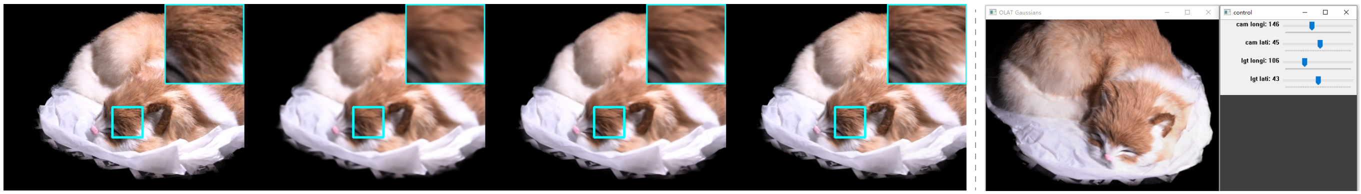
# OLAT Gaussians for Generic Relightable Appearance Acquisition

ZHIYI KUANG, State Key Laboratory of CAD & CG, Zhejiang University, China
YANCHAO YANG*, The University of Hong Kong (HKU), China
SIYAN DONG, The University of Hong Kong (HKU), China
JIAYUE MA, State Key Laboratory of CAD & CG, Zhejiang University, China
HONGBO FU, Hong Kong University of Science and Technology, China
YOUYI ZHENG*, State Key Laboratory of CAD & CG, Zhejiang University, China



cam longi: 146
cam lati: 45
lgt longi: 106
lgt lati: 43

Ground Truth DNL NRHints Ours Real-time Interaction

Fig. 1. Given multi-view one-light-at-a-time (OLAT) images illuminated by a moving point light, our OLAT Gaussians technique builds relightable representations for diverse objects. Besides achieving state-of-the-art rendering quality with more details in texture-rich areas than prior arts DNL [Gao et al. 2020] and NRHints [Zeng et al. 2023], our technique also supports real-time modification of views and lights.

One-light-at-a-time (OLAT) images sample a broader range of object appearance changes than images captured under constant lighting and are superior as input to object relighting. Although existing methods have produced reasonable relighting quality using OLAT images, they utilize surface-like representations, limiting their capacity to model volumetric objects, such as furs. Besides, their rendering process is time-consuming and still far from being used in real-time applications. To address these issues, we propose OLAT Gaussians to build relightable representations of objects from multi-view OLAT images. We build our pipeline on 3D Gaussian Splatting (3DGS), which achieves real-time high-quality rendering. To augment 3DGS with relighting capability, we assign each Gaussian a learnable feature vector, serving as an index to query the objects' appearance field. Specifically, we decompose the appearance field into an incident illumination function and a scattering function. The former accounts for light transmittance and foreshortening effects, while the latter represents the object's material properties to scatter light. Rather than using an off-the-shelf physically-based parametric rendering formulation, we model both functions using multi-layer perceptrons (MLPs). This makes our method suitable for various objects, e.g., opaque surfaces, semi-transparent volumes, furs, fabrics, etc. Given a camera view and a point light position, we compute each Gaussian's color as the product of the light intensity, the incident illumination value, and the scattering value, and then render the target image through the 3DGS rasterizer. To enhance rendering quality, we further utilize a proxy mesh to provide OLAT Gaussians with normals to improve highlights and visibility cues to improve shadows. Extensive experiments demonstrate that our method produces state-of-the-art rendering quality with significantly more details in texture-rich areas than previous methods. Our method also achieves real-time rendering, allowing users to interactively modify camera views and point light positions to get immediate rendering results, which are not available from the offline rendering of previous methods.

CCS Concepts: • **Computing methodologies** → Rendering; Relighting..

Additional Key Words and Phrases: 3D Gaussian Splatting, Scene Relighting, Appearance Acquisition

**ACM Reference Format:**
Zhiyi Kuang, Yanchao Yang, Siyan Dong, Jiayue Ma, Hongbo Fu, and Youyi Zheng. 2024. OLAT Gaussians for Generic Relightable Appearance Acquisition. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 03–06, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687671

## 1 Introduction

Object relighting is a long-standing goal in computer graphics and has wide applications, such as creating digital doubles of cultural relics and showcasing commodities online. Since the appearances of real-world objects result from the complex interactions of their geometry, material properties, and environmental lighting, it is challenging to disentangle only the objects' attributes, making relighting a highly ill-posed problem.

One branch of existing relighting methods reconstructs object appearances from multi-view images captured under constant environmental lighting. It allows casual capturing processes (such as recording a video surrounding the objects simply using a cellphone camera) but only captures a limited range of appearance changes.

*Corresponding authors

Authors' Contact Information: Zhiyi Kuang, State Key Laboratory of CAD & CG, Zhejiang University, China, 12021227@zju.edu.cn; Yanchao Yang, The University of Hong Kong (HKU), China, yanchaoy@hku.hk; Siyan Dong, The University of Hong Kong (HKU), China, siyandong.3@gmail.com; Jiayue Ma, State Key Laboratory of CAD & CG, Zhejiang University, China, 22221306@zju.edu.cn; Hongbo Fu, Hong Kong University of Science and Technology, China, hongbofu@ust.hk; Youyi Zheng, State Key Laboratory of CAD & CG, Zhejiang University, China, youyizheng@zju.edu.cn.

*SA Conference Papers '24, December 03–06, 2024, Tokyo, Japan*

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687671

Another branch utilizes well-designed light patterns to sample a wider range of appearance changes and achieves more accurate reconstruction than methods with constant lighting. Following the latter style, we choose Deferred Neural Lighting [Gao et al. 2020]'s OLAT capturing setup, which consists of a handheld camera to capture multi-view images and a single moving point light source to light the objects from different positions. This lightweight setup provides rich lighting information while maintaining a convenient capturing process without heavy equipment.

Existing methods have produced plausible relighting results by representing objects using meshes [Chen et al. 2021; Hasselgren et al. 2022; Munkberg et al. 2022], neural implicit functions [Boss et al. 2021a,b; Liang et al. 2023a; Liu et al. 2023; Zhang et al. 2021b,c], and 3D Gaussians [Jiang et al. 2023; Liang et al. 2023b]. However, most of them rely on specific shading functions formulated by a sparse set of parameters, such as the prevailing microfacet material model [Burley and Studios 2012; Walter et al. 2007], limiting their usage mainly in objects of solid opaque surfaces. Recently, NRHints [Zeng et al. 2023] leverages an MLP network to model the rendering process and performs well on various objects. Nevertheless, since NRHints utilizes NeuS [Wang et al. 2021]'s SDF surface representation, it produces blurry artifacts for extremely thin structures (such as furs and fabrics). Moreover, the volume rendering of vanilla NeuS takes minutes to render a single frame, which is still far from efficient usage.

To this end, we introduce OLAT Gaussians to build relightable representations of objects and render relighted images with high-fidelity details and real-time performance, as shown in Fig. 1. With captured OLAT images, we first build a proxy mesh using Instant-NeuS+, which we implement on Instant-NSR-pl [Guo 2022] to reconstruct object geometry from OLAT images. The proxy mesh provides OLAT Gaussians with vertex samples for shape initialization, depth-to-light for shadow mapping, and reference normals. After initialization, we assign each Gaussian a learnable feature vector to query the objects' appearance field, which is decomposed into incident illumination and scattering functions. We model both functions using trainable MLPs rather than a fixed rendering formulation to represent diverse materials. To aid OLAT Gaussians with geometry information, we condition the incident illumination function with the light visibility from shadow mapping, accounting for reasonable occlusion. Furthermore, we assign each Gaussian a learnable normal (supervised by mesh normals) to condition the scattering function to improve specular highlights. In addition, we explicitly attenuate the light intensity by a physically correct inverse square rule. Finally, at a given camera view with a point light position, we compute each Gaussian's color as the product of the light intensity, the incident illumination value, and the scattering value, and then use the 3DGS rasterizer to produce the relighted image.

