Joint Optimization of Triangle Mesh, Material, and Light from Neural Fields with Neural Radiance Cache

JIAKAI SUN, Zhejiang University, China

ZHANJIE ZHANG, Zhejiang University, China

GUANGYUAN LI, Zhejiang University, China

TIANYI CHU, Zhejiang University, China

LEI ZHAO, Zhejiang University, China
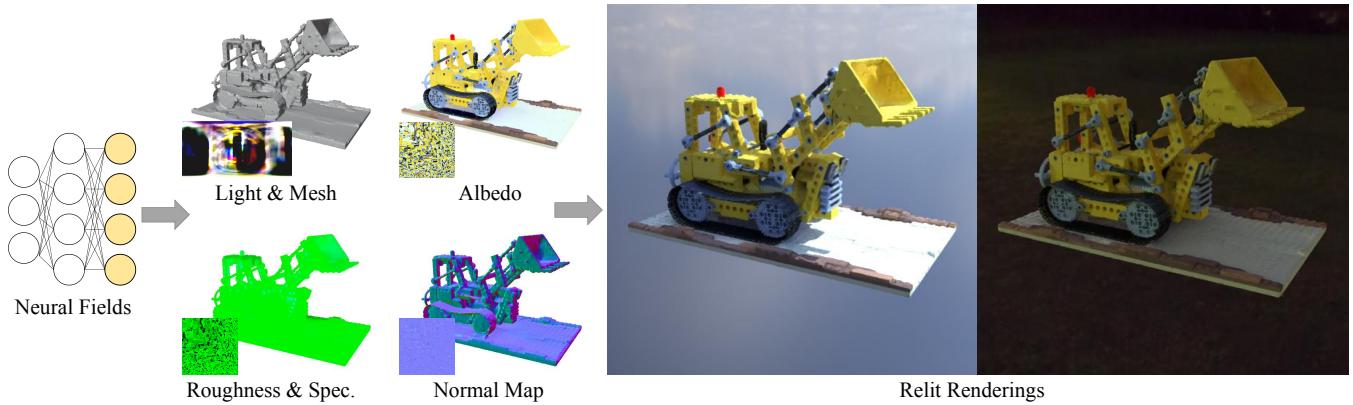
WEI XING, Zhejiang University, China

Fig. 1. Given the neural field representing a scene, our framework aims to jointly optimize triangle mesh, PBR material, and HDR light probe, which is directly compatible with modern graphics pipelines, facilitating integration with digital content creation tools and game engines. The Lego model is from NeRF synthetic dataset (CC-BY-3.0), and the relit renderings are rendered with Blender Cycles and Light probes from Poly Haven (CC0).

Traditional inverse rendering techniques are based on textured meshes, which naturally adapts to modern graphics pipelines, but costly differentiable multi-bounce Monte Carlo (MC) ray tracing poses challenges for modeling global illumination. Recently, neural fields have demonstrated impressive reconstruction quality but fall short in modeling indirect illumination. In this paper, we introduce a simple yet efficient inverse rendering framework that combines the strengths of both methods. Specifically, given pre-trained neural field representing the scene, we can obtain an initial estimate of the signed distance field (SDF) and create a Neural Radiance Cache (NRC), an enhancement over the traditional radiance cache used in real-time rendering. By using the former to initialize differentiable marching tetrahedrons (DMTet) and the latter to model indirect illumination, we can compute the global illumination via single-bounce differentiable MC ray tracing and jointly optimize the geometry, material, and light through back-propagation. Experiments demonstrate that, compared to previous methods, our approach effectively prevents indirect illumination effects from being baked into materials, thus obtaining the high-quality reconstruction of triangle mesh, Physically-Based (PBR) material, and High Dynamic Range (HDR) light probe.

CCS Concepts: • **Computing methodologies → Rendering**; **Computer vision**.

Additional Key Words and Phrases: Inverse Rendering, MC Ray Tracing, Neural Rendering, Neural Fields, Radiance Caching

1 INTRODUCTION

Coordinate-based neural networks, also known as neural fields [Xie et al. 2022], are gaining increasing popularity in the field of computer vision and graphics. Neural Radiance Fields (NeRF) [Mildenhall et al. 2020] and its variants [Barron et al. 2021, 2022; Chen et al. 2022; Karnewar et al. 2022; Müller et al. 2022; Sun et al. 2022b; Verbin et al. 2022] represent radiance fields as neural fields, leveraging differentiable volume rendering to synthesize photo-realistic novel views. In addition, surface-based methods [Azinović et al. 2022; Niemeyer et al. 2020; Oechsle et al. 2021; Wang et al. 2021; Yariv et al. 2021, 2020] model Signed Distance Fields (SDF) using neural fields, enabling accurate geometric reconstruction. These neural field-based method have significantly advanced the development of inverse rendering—a longstanding task in computer vision and graphics to estimate physical attributes of a scene from images. Recent works [Boss et al. 2021a,b; Zhang et al. 2022a, 2021a,b, 2022b] utilizing neural fields for geometry, material, and/or light decomposition has yielded impressive results. However, as these methods are not physically-based, they can only recover basic illumination effects and tend to bake complex effects such as shadows, inter-reflections, and color bleeding into materials. Furthermore, neural fields are not directly compatible with modern graphics pipelines, which poses limitations on its practical application.

In contrast, textured meshes naturally fit into modern graphics pipelines and can be optimized via physically-based differentiable rendering [Li et al. 2018; Loubet et al. 2019; Vicini et al. 2021; Zhang et al. 2020], enabling an unbiased estimation of attributes of a scene. Nonetheless, multi-bounce differentiable Monte Carlo (MC) rendering poses a steep increase in computational time and memory requirements for gradient-based optimization, particularly when aiming to jointly optimize geometry, material, and light. Consequently, previous state-of-the-art methods for joint reconstruction [Hasselgren et al. 2022; Munkberg et al. 2022] have primarily focused on direct illumination, i.e., environment lighting with shadows. Although these approaches yield convincing results, they fall short in decoupling material reflectance from indirect illumination effects.

To harness the reconstruction capabilities of neural fields and the efficient performance of modern graphics pipelines, we propose JOC[1] (pronounced as "joke"), a framework that integrates cutting-edge techniques from computer vision and graphics. JOC allows for capturing global illumination using single-bounce differentiable MC rendering with a modest increase in cost, and jointly optimizes geometry, material, and light via back-propagation, as shown in Figure 1. Specifically, we first obtain a Neural Radiance Cache (NRC) [Müller et al. 2021]—a low-cost and highly generalizable version of traditional radiance cache—from a trained neural field. The trained neural field offers an additional advantage: a robust initial estimate of the SDF for the Differentiable Marching Tetrahedrons (DMTet) [Shen et al. 2021], preventing mismatches between the triangle mesh obtained from DMTet and the NRC. By integrating the NRC into a highly optimized ray tracing pipeline [Parker et al. 2010], we can jointly optimize triangle meshes, Physically-Based (PBR) materials, and High Dynamic Range (HDR) light probe. Compared to previous inverse rendering methods based on textured meshes, JOC separates indirect illumination effects and material, yielding high quality reconstruction. Notably, we make no assumptions about the neural fields, other than it being capable of synthesizing novel views and estimating depth via ray casting[2]. This allows us to train the NRC from the majority of neural field-based methods, facilitating seamless integration with the latest advances in the area of neural fields.

