

GlossyGS: Inverse Rendering of Glossy Objects with 3D Gaussian Splatting

Shuichang Lai, Letian Huang, Jie Guo, Kai Cheng, Bowen Pan, Xiaoxiao Long, Jiangjing Lyu, Chengfei Lv,
Yanwen Guo

Abstract—Reconstructing objects from posed images is a crucial and complex task in computer graphics and computer vision. While NeRF-based neural reconstruction methods have exhibited impressive reconstruction ability, they tend to be time-consuming. Recent strategies have adopted 3D Gaussian Splatting (3D-GS) for inverse rendering, which have led to quick and effective outcomes. However, these techniques generally have difficulty in producing believable geometries and materials for glossy objects, a challenge that stems from the inherent ambiguities of inverse rendering. To address this, we introduce *GlossyGS*, an innovative 3D-GS-based inverse rendering framework that aims to precisely reconstruct the geometry and materials of glossy objects by integrating material priors. The key idea is the use of micro-facet geometry segmentation prior, which helps to reduce the intrinsic ambiguities and improve the decomposition of geometries and materials. Additionally, we introduce a normal map prefiltering strategy to more accurately simulate the normal distribution of reflective surfaces. These strategies are integrated into a hybrid geometry and material representation that employs both explicit and implicit methods to depict glossy objects. We demonstrate through quantitative analysis and qualitative visualization that the proposed method is effective to reconstruct high-fidelity geometries and materials of glossy objects, and performs favorably against state-of-the-arts.

Index Terms—Inverse Rendering, Neural Rendering

I. INTRODUCTION

RECONSTRUCTING objects with realistic materials using images from multiple viewpoints is a crucial and complex task in computer graphics and computer vision. This challenge is exacerbated when dealing with images taken in uncontrolled settings where the lighting conditions are unknown. Several methods [1]–[7] build on Neural Radiance Fields (NeRFs) [8], using neural networks to depict objects' materials and geometry. However, the required dense sampling for rays undermines the efficiency of these inverse rendering approaches. Recently, 3D Gaussian Splatting (3D-GS) [9] has emerged as an effective strategy, offering high quality and real-time performance.

Recent studies [10]–[12] try to build inverse rendering pipelines based on 3D-GS. However, these methods mostly

Shuichang Lai and Letian Huang contributed equally.

Shuichang Lai, Bowen Pan, Jiangjing Lyu and Chengfei Lv are with Alibaba Group (e-mail: laishuichang.lsc@alibaba-inc.com; bowen.pbw@alibaba-inc.com; jiangjing.ljj@alibaba-inc.com; chengfei.lcf@alibaba-inc.com).

Letian Huang, Jie Guo and Yanwen Guo are with the National Key Lab for Novel Software Technology, Nanjing University, Nanjing 210000, China (e-mail: lthuang@smail.nju.edu.cn; guojie@nju.edu.cn; ywguo@nju.edu.cn).

Kai Cheng is with the school of artificial intelligence and data science, University of Science and Technology of China, China (e-mail: chengkai21@mail.ustc.edu.cn).

Xiaoxiao Long is with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: xxlong@connect.hku.hk).

Jie Guo is the corresponding author.

perform well on non-specular objects. Till now, several challenges remain: specular highlights and environmental reflections introduce ambiguities, complicating the differentiation between inherent textures and reflections, thus leading to low-quality geometry reconstruction. Furthermore, microscopic variations in the normals of glossy surfaces can change how they look from distance. This is because tiny folds or protrusions can make microscopic smoothness appear rough. This nonlinear relationship between microscopic normal changes and appearance challenges 3D Gaussian-based forward shading methods, leading to inaccuracies in surface normal estimation.

In this paper, we introduce *GlossyGS*, a novel 3D-GS-based inverse rendering framework designed to accurately identify the physical properties of glossy objects by incorporating material prior. The key idea of *GlossyGS* consists of three aspects. Firstly, we introduce a *normal map prefiltering* strategy that modifies the shading order by blending the normals first. This approach improves the accuracy of simulating the normal distribution of glossy surfaces. Secondly, we propose the *micro-facet geometry segmentation prior* to adjust the micro-facet normal distribution and the geometry of reflective surfaces. This prior significantly reduces the inherent ambiguities in inverse rendering and enhances the decomposition of both geometry and material properties. A micro-facet geometry segmentation model is developed to predict this prior using semantic representations derived from large vision models. Finally, we design a *hybrid geometry and material representation* that utilizes anchors to explicitly represent the macroscopic coarse geometry while implicitly employing feature vectors associated with these anchors to predict multiple neural Gaussians. These neural Gaussians are essential for precisely simulating the normal distribution and material properties of the micro-facet, offering a detailed and realistic representation of glossy objects' surfaces. Extensive experimental results show that our approach can reconstruct high-fidelity geometries and materials of glossy objects, leading to state-of-the-art rendering for relighting. To summarize, our main contributions are three-fold.

- We propose the micro-facet geometry segmentation prior to minimize the inherent ambiguities in inverse rendering, consequently enhancing the decomposition of both geometry and material properties.
- We introduce a normal map prefiltering strategy that changes the shading order, enhancing the simulation of the normal distribution for glossy surfaces.
- We design a hybrid geometry and material representa-

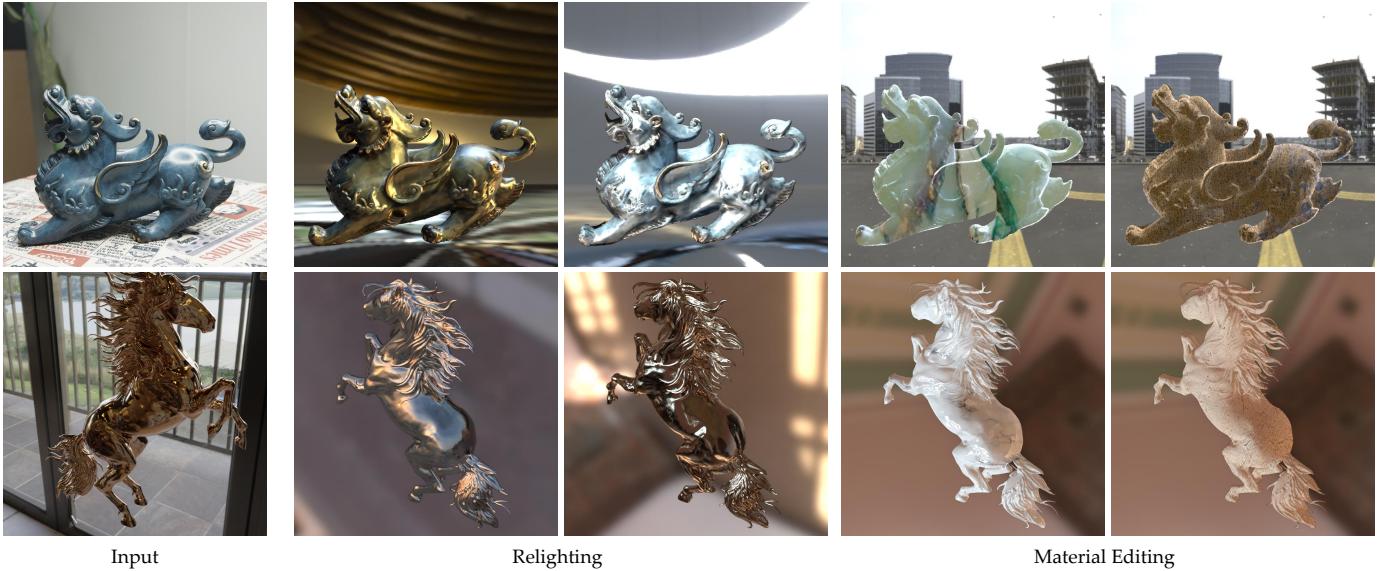


Fig. 1. We present an innovative 3D-GS-based inverse rendering framework that aims to precisely reconstruct the geometries and materials of glossy objects by integrating priors from glossy appearance analysis. The well disentangled geometric and material properties allow us to achieve high-quality relighting (the second and third columns) and material editing (the fourth and fifth columns), especially on highly glossy objects.

tion tailored for glossy objects, achieving state-of-the-art inverse rendering results.

