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

Deheng Zhang[2†*] Jingyu Wang[1*] Shaofei Wang[1] Marko Mihajlovic[1]
Sergey Prokudin[1] Hendrik P.A. Lensch[2] Siyu Tang[1]
[1]ETH Zürich [2]University of Tübingen

Abstract

Inverse rendering aims to reconstruct the 3D geometry, material parameters, and lighting conditions in a 3D scene from multi-view input images. To address this problem, some recent methods utilize a neural field combined with a physically based rendering model to reconstruct the scene parameters. Although these methods achieve impressive geometry reconstruction for glossy objects, the performance of material estimation and relighting remains limited. In this paper, we propose a novel end-to-end relightable neural inverse rendering system that achieves high-quality reconstruction of geometry and material properties, thus enabling high-quality relighting. The cornerstone of our method is a two-stage approach for learning a better factorization of scene parameters. In the first stage, we develop a reflection-aware radiance field using a neural signed distance field (SDF) as the geometry representation and deploy an MLP (multilayer perceptron) to estimate indirect illumination. In the second stage, we introduce a novel information-sharing network structure to jointly learn the radiance field and the physically based factorization of the scene. For the physically based factorization, to reduce the noise caused by Monte Carlo sampling, we apply a split-sum approximation with a simplified Disney BRDF and cube mipmap as the environment light representation. In the relighting phase, to enhance the quality of indirect illumination, we propose a second split-sum algorithm to trace secondary rays under the split-sum rendering framework. Furthermore, there is no dataset or protocol available to quantitatively evaluate the inverse rendering performance for glossy objects. To assess the quality of material reconstruction and relighting, we have created a new dataset with ground truth BRDF parameters and relighting results. Our experiments demonstrate that our algorithm achieves state-of-the-art performance in inverse rendering and relighting, with particularly strong results in the reconstruction of highly reflective objects.

1. Introduction

Inverse rendering is an ill-posed problem in computer graphics, which aims to reconstruct the 3D geometry, material parameters (texture, metallic, and roughness), as well as light conditions in a 3D scene given multi-view input images and corresponding camera poses. With the reconstructed assets, we can further change the light condition to render the object in new environments. This will serve as an essential step in 3D assets acquisition for the gaming and movie industries. To solve this problem, traditional inverse rendering algorithms [32] make the forward rendering process differentiable and optimize the scene parameters to achieve the required factorization. However, reconstructing complex real-world geometry remains a challenge within these pipelines. Recently, the neural radiance field (NeRF) [29] has demonstrated a superior ability to achieve novel view synthesis. Follow-up works such as NeuS [44], and VolSDF [50] further improve the quality of geometry reconstruction using neural signed distance functions. To improve the performance for glossy objects, Ref-NeRF [43] further changes the directional encoding to consider the reflected direction and roughness into the reconstruction pipeline. However, these methods only recover the geometry of the 3D object and do not reconstruct the material and light information in the scene, thus they can only achieve novel view synthesis without the ability to relight the object. Utilizing the neural radiance field as the geometry representation, neural inverse rendering methods [3, 15, 18, 22, 25, 28, 38, 41, 49, 54] further factorize the radiance field with physically based rendering formulation. However, none of the current algorithms simultaneously achieve high-quality scene parameter reconstruction, novel view synthesis, and relighting for glossy objects. Additionally, the currently available glossy object datasets do

*Equal contribution.
†The work was performed in ETH Zürich.
Project page: https://dehezhang2.github.io/RISE-SDF/

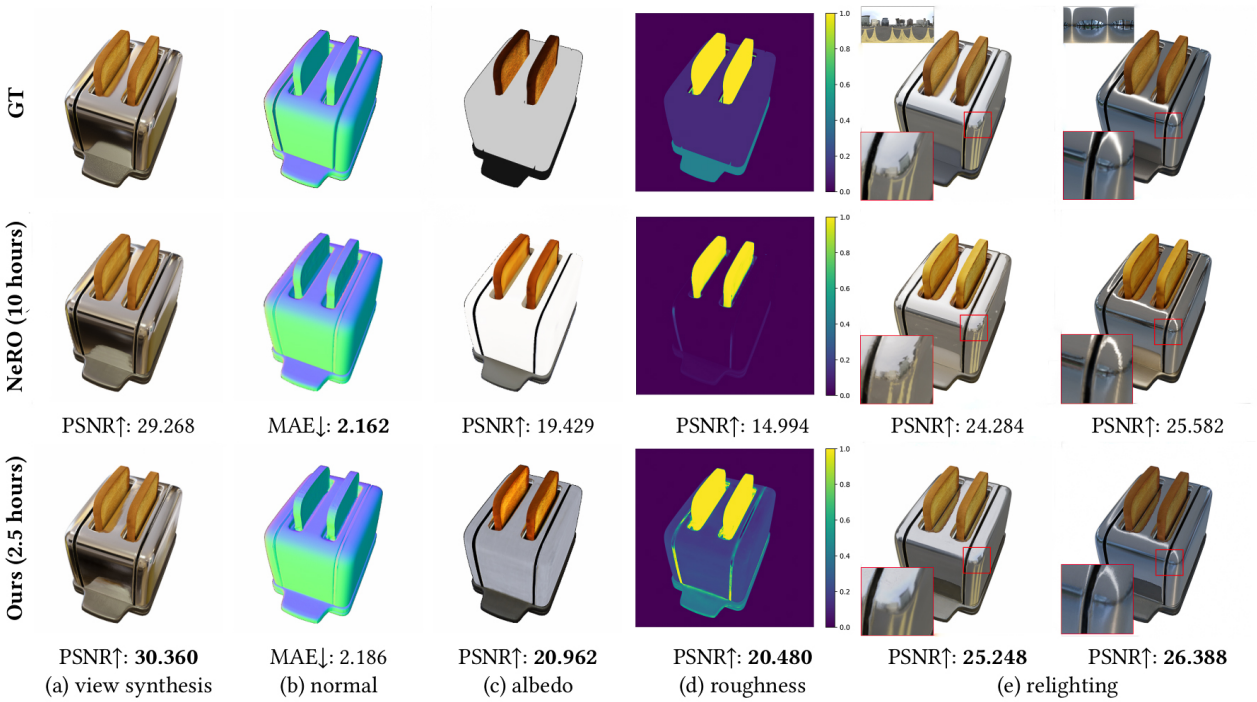


Figure 1. **RISE-SDF.** We present RISE-SDF, a method for reconstructing the geometry and material of glossy objects while achieving high-quality relighting. Our results, compared with the state-of-the-art method [25], show superior albedo and roughness estimation with significantly less training time. As an end-to-end relightable model, our algorithm generates high-quality relighting images without noise or aliasing.

not provide ground truth material parameters or relighting information, making it difficult to evaluate model performance.