2   RELATED WORK

Neural Fields. In recent years, neural field-based methods demonstrate exceptional capabilities in 3D reconstruction tasks, attracting widespread attention from researchers in computer vision and graphics. By parameterizing radiance fields using neural fields and utilizing differentiable volume rendering, NeRF [Mildenhall et al. 2020] is able to synthesize photorealistic novel views. This remarkable performance inspires a series of subsequent works that enhance various aspects of the method, including reducing training and/or rendering costs [Chen et al. 2022; Fridovich-Keil and Yu et al. 2022; Karnewar et al. 2022; Müller et al. 2022; Sun et al. 2022b; Takikawa et al. 2022], and improving performance in handling sparse inputs [Jain et al.

2021; Niemeyer et al. 2022; Sun et al. 2023; Yu et al. 2021] and challenging scenarios [Barron et al. 2021, 2022; Martin-Brualla et al. 2021; Mildenhall et al. 2022]. These variants spur the community to create some sophisticated models like Nerfacto [Tancik et al. 2023], which integrates a variety of advanced techniques. Some neural implicit surface reconstruction methods [Oechsle et al. 2021; Wang et al. 2021; Yariv et al. 2021], exemplified by NeuS [Wang et al. 2021], modify the volume rendering formula used by NeRF and use neural fields to parameterize SDF, achieving high-quality surface reconstructions. This approach further inspires works that enhance geometric consistency [Fu et al. 2022], outdoor scene performance [Sun et al. 2022a], and robustness in sparse settings [Yu et al. 2022b]. Inspired by Nerfacto, the community develops NeuS-facto [Yu et al. 2022a], a high-performance NeuS-like model that excels in various aspects. Despite the powerful reconstruction capabilities of neural field-based methods, they are not directly applicable to modern graphics pipelines, which limits their practical use.

Inverse Rendering. Inverse rendering is a longstanding challenge in computer vision and graphics. Most early works [Dong et al. 2014; Gardner et al. 2003; Ghosh et al. 2009; Guarnera et al. 2016; Xia et al. 2016] involve stringent conditions for scene capture, such as the need for specific lighting or complex camera setups. These extra settings provide sufficient priors. However, the joint recovery of geometry, material, and light from casually captured image sets under unknown illumination remains a severely ill-posed problem.

Recently, a series of works [Boss et al. 2021a,b; Zhang et al. 2021a,b, 2022b] have leveraged the powerful reconstruction capabilities of neural fields to tackle this issue. NeRFactor [Zhang et al. 2021b], NeRD [Boss et al. 2021a], and Neural-PIL [Boss et al. 2021b] employ NeRF-like geometry representations, simplified lighting models, and custom rendering pipelines to achieve decomposition of shape and reflectance under unknown Illumination, allowing for view synthesis and relighting. Notably, among these methods, NeRFactor directly utilizes pre-trained NeRF, while NeRD and Neural-PIL require their own neural field training to meet model requirements. PhySG [Zhang et al. 2021a], by utilizing neural fields to parameterize SDF and diffuse albedo, and using spherical Gaussians (SG) to represent environment map and specular bidirectional reflectance distribution function (BRDF), achieves material editing and re-lighting under the assumption of non-spatially-varying reflectance. InvRender [Zhang et al. 2022b] trains a visibility multi-layer perceptron (MLP) and an indirect illumination MLP leveraging PhySG. With these MLPs and its custom rendering pipeline, it achieves inverse rendering that accounts for global illumination.

While neural field-based inverse rendering methods exhibit impressive performance, they require additional processing to be compatible with modern digital content creation (DCC) tools and game engines, which might sacrifices reconstruction quality. In contrast, when taking textured meshes to represent the scene, these issues do not arise. NVDIFFREC [Munkberg et al. 2022] utilizes DMTet [Shen et al. 2021] and differentiable rasterizer with deferred shading to jointly optimize mesh, material, and lighting from images. Yet, due to its simplified shading model, shadows are baked into the material. NVDIFFRECMC [Hasselgren et al. 2022] extends NVDIFFREC with

---

[1]**J**oint **O**ptimization of Triangle Mesh, Material, and Light from Neural Fields with Neural Radiance **C**ache
[2]This assumption will be implicit in all subsequent sections for simplicity unless otherwise stated.

a differentiable MC renderer and a differentiable denoiser, effectively separating shadow and material. However, considering the increased noise level and drastically increased iteration time caused by multi-bounce path tracing, NVDIFFRECMC only considers direct lighting, leading to coupling of indirect illumination effects with the material.

Radiance Caching. Caching of radiometric quantities, a technique now prevalent in modern game engines for real-time global illumination, owes its inception to the seminal work of [Ward et al. 1988] on diffuse interreflection. Later works [Greger et al. 1998; Krivanek et al. 2005; Ren et al. 2013; Scherzer et al. 2012; Silvennoinen and Lehtinen 2017] enhance the performance and/or generalization of caching algorithms. However, achieving both of these objectives often requires hand-crafted priors and sophisticated data structure.

With the advent of neural rendering, a line of works [Hadadan et al. 2021; Müller et al. 2020, 2021; Ren et al. 2013] represent the radiance cache with neural networks to circumvent these requirements. Among these methods, NRC [Müller et al. 2021] stands out by employing a fully fused MLP for online training, thereby achieving robust real-time radiance caching with predictable performance and resource consumption. JOC utilizes a streamlined version of NRC, designed to be compatible with most off-the-shelf neural fields techniques.

Concurrent works. Several concurrent works [Hadadan et al. 2023; Wang et al. 2023] also combine neural fields and explicit representations. We introduce these works and highlight the differences between them and JOC as follows:

- FEGR [Wang et al. 2023] represents the intrinsic properties of the scene using a neural field and estimate the G-buffer with volumetric rendering. However, it cannot model indirect illumination.
- InvGI [Hadadan et al. 2023] leverages a neural field both to represent the radiance function, and to account for global illumination without building path integrals. However, InvGI does not allow for joint optimization of geometry, material, and light.

3 BACKGROUND

As the main idea of JOC involves training NRC from existing Neural Fields, this section will focus on the introduction of Neural Fields in 3D Reconstruction and Neural Radiance Cache.

