NeuS-PIR: Learning Relightable Neural Surface using Pre-Integrated Rendering

Shi Mao
KAUST
Thuwal, Saudi Arabia
shi.mao@kaust.edu.sa

Chenming Wu
Baidu Research
Beijing, China
wuchenming@baidu.com

Zhelun Shen
Baidu Research
Beijing, China
shenzhelun@baidu.com

Yifan Wang
University of Washington
Seattle, USA
yifan1@cs.washington.edu

Dayan Wu
IIE, CAS
Beijing, China
wudayan@iie.ac.cn

Liangjun Zhang
Baidu Research
Beijing, China
liangjunzhang@baidu.com

Abstract

In this paper, we propose NeuS-PIR, a novel approach for learning relightable neural surfaces using pre-integrated rendering from multi-view image observations. Unlike traditional methods based on NeRF or discrete mesh representations, our approach employs an implicit neural surface representation to reconstruct high-quality geometry. This representation enables the factorization of the radiance field into two components: a spatially-varying material field and an all-frequency lighting model. By jointly optimizing this factorization with a differentiable pre-integrated rendering framework and material encoding regularization, our method effectively addresses the ambiguity in geometry reconstruction, leading to improved disentanglement and refinement of scene properties. Furthermore, we introduce a technique to distill indirect illumination fields, capturing complex lighting effects such as interreflections. As a result, NeuS-PIR enables advanced applications like relighting, which can be seamlessly integrated into modern graphics engines. Extensive qualitative and quantitative experiments on both synthetic and real datasets demonstrate that NeuS-PIR outperforms existing methods across various tasks. The source code is included in the supplementary material and will be released publicly upon acceptance.

Keywords: Inverse Rendering, Pre-integrated Rendering, Neural Implicit Representation

1. Introduction

Recovering an object's geometry, material properties, and illumination from captured images – commonly known as inverse rendering – has long been a challenging problem in computer vision and graphics. These recovered properties are crucial for a wide range of applications, including view synthesis, relighting, and object insertion [30]. However, the task of inverse rendering is inherently difficult due to its underconstrained nature. To mitigate this challenge, various approaches have been proposed, often relying on additional inputs such as scanned geometry, predefined lighting conditions, multiple images captured under different lighting setups, or simplifying assumptions like uniform material properties for the object [3, 2, 46].

The emerging trend of neural representations has demonstrated remarkable capabilities in geometry reconstruction [39] and novel view synthesis [28]. In these neural

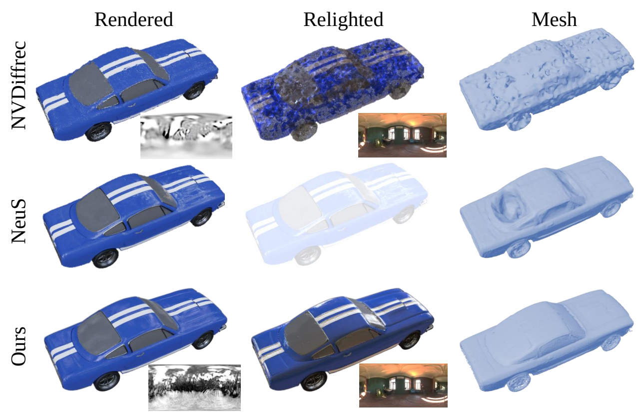

Figure 1: Comparison of results from our proposed method (bottom row) and NVDiffrec [30] (top row). Our method simultaneously learns geometry, material, and illumination within the neural implicit field. The relighted images and reconstructed geometry produced by our approach show significant improvements over NVDiffrec. Building on NeuS [39], which focuses on geometry reconstruction without factorization, our method benefits from the joint learning of material and illumination, allowing it to better preserve geometry, particularly in highly reflective regions.

representation-based methods, material properties and illumination are often intertwined. Consequently, they cannot be directly applied to tasks like relighting or material editing. While there have been attempts to decouple neural representations [48, 6, 35], major limitations are preventing them from being readily used for relighting. As NeRF-based methods model geometry as a volume density field without sufficient surface constraints, it contains artifacts that may not be noticeable in view synthesis but problematic in extracted high-quality surfaces with reliable normal. *Signed Distance Function* (SDF) is introduced into volume rendering to supervise the geometry depiction explicitly and has achieved improved results [39, 31, 42]. However, these methods focus more on geometry reconstruction rather than inverse rendering. Other methods like NVDiffrec [30] adopt hybrid approach that uses implicit SDF field and explicit mesh and proposes an efficient differential rendering pipeline for reconstruction, but such geometry representation might lead to inaccurate topology due to predefined SDF-grid. Overall, existing inverse rendering approaches [4, 48, 30] mostly focus on decomposing material and illumination while geometry quality is not prioritized.

In this work, we introduce **NeuS-PIR**, a novel framework that jointly recovers high-fidelity geometry, realistic material properties, and accurate all-frequency illumination from multi-view images, as shown in Fig. 1. Our approach builds on the implicit geometry representation of NeuS, extending it with a new **pre-integrated rendering (PIR)** network to enable simultaneous optimization of geometry, material, and lighting. By leveraging the **smooth surface reconstruction** capabilities of NeuS [39], we ensure stable and feasible joint optimization of geometry and appearance. Unlike methods such as NVDiffrec, which rely on discretized mesh representations that often struggle to reconstruct smooth surfaces due to reliance on good initialization, our implicit representation eliminates these initialization dependencies and produces more precise surface normals. To handle complex illumination, NeuS-PIR advances pre-integrated rendering by learning from a high-frequency environment map (a differentiable $6 \times 512 \times 512$ cube map), enabling superior reconstruction of shiny and reflective surfaces. In contrast, existing methods like Neural-PIL and PhySG often rely on low-frequency illumination representations to reduce computational costs, resulting in a loss of detail and degraded illumination quality. Additionally, we introduce a **radiance field** that guides the learning process, supported by customized regularization terms on geometry, lighting, and material properties to ensure stable and robust training. Moreover, NeuS-PIR distills an **indirect illumination field** from the learned representations, improving the modeling of complex lighting effects such as interreflections. All components are jointly optimized through a differentiable pre-integrated rendering framework, leading to state-of-the-art performance in tasks such as relighting and object insertion. In summary, our main technical contributions are:

- a framework that leverages a neural implicit surface and pre-integrated rendering to factorize the scene into geometry, material, and illumination, leading to less degraded geometry and better disentanglement, thereby supporting relighting,
- a joint optimizing scheme using re-computed environment map with material encoding regularization that encourages sparsity and consistency, which enables all-frequency illumination recovery with improved generalization ability compared to data-driven latent-space methods [6, 48], yet can be optimized efficiently, and
- a modular distillation method to distil indirect illuminations field from learned representations by jointly optimizing it with the direct illumination, which further addresses the complex lighting effect like interreflection.

2. Related Work

2.1. Multi-view Reconstruction

Explicit Reconstruction from images can be achieved through various methods, including triangle meshes [24], tetrahedral meshes [30], voxels [23], hierarchical octrees [15], atlas surfaces [14], or hybrid explicit/implicit representations [33]. A key advantage of explicit representations is their compatibility with downstream rendering in industrial engines. However, optimizing explicit representations is often sensitive to hyperparameters and initialization, which can sometimes lead to reconstruction failures due to topological inconsistencies. In contrast, neural implicit fields tend to offer more stable optimization processes.

