Multi-times Monte Carlo Rendering for Inter-reflection Reconstruction

Tengjie Zhu *   **Zhuo Chen** *   **Jingnan Gao**   **Yichao Yan**[†]   **Xiaokang Yang**

MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
`{zhutengjie, ningci5252, gjn0310, yanyichao, xkyang}@sjtu.edu.cn`

Abstract

Inverse rendering methods have achieved remarkable performance in reconstructing high-fidelity 3D objects with disentangled geometries, materials, and environmental light. However, they still face huge challenges in reflective surface reconstruction. Although recent methods model the light trace to learn specularity, the ignorance of indirect illumination makes it hard to handle inter-reflections among multiple smooth objects. In this work, we propose Ref-MC^2 that introduces the multi-times Monte Carlo sampling which comprehensively computes the environmental illumination and meanwhile considers the reflective light from object surfaces. To address the computation challenge as the times of Monte Carlo sampling grow, we propose a specularity-adaptive sampling strategy, significantly reducing the computational complexity. Besides the computational resource, higher geometry accuracy is also required because geometric errors accumulate multiple times. Therefore, we further introduce a reflection-aware surface model to initialize the geometry and refine it during inverse rendering. We construct a challenging dataset containing scenes with multiple objects and inter-reflections. Experiments show that our method outperforms other inverse rendering methods on various object groups. We also show downstream applications, *e.g.*, relighting and material editing, to illustrate the disentanglement ability of our method. Our project page: `https://zhutengjie.github.io/Ref-MC2/`.

1   Introduction

Neural Radiance Fields (NeRF) [26] and 3DGS [16] have demonstrated their excellent performance on novel view synthesis. However, it is difficult to directly apply their reconstructed 3D model to the current industrial pipeline, leading to the lack of flexibility in many downstream applications, *e.g.*, relighting and material editing. To better cooperate with mature techniques, inverse rendering bases the physical rendering [24] and utilizes the neural network to learn disentangled materials that can be seamlessly plunged into the industrial pipeline for further manipulation.

Previous methods [28, 9, 45, 57] have explored how to disentangle geometry, diffuse, roughness, metalness, and environmental light from multi-view images, but they still face challenges in shading and reflective objects. Specifically, Nvdiffrec [28] proposes a differentiable pipeline that enables gradient-based optimization on both meshes and volumetric textures. These methods ignore the shadow when modeling the illumination, resulting in failed disentanglement for diffuse and shading appearances. To model more realistic shadings, Nvdiffrecmc [9] further incorporates ray tracing and Monte Carlo sampling [27] into inverse rendering, significantly improving the decomposition of shape, materials, and lighting. However, it ignores the indirect illuminations during path tracing

---

and treats rays attacking object surfaces as sheltered illumination. As a consequence, methods like Nvdiffrecmc increase the ambiguity of geometry reconstruction and undermine material learning. Unfortunately, the scene with multiple inter-reflections is common in the real world. The failure in these scenes hinders the wider applications of these methods. Recent methods [17, 40, 22] take inter-reflections into account and can reconstruct the reflective objects well. However, the implicit representations of these methods for materials and renders sacrifice their scalability to downstream tasks. Recently, Nefii [45] further incorporated the implicit neural radiance to estimate ray tracing, alleviating the ambiguity between materials and indirect illuminations. However, this leads to a huge computational consumption for path tracing. Neural Microfacet Fields [22] employs two-bounce sampling for indirect light modeling, but its geometry field limits flexibility for downstream tasks.

In this paper, we proposed a full inverse rendering method, Ref-MC^2, which considers inter-reflections during ray tracing to improve the decomposition of explicit materials and environmental lighting. The core of our method is to use **Multi-times Monte Carlo integration** [27] and **BRDF** [24] rendering to approximate the indirect illumination at multiple reflection points along the light propagation path. Although our method takes advantage of hardware-accelerated ray tracing to model the indirect illumination, we still face two major challenges brought by the multi-times Monte Carlo sampling. 1) **Efficiency**: the explosive computational growth from multiple times is too heavy for the hardware algorithm only. 2) **Geometry**: the geometry quality greatly affects the calculation of indirect light and the decomposition of the materials, because the error will accumulate over and over again as the times of Monte Carlo sampling increase.

To address these challenges, we correspondingly propose two strategies. 1) For efficiency, we flatten the multi-times sampling into sequential single-time sampling. In a Lambert model [31], the diffuse light is independent of the direction of reflection and thus can be presented as a map to query at any time. When we trace the indirect light from an object, instead of recalculating the diffuse component, we can take the value directly from the diffuse map. It can be optimized through self-supervision. Therefore, we only need to sample the specular component within a small lobe along around the reflective direction. 2) For geometry, we refer to the SDF-based methods [41, 17, 6] and replace the common positional encoding with the Sphere Gaussian encoding to get an accurate initial geometry for reflective objects. We use this geometry to initialize Flxicubes [35] that optimizes surface meshes based on the gradient, and further fine-tunes the Flexicubes in a differentiable pipeline. To evaluate our framework, we construct a dataset containing difficult scenes which contain strong inter-reflections between multiple surfaces. Extensive experiments demonstrate that our framework can successfully decompose indirect illumination and materials. In summary, our contributions are:

- We propose a full inverse rendering method that employs multi-times Monte Carlo sampling to correctly decompose indirect illumination and materials.
- We reduce the computational consumption when tracing indirect illumination by self-supervising the diffuse map based on the Lambert model.
- We refine the SDF-based architecture with Spherical Gaussian encoding to obtain a high-quality initial geometry which further releases the accumulated error during multi-times sampling.
- We construct a dataset to evaluate the performance on indirect illumination.

