NeRF as Non-Distant Environment Emitter in Physics-based Inverse Rendering

JINGWANG LING, Tsinghua University, China
RUIHAN YU, Tsinghua University, China
FENG XU†, Tsinghua University, China
CHUN DU, Tibet University, China
SHUANG ZHAO, University of California, Irvine, USA

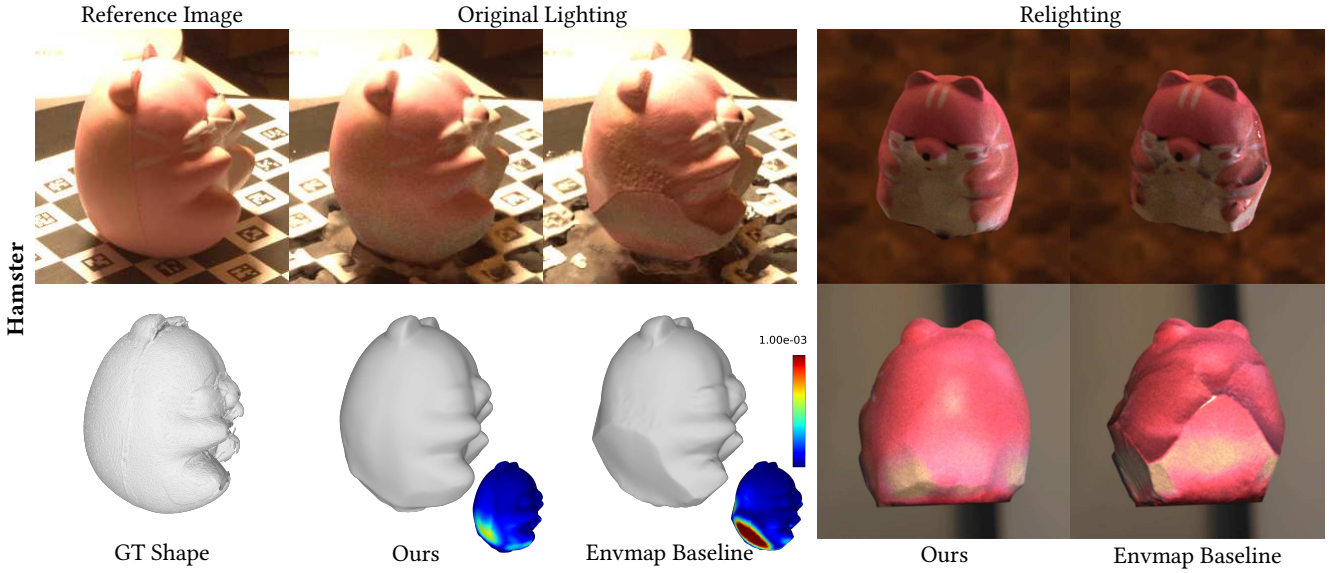

Fig. 1. We perform physics-based inverse rendering on real captured data featuring non-distant lighting. In this scenario, the commonly used environment map inaccurately models the lighting and leads to artifacts visible in relighting and shape reconstruction results. In contrast, our proposed NeRF emitter can accurately model non-distant lighting to achieve high-quality inverse rendering.

Physics-based inverse rendering aims to jointly optimize shape, materials, and lighting from captured 2D images. Here lighting is an important part of achieving faithful light transport simulation. While the environment map is commonly used as the lighting model in inverse rendering, we show that its distant lighting assumption leads to spatial invariant lighting, which can be an inaccurate approximation in real-world inverse rendering. We propose to use NeRF as a spatially varying environment lighting model and build an inverse rendering pipeline using NeRF as the non-distant environment emitter. By comparing our method with the environment map on real and synthetic datasets, we show that our NeRF-based emitter models the scene lighting more accurately and leads to more accurate inverse rendering. Project page and video: nerfemitterpbir.github.io.

1 INTRODUCTION

Reconstructing the shape, material, and lighting of an object from 2D images is a long-standing problem in computer graphics and vision, with many applications such as relighting and scene editing. With recent advances, physics-based inverse rendering has become popular as it could more faithfully simulate the light interactions within the real captured scene, leading to physically accurate reconstruction. Physics-based inverse rendering involves constructing the rendering equation in a differentiable way to establish the connections between the input images and the to-be-solved scene parameters. Then it is possible to minimize a rendering loss between the rendered and captured images by differentiating the rendering equation to obtain gradients and using gradient descent to optimize scene parameters, i.e. shape, material, and lighting.

To accurately simulate the light transport for inverse rendering, an accurate scene model is essential. However, in commonly used object-centric capture scenarios, the number of captured images is limited, and the camera positions are relatively localized, making it challenging to reconstruct the entire scene. Therefore, inverse rendering approaches often model the object of interest as shape and materials and approximate the rest of the scene as environment lighting. The environment map has become a popular choice to approximate lighting around objects. However, we show that in scenes where the light source is not infinitely far away, an environment

†Corresponding author.
Authors' addresses: Jingwang Ling, Tsinghua University, China, lingjw20@mails.tsinghua.edu.cn; Ruihan Yu, Tsinghua University, China, aurораryan0301@gmail.com; Feng Xu†, Tsinghua University, China, xufeng2003@gmail.com; Chun Du, Tibet University, China, duchun@utibet.edu.cn; Shuang Zhao, University of California, Irvine, USA, shz@ics.uci.edu.

map becomes a poor approximation, leading to inaccurate inverse rendering results.

The key is that under the distant lighting assumption of the environment map, the lighting distribution is spatially invariant. When the light source is not distant, it exhibits strong parallax effects as the light comes from different directions at different object surface positions. A spatially invariant lighting model now becomes a poor approximation of the actual lighting conditions at different object surface positions. To accurately handle the non-distant lighting, a representation that can synthesize spatially varying incoming radiance distribution is required.

We find that the neural radiance field (NeRF) [Mildenhall et al. 2020] is suitable to represent the real spatially varying lighting. Different from the environment map which essentially models a 2D radiance field that is infinitely distant, NeRF models a 3D radiance field that resides on the non-distant volumetric densities. By using an HDR NeRF to model the unbounded scene that surrounds the objects, we can synthesize 3D-consistent incoming radiance at any object shading point, just as NeRF excels in novel-view synthesis.

Therefore, to achieve more accurate inverse rendering when the lighting is not distant, we propose a technique to use a neural radiance field (NeRF) to model the environmental lighting. We model the scene in a hybrid manner, where objects are represented by surfaces and materials, while the surrounding environment lighting is modeled by a NeRF. To render this hybrid scene, we derive the rendering equation taking into account both geometric surfaces and NeRF. To adapt NeRF into physics-based inverse rendering, we devise an emitter importance sampling approach for NeRF to reduce the rendering variance. We build an inverse rendering pipeline that jointly reconstructs shape, material, and NeRF lighting from the captured images effectively. We capture both real and synthetic datasets featuring non-distant lighting and compare our method with the physics-based inverse rendering pipeline which uses the environment map as the lighting model. Our results show that when the lighting is not infinitely distant, an environment map inaccurately models the lighting and leads to artifacts in inverse rendering results. In contrast, our proposed NeRF emitter accurately models non-distant lighting and achieves high-quality inverse rendering.

To the best of our knowledge, we are the first to use NeRF to represent the environment lighting with better capability to model the non-distant lighting effects. Furthermore, we successfully involve the NeRF lighting in the inverse rendering pipeline with a novel emitter importance sampling method to reconstruct the geometry, material, and lighting of a scene with largely improved performance.

