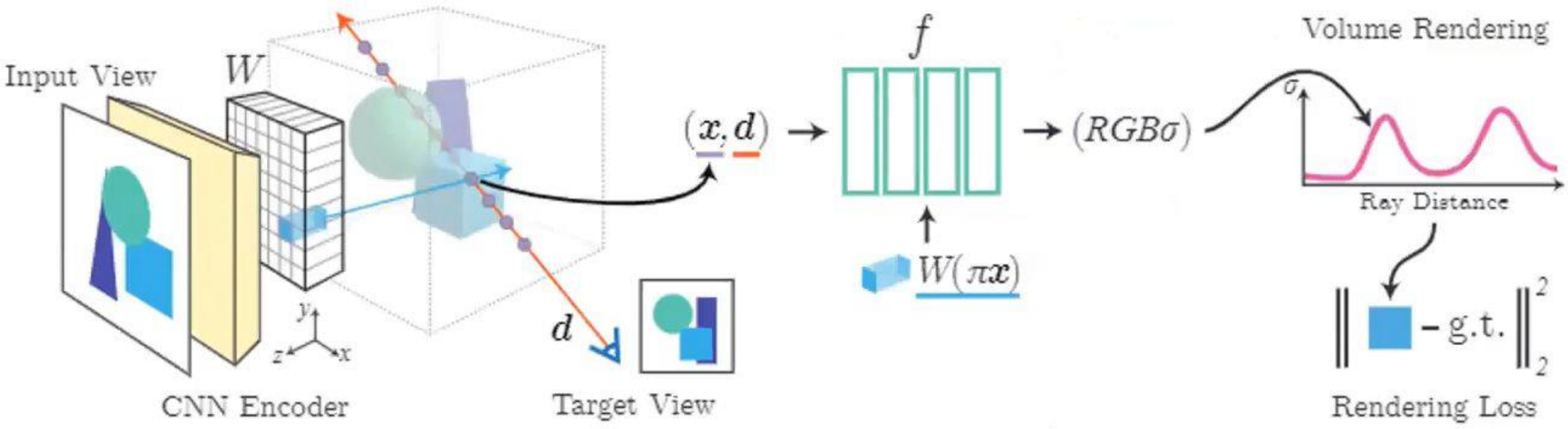
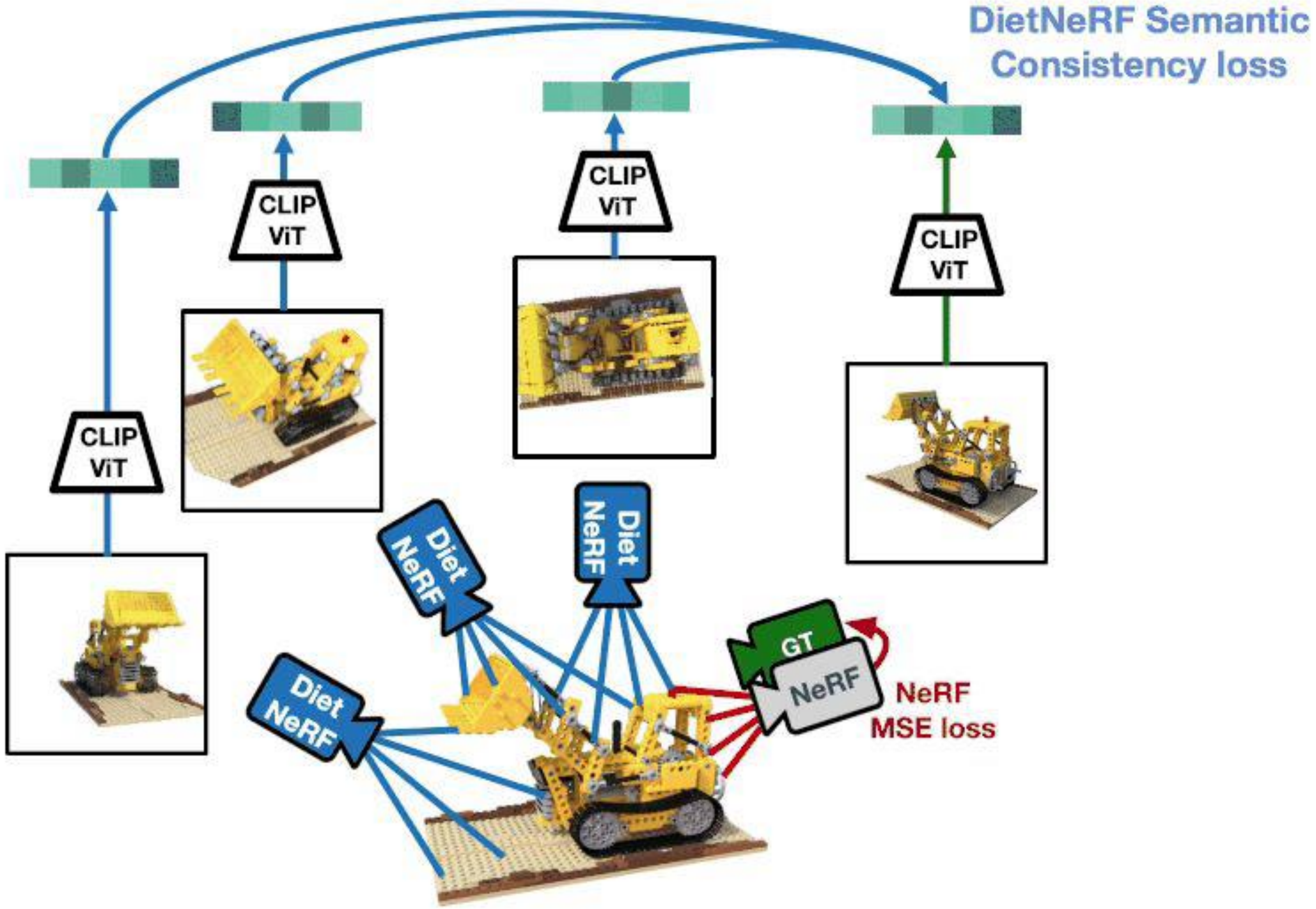
NeX & Ref-NeRF

