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

Shi Mao Chenming Wu* Zhelun Shen Liangjun Zhang
Robotics and Auto-Driving Lab (RAL), Baidu Research
mssheldonmao@gmail.com, {wuchenming, shenzhelun, liangjunzhang}@baidu.com

Abstract

Recent advances in neural implicit fields enables rapidly reconstructing 3D geometry from multi-view images. Beyond that, recovering physical properties such as material and illumination is essential for enabling more applications. This paper presents a new method that effectively learns relightable neural surface using pre-intergrated rendering, which simultaneously learns geometry, material and illumination within the neural implicit field. The key insight of our work is that these properties are closely related to each other, and optimizing them in a collaborative manner would lead to consistent improvements. Specifically, we propose NeuS-PIR, a method that factorizes the radiance field into a spatially varying material field and a differentiable environment cubemap, and jointly learns it with geometry represented by neural surface. Our experiments demonstrate that the proposed method outperforms the state-of-the-art method in both synthetic and real datasets.

1. Introduction

3D reconstruction from multi-view images is a fundamental task in computer vision. Conventional approaches involve a long workflow, such as structure-from-motion, correspondence matching, mesh reconstruction, and texturing [21, 36]. While this paradigm can be effective, it is also prone to errors, thus manual correction is often required. Recently, learning-based approaches have shown promise in the task of object reconstruction, and achieved impressive results in terms of geometry quality. Beyond geometry, recovering physical properties such as material and illumination is also essential for real-world applications, such as view synthesis [45, 5, 38], relighting [19, 1, 7], object insertion [16, 26, 44], material editing [46, 33]. In other words, along with the physical properties, 3D reconstruction can have wider range of applications in computer vision and graphics, such as AR/VR, autonomous driving simulation, and films, etc.

The task of recovering an object's geometry and material as well as illumination from collected images is *inverse rendering*, which enables rendering objects from novel viewpoints under different illumination conditions. We follow the recent success of neural implicit field to address this problem. NeRF [31, 51, 2, 40, 32, 39, 14] propose to use neural implicit field and differential volume rendering for novel view synthesis, which have promise in producing high-quality geometry reconstruction. NeRF-based methods target on synthesizing photorealstic novel views, and they often struggle to recover fine-grained geometry details. To tackle this problem, *Signed Distance Function* (SDF) is introduced into volume rendering to explicitly supervise the geometry depiction, and has achieved improved results [42, 34, 48]. Overall, existing inverse rendering approaches [5, 52, 7] mostly focus on extending NeRF to decomposing material and illumination while geometry qual-

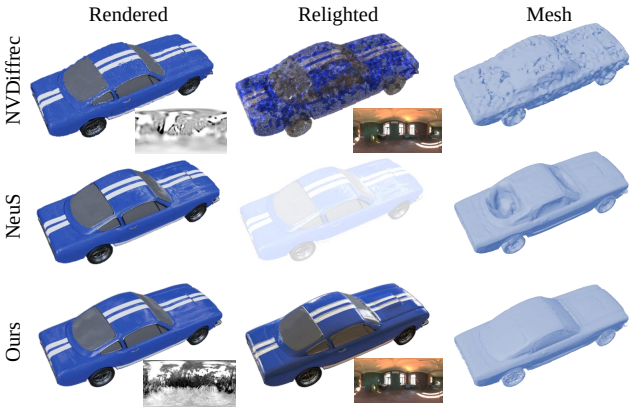

Figure 1: Our proposed method simultaneously learns geometry, material, and illumination within the neural implicit field. Our results are shown in the bottom row of the figure. Compared to NVDiffrec [33] shown in the top row, the relighted image and reconstructed geometry of our method are significantly better. Compared to NeuS [42], which is used for geometry recovery, our mesh benefits from the material and illumination learning, and can well preserve the geometry on the highly-reflective regions, while NeuS fails in reconstructing these regions. Note that NeuS does not support relighting.

*Correspondence

ity is not prioritized.

In our work, we propose a novel method for learning relightable neural impilicit surface using pre-integrated rendering. Unlike previous approaches, our method is built upon a high-quality neural implicit approach – NeuS, which interprets the scene as a SDF function over the learning space. We further enhance the method by decomposing material and illumination properties from the radiance field. As a result, our approach prioritizes not only geometry, but material and illumination simultaneously. Figure 1 shows an example demonstrating our proposed method continuously improves the reconstruction quality in terms of all the three aspects, and achieves competitive performance compared to the most related work – NVDiffrec [33], which uses differentiable marching tetrahedrons to optimize surface mesh through gradient-based optimization. Specifically, our proposed method is build on neural surface representation, which we extend it to support modeling an spatially varying material field and a differentiable environment cubemap. All of our geometry, material and lighting representations are jointly optimized by the re-rendered images using pre-integrated rendering. In summary, our work makes the following contributions:

- We propose a new method, namely NeuS-PIR, which is based on neural implicit surface and pre-integrated rendering, to factorize object geometry, material and illumination. Consequently, our method allows high-quality relighting in novel conditions.
- Our proposed jointly optimizing scheme enables the geometry, material, and illumination to benefit each other during training in a collaborative manner. This leads to an improved factorization result and alleviates degraded geometry.
- We adopt a regularization method for material encoding that encourages sparse and consistent material factorization. This method has the potential to generalize to more scenes compared to methods that use data-driven priors learned from BRDF datasets [7].

2. Related Work

2.1. Multi-view Reconstruction

Reconstructing 3D models from multi-view images is a long-standing research problem. Structure-from-motion (SfM) [36] and multi-view stereo (MVS) [17] are commonly used methods for this problem. Existing methods can be categorized based on how they represent object surfaces, either explicitly, such as point cloud or triangle mesh, or implicitly, such as signed distance field. These methods typically match corresponding pixels between images to estimate their depth values for framewise fusion using explicit representation [21, 11] or use a set of occupied voxels to store implicit distance fields describing the surface [13]. More recently, neural representation has demonstrated its ability to reconstruct 3D objects with less error and more efficiency compared to classic multi-view reconstruction methods.

Explicit Reconstruction from images can be achieved in several ways, including triangle mesh [28], tetrahedral mesh [33], voxel [27], hierarchical octree [20], atlas surface [18], or explicit/implicit hybrid [37]. One significant advantage of explicit representation is the reconstructed models are compatible with downstream applications, such as rendering on industrial engine. In contrast, implicit fields need to be converted to explicit representation and may introduce rounding errors. Optimizing explicit representation is sensitive to hyperparameters, which can sometimes result in reconstruction failures due to topological inconsistency. As opposite to that, neural implicit fields are more stable.

Neural Implicit Field Modeling uses neural networks to represent spatial numerical fields for modeling 3D objects or scenes. Neural Radiance Field (NeRF) [31] and its variants (e.g., MipNeRF [2]) use coordinate-based MLP to spatially encode the volumetric radiance space, and geometry and color of an arbitrary point inside this space can be queried by MLP, allowing for rendering by casting rays and integrating all the queried values. While NeRF-like methods can synthesize high-quality novel views, the intrinsic geometry is not explicitly optimized. Recent advances in neural surface reconstruction carefully design the optimization framework that can optimize the geometry using photometric loss. For example, UniSurf [34] proposes to sharpen the sampling distribution to align the volumetric field to the surface. VolSDF [48] converts the density function in NeRF to a learnable SDF transformation, and samples points along the casting rays according to the error bound of opacity. NeuS [42] and its follow-up [43, 29] provide an unbiased and occlusion-aware solution to convert the density function in NeRF to SDF. Although these methods can reconstruct plausible geometry and render high-quality novel views, the material and illumination are factorized into the model itself, resulting unsatisfactory rendered results when background scenes are different.