2 RELATED WORK

Neural Radiance Fields (NeRF) [Mildenhall et al. 2020] employ neural networks to model the radiance field in 3D scenes, thereby enabling novel-view synthesis. Mip-NeRF 360 [Barron et al. 2022] proposes a scene contraction method that extends NeRF to unbounded scenes, enabling NeRF to model the surround environment of an object of interest. Instant-NGP [Müller et al. 2022] accelerates NeRF by proposing a hybrid grid-network field representation. NeRFStudio [Tancik et al. 2023] and Zip-NeRF [Barron et al. 2023] combine the works by Barron et al. [2022]; Müller et al. [2022] to accelerate the modeling of an unbounded environment using NeRF. RawNeRF [Mildenhall et al. 2022] and VR-NeRF [Xu et al. 2023a] explore extending NeRF to high dynamic range, thereby improving the novel-view synthesis quality in dark-light or virtual reality scenarios. Given that light sources in physics-based rendering are often HDR, combining HDR technique into environment NeRF holds the potential for NeRF to model an HDR environment emitter.

Based on the powerful modeling capabilities of neural networks, there are research works exploring the use of neural networks to represent complex luminaires. The works by Condor and Jarabo [2022]; Zhu et al. [2021] utilizes neural networks to model the light field or radiance field of virtual luminaries, thereby accelerating physics-based rendering by reducing the variance. DMRF [Qiao et al. 2023] considers inserting a virtual mesh into the trained NeRF and simulate the interactions between them. These works focus on using pretrained networks to model light in forward rendering, while our work considers employing NeRF in inverse rendering as an emitter to reconstruct the scene parameters.

Differentiable Rendering involves differentiably constructing the rendering equation to establish connections between input images and the to-be-solved scene parameters. Then we can differentiate the rendering equation to obtain gradients, enabling the use of gradient descent for optimizing scene parameters. A key challenge in differentiable rendering lies in the differentiation of visibility discontinuity related to shapes. The work by [Li et al. 2018; Zhang et al. 2020, 2019] explicitly samples the discontinuity edges to estimate the boundary term with respect to shape. Another approach based on warped-area sampling [Bangaru et al. 2020] resolves shape discontinuity by converting boundary integrals into area integrals. This approach can be extended from mesh to signed distance funtion to enable differentiable SDF rendering [Bangaru et al. 2022; Vicini et al. 2022], and is also recently extended to path space to differentiate path integrals [Xu et al. 2023b]. Differentiating visibility-related discontinuity is essential to enable shape optimization in inverse rendering.

Inverse Rendering aims to use gradient descent to minimize a rendering loss between rendered and captured images to reconstruct the scene shape, material, and lighting parameters. An accurate lighting model is essential to achieve high-quality reconstruction via inverse rendering. However, current inverse rendering methods often assume simple light models, which restrict the capture setup or lead to an inaccurate approximation of the real-world scene lighting. For example, the works by Bi et al. [2020a,b]; Zhang et al. [2022a] require co-located camera-light setup. Other methods assume a point light or directional light [Ling et al. 2023; Yang et al. 2022a,b]. These assumptions require a specific captured setup in a controlled environment. To perform inverse rendering in an ordinary environment, the environment map [Debevec and Malik 1997] is a commonly used lighting approximation. Various inverse rendering works aim to reconstruct a single object [Boss et al. 2021a,b; Jin et al. 2023; Srinivasan et al. 2021; Sun et al. 2023; Zhang et al. 2021, 2022b], outdoor buildings [Rudnev et al. 2021] or humans [Chen and Liu 2022], all using the environment map as the scene lighting approximation. However, an environment map assumes the scene lighting is infinitely distant, which is rarely the case in real-world capture setups. In this work we show that when the assumption is violated,

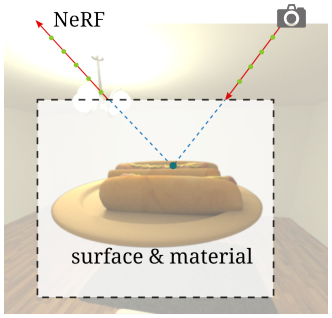

Fig. 2. The region inside the bounding box is modeled as surface and material, while the region outside the bounding box is modeled by NeRF as environment lighting.

the environment map becomes a poor approximation of the scene lighting, leading to inaccuracies and artifacts in inverse rendering. In contrast, our proposed NeRF-based emitter can robustly handle the scenarios with more accurate lighting that forgoes the distant lighting assumption.

3 NERF-BASED NON-DISTANT EMITTERS

3.1 Preliminaries

Here, we review differentiable surface rendering [Nimier-David et al. 2020] and NeRF [Mildenhall et al. 2020], comparing their similarities and differences. In physics-based rendering, the image pixels I_1, I_2, ..., I_k, ..., I_N individually measure the integral of the product of the sensor importance function W and the incident light intensity L_i at the position $\mathbf{p}$:

$$I_k = \int_{\mathcal{A}} \int_{\mathcal{S}^2} W_k(\mathbf{p}, \omega) L_i(\mathbf{p}, \omega) d\omega^{\perp} d\mathbf{p}, \tag{1}$$

where ω is the light direction and $\omega^{\perp}$ is the projected solid angle. Regarding $L_i(\mathbf{p}, \omega)$, there are differences between surface rendering and volume rendering due to the distinct assumptions made about the scene. Surface rendering assumes a vacuum between surfaces, where the incoming radiance is equal to the outgoing radiance at the first intersection point of the ray, therefore we have

$$L_i(\mathbf{p}, \omega) = L_i^s(\mathbf{p}, \omega) = L_o(\mathbf{r}(\mathbf{p}, \omega, t_0), -\omega), \tag{2}$$

where t_0 is the distance of the closest intersection point $\mathbf{r}(\mathbf{p}, \omega, t_0)$ to $\mathbf{p}$. The calculation of outgoing radiance L_o from the surface involves a spherical integral, which is expressed by the rendering equation

$$L_o(\mathbf{p}, \omega) = L_e(\mathbf{p}, \omega) + \int_{\mathcal{S}^2} L_i(\mathbf{p}, \omega') f_s(\mathbf{p}, \omega, \omega') d\omega'^{\perp}. \tag{3}$$

Here, L_e represents the surface's emitted radiance, and f_s denotes the BSDF properties of the surface. The L_i on the right-hand side involves recursive computation and can be implemented using path tracing.

On the other hand, NeRF makes different assumptions about the scene. It assumes that the scene is filled with volumetric density that emits light, without the presence of surfaces or volumetric scattering. In this case, the computation of L_i is given by the (non-scattering) volume rendering equation

$$L_i(\mathbf{p}, \omega) = L_i^v(\mathbf{p}, \omega) = \int_0^{\infty} T(\mathbf{p}, \omega, t) \sigma(\mathbf{r}(\mathbf{p}, \omega, t)) \mathbf{c}(\mathbf{r}(\mathbf{p}, \omega, t), -\omega) dt, \tag{4}$$

where σ is the volumetric density, $\mathbf{c}$ is the emitting radiance, and T is the accumulated transmittance modeling the absorption effect of occluding densities

$$T(\mathbf{p}, \omega, t) = \exp\left(-\int_0^t \sigma(\mathbf{r}(\mathbf{p}, \omega, s)) ds\right). \tag{5}$$

The assumptions made by NeRF bring the benefit of significantly simplifying the process of light transport in the scene. This simplification facilitates the reconstruction of radiance distribution information in the scene for novel view synthesis [Mildenhall et al. 2020]. In this paper, we will demonstrate that NeRF exhibits similar advantages in inverse rendering, modeling the light distribution in the scene to serve as the light source for inverse rendering.