20214609 Jaemin Cho

Recap: pixelNeRF



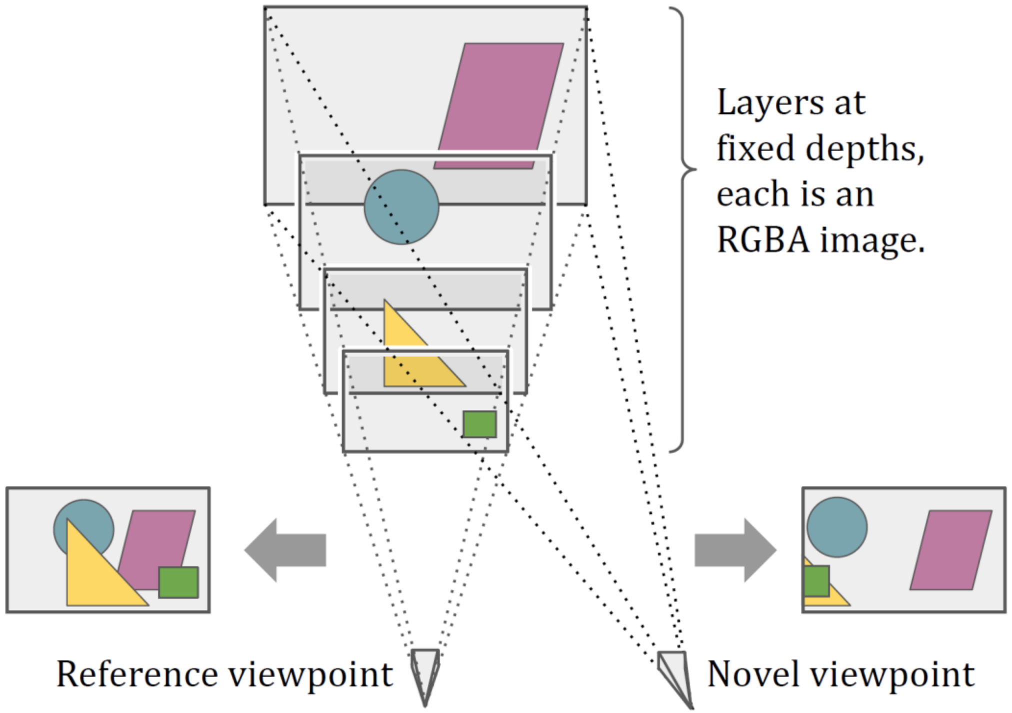
Recap: DietNeRF

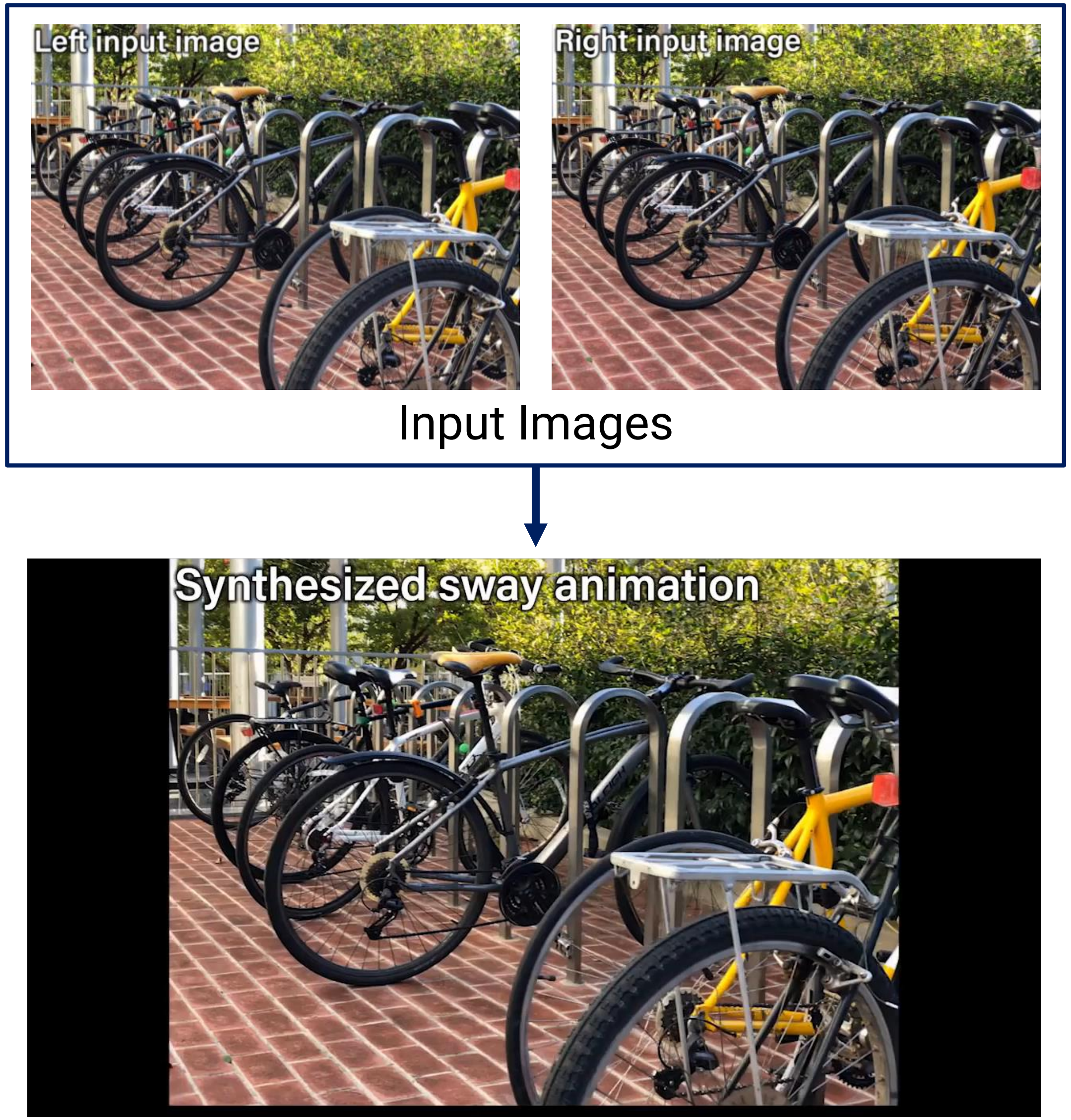


NeX: Real-time View Synthesis with Neural Basis Expansion

Suttisak Wizadwongsa et al., *CVPR*, 2021

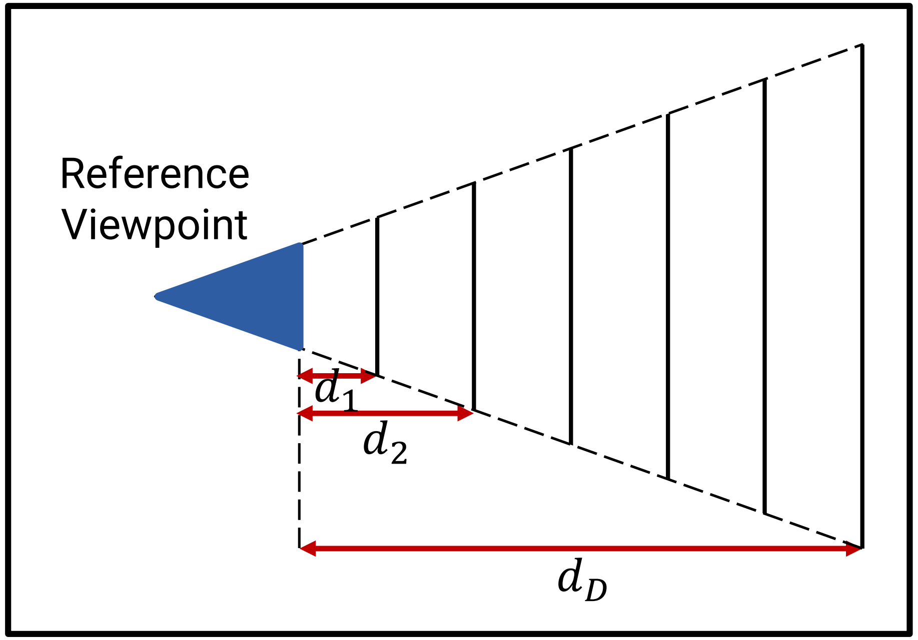
Original Multiplane Image (MPI)



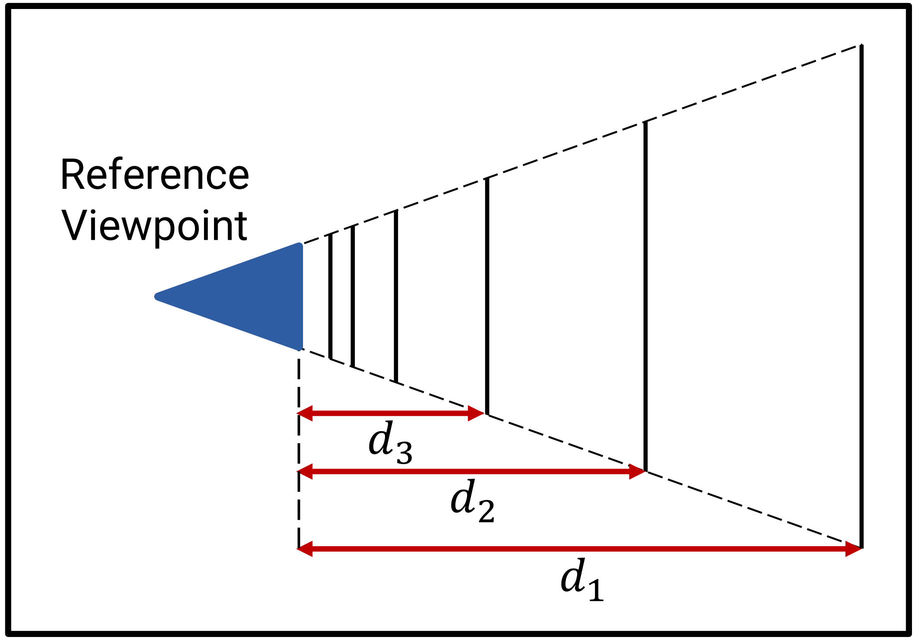


Original MPI

- Equidistant placed for depth space : $d_2 - d_1 = d_3 - d_2 = \cdots = d_D - d_{D-1}$
- Equidistant placed for **inverse depth space** : $\frac{1}{d_2} - \frac{1}{d_1} = \frac{1}{d_3} - \frac{1}{d_2} = \cdots = \frac{1}{d_D} - \frac{1}{d_{D-1}}$
  (= equidistant placed for **disparity space**)



For depth space



For inverse depth space

Original MPI: Rendering Process

i^{th} plane of MPI = alpha($\alpha_i \in R^{H \times W \times 1}$) + RGB color($c_i \in R^{H \times W \times 3}$)

$\mathrm{MPI} = A(\{\alpha_1, \alpha_2, \dots, \alpha_D\}) + \mathrm{C}(\{c_1, c_2, \dots, c_D\})$

1. Make new MPI by **warping** all planes to the **target view**

   $\text{new MPI} = \mathrm{W(A)} + \mathrm{W(C)}, \quad \text{W is a homography warping function}$

2. Render image in new MPI

   $\text{image in vew view } (\hat{I}) = \mathrm{O(W(A), W(C))}$

$$O(A,C) = \sum_{d=1}^{D} c_d T_d(A)\,, \;\; T_d(A) = \alpha_d \prod_{i=d+1}^{D} (1 - \alpha_i)$$

Original MPI: Conclusion

- Advantage
  - Express occlusion, thin structure, or planar reflection with **simple structure**
  - **Real-time** rendering

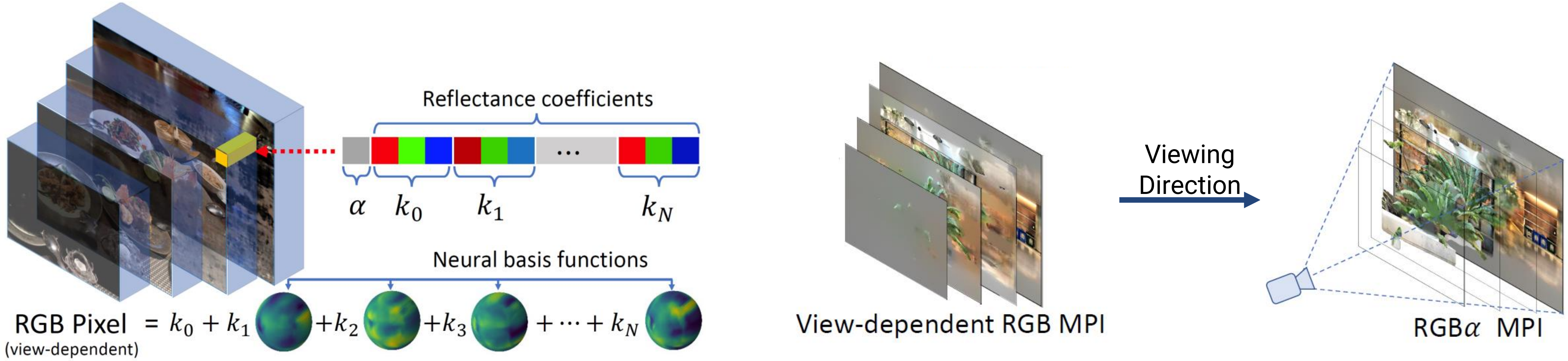
- Limitation - **RGBα** representation
  - Each pixel has **constant value** regardless of **viewing direction**
  - Only works well on the **diffuse surfaces**