Neural Implicit Field Modeling leverages neural networks to represent spatial fields for modeling 3D objects and scenes. Neural Radiance Fields (NeRF) [28], along with its variants (e.g., MipNeRF [1]), employ coordinate-based MLPs to encode volumetric radiance fields. These models allow querying the geometry and color of arbitrary points in space by integrating values along casting rays, enabling high-quality novel view synthesis. However, NeRF-like methods do not directly optimize the underlying geometry.

Recent advancements in neural surface reconstruction have introduced frameworks that optimize geometry using photometric losses. For instance, UniSurf [31] refines the sampling distribution to better align the volumetric field with the surface, while VolSDF [42] transforms NeRF's density function into a learnable Signed Distance Function (SDF), sampling points along casting rays based on opacity error bounds. NeuS [39] and its successors [40, 26] provide unbiased, occlusion-aware solutions by converting NeRF's density function into an SDF.

While these techniques can reconstruct plausible geometry and render high-quality novel views, material properties and illumination are baked into the model, making it difficult to adapt to different background scenes. Our work extends NeuS by introducing material and illumination factorization, enabling downstream applications such as relighting and object insertion.

2.2. Material and Illumination Estimation

Material and Illumination Estimation for reconstructed objects is a complex task, with many prior approaches requiring known lighting conditions for inverse rendering. Deep Reflectance Volumes [3] and Neural Reflectance Field [2] leverage differentiable volume ray marching frameworks to supervise the reconstruction of neural reflectance volumes and reflectance fields, respectively. Deferred Neural Lighting [12] applies deferred rendering using proxy geometry and neural texture, followed by neural rendering, to enable free-viewpoint relighting. Another mesh-based method [27] jointly optimizes mesh and SVBRDF using a differentiable renderer specialized for collocated configurations. IRON [46] proposes a two-stage approach where a signed distance field (SDF) is first used to recover geometry, followed by material optimization. These methods rely on photometric images, which require a more involved data capture process. In contrast, our approach jointly optimizes geometry, material, and illumination using only input images.

Inverse Rendering with Multi-View Estimation has seen recent progress, allowing for material and illumination estimation from more relaxed multi-view settings. NeRFactor [48] uses a set of MLPs to model light source visibility, normal maps, surface albedo, and material properties at surface points, building on a pre-trained NeRF. PhySG [47] assumes the scene is under fixed illumination, while NeRD [4] extends this to both fixed and varying illumination conditions. Neural-PIL [6] introduces a neural pre-integrated lighting method to replace spherical Gaussians, enabling the estimation of high-frequency lighting details. Ref-NeRF [37] replaces MipNeRF's [1] view-dependent outgoing radiance parametrization with reflected radiance to model environment lighting and surface roughness. NeILF [41] models illumination with a fully 5D light field, naturally handling occlusions and indirect lighting, while NeILF++ [45] removes the need for pre-reconstructed geometry.

Rather than relying solely on implicit field representations, NVDiffrec [30] adopts a hybrid approach, combining an implicit SDF field with an explicit mesh and proposing an efficient differentiable rendering pipeline for reconstruction. NVDiffrecmc [16] builds on NVDiffrec's representation but incorporates ray tracing and Monte Carlo integration for more realistic shading. TensoIR [19] opts for tensor factorization and neural fields, moving away from MLP-based fields. ENVIDR [22] introduces an implicit differentiable renderer for inverse rendering, achieving strong performance, but its integration with modern graphics pipelines—such as relighting with off-the-shelf HDR environment maps—remains challenging. Concurrently, several works have emerged. NeRO [25] uses a similar geometry representation and learning strategy, focusing on reflective objects, while our approach is designed for inverse rendering of general objects. NeuralPBIR [36] proposes a two-stage pipeline that first reconstructs an imperfect geometry and then applies physics-based inverse rendering for higher-quality factorization. However, their method simplifies BRDF rendering by disregarding metallic parameters, which limits its applicability to highly reflective surfaces, such as the toaster shown in Fig. 6 in our work.

Recent advances in 3D Gaussian Splatting (3DGS) [21] have introduced new representations and factorization techniques [11], which significantly facilitate inverse rendering [17, 13, 34]. However, Gaussian splatting is an underdefined explicit geometry, and mesh representations remain more suitable for geometric manipulation tasks such as deformation.

3. Methodology

Given a collection of multi-view images and their corresponding camera poses, our goal is to reconstruct the object's geometry and its surrounding environment's illumination. Specifically, our factorization approach learns an implicit representation of the object's geometry and material properties, as well as the illumination independent of the object. As illustrated in Fig. 2, for each point $\boldsymbol{x} \in \mathbb{R}^3$, our model outputs several properties: the *signed distance field* (SDF) value $s \in \mathbb{R}$, the diffuse albedo $\boldsymbol{k_d} \in [0,1]^3$, the roughness $r \in [0,1]$, the metallic parameter $m \in [0,1]$, and the environment illumination, represented by a cubemap $I \in \mathbb{R}^{H \times W \times 6}$. Once training is complete, our method enables the rendering of reconstructed objects under various environment maps, a process known as *relighting*. This can be achieved using either the jointly estimated environment maps or externally provided ones. The rendering process is governed by the principles of volume rendering. Additionally, high-quality surface meshes with material properties can be extracted from the learned representation using standard mesh extraction algorithms, such as the marching cubes algorithm or its variants, facilitating downstream applications.

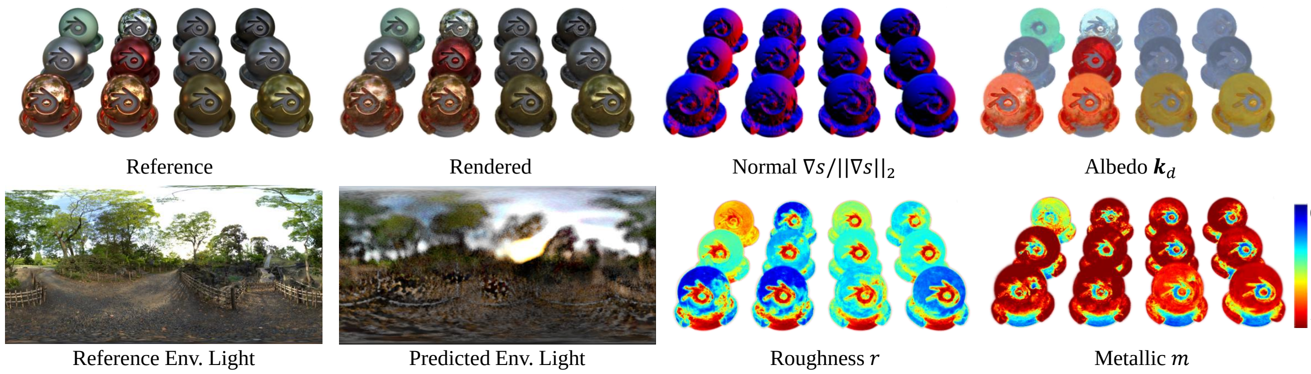

Figure 2: Our method factorizes a scene into geometry, material, and illumination. We display the reference and predicted environment illumination as a latitude-longitude converted environment cubemap. The roughness and metallic are visualized using a jet color map ranging from 0 to 1.