For inverse rendering, whether dealing with surface or volume scenes, it is necessary to differentiate the rendering equation to compute gradients. In differentiable surface rendering, it is necessary to differentiate Equation 3 to obtain

$$\partial_{\mathbf{x}} L_o(\mathbf{p}, \omega) = \partial_{\mathbf{x}} L_e(\mathbf{p}, \omega) + \int_{\mathcal{S}^2} [\partial_{\mathbf{x}} L_i(\mathbf{p}, \omega') f_s(\mathbf{p}, \omega, \omega') \tag{6}$$

$$+ L_i(\mathbf{p}, \omega') \partial_{\mathbf{x}} f_s(\mathbf{p}, \omega, \omega')] d\omega'^{\perp}. \tag{7}$$

Note that the differentiation operation involves a spherical integral with respect to the scene parameters, specifically $\partial_{\mathbf{x}} L_i(\mathbf{p}, \omega')$. This should be carefully considered when computing derivatives related to the light source. Additionally, this derivation overlooks the discontinuities introduced by changes in scene geometry. It is essential to employ additional techniques, such as reparameterization [Vicini et al. 2022], to address the discontinuities arising from geometric variations.

In contrast, the differentiation of NeRF is relatively straightforward. After discretizing the continuous line integral through quadrature from ray marching, the process becomes trivially differentiable. Gradients can be computed using an automatic differentiation (AD) framework, such as PyTorch.

3.2 Hybrid Rendering of Surfaces and NeRF

To accurately model the lighting effects from the environment surrounding the object, we propose the use of a neural radiance field (NeRF) to model environmental lighting. We adopt a hybrid scene representation of surfaces and NeRF. Based on the bounding region of the object of interest, we divide the scene into two regions: the internal region, where the object is represented by surfaces and materials, and the external region, which is represented by the NeRF-based environmental lighting. The scene layout is shown in Figure 2. To render the hybrid scene, we define the rendering equation taking into account both geometric surfaces and NeRF. For our scene assumption, the incoming radiance L_i is composed of $L_i'^s$ from the surface and $L_i'^v$ from NeRF:

$$L_i(\mathbf{p}, \omega) = L_i'^s(\mathbf{p}, \omega) + L_i'^v(\mathbf{p}, \omega). \tag{8}$$

For the surface component $L_i'^s$, it generally follows the surface light transport equation. However, here, we also need to consider the

occlusion from NeRF:

$$L_i'^s(\mathbf{p},\omega) = T(\mathbf{p},\omega,t_0)L_o(\mathbf{r}(\mathbf{p},\omega,t_0),-\omega), \tag{9}$$

where T is the accumulated transmittance of NeRF as defined in Equation 5. The NeRF contribution $L_i'^v$ is similar to Equation 4, except that we integrate from 0 to t_0 instead of ∞, where t_0 is the distance of the first surface intersection. t_0 goes to ∞ when there is no surface intersection.

An important notice is that computing L_o in Equation 9 follows the rendering equation in Equation 3, which contains a recursive L_i term. L_i here also contains both surface and volumetric components. Therefore, following Equation 8, we need to compute NeRF not only at the camera ray, but also at each secondary ray on the light path. When the light path finally goes out of the object bounding region and goes to infinity, we query NeRF to calculate the incoming radiance. NeRF essentially serves as a light source in this case.

Discussion. In the earlier derivations, it seems that the computational complexity of NeRF involves a proportionality with the length of the light path, leading to increased computational complexity. However, with specific scene segmentation, the number of rays querying NeRF is not proportional to the length of the light path. In the bounding region where the object is located, the density is consistently zero according to our scene assumption. If the region in which the object is located is convex, which holds in our case as a bounding box, then the line connecting any two points within this region remains within the same region. Consequently, the density at any point along the line between two points is zero, allowing the omission of the volume rendering term $L_i'^v$, making $L_i(\mathbf{p},\omega) = L_i'^s(\mathbf{p},\omega)$ for any ray originating from and ending inside the region. Consequently, when the camera is positioned outside the object's bounding box, any given light path enters the bounding box at most once and exits the bounding box at most once. Thus, each light path involves at most two rays for NeRF computation.

3.3 Emitter Importance Sampling for NeRF

To apply the NeRF light source to physically-based rendering, Monte Carlo sampling is required. Specifically, it is necessary to be able to sample the direction and compute the probability density of a specific sampled direction. Performing importance sampling for NeRF involves unique complexities, given that NeRF, as a volumetric light source, is composed of a density field with no fixed topology. Therefore, it is necessary to devise a specific importance sampling approach for NeRF.

Our base idea is to approximate regions of significant radiance contribution in NeRF using relatively simple geometric primitives. Specifically, we employ a Gaussian Mixture Model to fit a point cloud extracted from the bright regions of NeRF. This Gaussian Mixture Model serves as a proxy model for NeRF during importance sampling. First, we sample random rays originating from random positions inside the object bounding region. We use Equation 4 to compute the radiance along this ray and record the depth value that maximally contributes to the rendered radiance

$$t_{\max} = \arg\max_t T(\mathbf{p},\omega,t)\sigma(\mathbf{r}(\mathbf{p},\omega,t))\mathbf{c}(\mathbf{r}(\mathbf{p},\omega,t),-\omega). \tag{10}$$

Then we construct a sample point of position $\mathbf{r}(\mathbf{p},\omega,t_{\max})$ with weight $Y(L_i^v(\mathbf{p},\omega))$ to the point cloud, where Y is the luminance operator that converts RGB radiance to monochrome luminance. We aim to approximate the radiance distribution of the point cloud via fitting an isotropic Gaussian Mixture Model of $M = 64$ lobes. To efficiently minimize rendering variance using a limited number of lobes, inspired by MIS compensation [Karlík et al. 2019], we subtract the mean of the weights $\bar{Y}(L_i^v(\mathbf{p},\omega))$ from the weights of all point clouds. Only point clouds with positive weights are retained. As a result, the remaining point cloud can focus on accurately capturing the shape of very bright regions, while remaining unbiased via the multiple-importance sampling combined with BSDF sampling. Based on the compensated weights, we cluster the point cloud into M isotropic Gaussians, each characterized by a mean μ_i, covariance $\kappa_i^{-1}\mathbf{I}$ and weight λ_i, with $i \in [1, M]$. When performing emitter importance sampling for a shading point $\mathbf{p}$, we first project the isotropic Gaussians into von Mises–Fisher distributions $(\widehat{\mu}_i, \widehat{\kappa}_i, \widehat{\lambda}_i)$

$$\widehat{\mu}_i = \frac{\mu_i - \mathbf{p}}{\|\mu_i - \mathbf{p}\|_2},\ \widehat{\kappa}_i = \frac{\kappa_i}{\|\mu_i - \mathbf{p}\|_2},\ \widehat{\lambda}_i = \lambda_i, \tag{11}$$

and use the projected vMF lobes to perform sampling according to the work by Jakob [2012]. During inverse rendering, as NeRF updates its parameters in the joint optimization, the brightness distribution may change. To ensure that the sampling distribution matches the updated NeRF, we repeat the above process to establish an updated Gaussian mixture model every few optimization iterations.