Approach

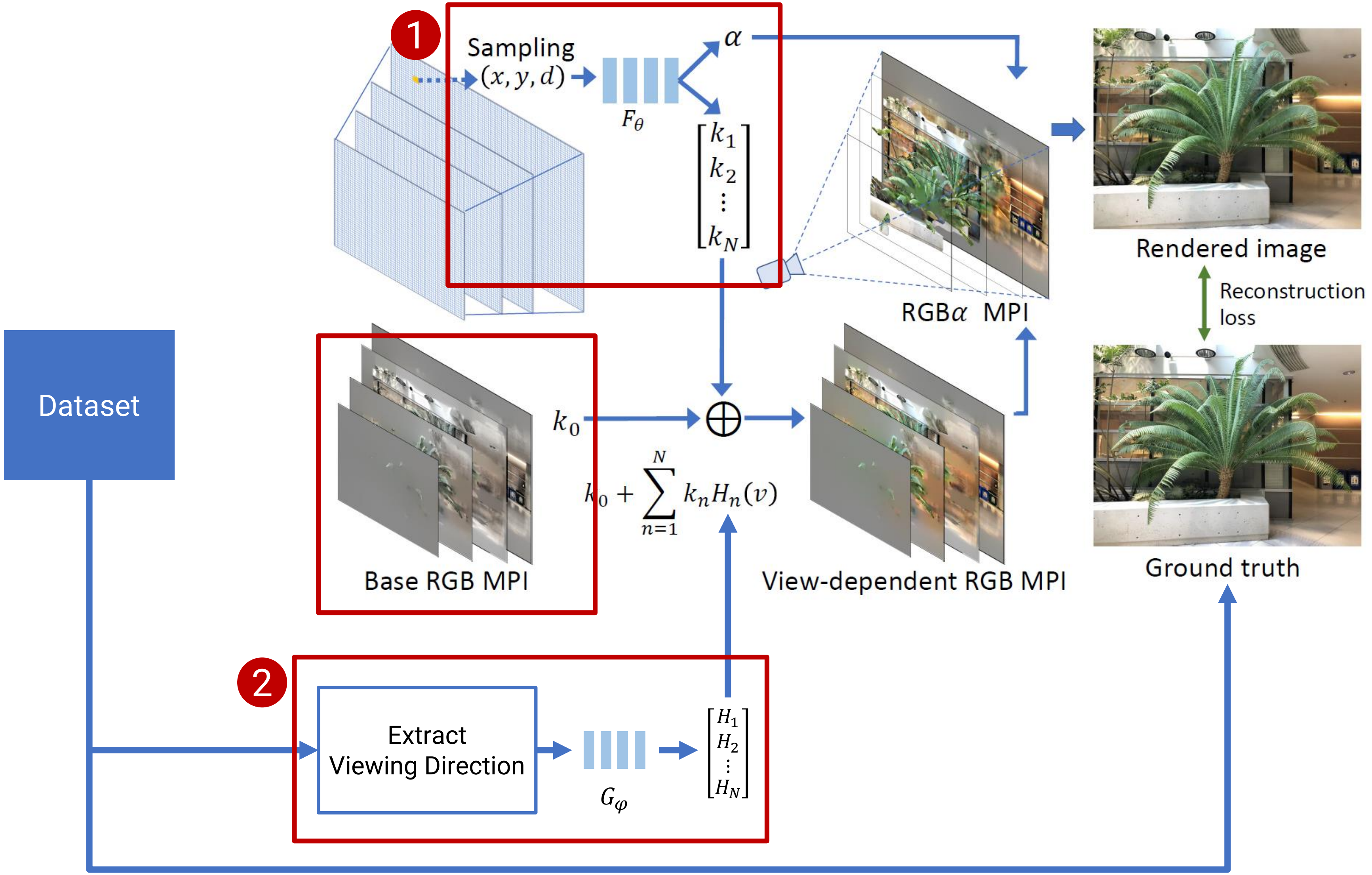
- Allow for **view-dependent** modeling in MPI
  - Parameterizing each color value as a function of the **viewing direction**

- Traditional RGB MPI : $each\ pixel\ - (c, \alpha)$
- View-dependent RGB MPI : $each\ pixel\ - (k_0, k_1, \dots, k_N, \alpha)$

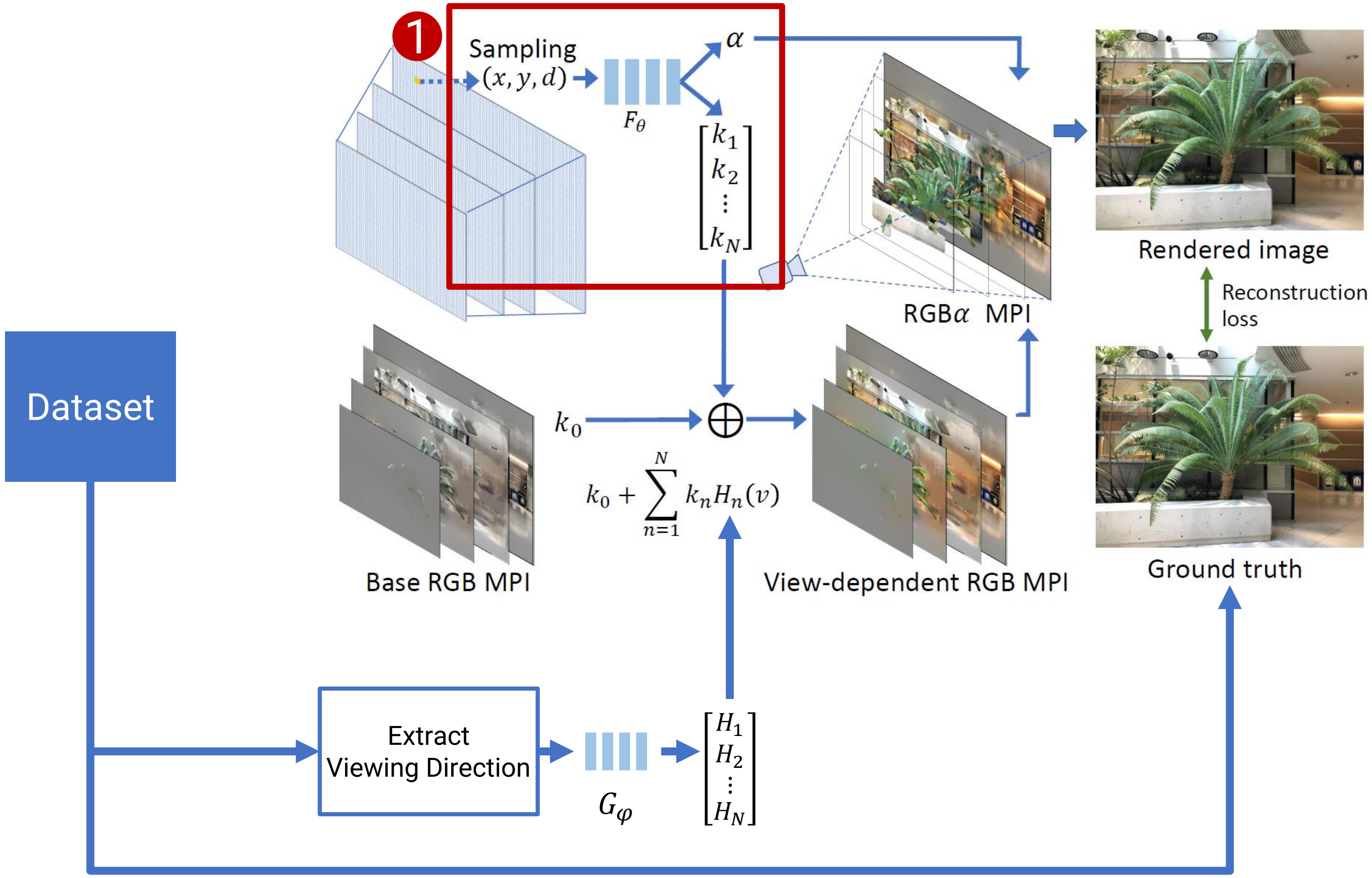
$$C^P(v) = k_0^P + \sum_{n=1}^{N} k_n^P H_n(v), \quad H_n(v): (v): R^3 \rightarrow R$$



