Supplementary Material: Neural Directional Encoding for Efficient and Accurate View-Dependent Appearance Modeling

A. Additional Implementation Details

A.1. Cone tracing footprint

In Sec. 4.2, we choose the cone to cover the (cosine weighted) GGX distribution [8] centered in the reflected direction $\boldsymbol{\omega}_r$. Assuming $\boldsymbol{\omega}_r = (0, 0, 1)$, the distribution D with roughness ρ in spherical coordinates (θ, ϕ) can be written as:

$$D(\theta, \phi) = \frac{\alpha^2 \max(\cos\theta, 0)}{\pi(\cos^2\theta(\alpha^2 - 1) + 1)^2}, \ \alpha = \rho^2. \quad (1)$$

If we want the cone to cover a certain fraction T of the distribution, the polar angle θ should satisfy:

$$\begin{aligned} T &= \int_0^{2\pi}\int_0^{\theta} D(\theta', \phi)\sin\theta' d\theta' d\phi \\ &= \frac{1 - \cos^2\theta}{1 + \cos^2\theta(\alpha^2 - 1)} \\ \Rightarrow \cos\theta &= \sqrt{\frac{1 - T}{T(\alpha^2 - 1) + 1}}, \end{aligned} \quad (2)$$

which gives the base cone radius r_0:

$$r_0 = \cot\theta = \frac{\sqrt{1 - \cos^2\theta}}{\cos\theta} = \sqrt{\frac{T}{1 - T}}\rho^2. \quad (3)$$

We found $T = 75\%$ in practice gives good results, which suggests $r_0 = \sqrt{3}\rho^2$. Therefore, the footprint at $\mathbf{x}'_i$ from $\mathbf{x}$ is $r_i = \sqrt{3}\rho^2\|\mathbf{x} - \mathbf{x}'_i\|_2$.

A.2. Real-time application

We use a two-pass deferred shading in our real-time model. The first pass rasterizes the world-space position $\mathbf{x}$, normal $\mathbf{n}$, diffuse color $\mathbf{c}_d$, specular tint $\mathbf{k}_s$, spatial feature $\mathbf{f}$, and roughness ρ into the G-buffer. In the second pass, we then calculate the NDE $\mathbf{H}$, including a cubemap lookup for far-field feature $\mathbf{H}_f$ and the cone tracing of near-field feature $\mathbf{H}_n$, and decode it to get the specular color $\mathbf{c}_s$. The MLP evaluations are executed sequentially inside the pixel shader, and we implement the early ray termination trick [3, 9] to stop the cone tracing if the accumulated transmittance is below 0.01. Because small decoder MLPs tend to provide unstable geometry optimization, we use the fixed SDF network weight from our NDE trained with 64 MLP width when training other variants that use smaller decoder MLPs (Sec. 5.3).

	ENVIDR	Ref-NeRF	NDE (ours)
PSNR↑	22.67	*23.46*	**23.63**

Table 1. **PSNR on the Ref-NeRF Garden Spheres scene.**

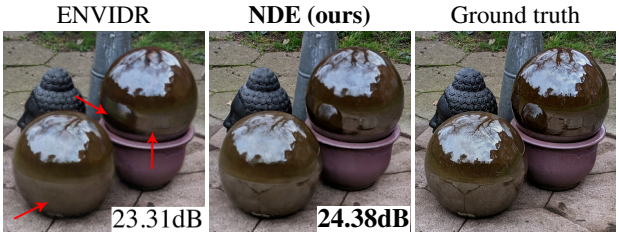

Figure 1. **Qualitative comparison on the Garden Spheres scene of Ref-NeRF real dataset.** Numbers shows the image PSNR; zoom in to see the difference.

