NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

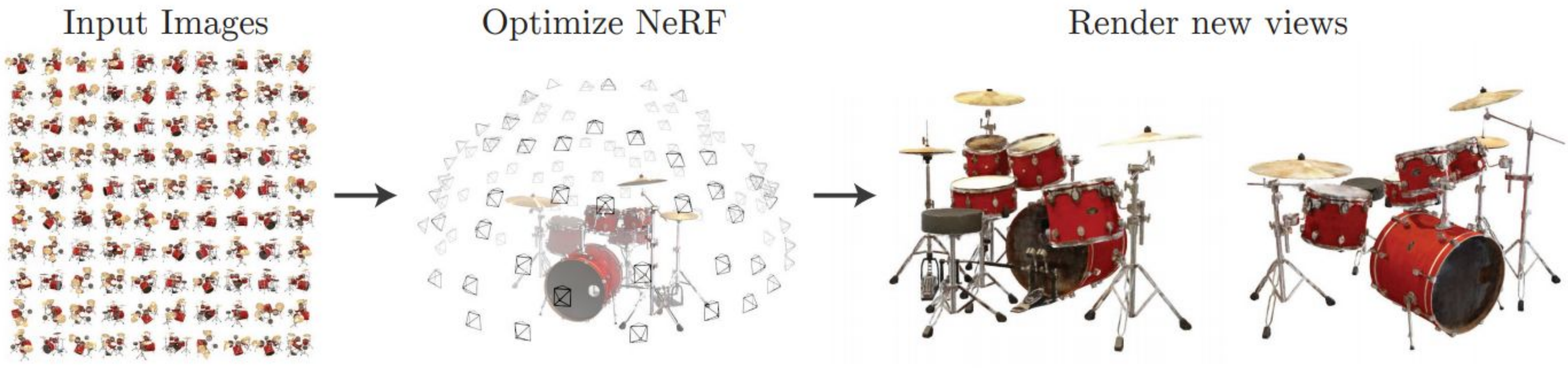
Ben Mildenhall[1⋆] Pratul P. Srinivasan[1⋆] Matthew Tancik[1⋆]
Jonathan T. Barron[2] Ravi Ramamoorthi[3] Ren Ng[1]