2 Related Work

2.1 Implicit Neural Representations

Neural implicit representations [23, 26, 32, 55, 44, 34, 10, 3, 4, 58, 36, 7] have achieved impressive success in many computer vision and computer graphics tasks. These methods use neural radiance fields to capture color and volume density, generating photo-realistic novel views through volume rendering [15]. However, the unconstrained volumetric representation of the original NeRF method leads to low-quality geometry. Recent 3D Gaussian Splatting (3DGS) [16, 11, 12, 20, 53] has gained popularity in novel view synthesis. Different from NeRF, it is an explicit representation that involves the optimization of multiple Gaussians to reconstruct 3D objects. 3DGS learns color and density in a volumetric point cloud, but it also fails to produce accurate geometry due to its discrete representation. Besides, these methods are all entangled learning, integrating all inherent materials and the environment map into the appearance. This limits the downstream applications in the current industrial pipeline. To produce a high-quality geometry, follow-up works [48, 43, 51, 30] use a

function to associate signed distance field (SDF) and volume density. The surface mesh can be extracted from neural implicit surfaces by Marching Cubes [19], and this 3D asset can be further applied in other applications. However, they usually perform badly in reconstructing the reflective objects due to the ambiguity of reflective appearance.

2.2 Neural Inverse Rendering

Although neural implicit surfaces have achieved impressive performance in geometry reconstructing and novel views synthesis, they do not obtain the fundamental materials of PBR which limits their flexibility in downstream tasks. Neural inverse rendering methods [17, 46, 8, 2, 54, 57, 13, 38, 21] introduce the physical rendering equation to estimate the disentangled diffuse and specular component from RGB images. They approximate the rendering equation based on neural networks or basis functions, *e.g.*, Spherical Gaussians [42, 47, 49, 54] and Spherical Harmonics [5, 1, 37, 52]. Nvdiffrec [28] introduces the **full inverse rendering** that estimates shape, materials, and environmental light into gradient-based optimization. However, it does not consider the shadows, leading to the entanglement of materials. Recent works [6, 41] extend SDF-based architectures with an additional appearance branch to model the reflections on the object. RefNeuS [6] introduces a reparametrization method to distinguish the reflective appearance from the diffuse appearance by a direction-relative process, but it fails to faithfully reconstruct non-reflective objects. UniSDF [41] proposes to use a weight-MLP to balance the reflective and non-reflective branches for different objects. However, these methods still face challenges in scenes with complex inter-reflections. Nvdiffrecmc [9] extends Nvdiffrec with Monte Carlo sampling to trace the light path but still ignores indirect illumination between objects. Further methods [17, 18, 45] consider indirect illumination in their design. ENVIDR [17] employs a neural renderer to learn the physical light interaction, without explicitly formulating the rendering equation. NeRO [18] applies the split-sum approximation to approximate the shading effects of both direct and indirect lights. Nefii [45] introduces ray tracing to the radiance field to model indirect illuminations. Neural Microfacet Fields [22] employs two-bounce sampling to accurately calculate indirect illumination. However, NMF represents geometry by a density field, which limits the scalability and flexibility for downstream tasks. In addition, these aforementioned methods do not fully disentangle the materials from RGB images, but decompose the appearance into reflective and diffuse color. It is hard to apply to the current industrial pipeline directly.eq: rendering equation In contrast, our Ref-MC^2 is a full inverse rendering method that also considers indirect illumination.

3 Method

3.1 Preliminaries

The rendering equation [14] is commonly used to compute the outgoing radiance $L_o(\mathbf{p}, \boldsymbol{\omega}_o)$ from the point $\mathbf{p}$ in outgoing direction $\boldsymbol{\omega}_o$:

$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) = \int_\Omega L_i(\mathbf{p}, \boldsymbol{\omega}_i) f(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) (\mathbf{n} \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \tag{1}$$

where $L_i(\mathbf{p}, \boldsymbol{\omega}_i)$ is the incoming radiance into $\mathbf{p}$ from the direction $\boldsymbol{\omega}_i$, $\mathbf{n}$ is the normal of the point $\mathbf{p}$, Ω is the hemisphere of directions above $\mathbf{p}$, $f(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ is the BSDF evaluated for $\boldsymbol{\omega}_i$ and the current incoming direction $\boldsymbol{\omega}_i$. The GGX [24] physics-based BSDF function is proposed to decompose the function into several physical terms. The function is described as:

$$f(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = f_d + f_s = f_d + \frac{DFG}{4(\mathbf{n} \cdot \boldsymbol{\omega}_i)(\mathbf{n} \cdot \boldsymbol{\omega}_o)}, \tag{2}$$

where f_d is the diffuse term and f_s is the specular term. D, F, and G are the microfacet distribution function, the Fresnel reflection coefficient, and the geometric attenuation, respectively.

Previous works [28] use the split sum function [25] to approximate the rendering equation, but it inevitably omits the shadow and indirect illumination. In contrast, Monte Carlo integration [27] is a simple but unbiased estimation method that comprehensively considers all the physical terms for the outgoing radiance. The Monte Carlo integration rendering equation is:

$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) \approx \frac{1}{N} \sum_{i=1}^N \frac{L_i(\mathbf{p}, \boldsymbol{\omega}_i) f(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) (\boldsymbol{\omega}_i \cdot \mathbf{n})}{p(\boldsymbol{\omega}_i)}, \tag{3}$$
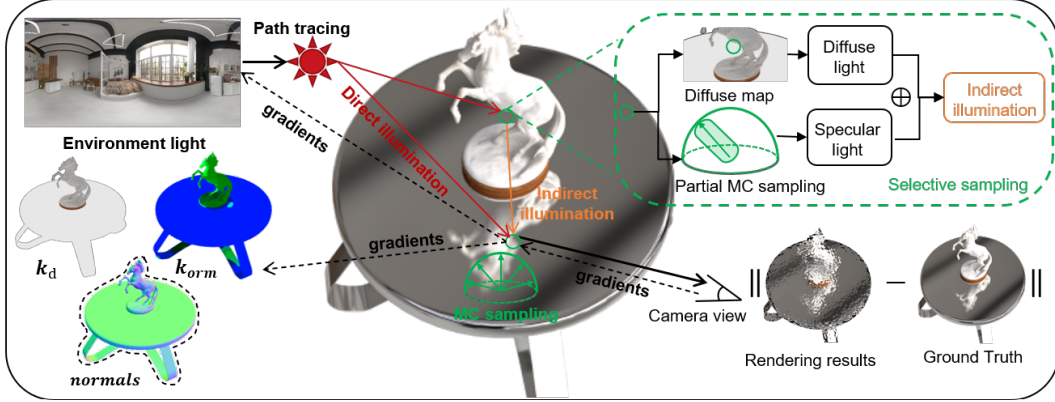
Figure 1: We perform Monte Carlo sampling at the viewpoint. When the sampling ray from the point is not blocked, it is the direct illumination from environmental lighting. When the sampling ray hits an object, we divide this indirect illumination from the object into diffuse light and specular light. We sample the diffuse light from a diffuse map that is optimized through self-supervision. For specular light, we only need to partially trace the rays in a small specular lobe along the reflective direction. The gradients are backward along the tracing path, and are passed to optimize k_d, k_{orm}, normals, and environment maps.