Experiments show that our decomposed MLP formulation is more versatile in representing various materials than the microfacet material model and produces more accurate results than feeding all inputs to a single network without decomposition. As a result, our method achieves state-of-the-art rendering quality and efficiency on diverse real-world objects, including opaque surfaces, semi-transparent volumes, furs, and fabrics.

To sum up, our main contributions are:

- We introduce OLAT Gaussians, a novel relightable representation that combines 3DGS with a well-designed decomposed MLP formulation and is applicable for diverse objects.
- We leverage a proxy mesh to assist OLAT Gaussians with normals to improve highlights and visibility cues to improve shadows.
- Our method achieves state-of-the-art rendering quality with significantly more details in texture-rich areas than previous methods, and real-time rendering performance.

## 2 Related Works

With profound advances in deep learning techniques, several differentiable neural architectures have been proposed to represent object appearance. We summarize representative works in image-based relighting and relightable 3D representations, including meshes, neural implicit functions, and 3D Gaussians. Please refer to comprehensive surveys [Dong 2019; Einabadi et al. 2021; Tewari et al. 2022] for more research in relighting object appearance.

*Image-based Relighting.* Deep learning has been introduced to generate scene appearances under different lighting conditions [Bemana et al. 2020; Debevec et al. 2000; Ren et al. 2015; Xu et al. 2018] from a collection of input images. Philip et al. [2019] first build a proxy mesh using multi-view stereo (MVS) and then use it to generate shadow maps and illumination buffers, which are fed to a convolutional neural network (CNN) to yield relighted images. Their model is trained on synthetic data but generates reasonable results on real-world data. Deferred Neural Lighting (DNL) [Gao et al. 2020] also utilizes a CNN as a renderer, which takes as input the projected textures and radiance cues rendered from a rough mesh with neural textures on each vertex and then produces rendering results. DNL's model typically requires 10,000 images to train. Neural Light Transport (NLT) [Zhang et al. 2021a] computes diffuse components using the Lambertian model and predicts non-diffuse components using a CNN conditioned by the non-diffuse residuals from nearby views. NLT's model requires 8,000 images captured by a light stage to train. Philip et al. [2021] extract source irradiance and mirror reflection from captured images as illumination cues and then feed both source and target illumination cues (specified by users) to a pretrained neural renderer to produce relighted images.

These image-based relighting methods define appearance attributes on meshes, which represent surface-like objects well but fall short of describing volumetric objects, such as furs and fabrics. Moreover, they rely on image-space renderers (mainly CNNs) and thus cannot optimize camera calibration errors common for real-world multi-view images. Instead of relying on meshes, we employ the 3DGS representation, which is more versatile in representing diverse objects and is able to jointly optimize camera poses through its differentiable rasterizer.

*Mesh Representation.* With rapid development in differentiable 3D renderers, the mesh-based representation is naturally adopted for inverse rendering, which optimizes objects' shape and material properties by backpropagating the losses between rendered and real images. DIB-R++ [Chen et al. 2021] starts with a given triangular mesh, then employs a rasterization-based renderer to obtain material maps, and finally computes shading results by sampling lighting in a deferred stage. Munkberg et al. [2022] utilize differentiable marching tetrahedrons [Shen et al. 2021] as the geometry representation and jointly optimize a triangular mesh, materials, and environmental lighting approximated by split sum. To improve material and light separation, Hasselgren et al. [2022] augment [Munkberg et al. 2022] by leveraging a Monte Carlo (MC) ray-tracing renderer and a neural denoiser to compensate for noises caused by MC sampling. Luan et al. [2021] also use an MC renderer and extend its usage to images illuminated by a light co-located with the camera view. Cai et al. [2022] leverage differentiable iso-surface extraction to convert implicit geometries into explicit ones for rendering to reproduce various light transport effects. Although reconstructed meshes are convenient to use directly in graphics rendering engines, they are limited to representing only solid opaque surfaces, as discussed previously. While using 3DGS to represent various objects, we also take advantage of mesh geometry to provide normals and visibility cues, thus improving our rendering quality further.

*Neural Implicit Representation.* Neural Radiance Fields (NeRF) [Mildenhall et al. 2021] has gained great success in novel view synthesis. It represents scenes as implicit radiance fields modeled by MLPs and employs volume rendering with two-pass sampling to generate realistic images at novel views. Since NeRF does not rely on explicit geometries, its implicit representation demonstrates exceptional versatility for representing a wide variety of scenes. Extensive research [Boss et al. 2021a, 2022, 2021b; Kuang et al. 2022; Li et al. 2022; Martin-Brualla et al. 2021; Rudnev et al. 2022; Srinivasan et al. 2021; Sun et al. 2021; Verbin et al. 2022; Xu et al. 2023; Yang et al. 2023, 2022; Yao et al. 2022; Zhang et al. 2021c; Zheng et al. 2021] extends NeRF with relighting capability by decomposing the radiance fields into material fields and environmental light. One common limitation of NeRF-based relighting methods is the coarse geometry modeled by an implicit density field, which lacks accurate occlusions and thus struggles to produce correct shadows. To mitigate this issue, LitNeRF [Sarkar et al. 2023] utilizes a proxy mesh generated by MVS to provide visibility cues and produces more reasonable shadows with sparse view images. Our 3DGS-based representation possesses comparable versatility of reproducing object appearance to NeRF and achieves better rendering efficiency by forwarding rendering, which is free of sampling passes.

An alternative to NeRF's density field is the neural signed distance function (SDF) [Park et al. 2019; Wang et al. 2021; Yariv et al. 2021]. NeuS [Wang et al. 2021] represents a surface as the zero-level set of an SDF and produces fine-grained details via unbiased and occlusion-aware sampling. [Fu et al. 2022; Li et al. 2023b; Wang et al. 2024; Zhu et al. 2023] use additional regularizations and training strategies to further improve the SDF geometry. SDF-based methods [Liang et al. 2023a; Liu et al. 2023; Lyu et al. 2022; Zhang et al. 2022a, 2021b, 2022b] regularize objects' shapes well and show promising relighting results. NRHints [Zeng et al. 2023] applies an SDF representation to OLAT images captured under [Gao et al. 2020]'s setup and produces accurate highlights and shadows with additional radiance cues from the underlying geometry. However, the SDF representation assumes a surface-like geometry, leading to sub-optimal recovered details for extremely thin structures such as furs and fabrics. In addition, the rendering efficiency of neural implicit representations suffers from the time-consuming volume rendering process. Although acceleration techniques [Fridovich-Keil et al. 2022; Müller et al. 2022; Wang et al. 2023] for NeRF have been proposed, they have not been widely utilized to build relightable scenes with real-time rendering performance. On the other hand, our method faithfully reproduces object details without the limitation of surface geometry assumption and achieves real-time rendering.

*3D Gaussian Representation.* 3DGS [Kerbl et al. 2023] innovatively employs an explicit 3D Gaussian representation and implements a highly efficient rasterizer to achieve real-time high-quality rendering. GS-IR [Liang et al. 2023b] and R3DG [Gao et al. 2023] achieve real-time relighting rendering from images captured under constant lighting. GaussianShader [Jiang et al. 2023] enhances 3DGS's relighting quality on reflective surfaces. We introduce 3DGS to build relightable representations from OLAT images to model accurate appearance changes. Without assumptions on the underlying geometry, our method recovers fine-grained details for various objects, including semi-transparent volumes, furs, and fabrics. We also utilize 3DGS's efficient rasterizer to achieve real-time rendering.