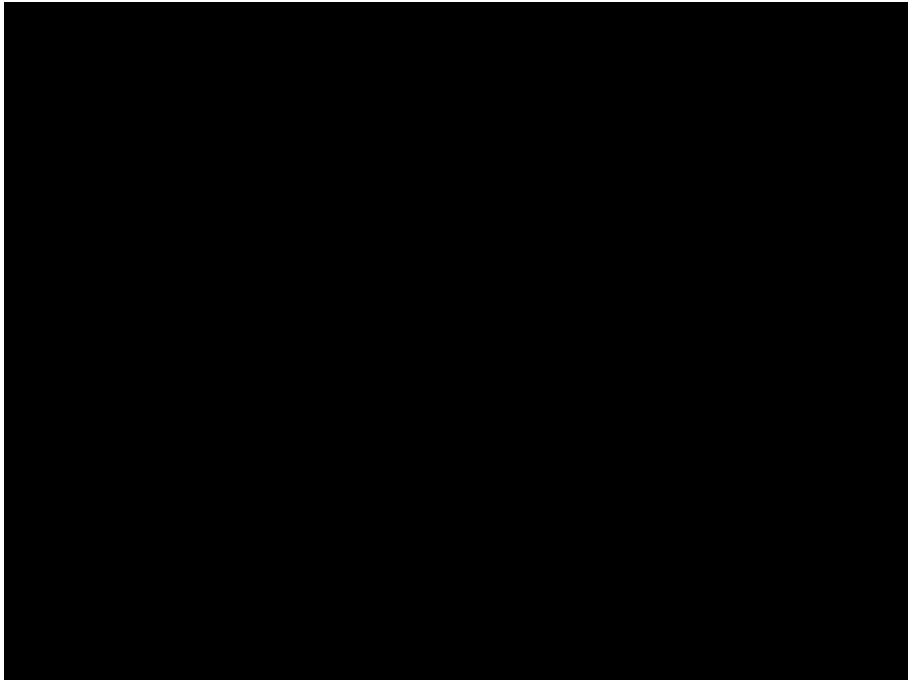
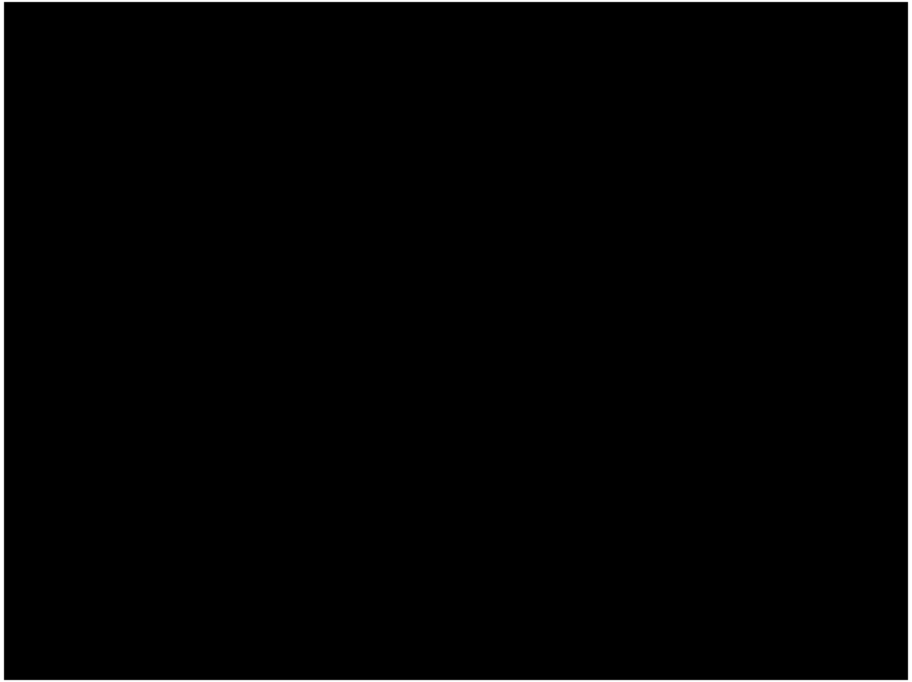
[1]UC Berkeley [2]Google Research [3]UC San Diego

Prepared by Allie 07/14/20

NeRF (Neural Radiance Field)

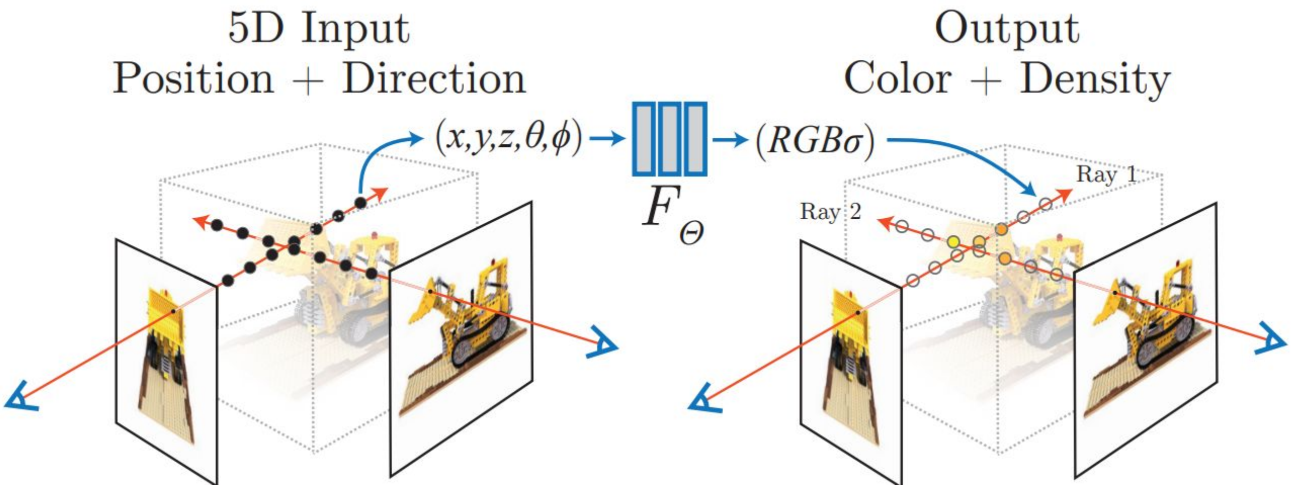
- Cool videos: https://www.matthewtancik.com/nerf
- The first continuous neural scene representation that is able to render high-resolution photorealistic novel views of real objects and scenes from RGB images captured in natural settings