II. RELATED WORK

A. 3D Gaussian Splatting

3D Gaussian Splatting (3D-GS) [9] has achieved high-fidelity real-time neural rendering by splatting the Gaussian primitives onto the image plane for a fast tile-based rasterization, instead of densely sampling points on the rays like Neural Radiance Fields (NeRF) [8], [13] for a slow volume rendering. Due to its real-time performance, some works attempt to integrate the efficient explicit representation into human modeling [14]–[19], dynamic scenes [20]–[22], large-scale scenes [23], [24] and slam [25]–[27], etc. Gaussian Editors [28], [29] facilitate editing of the appearance and geometry of 3D-GS through the diffusion model [30]–[32]. SuGaR [33] and 2D-GS [34], analyze the surface representation capability of Gaussians and propose a method for representing Gaussians’ normals, facilitating the extraction from Gaussians into traditional meshes. Our method also employs a high-fidelity normal map prefiltering strategy to facilitate the inverse rendering for glossy objects, which enables photo-realistic relighting and material editing.

Furthermore, there are some works focusing on addressing the current limitations of 3D-GS. Yu et al. [35] utilizes a 3D smoothing filter and a 2D mip filter to reduce aliasing when zooming in and zooming out. Huang et al. [36] proposes an optimal projection strategy by minimizing projection errors, resulting in photo-realistic rendering under various focal lengths and various camera models. Radl et al. [37] and Bulò et al. [38] conduct analysis and improvements based on 3D Gaussians’ depth sorting and density control respectively. Scaffold-GS [39] employs a hierarchical hybrid representation of scenes through the combination of explicit 3D Gaussians

and implicit MLP layers, significantly reducing the storage overhead of 3D-GS. Our method leverages the hybrid structure to infer neural Gaussians and neural materials and introduce the similarity of material within latent features to constrain material ambiguities. These designs facilitate the success of inverse rendering of glossy objects.

B. Inverse Rendering of Glossy Objects

Reconstruction and inverse rendering of a glossy object from mult-view images has been a challenging task due to the complex light interactions and ambiguities [40]–[49].

NeRO [2] use Integrated Directional Encoding (IDE) like Ref-NeRF [50] and employs a two-stage inverse rendering strategy. The first stage uses split-sum approximation [3] to accelerate integral calculations, while the second stage employs Monte Carlo sampling to refine materials. TensoIR [4] employs TensoRF [51] as its geometry representation and utilizes ray marching to compute indirect illumination and visibility. NMF [7] uses a microfacet reflectance model within a volumetric setting by treating each sample along the ray as a (potentially non-opaque) surface. However, these methods that rely on implicit representations [1]–[6], [50], [52], [53] involve densely sampled points along rays, leading to a significant decrease in the performance of inverse rendering.

Some works [10], [12] aim to employ 3D-GS [9] as an explicit representation of scenes in order to enhance the performance of inverse rendering. GaussianShader [12] decomposes the original color attributes into three components including diffuse, specular and residual color for modeling the color of reflective surfaces. In GS-IR [10], a technique similar to light probes [54] is employed to obtain the indirect light and ambient occlusion. However, these methods [10], [12] face challenges in representing accurate surfaces and materials using 3D Gaussians without correlation and suffer from the

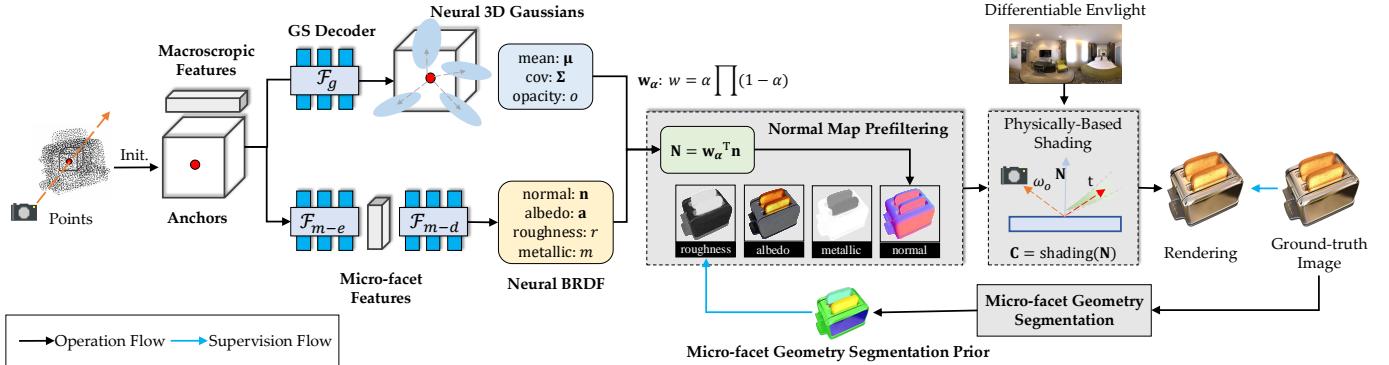


Fig. 2. Illustration of the inverse rendering pipeline for our *GlossyGS*. The initial points are used to generate anchors, which carry the macroscopic features. These features are fed to GS decoder and material encoder, generating neural 3D Gaussians and neural materials (BRDF). Normal map prefiltering is used as a prefilter for these neural Gaussians and materials to obtain the corresponding material maps. The final rendering is shading of these maps and differentiable environment lighting, supervised by ground-truth images and micro-facet geometry segmentation prior.

ambiguities between macroscopic normal and microscopic roughness, resulting in lower quality of inverse rendering. In contrast, our method effectively addresses these issues.

III. METHOD

In this section, we present our approach, *GlossyGS*, tailored for the inverse rendering of glossy objects, as illustrated in Fig. 2. To achieve fast and high-quality reconstruction of glossy objects, we employ a hybrid explicit-implicit geometry and material representation to infer neural Gaussians and materials (BRDFs). Our experiments reveal that the shading order of the Gaussians negatively impacts the geometry reconstruction of glossy objects. To mitigate this issue, we propose a normal map prefiltering strategy, which allows us to first generate material maps, subsequently using them to obtain the final rendered images through shading. Another factor affecting the reconstruction of glossy objects is the ambiguities arising from the coupling of microscopic and macroscopic normal distributions. To address this problem, we introduce a micro-facet geometry segmentation prior. Furthermore, we develop a segmentation model that learns this prior based on our custom-built dataset.

A. Preliminaries

3D Gaussian Splatting [9] 3D Gaussian Splatting takes the sparse point clouds generated during the SFM [55] calibration as input. From these points, it constructs a set of 3D Gaussians as graphical primitives explicitly representing the scene. Each graphic primitive is characterized by Gaussian mean μ and covariance matrix Σ , as well as opacity o to represent volumetric density, and is characterized by spherical harmonic (SH) coefficients to represent appearance.

The 3D primitives are projected to 2D screen space, and rasterized using α -blending. The α -blending weights are given as

$$w_k = \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j), \quad (1)$$

where α is given by $o \cdot g$. The projected Gaussian function g at pixel (u, v) is represented by:

$$g(u, v) = e^{\left(-\frac{1}{2}([u, v]^\top - \mu')^\top \Sigma'^{-1} ([u, v]^\top - \mu')\right)} \quad (2)$$

where $\mu' = \text{proj}(\mu)$ and $\Sigma' = \text{proj}(\Sigma)$ represent the projected 2D mean and covariance matrix, respectively. The pixel color at (u, v) is calculated using α -blending with spherical harmonics (SH) as follows:

$$C(u, v) = \sum_{k=1}^N \text{SH}_k(u, v) w_k, \quad (3)$$

where $\text{SH}_k(u, v)$ represents the color of the k^{th} primitive at pixel (u, v) . Expanding this method to include attributes like normal and shading gives as:

$$Y(u, v) = \sum_{k=1}^N y_k w_k, \quad (4)$$

a formula commonly used in inverse rendering tasks based on 3D-GS [10]–[12]. In this work, we utilize a modified 3D-GS approach for high-quality, real-time inverse rendering.