3.1 Neural Fields in 3D Reconstruction

Refer to [Xie et al. 2022], we define a Neural Field as a field that is parameterized fully or in part by a neural network which maps position (and optionally, some additional inputs) to corresponding attributes. For instance, NeRF [Mildenhall et al. 2020] maps position and viewing direction to density and color, while NeuS [Wang et al. 2021] maps these to value of the signed distance function and color. Recent works have demonstrated that neural fields are a powerful scene representation for 3D reconstruction from a set of posed camera images. Despite the diverse attributes computed by neural fields, the majority of these methods render neural fields to corresponding pixels by casting camera rays, and optimize the

parameters of the neural fields through gradient-based optimization to realize 3D reconstruction. To make this process more formal, we start from the measurement equation [Veach 1998]

$$I = \int_{\mathcal{A} \times \Omega} W_e(x, \omega) L_i(x, \omega) dx d\omega^\perp, \tag{1}$$

where I represents real-valued measurement. Each measurement corresponds to the integral of the outputs of a sensor that responds to the incident radiance L_i upon it with the sensor responsivity W_e over its area $\mathcal{A}$ and the upper hemisphere Ω. In the context of rendering images, each measurement I represents the value of a single pixel, while W_e is simplified to tone mapping and gamma correction. Methods based on neural fields also use the measurement equation to calculate pixel values, but they have different ways to compute L_i. For instance, IDR [Yariv et al. 2020] uses surface-based rendering, while NeRF uses volume rendering. However, they both compute L_i by casting rays and utilizing the attributes computed by the neural fields. Additionally, most methods based on neural fields can be used to estimate depth, i.e., the distance between the origin of the ray and its intersection point with the scene surface. For example, the depth can be estimated in NeRF-like model by a similar procedure as rendering RGB, while SDF-based model can directly use sphere tracing to find intersection points.

3.2 Neural Radiance Cache

In light transport calculations [Veach 1998], the incident radiance L_i in Equation (1) can be obtained using the exitant radiance L_o:

$$L_i(x, \omega) = L_o(x_\mathcal{M}(x, \omega), -\omega), \tag{2}$$

where $x_\mathcal{M}(x, \omega)$ is the ray-casting function, which returns the first point of the scene $\mathcal{M}$ visible from x in direction ω. Further, we can computes the exitant radiance L_o using the rendering equation [Kajiya 1986] :

$$L_o(x, \omega) = E(x, \omega) + \int_\Omega f(x, \omega_i, \omega) L_i(x, \omega_i) d\omega_i^\perp. \tag{3}$$

Here, w is the direction of L_i, and $f(x, \omega_i, \omega)$ is the BRDF. As $E(x, \omega)$ represents the emitted radiance L_e, NRC focuses on the remaining part of the equation, which is used to compute the scattered radiance L_s, given that it involves a challenging high-dimensional integral. To address this issue, NRC approximates L_s with a fully fused MLP expressed as:

$$NRC_\phi : (x, \omega, n, r, \alpha, \beta) \mapsto L_s, \tag{4}$$

caching the the scattered radiance $L_s \in \mathbb{R}^3$ at position $x \in \mathbb{R}^3$ and scattered direction $\omega \in \mathbb{S}^2$ with surface normal $n \in \mathbb{S}^2$, surface roughness $r \in \mathbb{R}$, diffuse reflectance $\alpha \in \mathbb{R}^3$ and specular reflectance $\beta \in \mathbb{R}^3$ as additional parameters. Here we omit the encoding for each parameter for simplicity, please refer to [Müller et al. 2021] for more details. Despite employing a minimal set of extra parameters to enhance its robustness, original NRC still requires material parameters r, α, and β, which are often inaccessible in many neural fields. Hence, we rely solely on x, ω, and n as input parameters, still yielding satisfactory approximations of indirect illumination.
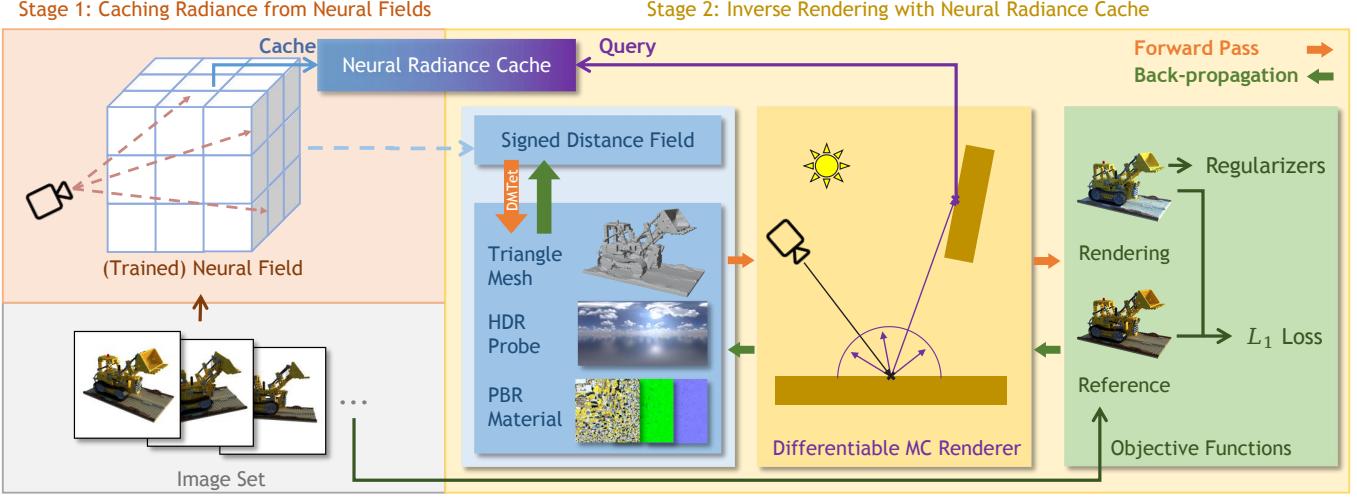
Fig. 2. Overview of JOC. Given a set of posed multi-view images and a corresponding neural field trained on it, JOC aims to reconstruct triangle mesh, PBR material and HDR light probe, which can be directly integrated into modern digital content creation tools and game engines. More specifically, JOC comprises two stages: In the first stage (Section 4.1), the Neural Radiance Cache (NRC) [Müller et al. 2021] is trained using the neural field, thereby 'caching' the radiance. We then jointly optimize the geometry, material, and light utilizing the Differentiable Marching Tetrahedrons (DMTet) [Shen et al. 2021] and a **single-bounce** differentiable Monte Carlo (MC) renderer combined with NRC in the second stage (Section 4.2). Note that we optionally leverage the neural field for a better initial estimate of the Signed Distant Field (SDF) for DMTet.
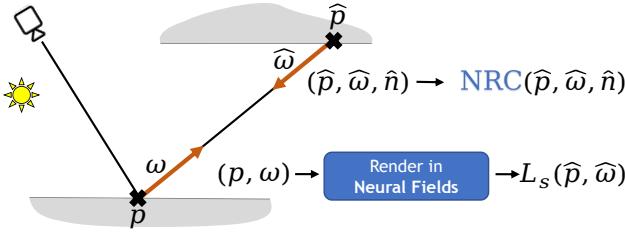

Fig. 3. Visualization of caching radiance. As neural fields are able to render RGB images and estimate depth, we can utilize them to capture scattered radiance at surface points, subsequently 'cache' it into the NRC.

4 METHOD

Given a set of multi-view images with associated foreground segmentation masks, camera poses, and a pre-trained neural field, JOC aims to reconstruct triangle meshes, material, and light directly suitable for modern DCC tools and game engines.

