

# RISE-SDF: A Relightable Information-Shared Signed Distance Field for Glossy Object Inverse Rendering

## Supplementary Material

Dataset	Material	Relighting	# Envmaps	Shiny Object
NeRF Synthetic [29]	✗	✗	0	✗
Shiny Blender [43]	✗	✗	0	✓
NeRO Synthetic [25]	✗	✓	3	✓
NeRFactor Synthetic [54]	✓	✓	8	✗
TensoIR Synthetic [15]	✓	✓	8	✗
Ours	✓	✓	9	✓

Table 3. **Comparison of the availability of the datasets.** We show the availability of ground truth material, relighting, number of environment maps for relighting, and availability glossy object. Ours is the first dataset with ground truth material and relighting for shiny objects.

### 6. Implementation Details

Our code is built on the instant-nsr-pl codebase [12]. We use a 512-resolution progressive hash grid with 16 levels. The geometry MLP has 2 layers with 128 neurons. The diffuse, specular, secondary MLP has 4 layers with 128 neurons. And the roughness and blending MLP has 2 layers with 128 neurons. The resolution of the environment map is  $6 \times 512 \times 512 \times 3$ . We start from the 4-th level of the hash grid and increase by 1 level for every 500 iterations. Regarding the hyperparameters, we use  $\lambda_c = 10$ ,  $\lambda_{eik} = 0.1$ ,  $\lambda_{curv} = 1$ . We use Adam optimizer with  $\beta_1 = 0.9$ ,  $\beta_1 = 0.999$ , and  $\epsilon = 10^{-12}$ . The first stage of our method is training for 10k iterations and the second stage for 70k iterations. All experiments are conducted on a single RTX 3090Ti GPU.

### 7. Shiny Inverse Rendering Synthetic Dataset

We built our own Shiny Inverse Rendering Dataset with an aligned BRDF model as no dataset with ground truth material and relighting results for glossy objects exists. We provide five scenes including *teapot*, *coffee*, *muscle car*, *toaster*, and *helmet* from the Shiny Blender dataset [43], with ground truth albedo, roughness, and relighting under nine different environment maps. Here are the steps to create the dataset:

- To align the BRDF during dataset generation and inverse rendering, we change the shader nodes of the five objects to the Principled BRDF with default parameters except for the metallic, roughness, and base color (albedo) in Blender [1].
- We render the objects under ten different light conditions, and choose one of them as the training light, and others for relighting.
- To export accurate ground truth albedo and roughness, we manually create the blender files using Diffuse BRDF with the albedo and roughness value in the Principled

BRDF model as the base color and render the diffuse pass.

In Table 3, we compare our new dataset to existing ones. To the best of our knowledge, ours is the first glossy dataset with an aligned BRDF model for forward and inverse rendering with accurate material and relighting ground truth.

### 8. Discussion on MLP Predictions in Stage 1

Using Eq. 4 and 5 in the main paper, we can divide the per-sample physically-based rendering equation into the following format:

$$\begin{aligned} \{\mathbf{c}_d\}_i &= (1 - m_i) \frac{\mathbf{a}_i}{\pi} \int_{H^2} L_{dir}(\mathbf{x}_i, \omega_j) \cos\theta_i d\omega_j \\ \{\mathbf{c}_s\}_i &= \int_{H^2} \frac{D(\rho_i) F(m_i, \mathbf{a}_i) G}{4|\mathbf{d} \cdot \mathbf{n}_i| |\omega_j \cdot \mathbf{n}_i|} L_{dir}(\mathbf{x}_i, \omega_j) \cos\theta_i d\omega_j \end{aligned} \quad (16)$$

Since the geometry is not well reconstructed in the first stage, the SDF is not converged near the surface. Therefore, the amount of volume rendering samples is large during the first stage since it is hard to prune the samples according to the SDF value. To save computation and to enhance training stability, we directly use MLPs to predict these two integration values. A blending weight is predicted to simulate the effect of metallic parameters. Note that we assume there is no indirect illumination in this equation.