Cook-Torrance BRDF [56] We follow the Cook-Torrance model [56] and formulate the bidirectional reflectance distribution function (BRDF) as follow:

$$f(\omega_i, \omega_o) = \underbrace{(1 - m) \frac{\mathbf{a}}{\pi}}_{\text{diffuse}} + \underbrace{\frac{DFG}{4(\omega_i \cdot \mathbf{n})(\omega_o \cdot \mathbf{n})}}_{\text{specular}}, \quad (5)$$

where $m \in [0, 1]$ is the metallic of the point, $\mathbf{a} \in [0, 1]^3$ is the albedo color of the point, \mathbf{n} denotes the surface normal, ω_i denotes the incident direction, ω_o denotes the outgoing view direction. D , F , and G refer to the normal distribution function, the Fresnel term, and the geometry term respectively. D , F , and G are generally determined by the metallic m , the roughness $r \in [0, 1]$ and the albedo \mathbf{a} . To tackle the intractable radiance integral, we adopt image-based lighting

(IBL) model and split-sum approximation [3] following [2]. Here, the lighting integral is approximated as:

$$c(\omega_o) = c_{\text{diffuse}} + c_{\text{specular}}(\omega_o), \quad (6)$$

$$c_{\text{diffuse}} = \mathbf{a}(1-m) \cdot \underbrace{\int_{\Omega} L(\omega_i) \frac{\omega_i \cdot \mathbf{n}}{\pi} d\omega_i}_{L_d}, \quad (7)$$

$$\begin{aligned} c_{\text{specular}}(\omega_o) &= \int_{\Omega} L(\omega_i) \frac{DFG}{4(\omega_o \cdot \mathbf{n})} d\omega_i \\ &\approx \underbrace{\int_{\Omega} \frac{DFG}{4(\omega_o \cdot \mathbf{n})} d\omega_i}_{M_s} \cdot \underbrace{\int_{\Omega} D(r, \mathbf{t}) L(\omega_i) d\omega_i}_{L_s}, \end{aligned} \quad (8)$$

where L_s denotes the integral of lights on the micro-facet normal distribution function $D(r, \mathbf{t})$, \mathbf{t} is the reflective direction, L_d denotes the diffuse light integral, M_s denotes the integral of BRDF.

B. Hybrid Geometry and Material Representation

Inspired by Scaffold-GS [39], we adopt a hybrid explicit-implicit geometry representation and employ neural networks to generate neural Gaussians to represent objects. Specifically, the sparse point cloud from COLMAP [55] is initialized as anchors, with each anchor equipped with a feature vector $\mathbf{f}_{\text{macro}}$, a scaling factor s , and learnable offsets. We then employ the GS decoder \mathcal{F}_g to generate neural Gaussians that represent the geometry of objects:

$$\mu, \Sigma, o = \mathcal{F}_g(\mathbf{f}_{\text{macro}}, \mathbf{x}_{\text{anchor}}) \quad (9)$$

where $\mathbf{x}_{\text{anchor}}$ denotes positions of the anchors and μ, Σ, o denote attributes of 3D Gaussians. We further extend this hybrid representation to BRDF materials. Since the $\mathbf{f}_{\text{macro}}$ contains complete Gaussian geometry and material information, we use the material encoder \mathcal{F}_{m-e} to generate high-dimensional micro-facet features

$$\mathbf{f}_{\text{micro}} = \mathcal{F}_{m-e}(\mathbf{f}_{\text{macro}}), \quad (10)$$

which are then decoded to obtain material attributes

$$r, m, \mathbf{n}, \mathbf{a} = \mathcal{F}_{m-d}(\mathbf{f}_{\text{micro}}) \quad (11)$$

where r, m are the roughness and metallic of the Gaussian respectively. And \mathbf{n}, \mathbf{a} are the normal and albedo color of the Gaussian, respectively. This representation not only reduces the storage cost of materials but also enhances the correlation between materials in local space for they share a common MLP. For more details about MLP structures, please refer to the supplementary material.

Building on this design, we devise a smooth, low-dimensional manifold for material representation. This contiguous manifold ensures smooth variations in material attributes, thereby preventing the abrupt changes caused by incorrect gradients. This approach not only stabilizes the optimization process but also enhances the robustness of our network, ensuring a more reliable and consistent performance in material representation. In practice, we apply a smoothness regularization on the decoder \mathcal{F}_{m-d} ,

$$\mathcal{L}_s = \|\mathcal{F}_{m-d}(\mathbf{f}_{\text{micro}}) - \mathcal{F}_{m-d}(\mathbf{f}_{\text{micro}} + \xi)\|_1 \quad (12)$$

where ξ denotes a random variable sampled from a normal distribution with mean 0 and variance 0.01.

Additionally, since our geometric and material attributes are still discrete, our hybrid representation retains the high performance of the vanilla 3D-GS [9] by projecting them onto the image plane and performing α -blending instead of dense sampling of points along rays.

C. Normal Map Prefiltering

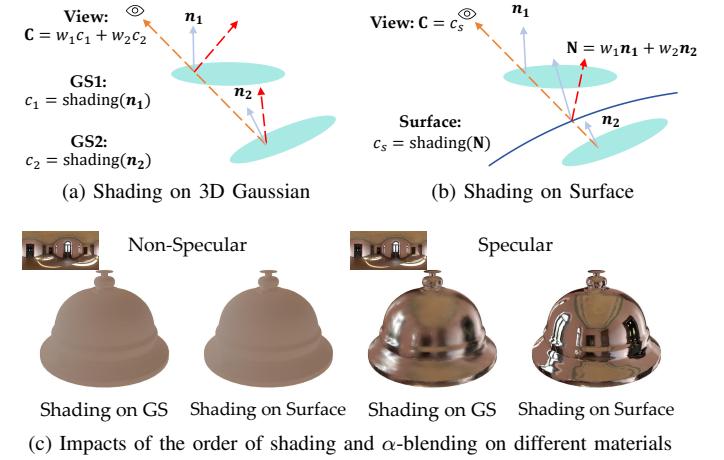


Fig. 3. **Normal Map Prefiltering.** $\mathbf{n}_1, \mathbf{n}_2, \mathbf{N}$ represent the normal of Gaussian 1, Gaussian 2, and the true surface, respectively. c_1, c_2, c_s represent the shading results of Gaussian 1, Gaussian 2, and the true surface, respectively. \mathbf{C} represents the color of the pixel, and w_1 and w_2 represent the weights for α -blending. (a) is the current approach used by 3D-GS-based IR methods [11], [12]. (b) On the other hand, our normal map prefiltering considers the nonlinearity of normals with respect to specular color (c) which has a significant impact on the rendering results of glossy objects.

Accurately predicting normals on reflective surfaces presents a formidable challenge. Through Eqn. (6), it can be inferred that shading of non-specular objects satisfies the following equation with respect to the surface normals \mathbf{n} :

$$\begin{aligned} \sum_{k=1}^N w_k c_{\text{diffuse}} &= \sum_{k=1}^N w_k \mathbf{a}(1-m) \cdot \int_{\Omega} L(\omega_i) \frac{\omega_i \cdot \mathbf{n}}{\pi} d\omega_i \\ &= \mathbf{a}(1-m) \cdot \int_{\Omega} L(\omega_i) \frac{\omega_i \cdot (\sum_{k=1}^N w_k \mathbf{n})}{\pi} d\omega_i. \end{aligned} \quad (13)$$

Diffuse color is linear in the normal. Eqn. (4) also tells that α -blending in 3D-GS [9] is a linear operation. Therefore, for diffuse objects, the order of α -blending and shading has little effect on the final color. However, specular reflection does not satisfy this linear relationship, i.e.,

$$\begin{aligned} \sum_{k=1}^N w_k c_{\text{specular}} &= \sum_{k=1}^N w_k \int_{\Omega} L(\omega_i) \frac{DFG}{4(\omega_o \cdot \mathbf{n})} d\omega_i \\ &\neq \int_{\Omega} L(\omega_i) \frac{DFG}{4(\omega_o \cdot (\sum_{k=1}^N w_k \mathbf{n}))} d\omega_i. \end{aligned} \quad (14)$$

Therefore, the order of shading and α -blending is crucial for inverse rendering of glossy objects. Unfortunately, current

methods mostly ignore the influence of this order, resulting in unsatisfactory inverse rendering results for glossy objects.

The above-mentioned two different orders are demonstrated in Fig. 3. They differ when the shading function is nonlinear in the normal. Most 3D-GS-based inverse rendering methods [10]–[12] employ the gradient of the depth map as the normal supervision strategy. Since the depth map represents the depth of the true surface, it is necessary to perform shading on the surface instead of shading on 3D Gaussians from this perspective. Furthermore, we demonstrate the visual impact of the shading order for different materials in Fig. 3c. Given the linear relationship between diffuse color and normal, the order of shading does not significantly affect the appearance of non-specular objects. However, for specular objects, performing shading first and then applying α -blending leads to a blurred appearance caused by the linear blending of the ambient lighting sampled from multiple reflective directions (the red arrows in Fig. 3a). This approach fails to accurately capture the environment reflections. In contrast, we reverse the order by blending the normals first, enabling us to maintain a more precise representation of the surface normal. By subsequently sampling the ambient lighting based on a single reflective direction (the red arrow in Fig. 3b), we achieve a more realistic rendering of specular objects. These different rendering results stem from the nonlinear relationship between the shading function and the normal for glossy objects. Our proposed normal prefiltering strategy successfully address this issue.