In this paper, we propose an end-to-end relightable inverse rendering system for glossy objects. First, we explore how to construct high-quality geometry for glossy objects as an initial step in appearance parameter estimation. We find that an accelerated version of NeuS [44] with a reflection-aware radiance field yields the best results for glossy geometry. More specifically, we use three networks to predict the diffuse, specular direct illumination, the blending weights, and an additional MLP to predict the indirect illumination solely for the surface intersection. In stage two, we propose an information-sharing MLP structure to learn the appearance parameters and the environment light. Due to the observation that Monte Carlo integration generates noise, we adopt the split-sum approximation [16]. Our method uses a pre-integrated Disney BRDF [5], stored in a 2D lookup texture (LUT), as the physically based appearance model. Additionally, a differentiable multi-level mipmap serves as the environment map. We further train this representation jointly with the radiance field in stage one to maintain a high-quality geometry during the optimization. For the relighting, we propose a second, modified split-sum algorithm evaluated on secondary rays to achieve consistent relighting that accounts for indirect illumination in the scene. To evaluate our model's performance, we introduce a new synthetic dataset for glossy objects with ground truth relighting and material maps. Our results demonstrate that our algorithm not only excels in novel view synthesis and geometry reconstruction quality but also achieves state-of-the-art performance as an end-to-end relightable radiance field.

2. Related Work

2.1. Traditional Rendering

Forward Rendering. Physically based rendering (PBR) seeks to render images in a way that simulates light transport of scenes in real-world. It usually employs physically plausible bidirectional reflection distribution functions (BRDFs) to describe the way how light is scattered by object surfaces. Microfacet model [9] is the most widely used BRDF model for PBR. It simulates a macro surface as a collection of microfacets. Contrary to earlier BRDF models [34], the microfacet model is considered more physically plausible as it satisfies reciprocity and energy conservation. While there exists microfacet-based diffuse BRDF model [33], in this work we follow modern rendering pipelines [5, 16] and focus on specular microfacet BRDFs in which microfacets are assumed to be perfect mirrors. As an important component of the microfacet model, the normal distribution function (NDF) describes the distribution

of the orientations (*i.e.* normals) of microfacets on the surface. GGX [42] is one commonly used NDF because of its simplicity for evaluation. Popular PBR models such as the Disney BRDF [5] uses GGX model and variants as its core relfection models. Later, the split-sum approximation [16] is proposed to achieve real-time rendering in game engines. It introduces a multi-level mipmap to store pre-convolved environment light, such that it can be efficiently queried to render images without time-consuming Monte Carlo sampling. We also employ split-sum in our inverse rendering pipeline mainly due to its efficiency.

Inverse Rendering. Inverse rendering aims to recover the geometry, material, and light condition given the 2D images. In our work, we assume the camera pose is known, the light source is far away from the object, and the BRDFs are isotropic. The traditional inverse rendering algorithm [36] represents both BRDF and environment light with spherical harmonics. Later methods apply differentiable rasterization [7, 24, 26] or ray tracing [2, 20, 32] to reconstruct the material. However, Zeltner et al. [52] show that it is challenging to make Monte Carlo sampling differentiable without increasing the noise and suggests using a PDF proportional to the differentiation of the GGX distribution [42]. Our work finds that using the split-sum approximation [16] bypasses this problem and achieves noise-free reconstruction results.

2.2. Neural Radiance Field

Novel View Synthesis. NeRF [29] proposes a radiance field-based representation of the 3D scene. The method is trained using 2D multi-view images for which the camera pose is provided and uses two MLPs to predict the density σ and radiance $\mathbf{c}$ of a given sample on the ray. Then the volumetric rendering equation is applied to render the image for different views. Since the information is stored in 3D, view consistency is ensured. Variants of NeRF modify the structure to either improve the rendering quality or speed up the training of NeRF. For example, Ref-NeRF [43] uses the reflected direction of the camera ray as the input of the MLP to improve the rendering quality for specular materials. [47] further improve the glossy reconstruction by using a neural directional encoding framework to separate the direct and indirect illumination encoding. Chen et al. [6], Fridovich-Keil et al. [10], Müller et al. [30] modify the data structure of the radiance field to improve the training speed of NeRF. In our paper, we use instant-NGP [30] with the hash grid as our backbone model for radiance prediction, since this algorithm converges faster compared to the original NeRF. Furthermore, recently proposed Gaussian Splatting [17] utilizes 3D Gaussians with center and covariance matrix as the 3D representation, and uses rasterization-based rendering to achieve a higher reconstruction speed. However, since the normal information is not well reconstructed in this method, which is essential during inverse rendering, we do not apply this rendering technique as our backbone.

Neural Geometry Reconstruction. Based on the density representation in NeRF [29], some methods [44, 50] construct neural SDF fields with functions to transform the signed distance to density. Follow-up methods [21, 45, 56] further improve the speed and quality of the geometry reconstruction through hash encoding and progressive training. Yariv et al. [51] extend the reconstruction to unbounded scenes. In our paper, we use the pipeline proposed in [56] as the fundamental geometry representation and use the progressive training proposed in [21] to improve the quality.

2.3. Neural Inverse Rendering

Following the idea of implicit representation in NeRF [29], neural inverse rendering algorithms use MLPs to store the material information. NDR and NDRMC [13, 31] use an MLP to predict the BRDF material parameters, and DMTet [39] as the geometry representation. Compared with neural SDF, these methods result in discontinuous artifacts during the reconstruction. NeRFactor [54], NeRD [3], Neural-PIL [4], NeRV [41], Neural Transfer Field [27], InvRender [55] and TensoIR [15] use a density field as the geometry representation and an environment tensor with Monte Carlo sampling for the light reconstruction. To solve the ambiguity of the base color and environment light, [8, 19, 35] show the importance of adding a material prior to inverse rendering. However, these methods fail to reconstruct the material properties of glossy objects. To improve the performance for glossy objects, Neural microfacet fields (NMF) [28] use the microfacet model [9] with the GGX distribution [42] to improve the reconstruction quality, but the geometry and relighting quality are still sub-optimal as they use radiance representation and Monte Carlo sampling for rendering. To improve the geometry reconstruction, NeRO [25] and ENVIDR [22] use neural SDF as the geometry representation, but they are still limited in terms of relighting availability and quality. Their neural representation of the environment map means they can only achieve relighting via pre-distillation or export to the Blender [1] renderer, which results in aliasing or blurriness. Gaussian splatting-based inverse rendering [11, 14, 23, 40, 48] uses the 3D Gaussian as the geometry representation to improve the inverse rendering speed, but these methods either have bad reconstruction on glossy objects or have bad geometry due to the Gaussian representation. Our algorithm uses SDFs as the geometry representation and the split-sum approximation as the light representation. More importantly, to achieve better relighting quality, we apply a second split algorithm to trace indirect illumination. Compared with NeRO and ENVIDR, we can directly switch the environment map with an HDR file without any initialization or data export and achieve high-quality relighting.