### 9. Split-sum Approximation

Given the sample location  $\mathbf{x}_i$ , ray direction  $\mathbf{d}$ , and the normal direction  $\mathbf{n}_i$ , we write the rendering equation and the corresponding Monte Carlo integration:

$$\begin{aligned} \mathbf{c}_i^{pbr}(\mathbf{x}_i, \mathbf{d}) &= \int_{H^2} f_r(\mathbf{x}_i, \mathbf{d}, \omega_j) L_i(\mathbf{x}_i, \omega_j) (\mathbf{n}_i \cdot \omega_j) d\omega_j \\ &\approx \frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} \frac{f_r(\mathbf{x}_i, \mathbf{d}, \omega_j) L_i(\mathbf{x}_i, \omega_j) (\mathbf{n}_i \cdot \omega_j)}{p(\omega_j; \hat{\mathbf{d}}_i, \rho_i)} \\ &\approx \left( \frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} \frac{f_r(\mathbf{x}_i, \mathbf{d}, \omega_j) (\mathbf{n}_i \cdot \omega_j)}{p(\omega_j; \hat{\mathbf{d}}_i, \rho_i)} \right) \\ &\quad \cdot \left( \frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} L_i(\mathbf{x}_i, \omega_j) \right) \end{aligned} \quad (17)$$

This integration can be approximated by the multiplication of separate sums. If we write the split-sum Monte Carlo

integration back to the continuous form:

$$\mathbf{c}_i^{pbr}(\mathbf{x}_i, \mathbf{d}) \approx \int_{H^2} f_r(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) d\omega_j \cdot \int_{H^2} L_i(\mathbf{x}_i, \omega_j) p(\omega_j; \hat{\mathbf{d}}_i, \rho_i) d\omega_j \quad (18)$$

The first integration is called as BSDF integral, and the second integration is called the light integral. The light integration is approximated by a multi-level mipmap. Consistent with [31], we use cube maps (with resolution  $6 \times 512 \times 512$ ). The base level corresponds to the smallest roughness value, and increases among the mip-levels. For each level, the mipmap is computed by average pooling the base level followed by a convolution using the GGX distribution with the corresponding roughness as the kernel. The mipmap is implemented as a differentiable function with respect to  $\hat{\mathbf{d}}_i$  and roughness  $\rho$ :

$$\int_{H^2} L_i(\mathbf{x}_i, \omega_j) p(\omega_j; \hat{\mathbf{d}}_i, \rho) d\omega_j \approx \text{Mipmap}(\hat{\mathbf{d}}_i, \rho) \quad (19)$$

In our paper, the BRDF is defined as a simplified version of the Disney BSDF as in Eq. 5 in the main paper. The fresnel term  $F$  is defined as:

$$F = F_0 + (1 - F_0)(1 - \omega_j \cdot \mathbf{h})^5, \quad (20)$$

where  $F_0 = 0.04 * (1 - m) + m * \mathbf{a}$  is the simplified basic reflection ratio, and  $\mathbf{h}$  is the half-way vector between  $-\mathbf{d}$  and  $\omega_j$ . Since we have both the diffuse and the specular parts in the BSDF, we can separate the rendering equation into the diffuse and specular parts. For the diffuse part, we directly extract the albedo outside of the integrand, and use the normal direction and largest roughness to query the mipmap:

$$\begin{aligned} \mathbf{l}_i^d &= \text{Mipmap}(\mathbf{n}_i, \rho_{max}) \\ \{\mathbf{c}_d^{pbr}\}_i &= ((1 - m_i) \mathbf{a}_i \int_{H^2} \frac{(\mathbf{n}_i \cdot \omega_j)}{\pi} d\omega_j) \mathbf{l}_i^d \\ &= (1 - m_i) * \mathbf{a}_i * \mathbf{l}_i^d \end{aligned} \quad (21)$$

For the specular part of BSDF integration, if we substitute the Fresnel in the BSDF:

$$\begin{aligned} &\int_{H^2} f_r(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) d\omega_j \\ &= F_0 \int_{H^2} \frac{f_r(\mathbf{x}_i, \mathbf{d}, \omega_j)}{F} (1 - (1 - \omega_j \cdot \mathbf{h})^5)(\mathbf{n}_i \cdot \omega_j) d\omega_j \\ &\quad + \int_{H^2} \frac{f_r(\mathbf{x}_i, \mathbf{d}, \omega_j)}{F} (1 - \omega_j \cdot \mathbf{h})^5 (\mathbf{n}_i \cdot \omega_j) d\omega_j \end{aligned} \quad (22)$$

This leaves two integrations only dependent on  $\rho$  and  $\mathbf{n}_i \cdot \omega_j$ , then we can precompute the result and store it to a 2D LUT:

$$\begin{aligned} &\int_{H^2} f_r(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) d\omega_j \\ &= F_0 * F_1(\rho, \mathbf{n}_i \cdot \mathbf{d}), + F_2(\rho, \mathbf{n}_i \cdot \mathbf{d}), \\ \mathbf{l}_i^s &= \text{Mipmap}(\hat{\mathbf{d}}_i, \rho), \\ \{\mathbf{c}_s^{pbr}\}_i &= (F_0 * F_1 + F_2) * \mathbf{l}_i^s. \end{aligned} \quad (23)$$