D. Micro-facet Geometry Segmentation Prior

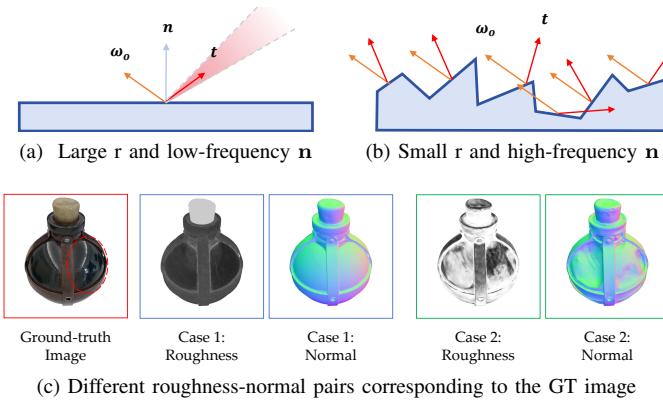


Fig. 4. Ambiguities Between Macroscopic Normal and Microscopic Roughness. ω_o, n, t, r represent outgoing view direction, normal, reflected light, and roughness respectively. (a) represents large roughness and simple normal, while (b) is the opposite, with complex normal and each plane can be considered as a specular reflection with smaller roughness. Both can represent the same reflective object, leading to the occurrence of Case 1 and Case 2 in (c).

The failure in reconstructing glossy objects is mainly attributed to the ambiguities arising from the coupling of microscopic and macroscopic normal distributions introduced by the micro-facet model [56]. The initial motivation behind the introduction of the micro-facet model is to represent complex materials in situations where the geometry is not excessively complex, thus reducing the storage overhead associated with geometry. The success of this model relies on the ability of

the microscopic normal distribution $D(r, t)$ characterized by roughness r to produce visual effects similar to the macroscopic normal n . However, achieving similar visual effects poses a challenge for inverse rendering tasks.

As shown in Fig. 4a and 4b, a low-frequency macroscopic normal distribution with a large roughness and a high-frequency macroscopic normal distribution with a small roughness can model the same glossy appearance. This leads to different roughness-normal pairs for the individual object as shown in Fig. 4c. The introduction of environmental light information leads to visual color inconsistencies in parts that were expected to have the same roughness, while parts that were expected to have different roughness exhibit visual color consistency as shown in Fig. 4c. This complicates the decoupling of microscopic roughness and macroscopic normal, making it challenging to obtain accurate solutions through simple inverse rendering gradient backpropagation.

Ideally, we can directly apply roughness priors to narrow down the solution space, effectively alleviating ambiguities. However, we still encounter challenges in estimating roughness, particularly when working with a single image, as it is impossible to completely separate this from the unknown lighting conditions. To address this issue, we propose a prior for segmenting the geometry of micro-facets, which helps eliminate cases like Case 2 in Figure 4c. In these cases, the solutions satisfy the rendering equations [57] but do not correspond to real-world glossy objects, unlike Case 1 shown in the same figure. In specific terms, we segment the roughness of glossy objects, with each segment representing a consistent roughness value. This prior serves to constrain the micro-facet normal distribution and geometry of reflective objects.

E. Micro-facet Geometry Segmentation Model

Similar to semantic segmentation, the micro-facet geometry segmentation model can be formulated as a mapping function from the image domain to the output label domain: $I \rightarrow B$, which predicts a pixel-wise category label $B \in \{0, \dots, C\}^{H \times W}$, where H and W denote the image size, C is the total number of roughness categories. Given that an object is usually composed of a limited number of materials, we assume that a single object can be composed of up to 6 roughness categories. Accordingly, we sort the values in the roughness map and select the top C most frequently occurring values to filter the corresponding pixels for each category. Pixels that are not selected are categorized as label 0, excluding the background.

During training, we employ the cross-entropy loss defined as follows:

$$\mathcal{L}_{ce}(I, B) = - \sum_{u=1}^H \sum_{v=1}^W \sum_{c=1}^C b_{(u,v),c} \log p_{(u,v)}(c | I), \quad (15)$$

where (u, v) are the pixel coordinates in the input image I , c is the category index, $b_{(u,v),c} \in \{0, 1\}$ is entry in the one-hot vector of the ground-truth label, and $p_{(u,v)}(c | I)$ is the predicted category probability for category c at pixel (u, v) .

Since roughness is an intrinsic characteristic of a material, it can be challenging to observe directly. To address this issue,

we utilize existing pre-trained vision models as the backbone for our optimization process. DINOv2 [58] models have been widely adopted across various semantic-related tasks, such as image retrieval and semantic segmentation, demonstrating robust performance even when using frozen weights without any fine-tuning. To leverage this impressive semantic capability in our segmentation model, we propose incorporating the DINOv2 encoder for feature extraction. Specifically, we utilize the DINOv2 encoder to capture rich feature representations and pair it with the DPT decoder [59] to perform effective segmentation. This design enables the network to learn to distinguish similar roughness regions by leveraging semantic-aware representations derived from DINOv2, utilizing our custom-built dataset. As shown in Fig. 5, the segmentation model can accurately partition regions of varying roughness, thereby ensuring that the reconstructed roughness is both smooth and reasonable.

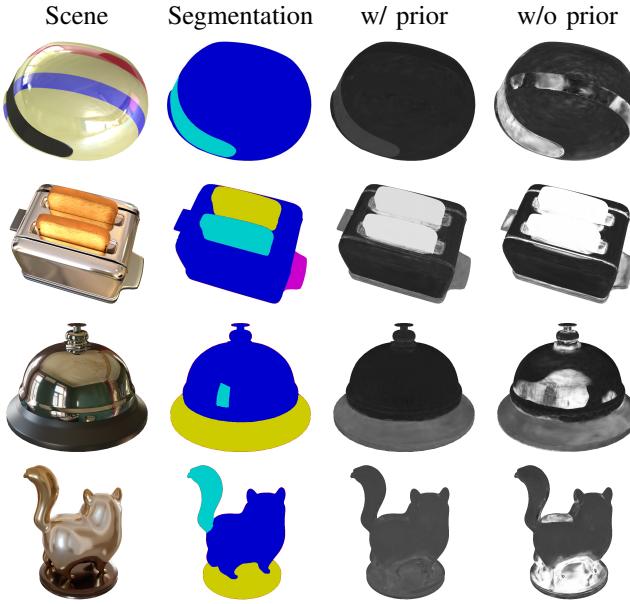


Fig. 5. Visualizing the results of micro-facet geometry segmentation and the effect of the segmentation prior. Without the segmentation prior, the reconstructed roughness is noisy and lacks smoothness. Note that there is no available ground-truth roughness in these examples.

F. The Inverse Rendering Framework

Based on the micro-facet geometry segmentation prior and the normal map prefiltering strategy, we propose a method for reconstructing high-fidelity geometries and materials of glossy objects from a set of posed images I^s . An overview of our approach is presented in Fig. 2. Initially, we obtain the geometric and material attributes of each Gaussian through network inference. Subsequently, using α -blending as described in Eqn. (4), we prefilter the maps for each attribute. We then utilize these maps along with the rendering equation [57], to perform shading and produce the final image, denoted as \hat{I} . Due to the fact that the maps represent the attributes of the true surface rather than the 3D Gaussians, and considering the nonlinearity of shading with respect to the normal, our

approach yields more accurate results compared to directly shading on 3D Gaussians. Furthermore, the calculation of environment mipmaps follows the same methodology as that of [10], [12].

During training, the learnable parameters and MLPs are jointly optimized by minimizing the reconstruction loss \mathcal{L} , represented as:

$$\mathcal{L} = \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_e \mathcal{L}_e + \lambda_n \mathcal{L}_n. \quad (16)$$

\mathcal{L}_c denotes the combined image loss, similar to the approach used in Scaffold-GS [39], defined as follows:

$$\mathcal{L}_c = \mathcal{L}_1 + \lambda_{SSIM} \mathcal{L}_{SSIM} + \lambda_{vol} \mathcal{L}_{vol}. \quad (17)$$

In this equation, the parameters λ_{SSIM} and λ_{vol} are set to 0.2 and 0.001, respectively.