A.3. Spatial mip-mapping strategies

We introduce mip-mapping strategies of spatial encodings in Sec. 5.3 using either a triplane [2, 4] or a hash grid [6]. Let $\mathbf{T}_{xy}, \mathbf{T}_{yz}, \mathbf{T}_{zx}$ denote the three 2D planes of the triplane $\mathbf{T}$. A mip-mapped query at location $\mathbf{x} = (x, y, z)$ of mip level λ is given by:

$$\begin{aligned} &\text{mipmap}(\mathbf{T}(\mathbf{x}), \lambda) = \\ &\qquad \bigoplus_{\mathbf{u}\in U} \text{lerp}(\mathbf{T}_{\mathbf{u}}^{\lfloor\lambda\rfloor}(\mathbf{u}), \mathbf{T}_{\mathbf{u}}^{\lceil\lambda\rceil}(\mathbf{u}), \lambda - \lfloor\lambda\rfloor), \\ &U = \{(x, y), (y, z), (z, x)\}, \mathbf{T}_{\mathbf{u}}^{k} = \text{downsample}(\mathbf{T}_{\mathbf{u}}, k), \end{aligned} \quad (4)$$

where $\bigoplus$ is the concatenation operation. For a hash grid feature $\mathbf{F}$ with l^{th} level feature $\mathbf{F}_l$ (beginning from the finest resolution), its mip-mapping is given by:

$$\text{mipmap}(\mathbf{F}(\mathbf{x}), \lambda) = \bigoplus_{l} \text{clamp}(l + 1 - \lambda, 0, 1)\mathbf{F}_l(\mathbf{x}). \quad (5)$$

B. Additional Results

For the unbounded real scene evaluation, we provide the results on the Garden Spheres scene of Ref-NeRF real dataset [7] in Tab. 1 and Fig. 1. It can be seen that our method is able to recover more interreflection details in real-world compared to other baselines. Considering perceptual measures are more reasonable for reflection quality comparison, we additionally show the FLIP [1] metric on synthetic

Method	Mat.	Teapot	Toaster	Car	Ball	Coffee	Helmet	Mean
NeRO	0.082	0.012	*0.097*	*0.049*	0.058	0.039	0.083	0.060
ENVIDR	0.062	0.030	0.098	0.056	*0.037*	0.046	*0.049*	0.054
Ref-NeRF	**0.023**	*0.011*	0.108	0.071	0.038	**0.030**	0.072	*0.050*
3DGS	0.042	**0.007**	0.153	0.051	0.104	*0.032*	0.073	0.066
NDE	*0.039*	**0.007**	**0.065**	**0.038**	**0.027**	0.035	**0.035**	**0.035**

Table 2. **FLIP metric on synthetic scenes.**

	NeRO	ENVIDR	Ref-NeRF	**NDE**	**NDE-RT**
PSNR ↑	28.75	31.29	28.18	**34.08**	*32.97*
SSIM ↑	0.956	0.969	0.946	**0.985**	*0.984*
LPIPS ↓	0.046	0.022	0.030	**0.008**	*0.010*

Table 3. **Quantitative results on the teaser scene.**

	Mat.	Teapot	Toaster	Car	Ball	Coffee	Helmet	Mean
PSNR ↑								
3DGS	29.98	45.69	20.99	27.25	27.65	32.31	28.26	30.30
NDE-RT	**30.28**	**47.02**	**28.31**	**28.91**	**43.23**	**34.21**	**36.38**	**35.48**
SSIM ↑								
3DGS	0.960	0.997	0.895	0.930	0.937	**0.972**	0.951	0.949
NDE-RT	**0.967**	**0.998**	**0.954**	**0.962**	**0.994**	**0.972**	**0.987**	**0.976**
LPIPS ↓								
3DGS	0.034	0.007	0.126	0.048	0.162	0.078	0.080	0.076
NDE-RT	**0.020**	**0.003**	**0.051**	**0.030**	**0.023**	**0.041**	**0.019**	**0.027**

Table 4. **Per-scene comparison with 3DGS on synthetic scenes.**

Method	Mat.	Teapot	Toaster	Car	Ball	Coffee	Helmet	Mean
Hashgrid	30.12	46.46	25.83	29.94	36.41	33.25	34.08	33.73
Pos. enc.	**31.53**	**49.12**	**30.32**	**30.39**	**44.66**	**36.57**	**37.77**	**37.19**

Table 5. **Comparison of geometry encoding on synthetic scenes in PSNR.** "Pos. enc." denotes positional encoding.

scenes in Tab. 2. Overall, our method still demonstrates higher rendering quality compared to other baselines.