where $\boldsymbol{\omega}_i$ is the ith sample drawn from density p. As the number of samples grows, the estimation variance reduces, but the computation increases. Multiple importance sampling [39] (MIS) is proposed to inhibit the computational consumption. When a function can be expressed as a multiplication of n functions, it draws n_i samples $\boldsymbol{\omega}_{i,j}$ from n sampling distributions p_i in turn. The MIS Monte Carlo estimator for the rendering equation is:

$$L_o\left(\mathbf{p}, \boldsymbol{\omega}_o\right) = \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} W_i \frac{F_i\left(\mathbf{p}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_{i,j}\right)}{p_i}, \tag{4}$$

where $p_i \propto$ the i-th multiplication function of the integrated function of the rendering equation and W_i is the the balance heuristic weighting function.

3.2 Multi-times Monte Carlo Sampling

As discussed in Sec. 3.1, the split sum approximation ignores the shadows and the indirect illumination. Nvdiffrecmc notes this and accounts for shadows using the Monte Carlo integration method. However, they give up continuously tracing the light rays attacking the surface of objects and treat them as zero illumination. It leads to bad performance in scenes with inter-reflections. Therefore, we propose the Ref-MC^2 method to trace the light rays continuously. When taking into account indirect illumination, based on [14], the rendering equation can be expressed in this version:

$$L_o\left(\mathbf{p}, \boldsymbol{\omega}_o\right) = \int_{\Omega} L_i\left(r\left(\mathbf{p}, \boldsymbol{\omega}_i\right), -\boldsymbol{\omega}_i\right) f\left(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o\right)\left(\mathbf{n} \cdot \boldsymbol{\omega}_i\right) d\boldsymbol{\omega}_i. \tag{5}$$

In this equation, $r\left(\mathbf{p}, \boldsymbol{\omega}_i\right)$ represents the location of a surface point on the object surface hit by a ray cast from $\mathbf{p}$ in direction $\boldsymbol{\omega}_i$ for the first time. The corresponding Monte Carlo integration that considers the indirect illumination can then be expressed as:

$$L_o\left(\mathbf{p}, \boldsymbol{\omega}_o\right) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{L_i\left(r\left(\mathbf{p}, \boldsymbol{\omega}_i\right), -\boldsymbol{\omega}_i\right) f\left(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o\right)\left(\boldsymbol{\omega}_i \cdot \mathbf{n}\right)}{p\left(\boldsymbol{\omega}_i\right)}. \tag{6}$$

It can be interpreted as that when the sampling ray is not blocked, it can be seen as direct illumination from environmental lighting. When the surface point of an object blocks the sampling ray, it needs to be continuously traced from this surface point. This continuous tracing is an iterated process of Monte Carlo integration. It is noteworthy that when continuous sampling at the blocking surface point it is

(a) The architecture of previous SDF works   (b) The architecture for inter-reflection   Geometry Comparison
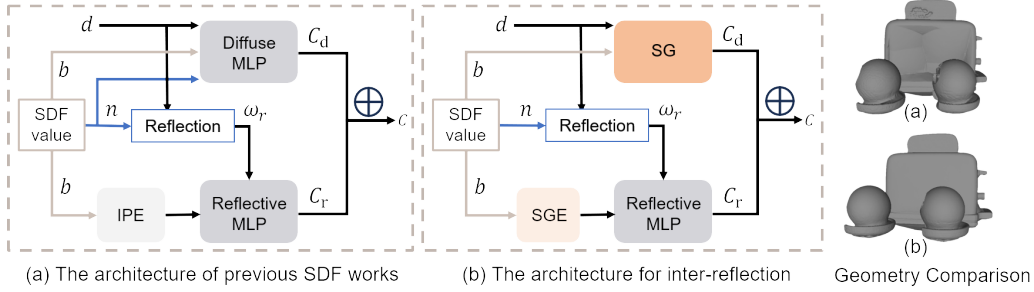
Figure 2: **Differences** between previous SDF architectures and the architecture for inter-reflections.

unnecessary to consider the indirect illumination like calculating the outgoing radiance at initial point **p**. This is because the energy of light gradually degrades and the impact of light after twice reflections is negligible compared to the explosively increased computational load. However, the additional computational load of ray tracing at a depth of two is still huge. To reduce our computational load, we further propose to approximate the diffuse light and transfer the computations.

In Disney PBR equation [24], the diffuse term f_d is:

$$f_d = \frac{c_{\text{diff}}}{\pi} \left(1 + (F_{D90} - 1)(1 - (\mathbf{n} \cdot \boldsymbol{\omega}_i))^5 \right) \left(1 + (F_{D90} - 1)(1 - (\mathbf{n} \cdot \boldsymbol{\omega}_o))^5 \right), \tag{7}$$

$$F_{D90} = 0.5 + 2r \cos^2 \theta_d, \tag{8}$$

where c_{diff} is the diffuse albedo of the material. In this equation, f is related to the direction of the incoming radiance, which can yield more realistic results. However, using this equation to calculate the diffuse light for indirect illumination creates unnecessary extra computation. Because the term f_d is related to the normal n and the direction of rays ω_o, we cannot get the diffuse map before deep ray tracing. The following term is to introduce the roughness for diffuse light to avoid too dark edges in extremely low grazing angles. However, using this equation to compute the diffuse light for indirect illumination introduces unnecessary extra computation. To reduce the burden, we can follow H and approximate it using Lambertian diffuse lighting.:

$$f_d^{\text{ind}} = \frac{c_{\text{diff}}}{\pi}. \tag{9}$$

Where f_d^{ind} is the diffuse light of the indirect illumination. In this approximation, f_d^{ind} is independent on the incoming direction ω_i and the outgoing direction ω_o. Based on this, the diffuse part of indirect light can be simplified as:

$$L_{\text{diff}}^{\text{ind}}(\mathbf{p}) = \frac{c_{\text{diff}}}{\pi} \int_{\Omega} L_i(\mathbf{p}, \boldsymbol{\omega}_i)(\mathbf{n} \cdot \boldsymbol{\omega}_i) \, d\boldsymbol{\omega}_i, \tag{10}$$

where $L_{\text{diff}}^{\text{ind}}$ is the diffuse part of the indirect illumination. It no longer relates to the direction of the sampling ray, so we can present it via an MLP M. Compared to the diffuse part, the specular part is strongly related to the ray direction. It is a disadvantage that we cannot approximate the specular part like the diffuse part, but it is also an advantage that we only need to sample minor rays in a small specular lobe along the reflective directions. Therefore, we significantly reduce the computations of both the diffuse part and the specular part during the multi-times Monte Carlo Sampling.

3.3 Geometry Initialization

As seen in the equation 5, our indirect lighting highly depends on the geometry. However, the strong ambiguity of reflections makes it hard to directly learn high-quality disentangled shapes. Therefore, we flatten the joint learning process of both geometry and materials into separate learning of them and thus reduce the number of physical terms to be disentangled in each learning stage.

We first utilize SDF-based architectures [48, 6, 18] to learn an initial geometry. The two branches of the diffuse and reflective networks well disambiguate the appearance with reflections and empower the

SDF network to produce a high-quality geometry, as shown in Fig. 2 (a). However, this architecture encounters challenges of indirect illumination due to the expressive capacity of Integrated Positional Encoding (IPE). It performs well in general scenes but shows limitations in representing inter-reflections. Due to the capacity of Spheical Gaussians (SG) [50, 33] to represent radiance directions, we introduce SG encoding instead of IPE to enhance the expressive capacity of reflective MLP, as shown in Fig. 2 (b). Besides, we directly parameterize diffuse appearance as SG coefficients which is suitable for objects with multiple reflective surfaces. The geometry comparison is shown in Fig. 2.

The learned SDF can be then converted into a surface mesh using the Marching Cubes [19], which supports further tuning in the following inverse rendering pipeline. Different from Nvdiffrecmc which adopts DMTET for differentiable mesh optimization, we introduce Flexicubes [35] into our inverse rendering pipeline. Flexicubes use the SDF, weight, and the deformation for vertexes of the grid cells to extract the surface mesh using the DMC [29]. The mesh is converted into a differentiable representation that can be optimized based on gradients.

3.4 Training Objectives

The main objective is to minimize the photometric loss between rendered images and ground truth:

$$\mathcal{L}_{rgb} = ||C - C_{gt}||^2, \tag{11}$$

where C is the rendering result, and C_{gt} is the corresponding ground truth. Following previous work [56, 28, 9], we adopt a smoothness loss for diffuse and material as regularization:

$$\mathcal{L}_d = \sum_{\mathbf{x}_{\text{surf}}} |\boldsymbol{k}_d\left(\mathbf{x}_{\text{surf}}\right) - \boldsymbol{k}_d\left(\mathbf{x}_{\text{surf}} + \epsilon\right)|, \tag{12}$$

$$\mathcal{L}_{orm} = \sum_{\mathbf{x}_{\text{surf}}} |\boldsymbol{k}_{orm}\left(\mathbf{x}_{\text{surf}}\right) - \boldsymbol{k}_{orm}\left(\mathbf{x}_{\text{surf}} + \epsilon\right)|, \tag{13}$$

where $\mathbf{x}_{\text{surf}}$ represents the world coordinates of points on the object's surface. $\boldsymbol{k}_d(\mathbf{x}_{\text{surf}})$ and $\boldsymbol{k}_{orm}(\mathbf{x}_{\text{surf}})$ is the material of this point. ϵ is a randomly distributed vector of tiny deformations. A self-supervised loss is used to regulate the learned diffuse color:

$$\mathcal{L}_{\text{diff}} = \mathcal{L}_{rgb}\left(\mathbf{C}_{\text{diff}}, \boldsymbol{k}_{\text{diff}}\left(\boldsymbol{x}_{\text{surf}}\right)\right), \tag{14}$$

where $\mathbf{C}_{\text{diff}}$ is the diffuse light from the object surface obtained during rendering. $\boldsymbol{k}_{\text{diff}}(\mathbf{x}_{\text{surf}})$ is the diffuse light obtained by the MLP. Overall, the full loss function is:

$$\mathcal{L} = \mathcal{L}_{rgb} + \omega_1 \mathcal{L}_d + \omega_2 \mathcal{L}_{orm} + \omega_3 \mathcal{L}_{\text{diff}}, \tag{15}$$

where ω_1, ω_2 and ω_3 are three predefined scalars.

4 Experiments

4.1 Implementation Details

Dataset. We construct a dataset of multiple reflective objects based on the existing single objects to evaluate the performance. Our dataset consists of 16 groups of object compositions, most of which contain indirect illumination between reflective objects. We render the composed objects with various environmental lighting in the Blender engine. Each group contains 300 images, with 200 for the training set and 100 for the test set.

Experiment setup. We optimize the 3D model on 1 RTX 3090 GPU with 24G memory. We use the Adam optimizer for the material and the environment map with an initial learning rate of 0.03. The coefficients of loss function ω_1, ω_2, and ω_3 are set to 0.1, 0.05, and 1, respectively. The rate of Monte Carlo sampling is commonly set to 128 consistent with the setting in Nvdiffrecmc [9].

