NeRF :
Representing Scenes as Neural Radiance Fields for View Synthesis

조민지

Vision & Display Systems Lab.

Dept. of Electronic Engineering, Sogang University

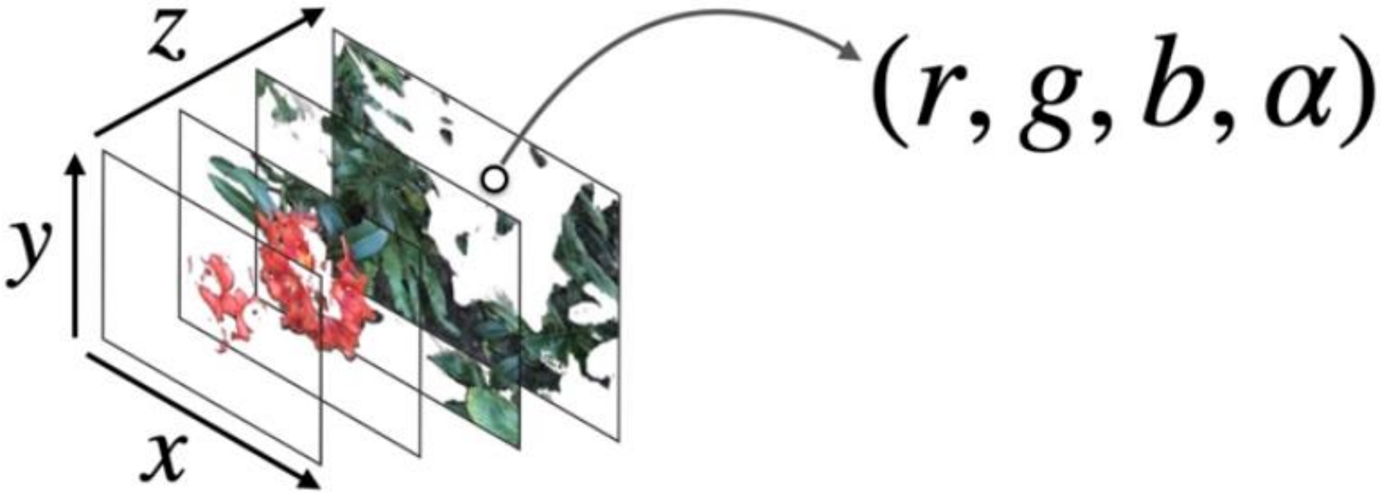
Outline

Intro

- 3D Rendering
  and View synthesis

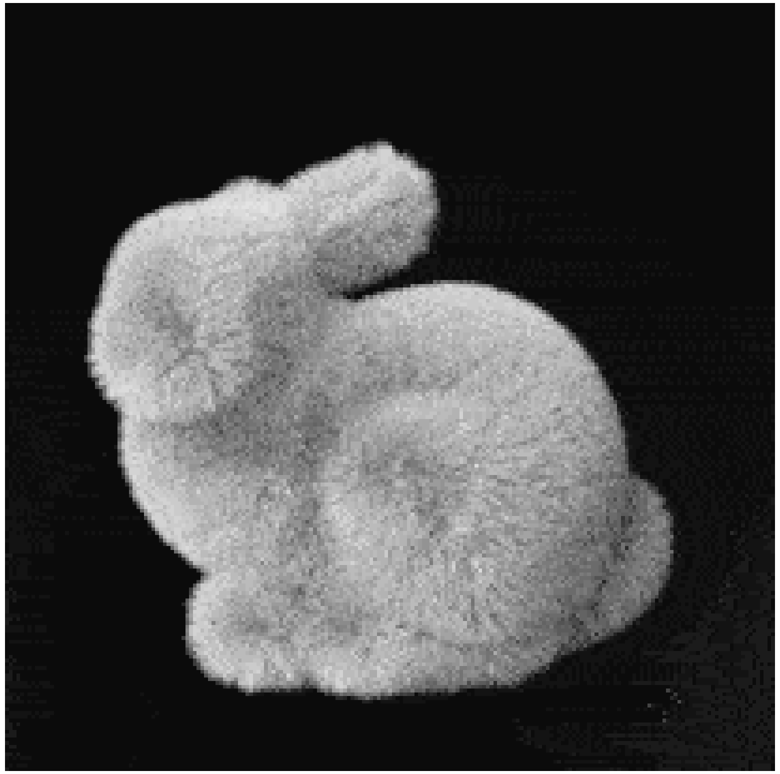
Previous works

- RGB$-\alpha$ volume rendering for synthesis
  - Large N-d array contains RGB$-\alpha$ information.



Previous works

- Neural Network as shape representation
  - Deep SDF
  - Occupancy network

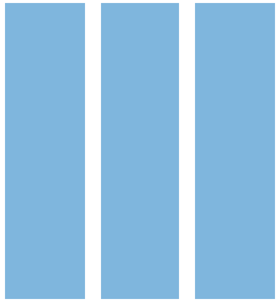


NeRF : Neural Radiance Field Scene Representation

- Neural Network as scene representation
  - Volume rendering with neural radiance fields
  - View synthesis and image-based rendering

Method

- Neural Network as a scene representation
  - Generalization? → X
  - Overfitted by 1 specific scene.
  - Weights of NN represents the scene!



F_{Ω}

Fully-connected neural network

(256 channels, 9 layers)

Method

- Representing a scene as a continuous 5D function

$$\Big(\overbrace{x,\ y,\ z}^{\text{Spatial location } \vec{x}},\ \underbrace{\theta,\ \phi}_{\text{Viewing direction } \vec{d}}\Big) \longrightarrow F_{\Omega} \longrightarrow \Big(\underbrace{r,\ g,\ b}_{\text{Output color } \vec{c}},\ \overbrace{\sigma}^{\text{Output density}}\Big)$$

Method

- Getting pose & bound from images
  - Capture images of [1] forward facing view or [2] 360 ° inward facing view.
  - Use COLMAP to get camera-to-world (c2w) transformation matrix.

Method

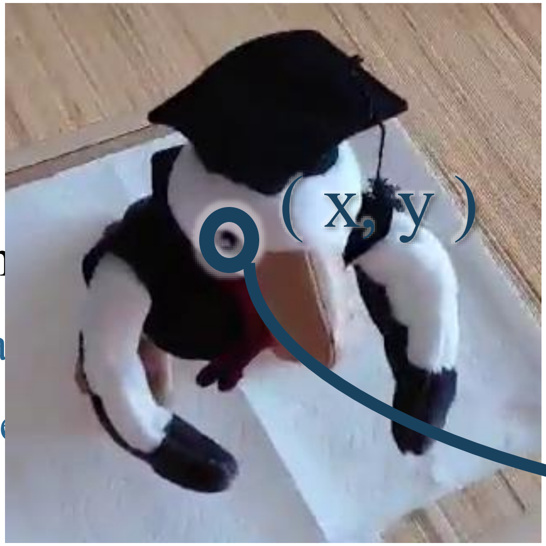
- Getting pose & bound from images
  - Capture images of [1] forward facing view or [2] 360 ° inward facing view.
  - Use COLMAP to get camera-to-world (c2w) transformation matrix.