## 3 Methodology

With captured multi-view OLAT images where a moving point light is the only emissive light source, we aim to design a generic 3DGS architecture for learning relightable representations of objects. We do not make any assumption on objects' geometry or material, which would limit our model's applicability. Therefore, we model the appearance field of objects using MLPs rather than a sparse set of material parameters.

On the other hand, 3DGS intrinsically does not guarantee accurate geometry and thus falls short of producing accurate shadows and highlights. To this end, we aid our model with geometry hints, including occlusions and normals computed from a proxy mesh. As a result, OLAT Gaussians reproduce high-fidelity details with real-time rendering performance based on 3DGS's highly efficient forward rendering.

### 3.1 Preliminary

*3D Gaussian Splatting.* 3DGS represents a scene by point clouds, with each point standing for the center (mean) $\boldsymbol{\mu}$ of a 3D Gaussian distribution:

$$G(\boldsymbol{x}|\boldsymbol{\mu},\Sigma) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{1}$$

where $\Sigma \in R^{3x3}$, an anisotropic covariance matrix, defines each Gaussian's shape. Each point is assigned a set of spherical harmonics (SH) coefficients to represent view-dependent colors and an opacity value $\alpha$ for accumulated volume rendering. During rendering, each 3D Gaussian is projected to 2D image space and rasterized to contribute to pixels that it covers. The final rendered image is the accumulated results of all visible Gaussians' contributions. We refer to their paper [Kerbl et al. 2023] for technical details.

### 3.2 Proxy Mesh

As shown in Fig. 2, we start by reconstructing a proxy mesh from OLAT images. Based on Instant-NSR-pl [Guo 2022], a fast NeuS

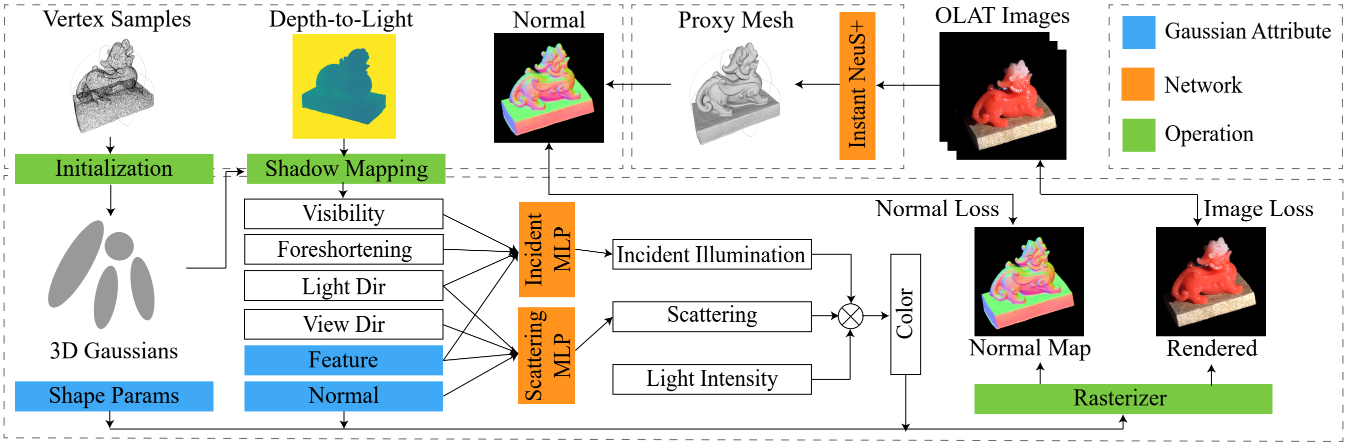


Fig. 2. Pipeline overview. With captured multi-view OLAT images, we first reconstruct a proxy mesh using Instant-NeuS+, a fast NeuS implementation. Next, vertices are sampled from the proxy mesh to initialize 3D Gaussians' shapes. We also use the proxy mesh to render depth-to-light and reference normal maps. The depth-to-light map is then used to provide light visibility through shadow mapping. Besides shape attributes, each Gaussian is assigned a learnable feature vector and a normal to condition the appearance field, which is decomposed into incident illumination and scattering functions. The color of each Gaussian is computed as the product of the light intensity, the incident illumination value, and the scattering value. Finally, we utilize 3DGS's rasterizer to render an image supervised by the ground truth. We also render a normal map supervised by the reference normal map to regularize the Gaussians' normals.

implementation with multiresolution hash encoding [Müller et al. 2022] and efficient sampling [Li et al. 2023a], we implement Instant-NeuS+ by additionally conditioning the color MLP with the incident light direction $\boldsymbol{w}_\mathrm{i}$, light distance $d$, visibility hints $v_\mathrm{cue}$, and highlight hints $h_\mathrm{cue}$ (borrowed from [Zeng et al. 2023]) to help solve the ambiguity between geometry and material:

$$\boldsymbol{c} = F_\mathrm{color}(\boldsymbol{f}_\mathrm{sdf}, \boldsymbol{n}_\mathrm{sdf}, \boldsymbol{w}_\mathrm{o}, \boldsymbol{w}_\mathrm{i}, d, v_\mathrm{cue}, h_\mathrm{cue}), \tag{2}$$

where $\boldsymbol{w}_\mathrm{o}$ is the view direction, $\boldsymbol{n}_\mathrm{sdf}$ is the normal, and $\boldsymbol{f}_\mathrm{sdf}$ is the feature vector produced by the SDF MLP of Instant-NSR-pl. The proxy mesh is extracted from the SDF MLP by Marching Cubes [Lorensen and Cline 1998].

Although other mesh reconstruction methods may also suffice, we find Instant-NeuS+ generally yields meshes of reasonable quality in efficient training time with no extra data or capturing process. Thus, we use it for our pipeline.

With the proxy mesh, we randomly sample $M$ vertices from it to initialize the 3D Gaussians' shapes. In addition to shape attributes (position $\boldsymbol{p}$, covariance $\Sigma$, and opacity $\boldsymbol{\alpha}$), we assign each Gaussian a learnable feature vector $\boldsymbol{f}_\mathrm{g}$ (initialized randomly) and a learnable normal $\boldsymbol{n}_\mathrm{g}$ (initialized as the mesh vertex normal), serving as input to the appearance field.

### 3.3 Decomposed MLP Shading

Instead of modeling the appearance field as one union MLP that accepts all inputs and decides the final appearance as in NRHints [Zeng et al. 2023], we decompose the appearance field into incident illumination and scattering MLPs. This decomposition helps these two functions focus on representing their own parts and achieves better brightness accuracy than one union MLP (see Fig. 7 and Tab. 4).

To render an image at a given camera view lit by a point light, we composite each Gaussian's observed color as the product of the attenuated light intensity (a point light is denoted as its position $\boldsymbol{x}_\mathrm{l}$ and intensity $L_\mathrm{e}$), the received incident illumination value, and the scattering value. In the following, we elaborate on each component.