The roughness constraint loss \mathcal{L}_e leverages the consistent roughness within segmented regions to supervise the roughness map I_ρ . Specifically, it is defined as follows:

$$\mathbb{S}(c) = \{I_\rho(u, v) \mid \hat{B}(u, v) = c\}, \quad (18)$$

$$\mathcal{L}_e = \sum_{c \in C} \sqrt{\frac{1}{|\mathbb{S}(c)| - 1} \sum_{i=1}^{|\mathbb{S}(c)|} (z_i - \bar{z})^2}, \quad (19)$$

where $z_i \in \mathbb{S}(c)$ and \bar{z} represents the mean value of the elements within $\mathbb{S}(c)$. Additionally, \hat{B} represents the segmented labels obtained from the micro-facet geometry segmentation model applied to the images \hat{I} . During the inverse rendering process, minimizing loss \mathcal{L}_e involves fitting a more accurate macroscopic normal distribution, thereby achieving improved geometry reconstruction of glossy objects.

\mathcal{L}_n is a normal loss that correlates the local geometry with the predicted normals by reducing the discrepancy between the gradient normal maps \hat{n}_D that are calculated from the rendered depth maps and the rendered normal maps \hat{n} , defined as:

$$\mathcal{L}_n = \|\hat{n} - \hat{n}_D\|. \quad (20)$$

The parameters λ_s , λ_e , and λ_n are all set to 0.01 in our experiments.

IV. IMPLEMENTATION DETAILS

In this section, we present our datasets, and training details.

a) *Datasets and Metrics:* We conduct experiments on two synthetic datasets, Shiny Blender [50] and Glossy Synthetic [2], and two real datasets, Stanford-ORB [60] and NeILF-HDR [61]. To further demonstrate the robustness of our method, we also captured two real scenes under indoor lighting. Following the official guidelines, images in the Stanford-ORB dataset are resized to a resolution of 512×512 for training and testing. We utilize tone mapping to convert high-dynamic range (HDR) images from the NeILF-HDR dataset into low-dynamic range (LDR) images. To evaluate the quality of the estimated normals, we compare estimated normal images with the ground truth using the mean angular error (MAE). Regarding rendering quality, we use PSNR, SSIM and LPIPS [62] to evaluate the similarity between the rendered images and the ground truth images.

TABLE I

QUANTITATIVE COMPARISON OF NORMAL RECONSTRUCTION AND RELIGHTING RESULTS ON THE SHINY BLENDER [50] DATASET. **BOLD** MEANS THE BEST PERFORMANCE AND UNDERLINE MEANS THE SECOND BEST.

	NeRO	NMF	Gshader	GSIR	Ours
PSNR↑	21.53	<u>25.17</u>	21.17	21.95	25.72
SSIM↑	0.858	<u>0.916</u>	0.867	0.857	0.93
LPIPS↓	0.144	<u>0.119</u>	0.133	0.152	0.103
normal MAE↓	3.02	1.81	6.81	9.06	<u>2.82</u>
training time↓	12h	4h	1.2h	<u>1.1h</u>	1h
rendering FPS↑	0.03	0.04	16	<u>30</u>	32

TABLE II

QUANTITATIVE COMPARISON OF NOVEL VIEW SYNTHESIS RESULTS ON THE GLOSSY SYNTHETIC [2] DATASET. **BOLD** MEANS THE BEST PERFORMANCE.

	NeRO	NMF	Gshader	GSIR	Ours
PSNR↑	29.73	28.22	28.65	25.35	30.46
SSIM↑	0.904	0.926	0.943	0.899	0.960
LPIPS↓	0.324	0.091	0.072	0.101	0.049

b) Micro-facet Geometry Segmentation Prior Model

Training: We collect training data by randomly picking 100,000 objects with roughness maps from the Objaverse dataset [63]. We render each object in ten random poses against five random HDR environment maps to capture a variety of realistic lighting conditions. The base learning rate of the pre-trained encoder is set as $5e-6$, while the randomly initialized decoder uses a $10\times$ larger learning rate. We employ the AdamW optimizer and reduce the learning rate by a factor of 0.8 every five epochs. Additionally, we apply horizontal flipping as a form of data augmentation. The training process runs for approximately three days on four NVIDIA A100 GPUs.

c) Inverse Rendering: The GlossyGS model is trained on a single NVIDIA V100 GPU with 32GB memory, completing the training process in approximately 1 hour for images at an 800×800 resolution. It achieves a rendering speed of around 30 FPS at the same resolution on the V100 GPU. Additionally, the learnable environment map has a resolution of 256×512 .

V. RESULTS

In this section, we compare our method with the following two types of baseline methods. (1) **3D-GS-based methods**: Gshader [12] and GSIR [10]. (2) **NeRF-based methods**: neural microfacet fields (NMF) [7] and NeRO [2]. For all these methods, we use their official implementations. While NeRF-based methods can achieve impressive reconstruction results, they are notably time-consuming.

A. Results on synthetic data

We conduct quantitative and qualitative evaluations on two synthetic datasets, examining normal reconstruction, material decomposition, novel view synthesis, and relighting.

Visual comparisons of the normal reconstruction on synthetic datasets are illustrated in Fig. 6, highlighting our

method’s capability in reconstructing surfaces exhibiting substantial reflection, alongside the recovery of detailed geometries (the nuanced folds of the clothing in the *Angel* scene). NeRO [2] exhibits remarkable reconstruction ability when applied to simple specular objects, but when applied to more complex scenarios, it tends to converge on local minima, leading to inaccurate geometries (pointed by the red box in the *Luyu* scene). NMF [7] can roughly recover geometry, but it tends to lose fine details (the missing door handle in the *Car* scene). Gshader [12] often incorrectly overfits highlights by exploiting geometric indentations, due to its forward shading mode. GSIR [10] excels in reconstructing non-specular surfaces but face difficulties with surfaces showing strong reflections. We report the quantitative metrics on the Shiny Blender [50] dataset in Tab. I. Our method demonstrates superior performance in relighting quality compared to baseline methods, highlighting the effectiveness of inverse rendering. Importantly, our method achieves the shortest training time, demonstrating a $4\times$ improvement over NeRF-based methods, which showcases its efficiency. Furthermore, we attain the highest relighting rendering frame rate, thanks to our hybrid geometry and material representation.

Additionally, we showcase the estimated BRDFs (albedo, roughness and metallic) in Fig. 7. Thanks to the micro-facet geometry segmentation prior and the roughness constraint loss, the roughness maps tend to be smooth and physically-plausible, which reduces the ambiguities in inverse rendering and enhances the decoupling of albedo and metallic. This contrasts with other methods that inaccurately assign lighting effects to albedo maps. The reconstructed environment maps, as illustrated in Fig. 8, further support our method’s ability to effectively decouple the BRDFs, allowing for a nearly complete restoration of the environment map. In contrast, GShader can only generate noisy images with many holes, while GSIR struggles with poor environmental maps due to its failure to decouple light from the albedo.

Our method excels in novel view synthesis and relighting, outperforming others, as evidenced in Tab. I and Tab. II. Fig. 10 highlights our success in capturing specular reflections in novel view synthesis. Furthermore, as shown in Fig. 9, our relighting results produce realistic visualizations and effectively recover specular highlights. NeRO mainly suffers from a loss of geometric details, leading to blurred highlight details (pointed by the red box in the *Horse* scene). NMF and GShader both encounter issues with geometries and materials, resulting in inferior relighting quality. Meanwhile, GSIR produces diffuse materials and struggles to model high-frequency effects.

B. Results on real data

We first evaluate our method on the NeILF-HDR dataset [61] and several real scenes captured by ourself, as shown in Fig. 11. We compare our method with the other four methods on normal reconstruction. As seen, our approach effectively handles surfaces with strong highlights and recovers geometric details. In contrast, GShader [12] demonstrates unstable geometry recovery, resulting in some voids, as shown in the