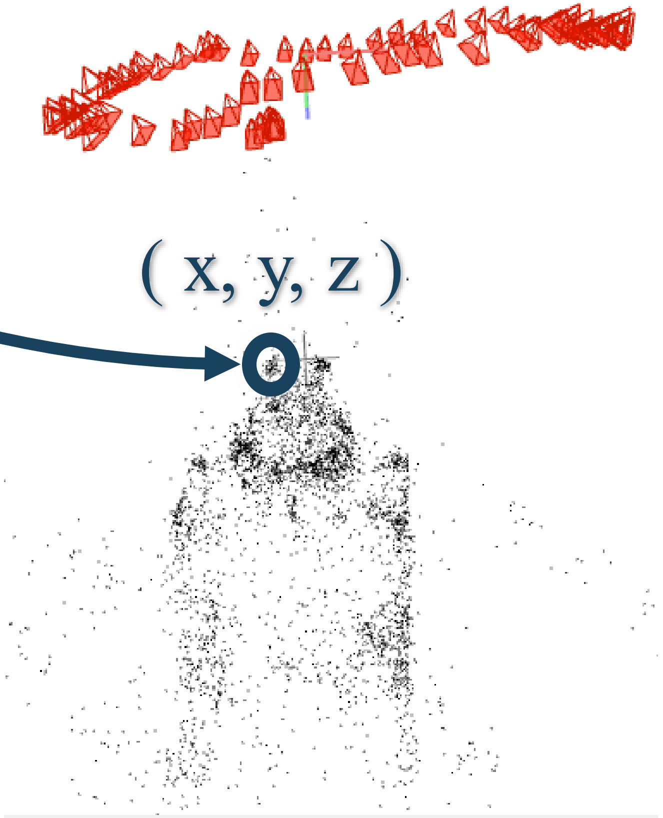
Method

- Getting pose & bound from images
  - Capture images of [1] forward facing view or [2] 360 ˚ inward facing view.
  - Use COLMAP to get camera-to-world (c2w) transformation matrix.

Method

- Getting pose & bound fron
  - Capture images of ¹ forwa
    or ² 360 ˚ inward facing vi





  - Use COLMAP to get camera-to-world (c2w) transformation matrix.

Method

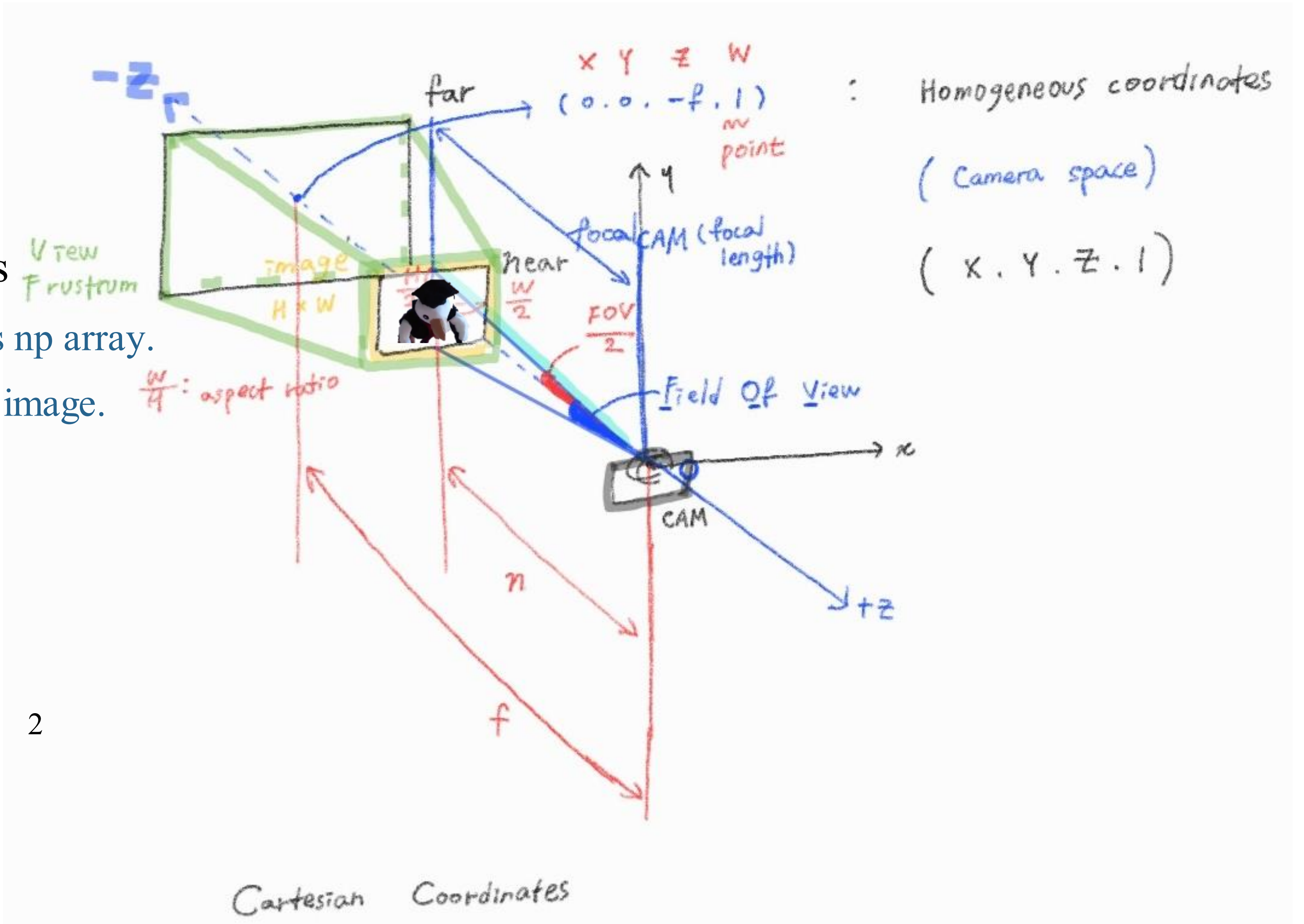
- Getting pose & bound from images
  - COLMAP generates poses_bounds np array.
  - Contains 1 x 17 matrix per 1 input image.

$$\Big\downarrow \text{Reshape}$$

$$\begin{array}{cc} & 4 \\ 3 & \begin{bmatrix} \text{C-to-W} \\ \text{Affine} \\ \text{transform} \end{bmatrix} \end{array} \begin{array}{c} 1 \\ \begin{bmatrix} H_{image} \\ W_{image} \\ focal\ length \end{bmatrix} \end{array} , \begin{array}{cc} 1 & \\ \begin{bmatrix} n \\ f \end{bmatrix} & 2 \end{array}$$

intrinsic parameters

depth values



Method

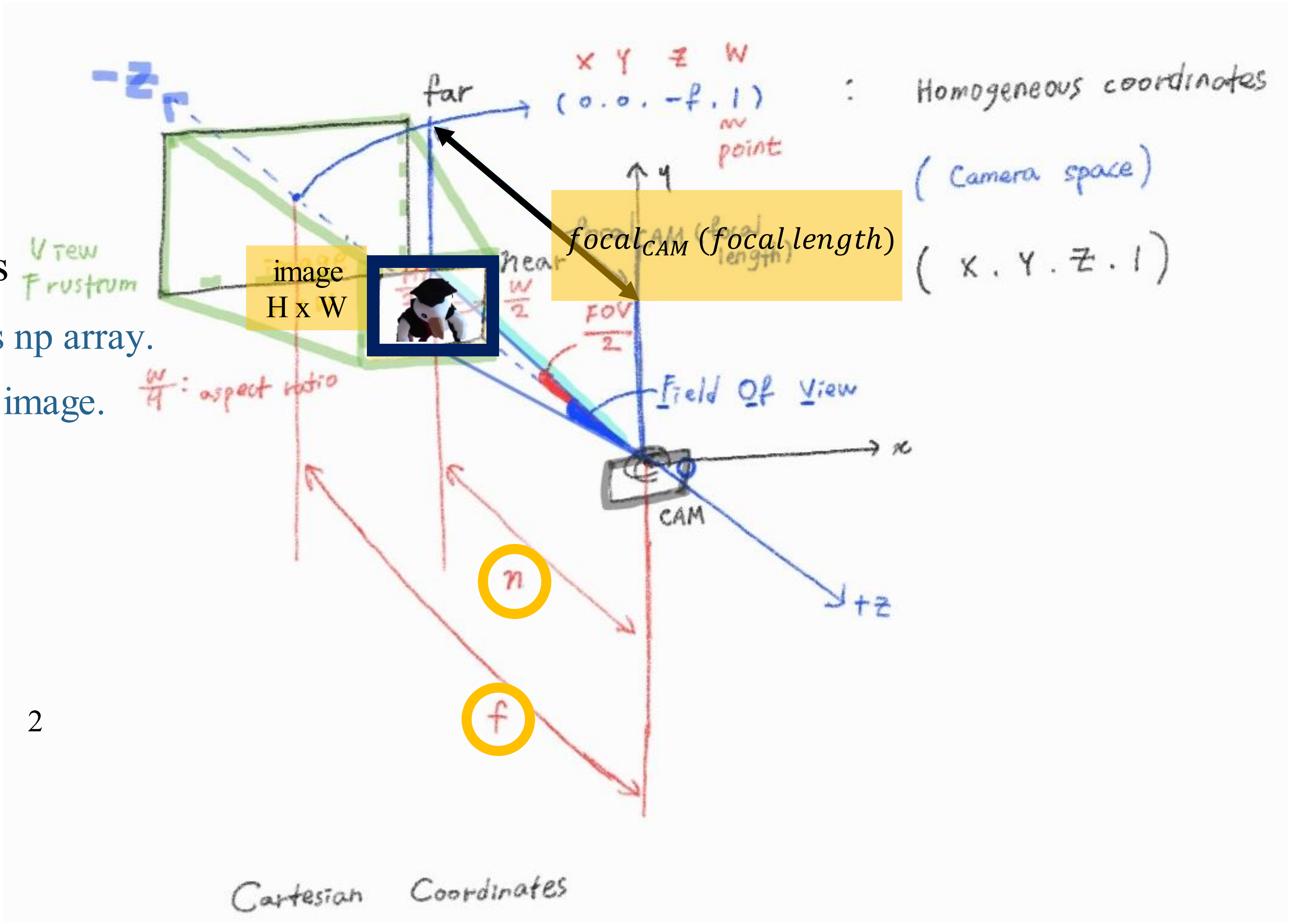
- Getting pose & bound from images
  - COLMAP generates poses_bounds np array.
  - Contains 1 x 17 matrix per 1 input image.