3.1. Prerequisites

3.1.1 NeuS for Geometry Modeling

Our method adopts Neural Implicit Surfaces (NeuS) [39] to represent an object's geometry, leveraging its ability to reconstruct high-quality surfaces as the zero-level set of an implicit SDF representation. NeuS employs a multi-layer perceptron (MLP) to learn both the SDF function f_{sdf} : $\boldsymbol{x} \mapsto s$, which maps a 3D position $\boldsymbol{x} \in \mathbb{R}^3$ to an SDF value $s \in \mathbb{R}$, and the appearance function f_{color} : $(\boldsymbol{x}, \boldsymbol{v}) \mapsto \boldsymbol{c}$, which maps a 3D position $\boldsymbol{x} \in \mathbb{R}^3$ and a viewing direction $\boldsymbol{v} \in \mathbb{S}^2$ to the corresponding RGB radiance $L \in [0,1]^3$.

NeuS renders images by accumulating radiance along rays cast through each pixel, following the standard volume rendering approach. Specifically, consider a pixel ray parameterized as $\{\boldsymbol{x}(t) = \boldsymbol{o} - t\boldsymbol{v} \mid t \geq 0\}$, where $\boldsymbol{o} \in \mathbb{R}^3$ is the camera origin, and $\boldsymbol{v} \in \mathbb{S}^2$ is the normalized direction pointing toward the camera center. The accumulated color for this pixel, $\boldsymbol{c}_p$, is computed as a weighted sum of the radiance along the ray:

$$\boldsymbol{c}_p(\boldsymbol{o}, \boldsymbol{v}) = \int_{t_n}^{t_f} w(t) L\left(\boldsymbol{x}(t), \boldsymbol{v}\right) \mathrm{d}t, \tag{1}$$

where $w(t)$ is a non-negative weight function, and the integration is performed from the near plane t_n to the far plane t_f of the camera. To ensure unbiased and occlusion-aware rendering, NeuS derives the weight function from the SDF as:

$$w(t) = \exp\left(-\int_0^t \rho(u)\mathrm{d}u\right)\rho(t), \tag{2}$$

where $\rho(t) = \max\left(\frac{-\mathrm{d}\Phi_\tau}{\mathrm{d}t}\left(s(t)\right)/\Phi_s\left(s(t)\right), 0\right)$ is referred to as the opaque density, and $\Phi_\tau(s) = (1 + e^{-\tau s})^{-1}$ is the Sigmoid function scaled by a factor τ. The learned factor τ is inversely proportional to the standard deviation of the density distribution near the zero level of the SDF. During training, the value of $1/\tau$ is expected to converge to zero as the zero-valued isosurfaces of the SDF progressively approach solid surfaces.

To enable relighting and material factorization, we model the outgoing radiance along the ray cast from a pixel using two branches. The first branch directly predicts the outgoing radiance based on the position, viewing direction, and surface normal using a radiance MLP. The second, material-aware branch captures material and illumination properties and computes the outgoing radiance using a split-sum approximation, as detailed in Sec. 3.1.2. Both branches share the same SDF module to ensure geometric consistency during training.

3.1.2 PIL for Material and Illumination Modeling

We adopt image-based lighting as our lighting model to decompose the radiance field into geometry, material, and lighting components, approximating the rendering equation with pre-integrated rendering. Following [20], the specular term in the rendering equation can be efficiently approximated using the *split-sum* approximation:

$$\int_\Omega L_i(\boldsymbol{l}) f_s(\boldsymbol{l}, \boldsymbol{v})(\boldsymbol{l} \cdot \boldsymbol{n})\mathrm{d}\boldsymbol{l} = I(\boldsymbol{r}; r) \int_\Omega f_s(\boldsymbol{l}, \boldsymbol{v})(\boldsymbol{l} \cdot \boldsymbol{n})\mathrm{d}\boldsymbol{l}, \tag{3}$$

where $L_i(\boldsymbol{l})$ represents the incident radiance from direction $\boldsymbol{l}$, $f_s(\boldsymbol{l}, \boldsymbol{v}; r, m)$ is the Cook-Torrance microfacet specular BRDF [10] parameterized by roughness r and metallic m, and $\boldsymbol{n}$ is the surface normal vector. The first term in the split-sum approximation, $I(\boldsymbol{r}; r)$, involves the importance sampling of the incident light radiance, modulated by the surface roughness r. Although this approximation incurs a minor loss in reflection accuracy at grazing angles, it allows us to pre-integrate this term from the environment map and query it using the reflection direction $\boldsymbol{r} = 2(\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{n} - \boldsymbol{v}$ as:

$$I(\boldsymbol{r}; r) = \int_{\Omega} L_i(\boldsymbol{l}) D(\boldsymbol{l}, \boldsymbol{r}; r)(\boldsymbol{l} \cdot \boldsymbol{r}) \mathrm{d}\boldsymbol{l}, \tag{4}$$

where $D(\boldsymbol{l}, \boldsymbol{v}; r)$ represents the GGX normal distribution function [38], which models the portion of microfacets that reflect light toward the viewer and is defined by the roughness r. The pre-integrated illumination is stored as mipmap levels of a learnable environment cubemap.

The second term of the split-sum approximation is independent of illumination and corresponds to integrating the specular BRDF f_s in a constant brightness environment. Using Schlick's approximation, the specular reflectance at normal incidence, F_0, can be factored out. Thus, the second term can be rewritten as F_0 modulated by scale and bias terms, which depend only on the material's roughness and the cosine between the viewing direction and the surface normal $(\boldsymbol{v} \cdot \boldsymbol{n})$:

$$\int_{\Omega} f_s(\boldsymbol{l}, \boldsymbol{v})(\boldsymbol{l} \cdot \boldsymbol{n}) \mathrm{d}\boldsymbol{l} = F_0 S\left((\boldsymbol{v} \cdot \boldsymbol{n}), r\right) + B\left((\boldsymbol{v} \cdot \boldsymbol{n}), r\right), \tag{5}$$

where the scale S and bias B can be pre-computed and stored in 2D lookup tables (LUTs) for efficient inference, as in [20]. We follow the common convention of setting F_0 as an interpolation between 0.04 (the specular reflectance of non-metallic materials) and the diffuse color $\boldsymbol{k}_d$ (the specular reflectance of metallic materials), based on the material's metallic value m:

$$F_0 = 0.04 \times (1 - m) + m\boldsymbol{k}_d. \tag{6}$$

Finally, the direct outgoing radiance $L_{dir}(\boldsymbol{v})$ observed from the viewing direction $\boldsymbol{v}$ is a blend of diffuse and specular components, formulated as:

$$L_{dir}(\boldsymbol{v}) = \boldsymbol{k}_d I_d + I_s\left(F_0 S\left((\boldsymbol{v} \cdot \boldsymbol{n}), r\right) + B\left((\boldsymbol{v} \cdot \boldsymbol{n}), r\right)\right), \tag{7}$$

where $I_d = I(\boldsymbol{n}; 1)$ represents the diffuse irradiance, and $I_s = I(\boldsymbol{r}; r)$ represents the specular irradiance. For further details on pre-integrated rendering, we refer readers to the presentation by UE4 [20].

3.2. Network Architecture

Fig. 3 illustrates the architecture of our proposed method. The SDF MLP learns the scene geometry, while the Radiance MLP captures the radiance field at a coarse level, using the geometry features from the SDF MLP and viewing directions. Following the pre-integrated rendering principle, further decompositions factorize the radiance into material and lighting components.