4.2 Comparison with Baseline

In this section, we compare our method with Nvdiffrec [28], Nvdiffrecmc [9], and Nefii [45] on our constructed reflective dataset, and the results are shown in Fig. 3. Nvdiffrec achieves photo-realistic results in the *metal balls* but fails on the smooth and glossy *table + horse*. Besides, Nvdiffrec cannot
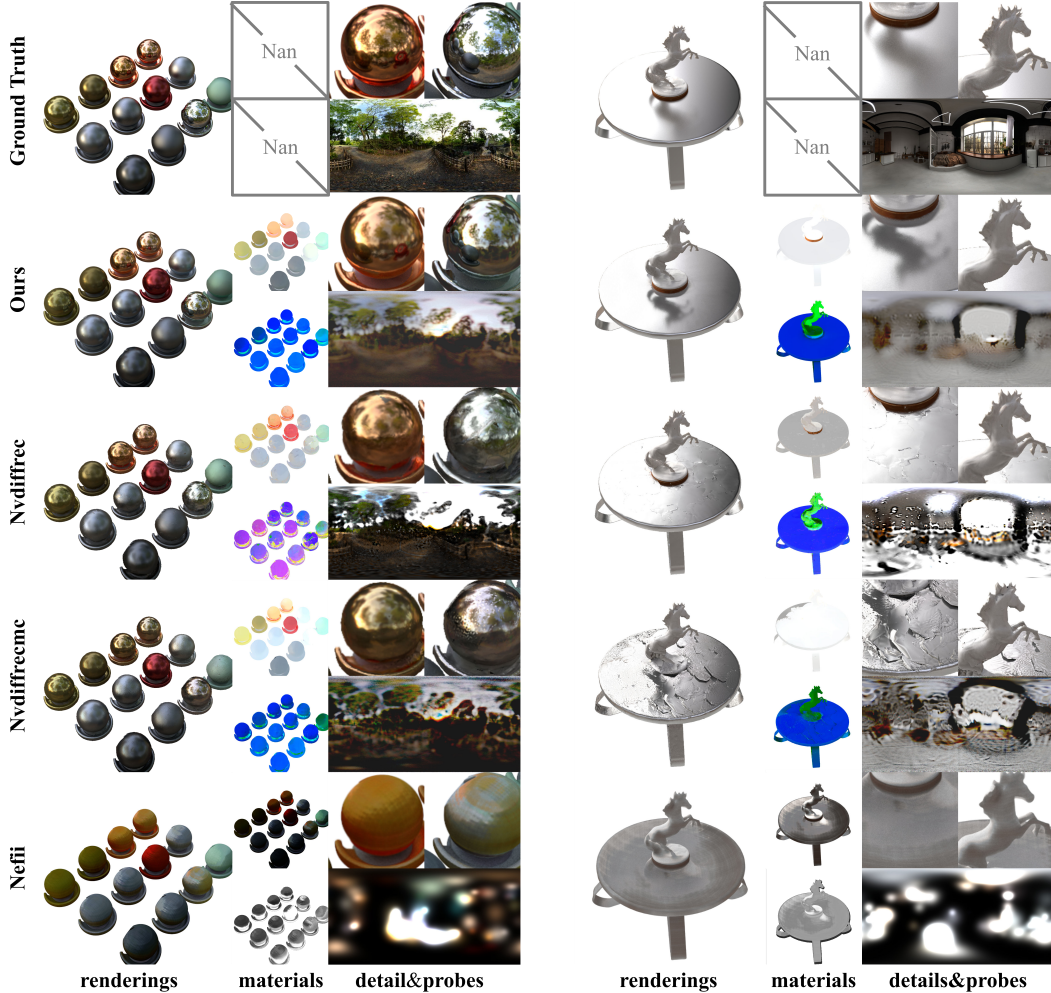
Figure 3: **Qualitative comparison**. The results of renderings, materials, and environment maps are presented. Note that, the material of Nefii contains only roughness without metalness. Our method achieves the best renderings with clear reflections, compared to other inverse rendering methods. Our method is also superior to others in the disentanglement of materials and environment maps.

Table 1: **Quantitative Comparisons**. Ours (x) means the x times of sampling in our method. Ours (1) is about 1.5h longer than Nvdiffrecmc [9] as we need additional time to learn the initial geometry.

Method	NDR [28]	NDRMC [9]	Nefii [45]	**Ours**	Ours (w/o Acc.)	Ours (w/o Geo.)	Ours (3)	Ours (1)
PSNR↑	26.9	25.7	22.3	28.1	28.0	24.6	28.4	27.3
Training time↓	30min	45min	20h	6.5h	11.5h	5h	26.5h	2.25h

well disentangle the materials because it does not consider the shading. In contrast, Nvdiffrecmc performs well in material learning, while its rendering results are bad. These two methods also suffer from indirect illumination from the reflection of inner objects, leading to low-quality environment maps. Nefii is the recent work that considers the indirect illumination in the radiance field, but it tends to produce low-reflective results and performs badly in these high specular objects. Compared to these methods, our method can achieve both photo-realistic rendering results and well-disentangled material learning. We can handle highly specular objects, *e.g.*, the metal table. The environment maps learned by our methods are also superior to others.
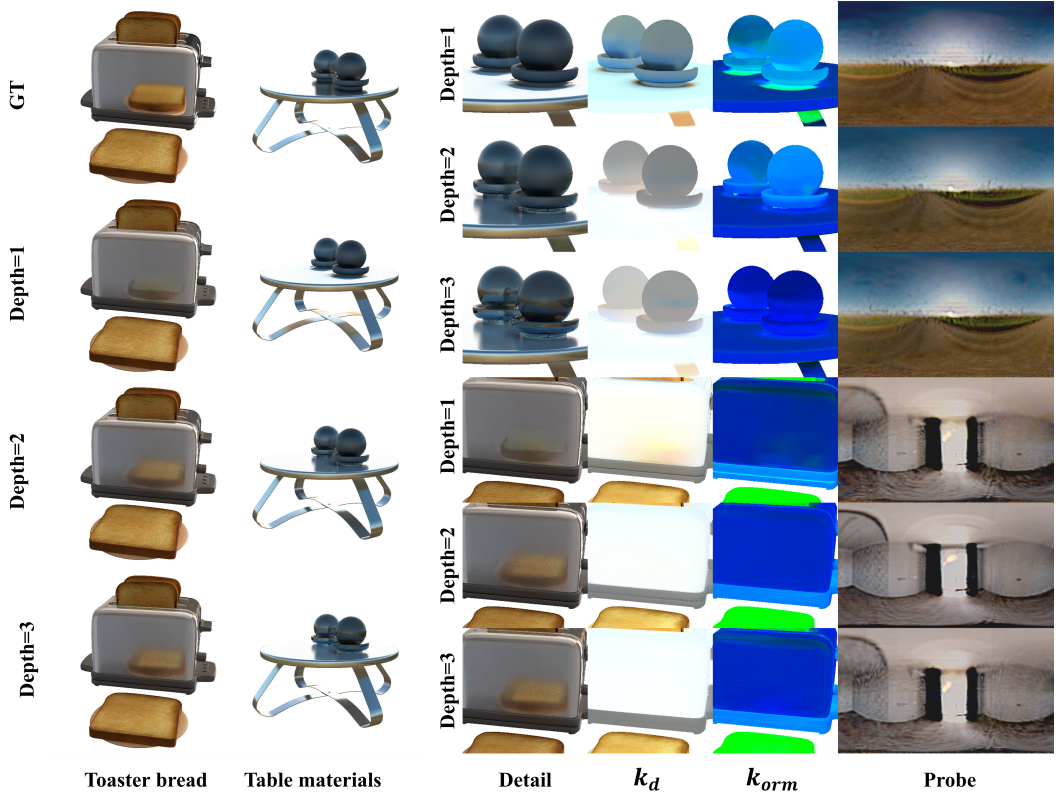
7