2.2. Material and Illumination Estimation

Estimating material and illumination for reconstructed objects is a challenging task. Previous methods require known lighting conditions for inverse rendering. Deep Reflectane Volumes [4] and Neural Reflectance Field [3] use differentiable volume ray marching framework to supervise the reconstruction of a neural reflectance volume and reflectance field respectively. Deferred Neural Lighting [15] applies deferred rendering using proxy geometry and neural texture followed by neural rendering to

Method	[31]	[42]	[5]	[52]	[50]	[7]	[33]	[22]	Ours
Geometry	NV	NS	NV	NV	NS	NV	Mesh	Mesh	NS
Factorize	-	-	✓	✓	✓	✓	✓	✓	✓
Detail Freq.	-	-	Low	Low	Low	Low	High	High	High

Table 1: Our method adopts neural surface to factorize material and illumination properties, and supports high-frequency details. A detailed comparison among existing approaches, including NeRF [31], NeuS [42], NeRD [5], NeRFactor [52], PhySG [50], Neural-PIL [7], NVDiffrec [33], and NVDiffrecmc [22]. NV: Neural Volume, NS: Neural Surface, Detail Freq.: detail frequency for illumination representation.

enable free-viewpoint relighting. Another mesh based method [30] jointly optimizes mesh and SVBRDF by a differentiable renderer specialized for collocated configurations. IRON [49] proposes a two-stage method that firstly uses signed distance field for recovering geometry and then optimizing material. These methods requires photometric images, leading to a more involved data capturing process, while our method jointly optimizes geometry, material and illumination with only images as input.

Recent efforts on inverse rendering enable us to estimate material and illumination from multi-view images. NeRV [38] assumes known illumination conditions, and represents the scene as a continuous volumetric function. However, its computational complexity is considerably high. NeRFactor [52] uses a set of MLPs to describe light source visibility, normal maps, surface albedo, and the material property on the surface point, on top of a pretrained NeRF model. PhySG [50] proposes to model the illumination using spherical Gaussian, which has a precondition that the scene is under a fixed illumination. NeRD [5] extends the fixed illumination condition to both fixed or varying illumination. Neural-PIL [7] proposes a neural pre-intergrated lighting method to replace the spherical Gaussians, which enables estimating high-frequency lighting details. Ref-NeRF [40] proposes a unified network architecture by replacing MipNeRF [2]'s parametrization of view-dependent outgoing radiance with a reflected radiance to model environment light and surface roughness. NeILF [47] proposes to use a fully 5D light field to model illuminations of any static scene, where occlusions and indirect lights are handled naturally. This method requires reconstructed geometry as input while ours jointly optimizes geometry and physcail properteis. Instead of only using implicit fields to describe the scenes, NVDiffrec [33] adopts a hybrid approach that uses implicit SDF field and explicit mesh together, and proposes an efficient differential rendering pipeline for reconstruction. The tightly coupled and long pipeline that combines two different structures exhibit unstable training performance, while our method performs better than it by jointly learning high-quality geometry, material and illumination using neural implicit surface representation with pre-integrated rendering techniques. The follow-up work NVDiffrecmc [22] adopts the same geometry and material representations, but incorporates ray tracing and Monte Carlo integration for more realistic shading, and a denoising model to address the noise caused by Monte Carlo integration. We show a detailed comparison among different approaches in Table 1.

3. Methodology

Given a collection of multi-view images and their corresponding camera poses, our goal is to reconstruct the object in camera views and its surrounding environment illumination. To be concrete, our factorization method learns an implicit representation of the object-related geometry and material, as well as the object-irrelevant illumination. As shown in Figure 2, for each point $\boldsymbol{x} \in \mathbb{R}^3$ as an input, our model outputs its *signed distance field* (SDF) value $s \in \mathbb{R}$, diffuse albedo $\boldsymbol{k_d} \in [0,1]^3$, roughness $r \in [0,1]$ and metallic $m \in [0,1]$, as well as the illumination represented by an environmental cubemap $I \in \mathbb{R}^{H,W,6}$. After the training finalized, our method enables us to render reconstructed objects under different environment maps representing the illumination. This is referred to as *relighting*, and can be achieved by either jointly estimated or externally defined environment maps. The rendering process follows volume rendering principle. In addition, high-quality surface mesh with material properties can be easily obtained from the learning representation using off-the-shelf mesh approximation algorithm (i.e., marching cube or its variants), and enables downstream applications.

3.1. Learning Geometry

Our method adopts Neural Implicit Surfaces (NeuS) [42] as the representation of an object's geometry, considering its ability to reconstruct high-quality surfaces as the zero-level set of implicit SDF representation. NeuS uses multi-layer perceptron (MLP) to learn both the SDF function $f_{sdf} : \boldsymbol{x} \mapsto s$ that maps a 3D position $\boldsymbol{x} \in \mathbb{R}^3$ to a real value $s \in \mathbb{R}$, and appearance mapping $f_{color} : (\boldsymbol{x}, \boldsymbol{v}) \mapsto \boldsymbol{c}$ that maps a 3D position $\boldsymbol{x} \in \mathbb{R}^3$ and viewing direction $\boldsymbol{v} \in \mathbb{S}^2$ to its corresponding color $\boldsymbol{c} \in [0,1]^3$.

NeuS renders an image by accumulating the radiance along the rays cast by pixels, following the standard volume rendering scheme. Specifically, given a pixel ray parameterized as $\{\boldsymbol{x}(t) = \boldsymbol{o} - t\boldsymbol{v} | t \geq 0\}$, where $\boldsymbol{o} \in \mathbb{R}^3$ represents its corresponding camera origin and $\boldsymbol{v} \in \mathbb{S}^2$ is its normalized direction pointing towards camera center, the accumulated color for this pixel can be computed as a weighted sum of colors along the ray:

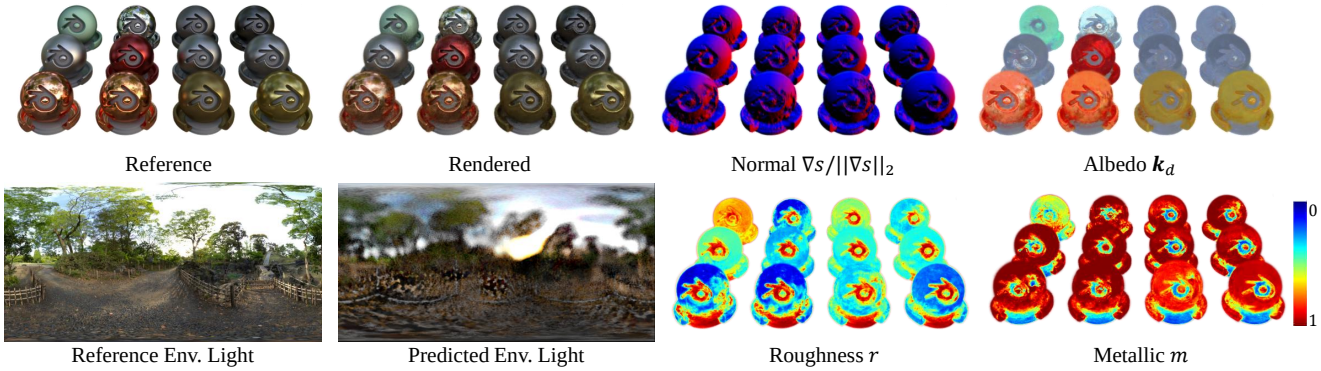

Figure 2: Our method factorize a scene into geometry, material and illumination. We display the reference and predicted environment illumination as a latitude-longitude converted environment cubemap. The roughness and metallic are visualized using jet color-map ranging from 0 to 1.