Training Network

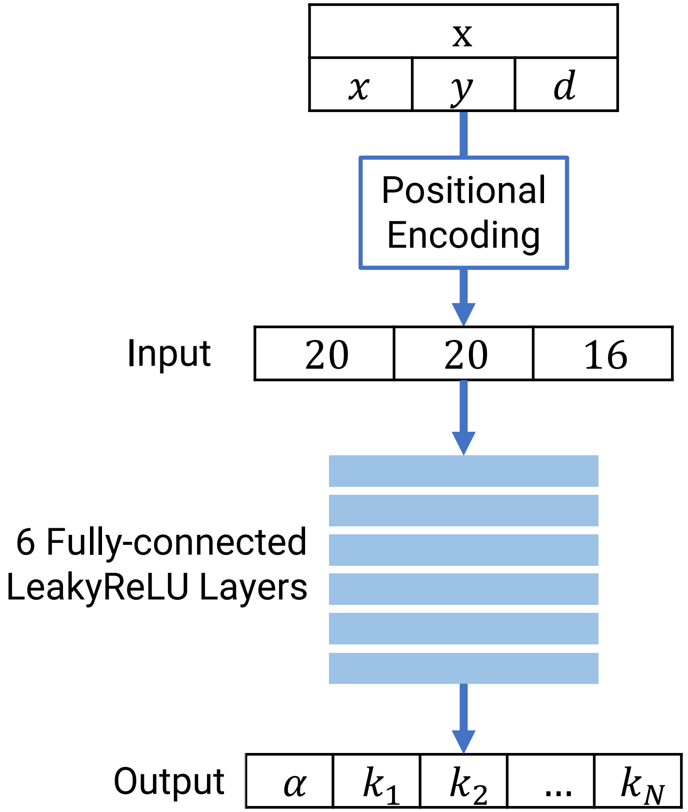


Training Network - Details

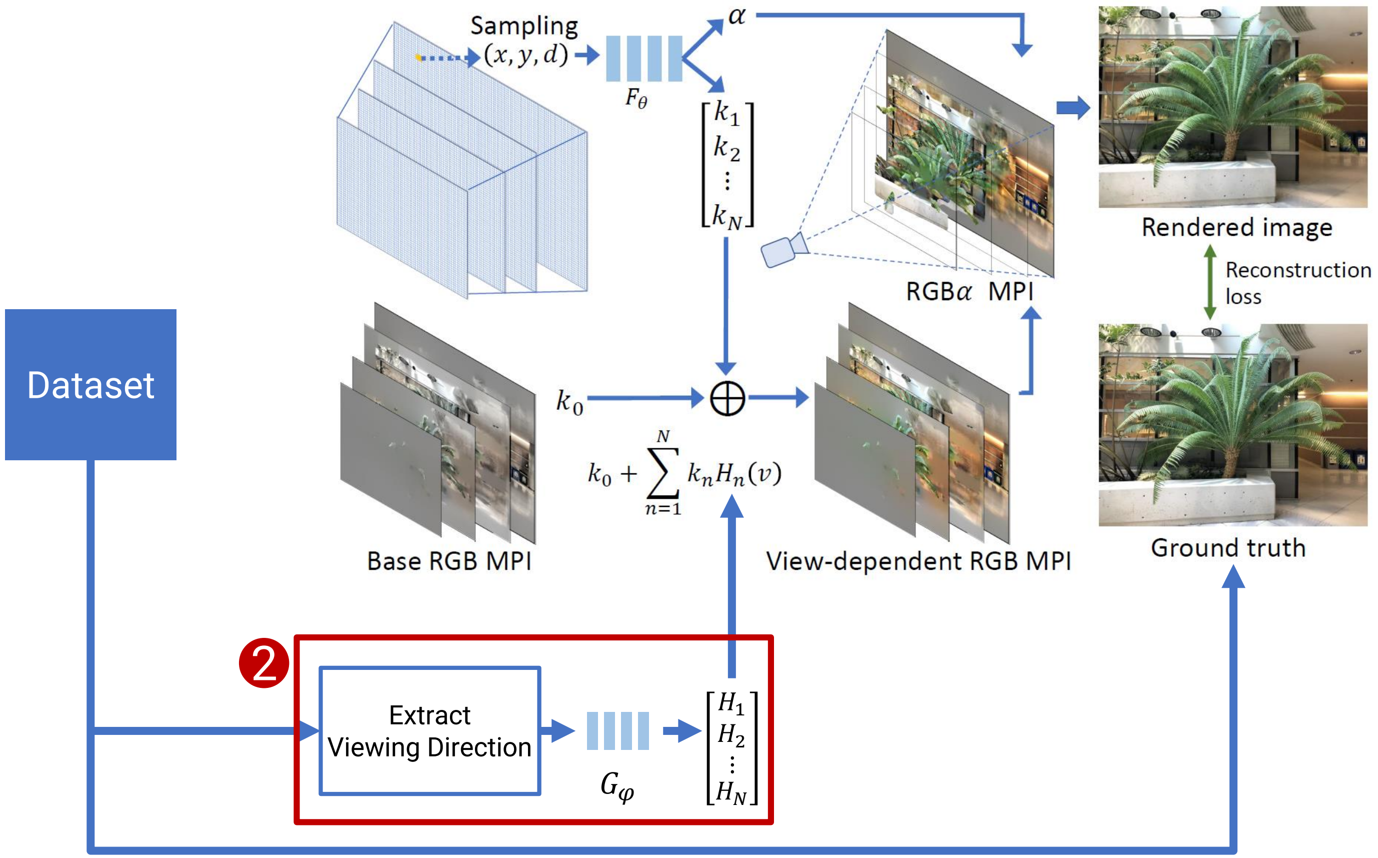


$F_\theta: (\mathrm{x}) \rightarrow (\alpha, k_1, k_2, \ldots, k_N)$

$\mathrm{x}\ is\ a\ pixel\ (x, y) at\ plane\ d : (x, y, d)$



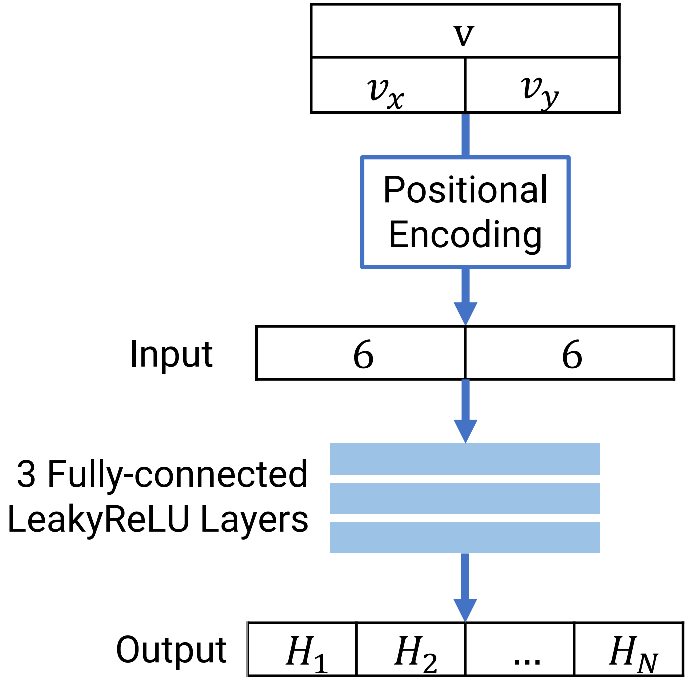
Training Network - Details



$G_\varphi : (\mathrm{v}) \rightarrow (H_1, H_2, \ldots, H_N)$

v *is the normalized viewing direction*: (v_x, v_y)

$v_z = \sqrt{1 - (v_x^2 + v_y^2)}$



Implicit-Explicit Modeling Strategy

- Base Color (k_0) Explicitly
  - Using positional encoding, MPI still produces **blurry results**
  - Reproducing **detail** and leads to **sharper results**

- Coefficient Sharing
  - N+1 coefficients for all pixels for all D planes → **expensive** for training and rendering
  - M planes **share** the same coefficients (not alpha)
  - Significant gain in **speed and model compactness** without **degradation in the visual quality**
  - 192 planes with M = 12

1^{st} to M^{th} planes : share coefficients set $\{k_0, k_1, \dots, k_N\}$

$M+1^{th}$ to $2M^{th}$ planes : share coefficients set $\{k_0, k_1, \dots, k_N\}$

$\vdots$

Loss Functions

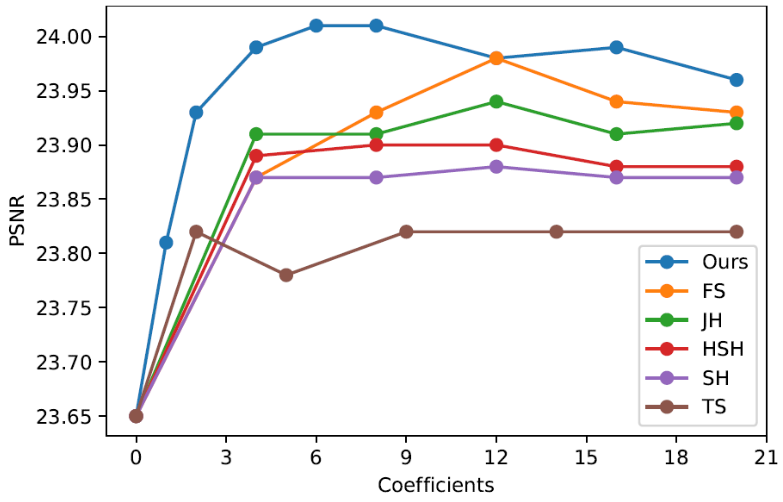
I_i : *rendered output image*, $\hat{I}_i$: *ground − truth image*

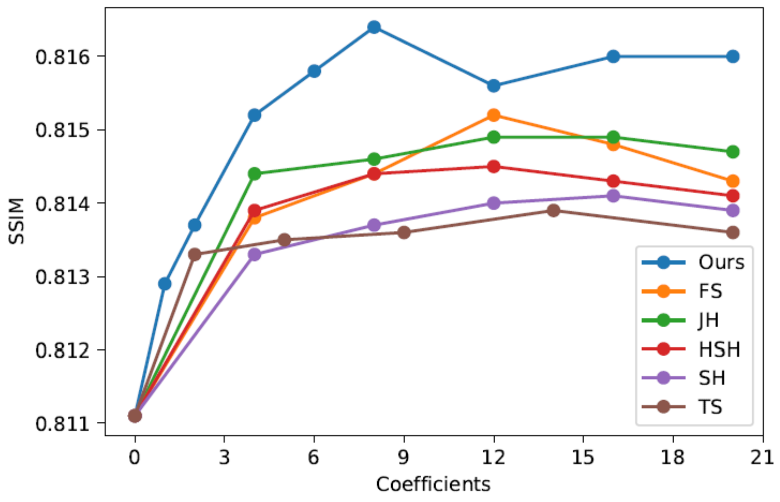
$$L_{total} = L_{rec}(\hat{I}_i, I_i) + \gamma TV(K_0)$$