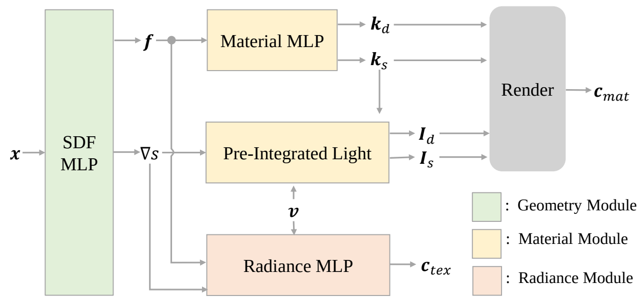

Figure 3: Network architecture of our proposed method. The SDF MLP learns geometry, and the radiance MLP learns the radiance field at a coarse level. Further decompositions factorize the material and illumination using the Material MLP and Pre-Integrated Light.

Geometry Module. The SDF MLP takes 3D positions $\boldsymbol{x}$ as input and outputs the feature $\boldsymbol{f}(\boldsymbol{x})$. The first channel of this feature represents the SDF value $s(\boldsymbol{x})$, and its gradient $\nabla s(\boldsymbol{x})$ is computed analytically. For positional encoding, we use trainable multi-resolution grids, efficiently supported by hashtables [29]. The feature is learned through an MLP.

Radiance Module. To initiate the training of the scene's geometry, the view-dependent color for each point is determined using an MLP. The radiance MLP takes as input the viewing direction $\boldsymbol{v}$, the positional feature $\boldsymbol{f}(\boldsymbol{x})$, and the unit surface normal $\nabla s(\boldsymbol{x}) / \|\nabla s(\boldsymbol{x})\|_2$. The viewing direction is encoded using spherical harmonics up to the 4th level. Finally, the output color $\boldsymbol{c}_{tex}$ is computed and integrated using Eq. 1.

Material Module. The material module decomposes the view-dependent outgoing radiance by modeling it as incident light modulated by surface material properties. The Material MLP, denoted as $Mat(\boldsymbol{f})$, takes the SDF feature $\boldsymbol{f}(\boldsymbol{x})$ as input and predicts the surface diffuse albedo $\boldsymbol{k}_d \in [0, 1]^3$, metallic value $m \in [0, 1]$, and roughness $r \in [0, 1]$. Following Munkberg et al. [30], the Material MLP also predicts a general occlusion term $o \in [0, 1]$, which accounts for indirect illumination and shadowing by modulating the outgoing radiance by $1 - o$. The occlusion term can be refined and distilled from the learned representations during the indirect illumination distillation process, as detailed in Sec. 3.4. The specular and occlusion properties of the material are collectively denoted as $\boldsymbol{k}_s = \{o, r, m\}$. Both $\boldsymbol{k}_d$ and $\boldsymbol{k}_s$ are learned through MLP layers, with sigmoid activations to constrain their values to the range $[0, 1]$.

To render the outgoing radiance under pre-integrated illumination, we follow the approach of Munkberg et al. [30], using a high-resolution cubemap as trainable parameters and pre-integrating $I(\boldsymbol{r}; r)$ for discrete roughness levels, stored as mipmaps. For a given roughness r, the value $I(\boldsymbol{r}; r)$ is retrieved using mipmap interpolation. The view-dependent radiance is then rendered using Eq. 7 and transformed into the sRGB space through gamma correction. Finally, the pixel color $\boldsymbol{c}_{mat}$ is volume-rendered using Eq. 1.

3.3. Loss and Regularization

We employ Mean Square Error (MSE), L1 loss, and binary cross-entropy (BCE) loss as supervision signals for the masked rendered images. Additionally, we introduce regularization terms for SDF, material, and lighting. Specifically, the image color loss is defined as:

$$\mathcal{L}_{\hat{\boldsymbol{c}}} = \lambda_{c1}||\hat{\boldsymbol{c}} - \boldsymbol{c}||_1 + \lambda_{c2}||\hat{\boldsymbol{c}} - \boldsymbol{c}||_2, \tag{8}$$

where $\hat{\boldsymbol{c}}$ represents the volume-rendered pixel color from either the material module $\boldsymbol{c}_{mat}$ or the texture module $\boldsymbol{c}_{tex}$. The mask loss is defined as:

$$\mathcal{L}_{mask} = \lambda_{mask}\mathrm{BCE}(mask, opa), \tag{9}$$

where BCE is the binary cross-entropy between the image mask and the accumulated opaque density along pixel rays.

To regularize the SDF field, we use both Eikonal and sparsity terms:

$$\mathcal{L}_{sdf} = \lambda_{se}||\nabla s - 1||_2^2 + \lambda_{ss}\exp(-\lambda_{sa}|s|_1). \tag{10}$$

The Eikonal term encourages the gradient of the SDF field to have unit length, while the sparsity term promotes a sparse distribution of the zero-crossing level set of the SDF. These regularization terms help in learning smooth and sparse solid surfaces.

Material estimation is challenging due to limited observations of each surface point. We mitigate this issue by introducing the prior that objects are generally composed of a limited number of distinct materials. Following this principle, we regularize the material representation in both feature space—promoting smoothness and sparsity—and image space—promoting local consistency.

In feature space, we adopt a material feature loss inspired by [50], regularizing the material-related latent feature f to be sparse, and ensuring that the mapping of the Material MLP is smooth with respect to small changes in the latent feature space:

$$\begin{aligned}\mathcal{L}_{mat}^{F} =& \lambda_{mf1}\sum_{i=2}^{F} D_{KL}(\mathrm{Bern}(0.05)||p(f_i)) \\ &+ \lambda_{mf2}||\mathrm{Mat}(\boldsymbol{f}) - \mathrm{Mat}(\boldsymbol{f} + \Delta\boldsymbol{f})||_1,\end{aligned} \tag{11}$$

where $p(\boldsymbol{f}_i)$ is the mean value of the i-th channel of the positional feature $\boldsymbol{f}$, representing the probability of non-zero values. The sparsity loss minimizes the KL-divergence between $p(\boldsymbol{f}_i)$ and a target Bernoulli distribution with probability 0.05, encouraging sparsity in the material features.

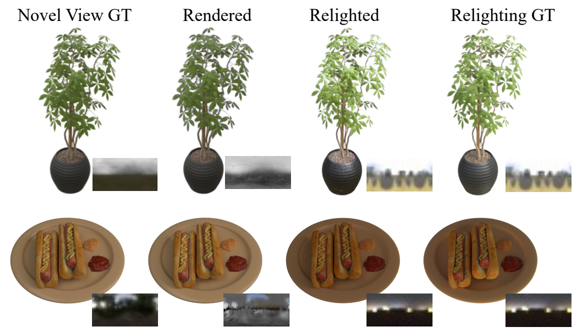

Figure 4: Novel view synthesis and relighting results produced by our proposed method.

The smoothness loss ensures that similar latent features (differing by a small $\Delta\boldsymbol{f} \sim \mathcal{N}(0, \epsilon)$) are mapped to similar material parameters via the Material MLP.