*Light Intensity.* Following the physically correct inverse-square law, the light intensity arriving at each Gaussian without occlusion is computed as:

$$L_\mathrm{int} = \frac{L_\mathrm{e}}{\|\boldsymbol{p} - \boldsymbol{x}_\mathrm{l}\|^2}. \tag{3}$$

*Incident Illumination Function.* This term stands for a Gaussian's received portion of unit light intensity, accounting for both light transmittance and foreshortening effects. To make this term geometry-aware, we decide each Gaussian's visibility $V(\boldsymbol{p}, \boldsymbol{w}_\mathrm{i}) \in \{0, 1\}$ ($\boldsymbol{w}_\mathrm{i} = \frac{\boldsymbol{x}_\mathrm{l} - \boldsymbol{p}}{\|\boldsymbol{x}_\mathrm{l} - \boldsymbol{p}\|}$ is the incident light direction) to the point light by shadow mapping: we first render a depth map of the proxy mesh at the point light view and then regard a Gaussian as visible if its distance to the point light is not greater than the observed depth (plus a small bias to reduce shadow acne) in the depth map. In addition, the foreshortening effect $max(\boldsymbol{n}_\mathrm{g} \cdot \boldsymbol{w}_\mathrm{i}, 0)$ is also taken into account. The incident illumination function is formulated as follows:

$$\boldsymbol{f}_\mathrm{i} = F_\mathrm{incident}(\boldsymbol{w}_\mathrm{i}, \boldsymbol{f}_\mathrm{g}, V(\boldsymbol{p}, \boldsymbol{w}_\mathrm{i}) max(\boldsymbol{n}_\mathrm{g} \cdot \boldsymbol{w}_\mathrm{i}, 0)), \tag{4}$$

The resulting incident illumination value $\boldsymbol{f}_\mathrm{i} \in [0, 1]$ is a scalar incorporating approximately non-tinted view-independent effects, including light transmittance without explicit supervision.

*Scattering Function.* To describe material properties to scatter light, the scattering function takes the incident light direction $\boldsymbol{w}_\mathrm{i}$, the view direction $\boldsymbol{w}_\mathrm{o}$, the Gaussian feature $\boldsymbol{f}_\mathrm{g}$, and the Gaussian normal $\boldsymbol{n}_{\mathrm{g}}$ as input, and then predicts the scattering value $\boldsymbol{f}_{\mathrm{s}} \in R^3$

$$\boldsymbol{f}_{\mathrm{s}} = F_{\mathrm{scatter}}(\boldsymbol{w}_{\mathrm{i}}, \boldsymbol{w}_{\mathrm{o}}, \boldsymbol{f}_{\mathrm{g}}, \boldsymbol{n}_{\mathrm{g}}). \quad (5)$$

Note that the scattering function does not conform to a fixed reflection rule and is able to incorporate both direct and indirect lighting to describe various materials.

With the above components, each Gaussian's observed color $\boldsymbol{c}_{\mathrm{g}}$ is computed as

$$\boldsymbol{c}_{\mathrm{g}} = L_{\mathrm{int}} \cdot \boldsymbol{f}_{\mathrm{i}} \cdot \boldsymbol{f}_{\mathrm{s}}. \quad (6)$$

Finally, we project 3D Gaussians to 2D and use 3DGS's rasterizer to volume-render all visible Gaussians to produce an image $\boldsymbol{I}_{\mathrm{g}}$ (with sRGB gamma correction applied) and supervise it by the ground truth image $\boldsymbol{I}^*$.

## 3.4 Training

*Normal Regularization.* To ensure Gaussian normals match local geometry, we use 3DGS's rasterizer to render Gaussian normals to a normal map $\boldsymbol{N}_{\mathrm{g}}$, which is then supervised by the reference mesh normal map $\boldsymbol{N}_{\mathrm{ref}}$.

*Loss.* We jointly train each Gaussian's attributes $\{\boldsymbol{p}, \Sigma, \alpha, \boldsymbol{f}_{\mathrm{g}}, \boldsymbol{n}_{\mathrm{g}}\}$, and the parameters of two global functions $F_{\mathrm{incident}}$ and $F_{\mathrm{scatter}}$ using losses of images and normal maps. The loss function of images is defined the same as 3DGS (L1 loss and a D-SSIM term), and the loss function of normal maps is L1. The total loss function is formulated as follows:

$$L = (1 - \lambda_1)L_1(\boldsymbol{I}_{\mathrm{g}}, \boldsymbol{I}^*) + \lambda_1 L_{D-SSIM}(\boldsymbol{I}_{\mathrm{g}}, \boldsymbol{I}^*) + \lambda_2 L_1(\boldsymbol{N}_{\mathrm{g}}, \boldsymbol{N}_{\mathrm{ref}}), \quad (7)$$

where we set $\lambda_1$=0.2 and $\lambda_2$=0.1 for all our experiments. We set 1e-3 as the learning rate of $\{\boldsymbol{f}_{\mathrm{g}}, \boldsymbol{n}_{\mathrm{g}}\}$, $F_{\mathrm{scatter}}$, and $F_{\mathrm{incident}}$. The learning rate and the schedule of $\{\boldsymbol{p}, \Sigma, \boldsymbol{\alpha}\}$ are the same as those of the 3DGS project. More implementation details are provided in our supplementary document.

## 4 Experiments

We run our method on 7 captured real-world OLAT scenes (4 from DNL [Gao et al. 2020] and 3 from NRHints [Zeng et al. 2023]). These scenes contain a wide variety of objects, including opaque surfaces, semi-transparent volumes, furs, and fabrics. Reproducing appearances from these scenes is challenging and requires the scene representation to be versatile and robust. Note that although DNL provides ~10,000 training images, NRHints has achieved comparable results with only 500~1,000 images as input. We use the same training sets as NRHints for an efficient training process. On the other hand, for a fair comparison, we use the same test sets (50-300 views of each case) for all methods.

## 4.1 Comparisons

We compare our method with DNL [Gao et al. 2020] and NRHints [Zeng et al. 2023], which produce state-of-the-art quality for OLAT object relighting, among methods using image-space renderers and methods using 3D differentiable representations, respectively. We use NRHints' pretrained models to generate its result images, while DNL's result images are provided. For cases from DNL, we crop out the center 512x512 resolution areas from their original 1024x1024 images to focus on the subjects since most of the rest areas are pure black. For cases from NRHints, we use the full 1024x1024 resolution. (An exception: The provided DNL's results on case *Cat* were trained on downscaled ground truth. Since the original ground truth is available, DNL's results on *Cat* are upscaled for comparisons.)

DNL produces accurate results with ~10,000 input images using 13 partition renderers, each trained on a subset of view directions. As reported in Tab. 1, though slightly lower on PSNR, our method achieves better SSIM and LPIPS than DNL using only 500~1,000 input images. As shown in Fig. 3, while visually close, our results possess more high-frequency details for furs (1st row), fabrics (3rd row), and dotted highlights (2nd and 4th rows). Moreover, since DNL utilizes image-space renderers that are intrinsically not differentiable to camera poses, it embeds camera calibration errors in each partition renderer. This results in "wobbling" artifacts during view moving when the renderers try to reproduce an incorrect camera pose. On the contrary, based on the 3D representation, our method can jointly optimize camera poses for higher accuracy, which is detailed in our supplementary document.

NRHints achieves comparable results to DNL while significantly reducing the number of the required training images to 500~1,000. Based on the NeuS representation, NRHints' results are reasonable but tend to be over-smooth due to the assumption of a solid surface. As shown in Fig. 3 and 4, our results demonstrate more faithful details in texture-rich areas than those of NRHints. As reported in Tab. 1 and 2, our method significantly improves LPIPS and also achieves better PSNR and SSIM than NRHints.

Besides better rendering quality, as reported in Tab. 3, our method trains a relightable model in around 1 hour, which is an order of magnitude less than DNL and NRHints. Furthermore, our method achieves real-time rendering with the highly efficient 3DGS rasterizer, while previous methods only support offline rendering.

## 4.2 Ablation Studies

*Normals.* Due to the discrete nature of 3DGS, it inherently cannot derive accurate normals that participate in the interaction between object materials and environmental lights, especially for surface-like objects. As shown in Fig. 5 and Tab. 4, our model produces inferior specular highlights without the additional normal of each Gaussian, proving that normals assist the scattering MLP in learning physically correct reflection.