C. Experiment Details

We provide the quantitative results on the teaser scene (Fig. 1 of the main paper) compared to the baselines in Tab. 3 and the per-scene comparison of our real-time model (NDE-RT) with 3DGS [5] in Tab. 4. Table 5 shows the comparison of different SDF encodings (Fig. 12 of the main paper). Table 6 and 7 show the per-scene quantitative results of our real-time and offline model with different MLP width (Width) on the synthetic dataset. In Fig. 2, we show the per-scene rendering results of both our offline (NDE) and real-time (NDE-RT) model on the synthetic dataset together with the reconstructed surface normals. The normals are masked by the foreground mask to get rid of floaters with the background color.

Width	Mat.	Teapot	Toaster	Car	Ball	Coffee	Helmet	Mean
PSNR ↑								
64	31.53	49.12	30.32	30.39	44.66	36.57	37.77	37.19
32	30.89	48.88	29.33	29.51	44.34	36.24	37.63	36.69
16	30.59	48.56	29.09	29.24	43.61	36.07	36.47	36.23
SSIM ↑								
64	0.972	0.999	0.968	0.968	0.995	0.979	0.990	0.982
32	0.968	0.999	0.961	0.962	0.994	0.977	0.989	0.979
16	0.965	0.998	0.959	0.960	0.994	0.977	0.986	0.977
LPIPS ↓								
64	0.017	0.002	0.039	0.024	0.022	0.033	0.014	0.022
32	0.021	0.002	0.057	0.032	0.021	0.033	0.017	0.026
16	0.023	0.002	0.058	0.034	0.021	0.034	0.022	0.028

Table 6. **Per-scene results of our offline models on synthetic scenes.** The first column suggests the decoder MLP width.

Width	Mat.	Teapot	Toaster	Car	Ball	Coffee	Helmet	Mean
PSNR ↑								
64	30.28	47.02	28.31	28.91	43.23	34.21	36.38	35.48
32	28.84	43.75	27.28	27.76	41.00	33.92	35.26	33.97
16	28.33	43.14	26.95	27.47	41.48	33.81	34.81	33.71
SSIM ↑								
64	0.967	0.998	0.954	0.962	0.994	0.972	0.987	0.976
32	0.955	0.996	0.945	0.954	0.992	0.971	0.984	0.971
16	0.951	0.996	0.942	0.951	0.992	0.971	0.982	0.969
LPIPS ↓								
64	0.020	0.003	0.051	0.030	0.023	0.041	0.019	0.027
32	0.030	0.006	0.068	0.039	0.025	0.043	0.026	0.034
16	0.033	0.007	0.070	0.043	0.024	0.044	0.030	0.036
FPS ↑								
64	55	130	55	70	65	30	58	66
32	220	400	150	200	210	110	190	211
16	350	600	250	330	300	200	290	331

Table 7. **Per-scene results of our real-time models on synthetic scenes.** The first column suggests the deocder MLP width.