## 10. Discussion on Indirect Illumination

To model the indirect illumination, we first assume that the indirect effect is only apparent for the specular part of the radiance. Therefore, we only modify the specular part  $\{\mathbf{c}_s\}_i$  to  $\{\mathbf{c}'_s\}_i$  in Eq. (16):

$$\begin{aligned} f_r^s(\mathbf{x}_i, \mathbf{d}, \omega_j) &= \frac{DFG}{4|\mathbf{d} \cdot \mathbf{n}_i| |\omega_j \cdot \mathbf{n}_i|} \\ \{\mathbf{c}'_s\}_i &= \int_{H^2} f_r^s(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) \\ &\quad ((1 - O)L_d(\mathbf{x}_i, \omega_j) + OL_{ind}(\mathbf{x}_i, \omega_j)) d\omega_j \\ &= (1 - O) \int_{H^2} f_r^s(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) L_d(\mathbf{x}_i, \omega_j) d\omega_j \\ &\quad + O \int_{H^2} f_r^s(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) L_{ind}(\mathbf{x}_i, \omega_j) d\omega_j \\ &= (1 - O) \{\mathbf{c}_s\}_i \\ &\quad + O \int_{H^2} f_r^s(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) L_{ind}(\mathbf{x}_i, \omega_j) d\omega_j \end{aligned} \quad (24)$$

After applying volume rendering to the equation above, we have:

$$\begin{aligned} C_{ind} &= \sum_{i=1}^N w_i \int_{H^2} f_r^s(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j) L_{ind}(\mathbf{x}_i, \omega_j) d\omega_j \\ \mathbf{C}'_s &= \sum_{i=1}^N w_i \{\mathbf{c}'_s\}_i = (1 - O) \sum_{i=1}^N w_i \{\mathbf{c}_s\}_i + OC_{ind} \\ &= (1 - O) \mathbf{C}_s + OC_{ind} \end{aligned} \quad (25)$$

Compared with NeRO [25], we use an MLP  $f_{ind}$  to directly predict  $C_{ind}$  instead of predicting the indirect illumination for each volume sample. More specifically, we use the normal vector  $\mathbf{N}$ , secondary ray direction  $\hat{\mathbf{d}}$ , and the geometry feature  $\beta_{\hat{\mathbf{x}}}$  at the expected intersection point. As shown in Fig. 7, our algorithm only has one secondary color MLP query. Assume there are  $N$  samples on the primary ray and  $M$  samples on the secondary ray ( $M \ll N$ ). For the indirect sampling in NeRO [25], since there is one MLP for opacity prediction and one MLP for indirect color for each sample on the primary ray, the total number of secondary MLP queries is  $2N$ . For TensoIR [15] and ENVIDR [22], since they compute the surface intersection first and then

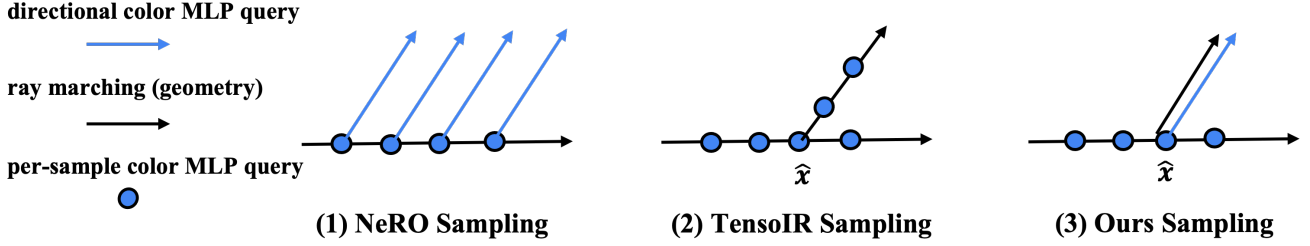


Figure 7. Indirect sampling method. Each blue arrow represents an MLP query using the secondary ray direction. Each black arrow represents a ray marching with multiple geometry MLP queries. Each blue dot represents color MLP queries for each sample. (1) For each sample along the primary ray, NeRO [25] queries an opacity MLP and an indirect color MLP to estimate the per-sample indirect illumination. (2) TensorIR [15] and ENVIDR [22] compute the expected surface intersection  $\hat{x}$  and apply a secondary ray marching. This is not efficient when there are multiple color MLPs in our case. (3) Our indirect sampling only queries the geometry MLP to compute the opacity and uses one color MLP only once for the expected surface intersection.

use secondary ray marching to compute the radiance field, the number of MLP queries is  $4M$  (in our case there is one density MLP and three color MLPs). In our algorithm, we only query the geometry MLP for indirect illumination, and query indirect color once for the expected intersection  $\hat{x}$ , thus the total query number is  $M + 1$ .