3.4 Differentiable Hybrid Rendering

In Section 3.2, we combined surface rendering and volume rendering to formulate the rendering equation for a hybrid scene. It requires proper differentiation for inverse rendering to be performed accurately. The initial point of attention lies in the differentiation of the surface rendering equation, as described in Equation 6, which involves the computation of $\partial_{\mathbf{x}} L_i(\mathbf{p},\omega')$. $\partial_{\mathbf{x}} L_i(\mathbf{p},\omega')$ computes the gradient of the light radiance with respect to arbitrary scene parameters $\mathbf{x}$. When the geometry of the scene is optimizable, the shading position $\mathbf{p}$, also the ray starting position of the NeRF, is gradient attached, thus it's essential to consider the case where $\mathbf{x} = \mathbf{p}$. To compute correct gradients, we need to get gradients of the queried NeRF radiance with respect to the ray starting position via automatic differentiation and propagate it to the differentiable surface rendering. As the ray direction, ω', is obtained through detached sampling in differentiable rendering [Zeltner et al. 2021], its derivative can be neglected.

Another aspect to consider is during the integration in volume rendering. Since our upper limit of integration, t_0, is dependent on the scene's shape parameters, differentiating with respect to shape will generate a boundary term. When differentiating $L_i'^v$, we have:

$$\partial_{\mathbf{x}} L_i'^v(\mathbf{p},\omega) = T(\mathbf{p},\omega,t_0)\sigma(\mathbf{r}(\mathbf{p},\omega,t_0))\mathbf{c}(\mathbf{r}(\mathbf{p},\omega,t_0),-\omega)\partial_{\mathbf{x}} t_0 \tag{12}$$

$$+ \int_0^{t_0} \partial_{\mathbf{x}}(T(\mathbf{p},\omega,t)\sigma(\mathbf{r}(\mathbf{p},\omega,t))\mathbf{c}(\mathbf{r}(\mathbf{p},\omega,t),-\omega))dt. \tag{13}$$

However, in our specific scene setup, $\mathbf{r}(\mathbf{p},\omega,t_0)$ is guaranteed to be within the bounding box of the object, and the density at its location, $\sigma(\mathbf{r}(\mathbf{p},\omega,t_0))$, is zero. Therefore, this boundary term can be safely ignored.

4 INVERSE RENDERING USING NERF EMITTER

Building upon the derivations above, we have implemented a pipeline that integrates NeRF as an emitter into physics-based inverse rendering. While previous inverse rendering methods typically require object foreground masks as input to exclude background influences, our pipeline can simultaneously obtain scene information from both foreground and background pixels and jointly optimize shape, material, and NeRF emitter based on the gradient of the rendering loss. However, physics-based inverse rendering often requires proper initialization. Therefore, we designed a multi-stage optimization process that use NeRF to aid the initialization of inverse rendering to facilitate the optimization process. Also, because of the widespread use of environment maps, existing datasets tend to avoid using nearby light sources, leading to a lack of thorough testing for scenes with close light sources. We established a capture system to thoroughly test such scenarios.

4.1 Staged optimization

Compared to surface-based inverse rendering, NeRF exhibits robustness to parameter initialization. Therefore, we first represent the entire scene, both inside and outside the bounding box using NeRF and perform ordinary NeRF training as initialization. We then perform TSDF fusion using the NeRF density within the object bounding box, and use the fusion result as object shape initialization. Then, we set the density inside the object bounding box to zero and model the bounding box region as object shape and material instead. At this point, NeRF has a reasonable initial lighting approximation, and the object's geometry has been initialized through NeRF. Finally, we perform inverse rendering until convergence.

4.2 Non-distant emitter capture system

To thoroughly test inverse rendering on scenes where the light is not infinitely distant, we establish a capture setup to capture a dataset featuring non-distant light sources. In addition, to address the ambiguity under the single-light condition, we provide multiple lighting conditions for inverse rendering within a single scene by rotating the object. Each scene setup features a light source near the object. We place the object on a turntable to rotate the object. For each rotation pose, we capture multi-view images using the camera. The camera features object-centric viewpoints, but lighting information could be observed from background pixels. To accurately capture the lighting information of the scene, we utilized a DSLR camera to capture HDR images.

5 IMPLEMENTATION DETAILS

5.1 HDR NeRF Parameterization and Training

Our NeRF model is based on the nerfacto proposed by Tancik et al. [2023]. To accurately model the light source using NeRF, we need to train NeRF to output High Dynamic Range (HDR) radiance values. Our inverse rendering pipeline takes HDR data as input. Following the approach in RawNeRF [Mildenhall et al. 2022], we use an exponential activation function $\exp(x-5)$ as the output activation function for NeRF radiance. During the NeRF pretraining stage and the inverse rendering stage, we utilize the relative L1 Loss as the rendering loss function, described as

$$\mathcal{L}_{\text{render}} = \frac{1}{N}\sum_{k=1}^{N}\left\|\frac{I_k - \widehat{I}_k}{I_k + \epsilon}\right\|_1, \tag{14}$$

where I_k is the rendered pixel, $\widehat{I}_k$ is the pixel on the captured image, and $\epsilon = 10^{-3}$ is a small number introduced to downweight excessively dark pixels. We use gradient scaling [Philip and Deschaintre 2023] to prevent NeRF from generating floaters. We follow the approach proposed by Tancik et al. [2023] to apply L_∞ contraction to the scene to handle the environmental lighting of unbounded scenes.

5.2 Integrating NeRF into Differentiable Rendering

Our inverse rendering pipeline is implemented based on combining the NeRFStudio [Tancik et al. 2023] and the Mitsuba 3 [Jakob et al. 2022] framework. The former is a NeRF framework containing neural networks, while the latter is a megakernel-based differentiable renderer. Due to the current limitation of calling neural networks within megakernels, integrating these two systems requires special techniques.

To begin with, we encapsulate NeRF and its importance sampling approach as an emitter within the Mitsuba 3 framework. Importance sampling is performed on Mitsuba 3 written in Dr.Jit, and during light source evaluation, the ray starting positions and directions are transferred to PyTorch for NeRF computation. The computed radiance or derivatives are then passed back to Mitsuba 3. To avoid calling neural networks within the megakernel, we split a rendering operation into two megakernels to separate the computation of light sources. Specifically, the first kernel generates rays to query the NeRF light source, and the second kernel gathers the NeRF evaluation to compute the pixel color. Instead of employing next event estimation, we utilize one-sample MIS combining emitter importance sampling and BSDF sampling. This allows deferring all light source queries for all light paths until the end of the light path.

Due to the large memory footprint of NeRF, it supports a limited ray batch size. However, differentiable rendering systems typically work on a full image or an image patch, which exceeds the maximum batch size that NeRF can handle. Therefore, we split the rays into batches and sequentially let NeRF process them. However, automatic differentiation systems rely on retaining computation graphs to make the NeRF evaluation operation differentiable. To avoid excessive GPU memory usage, we do not store computation graphs but recompute them during gradient computation, based on the checkpointing concept [Volin and Ostrovskii 1985]. Specifically, during forward rendering, we perform a detached gradient NeRF evaluation without retaining the computation graph. During backward propagation, we divide the queried rays into smaller batches, recover the computation graph for each batch of rays, and perform gradient computation via automatic differentiation. We record the random seeds during forward rendering and recover the seeds during backward propagation, ensuring that random operations are consistent.