3. Method

We first introduce the necessary concepts for volume rendering and physically based rendering (PBR) in Sec. 3.1. In Sec. 3.2, we describe the first stage of our framework that reconstructs the radiance field and the geometry. In Sec. 3.3, we combine volume rendering with traditional physically based split-sum approximation under the information-sharing MLP framework for material and lighting estimation. Furthermore, we introduce our proposed indirect illumination MLP and a novel second split-sum rendering algorithm in Sec. 3.4 to handle indirect illumination. This enables high-quality reconstruction and relighting with indirect illumination.

3.1. Preliminaries

Volume Rendering Model. The general volume rendering equation of NeRF is as follows:

$$\mathbf{C}(\mathbf{d}) = \sum\nolimits_{i=1}^{N} w_i \mathbf{c}_i, \text{ where } w_i = \Pi_{j=1}^{i-1}(1-\alpha_j)\alpha_i \quad (1)$$

here $\mathbf{d}$ represents the view direction, $\mathbf{c}_i$ represents the per-sample radiance. $\alpha_i = 1 - \exp(-\sigma_t^i)\delta_i$ where δ_i and σ_t^i represent interval length and density of sample i, respectively. Instead of using this density representation, we utilize NeuS [44] to better represent the underlying geometry as a neural signed distance field (SDF). The neural SDF defines a signed distance function $s : \mathbb{R}^3 \mapsto \mathbb{R}$, which is predicted by neural networks. The surface $\mathcal{S}$ is represented by the zero-level set of the neural SDF:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | s(\mathbf{x}) = 0\} \quad (2)$$

NeuS defines a S-density function $\phi_\sigma(\mathbf{x}) = \sigma e^{-\sigma \mathbf{x}}/(1 + e^{-\sigma \mathbf{x}})^2$ to convert the SDF value into density. The standard deviation of the density function is given by $1/\sigma$, which is learnable during the training process. To further make sure that the rendering equation is unbiased, NeuS introduces a normalized S-density as the rendering weight. As a result, the opacity α of the sample i becomes:

$$\alpha_i = \max\left(\frac{\Phi_\sigma(s(\mathbf{x}_i)) - \Phi_\sigma(s(\mathbf{x}_{i+1}))}{\Phi_\sigma(s(\mathbf{x}_i))}, 0\right) \quad (3)$$

Physically Based Rendering (PBR). In our PBR pipeline, we follow [28] and treat each sample during the volume rendering as a small local surface. Shading of each local surface follows the standard rendering equation:

$$\mathbf{c}_i^{pbr} = \int_{H^2} f_r(\mathbf{x}_i, \mathbf{d}, \omega_j) L_i(\mathbf{x}_i, \omega_j)(\mathbf{n}_i \cdot \omega_j) d\omega_j \quad (4)$$

where $-\mathbf{d}$ and ω_j are the outgoing and incoming directions (surface-to-camera direction and surface-to-light direction, respectively). This equation shows that the outgoing radiance $\mathbf{c}_i^{pbr}$ is the integration of the BRDF f_r multiplied by the incoming radiance and cosine term over the hemisphere defined by the surface normal $\mathbf{n}_i$. In our paper, the BRDF is defined as a simplified version of the Disney BRDF [5]:

$$f_r(\mathbf{x}_i, \mathbf{d}, \omega_j) = (1 - m_i)\frac{\mathbf{a}_i}{\pi} + \frac{D(\rho_i)F(m_i, \mathbf{a}_i)G}{4|\mathbf{d} \cdot \mathbf{n}_i||\omega_j \cdot \mathbf{n}_i|} \quad (5)$$

where albedo $\mathbf{a}_i$, metallic m_i, and roughness ρ_i are spatially varying material parameters, $\mathbf{n}_i$ is the surface normal. The first term of this equation represents the diffuse shading, while the second term corresponds to specular shading that is modeled by the microfacet model [9]. F stands for the Fresnel term. D is the normal distribution function (NDF) depending on roughness ρ. We use GGX [42] for D. G is the shadowing term, which models the shadowing effect between microfacets. In this work, we use the split-sum approximation [16] due to its efficiency in estimating the integral in Eq. (4). It approximates the radiance by pre-integration such that only a lookup per pixel is required to evaluate the reflected radiance finally:

$$\begin{aligned}\mathbf{c}_i^{pbr}(\mathbf{x}_i, \mathbf{d}) \approx &\left(\frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} \frac{f_r(\mathbf{x}_i, \mathbf{d}, \omega_j)(\mathbf{n}_i \cdot \omega_j)}{p(\omega_j; \hat{\mathbf{d}}_i, \rho_i)}\right) \\ &\cdot \left(\frac{1}{N_{mc}} \sum_{j=1}^{N_{mc}} L_i(\mathbf{x}_i, \omega_j)\right)\end{aligned} \quad (6)$$

given the sample location $\mathbf{x}_i$, ray direction $\mathbf{d}$, normal $\mathbf{n}_i$, and reflected ray direction $\hat{\mathbf{d}}_i$ w.r.t the normal. N_{mc} is the number of samples during pre-integration and p represents the probability of sampling the pair of directions.

3.2. Stage 1: Geometry Reconstruction

Geometry Representation. We use instant NGP (iNGP) [30], which consists of a multi-level hash encoding and a small multi-layer perceptron (MLP), to predict the neural SDF. As shown in Fig. 2, we use a progressive hash grid γ followed by a geometry MLP f_{geo} to predict the SDF value s and a geometry feature β:

$$s_i, \beta_i = f_{geo}(\gamma(\mathbf{x}_i)) \quad (7)$$

Following Eq. (1) and Eq. (3), we can compute the volume rendering weight w_i and normal vector $\mathbf{n}_i = \frac{\nabla_{\mathbf{x}_i} s_i}{||\nabla_{\mathbf{x}_i} s_i||}$ for each sample.