As illustrated in Figure 2, our framework consists of two stages: The first stage (Section 4.1) trains the NRC using the existing neural field. As NRC was designed for online training in real-time rendering, the traning time of this stage primarily depends on the rendering speed of the neural field. The second stage (Section 4.2) jointly optimizes the triangle mesh, material, and light through DMTet and single-bounce differentiable MC rendering combined with NRC.

4.1 Caching Radiance from Neural Fields

As introduced in Section 3.2, NRC is a MLP that takes position, scattered direction, and surface normal as inputs and outputs the exitant

radiance of position x in scattered direction ω. Following instant-ngp [Müller et al. 2022] , we use a multi-resolution hash grid [Müller et al. 2022] to encode positions and one-blob encoding [Müller et al. 2019] for the remaining input parameters.

Since the neural field can compute RGB value and estimate depth by casting rays, we can treat it as a function that

$$F_\theta : (o, \omega) \mapsto (L_i, p). \tag{5}$$

Here, $o \in \mathbb{R}^3$ is the ray origin, $\omega \in \mathbb{S}^2$ is the ray direction, $L_i \in \mathbb{R}^3$ is the incident radiance at (o, ω), and $p = x_\mathcal{M}(o, \omega) \in \mathbb{R}^3$ is the intersection point of the ray and the scene $\mathcal{M}$. According to Equation (2), the incident radiance L_i at (o, ω) is equivalent to the exitant radiance L_o at $(p, -\omega)$. By replacing o in Equation (5) with surface points p, we can obtain the exitant radiance L_o from $-\omega$, which is formulated as:

$$L_o(\hat{p}, -\omega), \hat{p} = F_\theta(p, \omega), \tag{6}$$

where $\hat{p} = x_\mathcal{M}(p, \omega)$. Assuming non-emitted objects, we get scattered radiance $L_s = L_o$ from surfaces from Equation (6), which can be used to train NRC. To be more specific, following InvRender [Zhang et al. 2022b], we randomly sample camera rays from a pre-trained neural field F_θ, estimate the depth to get surface points p, and then sample rays for each surface point. If these rays intersect with the surface, we record the intersection position $\hat{p}$, scattered direction $\hat{\omega} = -\omega$, and compute the analytic surface normal $\hat{n}$ of the intersection point for each intersection; see Figure 3 for the visualized process. Finally, we use the luminance-relative $\mathcal{L}_2$ loss [Müller

et al. 2021] to train the NRC, which is formulated as:

$$\mathcal{L}_2\left(L_s(\hat{p},\hat{\omega}), NRC_\phi(\hat{p},\hat{\omega},\hat{n})\right) = \frac{\left(L_s(\hat{p},\hat{\omega}) - NRC_\phi(\hat{p},\hat{\omega},\hat{n})\right)^2}{(NRC_\phi(\hat{p},\hat{\omega},\hat{n}))^2 + 0.01}, \quad (7)$$

where the gradient of the normalized factor is not back-propagated in optimization.

4.2 Inverse Rendering with Neural Radiance Cache

As a proof of concept, we extend NVDIFFRECMC [Hasselgren et al. 2022], which achieves joint optimization of triangle mesh, material, and light by employing a differentiable MC renderer that only consider direct illumination and a differentiable denoiser. In contrast, JOC enhance the renderer using NRC to account for global illumination, achieving superior reconstruction quality compared to NVDIFFRECMC.

4.2.1 Representations of Scene Parameters.

Geometry. We use SDF to represent geometry and leverage DMTet to convert it into a triangle mesh. SDF-based neural fields naturally provide an initial value for DMTet, while recent work [Lin et al. 2023] converts the density field to an SDF by subtracting a non-zero constant, allowing NeRF-like neural fields to also provide initial guess for the SDF. It should be noted that we can optimize geometry from scratch without relying on any initial estimate.

Material. We represent materials using Disney's PBR BRDF model and normal maps. These are widely used in modern game engines to enhance realism and scene details in physically-based shading models.

Light. We use an HDR light probe to represent illumination. This allows us to directly relight the reconstructed objects in modern renderers.

4.2.2 Global Illumination with a Single Bounce.
In the realm of physically-based rendering, the integral in the rendering equation are most often estimated using Monte Carlo sampling and ray tracing, as expressed by the following equation:

$$\begin{aligned} L_s(x,\omega) &= \int_\Omega f(x,\omega_i,\omega) L_i(x,\omega_i) d\omega_i^\perp \\ &\approx \frac{1}{N} \sum_{j=1}^N \frac{f(x,\omega_j,\omega) L_i(x,\omega_j)(\omega_j \cdot n)}{p(\omega_j)}, \end{aligned} \quad (8)$$

with N samples drawn from some distribution $p(\omega_j)$. Moreover, Multiple Importance Sampling (MIS) [Veach and Guibas 1995] is adopted, which is a prevalent sampling technique to reduce variance in Monte Carlo integration and is formulated as:

$$\sum_{i=1}^n \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{g(X_{i,j})}{p_i(X_{i,j})}, \quad w_i(x) = \frac{n_i p_i(x)}{\sum_k n_k p_k(x)}. \quad (9)$$

Here, p_i are differnet sampling distributions , $g(x)$ is the integrand and w_i are the weights. Please refer to [Hasselgren et al. 2022] for further details about the sampling strategy. In NVDIFFRECMC, as multi-bounce ray tracing comes with increased cost and noise, rays

only bounce once, that is, only shadow rays are traced. In this case, computing L_i simplifies to

$$L_i(p,\omega) = \begin{cases} \mathbf{E}(p,\omega) & \text{if } \omega \text{ is not blocked by } \mathcal{M} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where $\mathbf{E}(p,\omega)$ is the direct lighting estimation from the HDR light probe. However, if ω is blocked, $L_i(p,\omega)$ should represent the scattered radiance at the intersection point on the surface of $\mathcal{M}$. This shading model is clearly biased, resulting in indirect illumination being erroneously baked into the material. Nevertheless, the NRC serves as an effective solution for modeling indirect illumination. By modifying Equation (10) as:

$$L_i(p,\omega) = \begin{cases} \mathbf{E}(p,\omega) & \text{if } \omega \text{ is not blocked by } \mathcal{M} \\ NRC_\phi(\hat{p},-\omega,\hat{n}) & \text{otherwise} \end{cases} \quad (11)$$

where $\hat{p}$ is the intersection point and $\hat{n}$ is the surface normal. we are able to calculate global illumination via a single bounce, with an acceptable increase in computational cost.

4.2.3 Optimization.
Following [Hasselgren et al. 2022], we jointly optimize the geometry, material and light by minimizing the loss:

$$\mathcal{L} = \mathcal{L}_{\text{image}} + \lambda_{\mathbf{k}_d}\mathcal{L}_{\mathbf{k}_d} + \lambda_{\mathbf{k}_{\text{orm}}}\mathcal{L}_{\mathbf{k}_{\text{orm}}} + \lambda_n \mathcal{L}_n + \lambda_{n'}\mathcal{L}_{n'} + \lambda_{\text{light}}\mathcal{L}_{\text{light}}, \quad (12)$$

where $\mathcal{L}_{\text{image}}$ is simply the $\mathcal{L}_1$ loss between the tone mapped rendering and the input image $\mathbf{I}_{\text{GT}}$, and λs are corresponding weights for each regularizers.