*Visibility.* Shadows are caused by self-occlusion determined by whether the point light is visible to Gaussians. As shown in Fig. 5 and Tab. 4, our model fails to reproduce correct shadows without the visibility from shadow mapping, indicating that the incident illumination MLP successfully leverages the visibility to produce reasonable occluded shadows.

*Shading Formulations.* We validate our design choice of the color shading formulation by comparing it with two substituting solutions:

(1) Parametric PBR Shading. Similar to existing 3DGS-based relighting frameworks [Jiang et al. 2023; Liang et al. 2023b] for constant lighting, this formulation further decomposes the scattering function into per-Gaussian material attributes

Table 1. Quantitative comparisons on the 4 cases from DNL. Best and second-best metrics are highlighted. Our method achieves comparable PSNR with DNL and the best SSIM and LPIPS for all cases.

| | *Cat* | | | *Fish* | | | *Cluttered* | | | *Pixiu* | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| DNL [Gao et al. 2020] | 31.84 | 0.901 | 0.164 | 32.48 | 0.922 | 0.098 | 34.19 | 0.959 | 0.069 | 34.90 | 0.951 | 0.075 | 33.35 | 0.933 | 0.102 |
| NRHints [Zeng et al. 2023] | 31.21 | 0.900 | 0.152 | 31.01 | 0.909 | 0.112 | 33.76 | 0.959 | 0.072 | 33.68 | 0.948 | 0.079 | 32.42 | 0.929 | 0.104 |
| Ours | 31.94 | 0.935 | 0.099 | 31.92 | 0.933 | 0.072 | 34.59 | 0.971 | 0.046 | 34.18 | 0.953 | 0.067 | 33.15 | 0.948 | 0.071 |

Table 2. Quantitative comparisons on the 3 cases from NRHints. Better metrics are in **boldface**. Our method constantly outperforms NRHints on all metrics.

| | *Cat on Decor* | | | *Cup and Fabric* | | | *Pikachu* | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| NRHints [Zeng et al. 2023] | 36.061 | 0.976 | 0.070 | 37.555 | 0.981 | 0.055 | 34.464 | 0.969 | 0.066 | 36.023 | 0.975 | 0.064 |
| Ours | **36.537** | **0.981** | **0.054** | **38.646** | **0.986** | **0.043** | **34.963** | **0.973** | **0.061** | **36.715** | **0.980** | **0.053** |

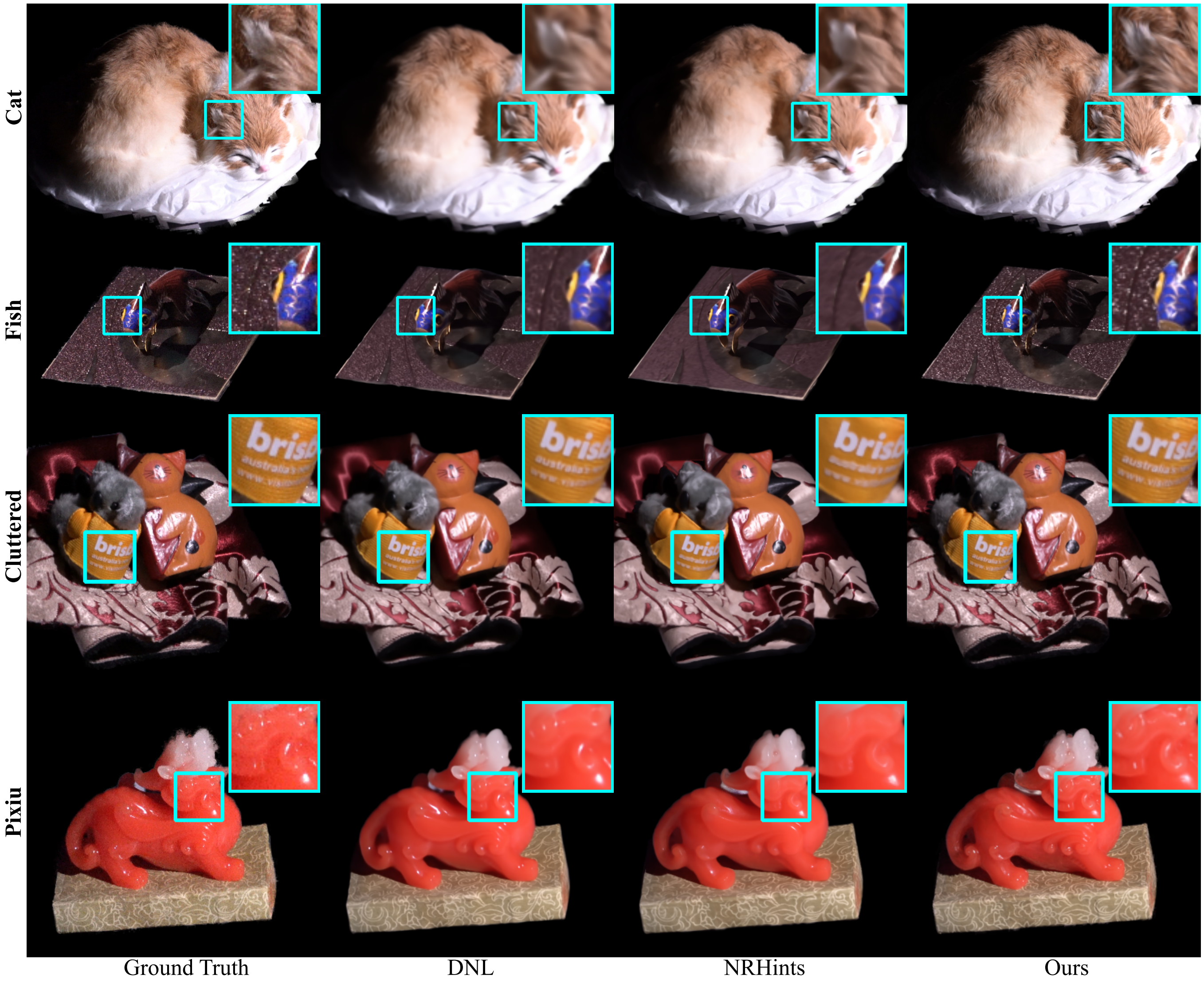


Fig. 3. Qualitative comparisons with DNL [Gao et al. 2020] and NRHints [Zeng et al. 2023], tested on the 4 real captured scenes from DNL. Our method produces more faithful details than DNL and NRHints, especially in texture-rich areas such as furs, fabrics, and dotted highlights.