## 11. Derivation of Second Split-sum

We define the illumination as:

$$L_i = \mathbb{1}[\rho > \rho_t] L_{dir} + \mathbb{1}[\rho \leq \rho_t] ((1 - O) * L_{dir} + O * L_{ind}), \quad (26)$$

which means we only consider the indirect light when the roughness is smaller than the threshold. We plug this equation into the light integral in Eq. (18), and the relighting light integral for the specular part becomes:

$$\begin{aligned} \mathbf{I}_{relight}^s = & (\mathbb{1}[\rho > \rho_t] + \mathbb{1}[\rho \leq \rho_t] * (1 - O)) * \mathbf{I}^s \\ & + \mathbb{1}[\rho \leq \rho_t] * O * \int_{H^2} L_{ind}(\hat{\mathbf{x}}, \omega_j) p(\omega_j; \hat{\mathbf{d}}, \rho) d\omega_j \end{aligned} \quad (27)$$

The light integral  $\mathbf{I}^s$  in the equation can be computed by volume rendering the per-sample light integral  $\mathbf{I}_i^s$ . For the second part, with a small enough  $\rho_t$ , the GGX distribution can be approximated as a delta function with an infinity value at the reflected direction  $\hat{\mathbf{d}}$ . So we simplify the equation above into the following form:

$$\begin{aligned} \mathbf{I}_{relight}^s \approx & (\mathbb{1}[\rho > \rho_t] + \mathbb{1}[\rho \leq \rho_t] * (1 - O)) * \mathbf{I}^s \\ & + \mathbb{1}[\rho \leq \rho_t] * O * L_{ind}(\mathbf{x}_i, \hat{\mathbf{d}}). \end{aligned} \quad (28)$$

## 12. Relighting Runtime Cost

In Tab. 4, we show the per-scene relighting runtime (fps) of NeRO (Blender) [25] and our pipeline. We can find that both methods cannot achieve interactive frame rates, while our method achieves higher quality with comparable runtime.

	NeRO	Ours*
teapot	0.481	0.461
musclecar	0.292	0.253
coffee	0.203	0.193
toaster	0.199	0.133
helmet	0.282	0.148
Avg.	0.292	0.238

Table 4. Relighting Runtime (fps) of NeRO and our pipeline.

## 13. Diffuse Synthetic Scene Result

In Tab. 5, we compare the qualitative metrics on TensorIR dataset [15], which contains diffuse objects. Since shadows are not explicitly considered in our pipeline, performance on more diffuse datasets is comparable to TensorIR but does not achieve state-of-the-art in every metric. One of the potential solutions to improve the quality is to add an MLP or a spherical harmonic grid to cache the shadow after stage one as done in GS-IR [23].

## 14. Discussions on Glossy Real Dataset

We generate ground truth object masks by projecting the ground truth mesh to the camera planes. We use the generated masks to train our model in company with mask loss as in NeuS[44]. Our model gives sub-optimal performance on this real dataset because it wrongly estimate near-field indirect illumination. The plate holding the objects is masked out and the indirect illumination MLP  $f_{ind}$  could not explain secondary shading effects. Further, we do not consider explicitly the reflections of the photographer on the object. For objects with a large part of the reflection of the photographer, our method would struggle to estimate the correct geometry, material, and environment light. We believe the results can be ameliorated by taking objects' surroundings into account and modeling explicitly the reflections of the

Scene	Method	Normal	Albedo			Novel View Synthesis			Relighting		
		MAE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Lego	NeRFactor	9.767	25.444	0.937	0.112	26.076	0.881	0.151	23.246	0.865	0.156
	InvRender	9.980	21.435	0.882	0.160	24.391	0.883	0.151	20.117	0.832	0.171
	TensoIR	5.980	25.240	0.900	0.145	34.700	0.968	0.037	27.596	0.922	0.095
	Ours	9.247	20.457	0.890	0.113	31.657	0.995	0.009	25.599	0.980	0.028
Hotdog	NeRFactor	5.579	24.654	0.950	0.142	24.498	0.940	0.141	22.713	0.914	0.159
	InvRender	3.708	27.028	0.950	0.094	31.832	0.952	0.089	27.630	0.928	0.089
	TensoIR	4.050	30.370	0.947	0.093	36.820	0.976	0.045	27.927	0.933	0.115
	Ours	4.515	22.756	0.961	0.075	37.866	0.997	0.007	26.665	0.977	0.038
Armadillo	NeRFactor	3.467	28.001	0.946	0.096	26.479	0.947	0.095	26.887	0.944	0.102
	InvRender	1.723	35.573	0.959	0.076	31.116	0.968	0.057	27.814	0.949	0.069
	TensoIR	1.950	34.360	0.989	0.059	39.050	0.986	0.039	34.504	0.975	0.045
	Ours	3.098	42.440	0.959	0.032	42.290	0.999	0.001	32.150	0.992	0.009
Ficus	NeRFactor	6.442	22.402	0.928	0.085	21.664	0.919	0.095	20.684	0.907	0.107
	InvRender	4.884	25.335	0.942	0.072	22.131	0.934	0.057	20.330	0.895	0.073
	TensoIR	4.420	27.130	0.964	0.044	29.780	0.973	0.041	24.296	0.947	0.068
	Ours	6.409	31.889	0.909	0.147	27.794	0.965	0.043	24.501	0.943	0.077
Avg.	NeRFactor	6.314	25.125	0.940	0.109	24.679	0.922	0.120	23.383	0.908	0.131
	InvRender	5.074	27.341	0.933	0.100	27.367	0.934	0.089	23.973	0.901	0.101
	TensoIR	4.100	29.275	0.950	0.085	35.088	0.976	0.040	28.580	0.944	0.081
	Ours	5.817	29.386	0.930	0.091	34.902	0.989	0.015	27.229	0.973	0.038