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

where $w(t)$ is a non-negative weight function, and we integrate it from near plane t_n to far plane t_f of the camera model. By enforcing the unbiased and occlusion-aware requirements, NeuS derives the weight function from SDF as:

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

where $\rho(t) = \max\left(\frac{-\mathrm{d}\Phi_\tau}{\mathrm{d}t}(s(t))}{\Phi_s(s(t))}, 0\right)$ is refereed to as 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 density distribution near the zero-level across the SDF. During training, $1/\tau$ is expected to converge to zero as the zero-valued isosurfaces of the SDF gradually approach to solid surfaces.

To enable relighting and material factorization, we model the outgoing radiance along the ray cast from a pixel in two branches. The first branch directly produces the outgoing radiance given the position, viewing direction, and surface normal using radiance MLP, while the second branch perceives material and illumination properties and render the outgoing radiance following split-sum approximation detailed in Section 3.2. Both of the branches shares the same SDF module to ensure the geometry consistency during learning.

3.2. Learning Material and Illumination

To decompose the radiance field into geometry, material and lighting components, we adopt image-based lighting as our lighting model and approximate the rendering equation with pre-integrated rendering. Following [24], the specular term in rendering equation can be approximated by *split sum* approximation:

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

where $L(\boldsymbol{l})$ represents the radiance from incident light direction $\boldsymbol{l}$, $f_s(\boldsymbol{l},\boldsymbol{v};r,m)$ denotes Cook-Torrance [12] microfacet specular BRDF, parameterized by roughness r and metallic m, and $\boldsymbol{n}$ is the surface normal vector. The first term in split sum approximation $I(\boldsymbol{r};r)$ involves an importance sampling of incident light radiance mediated by the surface roughness r. At a minor cost of losing lengthy reflection at grazing angles, this term is further approximated as viewing the surface point from the reflection direction, assuming that the surface normal is aligned to the reflection direction. Consequently, it can be pre-integrated from the environment map and queried from 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 normal distribution of GGX [41], which takes into account the percentage of microfacets that reflect light towards the viewer, and is defined by the roughness r. The precomputed illumination is stored in the mipmap levels of an environment cubemap.

The second term of the equation is irrelevant to illumination, and is equivalent to integrating specular BRDF f_s in a constant brightness environment. Using Schlick's equation, the specular reflectance at normal incidence F_0 can be factorized out. Therefore, the second term can be rewritten as F_0 modulated by its scale and bias, which are only related to the material's roughness and the cosine between the viewing angle and the surface's normal vector $(\boldsymbol{v}\cdot\boldsymbol{n})$.

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

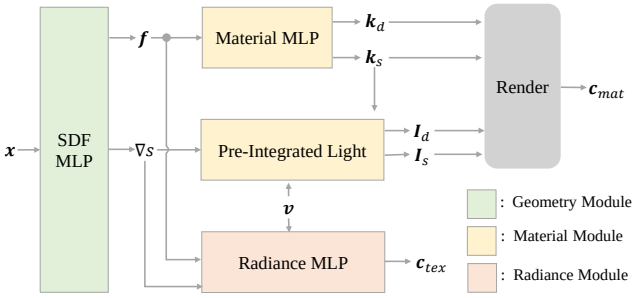

Figure 3: Network architecture of our proposed method. The SDF MLP learns the geometry, Radiance MLP learns the radiance field at a coarse level. Further decompositions factorize the material and illumination using the Material MLP and Pre-Integrated Light.

both the scale S and bias B can be pre-calculated and stored as 2D look-up table (LUT) for efficient inference. We adopt the convention from UE4 that sets F_0 as an interpolation from 0.04 (non-metallic material's specular reflectance) to diffuse color k_d (metallic material's specular reflectance) using material's metallic value m.

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

The ultimate shading model is a blend of the diffuse and specular terms and is formulated as follows.

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

where $I_d = I(\boldsymbol{n}; 1)$ refers to the diffuse irradiance and $I_s = I(\boldsymbol{r}; r)$ denotes specular irradiance. For more details of pre-integrated rendering, readers can refer to the presentation by UE4 [24].

3.3. Network Architecture

Figure 3 illustrates the architecture of our proposed method. The SDF MLP learns the geometry of the scene, while the Radiance MLP learns the radiance field of the scene at a coarse level, given the geometry features from SDF MLP and viewing directions. Further decompositions are applied to factorize the radiance into material and lighting factors, following the pre-integrated rendering principle.

Geometry Module. The SDF MLP takes 3D positions $\boldsymbol{x}$ as input and produces the feature $\boldsymbol{f}(\boldsymbol{x})$ as output. The first channel of the output feature represents its SDF value $s(\boldsymbol{x})$ and its gradient $\nabla s(\boldsymbol{x})$ is calculated analytically. Position encoding uses trainable multi-resolution grids that can be efficiently supported by hashtables [32]. The feature is learned through an MLP.

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

Material Module. The material module decomposes view-dependent outgoing radiance as incident light modulated by outgoing radiance. The Material MLP takes SDF features $\boldsymbol{f}(\boldsymbol{x})$ as input and generates surface diffuse albedo $\boldsymbol{k}_d \in [0,1]^3$, metallic $m \in [0,1]$ and roughness $r \in [0,1]$. In accordance with Munkberg et al. [33], Material MLP additionally produces an occlusion term $o \in [0,1]$ that takes into account the lack of indirect illumination and shadowing by modulating the computed outgoing radiance by $1-o$. We refer to all specular and occlusion properties of a material as $\boldsymbol{k}_s = \{o, r, m\}$. Both $\boldsymbol{k}_d$ and $\boldsymbol{k}_s$ are learned through MLP layers and activated by the sigmoid function to limit their value range to $[0,1]$.

To render the outgoing radiance using pre-integrated illumination, we follow the approach of Munkberg *et al*. [33]. They use high-resolution cubemap as trainable parameters and precompute $I(\boldsymbol{r}; r)$ for a set of discrete roughness levels as its mipmaps. To obtain a specific roughness r, we query $I(\boldsymbol{r}; r)$ using mipmap interpolation. The view-dependent radiance is then rendered using Eq. 7 and transformed to the S-RGB space via gamma correction. Finally, we integrate the rendered color $\boldsymbol{c}_{mat}$ using Eq. 1.

3.4. Loss and Regularization

We utilize the Mean Square Error (MSE), L1 loss and binary cross entropy to supervise the masked rendered images. We also include additional regularization terms for SDF, material, and light. Specifically, image color loss $\mathcal{L}_{\hat{\boldsymbol{c}}} = \lambda_{c1}||\hat{\boldsymbol{c}} - \boldsymbol{c}||_1 + \lambda_{c2}||\hat{\boldsymbol{c}} - \boldsymbol{c}||_2$, and $\hat{\boldsymbol{c}}$ represents volume rendered pixel color from either material module $\boldsymbol{c}_{mat}$ or texture module $\boldsymbol{c}_{tex}$. $\mathcal{L}_{mask} = \lambda_{mask}\mathrm{BCE}(mask, opa)$ is the binary cross entropy of image mask and the accumulated opaque density along pixel rays.

To regularize the SDF field, we use Eikonal and sparsity terms, i.e. $\mathcal{L}_{sdf} = \lambda_{se}||\nabla s - 1||_2^2 + \lambda_{ss}\exp(-\lambda_{sa}|s|_1)$. The first Eikonal term encourages the gradient of the SDF field to have a unit length, and the second term encourages the zero-valued level crossing of the SDF field to be sparse.