Figure 4: **Ablation study** on the depth of ray tracing, *i.e.*, the times of Monte Carlo sampling. The results with depth=1 show fewer and darker reflections compared to the ground truth. k_d maps also illustrate the limited capacity to disentangle the material from environmental light, for example, mistaking the diffuse color of the table as the color of the sky. In contrast, with depth=2 or 3, results show more realistic renderings and disentangled materials.
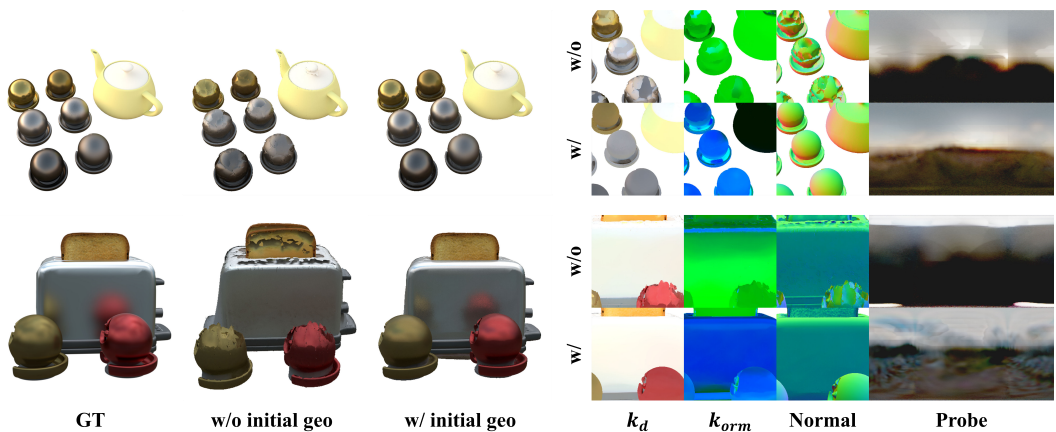

Figure 5: **Ablation study** on geometric initialization. As shown, a better-quality geometry can significantly improve the material learning and also refine the rendering results.

4.3 Multi-times Monte Carlo Sampling

In this section, we conduct an ablation study on the times of Monte Carlo Sampling, *i.e.* the depth of ray tracing. Multi-times sampling considers the indirect illumination, and thus successfully disentangles the environment light and inner reflective light. As shown in Fig. 4, the environment

maps learned by single-time sampling contain noise and shades that are actually the inner reflection. In contrast, multi-times sampling significantly releases the problem, producing a clearer environment map. Single-time sampling also undermines the rendering results, with darker inverted reflections. When the sampling ray is sheltered by objects during path tracing, the ray returns no light, resulting in a dark point. Multi-times sampling returns the reflective light of the sheltering points containing both diffuse and specular light. As a consequence, the rendering results of multi-times sampling show realistic appearance and reflections. Besides, it can be seen that results with 2-time sampling are comparable to results with 3-time sampling. Tab. 1 also quantitatively shows minimal improvement by deeper tracing. However, each additional sampling time induces a 4-times computation increment. Therefore, a 2-times sampling is a more cost-effective setting. Additionally, we also compare the efficiency between the sampling with and without acceleration. As shown in Tab. 1, our method can make the training speed twice as fast while keeping a comparable PSNR. The degree of acceleration is related to the complexity of geometry, and here we adopt the median in our dataset.

4.4 Geometry

In this section, we further explore the necessity of the initial geometry discussed in Sec. 3.3 and present our geometry reconstruction results. As shown in Fig. 5, without initial geometry, the rendering results show bumpy surfaces with badly learned materials in the area of hollow shape. Because multi-times sampling amplifies the errors introduced by geometry, the learned materials are even worse than the single-time sampling. When applying a good-quality initialization, the network bypasses the ambiguity brought by the geometry. It helps our method learn well-disentangled materials and finally produces high-quality rendering results. Quantitative comparison in Tab. 1 also demonstrate the necessity of geometry initialization. In addition, we showcase a real-scene geometry reconstruction result of the NeRO [18] dataset in Fig. 6. We further use Chamfer Distance to quantitatively evaluate our geometric quality and present the results in Tab. 2.
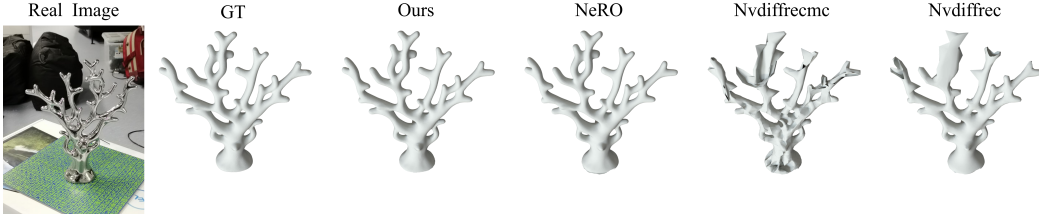

Figure 6: **Real scene geometry comparison**. We compare our reconstructed geometry with NeRO, Nvdiffrec and Nvdiffrecmc. Our reconstructed geometry demonstrates superior results.

