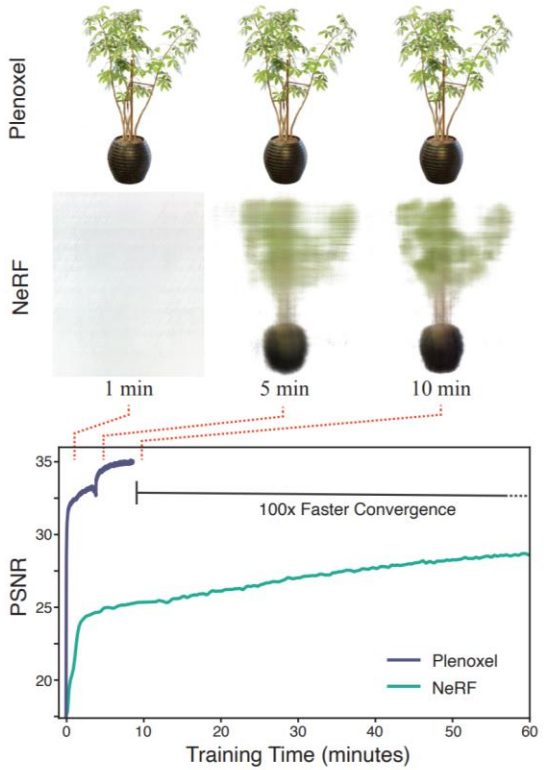


Figure 1. **Plenoxel: Plenoptic Volume Elements** for fast optimization of radiance fields. We show that direct optimization of a fully explicit 3D model can match the rendering quality of modern neural based approaches such as NeRF while optimizing over two orders of magnitude faster.

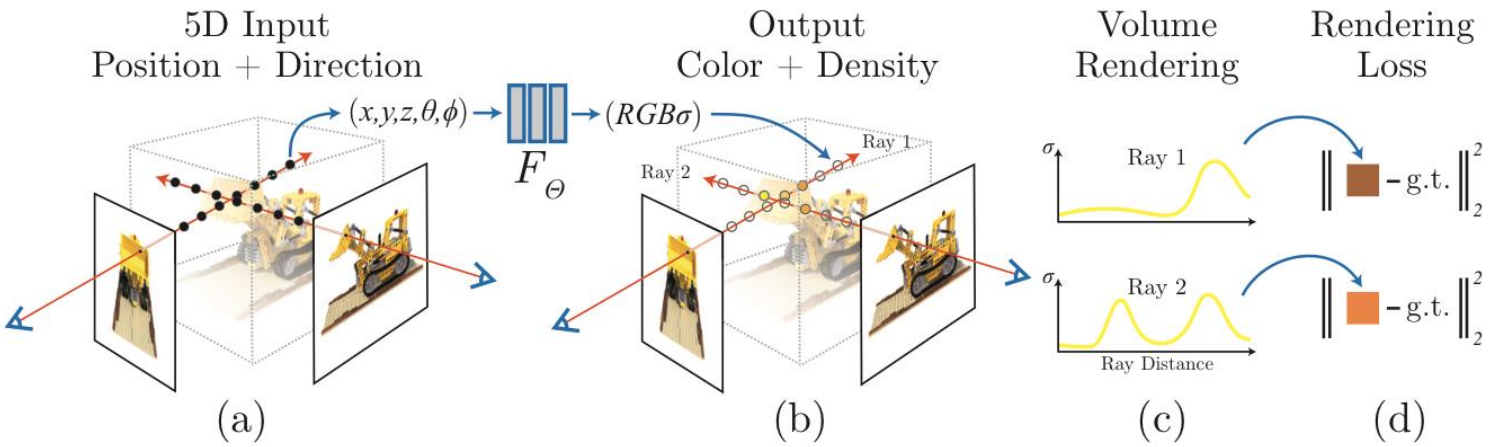
https://alexyu.net/plenoxels/

Volume Rendering

- Integrate over samples taken along the ray:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i$$