Material estimation can be erroneous due to limited observation of each individual surface point. However, this problem can be mitigated by introducing the prior that objects are usually made of a limited number of distinct materials. This allows for regularization of the material representation in feature space, promoting smoothness and sparsity, and in image space, promoting local consistency.

Specifically, we adopt material feature loss similar to

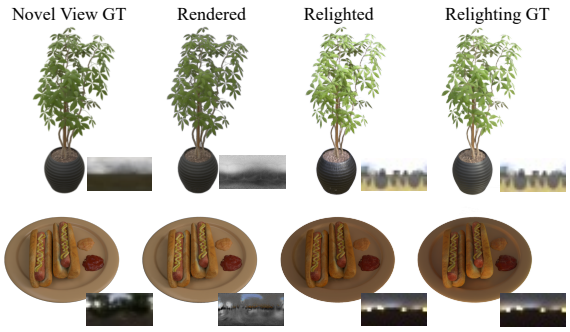

Figure 4: Novel view sysnthesis and relighting results produced by our proposed method.

Zhang *et al*. [53] as:

$$\begin{aligned}\mathcal{L}_{mat}^{F} =& \lambda_{mf1} \sum_{i=2}^{F} D_{KL}(\text{Bern}(0.05)||p(f_i)) \\ &+ \lambda_{mf2}||\text{Mat}(\boldsymbol{f}) - \text{Mat}(\boldsymbol{f} + \Delta\boldsymbol{f})||_1,\end{aligned} \tag{8}$$

here, $p(\boldsymbol{f}_i)$ is calculated as the mean of the i-th channel of the positional feature $\boldsymbol{f}$, which represents the probability of non-zero values. The sparsity loss minimizes its KL-divergence with a target Bernoulli distribution with probability 0.05, encouraging zero-values for a total of $F-1$ feature channels (excluding the first channel, which represents the SDF value). The smoothness loss encourages similar latent codes (differentiated by a small $\Delta\boldsymbol{f} \sim \mathcal{N}(0, \epsilon)$) to be mapped to similar material parameters through MaterialNet.

To regularize material in image space, we sample half of our rays using a patch-based method from images, to ensure the sampled points have similar roughness and metallic properties. Additionally, we limit the positions where occlusion can appear due to ambient occlusions, and regularize its amplitude accordingly. Formally, the material regularization in image space can be expressed as:

$$\begin{aligned}\mathcal{L}_{mat}^{I} =& \sum_{i=1}^{P} (\lambda_{mid}\delta_i(\boldsymbol{k}_d) + \lambda_{mir}\delta_i(r) + \lambda_{mim}\delta_i(m)) \\ &+ \lambda_{mio}||o||_2^2,\end{aligned} \tag{9}$$

where δ_i calculates the standard deviation of i-th image patch. We calculate the full material regularization as: $\mathcal{L}_{mat} = \mathcal{L}_{mat}^{F} + \mathcal{L}_{mat}^{I}$

To regularize the environment cubemap, we apply a white environment prior and regularize it using its mean absolute error (MAE). Specifically, we use the loss function $\mathcal{L}_{light} = \lambda_l \text{MAE}(I_{base})$, where I_{base} is the learned environment cubemap at the 0-th mipmap level.

In a nutshell, the total loss function we use in this paper

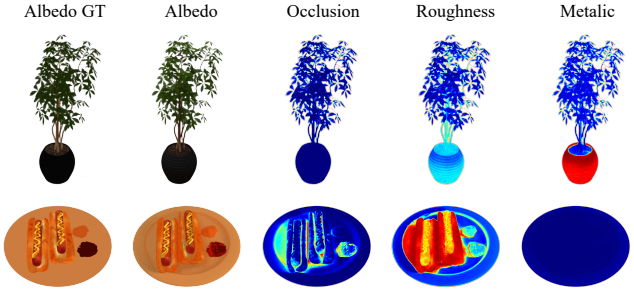

Figure 5: An example of material factorization by our proposed method.

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	<u>0.924</u>	0.092
NVDiffrecmc	<u>26.20</u>	**0.928**	**0.054**	**25.34**	**0.931**	0.072
Ours	**26.23**	<u>0.925</u>	<u>0.058</u>	<u>24.86</u>	0.921	<u>0.066</u>
w/o L_{mat}	26.07	0.923	0.062	24.60	<u>0.924</u>	**0.064**

Table 2: Quantitative evaluation on NeRFactor's synthesis dataset. Both NVDiffrec and NeRFactor metrics are as reported in [33] (In bold: best; Underline: second best).

is formulated as follows.

$$\mathcal{L} = \mathcal{L}_{\boldsymbol{c}_{mat}} + \mathcal{L}_{\boldsymbol{c}_{tex}} + \mathcal{L}_{mask} + \mathcal{L}_{sdf} + \mathcal{L}_{mat} + \mathcal{L}_{light}. \tag{10}$$

4. Experiments

4.1. Baselines

Our work is primarily related to two methods, NVDiffrec [33] and Neural-PIL [7]. Both of these 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 [22] adopts Monte Carlo Rendering for shading based on NVDIffrec's geometry and material representation, SAMURAI [6] additionally estimates camera pose for inverse rendering. We compare the above methods together with NeRFactor [52] and InvRender [54]. 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).

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

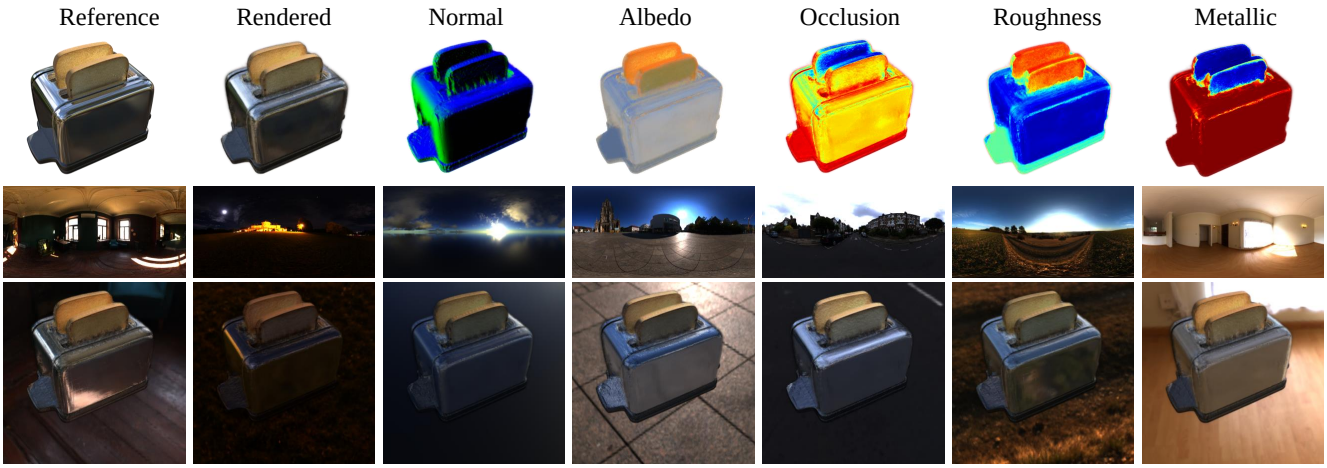

Figure 6: Material factorization and relighting on Ref-NeRF's Shiny scene. Top row: material factorization results. Middle row: high-frequency environment maps used for relighting. Bottom row: relighted results under the given corresponding illuminations for each column.

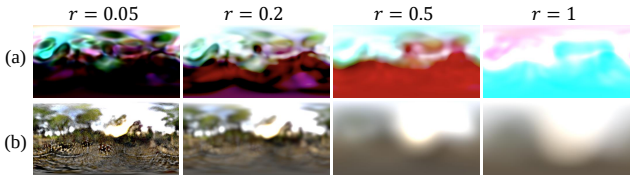

Figure 7: Different illumination modeling methods. (a) Our method equipped with Neural-PIL [7]'s environment modeling method. (b) Our proposed method.