Table 5. Per-scene results on the TensoIR synthetic datasets.

photographer.

## 15. Discussion on Limitations

In Fig. 8, we show an example of our limitations. In this case, since shadows are not well represented in our model, it causes ambiguity between the shadow and the geometry, thus generating some artifacts.

## 16. Per-Scene Results on the Synthetic Dataset

In Tab. 6, we provide the results for individual synthetic scenes mentioned in Sec. 4 of the main paper. Our method outperforms both baselines in all four scenes.

## 17. More Results

More results including relighting, material, normal reconstruction, information sharing ablation, and real dataset are shown in Fig. 8-12.

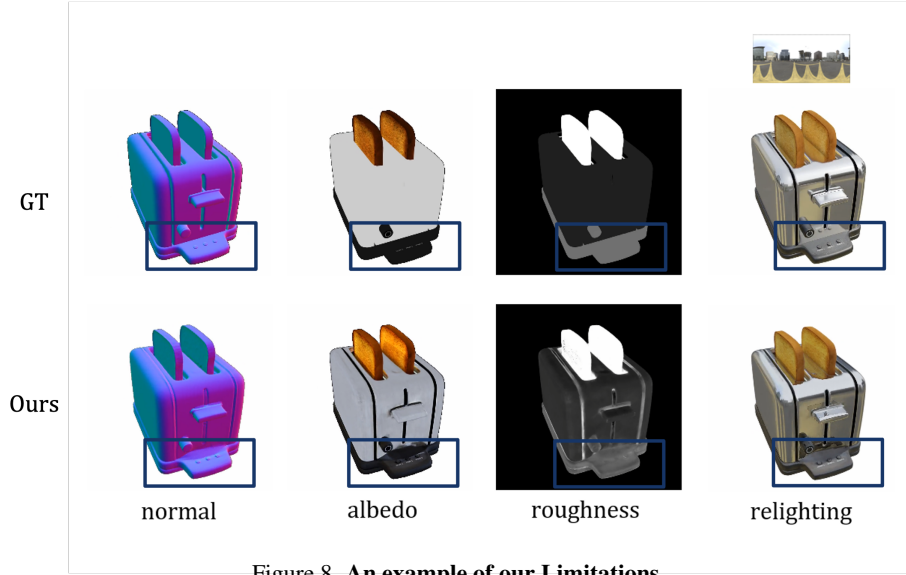


Figure 8. An example of our Limitations.