Reshape

$$\begin{array}{cc} & 4 \\ 3 & \left[\begin{array}{c}\text{C-to-W}\\ \text{Affine}\\ \text{transform}\end{array}\right]\end{array}\begin{array}{c}1\\ \begin{bmatrix}H_{image}\\ W_{image}\\ focal\ length\end{bmatrix}\end{array},\ \begin{array}{cc}1 & \\ \begin{bmatrix}n\\ f\end{bmatrix} & 2\end{array}$$

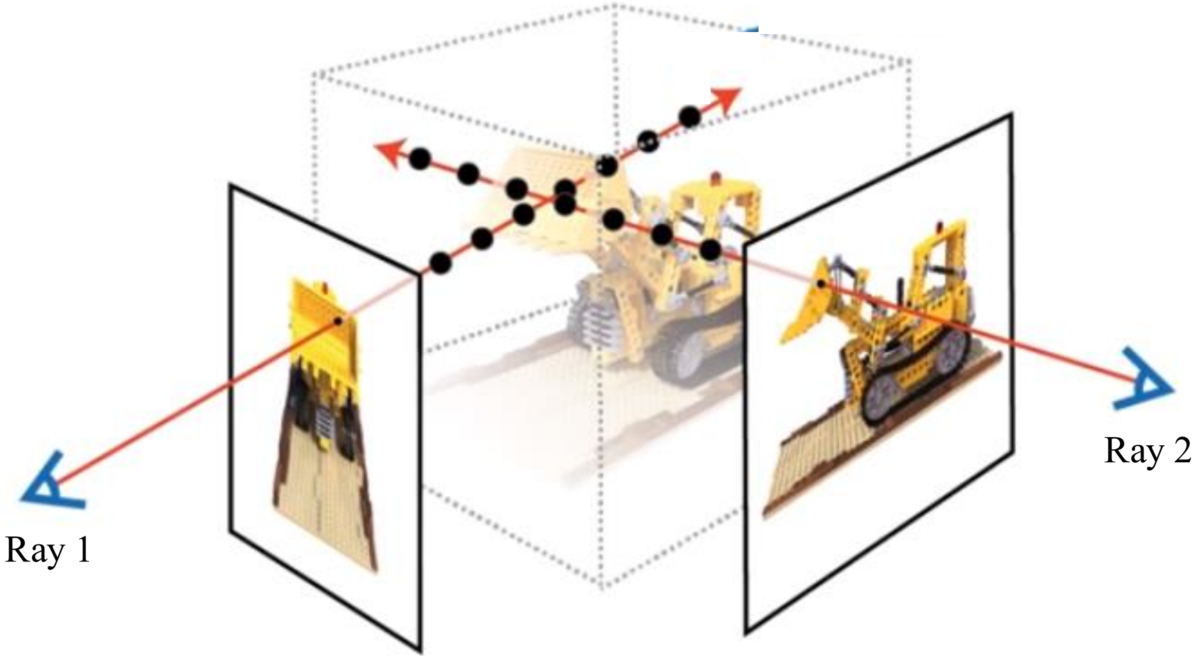
intrinsic parameters

depth values



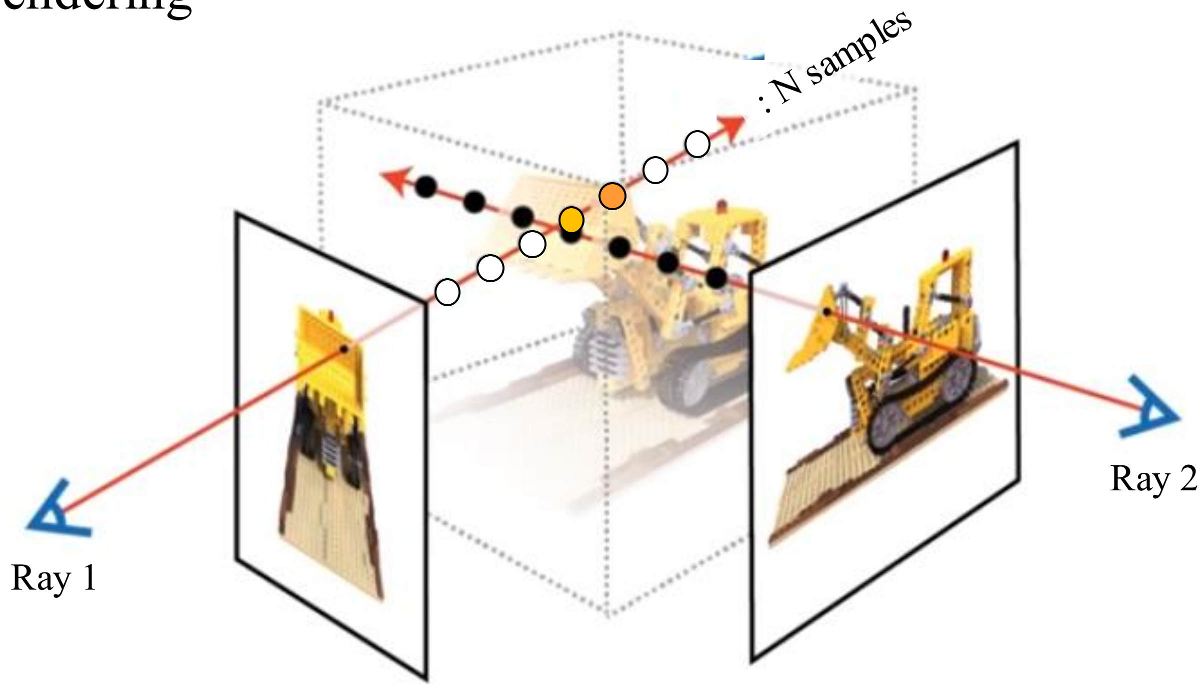
Method

- Generate views with traditional volume rendering
  - 1 pixel of each image = 1 ray