We implemented a multi-GPU training system to increase the parallelism of NeRF evaluation in inverse rendering. Only the first GPU contains a Mitsuba 3 scene and performs surface rendering.

During light evaluation and gradient computation, the first GPU is responsible for distributing NeRF ray queries to other GPUs and collecting computation results, thereby improving computational parallelism.

5.3 Differentiable Shape and Material Optimization

Our differentiable surface rendering is based on the differentiable SDF by Vicini et al. [2022]. This implementation can handle visibility gradients arising from shape discontinuities to enable shape optimization. We represent the shape and material as voxel grids with a resolution of 256^3. We perform inverse rendering of 320 iterations. During each iteration, we randomly sample 6 images from the dataset for training. Following the coarse-to-fine optimization approach, we start optimization using 128^2 resolution images and 64^3 voxel grids, and progressive upscale the image to 512^2 and voxel grid to 256^3 at iterations 128 and 256. Following Vicini et al. [2022], we use Laplacian loss to maintain the smoothness of the geometry and material volumes, and redistance the SDF during each iteration. Both shape and material voxel grids and the NeRF network are optimized via the Adam optimizer [Kingma and Ba 2015], with a learning rate of 3e-3 for shape, 2e-2 for materials and 1e-2 for NeRF. We use 512 primal and 128 adjoint samples per pixel.

5.4 Data Capture Details

Our dataset possesses two features, high dynamic range and multiple rotation poses, requiring specialized techniques for handling. To acquire HDR captures, we use a Canon 5D Mark III camera with bracketed exposures. Each HDR image is synthesized from seven differently exposed photographs using HDRutils [Hanji and Mantiuk 2023; Hanji et al. 2020]. This allows the HDR images to accurately reflect the physical radiance of the environment. Leveraging the turntable, we capture four rotational poses for each object, each time rotating the object by 90 degrees. We capture multi-view images for each rotation, and solve their camera intrinsic and extrinsic parameters using Metashape. To align the four rotational poses, we affix two ChArUco boards during the capture, one on the turntable and the other beside the turntable to estimate the transformation of the object bounding box with respect to the scene. Inspired by works that jointly optimize NeRF and camera transformations [Lin et al. 2021], we jointly optimize the rotation transformations during NeRF pretraining stage to refine the estimated transformations. We assign an appearance embedding for each rotation pose in NeRF to enable modeling environment lighting variations when the object rotates.

6 EXPERIMENTS

The used datasets are first introduced, followed by the comparison with the environment map emitter in the task of inverse rendering and the evaluation of our emitter importance sampling.

6.1 Datasets

Following the aforementioned capture setup, we acquired four real-world data samples in indoor environment. We place a lamp beside the object to simulate non-distant lighting conditions. Each sample consists of four rotation poses, and each rotation pose comprises 50-80 HDR multi-view images. We employed a Konica Minolta Vivid 9i scanner to obtain the ground truth geometry. Following the real-world data capture setup, we synthesized four synthetic data samples using Mitsuba 3. For each set of synthetic data, we place an object in an indoor scene and render it from four different rotation poses. For each rotation pose we synthesize 50 multi-view images.

6.2 Comparisons with Environment Map

We compare our proposed NeRF emitter with the environment map emitter, which is the most commonly used emitter in the current inverse rendering works. We compare reconstruction quality obtained from inverse rendering using different emitters. As our method proposes an emitter module, it can generally be applied to any physics-based inverse rendering system. For fair comparison that only showcases the differences from the emitter, we implement the environment map baseline method also based on Differentiable SDF rendering [Vicini et al. 2022], which takes a known environment map as input and optimizes SDF geometry and material.

We adapt the environment map baseline method to work with our data. First, we need to provide an environment map input for the baseline. For synthetic data, we render environment maps using the ground truth scene in Mitsuba 3. For real data, we use the captured HDR images to train a NeRF, and render environment maps from the rendered NeRF. We place a virtual spherical camera at the center of the object of interest and render environment maps. When rendering the environment map, we retain the object of interest in the scene without removing it. Instead, we set the near plane of the spherical camera at the object's bounding box. This approach captures global illumination effects caused by the object in the rendered environment map while preventing the spherical camera from being occluded. We render an environment map for each rotation pose to capture the global illumination variations caused by object rotations.

Second, the rendered background does not match the input images when using environment maps because of the parallax effects. Therefore, for the synthetic dataset, we generate ground truth object masks and set the background pixels on the input images to black. During inverse rendering, the baseline renders the object only and hides the environment map, therefore generating masked rendered images, matching the background-processed input images. For the real dataset, we lend the background modeling ability of our NeRF emitter to the baseline. Specifically, we use NeRF to render a background image behind the object bounding box and a semi-transparent image modeling the occlusions (if any) in front of the object bounding box. These images are composited with the rendered object image to get the final image in the baseline.

Third, the baseline shares a similar coarse-to-fine optimization procedure with our method. We use the same NeRF extracted geometry as the shape initialization for the baseline. After these adaptations, the key difference between our method and the baseline remains in the light source modeling: the baseline computes surface shading using an environment map as an emitter, while our method represents spatially varying lighting using a NeRF.

Figure 4 shows the qualitative comparison results of the rendering and shape reconstruction between our method and the environment

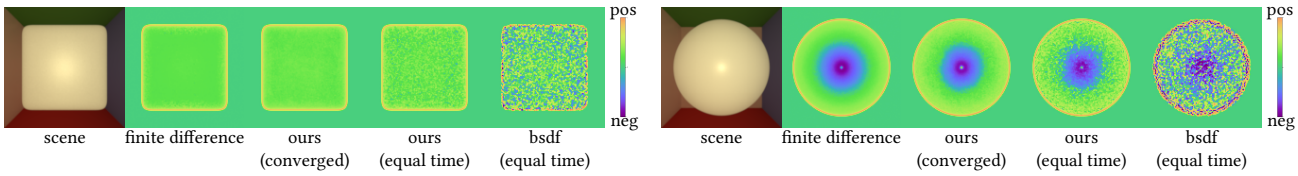

Fig. 3. Comparing the gradient image rendered by our emitter importance sampling and pure BSDF sampling.

Table 1. Quantitative Comparison with the environment map baseline on Novel View Synthesis, Relighting, and Shape Reconstruction.

Method	Novel View Synthesis			Relighting			Shape
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	CD
Ours	30.70	0.97	0.019	27.99	0.96	0.040	8.35e-6
Envmap Baseline	22.41	0.95	0.054	21.32	0.92	0.088	1.20e-4

map baseline on the synthetic datasets. We visualize rendered images in the original lighting, reconstructed shape, and relighting results in two novel lighting conditions. For the Head and Hotdog scene, we optimize the diffuse reflectance parameter using a diffuse BSDF following the work by Vicini et al. [2022]. In other scenes, we optimize the base color and roughness parameters using a principled BSDF material model. In the **Head** scene, the head casts shadows onto the shoulder and the chest. The baseline seems to match the cast shadows when rendering in the original lighting. However, the relighting results reveal local shape bumps. This is caused during inverse rendering to create local shadows because the shadows rendered by the environment map do not match the input images at shadow borders. The **Hotdog** scene visualizes inaccurate shadows rendered by the environment map, creating compromised rendering results in the original lighting. The artifacts are also visible during relighting and shape visualization. Since our method employs a more accurate light modeling using NeRF, the rendered shadows match the input images in the original lighting, and our relighting and shape results do not exhibit such issues. The **Teapot** scene exhibits glossy highlights. Due to the inaccuracy of the environment map, the rendered highlights in the baseline do not match the reference images in the original lighting. These lead to inaccurate material parameters and geometry bumps, visible in the relighting results. In contrast, our method can more closely match the glossy highlights, both in the original lighting and relighting results. The **Boar** scene is captured in a Cornell box, with a red wall on one side and a green wall on the other side. The baseline seems to render the illumination effects from the red wall on the boar's hind leg in the original lighting. However, the relighting results reveal baked illumination effects (green on the boar's face, red on the boar's hind leg). Our methods can both render the illumination effects in the original lighting and not bake the reflections in the relighting results.