$$\text{where } T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right)$$

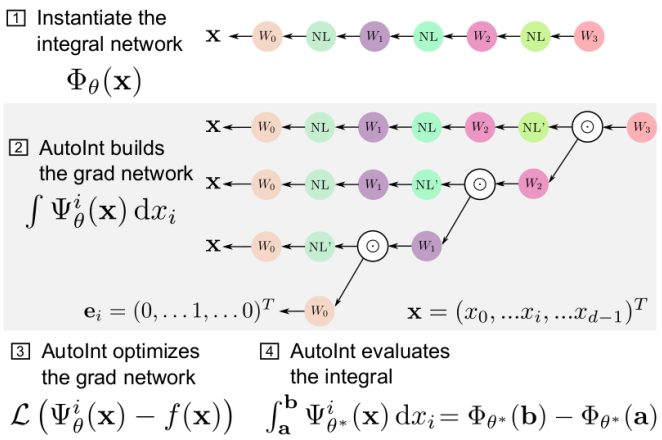


NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

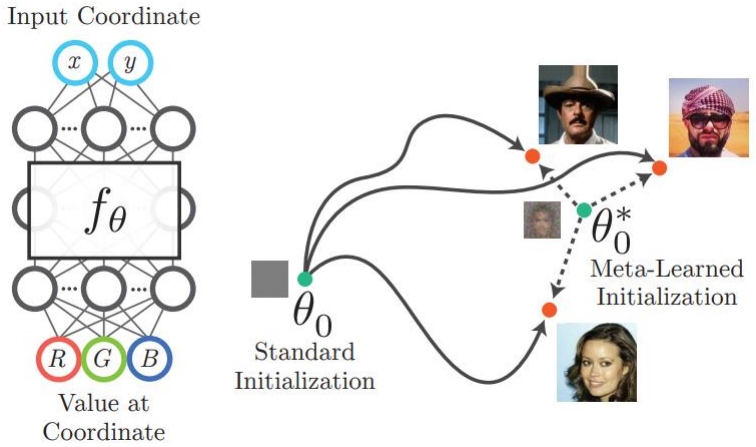
Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *European conference on computer vision*. Springer, Cham, 2020.

Accelerating NeRF

- Subdividing the 3D volume into regions [19, 35].
- AutoInt [18] restructures the coordinate-based MLP to compute ray integrals exactly, for more than 10× faster rendering with a small loss in quality.
- Learned Initializations [49] employs meta-learning on many scenes to start from a better MLP initialization, for both > 10× faster training and better priors when per-scene data is limited



AutoInt [18]



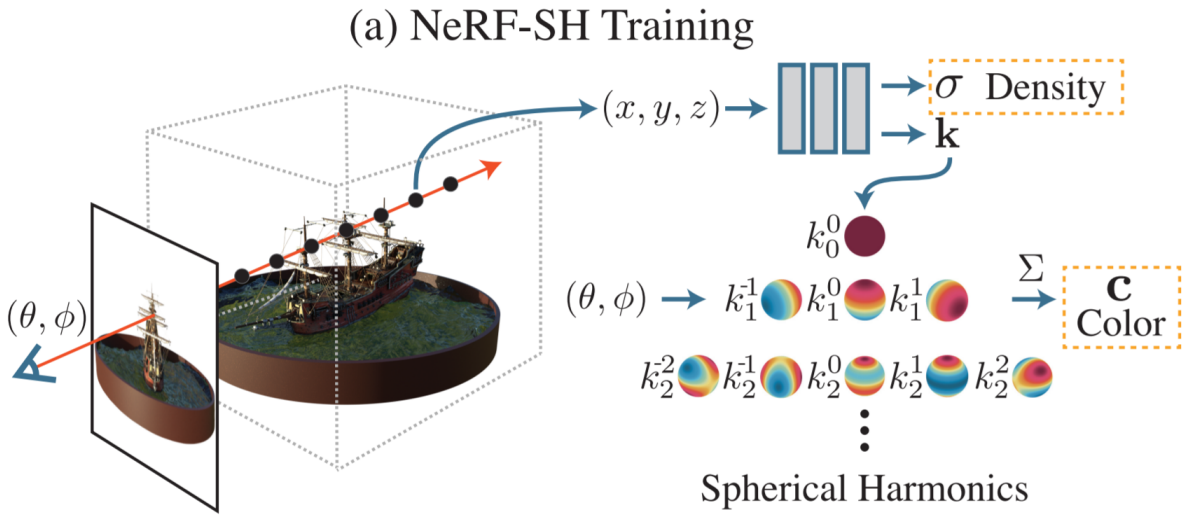
Learning initialization [49]