$\mathcal{L}_{\mathbf{k}_d}$, $\mathcal{L}_{\mathbf{k}_{\text{orm}}}$ and $\mathcal{L}_n$ are the smoothness regularizers for albedo $\mathbf{k}_d$, specular parameters $\mathbf{k}_{\text{orm}}$ and surface normal n with the template as:

$$\mathcal{L}_P = \frac{1}{|\mathbf{x}_{\text{surf}}|} \sum_{x \in \mathbf{x}_{\text{surf}}} |P(x) - P(x+\epsilon)|, \quad (13)$$

where P is $\mathbf{k}_d$, $\mathbf{k}_{\text{orm}}$ or n, x_{surf} denotes a set of surface points and $\epsilon \sim \mathcal{N}(0, \sigma = 0.01)$ is a small random displacement vector.

$\mathcal{L}_{n'}$ enforce the tangent space normal map close to $(0,0,1)$ to prevent the coupling between the normal map and the environment light, which is formulated as:

$$\mathcal{L}_{n'} = \frac{1}{|\mathbf{x}_{\text{surf}}|} \sum_{x \in \mathbf{x}_{\text{surf}}} 1 - \frac{n'(x) + n'(x+\epsilon)}{|n'(x) + n'(x+\epsilon)|} \cdot (0,0,1). \quad (14)$$

$\mathcal{L}_{\text{light}}$ is the monochrome image loss between the demodulated lighting terms and the reference image to disentangle material parameters and light,which is formulated as:

$$\mathcal{L}_{\text{light}} = |Y(\mathbf{c}_d + \mathbf{c}_s) - V(I_{\text{GT}})|, \quad (15)$$

where $Y(x)$ is the luminance operator, $V(x)$ computes the value component of HSV, $\mathbf{c}_d$ and $\mathbf{c}_s$ are diffuse and specular lighting, separately.

5 EVALUATION STUDIES

To the best of our knowledge, only a few works achieve joint optimization of triangle mesh, PBR material texture, and light under unknown illumination. Therefore, we use NVDIFFRECMC [Hasselgren et al. 2022], NVDIFFREC [Munkberg et al. 2022], and NeRFactor [Zhang et al. 2021b] as our baselines. The first two approaches share the same task as ours, while NeRFactor, as a representative of neural field-based methods, is taken as a baseline in their works. We

	Hotdog			Lego			Drums★			Ficus★		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRFactor	25.452	0.912	0.124	20.325	0.832	0.140	21.570	0.908	0.086	21.636	0.917	0.095
NVDIFFREC	29.460	0.933	0.091	19.483	0.818	0.135	22.388	0.912	**0.083**	**27.831**	**0.961**	**0.048**
NVDIFFRECMC	28.606	0.941	0.101	20.464	0.835	0.145	22.750	0.917	0.086	25.162	0.942	0.060
JOC	**30.688**	**0.948**	**0.080**	**22.065**	**0.854**	**0.128**	**22.949**	**0.919**	0.084	25.905	0.948	0.056

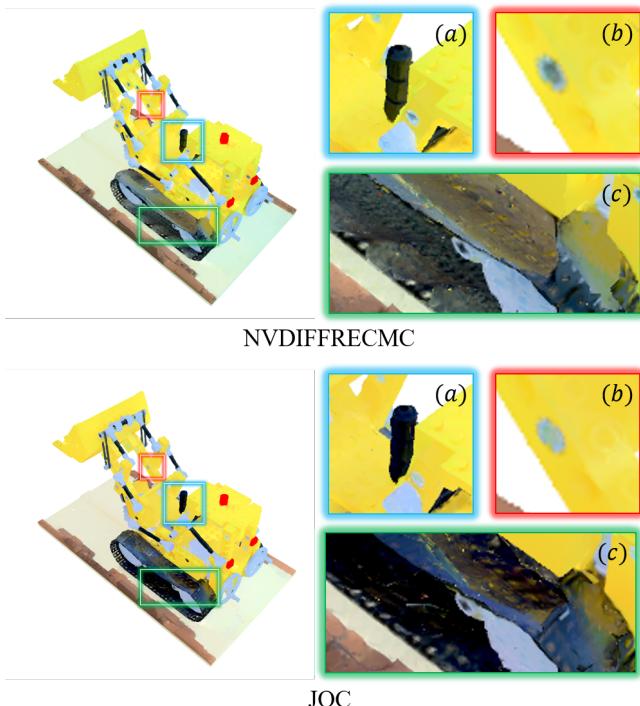|	Hotdog			Lego		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NVDIFFRECMC	25.516	0.926	0.124	22.359	0.865	0.126
JOC	**26.822**	**0.931**	**0.106**	**23.024**	**0.881**	**0.108**

NVDIFFRECMC

JOC

Fig. 4. Comparsion between reconstructed albedo of JOC and NVD-IFFRECMC [Hasselgren et al. 2022] from Lego scene in NeRF synthetic dataset. Thanks to modeling indirect illumination in inverse rendering, JOC successfully prevents baking (a) inter-reflections (b) holes and (c) color bleeding into albedo.

compare the quality of reconstruction with baselines in Section 5.1, and conduct ablation study to show the effect of leveraging better SDF initial estimate from neural fields in Section 5.2. Besides, we show the efficiency of NRC in Section 5.3. Our reconstructions are directly suitable for modern digital content creation tools and game engines, as shown in Figure 9.

Note that InvRender [Zhang et al. 2022b] also leverage off-the-shelf neural scene representation methods as priors, however, the reconstruction results of InvRender are implicit, which are unsuitable for modern graphics pipelines to render. Despite this, following InvRender, we utilize IDR [Yariv et al. 2020] as the neural fields in all presented experiment results.

5.1 Comparsions

Table 1 presents the quantitative results on the NeRFactor [Zhang et al. 2021b] synthetic dataset from individual scenes in terms of PSNR, SSIM, and LPIPS. Furthermore, we provide qualitative comparison in Figure 7 for relighting results and Figure 8 for reconstruction quality. Additionally, as the training images in NeRFactor dataset are lit under low frequency lighting, we conduct extra experiments with training images in NeRF [Mildenhall et al. 2020] synthetic dataset which contains high frequency lighting and show the results in Table 2. Following previous works [Hasselgren et al. 2022; Munkberg et al. 2022; Zhang et al. 2021b], we normalize the relit renderings by the average intensity of the reference. Our model achieves competitive performance in both quantitative and qualitative comparisons.

Figure 4 illustrates the advantage of modeling indirect illumination in inverse rendering, namely, avoiding the baking (a) inter-reflections (b) holes and (c) color bleeding into the material.

5.2 Ablation Study

Table 3 shows the effect of leveraging the neural field to initialize the SDF for DMTet in our experiments. For simple scene like Hotdog and Ficus, the better initial estimate leads to better performance. However, the surface prior limits the search space for geometric reconstruction, thus slightly degrading the reconstruction quality for complex scene as Drums and Lego.