Radiance Representation. The direct illumination is estimated using the spatial location $\mathbf{x}_i$ and reflected direction $\hat{\mathbf{d}}$ as inputs akin to Verbin et al. [43]. The reflected direction is computed as:

$$\hat{\mathbf{d}}_i = -2(\mathbf{d} \cdot \mathbf{n}_i)\mathbf{n}_i + \mathbf{d} \quad (8)$$

The encodings $e(\mathbf{x}_i)$ and $e(\hat{\mathbf{d}}_i)$ are respectively computed via positional and spherical harmonic encodings.

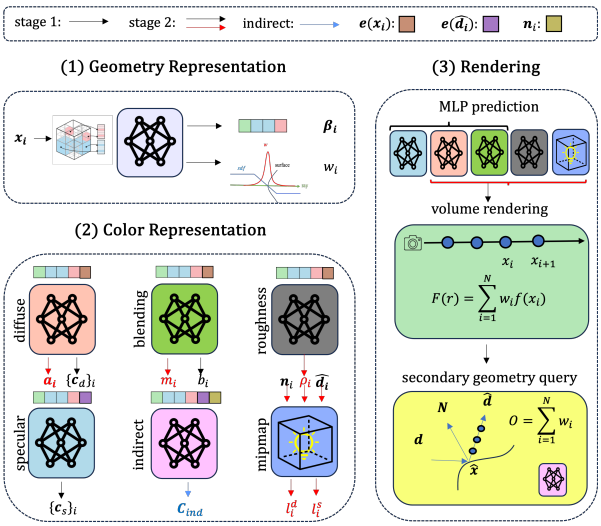


Figure 2. **Our pipeline. The colors of the features in the figure indicate different feature concatenation combinations.** (1) Given the location $\boldsymbol{x}_i$, the progressive hash grid with the geometry MLP predicts the geometry feature β_i and the corresponding volume rendering weight w_i via Eq. 1 and 3. (2) For the color representation, separate networks predict per-sample albedo $\boldsymbol{a}_i$, metallic m_i, and roughness ρ_i. They share the information between the direct volume rendering pipeline (black arrow) and the physically based rendering pipeline (red arrow). (3) The per-sample values (all except the blue value) are rendered via volume rendering. Then we compute the expected surface intersection $\hat{\boldsymbol{x}}$ and trace another ray to compute the occlusion probability O. Finally, the direct and indirect colors are blended via O.

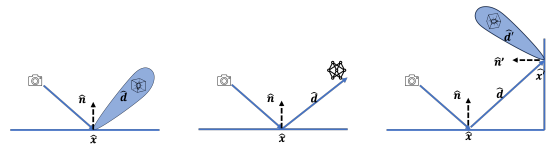
We define three MLPs, f_{diff}, f_{spec}, f_{blend}, and predict the diffuse color $\{\mathbf{c}_d\}_i = f_{diff}(e(\mathbf{x}_i))$, specular color $\{\mathbf{c}_s\}_i = f_{spec}(e(\hat{\mathbf{d}}_i))$, and the blending weight $b_i = f_{blend}(e(\mathbf{x}_i), \beta_i)$. We extend the original volume rendering equation Eq. (1) for these values as a function of sample location $\mathbf{x}_i$:

$$\begin{aligned} \{\mathbf{c}_d\}_i &= (1 - b_i) * \{\mathbf{c}_d\}_i; \quad \{\mathbf{c}_s\}_i = b_i * \{\mathbf{c}_s\}_i \\ \{\mathbf{C}_d, \mathbf{C}_s, \mathbf{N}, T\} &= \sum\nolimits_{i=1}^{N} w_i \{\mathbf{c}_d, \mathbf{c}_s, \mathbf{n}, t\}_i \end{aligned} \tag{9}$$

Note that t_i are the sample values during ray marching. In this way, we get the diffuse radiance $\mathbf{C}_d$ and specular radiance $\mathbf{C}_s$ under direct illumination. More details are discussed in the supplementary. We will describe how to model indirect illumination in Sec. 3.4

3.3. Stage 2: Material and Light Estimation

In Stage 2, we optimize materials and environment lighting via differentiable PBR, given the geometry and radiance field initialized in Stage 1. Note that in stage 2, geometry and the radiance field are optimized jointly with PBR parameters, contrary to [25, 27, 55]. The PBR appearance $\mathbf{c}^{pbr} = \mathbf{c}_d^{pbr} + \mathbf{c}_s^{pbr}$ is computed via the split-sum approximation as follows:



(a) 1st split-sum (b) indirect MLP (c) 2nd split-sum

Figure 3. **Different Illumination Inference Method**. (a) During training and relighting, we use the first split-sum to compute the direct illumination. (b) During training, we use an MLP to predict indirect illumination and blend with direct illumination using the occlusion probability. (c) During relighting, we use a second split-sum with one additional ray bounce to compute the indirect illumination.

$$\begin{aligned} \{\mathbf{c}_d^{pbr}\}_i &= (1 - m_i) * \mathbf{a}_i * \mathbf{l}_i^d(\mathbf{n}_i) \\ \{\mathbf{c}_s^{pbr}\}_i &= (F_0(m_i, \mathbf{a}_i) * F_1(\rho_i, \mathbf{n}_i \hat{\mathbf{d}}) \\ &\quad + F_2(\rho_i, \mathbf{n}_i \hat{\mathbf{d}})) * \mathbf{l}_i^s(\rho_i, \hat{\mathbf{d}}_i) \end{aligned} \tag{10}$$

where F_1, F_2 are integral values only dependent on the roughness and cosine between the normal and reflection direction, $\mathbf{l}_i^d, \mathbf{l}_i^s$ are light integrals approximated by a differentiable mipmap. More details and derivation of the split-sum approximation are discussed in the supplementary. Finally, we apply volume rendering for the diffuse and specular appearance of Stage 2:

$$\{\mathbf{C}_d^{pbr}, \mathbf{C}_s^{pbr}\} = \sum\nolimits_{i=1}^{N} w_i \{\mathbf{c}_d^{pbr}, \mathbf{c}_s^{pbr}\}_i$$