[19]: Liu, Lingjie, et al. "Neural sparse voxel fields." *arXiv preprint arXiv:2007.11571* (2020).
[35]: Rebain, Daniel, et al. "Derf: Decomposed radiance fields." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
[18]: Lindell, David B., Julien NP Martel, and Gordon Wetzstein. "Autoint: Automatic integration for fast neural volume rendering." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
[49]: Tancik, Matthew, et al. "Learned initializations for optimizing coordinate-based neural representations." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

NeRF-SH: NeRF with Spherical Harmonic

- SHs have been a popular low-dimensional representation for spherical functions and have been used to model Lambertian surfaces or even glossy surfaces.

  Lambertian reflectance is the property that defines an ideal "matte" or diffusely reflecting surface.



PlenOctree: a modified NeRF model (NeRF-SH)

Yu, Alex, et al. "Plenoctrees for real-time rendering of neural radiance fields." *arXiv preprint arXiv:2103.14024* (2021).

NeRF-SH: NeRF with Spherical Harmonic

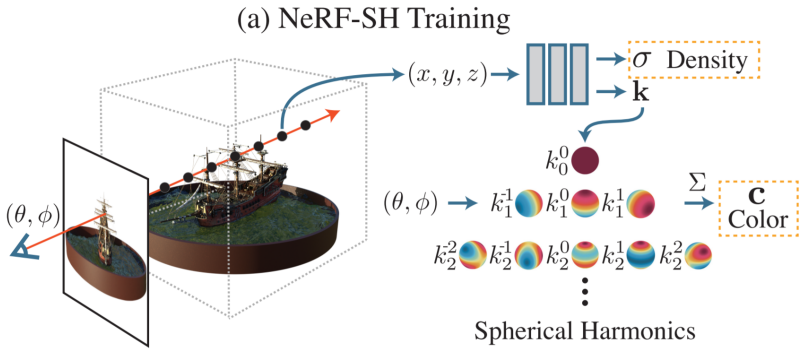
- Factorize the view-dependent appearance with the SH basis.
- Each $k_l^m \in \mathbb{R}^3$ is a set of 3 coefficients corresponding to the RGB components.

$$f(x) = (k, \sigma) \text{ where } k = (k_l^m)_{l:0 \le l \le l_{max}}^{m:-l \le m \le l}$$

$$c(\underbrace{d}_{\theta,\phi}; k) = \sigma\left(\sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} k_l^m Y_l^m(d)\right)$$

Spherical harmonic function

- Similar to Fourier coefficients.



PlenOctree: a modified NeRF model (NeRF-SH)

Yu, Alex, et al. "Plenoctrees for real-time rendering of neural radiance fields." *arXiv preprint arXiv:2103.14024* (2021).

Wikipedia. “Spherical harmonics” https://en.wikipedia.org/wiki/Spherical_harmonics