In image space, we sample half of our rays using a patch-based method to encourage local similarity in roughness and metallic properties. Additionally, we regularize the amplitude of the occlusion term. The image-space material loss is defined as:

$$\begin{aligned}\mathcal{L}_{mat}^{I} =& \sum_{i=1}^{P}(\lambda_{mid}\delta_i(\boldsymbol{k}_d) + \lambda_{mir}\delta_i(r) + \lambda_{mim}\delta_i(m)) \\ &+ \lambda_{mio}||o||_2^2,\end{aligned} \tag{12}$$

where δ_i computes the standard deviation of the i-th image patch, and P is the number of patches. The full material regularization is then computed as:

$$\mathcal{L}_{mat} = \mathcal{L}_{mat}^{F} + \mathcal{L}_{mat}^{I}. \tag{13}$$

The real-world illumination is generally neutral with balanced colors. Therefore, the cubemap is compared against an "averaged gray" version of itself to prevent severe color shift. Specifically, given G_{base} as the channel-averaged gray image of the learned environment cubemap I_{base} at the 0-th mipmap level:

$$\mathcal{L}_{light} = \lambda_l|I_{base} - G_{base}|_1, \tag{14}$$

In summary, the total loss function is a combination of the aforementioned losses:

$$\mathcal{L} = \mathcal{L}_{\hat{\boldsymbol{c}}_{mat}} + \mathcal{L}_{\hat{\boldsymbol{c}}_{tex}} + \mathcal{L}_{sdf} + \mathcal{L}_{mask} + \mathcal{L}_{mat} + \mathcal{L}_{light}. \tag{15}$$

3.4. Indirect Illumination

To handle complex indirect lighting, we distill the indirect illumination field from the accompanying radiance field

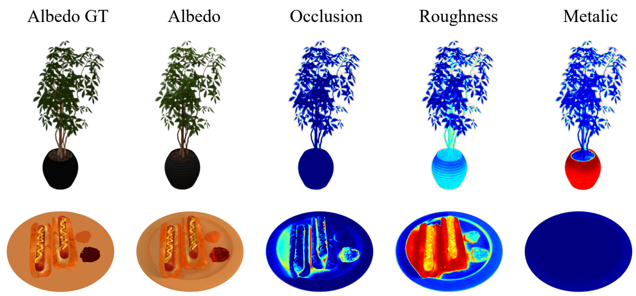

Figure 5: An example of material factorization by our proposed method.

and incorporate it during rendering when direct light is geometrically blocked. This indirect illumination distillation process is applied as a post-processing step after the initial training. Following InvRender [50], we parameterize the indirect illumination using $M = 12$ Spherical Gaussians (SGs) for spatial positions:

$$L_i^{ind}(\boldsymbol{\omega}, \boldsymbol{x}) = \sum_{k=1}^{M} G_{\boldsymbol{x}}(\boldsymbol{\omega}; \boldsymbol{\xi}_k, \lambda_k, \boldsymbol{\mu}_k), \tag{16}$$

where $\boldsymbol{\omega} \in \mathbb{S}^2$ is the queried incident direction at position $\boldsymbol{x}$. The parameters $\boldsymbol{\xi}_k \in \mathbb{S}^2$, $\lambda_k \in \mathbb{R}_+$, and $\boldsymbol{\mu}_k \in \mathbb{R}^3$ represent the axis, sharpness, and amplitude of each Gaussian lobe, respectively. The SG parameters are modeled using an MLP that takes only the position $\boldsymbol{x}$ as input. Similarly, a specular occlusion term $o_s(\boldsymbol{\omega}, \boldsymbol{x})$ is modeled to determine whether a direction $\boldsymbol{\omega}$ is geometrically blocked. The specular occlusion is hemispherically integrated to approximate the diffuse occlusion term $o_d(\boldsymbol{x})$. Following PhySG [47], the rendering equation can be efficiently integrated by approximating the GGX normal distribution using SGs as well. The outgoing radiance is computed as a combination of direct and indirect components, each weighted by the specular and diffuse occlusion terms:

$$L = L_{dir} + L_{ind}, \tag{17}$$
$$L_{dir} = (1 - o_s(r))L_{dir}^s + (1 - o_d)L_{dir}^d, \tag{18}$$
$$L_{ind} = o_s(r)L_{ind}^s + o_d L_{ind}^d, \tag{19}$$

Here, L_{dir} and L_{ind} represent the direct and indirect outgoing radiance, respectively, while the superscripts s and d denote the specular and diffuse components of each radiance. The outgoing radiances are weighted by the occlusion terms in a complementary manner: when the object's geometry obstructs the incoming light, the object is illuminated by indirect light; conversely, when there is no obstruction, the object receives direct illumination from the environment. For the specular component, we simplify the computation of occluded incident light by only considering occlusion in the reflection direction $\boldsymbol{r}$. We assume that the outgoing specular radiance is highly influenced by mirror reflection, and thus multiply the occlusion term by the outgoing radiance. Similarly, the diffuse occlusion effect is simplified by multiplying the integrated diffuse occlusion by the outgoing radiance, assuming that the incident light is uniformly occluded. Despite these simplifications, our experiments demonstrate satisfactory performance.

To learn the indirect illumination field, we begin by conducting hemisphere sampling at a surface point $\boldsymbol{x}$ to obtain secondary ray directions $\boldsymbol{\omega}_s$. A sphere tracing algorithm is then used to find intersections with other surfaces, denoted as $\boldsymbol{x}'$. The radiance extracted from the Radiance MLP, in the opposite sampling direction, is used to supervise the corresponding indirect illumination:

$$\mathcal{L}_{ind} = \lambda_{ind} \sum_{s \in S} ||L_{ind}(\boldsymbol{\omega}_s, \boldsymbol{x}) - L_{tex}(-\boldsymbol{\omega}_s, \boldsymbol{x}')||_2^2. \tag{20}$$

Additionally, depending on whether the sphere tracing algorithm finds an intersection along the sampled direction $\boldsymbol{\omega}_s$, denoted as $o_s'(\boldsymbol{\omega}_s, \boldsymbol{x})$, the specular occlusion term o_s is supervised using an MSE loss:

$$\mathcal{L}_{occ} = \lambda_{occ} \sum_{s \in S} ||o_s(\boldsymbol{\omega}_s, \boldsymbol{x}) - o_s'(\boldsymbol{\omega}_s, \boldsymbol{x})||_2^2. \tag{21}$$

4. Experiments

4.1. Baselines

Our work is primarily related to two methods, NVDiffrec [30] and Neural-PIL [6]. Both methods use pre-integrated illumination for lighting modeling but differ in their geometry modeling approaches. Additionally, recent works based on these methods are available: NVDiffrecmc [16] adopts Monte Carlo Rendering for shading based on NVDIffrec's geometry and material representation, SAMURAI [5] additionally estimates camera pose for inverse rendering. We also compare with NeRFactor [48] and InvRender [50], which are both implicit representation-based methods that adopt different representations for geometry and illuminations. To measure the image quality of relighting and albedo images, we use three quantitative metrics: Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). For geometry evaluation on synthesis dataset with ground truth geometry, Chamfer Distance (CD) is reported.