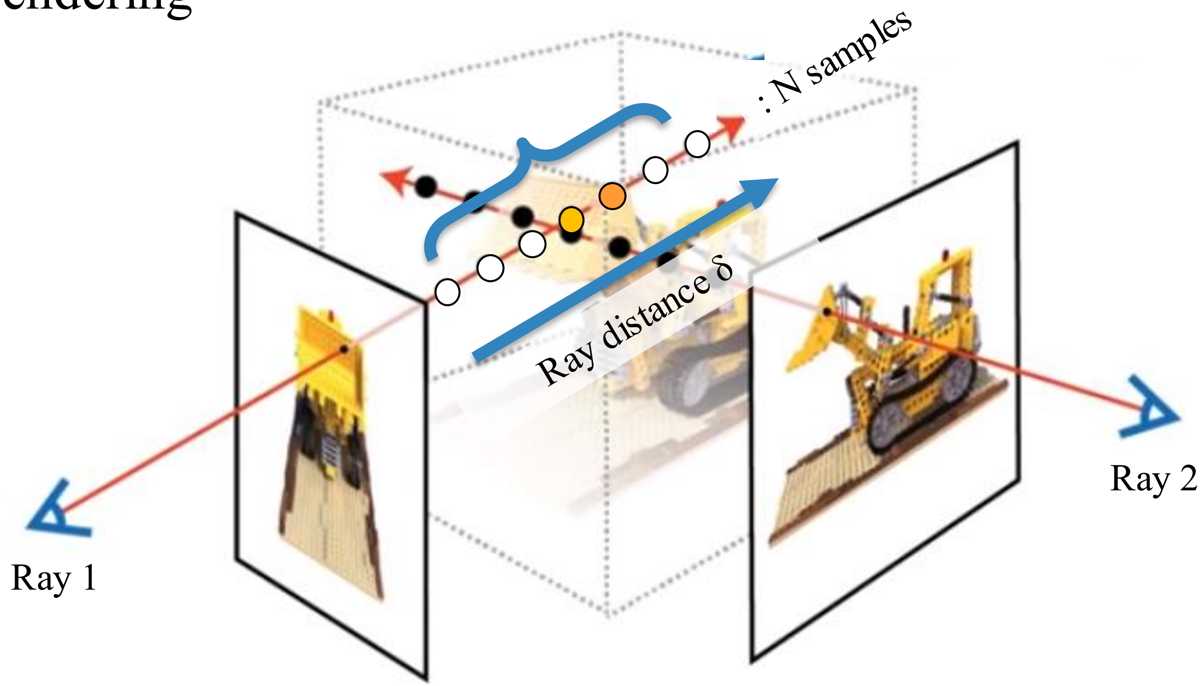
Method

- Generate views with traditional volume rendering
  - 1 pixel of each image = 1 ray



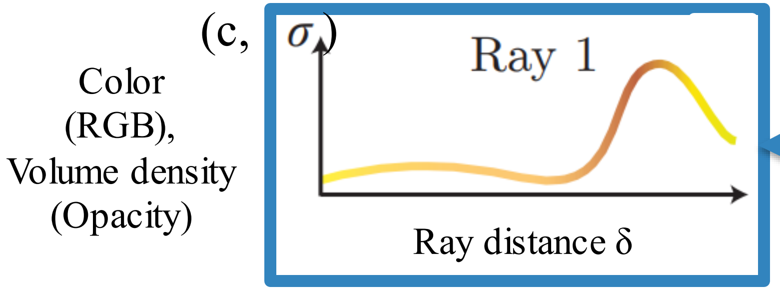
Method

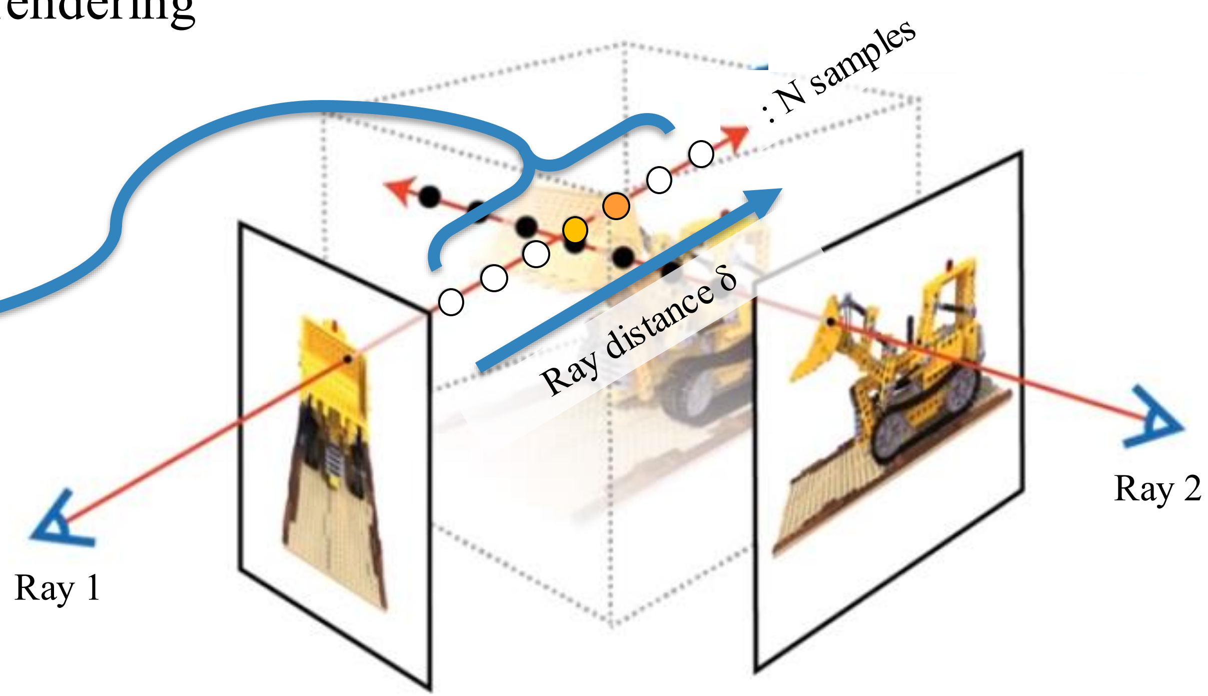
- Generate views with traditional volume rendering
  - 1 pixel of each image = 1 ray



Method

- Generate views with traditional volume rendering
  - 1 pixel of each image = 1 ray



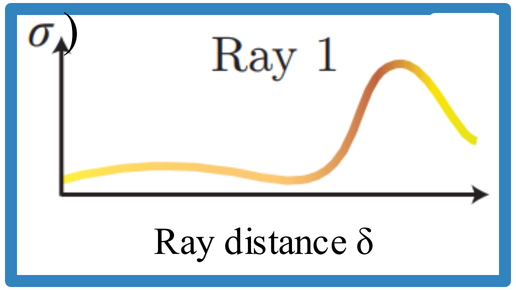


Method

- Generate views with traditional volume rendering
  - Optimize every ray with gradient descent (l2 loss)

Color (RGB), Volume density (Opacity)

(c, σ)



$: C(i)$

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

$$: \hat{C}(i) = \sum_{j=1}^{N} T_j \left(1 - \exp(-\sigma_j \delta_j)\right) c_j$$

$$T_j = \exp\left(-\sum_{k=1}^{j-1} \sigma_j \delta_j\right)$$

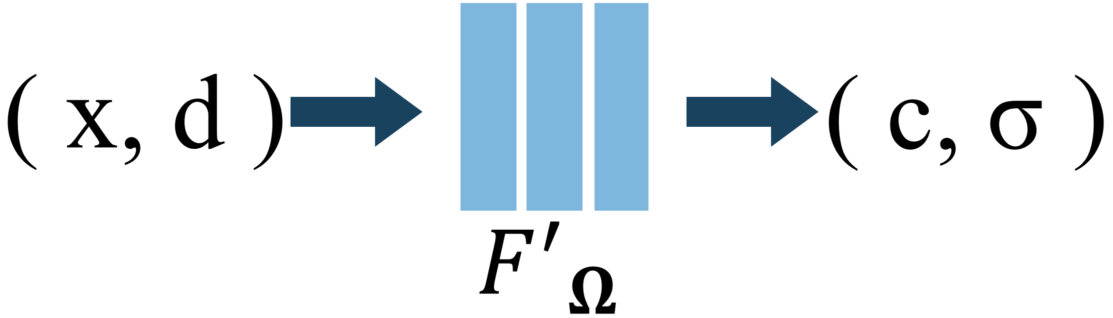
Method

- Optimizing a Neural Radiance Field
  - [1] (Sinusoidal) Positional encoding
  - For F_{Ω} : (x, d) → (c, σ) ,