Table 2: **Geometry qualitative comparison**. We use the Chamfer Distance ($\downarrow$) to evaluate our reconstructed geometry. As shown in the table, we obtain the best results in multiple scenes.

Dataset	Nvdiffrec [28]	Nvdiffrecmc [9]	NeRO [18]	Ours
Materials	0.016	0.016	0.0057	**0.0030**
Coral	0.28	0.25	0.13	**0.13**

4.5 Relighting and Material Editing

The disentangled environment map and material empower our methods to relight and edit reconstructed objects in the downstream application. Our method can also be seamlessly plugged into the industrial pipeline. As shown in Fig. 7, our method enables the arbitrary combination of reconstructed objects. We can easily edit their metalness, roughness, and albedo color by manipulating the learned materials. Due to the well-disentangled shading, our rendering results are natural and realistic in all five environment maps, even the point light in a dark environment. They also perform well after editing materials thanks to the well-learned material map. For example, after we increase the metalness and reduce the roughness of the teapot, the teapot clearly reflects the neighboring objects on its surface. In another case, where we extremely increase the reflectance and change the base color of the toaster, it accomplishes to reflect the scene and other objects. To showcase further applications
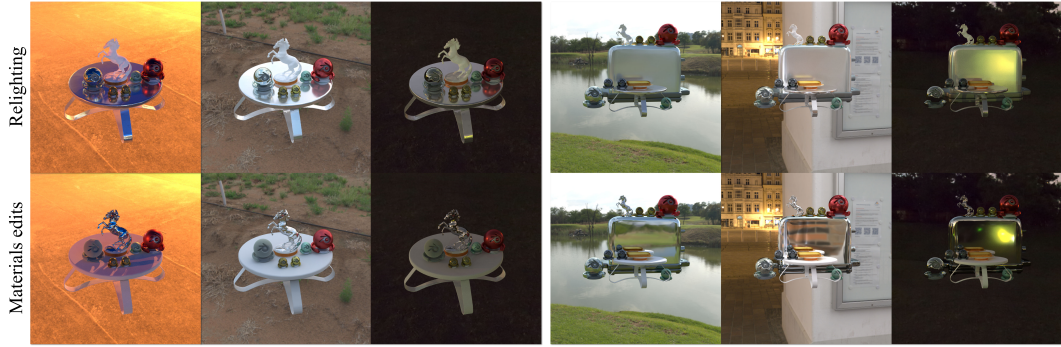
Figure 7: **Relighting and editing**. We compose the reconstructed objects in a unified scene and change the environmental light. The relighting results show our strong ability to disentangle the light and shading. We also perform material edits which shows the flexibility in wide applications.
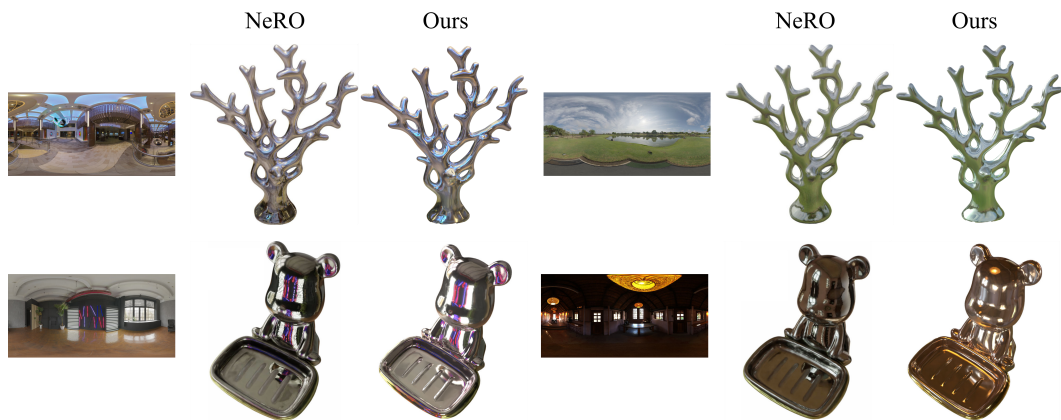

Figure 8: **Relighting results of real scenes**. Our method demonstrates more realistic results under several lighting conditions than NeRO.

of our method, we conduct experiments on real-world datasets captured by NeRO [18] and present the comparison results in Fig. 8. It can be seen that Ref-MC^2 achieves more realistic results under different lighting conditions.

5   Conclusion and Limitations

In conclusion, our Ref-MC^2 introduces multi-times Monte Carlo sampling into the inverse rendering pipeline to model the indirect illumination. It improves the performance in scenes with complex inter-reflections. However, increment of sampling times significantly increases computational consumption and makes the pipeline highly geometry-sensitive. To solve the challenge of computational efficiency, based on the Lambert model, we simplify the BRDF for indirect lighting, which allows us to reduce the number of ray traces. To improve the geometry quality, we adopt SDF-based architecture to get an initial geometry and refine a design with Spherical Gaussian encoding for reflective objects. We further use Flexicubes to take the initial mesh into the differentiable rendering pipeline that learns disentangled materials. Our Ref-MC^2 still has several limitations. The major limitation is that the 2-time sampling cannot handle extremely reflective objects, *e.g.*, mirrors, as the specular energy hardly degrades after reflections. Besides, the training time needs to further reduce in future time.

6   Acknowledgements

References

[1] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233, 2003.

[2] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P. A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *NeurIPS*, pages 10691–10704, 2021.