NeRF (Neural Radiance Field)

- Synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views
- “underlying continuous volumetric scene function” = a fully-connected network
    - Input : a single continuous 5D coordinate (spatial location (x, y, z) and viewing direction (θ, φ))
    - Output : the volume density and view-dependent emitted radiance at that spatial location.



Volume Rendering

- Create a 2D projection from a discretely sampled 3D data set
  - Given camera poses and intrinsics, render a 2D image from a 3D volumetric representation

$\sigma(\mathbf{x})$ probability of a ray terminating at an infinitesimal particle at location

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ camera ray

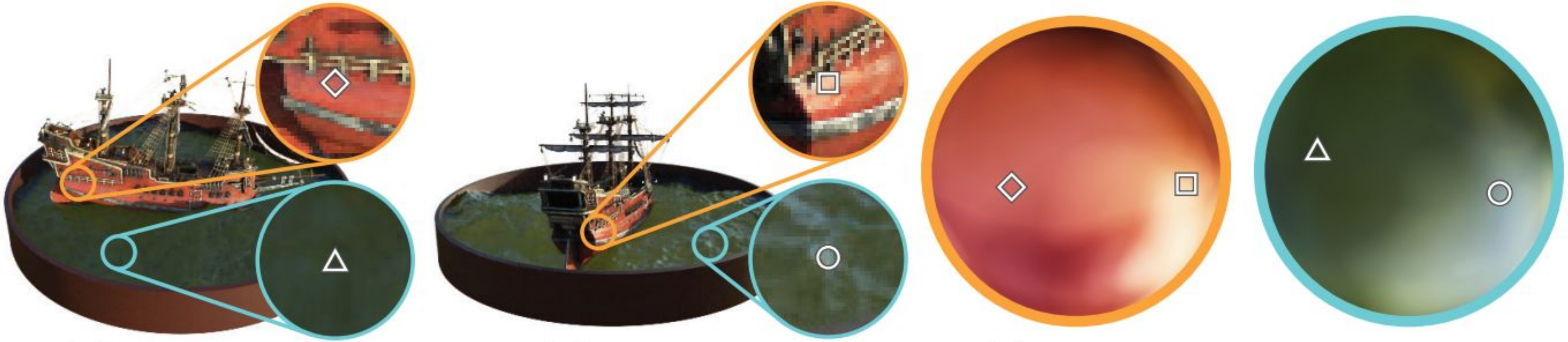
$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

volume density (independent of viewing direction d)

emitted color (change with viewing direction d)

accumulated transmittance along the ray from t_n to t (Derivation here link)