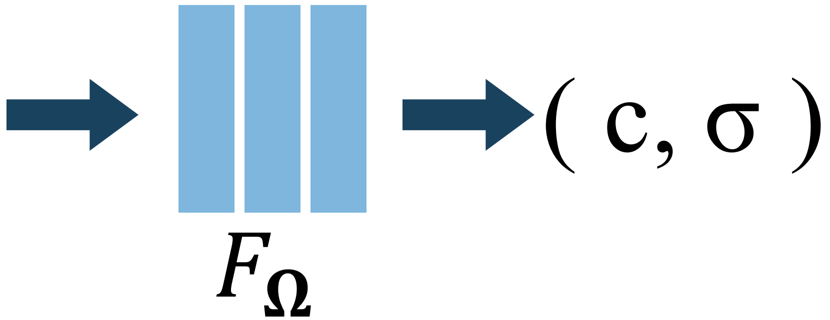
    $F_{\Omega} = F'_{\Omega} \circ \gamma$ (○ is elementwise product)

Method

- Optimizing a Neural Radiance Field
  - [1] (Sinusoidal) Positional encoding
  - For F_{Ω} : (x, d) → (c, σ) ,
    $F_{\Omega} = F'_{\Omega} \circ \gamma$ (∘ is elementwise product)



$$\left(\begin{matrix} \sin(\pi x) & \cos(\pi x) \\ \sin(2\pi x) & \cos(2\pi x) \\ \vdots & \vdots \\ \sin(2^{L-1}\pi x) & \cos(2^{L-1}\pi x) \end{matrix} , \quad \gamma(y) , \gamma(z) \right)$$



Method

- Optimizing a Neural Radiance Field
  - [1] (Sinusoidal) Positional encoding
  - For F_{Ω} : (x, d) $\rightarrow$ (c, σ) ,

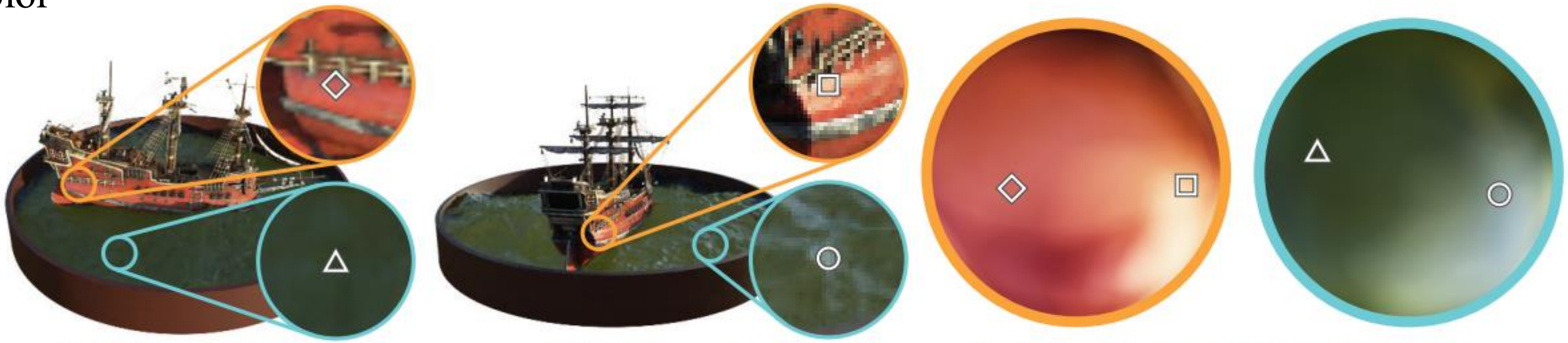
    $F_{\Omega} = F'_{\Omega} \circ \gamma$ (○ is elementwise product)

$$\gamma(p) = (\sin(\pi p), \cos(\pi p), \cdots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)) \quad (L = 10\ for\ \gamma(\mathrm{x}) \text{ and } 4\ for\ \gamma(\mathrm{d}))$$

  - $R^{2L} \leftarrow R$: map into higher dimensional space.

Method

- Optimizing a Neural Radiance Field
  - View-dependent RGB color



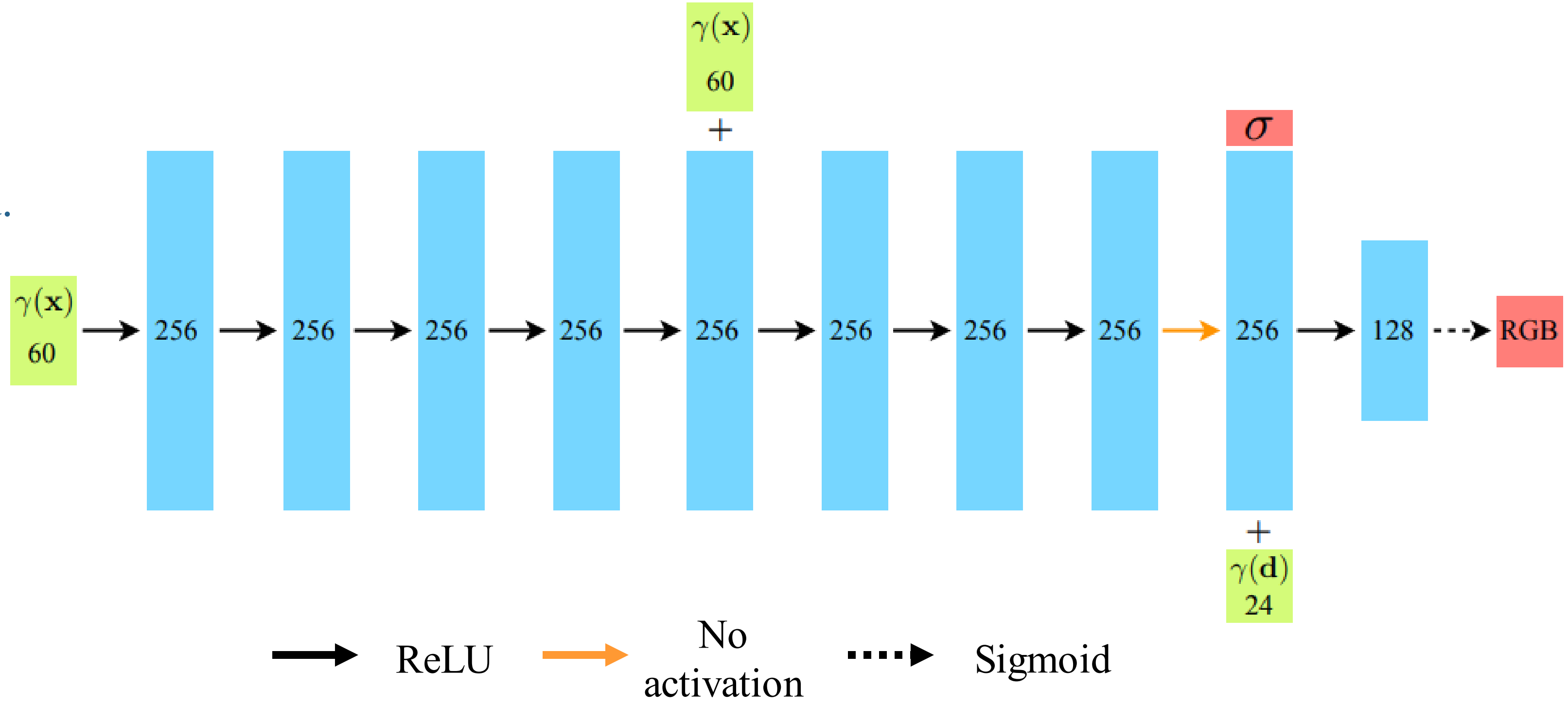
(a) View 1 (b) View 2 (c) Radiance Distributions

- Color (radiance) distributions of the same point on ships viewed by different angles.