Information Sharing between Stage 1 & 2. Comparing Eq. (10) and Eq. (9), we note that both blending weight b_i and metallic m_i are used to mix the diffuse and specular colors. Besides, diffuse color c_i^d and albedo $\mathbf{a}_i$ are both view-independent and low-entropy. We thus propose to share this information by expanding the output dimension of the MLP in Stage 1: $\{\mathbf{c}_i^d, \mathbf{a}_i\} = f_{diff}(e(\mathbf{x}_i); \beta_i), \{b_i, m_i\} = f_{blend}(e(\mathbf{x}_i), \beta_i)$, and create a new MLP for the roughness prediction $\rho_i = f_{rough}(e(\mathbf{x}_i), \beta_i)$. We use MLP to predict all parameters in both stages and only optimize the albedo and metallic parameters in stage 2. As shown in Fig. 2, we jointly optimize the PBR loss and the radiance field loss from Stage 1. This MLP sharing technique provides a novel way to initialize the material MLP, thus giving more stable material prediction as shown in supplementary Fig. 6.

3.4. Modeling Indirect Illumination

So far we have excluded indirect illumination from our model, whereas in practice modeling indirect illumination is crucial for reconstructing and rendering realistic scenes, especially when the scene is highly specular. We thus introduce two novel techniques to model indirect illumination effects: indirect illumination MLP for training and second split-sum for test-time relighting.

Indirect Illumination MLP. Observing that indirect illumination is often most relevant for the specular component of the incoming radiance, we add a residual term to refine the specular radiance in Stage 1 (Sec. 3.2). Note that we only estimate the indirect illumination once for one primary ray for efficiency. Specifically, we compute the expected depth values T, which is used to compute the expected surface intersection $\hat{\mathbf{x}} = \mathbf{o} + T\mathbf{d}$. Then, we use this surface location as input to geometry models $\{\gamma, f_{geo}\}$ to compute the corresponding feature $\beta_{\hat{\mathbf{x}}}$. As shown in Fig. 2 and Fig. 3(b), to better model the indirect illumination, we use another MLP f_{ind} to predict the indirect illumination $\mathbf{C}_{ind} = f_{ind}(\mathbf{N}, e(\hat{\mathbf{d}}), \beta_{\hat{x}})$, where $\hat{\mathbf{d}}$ is the reflected secondary ray direction along the expected normal $\mathbf{N}$. After that, we apply the ray marching along the secondary ray direction $\hat{\mathbf{d}}$ only to compute the volume rendering weights w_i and do not query the color MLPs. We use the weights to compute the occlusion probability $O = \sum_{i=1}^{N} w_i$ and use this value to blend the specular part as a weighted sum of direct and indirect illumination:

$$\begin{aligned} \mathbf{C} &= \mathbf{C}_d + (1-O) * \mathbf{C}_s + O * \mathbf{C}_{ind} \\ \mathbf{C}^{pbr} &= \mathbf{C}_d^{pbr} + (1-O) * \mathbf{C}_s^{pbr} + O * \mathbf{C}_{ind} \end{aligned} \tag{11}$$

More details are discussed in the supplementary.

Second Split-sum. During the relighting stage, we cannot reuse the learned indirect illumination MLP for unseen light conditions. Therefore, we propose a novel second split-sum (Fig. 3 (c)) for relighting. Unlike Monte Carlo rendering, which requires tracing multiple rays, we trace a single ray and use the split-sum to approximate integration over the reflection lobe. We compute the expected intersection $\hat{\mathbf{x}}$ and secondary direction $\hat{\mathbf{d}}$. Given the illumination term in Eq. (4), we further divide the incident radiance into direct and indirect illumination. As shown in [22], the indirect illumination mainly has a strong effect with a small roughness value on the specular part of the light. So we set a threshold ρ_t and only trace secondary rays for pixels with smaller roughness values:

$$\begin{aligned} L_i =& \mathbb{1}[\rho > \rho_t] L_{dir} \\ &+ \mathbb{1}[\rho \leq \rho_t]((1-O) * L_{dir} + O * L_{ind}) \end{aligned} \tag{12}$$

where $\mathbb{1}[.]$ is an indicator function that returns one if the condition holds and zero otherwise. Plugging this equation into the split-sum equation (in the appendix), we derive the following adjustment equation for the light integral during relighting:

$$\begin{aligned} \mathbf{l}^s_{relight} \approx& (\mathbb{1}[\rho > \rho_t] + \mathbb{1}[\rho \leq \rho_t] * (1-O)) * \mathbf{l}^s \\ &+ \mathbb{1}[\rho \leq \rho_t] * O * L_{ind}(\hat{\mathbf{x}}, \hat{\mathbf{d}}) \end{aligned} \tag{13}$$

where ρ and $\mathbf{l}^s$ are computed by volume rendering the per-sample roughness ρ_i and per-sample light integral $\mathbf{l}^s_i$. We give the full derivation in supplementary. Applying ray marching along the secondary ray $\{\hat{\mathbf{x}}, \hat{\mathbf{d}}\}$ the expected second surface intersection $\hat{\mathbf{x}'}$ is computed. The indirect light L_{ind} is evaluated by following Eq. (10):

$$L_{ind}(\hat{\mathbf{x}}, \hat{\mathbf{d}}) = \mathbf{c}_d^{pbr}(\hat{\mathbf{x}'}, \hat{\mathbf{d}}) + \mathbf{c}_s^{pbr}(\hat{\mathbf{x}'}, \hat{\mathbf{d}}) \tag{14}$$

Training Losses. To improve the geometry quality, we further apply the progressive hash grid and finite differences for SDF gradient estimation in the Eikonal [21] and curvature [37] losses. Finally, we compute the photometric loss, and define the whole optimization loss function:

$$\begin{aligned} &\mathcal{L}_c = ||\mathbf{C} - \mathbf{C}_{gt}||_2^2; \;\; \mathcal{L}_{pbr} = ||\mathbf{C}^{pbr} - \mathbf{C}_{gt}||_2^2 \\ &\mathcal{L}_{stage\ 1} = \lambda_c \mathcal{L}_c + \lambda_{eik}\mathcal{L}_{eik} + \lambda_{curv}\mathcal{L}_{curv} \\ &\mathcal{L}_{stage\ 2} = \lambda_c \mathcal{L}_c + \lambda_c \mathcal{L}_{pbr} + \lambda_{eik}\mathcal{L}_{eik} + \lambda_{curv}\mathcal{L}_{curv} \end{aligned} \tag{15}$$

4. Experiments

4.1. Experiment Protocol

Baselines. We compare our method with baselines focusing on inverse rendering for glossy objects. These methods include DMTet-based methods NDR [31] and NDRMC [13], NeRF-based method NMF [28], neural SDF-based methods NeRO [25], ENVIDR [22], and 3DGS-based method GaussianShader [14]. Note that not all of these methods are end-to-end relightable. NeRO [25] and ENVIDR [22] use neural representation for the environment map, thus their relighting requires additional neural network training per environment map. We build our own shiny inverse rendering dataset as discussed in the supplementary material, and use NeRO [25] real dataset to evaluate the models.