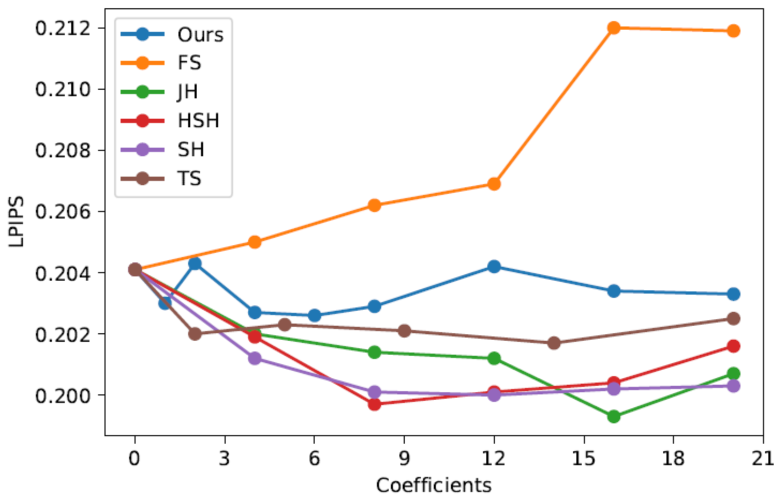
- Reconstruction Loss (L_{rec})
  - MSE Loss + Gradient Loss
  - $L_{rec}(\hat{I}_i, I_i) = \|\hat{I}_i - I_i\|^2 + \omega\|\nabla\hat{I}_i - \nabla I_i\|$
- Total Variation (TV)

Experiment Results

- Number of Basis Coefficients
  - FS : Fourier Series
  - JH : Jacobi Spherical Harmonics
  - HSH : Hemispherical Harmonics
  - SH : Spherical Harmonics
  - TS : Taylor Series







Experiment Results

- Evaluation on different modeling strategies
  - Alpha transparency (A)
  - Base color (K_0)
  - View-dependent coefficients ($K_1, \ldots, K_n$)

Method			Metric		
A	K_0	$K_1, \ldots, K_n$	PSNR ↑	SSIM ↑	LPIPS ↓
Ex	Ex	Ex	24.57	0.857	0.292
Ex	Ex	Im	24.47	0.854	0.300
Ex	Im	Ex	24.55	0.857	0.296
Ex	Im	Im	24.44	0.854	0.302
Im	Ex	Ex	26.30	0.901	0.204
Im	**Ex**	**Im**	**26.32**	**0.904**	**0.202**
Im	Im	Ex	25.82	0.883	0.279
Im	Im	Im	25.63	0.878	0.301

Experiment Results

- Real forward-facing dataset

Method	PSNR ↑	SSIM ↑	LPIPS ↓
SRN [34]	21.82	0.744	0.464
LLFF [21]	24.41	0.863	0.211
NeRF [22]	26.76	0.883	0.246
NeX (Ours)	**27.26**	**0.904**	**0.178**

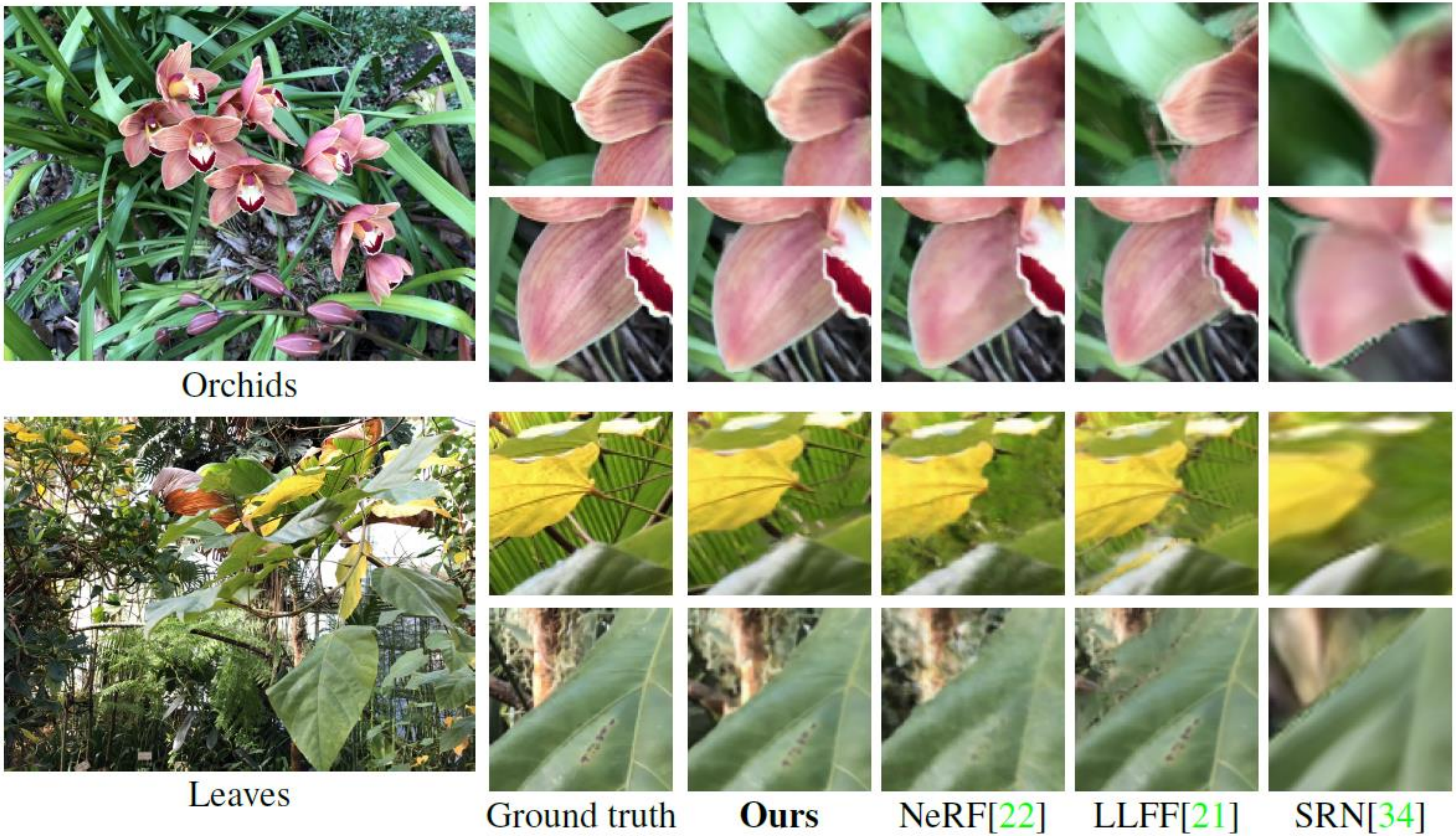
- Shiny dataset

Method	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [22]	25.60	0.851	0.259
NeX (Ours)	**26.45**	**0.890**	**0.165**

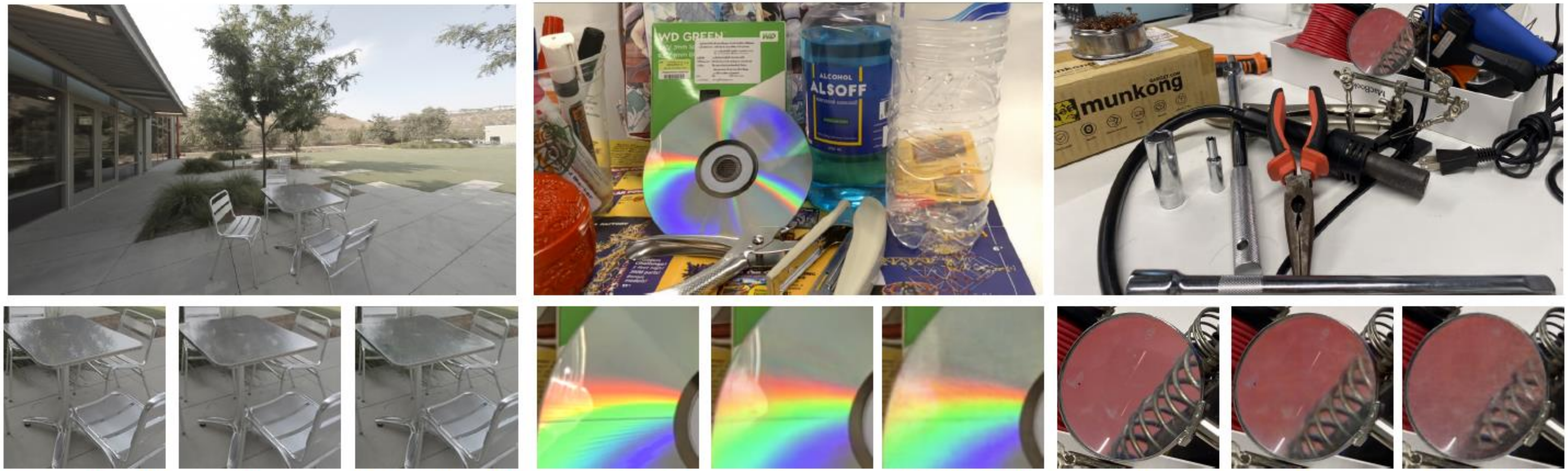
- Space dataset

Method	PSNR↑	SSIM ↑	LPIPS ↓
Soft3D [24]	31.57	0.964	0.126
Deepview[6]	31.60	0.978	0.085
NeX (Ours)	**35.84**	**0.985**	**0.083**

Experiment Results



Experiment Results



Ground truth | **Ours** | DeepView[6] | Ground truth | **Ours** | NeRF[22] | Ground truth | **Ours** | NeRF[22]