References

[1] Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D Fairchild. Flip: A difference evaluator for alternating images. *Proc. ACM Comput. Graph. Interact. Tech.*, 3(2):15–1, 2020. 1

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 1

[3] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021. 1

[4] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 2023. 1

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM TOG*, 2023. 2

[6] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *SIGGRAPH*, 2022. 1

[7] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler,

Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 1

[8] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *EGSR*, 2007. 1

[9] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 1

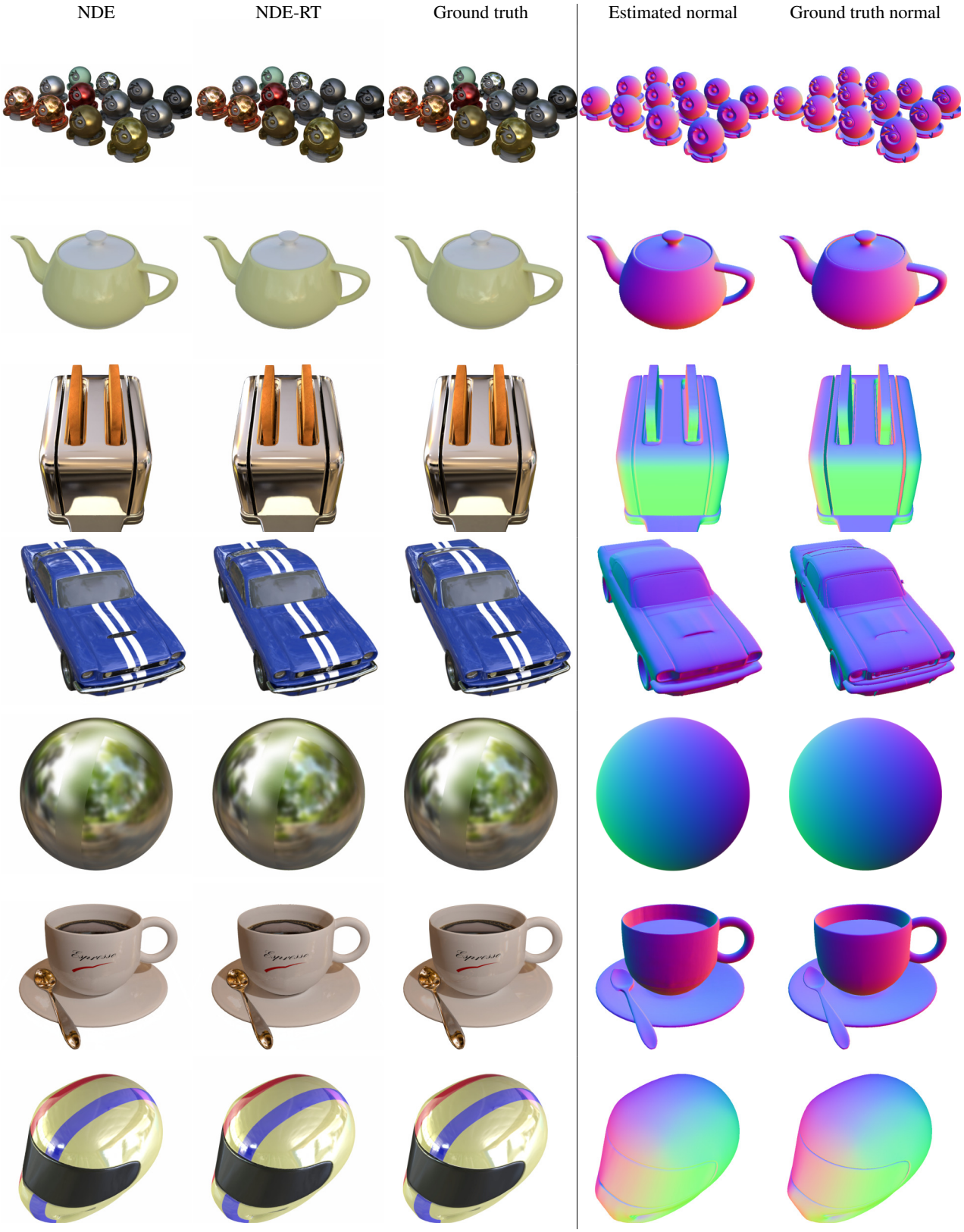

Figure 2. **Qualitative results on each synthetic scene** for our offline (NDE) and real-time (NDE-RT) methods.