[3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, volume 13692, pages 333–350, 2022.

[4] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *ICCV*, pages 4159–4171, 2023.

[5] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

[6] Wenhang Ge, Tao Hu, Haoyu Zhao, Shu Liu, and Ying-Cong Chen. Ref-neus: Ambiguity-reduced neural implicit surface learning for multi-view reconstruction with reflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4251–4260, 2023.

[7] Yuan-Chen Guo, Yan-Pei Cao, Chen Wang, Yu He, Ying Shan, and Song-Hai Zhang. Vmesh: Hybrid volume-mesh representation for efficient view synthesis. In *SIGGRAPH Asia*, pages 17:1–17:11, 2023.

[8] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *CVPR*, pages 18388–18397, 2022.

[9] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light, and material decomposition from images using monte carlo rendering and denoising. *Advances in Neural Information Processing Systems*, 35:22856–22869, 2022.

[10] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, pages 19717–19726, 2023.

[11] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*, 2024.

[12] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023.

[13] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *CVPR*, pages 165–174, 2023.

[14] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

[15] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984.

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

[17] Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. ENVIDR: implicit differentiable renderer with neural environment lighting. In *ICCV*, pages 79–89, 2023.

[18] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Transactions on Graphics (TOG)*, 42(4):1–22, 2023.

[19] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

[20] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *CVPR*, 2024.

[21] Jipeng Lv, Heng Guo, Guanying Chen, Jinxiu Liang, and Boxin Shi. Non-lambertian multispectral photometric stereo via spectral reflectance decomposition. In *IJCAI*, 2023.

[22] Alexander Mai, Dor Verbin, Falko Kuester, and Sara Fridovich-Keil. Neural microfacet fields for inverse rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 408–418, 2023.

[23] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021.

[24] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*, pages 1–7. 2012.

[25] Stephen McAuley, Stephen Hill, Adam Martinez, Ryusuke Villemin, Matt Pettineo, Dimitar Lazarov, David Neubelt, Brian Karis, Christophe Hery, Naty Hoffman, et al. Physically based shading in theory and practice. In *ACM SIGGRAPH 2013 Courses*, pages 1–8. 2013.

[26] 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 *ECCV*, 2020.

[27] William J Morokoff and Russel E Caflisch. Quasi-monte carlo integration. *Journal of computational physics*, 122(2):218–230, 1995.

[28] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022.

[29] Gregory M Nielson. Dual marching cubes. In *IEEE visualization 2004*, pages 489–496. IEEE, 2004.

[30] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021.

[31] Michael Oren and Shree K Nayar. Generalization of lambert's reflectance model. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246, 1994.

[32] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pages 5845–5854, 2021.

[33] Christian Reiser, Stephan Garbin, Pratul Srinivasan, Dor Verbin, Richard Szeliski, Ben Mildenhall, Jonathan Barron, Peter Hedman, and Andreas Geiger. Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis. *ACM Transactions on Graphics (TOG)*, 43(4):1–14, 2024.

[34] Christian Reiser, Richard Szeliski, Dor Verbin, Pratul P. Srinivasan, Ben Mildenhall, Andreas Geiger, Jonathan T. Barron, and Peter Hedman. MERF: memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM Trans. Graph.*, 42(4):89:1–89:12, 2023.

[35] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.

[36] Zixi Shu, Ran Yi, Yuqi Meng, Yutong Wu, and Lizhuang Ma. Rt-octree: Accelerate plenoctree rendering with batched regular tracking and neural denoising for real-time neural radiance fields. In *SIGGRAPH Asia*, pages 99:1–99:11, 2023.

[37] Peter-Pike J. Sloan, Jan Kautz, and John M. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.*, 21(3):527–536, 2002.

[38] Jiajun Tang, Haofeng Zhong, Shuchen Weng, and Boxin Shi. Luminaire: Illumination-aware conditional image repainting for lighting-realistic generation. In *NeurIPS*, 2023.

[39] Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

[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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[41] Fangjinhua Wang, Marie-Julie Rakotosaona, Michael Niemeyer, Richard Szeliski, Marc Pollefeys, and Federico Tombari. Unisdf: Unifying neural representations for high-fidelity 3d reconstruction of complex scenes with reflections. *arxiv preprint arXiv:2312.13285*, 2023.

[42] Jiaping Wang, Peiran Ren, Minmin Gong, John M. Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph.*, 28(5):133, 2009.

[43] 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. *arXiv preprint arXiv:2106.10689*, 2021.

[44] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[45] Haoqian Wu, Zhipeng Hu, Lincheng Li, Yongqiang Zhang, Changjie Fan, and Xin Yu. Nefii: Inverse rendering for reflectance decomposition with near-field indirect illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2023.

[46] Xiuchao Wu, Jiamin Xu, Zihan Zhu, Hujun Bao, Qixing Huang, James Tompkin, and Weiwei Xu. Scalable neural indoor scene rendering. *ACM Trans. Graph.*, 41(4):98:1–98:16, 2022.

[47] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Trans. Graph.*, 32(6):209:1–209:11, 2013.

[48] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

[49] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In *SIGGRAPH*, pages 46:1–46:9, 2023.

[50] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.

[51] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

[52] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, pages 5732–5741, 2021.

[53] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. *CVPR*, 2024.

[54] 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 *CVPR*, pages 5453–5462, 2021.

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

[56] 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 Transactions on Graphics (ToG)*, 40(6):1–18, 2021.

[57] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, pages 18622–18631, 2022.

[58] Youjia Zhang, Teng Xu, Junqing Yu, Yuteng Ye, Yanqing Jing, Junle Wang, Jingyi Yu, and Wei Yang. Nemf: Inverse volume rendering with neural microflake field. In *ICCV*, pages 22862–22872, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction accurately reflect the contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [No]

Justification: The data and code have not been provided to reviewers.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the data and code in a few weeks.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the implement details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Results are accompanied by error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report compute resources in the experiment set up.

Guidelines:
- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The research conforms with that.

Guidelines:
- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss them in the introduction and experiments.

Guidelines:
- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: NA.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited them.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have discussed them in section experiment.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: NA.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: NA.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.