Scene	Method	Normal	Roughness	Albedo			Novel View Synthesis			Relighting		
		MAE ↓	PSNR ↑	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Teapot	NDR	5.43	28.32	31.47	0.98	0.02	33.17	0.98	0.02	26.78	0.93	0.03
	NDRMC	4.63	27.69	30.55	0.99	0.02	27.98	0.96	0.04	27.20	0.93	0.04
	NMF	5.18	26.25	19.42	0.76	0.21	33.39	0.98	0.02	24.41	0.96	0.03
	ENVIDR	1.61	-	-	-	-	37.16	0.98	0.02	29.22	0.97	0.03
	GShader	7.24	-	-	-	-	34.84	0.97	0.02	26.41	0.96	0.03
	NeRO	1.23	17.50	29.42	0.98	0.02	35.99	0.99	0.01	32.48	0.98	0.02
	Ours (full model)	0.78	38.86	34.26	0.99	0.01	38.55	0.99	0.01	32.55	0.99	0.02
	Ours (full model, 5 hrs)	0.76	38.61	34.41	0.99	0.01	39.15	0.99	0.01	32.46	0.99	0.03
Coffee	NDR	10.44	23.59	20.99	0.91	0.13	25.98	0.90	0.10	22.25	0.82	0.15
	NDRMC	8.44	23.60	22.43	0.95	0.08	23.93	0.87	0.15	23.34	0.86	0.14
	NMF	4.04	23.13	13.79	0.80	0.22	26.52	0.91	0.08	21.38	0.89	0.10
	ENVIDR	7.58	-	-	-	-	26.13	0.89	0.13	24.72	0.87	0.12
	GShader	7.06	-	-	-	-	26.83	0.91	0.11	23.59	0.90	0.10
	NeRO	3.13	23.22	19.01	0.92	0.12	27.10	0.94	0.09	25.88	0.92	0.09
	Ours (full model)	3.43	29.24	16.91	0.89	0.12	27.04	0.99	0.02	25.03	0.98	0.03
	Ours (full model, 5 hrs)	3.36	29.24	16.92	0.89	0.11	26.93	0.99	0.02	25.49	0.98	0.03
Car	NDR	5.37	22.02	20.19	0.92	0.11	29.73	0.95	0.04	22.46	0.86	0.09
	NDRMC	5.12	21.61	20.07	0.92	0.11	27.45	0.93	0.06	26.40	0.90	0.06
	NMF	2.90	26.09	14.36	0.85	0.13	31.31	0.96	0.02	23.31	0.93	0.04
	ENVIDR	3.32	-	-	-	-	31.55	0.95	0.05	26.43	0.93	0.06
	GShader	5.71	-	-	-	-	31.62	0.95	0.05	25.74	0.93	0.05
	NeRO	5.95	23.70	22.48	0.92	0.10	26.98	0.94	0.06	26.37	0.93	0.06
	Ours (full model)	2.43	25.65	25.14	0.95	0.16	32.06	0.99	0.01	28.20	0.99	0.03
	Ours (full model, 5 hrs)	2.22	27.74	25.39	0.95	0.04	32.65	0.99	0.01	28.32	0.99	0.03
Toaster	NDR	6.05	18.50	14.99	0.87	0.16	28.27	0.93	0.07	15.82	0.71	0.25
	NDRMC	4.47	18.15	16.27	0.89	0.12	25.29	0.89	0.14	22.13	0.85	0.15
	NMF	2.62	14.44	9.59	0.69	0.30	29.82	0.94	0.04	17.97	0.86	0.11
	ENVIDR	3.26	-	-	-	-	28.64	0.91	0.10	22.50	0.86	0.12
	GShader	4.88	-	-	-	-	28.47	0.92	0.10	21.30	0.88	0.11
	NeRO	2.16	14.99	19.43	0.88	0.19	29.27	0.94	0.08	25.70	0.91	0.09
	Ours (full model)	2.19	20.48	20.96	0.89	0.09	30.36	0.99	0.01	25.27	0.98	0.03
	Ours (full model, 5 hrs)	2.08	20.48	19.15	0.90	0.09	30.86	0.99	0.01	25.38	0.98	0.03
Helmet	NDR	2.09	24.83	24.07	0.92	0.09	29.41	0.94	0.08	17.98	0.76	0.20
	NDRMC	1.18	27.95	23.70	0.92	0.09	26.77	0.92	0.12	26.68	0.90	0.10
	NMF	0.78	20.57	14.03	0.69	0.19	31.54	0.96	0.03	20.88	0.89	0.08
	ENVIDR	0.89	-	-	-	-	34.02	0.95	0.06	24.75	0.91	0.07
	GShader	6.00	-	-	-	-	26.23	0.91	0.09	21.55	0.88	0.11
	NeRO	6.00	21.77	24.19	0.92	0.06	32.33	0.98	0.03	30.14	0.95	0.06
	Ours (full model)	0.52	33.07	33.92	0.97	0.01	32.25	0.99	0.01	29.43	0.99	0.03
	Ours (full model, 5 hrs)	0.48	34.31	33.92	0.98	0.01	32.87	0.99	0.01	29.43	0.99	0.02

Table 6. Per-scene results on the synthetic datasets.