In Table 1, we quantitatively compare our method with the environment map baseline on the synthetic datasets. We use the ground truth object masks to set background pixels to black when evaluating novel view synthesis and relighting, to more accurately reflect the quality of the object foreground and exclude the effect of the inaccurate background modeled by the environment map in the baseline. The PSNR metrics are computed and averaged only within the object foreground mask. Our method is consistently better than the environment map baseline on novel view synthesis, relighting, and shape reconstruction.

Figure 5 and 1 show the comparison results on the real-world data between our method and the environment map baseline. In the **Cabbage**, **Hamster** and **Dog** scenes, we can observe attached shadows in the original lighting. It can be seen in the relighting results that many attached shadows are baked into the baseline, causing dark artifacts. Compared to the baseline, our method generates more reasonable relighting results, indicating more accurate material reconstruction. The **RealChair** scene exhibits strong self-shadows. We can see the baseline renders inaccurately cast shadows in the original lighting. The mismatched shadow borders are also baked into the reconstructed materials, visible during relighting. Our method more accurately renders the cast shadows in the original lighting and produces cleaner relighting results. We visualize the shape errors as heatmaps in the insets. From the visualized shape and error maps, we can see that our reconstructed shape is more accurate than the baseline.

6.3 Ablation Study

We compare our proposed emitter importance sampling for NeRF with pure BSDF sampling as shown in Figure 3. In the first row, we put a cube in a Cornell box-like NeRF, which is trained on multi-view images rendered in a Cornell box. We visualize the gradient image with respect to the height of the cube. In the second row, we put a sphere in the same Cornell box-like NeRF and compute the gradient image with respect to the sphere radius. We can see that when using high samples per pixel (8192 spp), our rendered gradient images closely match the finite difference reference (65536 spp). When using the same samples per pixel (256 spp), our rendered gradient images have much lower variance than the gradient images rendered by pure BSDF sampling.

7 LIMITATION

The main drawback of using our proposed NeRF emitter is the increased computational cost in evaluating NeRF. With our multi-GPU implementation, our inverse rendering optimization takes about 4.5 hours on 8 RTX 4090 GPUs, while the baseline using the environment map completes in about one hour on one RTX 4090. However, accelerating NeRF is an active research topic and our method will benefit from the process in this direction, for example, recent works like Adaptive Shells [Wang et al. 2023a] can potentially accelerate our approach by increasing NeRF evaluation speed. Research progress that reduces the samples per pixel in inverse rendering like the works by Chang et al. [2023]; Nicolet et al. [2023]; Wang et al.

[2023b] may also accelerate our approach by reducing the number of NeRF evaluation rays.

Our emitter sampling method for NeRF does not consider that the radiance function $\mathbf{c}$ of NeRF is directional-dependent. This assumption works well under diffuse emitters, but can lead to increased rendering variance when the emitter is highly directional dependent. It also does not consider occlusion introduced by the surface to NeRF which should reduce the sampling probability of the occluded emitters. However, as an initial step in the emitter importance sampling on NeRF emitters in inverse rendering, our proposed method can achieve more accurate light modeling and effective light sampling in inverse rendering using NeRF as an emitter. Future work can explore this direction to further reduce the rendering variance when using NeRF as an emitter.

REFERENCES

Sai Praveen Bangaru, Michaël Gharbi, Fujun Luan, Tzu-Mao Li, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Frédo Durand. 2022. Differentiable Rendering of Neural SDFs through Reparameterization. In *SIGGRAPH Asia 2022 Conference Papers, SA 2022, Daegu, Republic of Korea, December 6-9, 2022*, Soon Ki Jung, Jehee Lee, and Adam W. Bargteil (Eds.). ACM, 5:1–5:9. https://doi.org/10.1145/3550469.3555397

Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020), 245:1–245:18. https://doi.org/10.1145/3414685.3417833

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 5460–5469. https://doi.org/10.1109/CVPR52688.2022.00539

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 19640–19648. https://doi.org/10.1109/ICCV51070.2023.01804

Sai Bi, Zexiang Xu, Pratul P. Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David J. Kriegman, and Ravi Ramamoorthi. 2020a. Neural Reflectance Fields for Appearance Acquisition. *CoRR* abs/2008.03824 (2020). arXiv:2008.03824 https://arxiv.org/abs/2008.03824

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David J. Kriegman, and Ravi Ramamoorthi. 2020b. Deep Reflectance Volumes: Relightable Reconstructions from Multi-view Photometric Images. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 12348)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 294–311. https://doi.org/10.1007/978-3-030-58580-8_18

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 *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 12664–12674. https://doi.org/10.1109/ICCV48922.2021.01245

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 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 10691–10704. https://proceedings.neurips.cc/paper/2021/hash/58ae749f25eded36f486bc85feb3f0ab-Abstract.html

Wesley Chang, Venkataram Sivaram, Derek Nowrouzezahrai, Toshiya Hachisuka, Ravi Ramamoorthi, and Tzu-Mao Li. 2023. Parameter-space ReSTIR for Differentiable and Inverse Rendering. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, Erik Brunvand, Alla Sheffer, and Michael Wimmer (Eds.). ACM, 18:1–18:10. https://doi.org/10.1145/3588432.3591512

Zhaoxi Chen and Ziwei Liu. 2022. Relighting4D: Neural Relightable Human from Videos. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIV (Lecture Notes in Computer Science, Vol. 13674)*, Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, 606–623. https://doi.org/10.1007/978-3-031-19781-9_35

Jorge Condor and Adrián Jarabo. 2022. A Learned Radiance-Field Representation for Complex Luminaires. In *33rd Eurographics Symposium on Rendering, EGSR 2022 - Symposium Track, Prague, Czech Republic, 4-6 July 2022*, Abhijeet Ghosh and Li-Yi Wei (Eds.). Eurographics Association, 49–58. https://doi.org/10.2312/SR.20221155

Paul E. Debevec and Jitendra Malik. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, Los Angeles, CA, USA, August 3-8, 1997*, G. Scott Owen, Turner Whitted, and Barbara Mones-Hattal (Eds.). ACM, 369–378. https://doi.org/10.1145/258734.258884

Param Hanji and Rafal K. Mantiuk. 2023. Robust Estimation of Exposure Ratios in Multi-Exposure Image Stacks. *IEEE Trans. Computational Imaging* 9 (2023), 721–731. https://doi.org/10.1109/TCI.2023.3301338

Param Hanji, Fangcheng Zhong, and Rafal K. Mantiuk. 2020. Noise-Aware Merging of High Dynamic Range Image Stacks Without Camera Calibration. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 12537)*, Adrien Bartoli and Andrea Fusiello (Eds.). Springer, 376–391. https://doi.org/10.1007/978-3-030-67070-2_23