4.2. Implementation Details

We jointly optimize scene geometry, material, and environmental illumination using both image loss and parameter regularization. For the image loss, we prioritize Mean

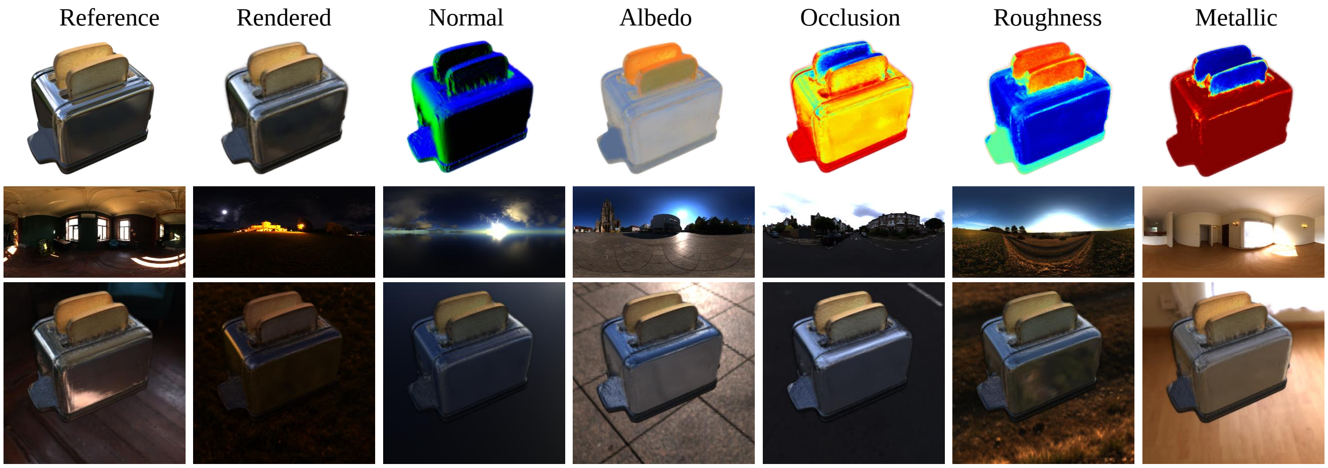

Figure 6: Material factorization and relighting on Ref-NeRF's Shiny scene. Top row: material factorization results. Middle row: high-frequency environment maps used for relighting. Bottom row: relighted results under the given corresponding lights.

Method	Relighting			Albedo		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeuS	21.83	0.913	0.070	-	-	-
NeRFactor	23.78	0.907	0.112	23.11	0.917	0.094
NVDiffrec	24.53	0.914	0.085	24.75	0.924	0.092
NVDiffrecmc	26.20	**0.928**	**0.054**	**25.34**	**0.931**	0.072
Ours	**26.23**	0.925	0.058	24.86	0.921	0.066
w/o $\mathcal{L}_{mat}$	26.07	0.923	0.062	24.60	0.924	**0.064**
w/o $\mathcal{L}_{sdf}$	24.05	0.904	0.085	24.61	**0.929**	**0.065**

Table 1: Quantitative evaluation on NeRFactor's synthesis dataset. Both NVDiffrec and NeRFactor metrics are as reported in [30] (In bold: best; Underline: second best).

Squared Error (MSE) loss by setting $\lambda_{c1} = 1$, $\lambda_{c2} = 10$, and $\lambda_{mask} = 0.1$. For parameter regularizations, we set $\lambda_{se} = 0.1$, $\lambda_l = 0.1$, $\lambda_{mio} = 0.001$, and all other parameters as default at 0.01. Our model is optimized using the Adam optimizer with a learning rate of 0.01. The learning rate is scheduled by a 500-step warming-up stage, starting from 1% and rising to 100%, followed by an exponential decay until the end of training. Material and radiance modules are scheduled asymmetrically to facilitate geometry initialization. Our experiment was conducted on 2 NVIDIA Tesla V100 GPUs, with a training time of approximately 1.5 hours for a total of 40,000 steps.

4.3. Experiment on Synthetic Dataset

NeRFactor's Relight Dataset. Following NeRFactor [48], four synthetic scenes originally released by NeRF [28] are relighted with eight different low-frequency environment illuminations, and evaluated over eight uniformly

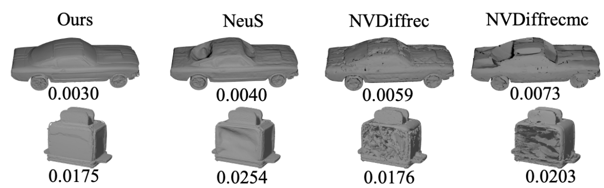

Figure 7: Evaluation of the geometry on Shiny Dataset. Chamfer Distances are shown beneath each example image.

sampled novel views. We compare our method with baseline NeRFactor [48], NVDiffrec [30], and NVDiffrecmc [16] on NeRFactor's Blender dataset on relighting and albedo reconstruction qualities. As material and fixed illumination can only be resolved up to a relative scale, we adopt the convention that scales the predicted albedo image by a color-tuning factor that matches the average of ground-truth albedo. As shown in Tab. 1, our method outperforms both NeRFactor and NVdiffrec and achieves on-par relighting performance with NVDiffrecmc. We attribute the performance gain to adopting efficient implicit neural surface representation together with pre-integrated rendering, whereas NVDiffrecmc's performance gain is contributed by novel Monte Carlo shading. We list NeuS here as a baseline of non-relightable methods for comparison. The qualitative results are visualized in Fig. 4, where we render the image using a constructed environment map from a novel view and relight it with a given illumination.

Shiny Scenes. As NeRFactor's Relight dataset contains mostly Lambertian surfaces illuminated by low-frequency environment light, we further evaluate our method on Ref-NeRF's Shiny scenes [37], which contains shiny objects. To evaluate the relighting ability in a more challenging scenario, We relight the Blender models of car and toaster using Blender Cycles renderer with 7 different high-frequency environment lights. Fig. 6 shows the material factorization and relighting results on the toaster scene. The bread and toaster material were correctly predicted, and the relighting results blend in the novel illumination and exhibit consistent non-Lambertian reflection on the surface (notice the reflection of the environment map visible on the toaster).

We further show evaluations of the reconstructed geometry on the Shiny dataset in Fig. 7. The Chamfer distance is calculated by uniformly sampling 5,000 points from the mesh and comparing them to the samplings from ground truth models. Our method addresses the geometric ambiguity often found in NeuS by explicitly factoring the illumination, which makes it easier for the model to interpret shiny objects. Therefore effectively maintains the convex geometry on the highly reflective region of both the car and toaster model. Additionally, our method reproduces smoother surfaces compared to NVDiffrec/mc, showing the superiority of using continuous implicit neural surface representation over a discretized representation.

4.4. Experiment on Real-World Dataset

To evaluate our method on real-world scenes, we follow Wang et al. [39] in adopting Common Objects in 3D (CO3D) dataset [32] and evaluating a subset of cars. The CO3D dataset is a collection of multi-view images captured in outdoor settings, containing detailed annotations such as ground-truth camera pose, intrinsic, depth map, object mask, and 3D point cloud. This dataset was gathered through real-world video capture and presents a significant challenge to reconstruction algorithms due to the presence of highly reflective and low-textured surfaces like dark windows, and metallic paint, which are non-Lambertian. As the ground-truth object mask is directly produced by off-the-shelf software, up to 8% of the masks are wrong. In our experiment, we filtered out the incorrectly masked images by first computing the distribution of masked percentages of all images in a scene and then dropping the images whose masked percentage is below the second mode threshold if the distribution is multimodal.