View-dependent emitted radiance

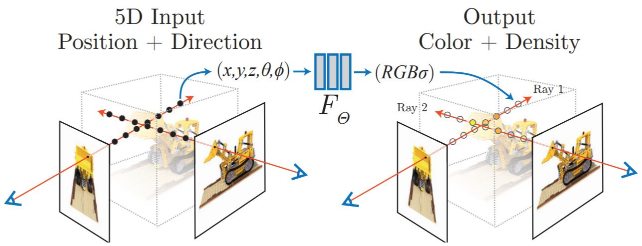


(a) View 1 (b) View 2 (c) Radiance Distributions

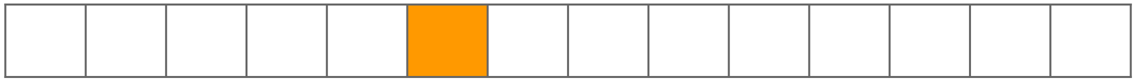
Volume Rendering (discretized)

- Numerically estimate this continuous integral

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right]$$



t_n Draw uniform distribution t_f



$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\boxed{\sigma_i}\delta_i))\boxed{c_i},\ \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1}\boxed{\sigma_j}\delta_j\right)$$

Network outputs

Improve the Quality

- Positional encoding
  - Map the input to high dimensional space before sending into the network
  - Preserve high frequency details
  - Do x, y, z separately to position(xyz) and d(ray direction)

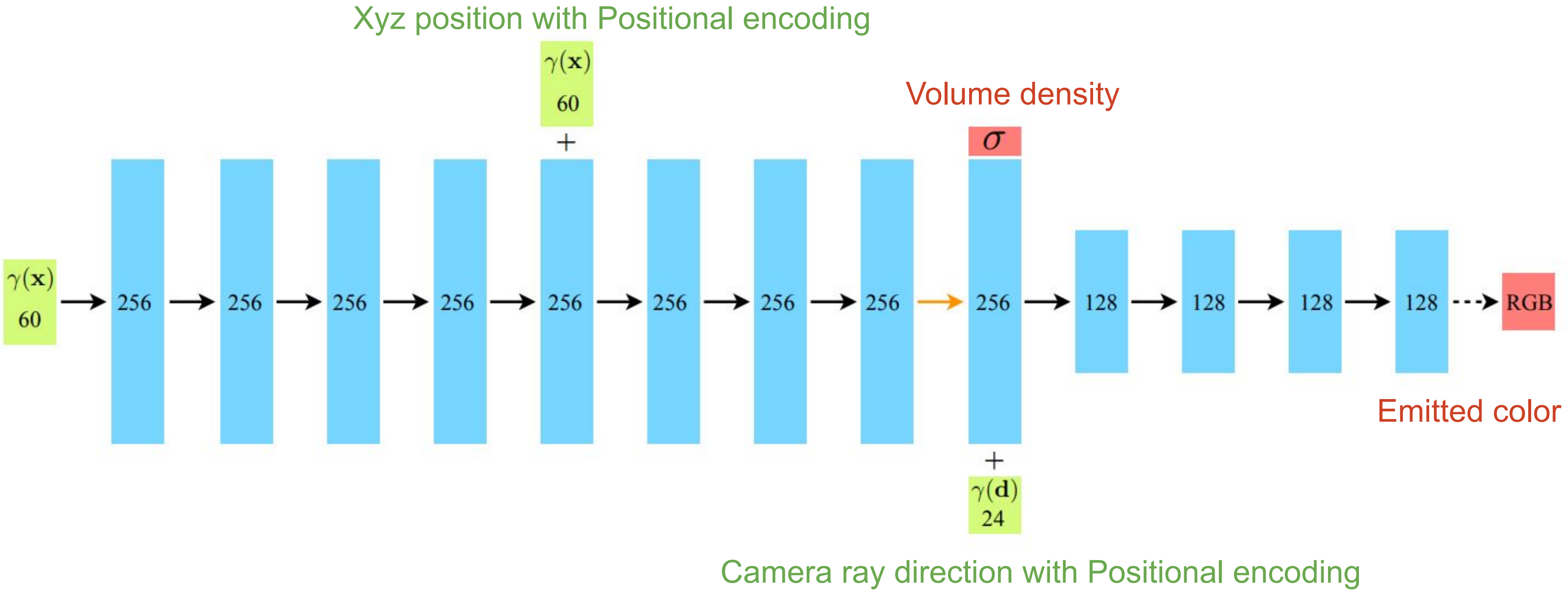
$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$