Wenzel Jakob. 2012. Numerically stable sampling of the von Mises-Fisher distribution on S^2 (and other tricks). *Interactive Geometry Lab, ETH Zürich, Tech. Rep* (2012), 6.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. DR.JIT: a just-in-time compiler for differentiable rendering. *ACM Trans. Graph.* 41, 4 (2022), 124:1–124:19. https://doi.org/10.1145/3528223.3530099

Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. 2023. TensoIR: Tensorial Inverse Rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 165–174. https://doi.org/10.1109/CVPR52729.2023.00024

Ondrej Karlík, Martin Sik, Petr Vévoda, Tomás Skrivan, and Jaroslav Krivánek. 2019. MIS compensation: optimizing sampling techniques in multiple importance sampling. *ACM Trans. Graph.* 38, 6 (2019), 151:1–151:12. https://doi.org/10.1145/3355089.3356565

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6 (2018), 222. https://doi.org/10.1145/3272127.3275109

Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 5721–5731. https://doi.org/10.1109/ICCV48922.2021.00569

Jingwang Ling, Zhibo Wang, and Feng Xu. 2023. ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 175–185. https://doi.org/10.1109/CVPR52729.2023.00025

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 16169–16178. https://doi.org/10.1109/CVPR52688.2022.01571

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 *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12346)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 405–421. https://doi.org/10.1007/978-3-030-58452-8_24

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 41, 4 (2022), 102:1–102:15. https://doi.org/10.1145/3528223.3530127

Baptiste Nicolet, Fabrice Rousselle, Jan Novák, Alexander Keller, Wenzel Jakob, and Thomas Müller. 2023. Recursive Control Variates for Inverse Rendering. *ACM Trans. Graph.* 42, 4 (2023), 62:1–62:13. https://doi.org/10.1145/3592139

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 146. https://doi.org/10.1145/3386569.3392406

Julien Philip and Valentin Deschaintre. 2023. Radiance Field Gradient Scaling for Unbiased Near-Camera Training. *CoRR* abs/2305.02756 (2023). https://doi.org/10.48550/ARXIV.2305.02756 arXiv:2305.02756

Yi-Ling Qiao, Alexander Gao, Yiran Xu, Yue Feng, Jia-Bin Huang, and Ming C. Lin. 2023. Dynamic Mesh-Aware Radiance Fields. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 385–396. https://doi.org/10.1109/ICCV51070.2023.00042

Viktor Rudnev, Mohamed Elgharib, William A. P. Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. 2021. Neural Radiance Fields for Outdoor Scene Relighting. *CoRR* abs/2112.05140 (2021). arXiv:2112.05140 https://arxiv.org/abs/2112.05140

Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. 2021. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *IEEE Conference on Computer Vision and Pattern*

Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 7495–7504. https://doi.org/10.1109/CVPR46437.2021.00741

Cheng Sun, Guangyan Cai, Zhengqin Li, Kai Yan, Cheng Zhang, Carl Marshall, Jia-Bin Huang, Shuang Zhao, and Zhao Dong. 2023. Neural-PBIR Reconstruction of Shape, Material, and Illumination. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 18000–18010. https://doi.org/10.1109/ICCV51070.2023.01654

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, Justin Kerr, and Angjoo Kanazawa. 2023. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, Erik Brunvand, Alla Sheffer, and Michael Wimmer (Eds.). ACM, 72:1–72:12. https://doi.org/10.1145/3588432.3591516

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2022. Differentiable signed distance function rendering. *ACM Trans. Graph.* 41, 4 (2022), 125:1–125:18. https://doi.org/10.1145/3528223.3530139

Yu. M. Volin and G. M. Ostrovskii. 1985. Automatic computation of derivatives with the use of the multilevel differentiating technique—1. Algorithmic basis. *Computers & Mathematics With Applications* 11 (1985), 1099–1114. https://api.semanticscholar.org/CorpusID:120003155

Yu-Chen Wang, Chris Wyman, Lifan Wu, and Shuang Zhao. 2023b. Amortizing Samples in Physics-Based Inverse Rendering Using ReSTIR. *ACM Trans. Graph.* 42, 6 (2023), 214:1–214:17. https://doi.org/10.1145/3618331

Zian Wang, Tianchang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Müller, and Zan Gojcic. 2023a. Adaptive Shells for Efficient Neural Radiance Field Rendering. *ACM Trans. Graph.* 42, 6 (2023), 260:1–260:15. https://doi.org/10.1145/3618390

Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Aljaz Bozic, Dahua Lin, Michael Zollhöfer, and Christian Richardt. 2023a. VR-NeRF: High-Fidelity Virtualized Walkable Spaces. In *SIGGRAPH Asia 2023 Conference Papers, SA 2023, Sydney, NSW, Australia, December 12-15, 2023*, June Kim, Ming C. Lin, and Bernd Bickel (Eds.). ACM, 43:1–43:12. https://doi.org/10.1145/3610548.3618139

Peiyu Xu, Sai Praveen Bangaru, Tzu-Mao Li, and Shuang Zhao. 2023b. Warped-Area Reparameterization of Differential Path Integrals. *ACM Trans. Graph.* 42, 6 (2023), 213:1–213:18. https://doi.org/10.1145/3618330

Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. 2022a. PS-NeRF: Neural Inverse Rendering for Multi-view Photometric Stereo. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13661)*, Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, 266–284. https://doi.org/10.1007/978-3-031-19769-7_16

Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. 2022b. S^3-NeRF: Neural Reflectance Field from Shading and Shadow under a Single Viewpoint. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/0a630402ee92620dc2de3b704181de9b-Abstract-Conference.html

Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo estimators for differential light transport. *ACM Trans. Graph.* 40, 4 (2021), 78:1–78:16. https://doi.org/10.1145/3450626.3459807

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 143. https://doi.org/10.1145/3386569.3392383

Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A differential theory of radiative transfer. *ACM Trans. Graph.* 38, 6 (2019), 227:1–227:16. https://doi.org/10.1145/3355089.3356522

Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. 2022a. IRON: Inverse Rendering by Optimizing Neural SDFs and Materials from Photometric Images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 5555–5564. https://doi.org/10.1109/CVPR52688.2022.00548

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021. PhySG: Inverse Rendering With Spherical Gaussians for Physics-Based Material Editing and Relighting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 5453–5462. https://doi.org/10.1109/CVPR46437.2021.00541

Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. 2022b. Modeling Indirect Illumination for Inverse Rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 18622–18631. https://doi.org/10.1109/CVPR52688.2022.01809

Junqiu Zhu, Yaoyi Bai, Zilin Xu, Steve Bako, Edgar Velázquez-Armendáriz, Lu Wang, Pradeep Sen, Milos Hasan, and Ling-Qi Yan. 2021. Neural complex luminaires: representation and rendering. *ACM Trans. Graph.* 40, 4 (2021), 57:1–57:12. https://doi.org/10.1145/3450626.3459798