(a) Spaces dataset: Scene 056 (b) Shiny dataset: CD (c) Shiny dataset: Tools

Experiment Results

- Need **long time** and high number of input views for training
- Cannot completely synthesize view dependent effect (ex. **sharp highlights**, or **refraction**)

Ground truth **Ours** Ground truth **Ours**

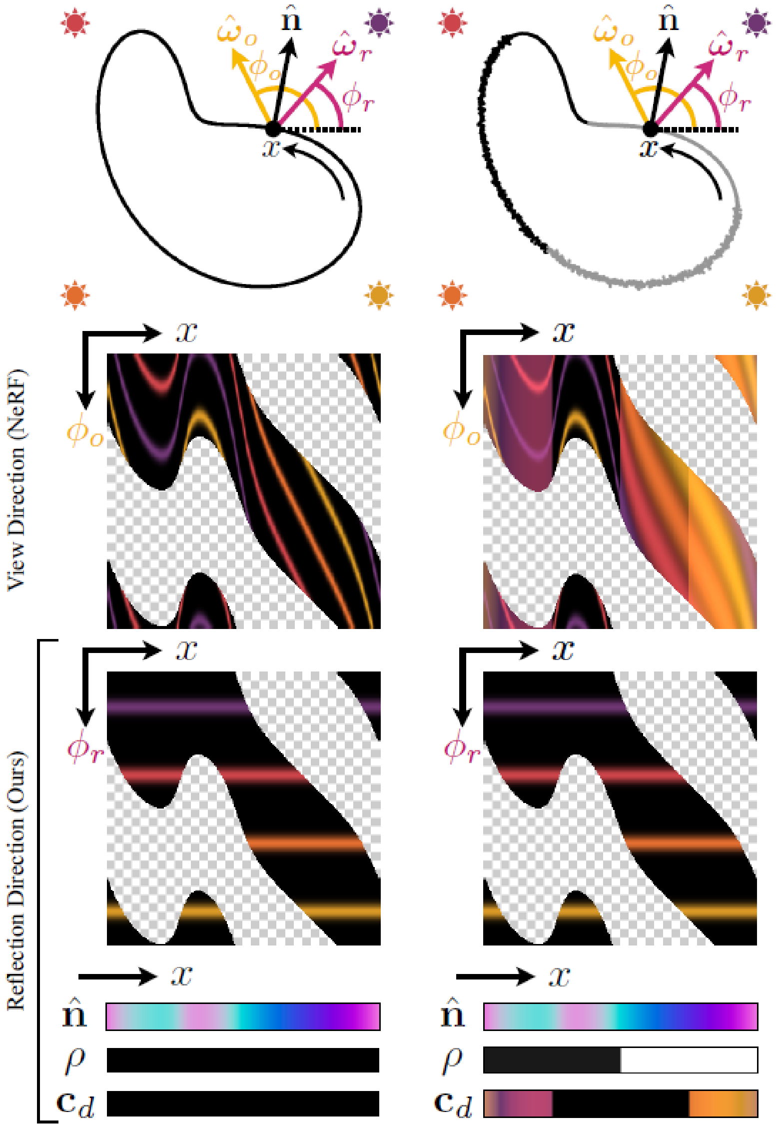
Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields

Dor Verbin et al., *CVPR*, 2022

Approach

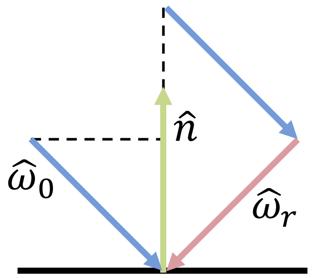
Original NeRF

- Using viewing direction as a input of outgoing radiance function
  - Poorly suited for interpolation
- Fake specular reflection
  - Emitters in side the object
  - Objects with semitransparent or foggy shells

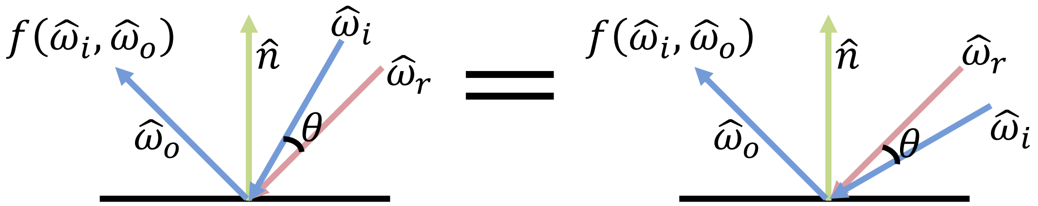


Reflection Direction Parameterization

- Convert view direction to reflection of the view direction
  - $\widehat{\omega}_r = 2(\widehat{\omega}_0 \cdot \hat{n})\hat{n} - \widehat{\omega}_0$
  - $\widehat{\omega}_0$: unit vector from camera

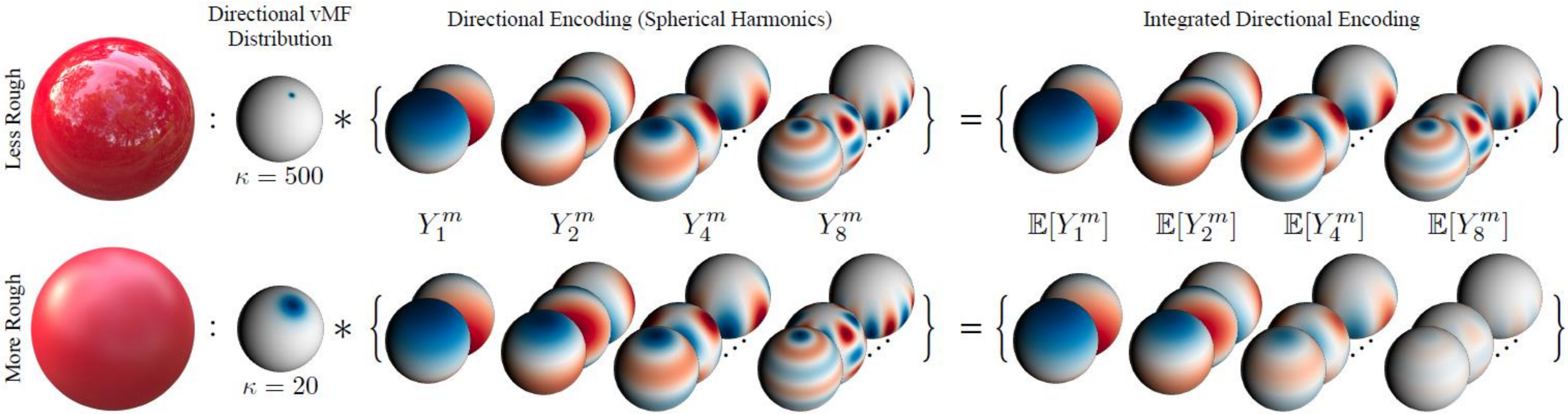


- BRDF : rotationally-symmetric about reflected view direction
  - $f(\widehat{\omega}_i, \widehat{\omega}_o) = p(\widehat{\omega}_r \cdot \widehat{\omega}_i)\ for\ some\ function\ p$
  - Neglecting interreflections and self-occlusions



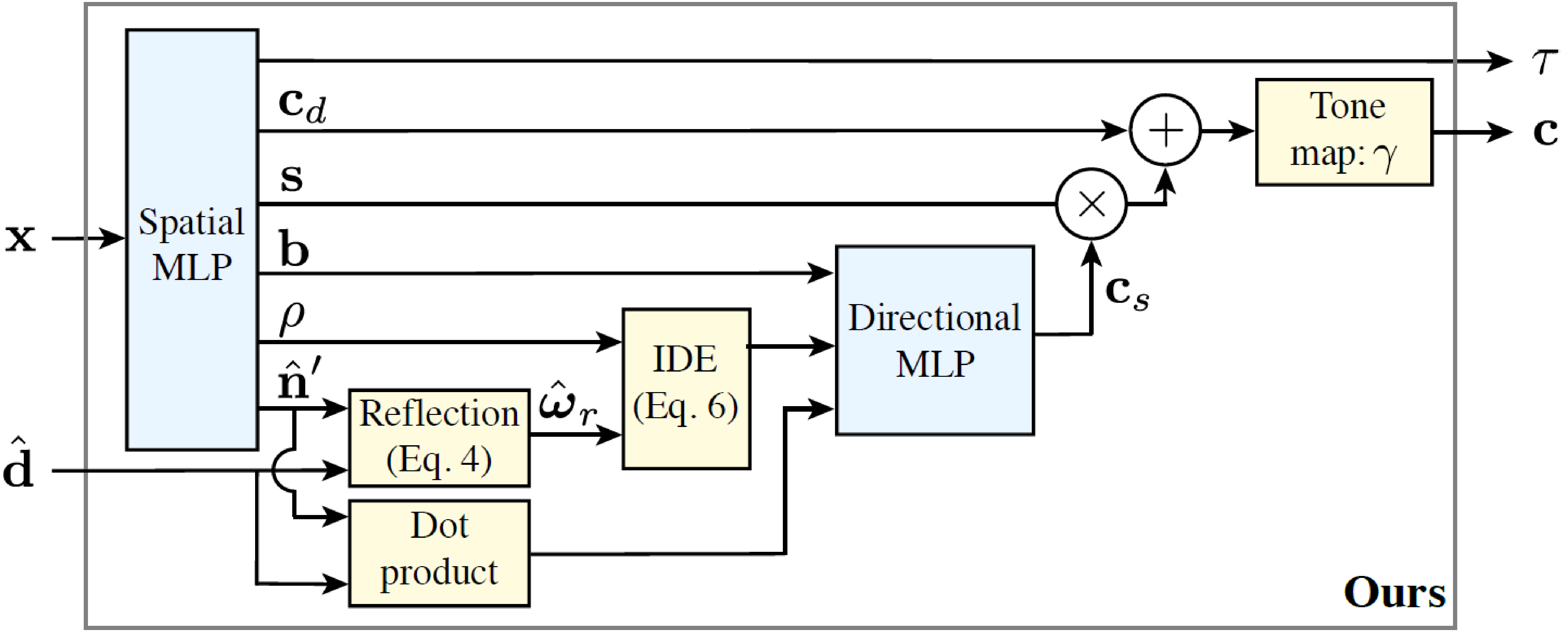
Integrated Directional Encoding

- Radiance cannot be represented as a function of reflection direction alone
  - High roughness → radiance changes slowly
- Encode reflection directions with a set of spherical harmonics: $\{Y_l^m\}$
- Encode distribution of reflection vectors instead of single vector: $vMF(\widehat{\omega}_r, \kappa)$
  - Can reason about roughness
- $IDE(\widehat{\omega}_r, \kappa) = \left\{E_{\widehat{\omega}\sim vMF(\widehat{\omega}_r,\kappa)}[Y_l^m(\widehat{\omega})] : (l,m) \in M_L\right\}, \quad M_L = \{(l,m) : l = 1, \dots, 2^L, m = 0, \dots, l\}$
- $E_{\widehat{\omega}\sim vMF(\widehat{\omega}_r,\kappa)}[Y_l^m(\widehat{\omega})] \approx \exp\left(-\frac{l(l+1)}{2\kappa}\right) Y_l^m(\widehat{\omega}_r)$