We compare our method with NVDiffrec [30], NVdiffrecmc [16], SAMURAI [5], NeuralPIL [6], and InvRender [50] on a subset of 10 car scenes with relatively complete 360° viewing directions in CO3D dataset. As shown in Tab. 2, our results are significantly better than other methods of novel view synthesis. It is worth noting that NVDiffrecmc performs worse than NVDiffrec because NVDiffrecmc enforces additional regularization[16] and also tends to degrade the geometry as shown in Fig. 7. A more detailed qualitative comparison with most related NVDiffrec methods is visualized in Fig. 8. Although its rendered novel views look realistic in the interpolated viewing direction, NVDiffrec fails to reconstruct a smooth mesh, and the geometry artifacts lead to noisy material factorization and unrealistic relighting results. We attribute this to the fact that NVDiffrec uses differentiable marching tetrahedrons with a fixed number of vertices to represent geometry, limiting its ability to represent geometry and behaving unstably under limited views.

Table 2: Quantitative evaluation on CO3D dataset. [32]

Method	Novel View		
	PSNR↑	SSIM ↑	LPIPS ↓
NVDiffrec [30]	26.29	0.925	0.086
NVDiffrecmc [16]	24.45	0.911	0.107
SAMURAI [5]	24.88	0.901	0.118
NeuralPIL [6]	25.42	0.915	0.092
InvRender [50]	24.94	0.919	0.092
Ours	**29.03**	**0.935**	**0.046**

4.5. Indirect Illumination

We further show the additional indirect illumination distillation in Fig. 9 using the Toaster scene. The model is initialized using the pre-trained geometry, radiance and material module and fine-tuned for another 20K epochs by adding supervision for indirect illumination $\mathcal{L}_{ind}$ and occlusion $\mathcal{L}_{occ}$ as described in Sec. 3.4. As shown in Fig. 9, the learned occlusion terms for specular and diffuse occlusion help in distinguishing the contribution of direct and indirect illumination between the parallel breads as well as recognizing the indirect color bleeding on the shiny surface of the toaster machine. This decomposition leads to more consistent albedo prediction on the bread, eliminating the effect of complex indirect illumination. This shows that modular indirect illumination distillation can work effectively based on our learned representation.

4.6. Ablation Study

Material and SDF Regularization. We perform an ablation study by comparing the result of NeRFactor's synthesis dataset with and without a certain factor. The results are summarized in Tab. 1. We observe only limited performance gain in relighting and albedo reconstruction regarding PSNR. We attribute this to the fact that the dataset already contains high-quality input views to regularize materials. The SDF loss played a crucial role in solving the ambiguity between geometry and material property, as it penalizes noises in the reconstructed surface normals. As a result, the quality of relighting results sees a large improvement with the SDF loss.

Neural-PIL vs. Pre-computed Environment map. We compare our choice of using a trainable environmental cubemap and explicitly computing the pre-integration

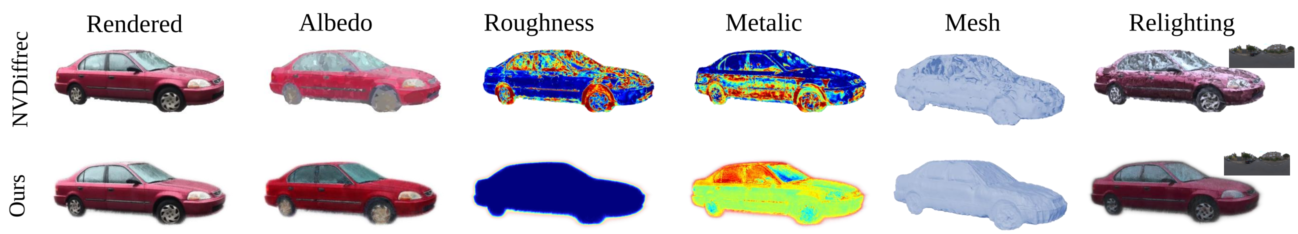

Figure 8: A comparison between our proposed method and NVDiffrec [30] on real-world CO3D dataset. Although NVDiffrec shows promising results on rendered novel views, it fails to reconstruct a plausible mesh. The geometry artifacts lead to noisy material factorization and unrealistic relighting. our method with high-quality neural implicit field representation results in better overall results.

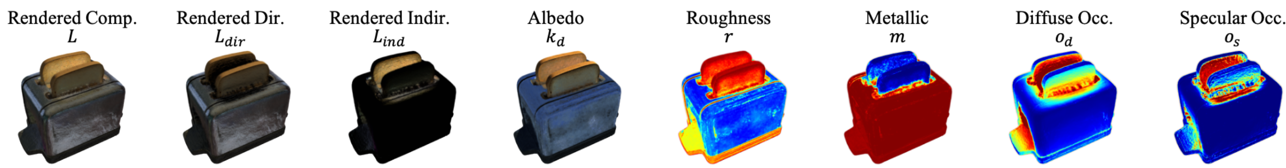

Figure 9: Indirect illumination distillation. The first three are rendered images using composed illumination L, direct illumination L_{dir}, and indirect illumination L_{ind} respectively in Eq. 17. The middle three images are fine-tuned material properties, and the last two images are distilled diffuse occlusion o_d and specular occlusion o_s for Eq. 18, and Eq. 19 respectively.

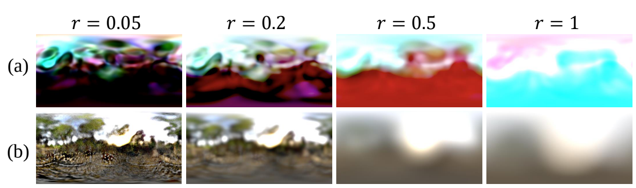

Figure 10: Different illumination modeling methods. (a) Using Neural-PIL [6]'s environment illumination modeling. (b) Our proposed method.

as the lighting representation with Neural-PIL, which uses FILM-SIREN layers [6] to learn the pre-integrated illumination by taking different roughness as network input. We find that Neural-PIL does not impose strict integration relations among the queried environment maps across different roughness levels. We implement Neural-PIL's method as an alternative to our illumination model and evaluate its ability to reconstruct the environment illumination using NeRF's material scenes. As shown in Fig. 10, when queried at roughness 1, the environment map drastically differs from the previous levels of pre-filtered environment maps and has a complementary color. This erroneous illumination prediction violated the inherent relations across different roughness levels and therefore leads to worse material reproducing. On the contrary, our implementation learns high-quality illumination, which is consistent across different roughness levels by design and validated in Fig. 10. Additionally, to regularize illumination, Neural-PIL resorts to training another auto-encoder network to learn illumination latent codes using data-driven methods. In comparison, our method purely deduces illumination from the scene without the need for extra data and thus reduces the induced bias (e.g. blueish style because more sky environment maps are collected, and lower resolution as the collected environment is down-sampled before training the auto-encoder).

5. Conclusion

In this paper, we presented NeuS-PIR, a novel approach for learning relightable neural surfaces via pre-integrated rendering. Our method jointly optimizes a neural implicit surface, a spatially varying material field, and a differentiable environment cubemap, leading to high-quality reconstruction and relighting results. Furthermore, we demonstrated how indirect illumination fields can be effectively distilled from the learned representations. Experimental evaluations show that NeuS-PIR outperforms existing methods in both reconstruction accuracy and relighting performance, showcasing its potential for a wide range of applications in computer vision and graphics.