5.3 Efficiency of Neural Radiance Cache

Caching. InvRender [Zhang et al. 2022b] is another work that utilizes a neural network to model indirect illumination. To be specific, InvRender takes a pre-trained IDR [Yariv et al. 2020] as the outgoing radiance field, and trains a network that maps a 3D location to its

Table 3. Ablation study on relighting results for NeRFactor dataset with per-scene quality metrics. The presented metrics are the averge over eight different view points under eight different light probes.

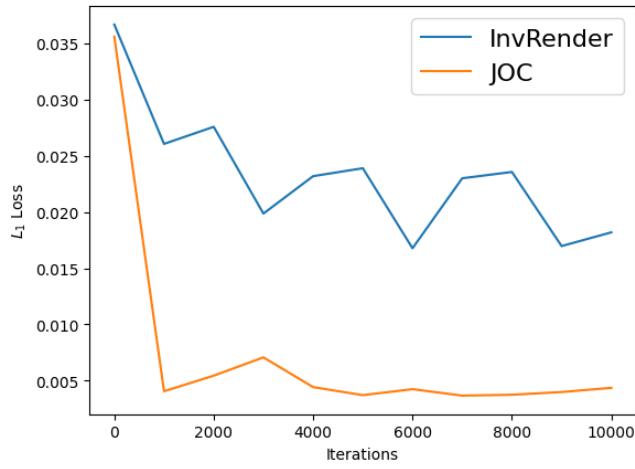| | Hotdog | | | Lego | | | Drums | | | Ficus | | |
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
JOC *w/o SDF Init.*	29.570	0.941	0.092	**22.091**	0.853	0.128	**23.001**	**0.920**	**0.082**	25.338	0.944	0.059
JOC	**30.688**	**0.948**	**0.080**	22.065	**0.854**	0.128	22.949	0.919	0.084	**25.905**	**0.948**	**0.056**

Fig. 5. Comparing the training processes of InvRender and JOC (NRC) for representing indirect illumination: InvRender predicts the indirect illumination received by surface points and represents it with SGs, training by supervising the predicted incoming radiance with the corresponding outgoing radiance. In contrast, JOC directly predicts the scattered radiance at surface points via NRC. For a fair comparison, JOC also employs L_1 loss as the objective function in the experiment depicted. It is evident that JOC achieves lower loss within fewer iteration steps. Both models exhibit similar iteration time (41 minutes versus 38 minutes for 10,000 iterations)
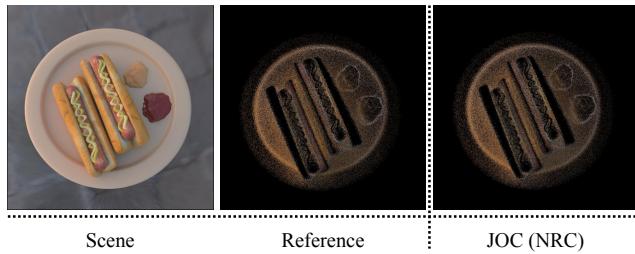.

Fig. 6. The ability of NRC to cache the scattered radiance. NRC faithfully reproduces the scattered radiance captured from IDR [Yariv et al. 2020]

indirect incoming illumination represented as a mixture of SGs. In contrast, NRC directly caches the exitant radiance of a 3D location with the scattered direction and the analytic normal. Figure 5 illustrates the training of NRC compared to InvRender on the Hotdog scene in the dataset provided by InvRender. Besides, Figure 6 shows the ability of NRC to cache the scattered radiance.

Querying. We modify the renderer of NVDIFFRECMC [Hasselgren et al. 2022] which is based on Optix7 [Parker et al. 2010] for utilizing the NRC. To be specific, we record all requisite information after each sampling and pass it to the NRC within Python, resulting in additional overheads such as Optix start-up costs, thread synchronization, and data I/O. Despite our implementation, as a proof of concept, not being highly optimized, these costs remain manageable: With 128 samples per pixel (spp), the iteration time in comparison to NVDIFFRECMC has increased by 40%, while demanding an additional 20% of GPU memory. Yet, to model indirect illumination in a differentiable MC renderer, these overheads are deemed justifiable, considering the prohibitive time and memory consumption associated with conventional multi-bounce differentiable MC renderer. For instance, in a conventional differentiable Monte Carlo renderer, the storage requirement grows linearly with number of bounces and quickly exhausts all available GPU memory. In contrast, we directly simulate infinite-bounce transport via NRC [Müller et al. 2021]. For a more comprehensive understanding of the costs associated with multi-bounce differentiable MC rendering, please refer to [Vicini et al. 2021].

6 DISCUSSION

Although the encoding techniques we utilize for the NRC slightly relax the need for fine geometry [Müller et al. 2022; Müller et al. 2019], a neural field capable of representing relatively accurate geometry is still necessary. Furthermore, while NVDIFFRECMC [Hasselgren et al. 2022] supports the reconstruction of transparency objects, our approach has not yet been extended to handle such objects. Future work could potentially explore enhancing the robustness of the NRC suitable for neural fields and expanding support for a wider variety of objects.