Fig. 3: A visualization of view-dependent emitted radiance. Our neural radiance field representation outputs RGB color as a 5D function of both spatial position **x** and viewing direction **d**. Here, we visualize example directional color distributions for two spatial locations in our neural representation of the *Ship* scene. In (a) and (b), we show the appearance of two fixed 3D points from two different camera positions: one on the side of the ship (orange insets) and one on the surface of the water (blue insets). Our method predicts the changing specular appearance of these two 3D points, and in (c) we show how this behavior generalizes continuously across the whole hemisphere of viewing directions.

Method

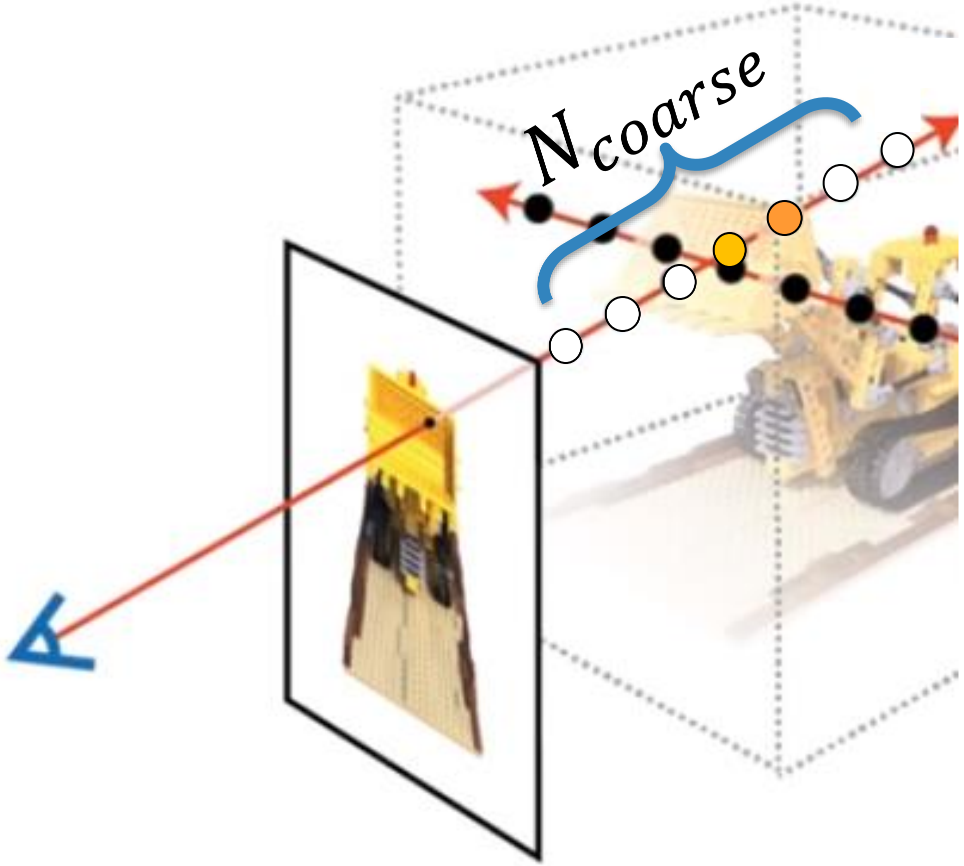
- Optimizing a Neural Radiance Field
  - View-dependent RGB color
  - Predict σ as function of x,
    predict c as function of x and d.



Method

Method

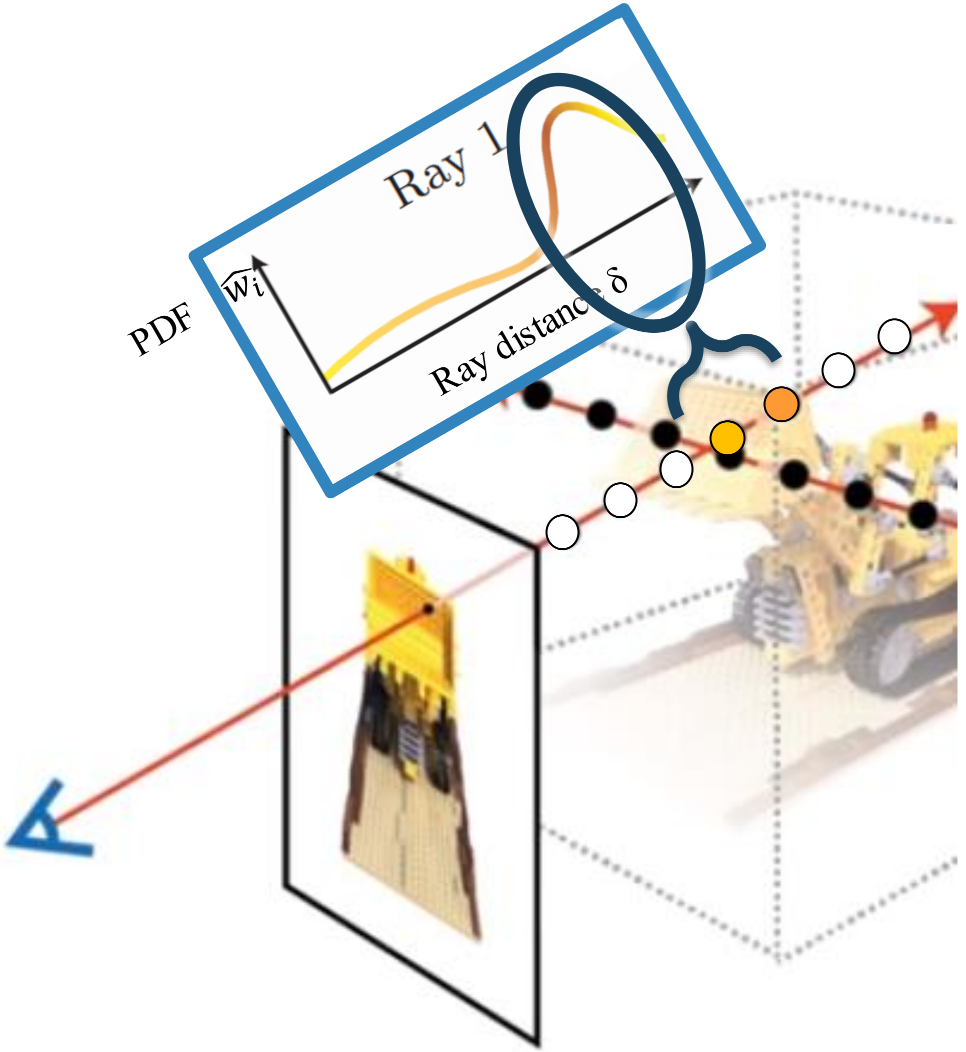
- Optimizing a Neural Radiance Field
  - ³ Hierarchical volume sampling
  - Render by 2 networks : coarse & fine



Method

- Optimizing a Neural Radiance Field
  - [3] Hierarchical volume sampling
  - Render by 2 networks : $coarse$ & $fine$
  - Get Probability Distribution Function by normalizing weights from N_{coarse}.