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 [52], four synthetic scenes originally released by NeRF [31] are relighted with eight different low-frequency environment illuminations, and evaluated over eight uniformly sampled novel views. We compare our method with NeRFactor [52], NVDiffrec [33], and NVDiffrecmc [22] 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 scale the predicted albedo image by a color-tuning factor that matches the average of ground-truth albedo. As shown in Table 2, our method outperforms both NeRFactor and NVdiffrec and achieves on-par relighting performance with NVDiffrecmc. We attributes the performance gain by 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 method for comparison. Qualitative result are visualized in Figure 4, where we render the image using constructed environment map from 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 [40], 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. Figure 4 shows the material factorization and relighting results on the toaster scene. The material of both the bread and toaster 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). As shown in Figure 1, our reconstructed geometry is even better than NeuS. We attribute this advantage to the fact that our method models the reflected illumination during training. This makes it easier for the model to interpret shiny objects.

4.4. Experiment on Real-World Dataset

To evaluate our method on real-world scenes, we follows Wang *et al*. [42] in adopting Common Objects in 3D (CO3D) dataset [35] and evaluating on a subset of cars. The

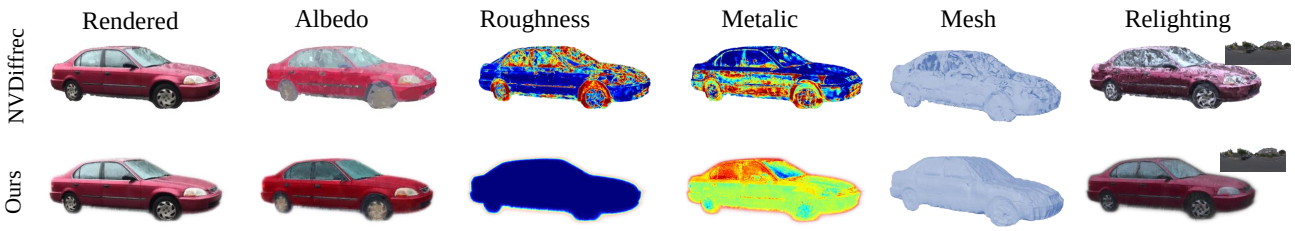

Figure 8: A comparison between our proposed method and NVDiffrec [33] on real-world dataset – CO3D [35]. Although NVDiffrec shows promising results on rendered novel views, it fails to reconstruct a plausible mesh. The geometry artificats lead to noisy material factorization and unrealstic relighting. In contrast, our method uses high-quality neural implicit field representation and jointly optimize geometry, material and illumination in a collaborative manner, resulting in better overall results.

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 [33], NVdiffrecmc [22], SAMURAI [6], NeuralPIL [7], and InvRender [54] on a subset of 10 car scenes with relatively complete 360° viewing directions in CO3D dataset. As shown in Table 3, our results are significantly better than other methods on novel view synthesis. It is worth noting NVdiffrecmc performs worse than NVDiffrec because NVdiffrecmc enforces additional regularization. A more detailed qualitative comparison with most related NVDiffrec method is visualize in Figure 8. Although its rendered novel-view seems realistic in 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 fixed number of vertices to represent geometry, limiting its ability to represent geometry and behaving unstably under limited views.

4.5. Ablation Study

Material 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 Table 2. We observe only limited performance gain in relighting and albedo reconstruction in terms of PSNR. We attribute this to the fact that the dataset already contains high-quality input views to regularize materials.

Method	Novel View		
	PSNR↑	SSIM ↑	LPIPS ↓
NVDiffrec [33]	26.29	0.925	0.086
NVDiffrecmc [22]	24.45	0.911	0.107
SAMURAI [6]	24.88	0.901	0.118
NeuralPIL [7]	25.42	0.915	0.092
InvRender [54]	24.94	0.919	0.092
Ours	**29.03**	**0.935**	**0.046**

Table 3: Quantitative evaluation on CO3D dataset. [35]

Neural-PIL vs. Pre-computed Environment map. We compare our choice of using trainable environmental cubemap and explicitly computing the pre-integration as the lighting representation with Neural-PIL, which uses FILM-SIREN layers [7] 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 levels of roughness. 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 Figure 7, when queried at roughness 1, the environment map is drastically different from the previous levels of pre-filtered environment maps and has the complimentary color. To regularize illumination, Neural-PIL resorts to train another auto-encoder network to learn illumination latent codes using data-driven methods. While our implementation can learn high-quality illumination, which is naturally consistent across different levels of roughness, purely from the scene.

5. Conclusion

In this paper, we present a novel method called NeuS-PIR for effectively learning relightable neural surfaces using pre-integrated rendering. We propose to simultaneously train a neural implicit surface, a spatially varying material field, and a differentiable environment cubemap using pre-integrated rendering. Our experiments demonstrate that our proposed method is superior to existing approaches in terms of reconstruction quality and relighting fidelity, which has potential to be applied in many applications in computer vision and graphics.

Acknowledgement

The authors would like to thank Jiadai Sun and Lina Liu for helping proofread the paper, Jin Fang for the discussion in the early stage of this project.

References

[1] Louis-Philippe Asselin, Denis Laurendeau, and Jean-François Lalonde. Deep svbrdf estimation on real materials. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, pages 1157–1166. IEEE, 2020.

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul 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.

[3] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020.

[4] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 294–311. Springer, 2020.

[5] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[6] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T. Barron, Hendrik P.A. Lensch, and Varun Jampani. SAMURAI: Shape And Material from Unconstrained Real-world Arbitrary Image collections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[7] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 10691–10704, 2021.

[8] Zhiqin Chen, Andrea Tagliasacchi, Thomas Funkhouser, and Hao Zhang. Neural dual contouring. *ACM Trans. on Graphics (TOG)*, 41(4):1–13, 2022.

[9] Zhiqin Chen and Hao Zhang. Neural marching cubes. *ACM Trans. on Graphics (TOG)*, 40(6):1–15, 2021.

[10] Evgeni Chernyaev. Marching cubes 33: Construction of topologically correct isosurfaces. Technical report, 1995.

[11] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 628–644. Springer, 2016.

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

[13] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 303–312, 1996.

[14] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5501–5510, 2022.

[15] Duan Gao, Guojun Chen, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deferred neural lighting: free-viewpoint relighting from unstructured photographs. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.

[16] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Trans. on Graphics (TOG)*, 2017.

[17] Michael Goesele, Brian Curless, and Steven M Seitz. Multi-view stereo revisited. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, volume 2, pages 2402–2409. IEEE, 2006.

[18] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 216–224, 2018.

[19] Tom Haber, Christian Fuchs, Philippe Bekaer, Hans-Peter Seidel, Michael Goesele, and Hendrik PA Lensch. Relighting objects from image collections. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 627–634. IEEE, 2009.

[20] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, pages 412–420. IEEE, 2017.

[21] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[22] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022.

[23] Eldar Insafutdinov, Dylan Campbell, João F Henriques, and Andrea Vedaldi. Snes: Learning probably symmetric neural surfaces from incomplete data. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings*, pages 367–383. Springer, 2022.

[24] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013.

[25] Ruilong Li, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: A general nerf accleration toolbox. *arXiv preprint arXiv:2210.04847*, 2022.

[26] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2475–2484, 2020.

[27] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2916–2925, 2018.

[28] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 7708–7717, 2019.

[29] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2022.