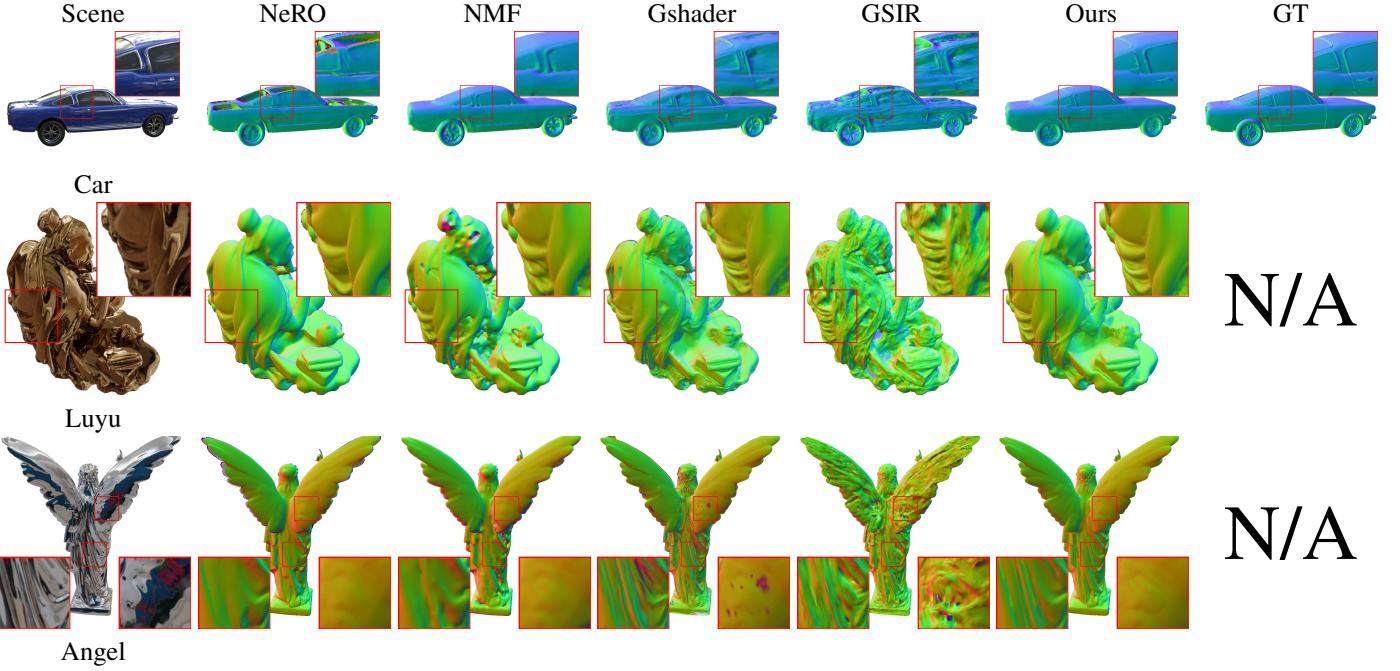


Fig. 6. Comparison of normal reconstruction among our method, NMF [7], NeRO [2], Gshader [12] and GSIR [10] on the Shiny Blender [50] (the first row), Glossy Synthetic datasets [2](the second and third rows). Note that there is no available ground-truth normal in the Glossy Synthetic dataset.



Fig. 7. Comparison of the estimated BRDFs among our method, NeRO [2] and GSIR [10] on the Shiny Blender dataset [50].

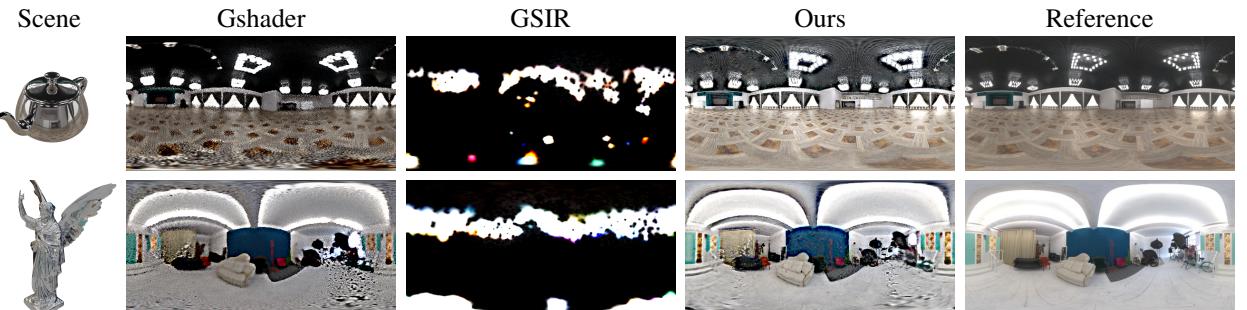


Fig. 8. Comparison of environment maps among our method, Gshader [12] and GSIR [10] on the Glossy Synthetic dataset [2].

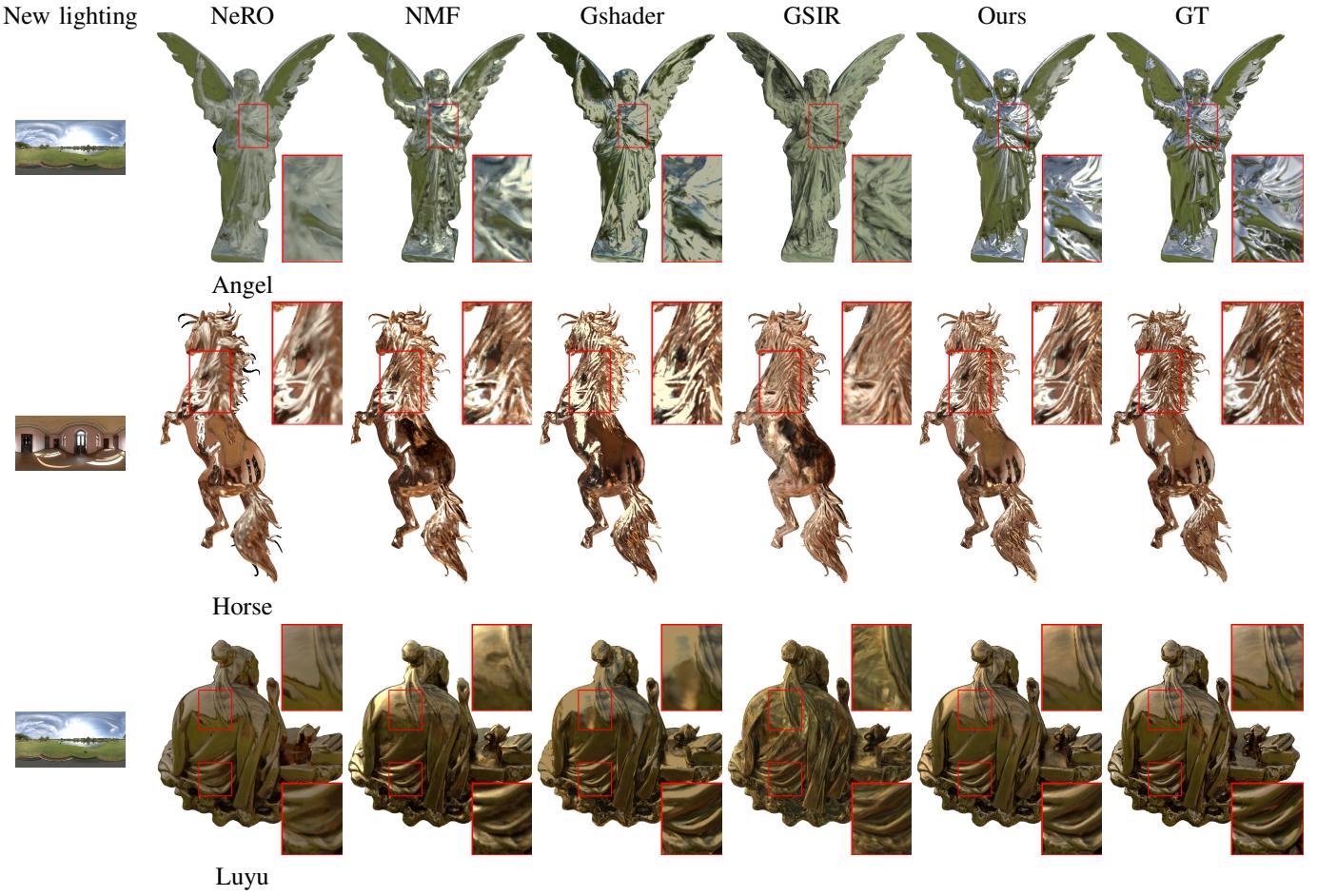


Fig. 9. Comparison of relighting results among our method, NMF [7], NeRO [2], Gshader [12] and GSIR [10] on the Glossy Synthetic dataset [2].



Fig. 10. Comparison of novel view synthesis among our method, NMF [7], NeRO [2], Gshader [12] and GSIR [10] on the Glossy Synthetic [2] (the first row), and Stanford-ORB dataset [60](the second row).

Qilin and *Brassgourd* scene. GS-IR [10] struggles to generate smooth geometric surfaces. NeRF-based methods (NeRO [2] and NMF [7]) can only generate coarse geometries without details (pointed by the red box in the *GoldenDragon* scene). The relighting results (as shown in the seventh column of Fig. 11) further demonstrate the reliability of our material reconstruction, as the specular highlights change appropriately with variations in lighting conditions.