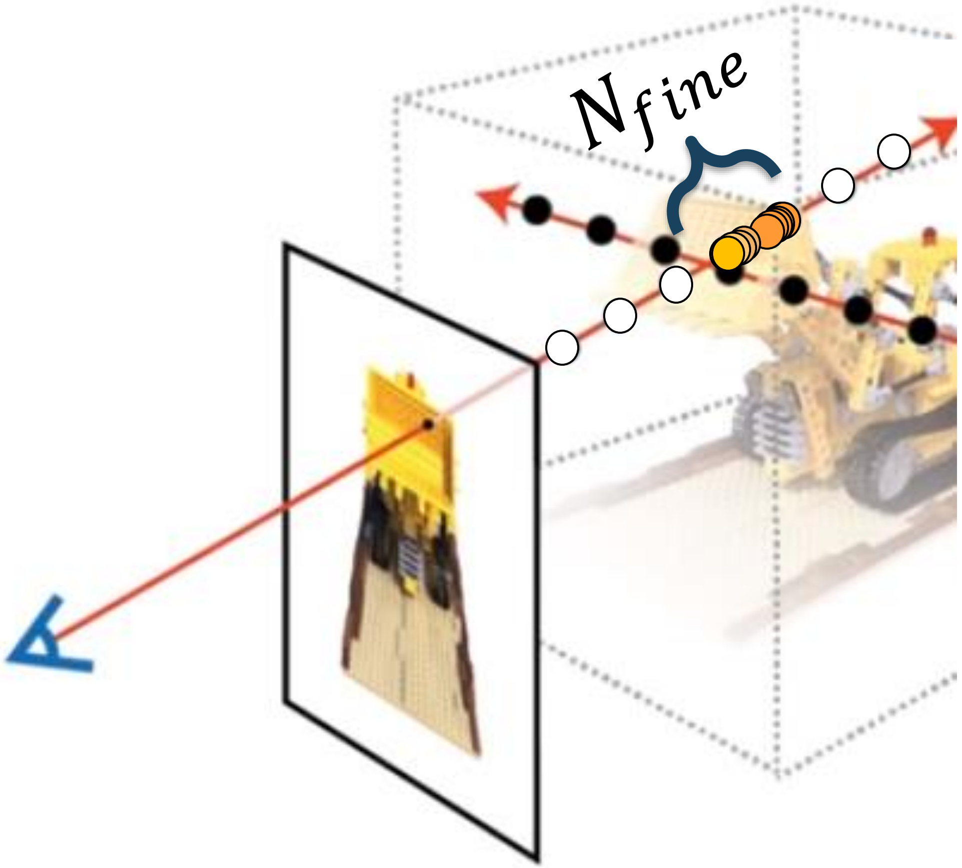
  - Given the output of N_{coarse} , produce more informed sample points N_{fine}.
  - $N_{fine} = N_{importance}$
  - Finally, use all $N_{coarse+fine}$ samples.



Method

- Optimizing a Neural Radiance Field
  - [3] Hierarchical volume sampling
  - Render by 2 networks : $coarse$ & $fine$
  - Get Probability Distribution Function by normalizing weights from N_{coarse}.

  - Given the output of N_{coarse} , produce more informed sample points N_{fine}.
  - $N_{fine} = N_{importance}$
  - Finally, use all $N_{coarse+fine}$ samples.



Results

- Ablation studies
  - Effect of 3 optimization methods

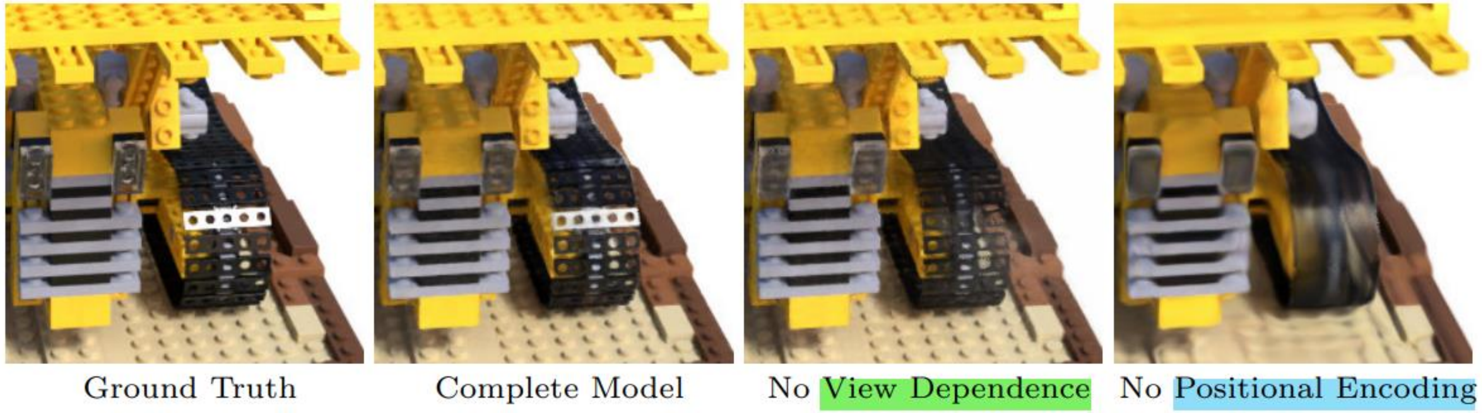
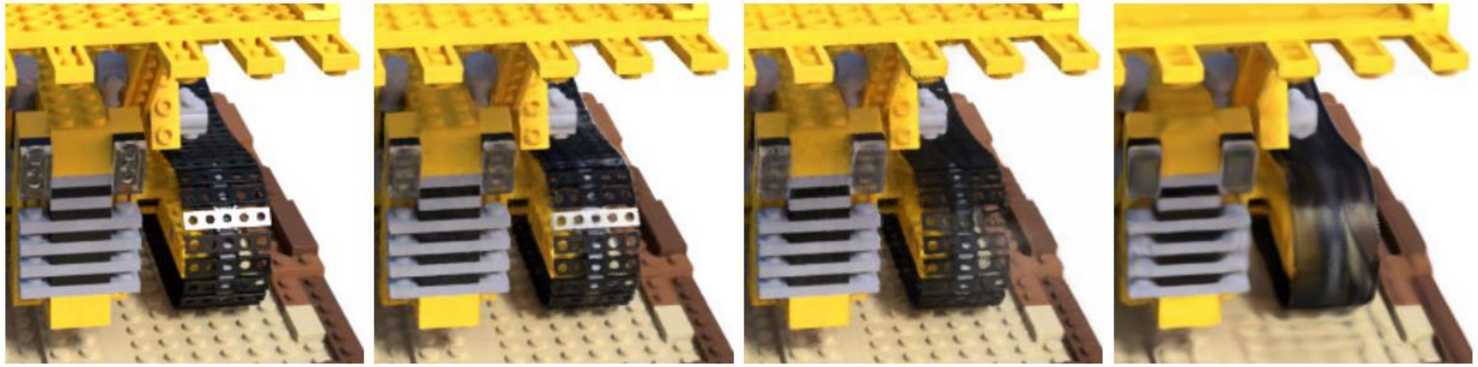


Fig. 4: Here we visualize how our full model benefits from representing view-dependent emitted radiance and from passing our input coordinates through a high-frequency positional encoding. Removing view dependence prevents the model from recreating the specular reflection on the bulldozer tread. Removing the positional encoding drastically decreases the model's ability to represent high frequency geometry and texture, resulting in an oversmoothed appearance.