[30] Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. Unified shape and svbrdf recovery using differentiable monte carlo rendering. In *Computer Graphics Forum*, volume 40, pages 101–113. Wiley Online Library, 2021.

[31] 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 *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

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

[33] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8280–8290, 2022.

[34] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5589–5599, 2021.

[35] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David 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.

[36] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 501–518. Springer, 2016.

[37] 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 *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 6087–6101, 2021.

[38] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7495–7504, 2021.

[39] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5459–5469, 2022.

[40] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5481–5490. IEEE, 2022.

[41] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007.

[42] 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 *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[43] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. *arXiv preprint arXiv:2212.05231*, 2022.

[44] Zian Wang, Wenzheng Chen, David Acuna, Jan Kautz, and Sanja Fidler. Neural light field estimation for street scenes with differentiable virtual object insertion. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 380–397. Springer, 2022.

[45] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Trans. on Graphics (TOG)*, 38(4):1–13, 2019.

[46] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 13779–13788, 2021.

[47] Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf: Neural incident light field for physically-based material estimation. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 700–716. Springer, 2022.

[48] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Proc. of*

the Conference on Neural Information Processing Systems (NeurIPS), volume 34, pages 4805–4815, 2021.

[49] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5565–5574, 2022.

[50] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5453–5462, 2021.

[51] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

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

[53] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 18643–18652, 2022.

[54] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022.

Supplementary Material

The supplementary material contains additional details and results of our approach. Technical details are provided in Section 1. Our proposed method is further ablated and explained in Section 2. Additional auxiliary results are presented in Section 3. Limitations are discussed in Section 4.

1. Implementation Details

1.1. Network Architecture

Geometry Module. We encode the geometry using multiresolution hash encoding [32] with 16 levels and 2 features per level. The coarest resolution is 16 and hash table size is 2^{19}. The original position (scaled by a factor of 2 and offset by -1) is included in the positional encoding. The multilayer perceptron (MLP) that learns positional features after hash encoding has one hidden layer with 64 neurons activated by ReLU, and the output positional feature dimension is set to 13. The geometric weight initialization is adopted from NeuS [42]. We do not apply any activation function in the output layer because the first element of its output represents the SDF value, and its range should not be restricted.

Radiance Module. The outgoing radiance is learned through a MLP with 2 hidden layers of 64 neurons, activated by ReLU from concatenated inputs. The output 3-channel RGB radiance is then activated by Sigmoid, which restricts it to the range of $[0, 1]$. The inputs to radiance MLP include the viewing direction (using sphere harmonics up to the 4^{th} level), positional feature, and the surface's unit normal vector.

Material Module. The spatially varying material field is learned through a MLP with two hidden layers of 64 neurons activated by ReLU from positional feature computed by the geometry module. The output is a 6-channel material feature activated by Sigmoid, where the first three channels represent diffuse albedo $\boldsymbol{k}_d$ and the remaining three channels represent specular and occlusion properties $\boldsymbol{k}_s = \{o, r, m\}$ for occlusion, roughness and metallic respectively. We use a differentiable cubemap with a resolution of $512 \times 512 \times 6$ to represent the illumination. Its pixel value are clipped to be greater than 0 during training to represent valid illumination. For visualization, the cubemap is converted to 512×1024 High Dynamic Range (HDR) image using the lat-long conversion.

1.2. Training Details

During training, we adopt dynamic ray sampling to update number of rays used for training. The initial number of rays for training is set to 256 and limited up to a maximum of 8192. To prevent the cosine value between viewing direction and normal vector from becoming invalid due to non-negative clipping, we use NeuS convention to smooth the cosine value by a cosine annealing ratio in the first 5000 steps [42]. We perform ray-marching using an occupancy grid with resolution 128 within a $[-rad, rad]^3$ cube range, which is updated every step to prune empty spaces whose opaque density is less than a threshold of 0.001 [25]. We set random value for masked areas during training to stabilize the geometry learning. For training CO3D data, all scenes are processed and recentered as in [23], we use a bounding box with $rad = 1.2$. For training the Shiny and NeRFactor's relight dataset, we use $rad = 1.5$. The scaling factor τ for $\Phi_\tau(s)$ in calculating the opaque density is initialized as e^3 and updated using Adam optimizer with a learning rate of 0.001. During training, we update the parameters of geometry MLP, radiance MLP, and scaling factor τ by an Adam optimizer, while updating the parameter of material MLP and differentiable cubemap using another Adam optimizer. Each optimizer has its own warm-up stage of 500 steps, followed by an exponential decay period. The material and illumination modules are only enabled after the first 10K steps in a standard training protocol.

Method	Novel View		
	PSNR↑	SSIM ↑	LPIPS ↓
NVDiffrec[33]	26.29	0.925	0.086
Neural-PIL[7]	25.42	0.915	0.092
Ours-0K	29.15	0.937	0.034
Ours-1K	29.13	0.938	0.033
Ours-10K (baseline)	29.03	0.935	0.046
w/o L_{mat}	29.16	0.937	0.035
w/o L_{light}	29.10	0.937	0.034

Table 4: Ablation study on CO3D dataset [35]

2. Additional Ablation Study

Additional Regularization. We present an ablation study on material and illumination regularizations for the CO3D dataset in Table 4. The results show a slight performance gain on novel view synthesis with fewer regularization.

Training Strategy. We vary our training strategies by introducing the material module during different phases of training. Table 4 shows the results on CO3D dataset with the introduction of the material module at 0, 1K, and 10K steps of training. The results show a slight performance gain when introducing the material module in the earlier phase. We attribute it to the fact that jointly optimizing geometry, material and illumination in a single-stage manner benefits the performance.

Method	Relighting			Albedo		
	PSNR↑	SSIM ↑	LPIPS ↓	PSNR↑	SSIM ↑	LPIPS ↓
NVDiffrec[33]	21.28	0.853	0.112	22.52	0.872	0.128
Ours	**24.73**	**0.932**	**0.041**	**24.80**	**0.931**	**0.036**

Table 5: Quantitative evaluation on the *car* scene of Shiny dataset.

Method	Relighting			Albedo		
	PSNR↑	SSIM ↑	LPIPS ↓	PSNR↑	SSIM ↑	LPIPS ↓
NVDiffrec[33]	17.50	0.772	0.213	15.49	0.848	0.179
Ours	**18.07**	**0.830**	**0.143**	**16.02**	**0.863**	**0.161**

Table 6: Quantitative evaluation on the *toast* scene of Shiny dataset.

3. Additional Results and Comparisons

3.1. Experiments on Synthetic Dataset

We further compare our method with NVDiffrec [33] on relighting and albedo reconstruction quality in two Shiny scenes. Quantitative results are shown in Table 5, 6 demonstrates our method consistently outperforms NVDiffrec [33] in these two scenes. Additionally, we visualize the geometry, material, and illumination decomposition as well as the relighting results in 7 different novel illuminations. Our relighting results are more plausible due to better learned geometry and material. The learned environment maps also have higher details, we believe are results of the learned high-quality geometry.

3.2. Experiments on Real-World Dataset

We provide detailed comparisons with our most related baseline methods: NVDiffrec [33] and Neural-PIL [7], on a real-world dataset. Specifically, we use car scenes from CO3D with IDs *425_59326_114632, 244_25999_52630, 417_57648_111091,415_57125_110159, 106_12662_23043, 336_34852_64130, 157_17286_33548, 421_58388_112532, 336_34811_63015, and 351_37072_67647* for evaluations. The quantitative novel view synthesis result are shown in Table 4, where our method outperforms both NVDiffrec [33] and Neural-PIL [7]. For Neural-PIL, We use the official codes and train each scene with 400K steps as default. The qualitative results are shown in Figure 11,12,13, and 14. Note Neural-PIL adopts a different material representation, so we do not visualize its roughness and specular maps using our jet colormap. The roughness value of Neural-PIL ranges from 0 to 1 and is mapped from black to white, and its specular value is the specular color. We observe that the environment map learned by Neural-PIL [7] lacks high-frequency details and tends to be blue-tinted. This may due to the bias of the dataset used to train the environment illumination latent. Our generated mesh tends to preserve smoothness while capturing geometry details in most of the scenes, while NVDiffrec [33] tends to be noisy and Neural-PIL [7] tends to be over-smoothed.