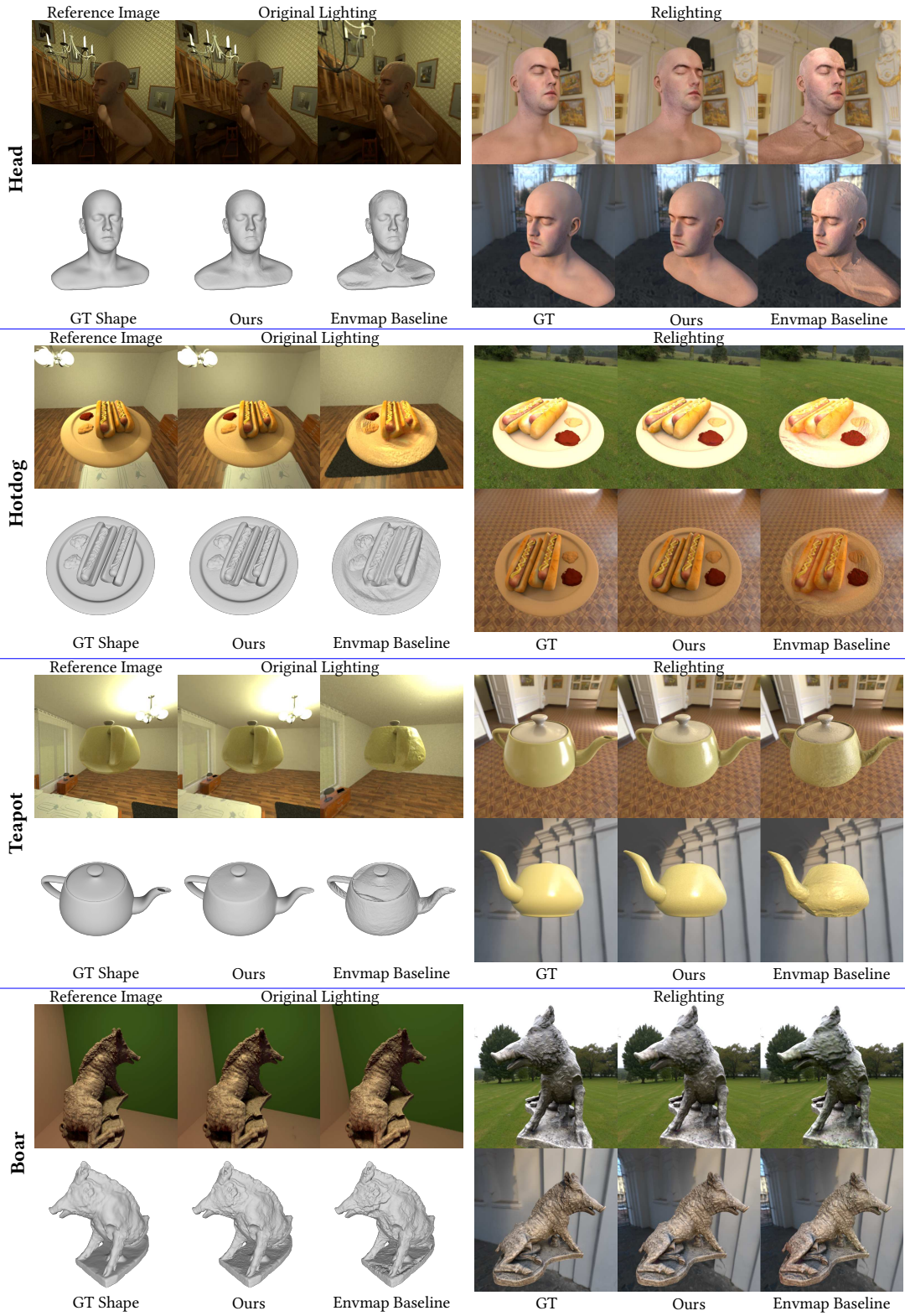

Fig. 4. Comparison with the environment map baseline on synthetic datasets.

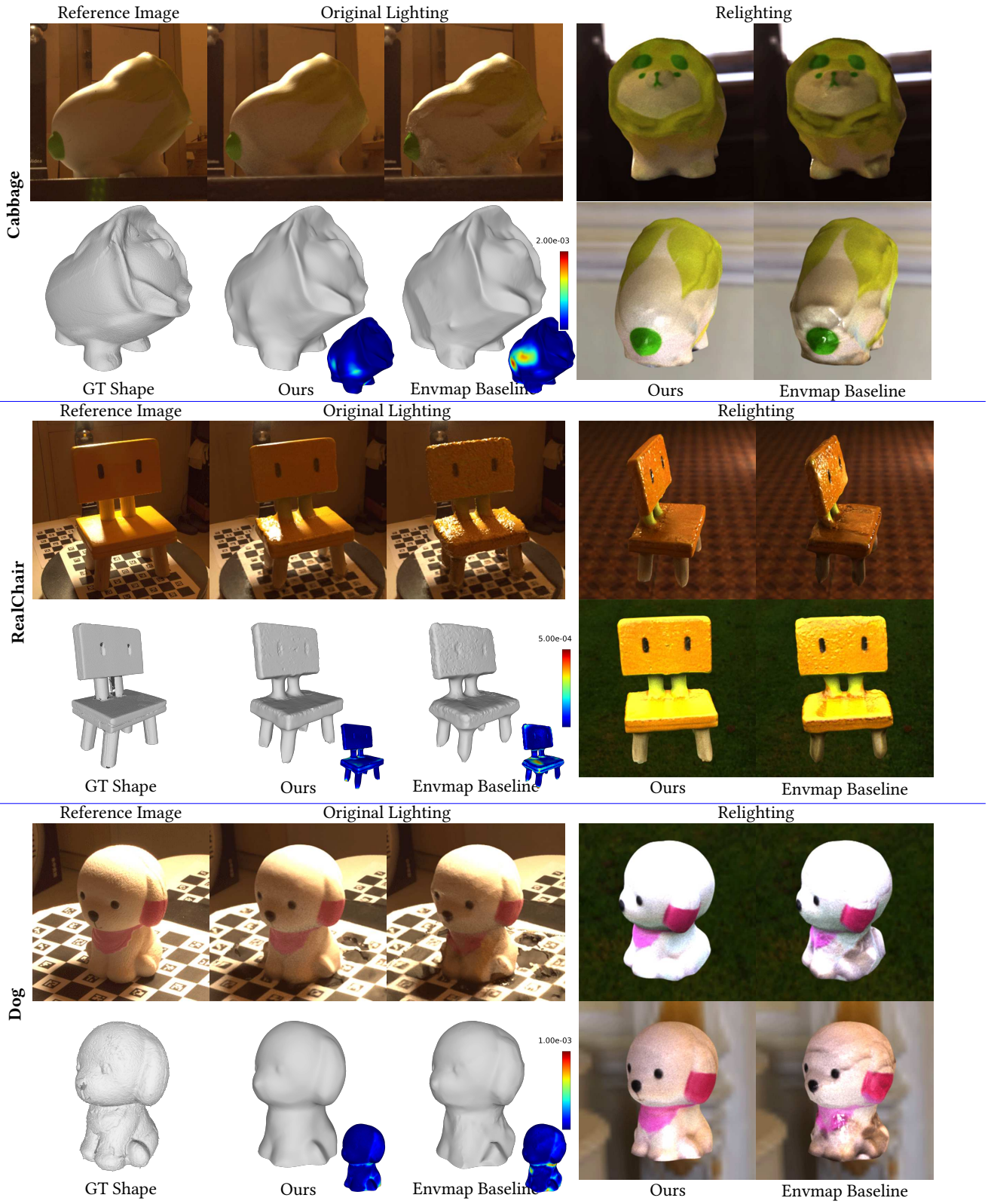

Fig. 5. Comparison with the environment map baseline on real datasets.