7 CONCLUSION

We propose a novel framework that reconstructs triangle meshes, Physically-Based materials, and High Dynamic Range light probes under unknown illumination. Compared to previous methods, we effectively model indirect illumination in modern graphics pipelines through the Neural Radiance Cache [Müller et al. 2021] trained on neural fields with minimal assumptions, thereby enhancing the separation of illumination effects and reconstructed materials. Qualitative and quantitative evaluations demonstrate the effectiveness of our method.
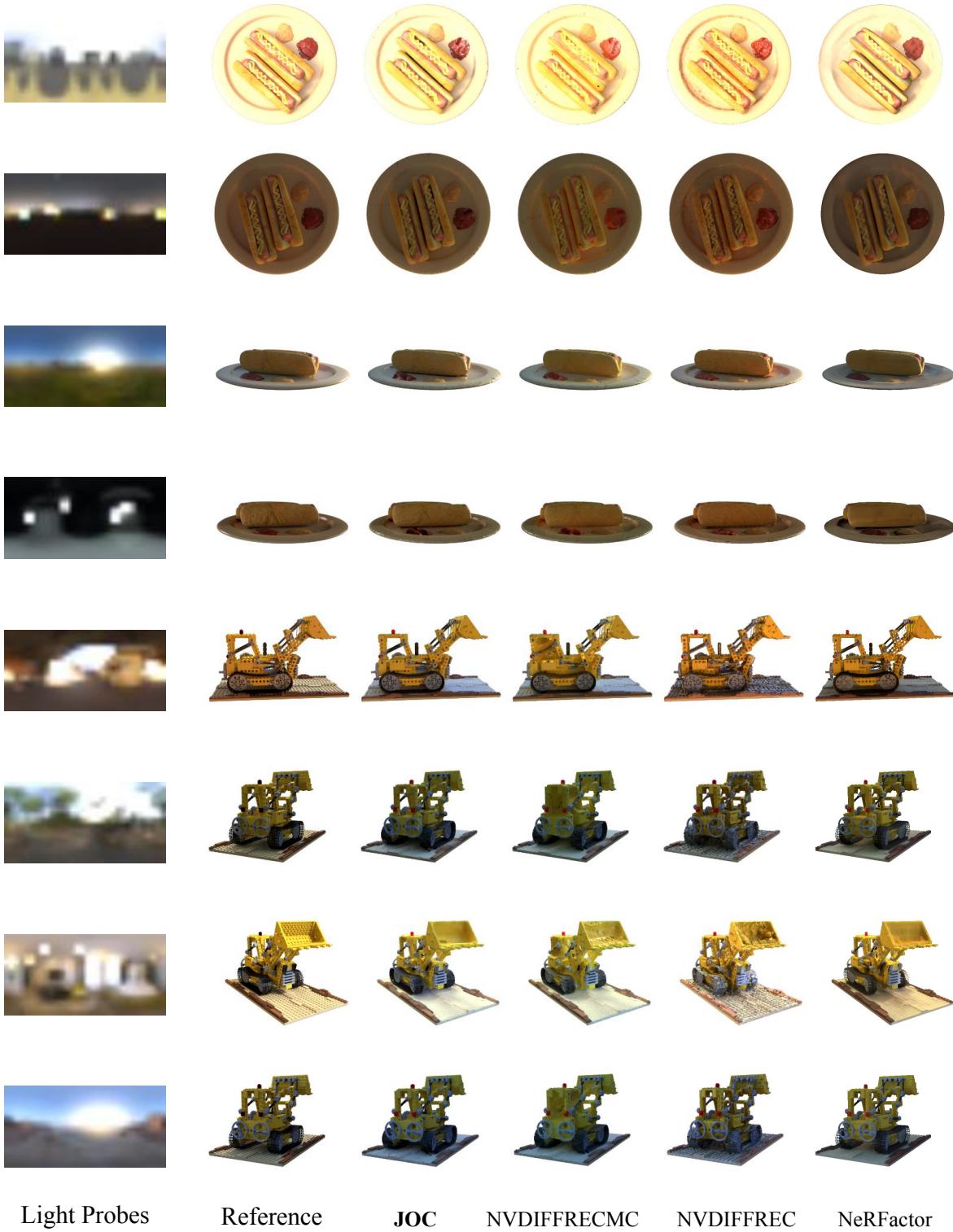
| Light Probes | Reference | **JOC** | NVDIFFRECMC | NVDIFFREC | NeRFactor |

Fig. 7. Qualitative comparisons on the NeRFactor [Zhang et al. 2021b] synthetic dataset. Please zoom-in for details.

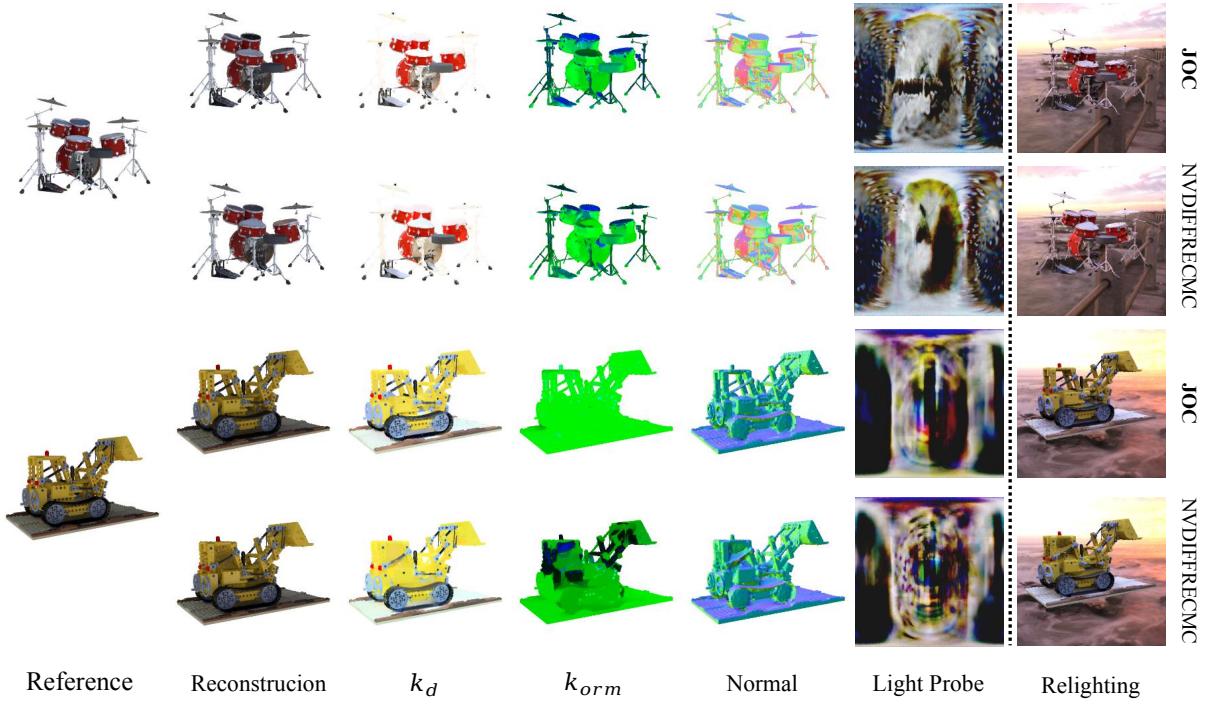| Reference | Reconstrucion | k_d | k_{orm} | Normal | Light Probe | Relighting |

Fig. 8. Reconstruction examples from the NeRFactor [Zhang et al. 2021b] synthetic dataset and relit renderings under a light probe from Poly Haven (CC0). Our results outperform the previous work [Hasselgren et al. 2022] in terms of visual quality, surface reconstruction and material separation. Please zoom-in for details.
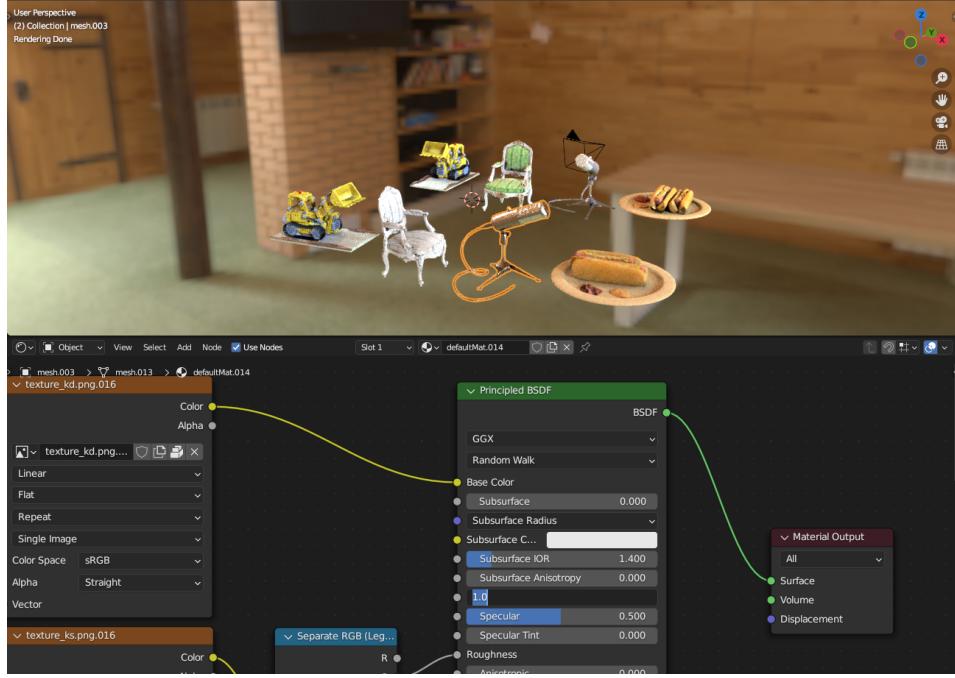

Fig. 9. We can use the reconstruction results in modern DCC tools , i.e., Blender, for various applications as scene editing and material editing.