4. Limitations

Although our method models direct illumination with high-frequency details, it is still challenging to handle complicated indirect illumination. Further investigation to account for self-occlusion and shadowing might help address the issue. We also observe unsatisfactory quality when the number of views is limited, which might be resolved by integrating data augmentations and diffusion methods. While our method can handle wild data, it is sometimes not very stable, improving the robustness when dealing with wild data is also important. We use marching cube algorithm [10] to export explicit representation (i.e., mesh), which may result in loss of accuracy. Recent methods such as neural marching cube [9] and neural dual contouring [8] might help to export better explicit results for downstream applications.

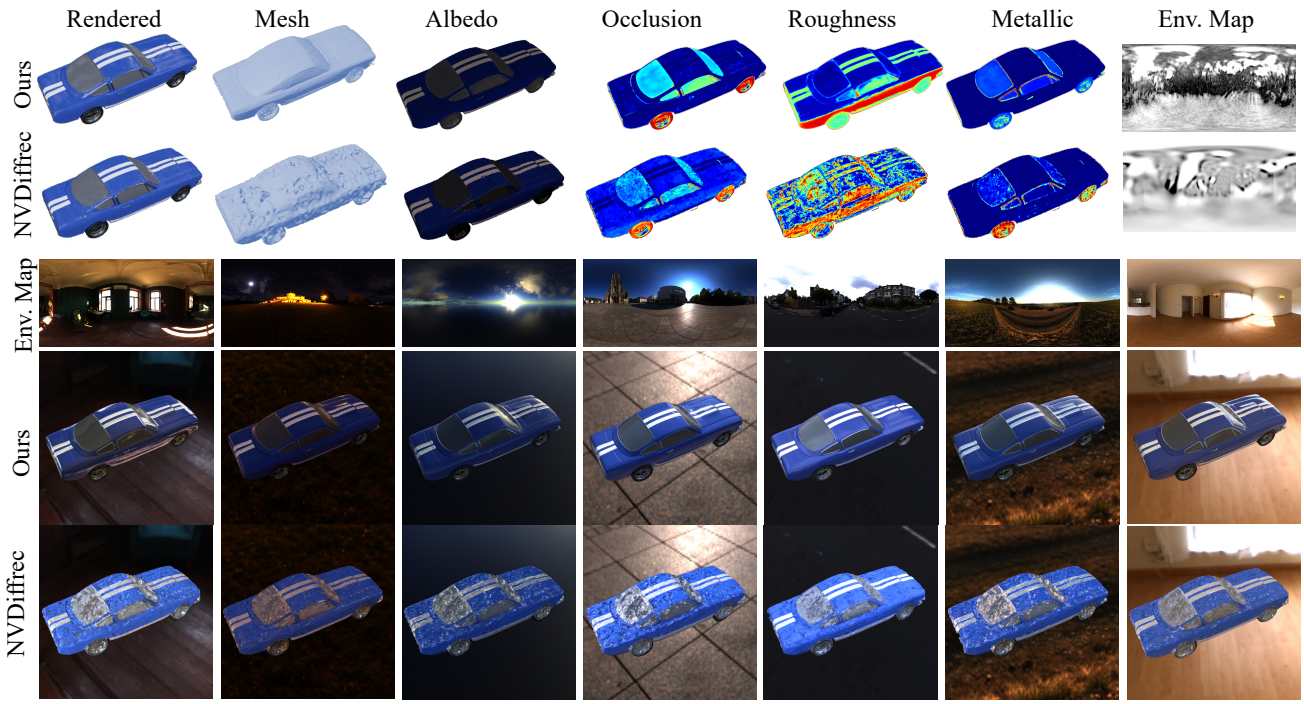

Figure 9: Visualization of Shiny Car Scene

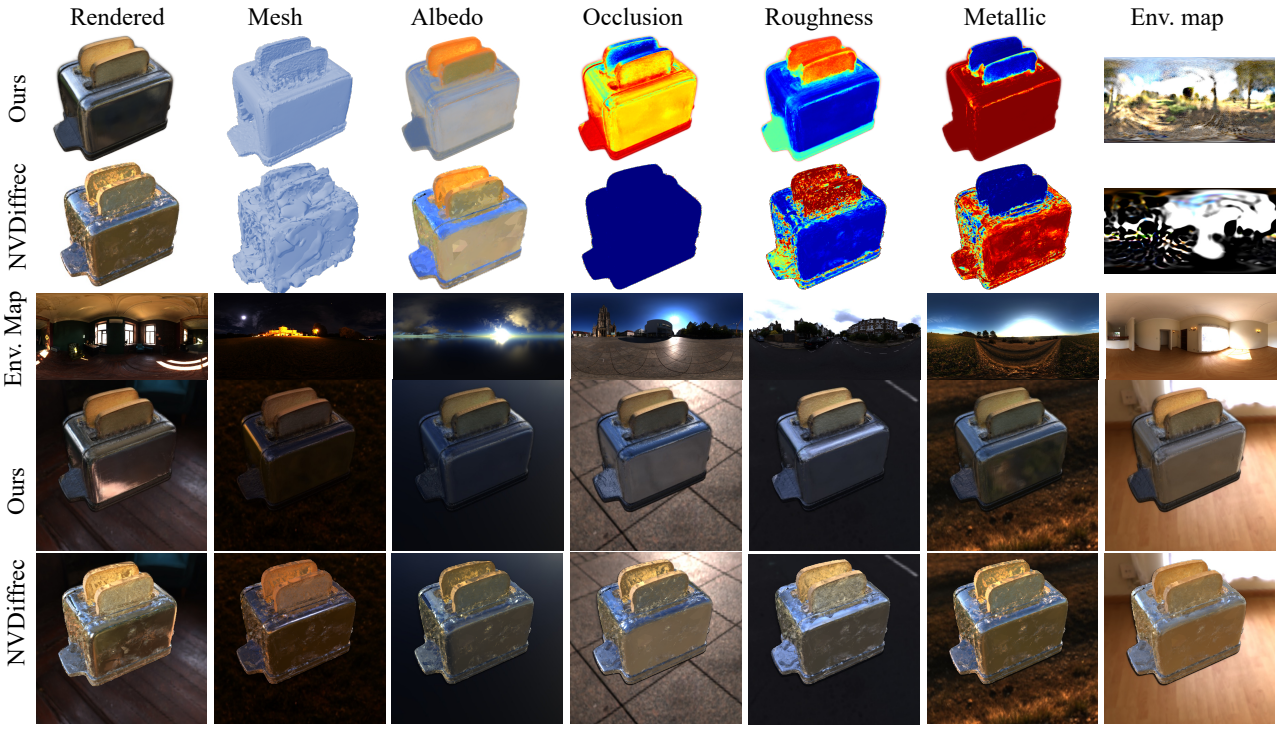

Figure 10: Visualization of Shiny Toaster Scene

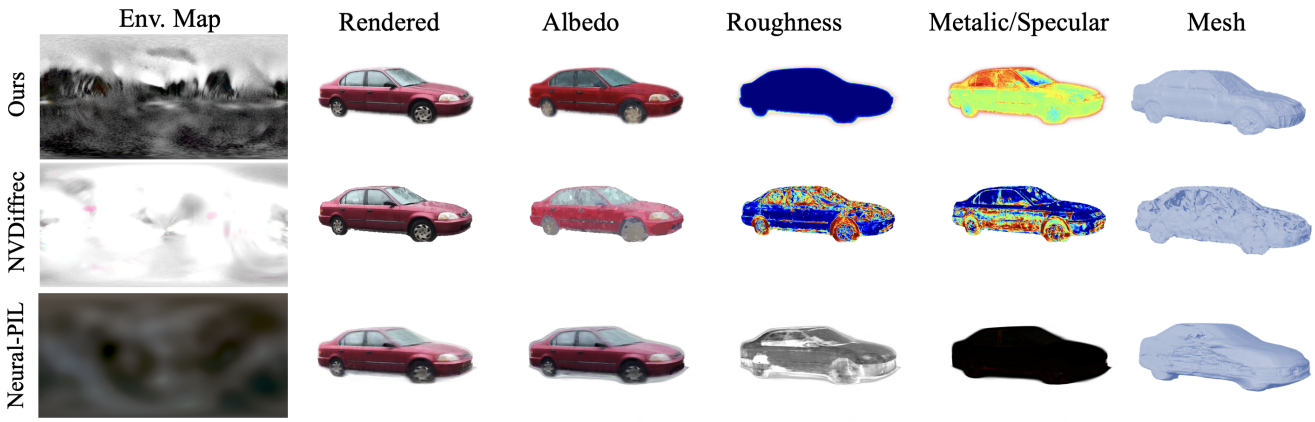

Figure 11: Visualization of scene car 425_59326_114632 in CO3D dataset

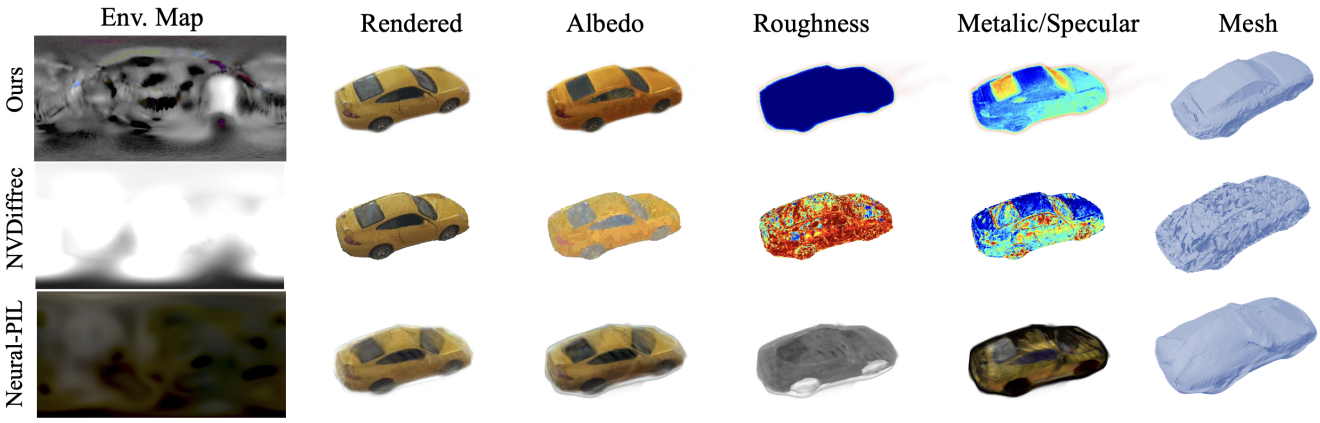

Figure 12: Visualization of scene car 351_37072_67647 in CO3D dataset

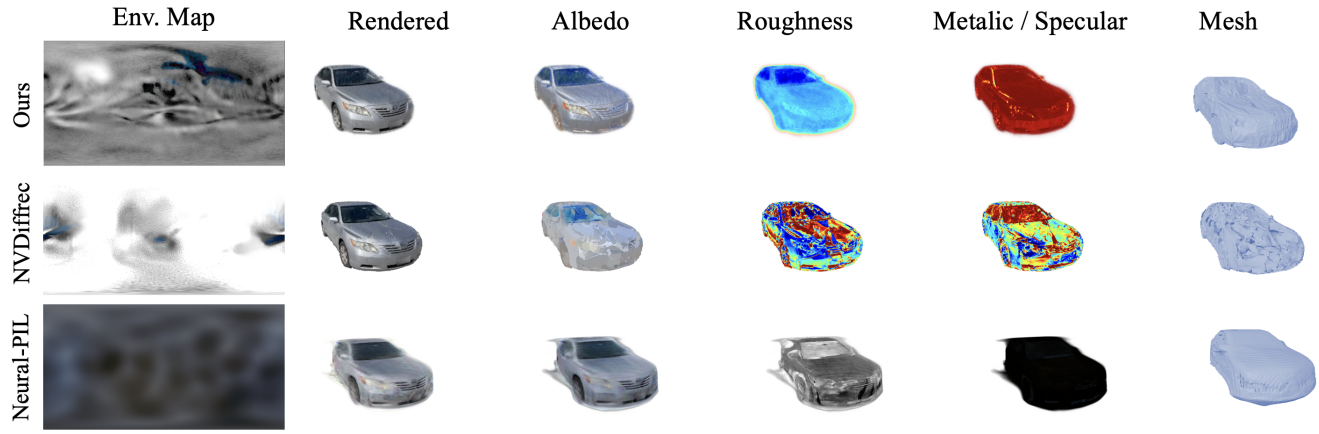

Figure 13: Visualization of scene car 244_25999_52630 in CO3D dataset

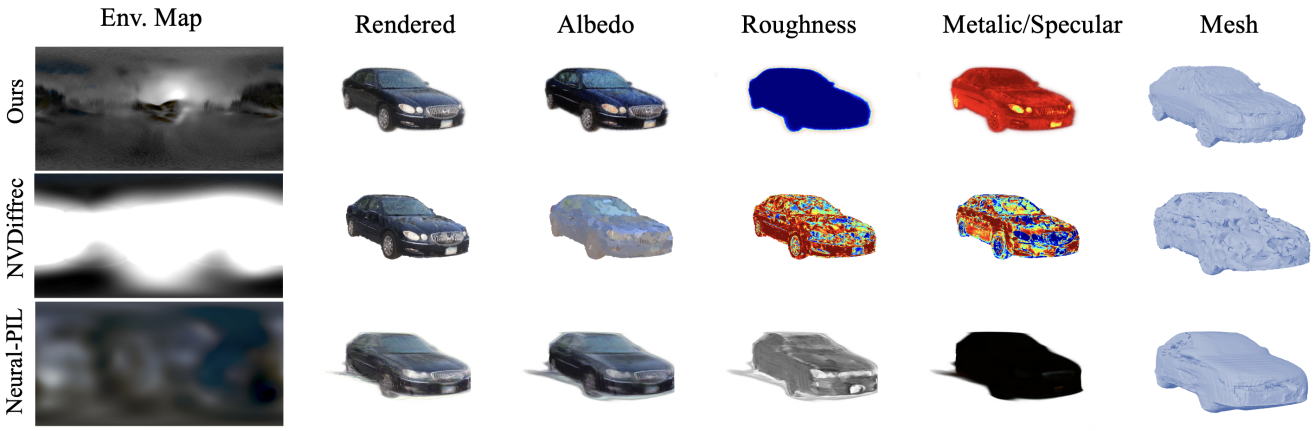

Figure 14: Visualization of scene car 415_57125 _110159 in CO3D dataset