Overview

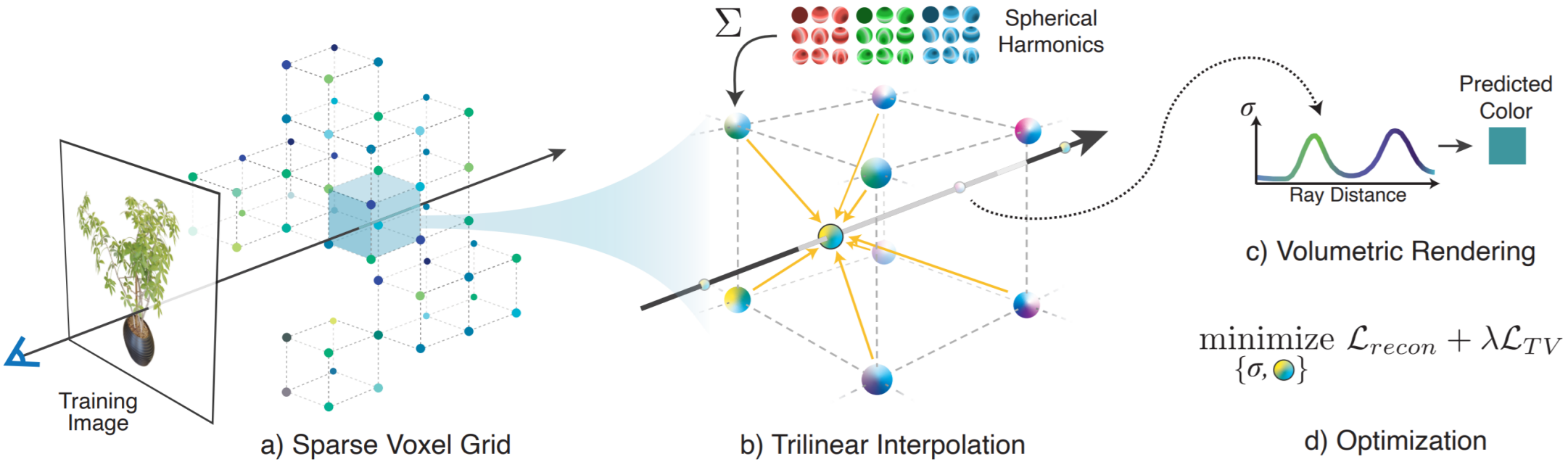
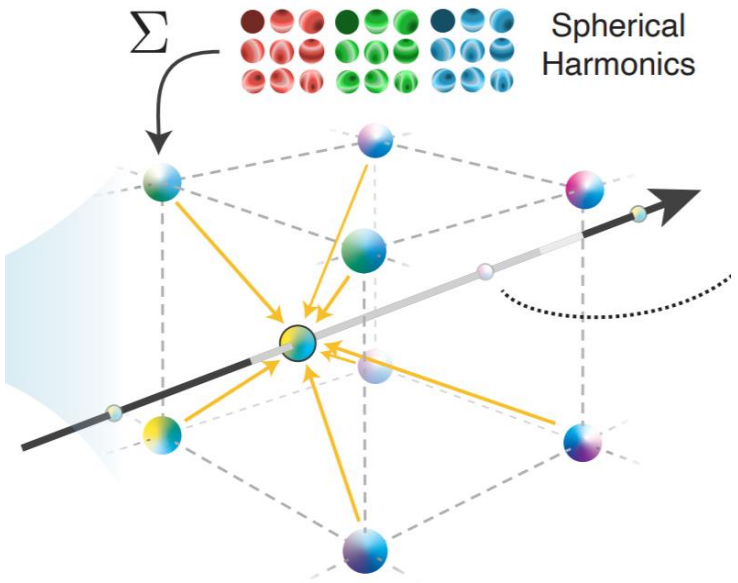


Figure 2. **Overview of our sparse Plenoxel model.** Given a set of images of an object or scene, we reconstruct a (a) sparse voxel ("Plenoxel") grid with density and spherical harmonic coefficients at each voxel. To render a ray, we (b) compute the color and opacity of each sample point via trilinear 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.

Interpolation

- Trilinear interpolation of opacity and harmonic coefficients stored at the nearest 8 voxels.
- Benefits of interpolation
  - Interpolation increases the effective resolution by representing sub-voxel variations in color and opacity.
  - Interpolation produces a continuous function approximation that is critical for successful optimization.



b) Trilinear Interpolation

	PSNR ↑	SSIM ↑	LPIPS ↓
Trilinear, 256^3	30.57	0.950	0.065
Trilinear, 128^3	28.46	0.926	0.100
Nearest Neighbor, 256^3	27.17	0.914	0.119
Nearest Neighbor, 128^3	23.73	0.866	0.176

Table 1. **Ablation over interpolation method.** Results are averaged over the 8 NeRF synthetic scenes. We find that trilinear interpolation provides dual benefits of improving effective resolution and improving optimization, such that trilinear interpolation at resolution 128^3 outperforms nearest neighbor interpolation at 256^3.

Coarse to Fine

- Coarse-to-fine strategy
    1. Begins with a dense grid at lower resolution
    2. Optimizes
    3. Prunes unnecessary voxels (threshold $T_i(1 - \exp(-\sigma_i \delta_i))$ or σ_i)
    4. Refines the remaining voxels by subdividing each in half in each dimension
    5. Continues optimizing

Optimization

- Mean squared error (MSE) and total variation (TV) regularization.
- Use RMSProp because of poor conditioning.