We also evaluate our method on four real glossy objects from the Stanford-ORB [60] dataset. The quantitative measurements of normal reconstruction and relighting results are

TABLE III
QUANTITATIVE COMPARISON OF NORMAL RECONSTRUCTION AND
RELIGHTING RESULTS ON THE STANFORD-ORB [60] DATASET. **BOLD**
MEANS THE BEST PERFORMANCE.

	NeRO	NMF	Gshader	GSIR	Ours
PSNR \uparrow	27.35	25.64	25.76	28.04	30.3
SSIM \uparrow	0.969	0.944	0.957	0.962	0.975
LPIPS \downarrow	0.051	0.048	0.041	0.033	0.022
normal MAE \downarrow	1.84	1.81	2.01	2.69	1.75

shown in Tab. III. Our method outperforms all the other

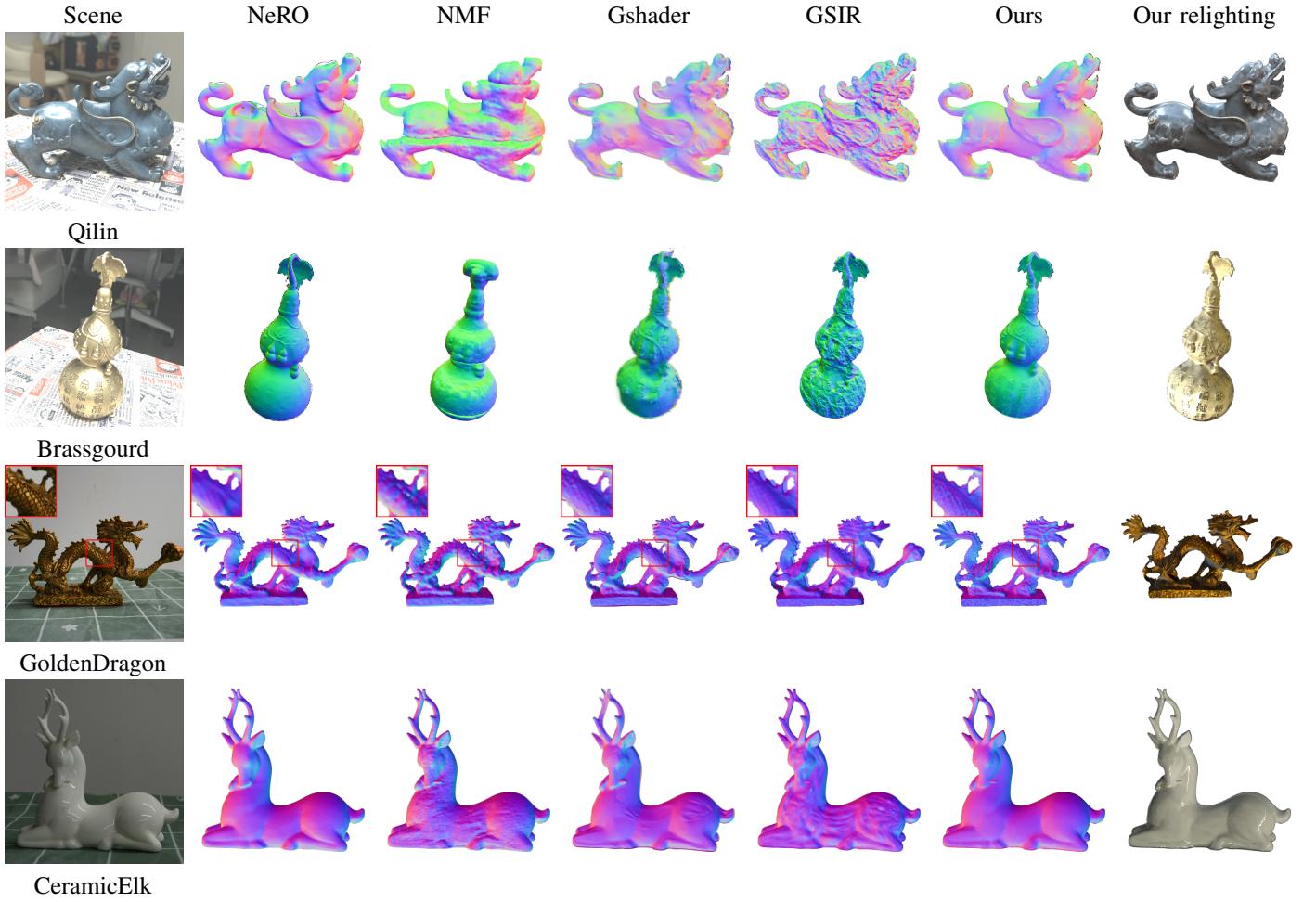


Fig. 11. Comparison of normal reconstruction and relighting results among our method, NMF [7], NeRO [2], Gshader [12] and GSIR [10] on the NEILF-HDR [61] dataset (the first and second rows) and real captured scenes (the third and forth rows).

TABLE IV

ABALATION STUDIES ON THE RELIGHTING RESULTS USING OBJECTS FROM THE GLOSSY SYNTHETIC DATASET [2]. **BOLD** MEANS THE BEST PERFORMANCE.

	PSNR↑	SSIM↑	LPIPS↓
w/o prior	23.51	0.898	0.098
w/o normal map prefILTERING	24.61	0.922	0.072
w/o hybrid	23.52	0.902	0.082
Ours	25.47	0.928	0.068

methods. We also present the visual results in Fig. 12 and Fig. 13. Again, we see that our method accurately reconstructs geometries and materials, displaying glossiness in real-scene relighting that matches the original objects. Conversely, prior methods struggle to model geometry in highly glossy regions, resulting in incorrect material properties and low-quality relighting effects.

C. Ablation studies

In this subsection, we evaluate several design choices in our pipeline by conducting ablation studies on them.

a) *Micro-facet Geometry Segmentation Prior*: We introduce the micro-facet geometry segmentation prior to enhance

the decomposition of geometry and material properties. To assess the effectiveness of the prior, we conduct a comparison with a baseline model trained without the prior. As depicted in Fig. 14, the baseline model incorrectly separates micro-scale roughness and macro-scale normal, resulting in inaccurate geometry. Utilizing prior knowledge, our method successfully generates the correct normals. The data presented in Tab. IV further confirm that the integration of the prior aids the model in generating reliable materials.

b) *Normal Map Prefiltering*: To assess the effectiveness of the strategy, we conduct an experiment by implementing the training without normal prefILTERING. The comparative analysis between the relighting results is displayed in Fig. 14. In the absence of normal prefILTERING, the relighting results tend to produce blurred visual appearance. However, our method is capable of clearly mapping environmental lighting, achieving results that are close to the ground truth.

c) *Hybrid Geometry and Material Representation*: We utilize a hybrid explicit-implicit geometry representation to generate neural Gaussians for representing glossy objects. To validate the effectiveness of this strategy for inverse rendering, we conducted an experiment using the explicit Gaussian design in the same manner as Gshader [12]. As shown in Fig.

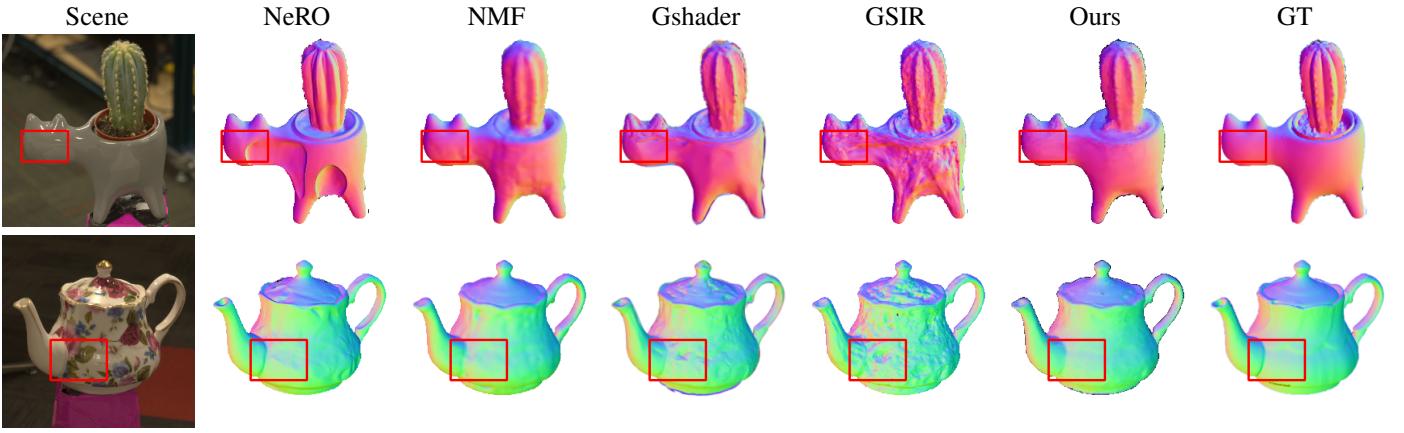


Fig. 12. Comparison of normal reconstruction among our method, NMF [7], NeRO [2], Gshader [12] and GSIR [10] on the Stanford-ORB dataset [60].