- Hierarchical volume sampling
  - Optimize two networks: one “coarse” and one “fine”
  - Use the output of the coarse network to adjust the sampling for the fine network
  - Compute the final rendered color of the ray using all the sample (fine + coarse)

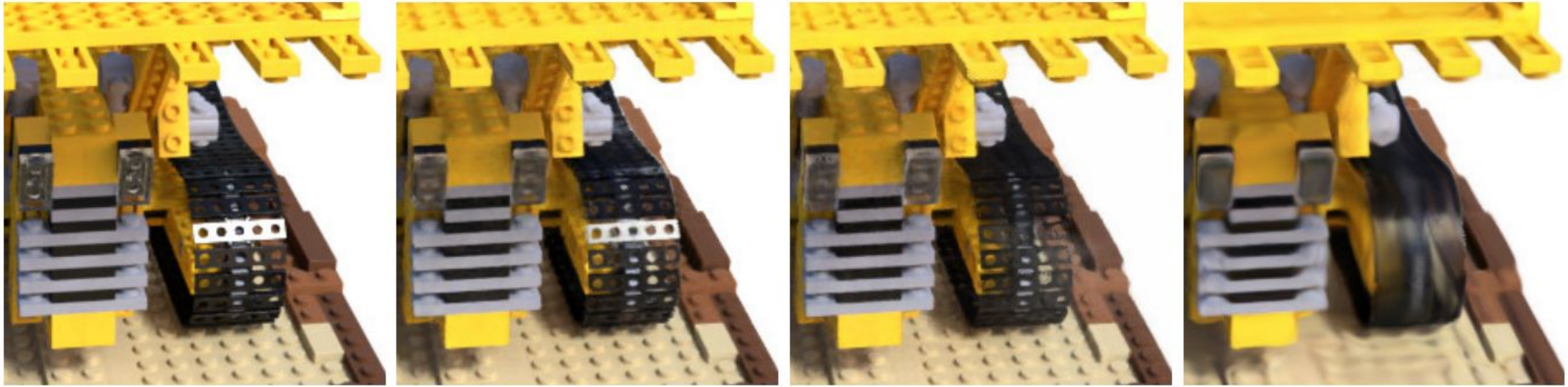
$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} \boxed{w_i} c_i \,, \qquad w_i = T_i(1 - \exp(-\sigma_i \delta_i))$$

Higher weight, denser sample (for the fine network)

Network Structure



Comparison



Ground Truth | Complete Model | No View Dependence | No Positional Encoding

Implementation

- Camera poses/intrinsics/scene bounds from COLMAP
- Optimize a separate neural continuous volume representation network for each scene
- Loss function (image color difference):