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda_{TV}\mathcal{L}_{TV}$$

$$\mathcal{L}_{recon} = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \left\| C(r) - \hat{C}(r) \right\|_2^2$$

$$\mathcal{L}_{TV} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V},d\in[D]} \sqrt{\Delta_x^2(v,d) + \Delta_y^2(v,d) + \Delta_z^2(v,d)}$$

For σ and SH coefficients

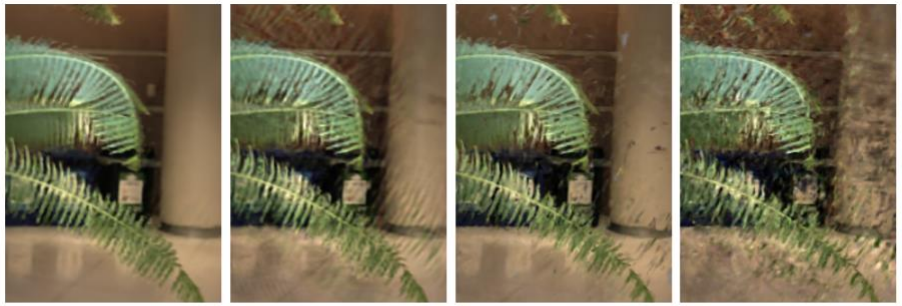


Figure 3. **Ablation over TV regularization.** Clear artifacts are visible in the forward-facing scenes without TV on both σ and SH coefficients, although PSNR does not always reflect this.

Implementation Details

- The speed of their implementation is possible in large part because the gradient of our Plenoxel model becomes very sparse very quickly.

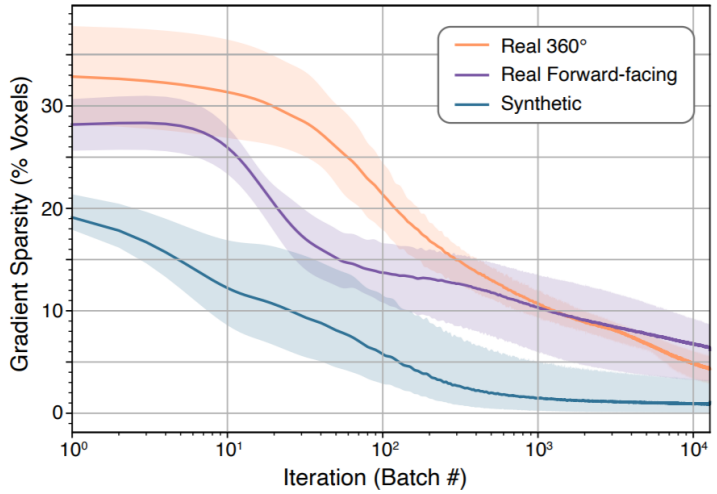


Figure 4. **Gradient sparsity.** The gradient becomes very sparse spatially within the first 12800 batches (one epoch for the synthetic scenes), with as few as 1% of the voxels updating per batch in the synthetic case. This enables efficient training via sparse parameter updates. The solid lines show the mean and the shaded regions show the full range of values among all scenes of each type.

Results

- Synthetic Scenes
  - Use the 8 scenes from NeRF: chair, drums, ficus, hotdog, lego, materials, mic, and ship.

Figure 5. **1 minute, 20 seconds.** Results on the synthetic scenes after 1 epoch of optimization, an average of 1 minute and 20 seconds.

	PSNR ↑	SSIM ↑	LPIPS ↓	Train Time
Ours	31.71	**0.958**	**0.049**	**11 mins**
NV [20]	26.05	0.893	0.160	>1 day
JAXNeRF [7, 26]	**31.85**	0.954	0.072	1.45 days
Ours	26.29	**0.839**	**0.210**	**24 mins**
LLFF [25]	24.13	0.798	0.212	—*
JAXNeRF [7, 26]	**26.71**	0.820	0.235	1.62 days
Ours	20.40	**0.696**	**0.420**	**27 mins**
NeRF++ [57]	**20.49**	0.648	0.478	∼4 days