Evaluation Protocol. During relighting evaluation, for NeRO [25], NDR [31], and NDRMC [13], we use Blender with SPP=4096 and 4 bounces of lights. For ENVIDR [22], we first pre-train and distill the environment maps to the MLP. Due to the inevitable ambiguity between the albedo and the environment map, we align each channel of the predicted albedo, roughness, and relighting with the ground truth before evaluation. We use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [46], and Learned Perceptual Image Patch Similarity (LPIPS) [53] as metrics to evaluate the novel view synthesis, albedo, and relighting. Roughness is evaluated using only PSNR, and the normals are evaluated using Mean Angle Error (MAE). For all the values, we calculate the average among all the test views in five scenes. For the relighting metrics, we calculate the average under all the different environment maps.

4.2. Results

Shiny Inverse Rendering. We first compare the quantitative metrics on our shiny inverse rendering dataset. We

Method	PB	E2ER	Normal	Roughness	Albedo			Novel View Synthesis			Relighting			Runtime
			MAE ↓	PSNR ↑	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	
NDR [31]	✓	✓	5.877	23.452	22.340	0.921	0.102	29.312	0.939	0.062	21.057	0.817	0.145	2 hrs
NDRMC [13]	✓	✓	4.767	23.782	22.602	0.935	0.084	26.280	0.914	0.100	25.149	0.890	0.100	2 hrs
NMF [28]	✓	✓	3.110	22.099	14.238	0.755	0.211	30.516	0.950	0.037	21.590	0.904	0.072	5 hrs
ENVIDR [22]	✗	✗	3.330	-	-	-	-	31.500	0.936	0.072	25.524	0.907	0.080	6 hrs*
GShader [14]	✗	✓	6.179	-	-	-	-	29.598	0.934	0.075	23.718	0.910	0.081	1.5 hrs
NeRO [25]	✓	✗	3.694	20.237	22.907	0.925	0.097	30.331	0.958	0.054	28.116	0.980	0.031	> 10 hrs
Ours (w.o. info-share)	✓	✓	2.191	28.216	25.634	0.931	0.080	31.220	0.988	0.016	27.252	0.984	0.031	2 hrs
Ours (w.o. indirect MLP)	✓	✓	1.947	28.310	25.170	0.925	0.084	30.451	0.990	0.018	27.300	0.984	0.030	2.5 hrs
Ours (with Monte-Carlo)	✓	✓	2.540	25.865	22.800	0.881	0.105	29.482	0.925	0.072	23.766	0.876	0.126	6 hrs
Ours (w.o. sec. split-sum)	✓	✓	-	-	-	-	-	-	-	-	27.230	0.984	0.030	-
Ours(full model)	✓	✓	1.869	29.460	25.839	0.941	0.078	32.050	0.993	0.012	28.096	0.985	0.030	2.5 hrs
Ours (full model, 5 hrs)	✓	✓	1.780	30.076	25.958	0.943	0.052	32.493	0.994	0.011	28.215	0.986	0.028	5 hrs

Table 1. **Quantitative comparisons on the shiny inverse rendering synthetic dataset.** PB: Physically Based. E2ER: End to End Relightable. ENVIDR runtime: ENVIDR requires 2 additional hours for distillation. Our results have significantly outperformed the baseline methods by reconstructing more accurate geometry, and material, thus achieving high-quality novel view synthesis and relighting results.

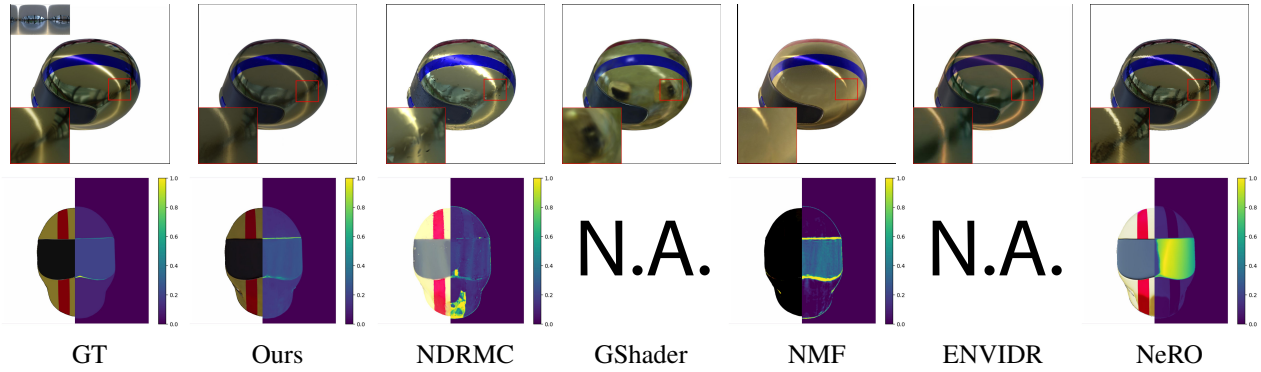


Figure 4. **Qualitative comparisons on the helmet scene. From top to bottom: relighting, material.** We can observe that other methods either have blurry, color-shifted results or aliasing, noisy effect under unseen illumination. And our material estimation also outperform other baselines.

categorize different methods according to whether they are physically-based models with valid BRDF modeling, and whether they are end-to-end relightable or not. We only evaluate the roughness and albedo metrics for physically-based models. The category and metrics are listed in Table 1. Our method shows superior performance compared with previous methods on all metrics. We also show the qualitative results for relighting and material estimation in Fig. 4. More results can be found in the supplementary material.