Limitations. Our approach exhibits suboptimal results when the number of input views is limited. This issue could potentially be mitigated by incorporating data augmentation techniques or diffusion models [44, 18, 43, 49]. While our method is capable of handling in-the-wild data, stability remains a challenge in certain cases. Inaccurate object mask and camera pose estimation affects the decomposition of our method. Enhancing robustness when processing such data is an important direction for future work. Additionally, we export explicit representations (i.e., meshes) using the marching cubes algorithm [9], which may introduce some loss of accuracy. Recent advances like neural marching cubes [8] and neural dual contouring [7] could provide more accurate explicit outputs for downstream tasks. Finally, our current approach to indirect illumination distillation relies on geometry occlusion multiplication to approximate more complex masked light integration, which could be further refined.

References

[1] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2, 3

[2] S. Bi, Z. Xu, P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 1, 3

[3] S. Bi, Z. Xu, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *European Conference on Computer Vision*, pages 294–311, 2020. 1, 3

[4] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2, 3

[5] M. Boss, A. Engelhardt, A. Kar, Y. Li, D. Sun, J. T. Barron, H. P. Lensch, and V. Jampani. Samurai: Shape and material from unconstrained real-world arbitrary image collections. In *Advances in Neural Information Processing Systems*, 2022. 7, 9

[6] M. Boss, V. Jampani, R. Braun, C. Liu, J. Barron, and H. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *Advances in Neural Information Processing Systems*, volume 34, pages 10691–10704, 2021. 2, 3, 7, 9, 10

[7] Z. Chen, A. Tagliasacchi, T. Funkhouser, and H. Zhang. Neural dual contouring. *ACM Transactions on Graphics*, 41(4):1–13, 2022. 11

[8] Z. Chen and H. Zhang. Neural marching cubes. *ACM Transactions on Graphics*, 40(6):1–15, 2021. 11

[9] E. Chernyaev. Marching cubes 33: Construction of topologically correct isosurfaces. Technical report, CERN, 1995. 11

[10] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982. 4

[11] J. Fan, F. Luan, J. Yang, M. Hašan, and B. Wang. Rng: Relightable neural gaussians. *arXiv preprint arXiv:2409.19702*, 2024. 3

[12] D. Gao, G. Chen, Y. Dong, P. Peers, K. Xu, and X. Tong. Deferred neural lighting: free-viewpoint relighting from unstructured photographs. *ACM Transactions on Graphics*, 39(6):1–15, 2020. 3

[13] J. Gao, C. Gu, Y. Lin, H. Zhu, X. Cao, L. Zhang, and Y. Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv preprint arXiv:2311.16043*, 2023. 3

[14] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–224, 2018. 2

[15] C. Häne, S. Tulsiani, and J. Malik. Hierarchical surface prediction for 3d object reconstruction. In *Proceedings of the International Conference on 3D Vision*, pages 412–420. IEEE, 2017. 2

[16] J. Hasselgren, N. Hofmann, and J. Munkberg. Shape, light, and material decomposition from images using monte carlo rendering and denoising. In *Advances in Neural Information Processing Systems*, 2022. 3, 7, 8, 9

[17] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, and W. Wang. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. 3

[18] H. Jin, Y. Li, F. Luan, Y. Xiangli, S. Bi, K. Zhang, Z. Xu, J. Sun, and N. Snavely. Neural gaffer: Relighting any object via diffusion. *arXiv preprint arXiv:2406.07520*, 2024. 10

[19] H. Jin, I. Liu, P. Xu, X. Zhang, S. Han, S. Bi, X. Zhou, Z. Xu, and H. Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023. 3

[20] B. Karis and E. Games. Real shading in unreal engine 4. *Proceedings of Physically Based Shading Theory Practice*, 4(3):1, 2013. 4, 5

[21] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 3

[22] R. Liang, H. Chen, C. Li, F. Chen, S. Panneer, and N. Vijaykumar. Envidr: Implicit differentiable renderer with neural environment lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[23] Y. Liao, S. Donne, and A. Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018. 2

[24] S. Liu, T. Li, W. Chen, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 2

[25] Y. Liu, P. Wang, C. Lin, X. Long, J. Wang, L. Liu, T. Komura, and W. Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. In *SIGGRAPH*, 2023. 3

[26] X. Long, C. Lin, P. Wang, T. Komura, and W. Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*. Springer, 2022. 3

[27] F. Luan, S. Zhao, K. Bala, and Z. Dong. Unified shape and svbrdf recovery using differentiable monte carlo rendering. *Computer Graphics Forum*, 40:101–113, 2021. 3

[28] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. 1, 2, 8

[29] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 5

[30] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 1, 2, 3, 5, 7, 8, 9, 10

[31] M. Oechsle, S. Peng, and A. Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5589–5599, 2021. 2

[32] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 9

[33] T. Shen, J. Gao, K. Yin, M.-Y. Liu, and S. Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 6087–6101, 2021. 2

[34] Y. Shi, Y. Wu, C. Wu, X. Liu, C. Zhao, H. Feng, J. Liu, L. Zhang, J. Zhang, B. Zhou, et al. Gir: 3d gaussian inverse rendering for relightable scene factorization. *arXiv preprint arXiv:2312.05133*, 2023. 3

[35] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2

[36] C. Sun, G. Cai, Z. Li, K. Yan, C. Zhang, C. Marshall, J.-B. Huang, S. Zhao, and Z. Dong. Neural-pbir reconstruction of shape, material, and illumination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18046–18056, 2023. 3

[37] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5481–5490. IEEE, 2022. 3, 8

[38] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007. 5

[39] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, 2021. 1, 2, 3, 4, 9

[40] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2022. 3

[41] Y. Yao, J. Zhang, J. Liu, Y. Qu, T. Fang, D. McKinnon, Y. Tsin, and L. Quan. Neilf: Neural incident light field for physically-based material estimation. In *European Conference on Computer Vision*, pages 700–716, 2022. 3

[42] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 4805–4815, 2021. 2

[43] C. Zeng, Y. Dong, P. Peers, Y. Kong, H. Wu, and X. Tong. Dilightnet: Fine-grained lighting control for diffusion-based image generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 10

[44] Z. Zeng, V. Deschaintre, I. Georgiev, Y. Hold-Geoffroy, Y. Hu, F. Luan, L.-Q. Yan, and M. Hašan. Rgb-x: Image decomposition and synthesis using material-and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 10

[45] J. Zhang, Y. Yao, S. Li, J. Liu, T. Fang, D. McKinnon, Y. Tsin, and L. Quan. Neilf++: Inter-reflectable light fields for geometry and material estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[46] K. Zhang, F. Luan, Z. Li, and N. Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5574, 2022. 1, 3

[47] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 3, 7

[48] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics*, 40(6):1–18, 2021. 2, 3, 7, 8

[49] Y. Zhang, Y. Liu, Z. Xie, L. Yang, Z. Liu, M. Yang, R. Zhang, Q. Kou, C. Lin, W. Wang, et al. Dreammat: High-quality pbr material generation with geometry-and light-aware diffusion models. *ACM Transactions on Graphics*, 43(4):1–18, 2024. 10

[50] Y. Zhang, J. Sun, X. He, H. Fu, R. Jia, and X. Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 6, 7, 9