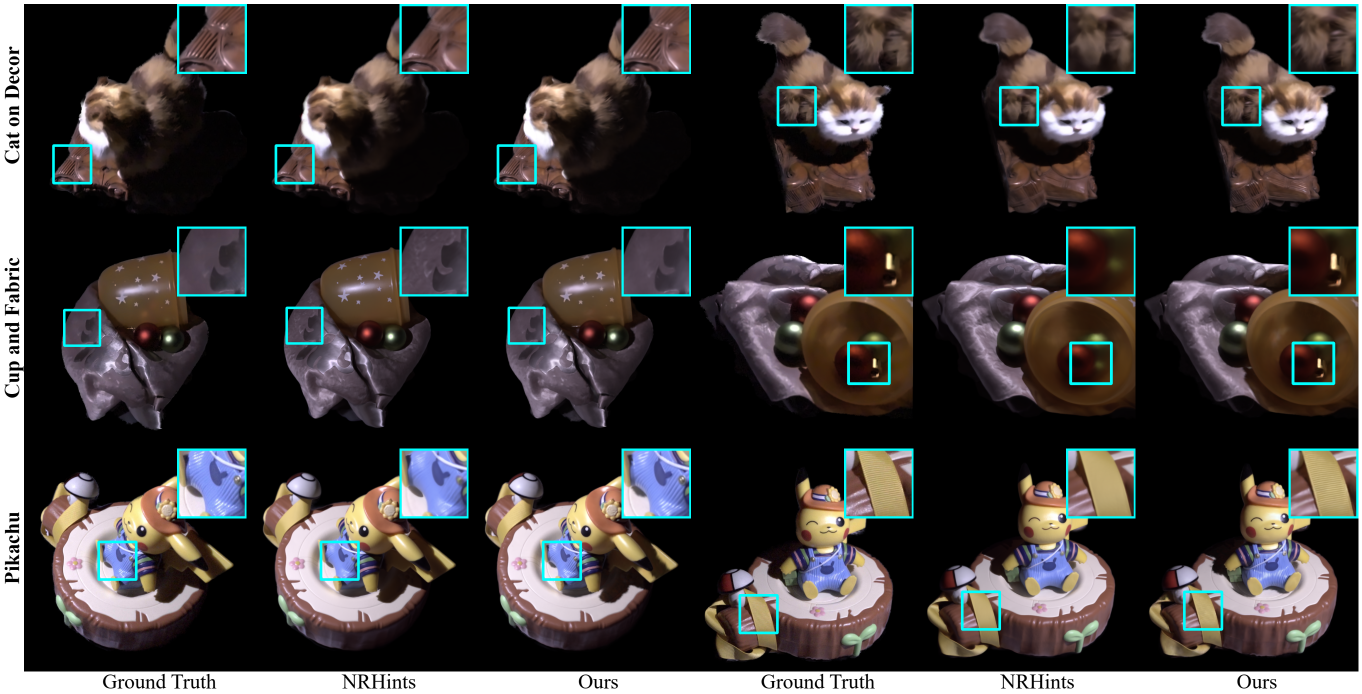


Fig. 4. Qualitative comparisons with NRHints [Zeng et al. 2023], tested on the 3 real captured scenes from NRHints. Our results possess more high-frequency details, especially for fine-grained textures and thin structures.

Table 3. Required resources and performance of the compared methods. Our method achieves real-time rendering with an order of magnitude fewer input images than DNL and an order of magnitude less training time than NRHints.

| | Input Images | Training (GPUs) | Inference |
|---|---|---|---|
| DNL [Gao et al. 2020] | ~10,000 | ~20h (13) | 1~2s |
| NRHints [Zeng et al. 2023] | 500~1,000 | ~20h (4) | ~1min |
| Ours | 500~1,000 | ~1h (1) | ~30ms |

Table 4. Quantitative results of the ablation study. Metrics are averaged from all 7 scenes. Our method achieves better rendering quality with normals and visibility as geometry cues. Our well-designed decomposed MLP shading demonstrates better performance than PBR shading and union shading.

| Ablation Variants | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| w/o normals | 34.5751 | 0.9613 | 0.0635 |
| w/o visibility | 33.9564 | 0.9596 | 0.0647 |
| w/o normals and visibility | 33.5752 | 0.9585 | 0.0659 |
| PBR Shading | 24.9865 | 0.8998 | 0.1146 |
| Union Shading | 33.8797 | 0.9603 | 0.0645 |
| Ours | **34.6826** | **0.9617** | **0.0629** |

(albedo $\boldsymbol{a}$, roughness $\rho$, metalness $m$) and uses occlusion-based light visibility. The PBR shading function [Walter et al. 2007] is defined as follows:

$$\boldsymbol{c} = \left((1-F)(1-m)\frac{\boldsymbol{a}}{\pi} + \frac{DFG}{4(\boldsymbol{w}_{\mathrm{o}}\cdot\boldsymbol{n})(\boldsymbol{w}_{\mathrm{i}}\cdot\boldsymbol{n})}\right)(\boldsymbol{w}_{\mathrm{i}}\cdot\boldsymbol{n})V(\boldsymbol{p},\boldsymbol{w}_{\mathrm{i}})L_{\mathrm{int}}, \qquad (8)$$

where $D$, $F$, and $G$ are the normal distribution function, the Fresnel equation, and the geometry function, respectively. $V(\boldsymbol{p},\boldsymbol{w}_{\mathrm{i}})$ is our visibility from shadow mapping. $\boldsymbol{n}$ is the normal, also supervised by mesh normals. $L_{\mathrm{int}}$ is the light intensity from Eq. 3.

As shown in Fig. 7, PBR shading produces plausible results only for materials (mainly opaque solid surfaces) that can be well described by the shading function in Eq. 8. It is beyond Eq. 8's capacity to model effects such as anisotropic reflection (1st row), semi-transparency (2nd row, the yellow cup), and specular interreflection (3rd row). Therefore, different shading functions are required to reproduce these diverse effects. To this end, we leverage the versatility of neural networks to describe various materials without handcrafting for each scene to achieve high-fidelity results.

(2) Union Shading. Inspired by NRHints [Zeng et al. 2023], it is straightforward to feed all inputs to a single color network that predicts the final color without decomposition. This network contains the same number of hidden layers with a doubled number of nodes in each layer as our two MLPs. However, as shown in Fig. 7 and Tab. 4, though visually similar, union shading results fall short of brightness accuracy. On the contrary, our method produces more accurate results benefiting from the decomposition of incident

illumination and scattering, which do not interfere with each other, as shown in Fig. 6.

## 5 Limitations and Discussion

*Indirect Lighting.* OLAT Gaussians are generic to produce certain indirect lighting effects. However, without the explicit guidance of multi-bounce reflection rays, our results of indirect lighting are less than perfect (e.g., 2nd row in Fig. 3, self-reflection on the baseboard). An additional indirect lighting field might improve our results further.

*Shadows.* Although the visibility from shadow mapping helps our method produce reasonable shadows, the shadow edges are not as sharp as the ground truth (Fig. 5). Since the sharp shadow edges change continuously with the moving point light, fuzzy artifacts arise when the edges appear at regions covered by sparse Gaussians that are insufficient to reproduce sharp edges. Densifying Gaussians with a dedicated strategy might alleviate this issue.

## 6 Conclusion

This paper has presented OLAT Gaussians to reconstruct objects' relightable appearance from OLAT images. With the well-designed shading decomposition, our method produces high-quality results on diverse materials, including opaque surfaces, semi-transparent volumes, furs, and fabrics. With an order of magnitude less training time than previous methods, our method reproduces more faithful details. Our method also achieves real-time rendering, which may support future interactive applications.

## Acknowledgments

We thank the reviewers for their insightful comments. We thank Chong Zeng for helping with the comparison experiments. This work was supported in part by NSF China (No. 62172363). Yanchao Yang was supported by the Early Career Scheme of the Research Grants Council (grant # 27207224), and the HKU-100 Award.

## References

Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2020. X-fields: Implicit neural view-, light-and time-image interpolation. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. 2021a. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 12684–12694.

Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan Barron, Hendrik Lensch, and Varun Jampani. 2022. Samurai: Shape and material from unconstrained real-world arbitrary image collections. *Advances in Neural Information Processing Systems* 35 (2022), 26389–26403.

Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. 2021b. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems* 34 (2021), 10691–10704.

Brent Burley and Walt Disney Animation Studios. 2012. Physically-based shading at disney. In *Acm Siggraph*, Vol. 2012. vol. 2012, 1–7.

Guangyan Cai, Kai Yan, Zhao Dong, Ioannis Gkioulekas, and Shuang Zhao. 2022. Physics-based inverse rendering using combined implicit and explicit geometries. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 129–138.

Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khamis, Or Litany, and Sanja Fidler. 2021. DIB-R++: learning to predict lighting and material with a hybrid differentiable renderer. *Advances in Neural Information Processing Systems* 34 (2021), 22834–22848.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques.* 145–156.

Yue Dong. 2019. Deep appearance modeling: A survey. *Visual Informatics* 3, 2 (2019), 59–68.

Farshad Einabadi, Jean-Yves Guillemaut, and Adrian Hilton. 2021. Deep neural models for illumination estimation and relighting: A survey. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 315–331.

Sara Fridovich-Keil, Alex 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.* 5501–5510.

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* 35 (2022), 3403–3416.

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

Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. 2023. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv preprint arXiv:2311.16043* (2023).

Yuan-Chen Guo. 2022. Instant Neural Surface Reconstruction. https://github.com/bennyguo/instant-nsr-pl.

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

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. 2023. GaussianShader: 3D Gaussian Splatting with Shading Functions for Reflective Surfaces. *arXiv preprint arXiv:2311.17977* (2023).

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42, 4 (2023), 1–14.

Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. 2022. Neroic: Neural rendering of objects from online image collections. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–12.

Quewei Li, Jie Guo, Yang Fei, Feichao Li, and Yanwen Guo. 2022. Neulighting: Neural lighting for free viewpoint outdoor scene relighting with unconstrained photo collections. In *SIGGRAPH Asia 2022 Conference Papers.* 1–9.

Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. 2023a. NerfAcc: Efficient Sampling Accelerates NeRFs. *arXiv preprint arXiv:2305.04966* (2023).

Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. 2023b. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8456–8465.

Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. 2023a. Envidr: Implicit differentiable renderer with neural environment lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 79–89.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. 2023b. Gs-ir: 3d gaussian splatting for inverse rendering. *arXiv preprint arXiv:2311.16473* (2023).

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

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

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

Linjie Lyu, Ayush Tewari, Thomas Leimkühler, Marc Habermann, and Christian Theobalt. 2022. Neural radiance transfer fields for relightable novel-view synthesis with global illumination. In *European Conference on Computer Vision.* Springer, 153–169.

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.* 7210–7219.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

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

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.* 8280–8290.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 165–174.

Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, and George Drettakis. 2019. Multi-view relighting using a geometry-aware network. *ACM Trans. Graph.* 38, 4 (2019), 78–1.

Julien Philip, Sébastien Morgenthaler, Michaël Gharbi, and George Drettakis. 2021. Free-viewpoint indoor neural relighting from multi-view stereo. *ACM Transactions on Graphics (TOG)* 40, 5 (2021), 1–18.

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image based relighting using neural networks. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 1–12.

Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. 2022. Nerf for outdoor scene relighting. In *European Conference on Computer Vision.* Springer, 615–631.

Kripasindhu Sarkar, Marcel C Bühler, Gengyan Li, Daoye Wang, Delio Vicini, Jérémy Riviere, Yinda Zhang, Sergio Orts-Escolano, Paulo Gotardo, Thabo Beeler, et al. 2023. LitNeRF: Intrinsic Radiance Decomposition for High-Quality View Synthesis and Relighting of Faces. In *SIGGRAPH Asia 2023 Conference Papers.* 1–11.

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. *Advances in Neural Information Processing Systems* 34 (2021), 6087–6101.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 7495–7504.

Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, and Ravi Ramamoorthi. 2021. Nelf: Neural light-transport field for portrait view synthesis and relighting. *arXiv preprint arXiv:2107.12351* (2021).

Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. 2022. Advances in neural rendering. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 703–735.

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

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

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. *arXiv preprint arXiv:2106.10689* (2021).

Xulong Wang, Siyan Dong, Youyi Zheng, and Yanchao Yang. 2024. InfoNorm: Mutual Information Shaping of Normals for Sparse-View Reconstruction. *arXiv preprint arXiv:2407.12661* (2024).

Zian Wang, Tianchang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Müller, and Zan Gojcic. 2023. Adaptive shells for efficient neural radiance field rendering. *arXiv preprint arXiv:2311.10091* (2023).

Yingyan Xu, Gaspard Zoss, Prashanth Chandran, Markus Gross, Derek Bradley, and Paulo Gotardo. 2023. Renerf: Relightable neural radiance fields with nearfield lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 22581–22591.

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. 2018. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)* 37, 4 (2018), 1–13.

Jing Yang, Hanyuan Xiao, Wenbin Teng, Yunxuan Cai, and Yajie Zhao. 2023. Light Sampling Field and BRDF Representation for Physically-based Neural Rendering. *arXiv preprint arXiv:2304.05472* (2023).

Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K Wong. 2022. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In *European Conference on Computer Vision.* Springer, 266–284.

Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2022. Neilf: Neural incident light field for material and lighting estimation. *Computer Vision–ECCV* (2022), 23–27.

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

Chong Zeng, Guojun Chen, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. 2023. Relighting neural radiance fields with shadow and highlight hints. In *ACM SIGGRAPH 2023 Conference Proceedings.* 1–11.

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.* 5565–5574.

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021b. 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.* 5453–5462.

Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. 2021a. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG)* 40, 1 (2021), 1–17.

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

Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. 2022b. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 18643–18652.

Quan Zheng, Gurprit Singh, and Hans-Peter Seidel. 2021. Neural relightable participating media rendering. *Advances in Neural Information Processing Systems* 34 (2021), 15203–15215.

Bingfan Zhu, Yanchao Yang, Xulong Wang, Youyi Zheng, and Leonidas Guibas. 2023. Vdn-nerf: Resolving shape-radiance ambiguity via view-dependence normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 35–45.

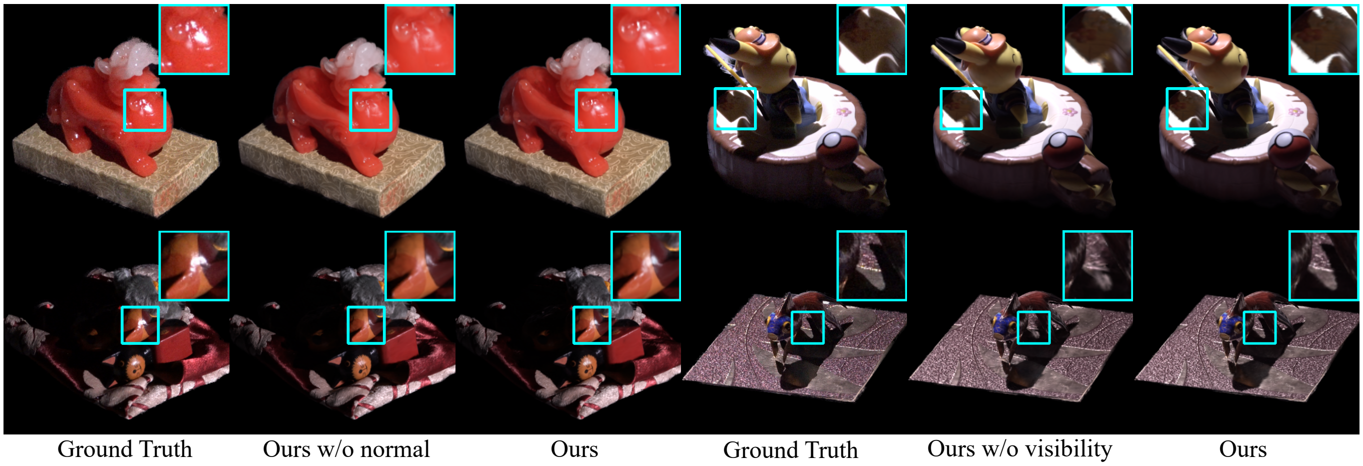


Fig. 5. Ablation study of normals and visibility. Our method produces more accurate specular highlights with Gaussian normals and sharper shadow edges with visibility.