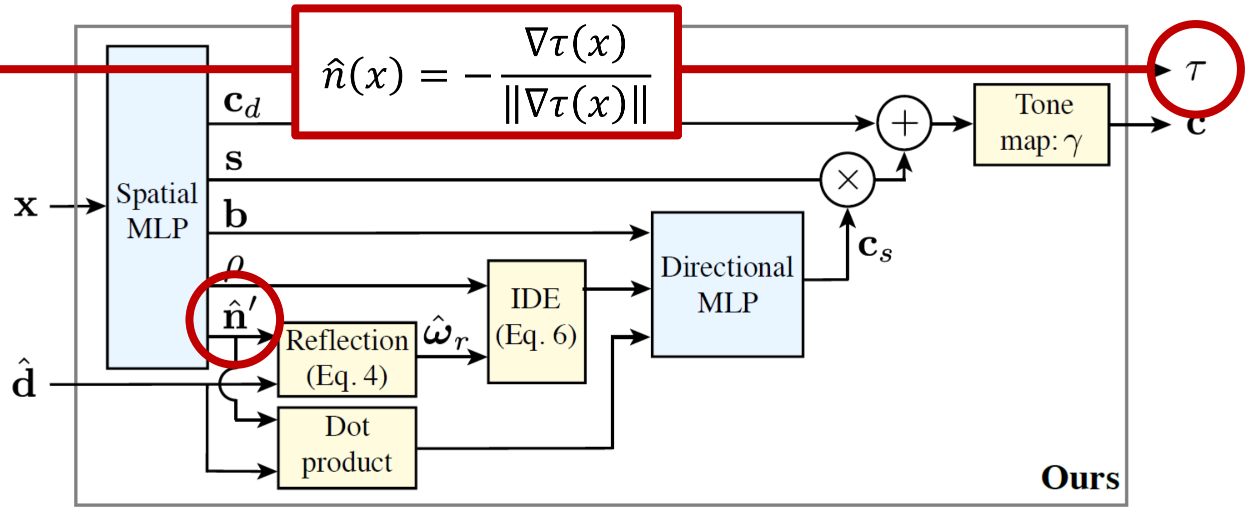
Architecture

- $c = \gamma(c_d + s \times c_s)$, $\quad$ *s is specular tint*

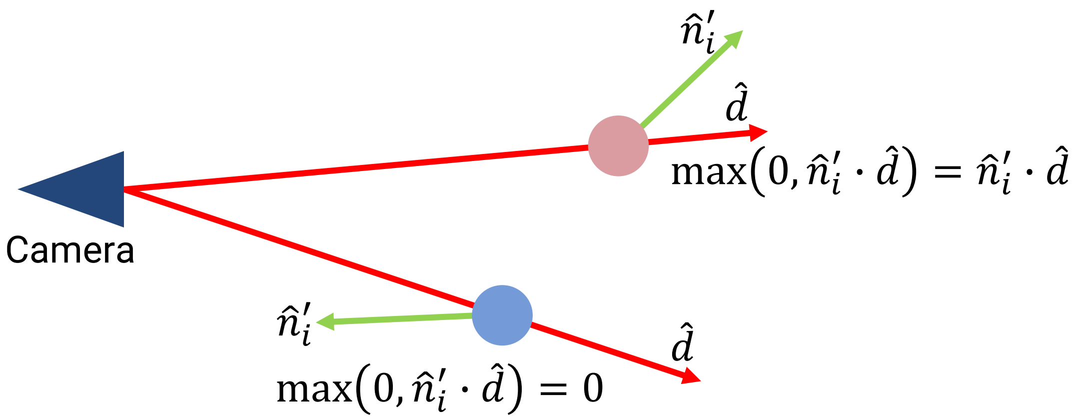


Accurate Normal Vectors

- Tie **predicted normal** and **density gradient normal**
  - $R_p = \sum_i w_i \|\hat{n}_i - \hat{n}_i{}'\|^2$
  - w_i is the weight of the ith sample along the ray



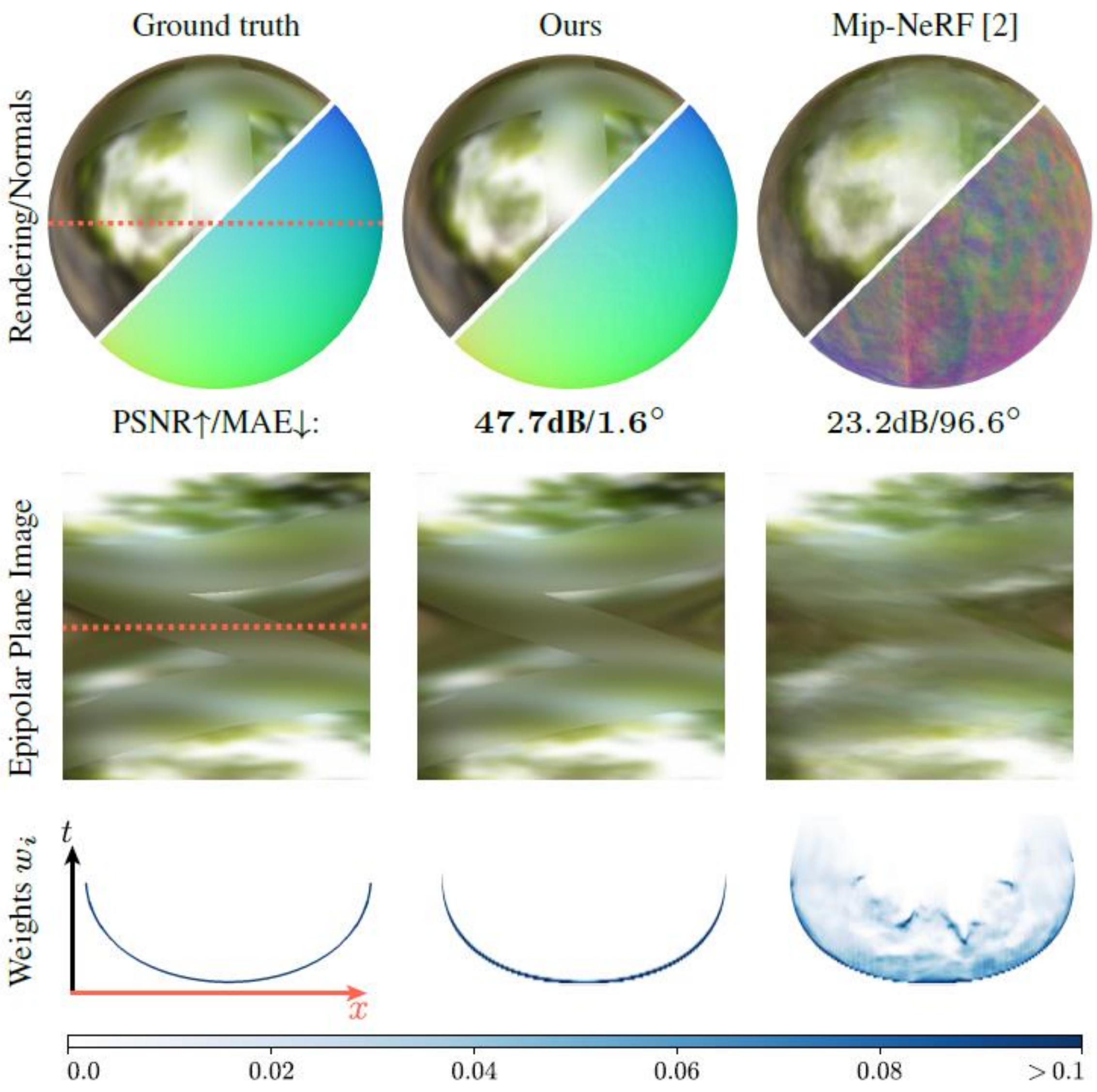
- Penalize normal vector that are **back-facing**
  - $R_o = \sum_i w_i \max(0, \hat{n}_i' \cdot \hat{d})^2$