Relighting. As shown in supplementary Fig. 9, we visualize the relighting results under different unseen light conditions during training. We can observe that NDRMC [13] and GShader [14] are limited by their low-quality geometry reconstructions. NMF [28] generates excessive noise due to the Monte Carlo Sampling. ENVIDR [22] generates blurry shifted colors due to the distillation of the HDR environment map to the MLP. NeRO [25] causes aliasing artifacts when exporting to Blender, which is a required step of their model for relighting.

Material & geometry reconstruction. The normal estimation is displayed in supplementary Fig. 11, while the albedo and roughness are in supplementary Fig. 10. Our method has the best quality on the material estimation thanks to our information-sharing architecture. For normal reconstruction, we can observe that NDRMC [13] and GShader [14] produce artifacts due to limitations of surface-based-rendering/3DGS on geometry reconstruction. NMF [28] has small bumps due to the instability of volume density field representation and Monte Carlo sampling. For SDF-based methods, ENVIDR [22] fails to reconstruct the geometry when indirect illumination appears, and NeRO [25] tends to reconstruct the over-smoothed surface.

Real Dataset. Further evaluation is carried out on the Glossy Real dataset from NeRO [25]. Following NeRO, we qualitatively show novel view synthesis, geometry reconstruction, and relighting results. We also quantitatively evaluate the Chamfer distance of the extracted meshes.

The per-scene Chamfer distance scores are listed in Table 2. The results of extracted meshes, novel-view-synthesis, and relighting are shown in supplementary Fig. 13. We achieve comparable performance to NeRO in geometry reconstruction. Our method falls short in material estimation because the assumption of a distant environment map is violated. Please refer to the appendix for a more detailed discussion.

	NDR*	NDRMC*	NeuS	NeRO	Ours(4hrs)*
Bunny	0.0047	0.0042	0.0022	0.0012	0.0018
Coral	0.0025	0.0022	0.0016	0.0014	0.0016
Maneki	0.0148	0.0117	0.0091	0.0024	0.0023
Bear	0.0104	0.0118	0.0074	0.0033	0.0095
Vase	0.0201	0.0058	0.0101	0.0011	0.0012
Avg.	0.0105	0.0071	0.0061	0.0019	0.0032

Table 2. **Geometry reconstruction quality in Chamfer Distance (CD↓) on the Glossy-Real dataset.** NDR, NDRMC, and our method use ground truth object mask.

4.3. Ablation Studies

Information Sharing. We compared our full pipeline's material and normal estimation against a version without information sharing in supplementary Fig. 13. While Stage 1 (1k iterations) in the full pipeline optimizes only the radiance field, we assessed the non-information-sharing model at both 5k iterations, matching material optimization time, and 15k iterations, aligning with geometry optimization time of the full model. Results indicate that information sharing enhances both geometry and material reconstruction, showing that extended training without information sharing still underperforms compared to the full model with less training iterations.

Indirect illumination. In Fig. 5, we visualize the relighting results using the fireplace environment map for the toaster scene using our full model, our model without indirect illumination prediction MLP, and our model without the second split-sum during relighting. The fully implemented model closely replicates the ground truth in terms of lighting and reflection accuracy. Models omitting indirect illumination and the second split-sum relighting show significant degradation. The former displays reduced realism in reflections, while the latter exhibits unnatural contrasts due to the indirect MLP trained in the different light conditions.

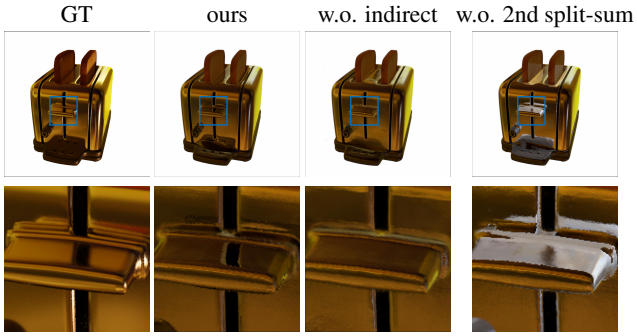


Figure 5. **Ablation study for indirect illumination.**

Monte Carlo vs. Split-sum. In our pipeline, we use the split-sum approximation instead of Monte Carlo sampling to avoid sampling noise. In Fig. 6, we compare the relighting results and the corresponding error map with respect to the ground truth using the split-sum approximation, Monte Carlo rendering using multiple-importance sampling (MIS) with 128 spp and 256 spp respectively. We can observe that with split-sum our rendering results have significantly less noise compared with Monte Carlo sampling.

4.4. Limitations

Although we achieve superior performance on the reconstruction, inverse rendering, and relighting of shiny objects, our system still has the following limitations. Our system is built on an isotropic BRDF model, which means we cannot handle transparent or anisotropic materials. Furthermore, although our approach can handle indirect illumination for reflective surfaces, shadows for diffuse surfaces are not well solved. Thus, our model shows sub-optimal performance point light environments. Furthermore, due to the inevitable ambiguity problem between the illumination, material parameter, and geometry, the estimated material could still deviate strongly from the ground truth in some cases.

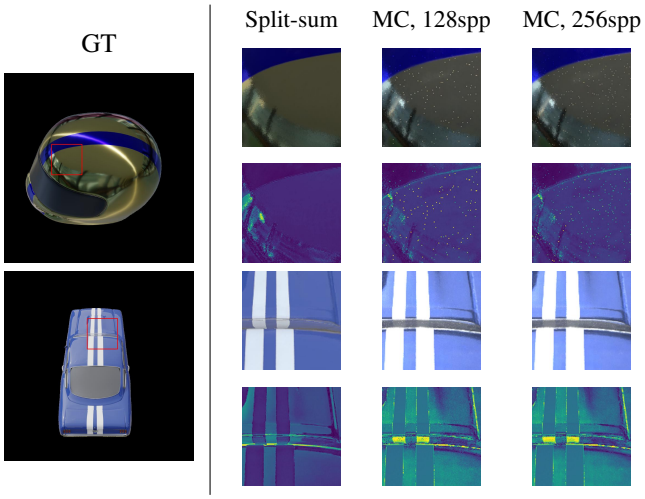


Figure 6. **Ablation study for Monte Carlo integration**. We visualize the relighting results for helmet and musclecar scene. Odd rows: relightings, even rows: error maps (same color map as we use for roughness maps).