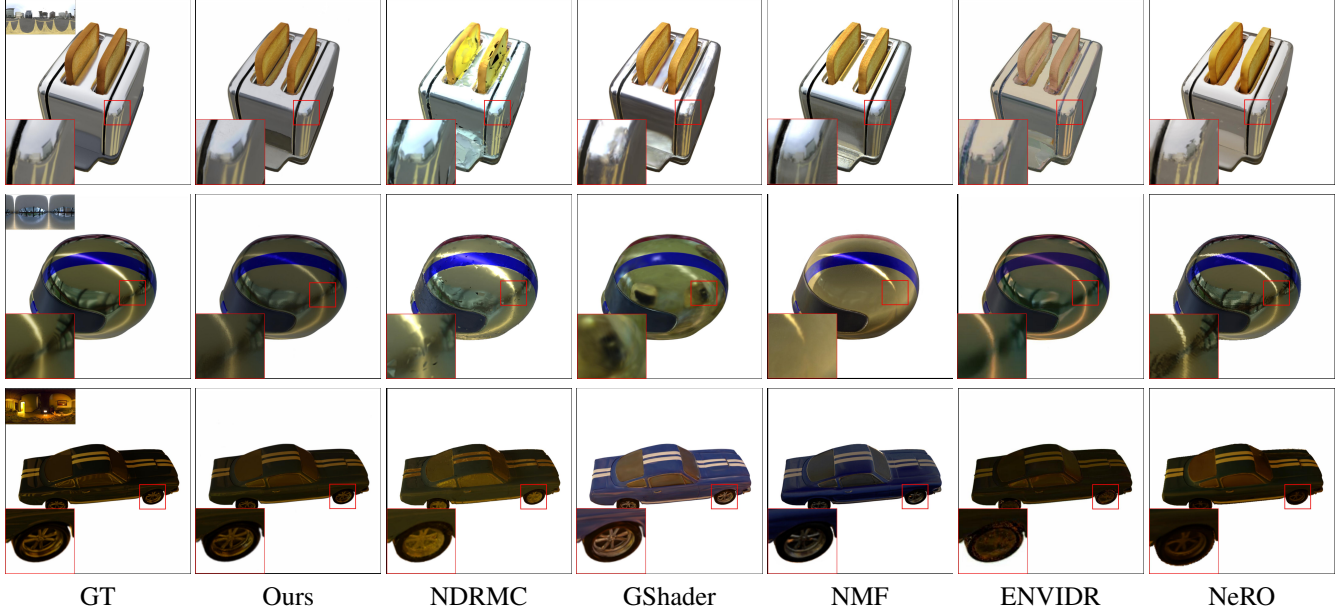


Figure 9. **Qualitative comparisons on relighted synthetic scenes. From top to bottom: toaster, helmet, car.** We can observe that other methods either have blurry, color-shifted results or aliasing, noisy effect under unseen illumination.

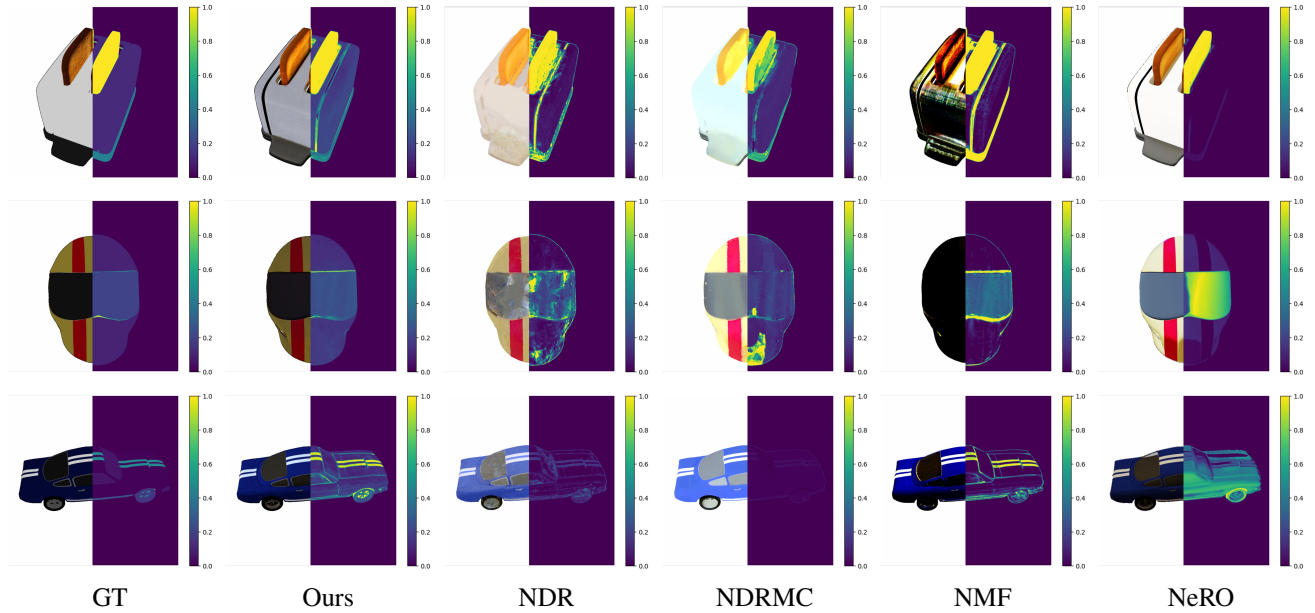


Figure 10. **Qualitative comparisons on material estimation. From top to bottom: toaster, helmet, car.** In each figure, we show albedo on the left and roughness on the right.