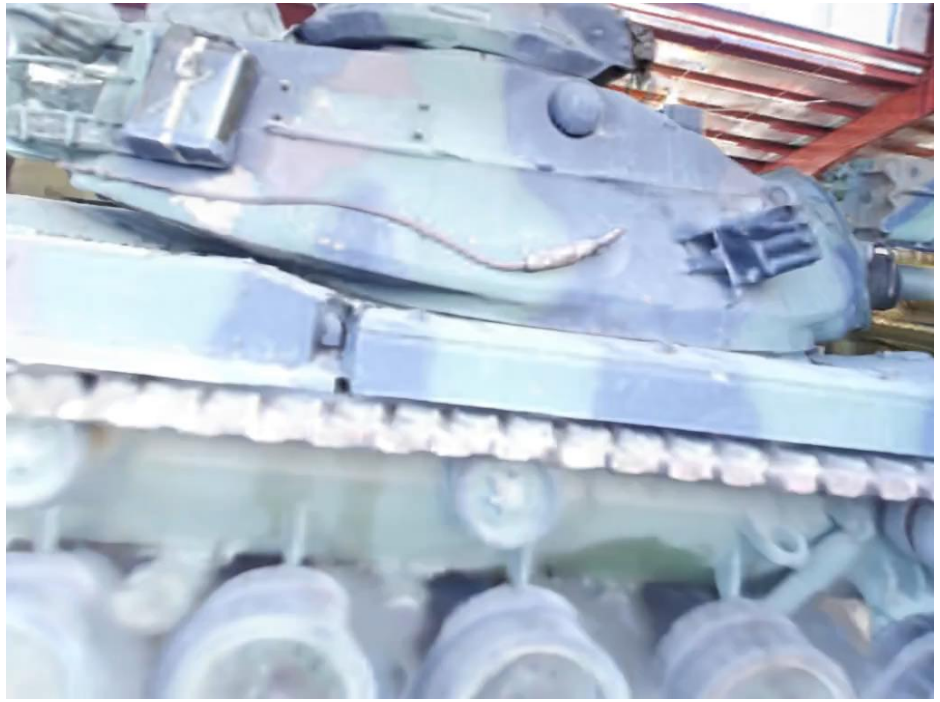
Table 2. **Results.** *Top:* average over the 8 synthetic scenes from NeRF; *Middle:* the 8 real, forward-facing scenes from NeRF; *Bottom:* the 4 real, 360° scenes from Tanks and Temples [15]. 4 of the synthetic scenes train in under 10 minutes. *LLFF requires pretraining a network to predict MPIs for each view, and then can render novel scenes without further training; this pretraining is amortized across all scenes so we do not include it in the table.

Results

- Real Forward-Facing Scenes

Results

- Real 360° Scenes



Ablation Studies

- Extensive ablation studies.

	PSNR ↑	SSIM ↑	LPIPS ↓
Ours: 100 images (low TV)	31.71	0.958	0.050
NeRF: 100 images [26]	31.01	0.947	0.081
Ours: 25 images (low TV)	26.88	0.911	0.099
Ours: 25 images (high TV)	28.25	0.932	0.078
NeRF: 25 images [26]	27.78	0.925	0.108

Table 3. **Ablation over the number of views.** By increasing our TV regularization, we exceed NeRF fidelity even when the number of training views is only a quarter of the full dataset. Results are averaged over the 8 synthetic scenes from NeRF.

Resolution	PSNR ↑	SSIM ↑	LPIPS ↓
512^3	31.71	0.958	0.050
256^3	30.57	0.950	0.065
128^3	28.46	0.926	0.100
64^3	26.11	0.892	0.139
32^3	23.49	0.859	0.174

Table 4. **Ablation over the Plenoxel grid resolution.** Results are averaged over the 8 synthetic scenes from NeRF.

Rendering Formula	PSNR ↑	SSIM ↑	LPIPS ↓
Max [22], used in NeRF [26]	30.57	0.950	0.065
Neural Volumes [20]	27.54	0.906	0.201

Table 5. **Comparison of different rendering formulas.** We compare the rendering formula from Max [22] (used in NeRF and our main method) to the one used in Neural Volumes [20], which uses absolute instead of relative transmittance. Results are averaged over the 8 synthetic scenes from NeRF.

Future Work

- The method exhibits different artifacts than neural methods.
- Future work may be able to adjust or mitigate these remaining artifacts by studying different regularization priors.

Figure 9. **Artifacts.** JAXNeRF and Plenoxel models both exhibit artifacts, but the artifacts are different, as shown here in the specularities in the synthetic drums scene. Note that some artifacts are unavoidable for any underdetermined inverse problem, but the specific artifacts vary depending on the priors induced by the model and regularizer.