Accurate Normal Vectors

$$R_p = \sum_i w_i \|\hat{n}_i - \hat{n}_i'\|^2$$

$$R_o = \sum_i w_i \max(0, \hat{n}_i' \cdot \hat{d})^2$$



Experiments

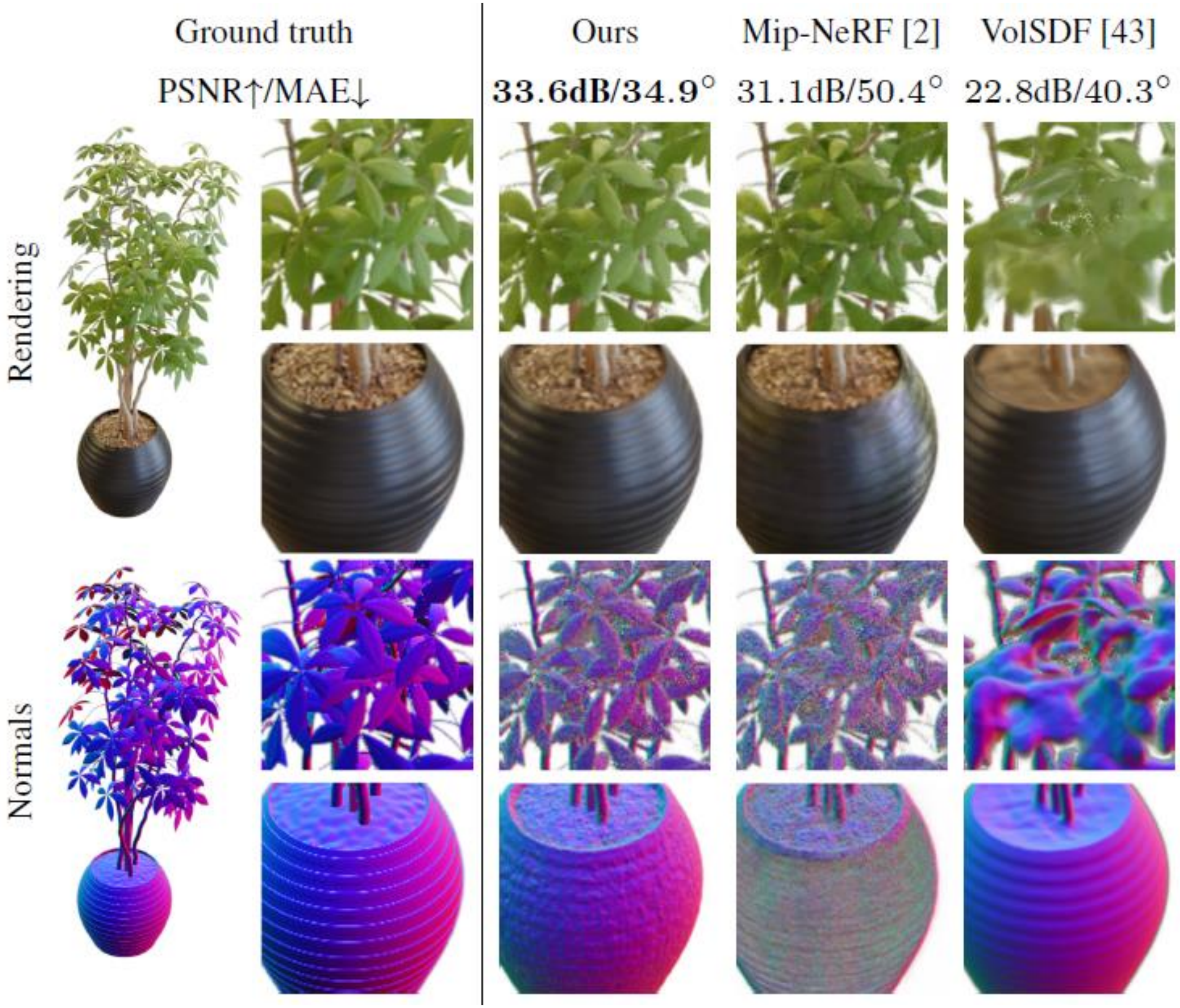
	PSNR ↑	SSIM ↑	LPIPS ↓	MAE° ↓
PhySG [45] (requires object masks)	26.21	0.921	0.121	8.46
Mip-NeRF [2]	29.76	0.942	0.092	60.38
Mip-NeRF, 8 layers	31.59	0.956	0.072	58.07
Mip-NeRF, 8 layers, w/ normals	31.39	0.955	0.074	58.27
Mip-NeRF, 8 layers, w/ $\mathcal{R}_o$	31.48	0.955	0.073	57.37
Ours, no reflection	29.47	0.944	0.084	16.19
Ours, no $\mathcal{R}_o$	31.62	0.954	0.078	52.56
Ours, no pred. normals	30.91	0.936	0.105	30.67
Ours, concat. viewdir	35.42	0.966	0.061	21.25
Ours, fixed lobe	35.52	0.965	0.061	26.46
Ours, no diffuse color	33.32	0.962	0.067	26.13
Ours, no tint	35.45	0.965	0.060	22.70
Ours, no roughness	33.39	0.963	0.065	25.96
Ours, PE	35.90	0.968	0.058	20.31
Ours	35.96	0.967	0.058	18.38

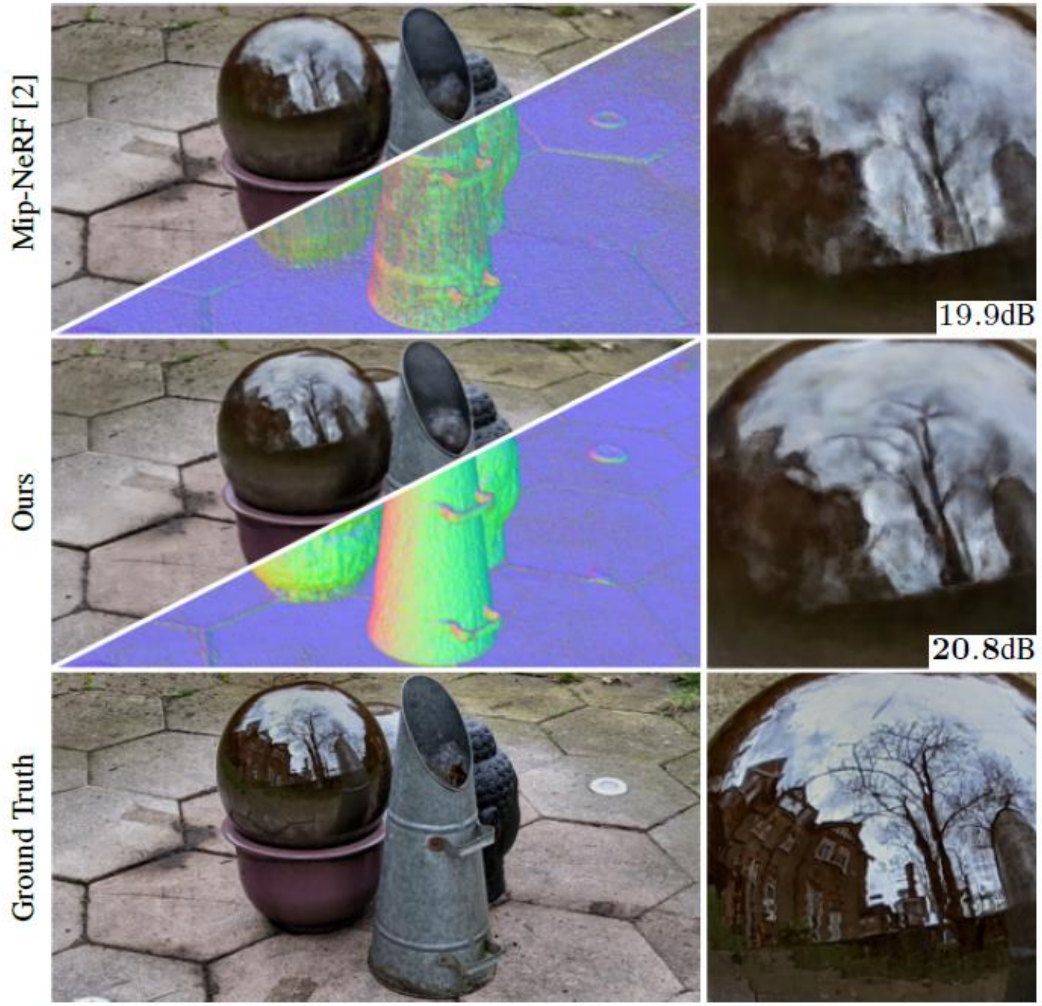
Table 1. Baseline comparisons and ablation study on our "Shiny Blender" dataset.

	PSNR ↑	SSIM ↑	LPIPS ↓	MAE° ↓
PhySG [45] (requires object masks)	20.60	0.861	0.144	29.17
VolSDF [43]	27.96	0.932	0.096	19.45
NSVF [19]	31.74	0.953	0.047	–
NeRF [24]	32.38	0.957	0.046	–
Mip-NeRF [2]	33.09	0.961	0.043	38.30
Ours, PE	33.90	0.965	0.039	24.16
Ours	33.99	0.966	0.038	23.22

Table 2. Results for our method compared to previous approaches on the Blender dataset [24].

Experiments





Scene Editing

Scene Editing

Conclusion

- Limitations
  - Increased computation
    - IDE is slightly slower than standard positional encoding
    - Back propagation is 25% slowly than mip-NeRF
  - Reparameterization does not explicitly model interreflections or non-distant illumination

- Contributions
  - Improve quality of view-dependent appearance and accuracy of normal vector
  - Represent outgoing radiance as interpretable components
    - normal, roughness, diffuse and specular color…

Quiz

1. Please select the elements that depend on viewing direction on NeX.
   ① Alpha Transparency
   ② Reflect Coefficients
   ③ Neural Basis functions

2. What element does Ref-NeRF use as input of IDE instead of view direction?
   (   ) of view direction

References

- Suttisak Wizadwongsa et al., NeX: Real-time View Synthesis with Neural Basis Expansion, CVPR, 2021
- Dor Verbin et al., Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields, CVPR, 2022