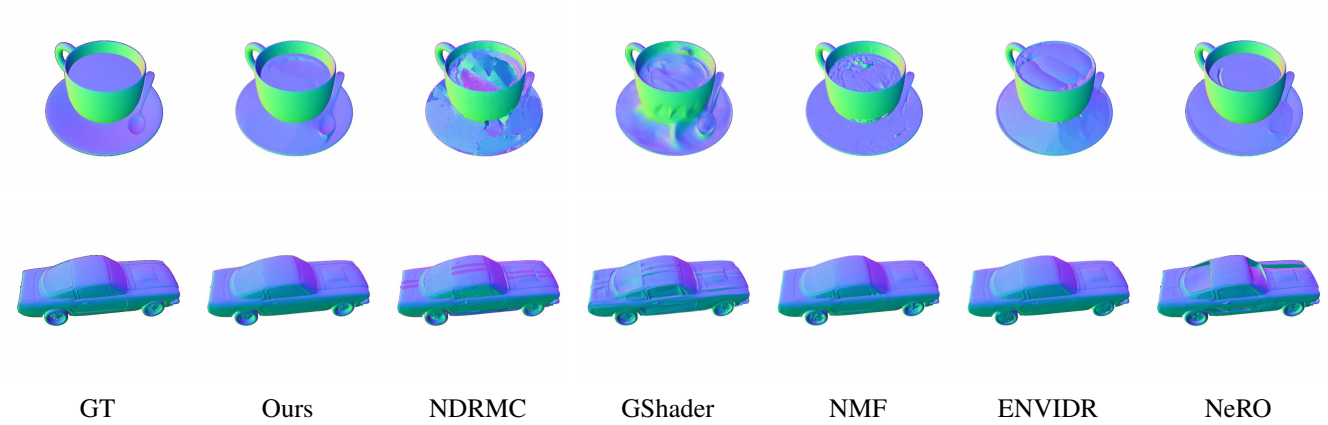


Figure 11. **Qualitative comparisons on normal reconstruction. From top to bottom: toaster, coffee, car.**

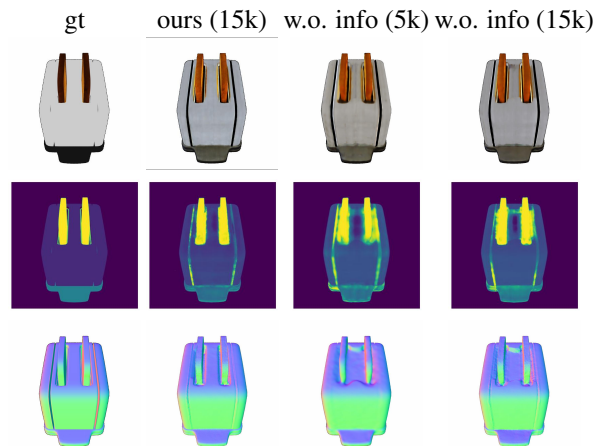


Figure 12. **Ablation studies.** We compare our full model with our model without information sharing (physically based rendering only).

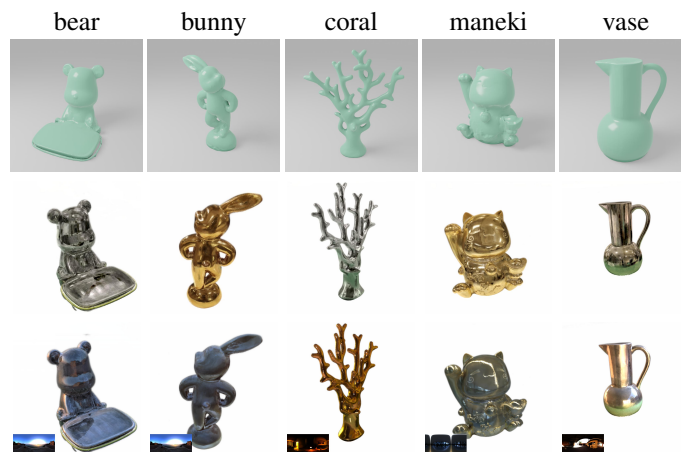


Figure 13. **Extracted mesh, Novel view synthesis and relighting of the Glossy-Real dataset.** Our model is trained for 4 hours with ground truth object mask. We extract meshes from SDF using marching cubes with a resolution of 512.