Results

- Ablation studies
  - Effect of 3 optimization methods

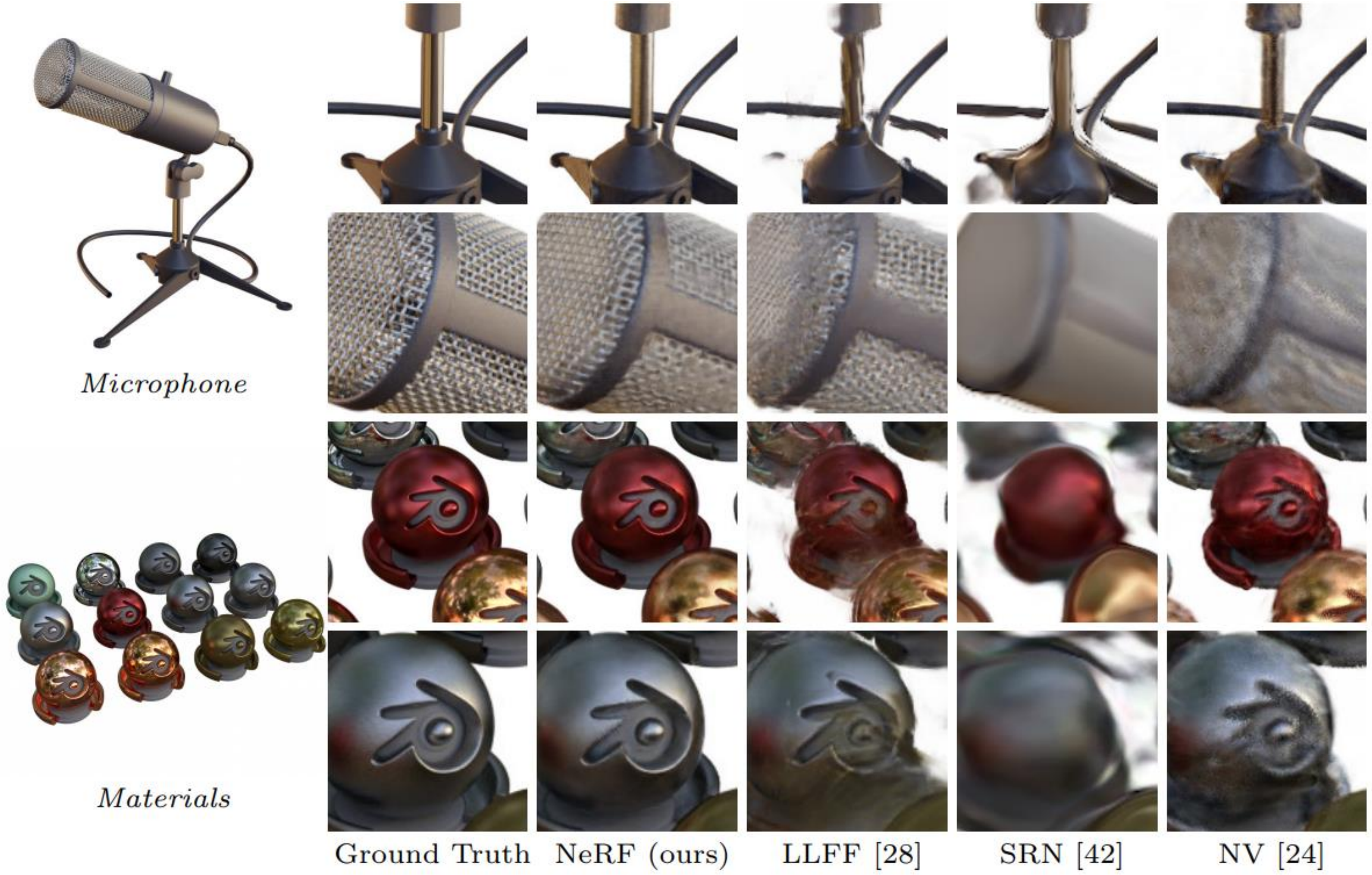


Ground Truth | Complete Model | No View Dependence | No Positional Encoding

	Input	#Im.	L	(N_c, N_f)	PSNR↑	SSIM↑	LPIPS↓
No PE, VD, H	xyz	100	-	(256, -)	26.67	0.906	0.136
No Pos. Encoding	$xyz\theta\phi$	100	-	(64, 128)	28.77	0.924	0.108
No View Dependence	xyz	100	10	(64, 128)	27.66	0.925	0.117
No Hierarchical	$xyz\theta\phi$	100	10	(256, -)	30.06	0.938	0.109
Complete Model	$xyz\theta\phi$	100	10	(64, 128)	**31.01**	**0.947**	**0.081**

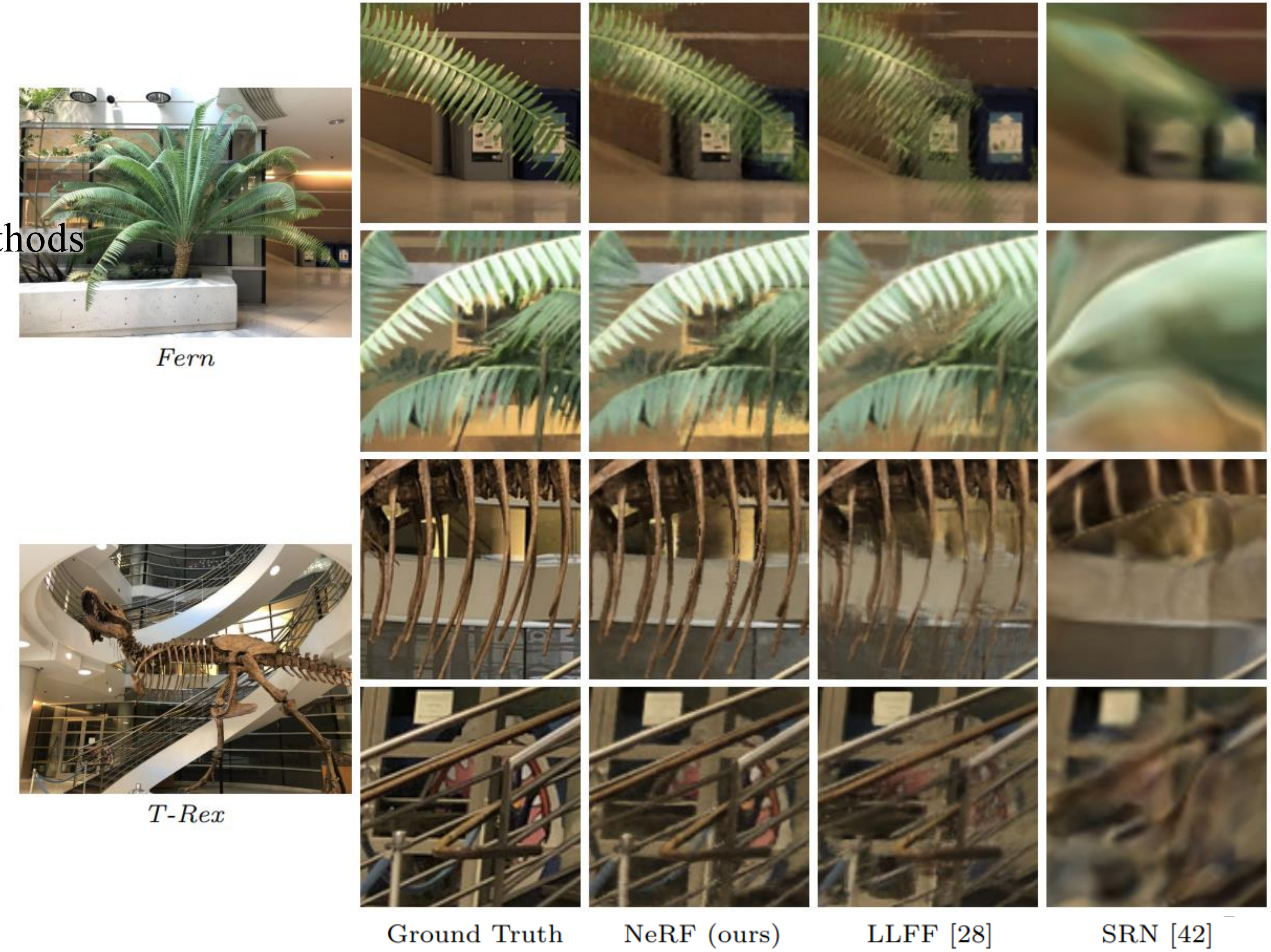
Results

- Comparisons with previous methods
  - Synthetic dataset



Results

- Comparisons with previous methods
  - Real world scenes



Results

Going forward

- Decrease training and inference time (currently 30 sec per high-resolution frame)
  - Mip-NeRF (A Multiscale Representation for Anti-Aliasing Neural Radiance Fields)
    https://jonbarron.info/mipnerf/
- Disentangle more graphics attributes to allow additional applications, such as relighting
  - NeRV (Neural Reflectance and Visibility Fields for Relighting and View Synthesis)
    https://pratulsrinivasan.github.io/nerv/
- Generalize to more than one scene without training from scratch and requiring fewer input images

감사합니다