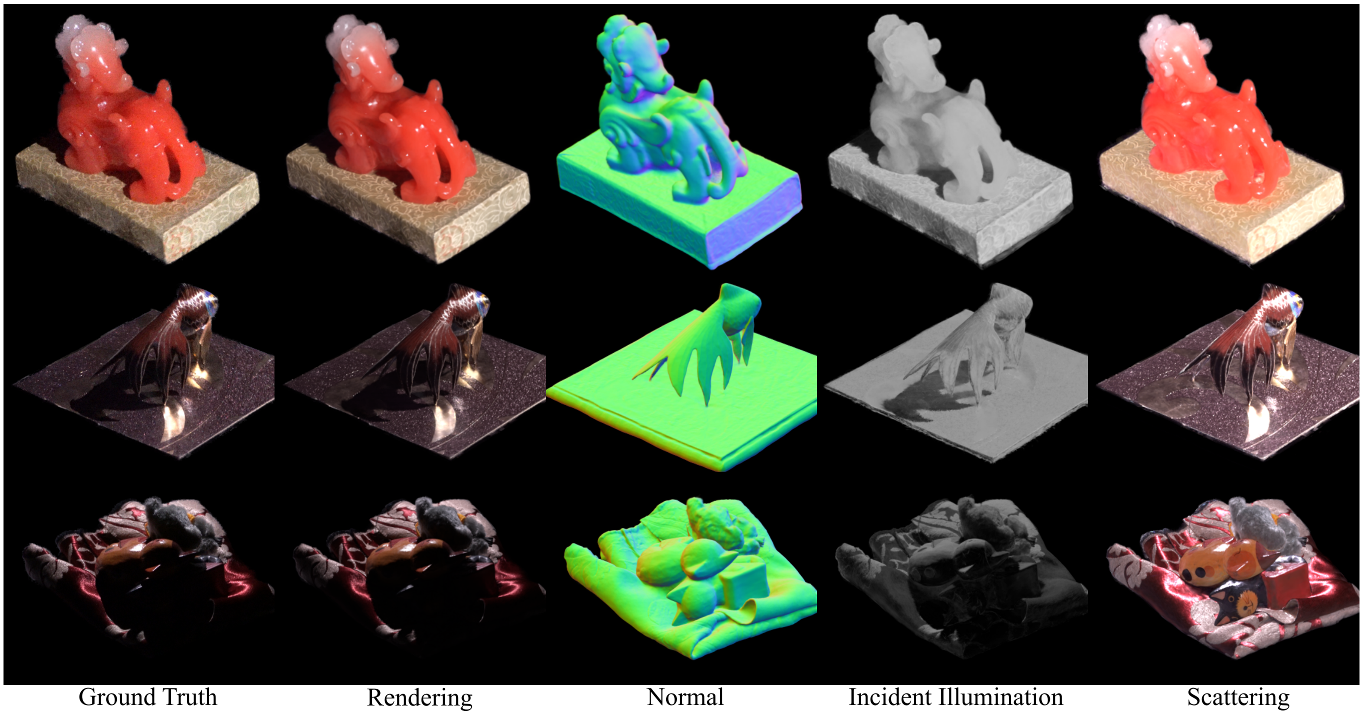


Fig. 6. Components of our rendering results. The normals match objects' underlying geometry well. The incident illumination and scattering parts are well disentangled.

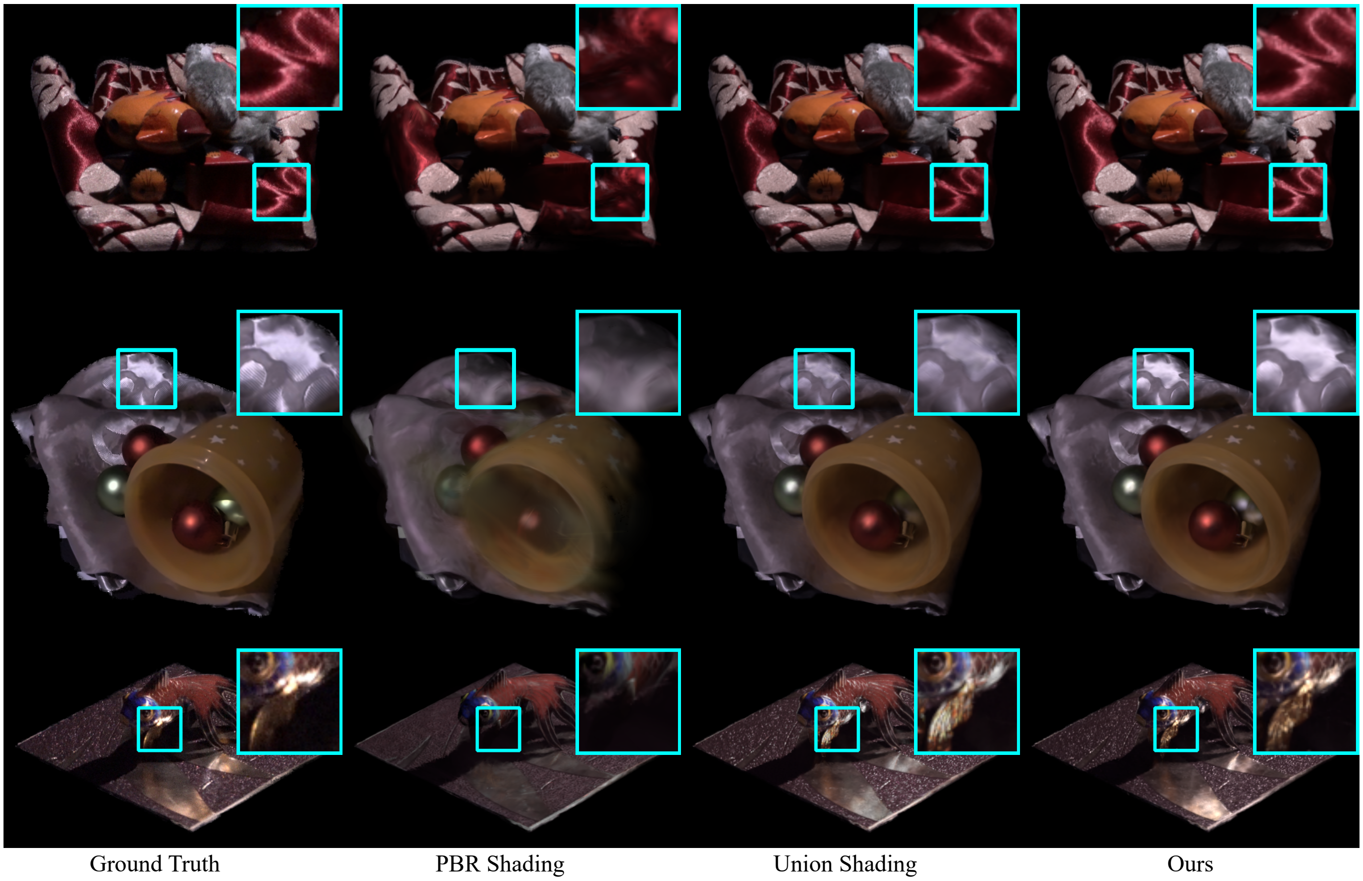


Fig. 7. Ablation study of different shading formulations. Our decomposed MLP shading is more versatile than PBR shading to reproduce various lighting effects, such as anisotropic highlights (1st row), semi-transparency (2nd row, the yellow cup), and specular interreflection (3rd row). Our shading achieves more accurate brightness than union shading.