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$

rays

Coarse network output

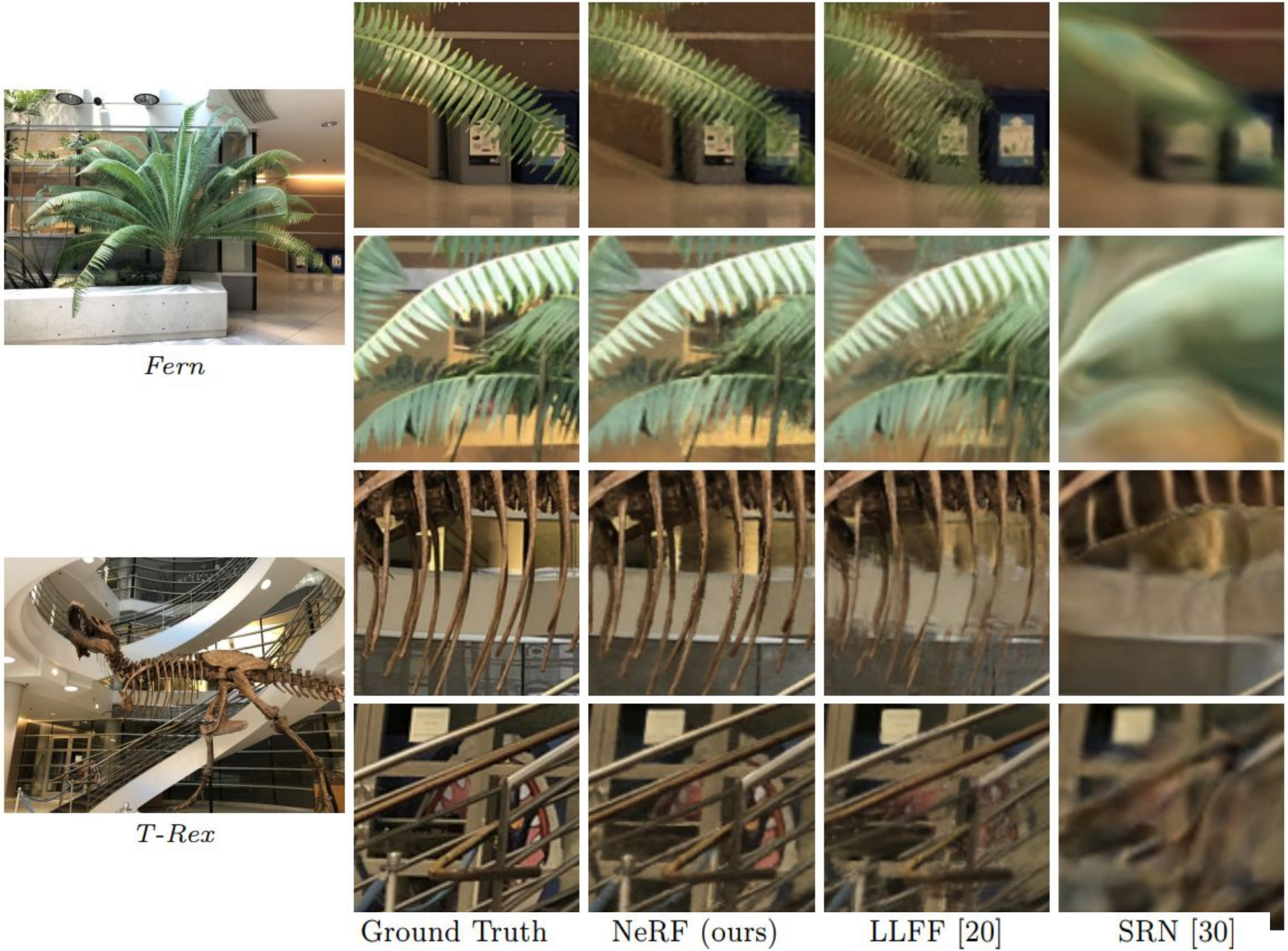
Ground truth

Fine network output

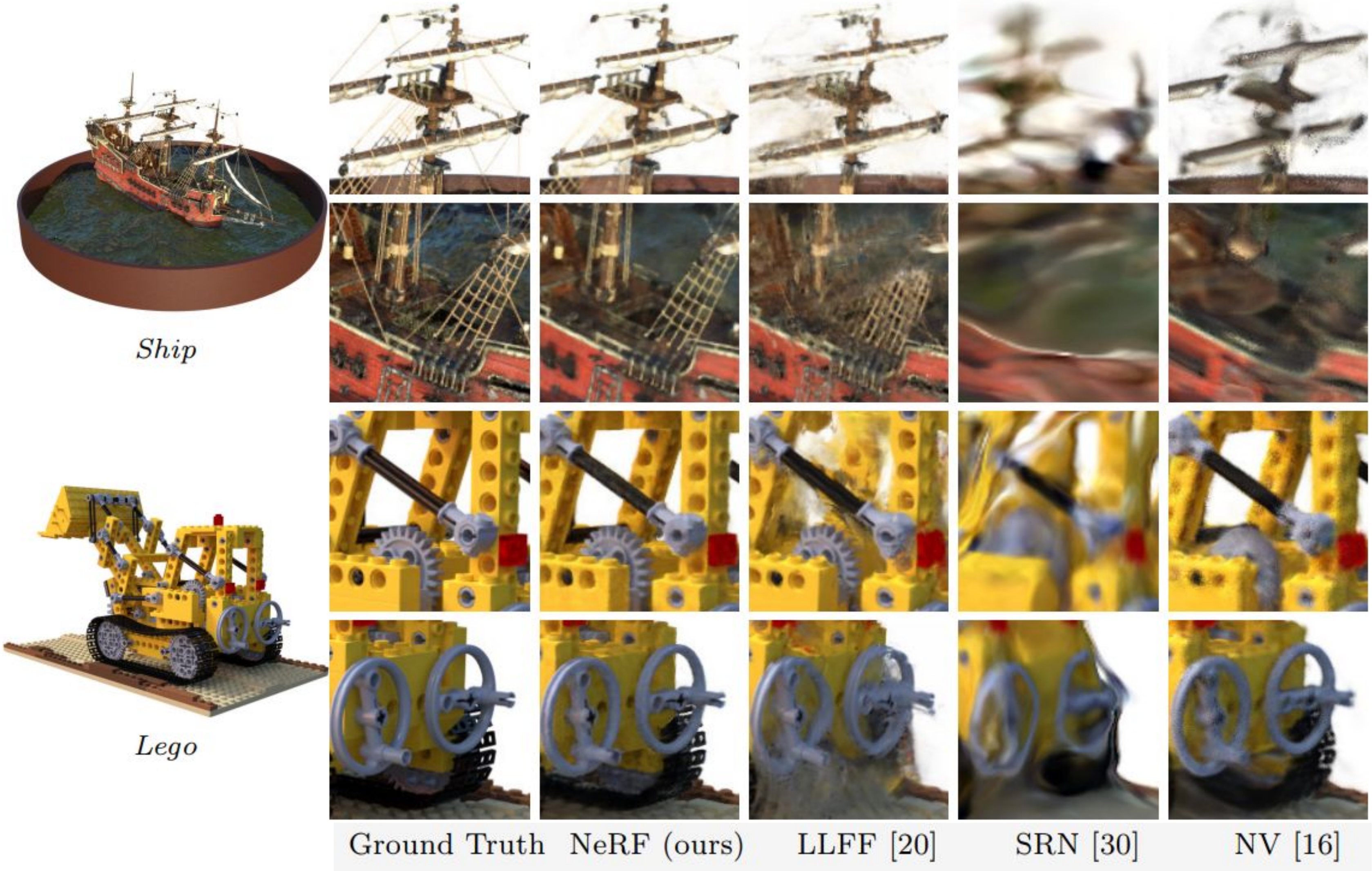
Results

Method	Diffuse Synthetic 360° [29]			Realistic Synthetic 360°			Real Forward-Facing [20]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SRN [30]	33.20	0.986	0.073	22.26	0.867	0.170	22.84	0.866	0.378
NV [16]	29.62	0.946	0.099	26.05	0.944	0.160	-	-	-
LLFF [20]	34.38	0.995	0.048	24.88	0.935	0.114	24.13	0.909	**0.212**
Ours	**40.15**	**0.998**	**0.023**	**31.01**	**0.977**	**0.081**	**26.50**	**0.935**	0.250

LPIPS is a perceptual metric: https://arxiv.org/pdf/1801.03924.pdf



Fern
T-Rex
Ground Truth
NeRF (ours)
LLFF [20]
SRN [30]



Ship
Lego
Ground Truth
NeRF (ours)
LLFF [20]
SRN [30]
NV [16]