5. Conclusion

We presented RISE-SDF, an end-to-end relightable inverse rendering system for glossy objects. One key idea of RISE-SDF is information-sharing between the radiance-based signed-distance field and the physically based inverse rendering field to improve the geometry and material estimation. We further introduced indirect illumination prediction during training and a second split-sum during relighting to improve the reconstruction and relighting quality on interreflections. Experiments demonstrate that our system has superior novel view synthesis, material reconstruction, geometry reconstruction, and relighting quality compared with the previous state-of-the-art.

References

[1] Blender. https://www.blender.org/. 3, 1

[2] Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 3

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *Int. Conf. Comput. Vis. (ICCV)*, 2021. 1, 3

[4] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P. A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2021. 3

[5] Brent Burley. Physically-based shading at disney. In *Proc. of SIGGRAPH*, 2012. 2, 3, 4

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Eur. Conf. Comput. Vis. (ECCV)*, 2022. 3

[7] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2019. 3

[8] Xi Chen, Sida Peng, Dongchen Yang, Yuan Liu, Bowen Pan, Chengfei Lv, and Xiaowei Zhou. Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination. *arxiv:2404.11593*, 2024. 3

[9] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Trans. on Graphics*, 1 (1):7–24, 1982. 2, 3, 4

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 3

[11] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024. 3

[12] Yuan-Chen Guo. Instant neural surface reconstruction. https://github.com/bennyguo/instant-nsr-pl, 2022. 1

[13] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light, and material decomposition from images using monte carlo rendering and denoising. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022. 3, 6, 7

[14] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2024. 3, 6, 7

[15] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023. 1, 3, 2

[16] Brian Karis. Real shading in unreal engine 4. In *Proc. of SIGGRAPH*, 2013. 2, 3, 4

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics*, 42(4): 139:1–139:14, 2023. 3

[18] Julian Knodt, Joe Bartusek, Seung-Hwan Baek, and Felix Heide. Neural ray-tracing: Learning surfaces and reflectance for relighting and view synthesis. *arxiv:2104.13562*, 2021. 1

[19] Georgios Kouros, Minye Wu, Sushruth Nagesh, Xianling Zhang, and Tinne Tuytelaars. Unveiling the ambiguity in neural inverse rendering: A parameter compensation analysis. *arxiv:2404.12819*, 2024. 3

[20] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. on Graphics*, 37(6):222:1–222:11, 2018. 3

[21] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023. 3, 6

[22] Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. Envidr: Implicit differentiable renderer with neural environment lighting. In *Int. Conf. Comput. Vis. (ICCV)*, 2023. 1, 3, 6, 7, 2

[23] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2024. 3

[24] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Int. Conf. Comput. Vis. (ICCV)*, 2019. 3

[25] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Trans. on Graphics*, 42(4): 114:1–114:22, 2023. 1, 2, 3, 5, 6, 7

[26] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *Eur. Conf. Comput. Vis. (ECCV)*, 2014. 3

[27] Linjie Lyu, Ayush Tewari, Thomas Leimkuehler, Marc Habermann, and Christian Theobalt. Neural radiance transfer fields for relightable novel-view synthesis with global illumination. In *Eur. Conf. Comput. Vis. (ECCV)*, 2022. 3, 5

[28] Alexander Mai, Dor Verbin, Falko Kuester, and Sara Fridovich-Keil. Neural microfacet fields for inverse rendering. In *Int. Conf. Comput. Vis. (ICCV)*, 2023. 1, 3, 4, 6, 7

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis. (ECCV)*, 2020. 1, 3

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 41(4): 102:1–102:15, 2022. 3, 4

[31] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting

from images. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 3, 6, 7, 2

[32] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. on Graphics*, 38(6):203:1–203:17, 2019. 1, 3

[33] Michael Oren and Shree K. Nayar. Generalization of lambert's reflectance model. In *Proc. of SIGGRAPH*, 1994. 2

[34] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975. 2

[35] Yohan Poirier-Ginter, Alban Gauthier, Julien Philip, Jean-François Lalonde, and George Drettakis. A Diffusion Approach to Radiance Field Relighting using Multi-Illumination Synthesis. *Computer Graphics Forum*, 2024. 3

[36] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proc. of SIGGRAPH*, 2001. 3

[37] Radu Alexandru Rosu and Sven Behnke. Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023. 6

[38] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Int. Conf. Comput. Vis. (ICCV)*, 2019. 1

[39] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2021. 3

[40] Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jingtuo Liu, Liangjun Zhang, Jian Zhang, Bin Zhou, Errui Ding, and Jingdong Wang. Gir: 3d gaussian inverse rendering for relightable scene factorization. *arxiv:2312.05133*, 2023. 3

[41] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021. 1, 3

[42] TS Trowbridge and Karl P Reitz. Average irregularity representation of a rough surface for ray reflection. *Journal of the Optical Society of America(JOSA)*, 65(5):531–536, 1975. 3, 4

[43] 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 *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 1, 3, 4

[44] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2021. 1, 2, 3, 4

[45] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Int. Conf. Comput. Vis. (ICCV)*, 2023. 3

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612, 2004. 6

[47] Liwen Wu, Sai Bi, Zexiang Xu, Fujun Luan, Kai Zhang, Iliyan Georgiev, Kalyan Sunkavalli, and Ravi Ramamoorthi. Neural directional encoding for efficient and accurate view-dependent appearance modeling. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2024. 3

[48] Tong Wu, Jia-Mu Sun, Yu-Kun Lai, Yuewen Ma, Leif Kobbelt, and Lin Gao. Deferredgs: Decoupled and editable gaussian splatting with deferred shading. *arxiv:2404.09412*, 2024. 3

[49] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K Wong. PS-NeRF: Neural inverse rendering for multi-view photometric stereo. In *Eur. Conf. Comput. Vis. (ECCV)*, 2022. 1

[50] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2021. 1, 3

[51] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In *Proc. of SIGGRAPH*, 2023. 3

[52] Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. Monte carlo estimators for differential light transport. *ACM Trans. on Graphics*, 40(4):78:1–78:16, 2021. 3

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 6

[54] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. on Graphics*, 40(6): 237:1–237:18, 2021. 1, 3

[55] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 3, 5

[56] Fuqiang Zhao, Yuheng Jiang, Kaixin Yao, Jiakai Zhang, Liao Wang, Haizhao Dai, Yuhui Zhong, Yingliang Zhang, Minye Wu, Lan Xu, and Jingyi Yu. Human performance modeling and rendering via neural animated mesh. *ACM Trans. on Graphics*, 41(6):235:1–235:17, 2022. 3