REFERENCES

Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. 2022. Neural RGB-D Surface Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6290–6301.

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. *IEEE International Conference on Computer Vision (ICCV)* (2021).

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. 2021a. NeRD: Neural Reflectance Decomposition from Image Collections. In *IEEE International Conference on Computer Vision (ICCV)*.

Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. 2021b. Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *European Conference on Computer Vision (ECCV)*.

Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. 2014. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Trans. Graph.* 33, 6 (2014), 1–12.

Fridovich-Keil and Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. 2022. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)* 35 (2022), 3403–3416.

Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. 2003. Linear light source reflectometry. *ACM Trans. Graph.* 22, 3 (2003), 749–758.

Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A Wilson, and Paul Debevec. 2009. Estimating specular roughness and anisotropy from second order spherical gradient illumination. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 1161–1170.

G. Greger, P. Shirley, P.M. Hubbard, and D.P. Greenberg. 1998. The irradiance volume. *IEEE Computer Graphics and Applications* 18, 2 (1998), 32–43. https://doi.org/10.1109/38.656788

Darya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. 2016. BRDF representation and acquisition. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 625–650.

Saeed Hadadan, Shuhong Chen, and Matthias Zwicker. 2021. Neural Radiosity. *ACM Trans. Graph.* 40, 6, Article 236 (dec 2021), 11 pages. https://doi.org/10.1145/3478513.3480569

Saeed Hadadan, Geng Lin, Jan Novák, Fabrice Rousselle, and Matthias Zwicker. 2023. Inverse Global Illumination using a Neural Radiometric Prior. arXiv:2305.02192 [cs.CV]

Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. 2022. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ajay Jain, Matthew Tancik, and Pieter Abbeel. 2021. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 5885–5894.

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

Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy Mitra. 2022. ReLU Fields: The Little Non-Linearity That Could. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 27, 9 pages. https://doi.org/10.1145/3528233.3530707

J. Krivanek, P. Gautron, S. Pattanaik, and K. Bouatouch. 2005. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics* 11, 5 (2005), 550–561. https://doi.org/10.1109/TVCG.2005.83

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Trans. Graph.* 37, 6, Article 222 (Dec. 2018), 11 pages. https://doi.org/10.1145/3272127.3275109

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3D: High-Resolution Text-to-3D Content Creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing Discontinuous Integrands for Differentiable Rendering. *ACM Trans. Graph.* 38, 6, Article 228 (nov 2019), 14 pages. https://doi.org/10.1145/3355089.3356510

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7210–7219.

Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. 2022. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16190–16199.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Trans. Graph.* 38, 5, Article 145 (Oct. 2019), 19 pages. https://doi.org/10.1145/3341156

Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. 2020. Neural Control Variates. *ACM Trans. Graph.* 39, 6, Article 243 (nov 2020), 19 pages. https://doi.org/10.1145/3414685.3417804

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-Time Neural Radiance Caching for Path Tracing. 40, 4 (2021). https://doi.org/10.1145/3450626.3459812

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. 2022. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8280–8290.

Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. 2022. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5480–5490.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Michael Oechsle, Songyou Peng, and Andreas Geiger. 2021. UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. In *International Conference on Computer Vision (ICCV)*.

Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al. 2010. Optix: a general purpose ray tracing engine. *ACM Trans. Graph.* 29, 4 (2010), 1–13.

Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. 2013. Global Illumination with Radiance Regression Functions. *ACM Trans. Graph.* 32, 4, Article 130 (jul 2013), 12 pages. https://doi.org/10.1145/2461912.2462009

Daniel Scherzer, Chuong H Nguyen, Tobias Ritschel, and Hans-Peter Seidel. 2012. Preconvolved radiance caching. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1391–1397.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ari Silvennoinen and Jaakko Lehtinen. 2017. Real-time global illumination by precomputed local reconstruction from sparse radiance probes. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022b. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. 2022a. Neural 3D Reconstruction in the Wild. In *ACM SIGGRAPH 2022 Conference Proceedings*.

Jiakai Sun, Zhanjie Zhang, Jiafu Chen, Guangyuan Li, Boyan Ji, Lei Zhao, and Wei Xing. 2023. VGOS: Voxel Grid Optimization for View Synthesis from Sparse Inputs. *arXiv preprint arXiv:2304.13386*.

Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. 2022. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9.

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. 2023. Nerfstudio: A Modular Framework for Neural Radiance Field Development. *arXiv preprint arXiv:2302.04264* (2023).

Eric Veach. 1998. *Robust Monte Carlo methods for light transport simulation.* Stanford University.

Eric Veach and Leonidas J Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 419–428.

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. 2022. Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths Using Constant Memory and Linear Time. *ACM Trans. Graph.* 40, 4, Article 108 (jul 2021), 14 pages. https://doi.org/10.1145/3450626.3459804

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. 2023. Neural Fields meet Explicit Geometric Representation for Inverse Rendering of Urban Scenes. arXiv:2304.03266 [cs.CV]

Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. 1988. A Ray Tracing Solution for Diffuse Interreflection. *SIGGRAPH Comput. Graph.* 22, 4 (jun 1988), 85–92. https://doi.org/10.1145/378456.378490

Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. 2016. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Trans. Graph.* (Nov 2016), 1–12. https://doi.org/10.1145/2980179.2980248

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2022. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 641–676.

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

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. 2020. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020).

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4578–4587.

Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. 2022a. SDF-Studio: A Unified Framework for Surface Reconstruction. https://github.com/autonomousvision/sdfstudio

Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. 2022b. MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-Space Differentiable Rendering. *ACM Trans. Graph.* 39, 4 (2020), 143:1–143:19.

Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. 2022a. IRON: Inverse Rendering by Optimizing Neural SDFs and Materials from Photometric Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021a. PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. 2021b. NeRFactor: Neural Factorization of Shape and Reflectance under an Unknown Illumination. *ACM Trans. Graph.* 40, 6, Article 237 (dec 2021), 18 pages. https://doi.org/10.1145/3478513.3480496

Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. 2022b. Modeling Indirect Illumination for Inverse Rendering. In *CVPR*.