Fig. 13. Comparison of relighting among our method, NMF [7], NeRO [2], Gshader [12] and GSIR [10] on the Stanford-ORB [60] dataset.

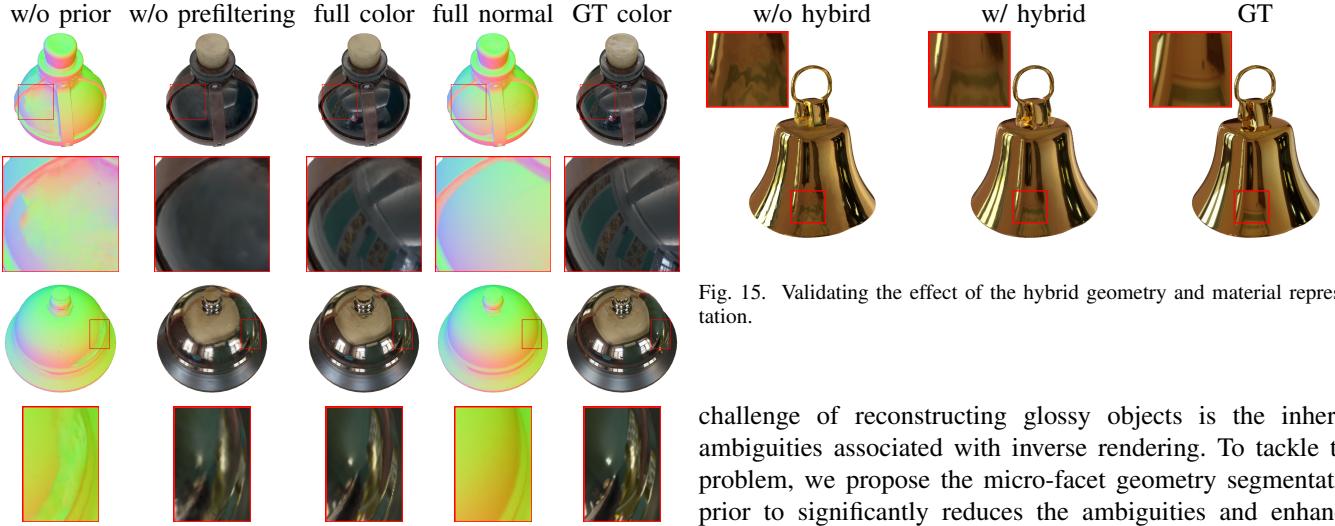


Fig. 14. Validating the effect of the micro-facet geometry segmentation prior and the normal map prefiltering in training.

15, without hybrid geometry and material representation, the explicit Gaussian fails to produce realistic relighting results. Quantitative metrics in Tab. IV further validate this.

VI. CONCLUSION

We have presented a new framework aimed at the efficient reconstruction of glossy objects with 3D-GS. The key

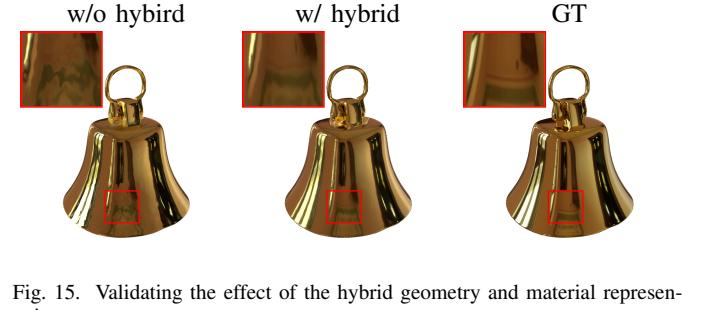


Fig. 15. Validating the effect of the hybrid geometry and material representation.

challenge of reconstructing glossy objects is the inherent ambiguities associated with inverse rendering. To tackle this problem, we propose the micro-facet geometry segmentation prior to significantly reduces the ambiguities and enhances the decomposition of both geometry and material properties. Additionally, we introduce a normal map prefiltering strategy that enhances the GlossyGS's ability to more accurately simulate the normal distribution of glossy surfaces. Finally, we introduce a hybrid geometry and material representation tailored for glossy objects. Extensive experiments on both synthetic and real datasets demonstrate consistently superior performance of the proposed method.

Limitations and future work. Although GlossyGS can accurately reconstruct the geometries and materials of glossy objects without sacrificing performance, it still has some limitations. One problem is that our method assumes distant

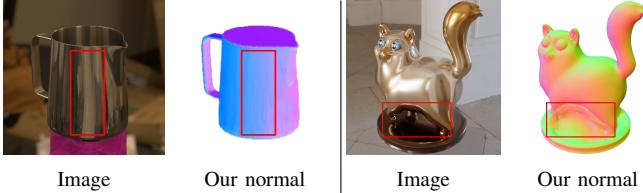


Fig. 16. Limitations. Our method struggles to remove interreflections, resulting in incorrect geometries.

illumination, which means near-field lighting effects are not considered. In reality, glossy objects often capture reflections from the photographer, which change with camera angles, as shown in Fig. 16. Additionally, our method does not effectively handle interreflections due to our rendering scheme. In such scenarios, while our carefully designed prior and prefiltering strategy can minimize the impact of these reflections on geometry, it still struggles to produce accurate geometries. One potential solution is to model dynamic illumination similarly to NeRO [2]. Furthermore, our method struggles with concave surfaces due to viewpoint limitations, a common challenge in geometry reconstruction. We plan to address this issue in future work.

REFERENCES

- [1] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, “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*, 2021, pp. 5453–5462.
- [2] Y. Liu, P. Wang, C. Lin, X. Long, J. Wang, L. Liu, T. Komura, and W. Wang, “Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–22, 2023.
- [3] B. Karis and E. Games, “Real shading in unreal engine 4,” *Proc. Physically Based Shading Theory Practice*, vol. 4, no. 3, p. 1, 2013.
- [4] H. Jin, I. Liu, P. Xu, X. Zhang, S. Han, S. Bi, X. Zhou, Z. Xu, and H. Su, “Tensoir: Tensorial inverse rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 165–174.
- [5] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, “Nerv: Neural reflectance and visibility fields for relighting and view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7495–7504.
- [6] C. Zeng, G. Chen, Y. Dong, P. Peers, H. Wu, and X. Tong, “Relighting neural radiance fields with shadow and highlight hints,” in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–11.
- [7] A. Mai, D. Verbin, F. Kuester, and S. Fridovich-Keil, “Neural microfacet fields for inverse rendering,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 408–418.
- [8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.
- [10] Z. Liang, Q. Zhang, Y. Feng, Y. Shan, and K. Jia, “Gs-ir: 3d gaussian splatting for inverse rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 21644–21653.
- [11] J. Gao, C. Gu, Y. Lin, H. Zhu, X. Cao, L. Zhang, and Y. Yao, “Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing,” *arXiv preprint arXiv:2311.16043*, 2023.
- [12] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, and Y. Ma, “Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 5322–5332.
- [13] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.
- [14] S. Hu and Z. Liu, “Gauhuman: Articulated gaussian splatting from monocular human videos,” *arXiv preprint arXiv:2312.02973*, 2023.
- [15] L. Hu, H. Zhang, Y. Zhang, B. Zhou, B. Liu, S. Zhang, and L. Nie, “Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians,” *arXiv preprint arXiv:2312.02134*, 2023.
- [16] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner, “Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians,” *arXiv preprint arXiv:2312.02069*, 2023.
- [17] A. Moreau, J. Song, H. Dhamo, R. Shaw, Y. Zhou, and E. Pérez-Pellitero, “Human gaussian splatting: Real-time rendering of animatable avatars,” *arXiv preprint arXiv:2311.17113*, 2023.
- [18] M. Kocabas, J.-H. R. Chang, J. Gabriel, O. Tuzel, and A. Ranjan, “Hugs: Human gaussian splats,” *arXiv preprint arXiv:2311.17910*, 2023.
- [19] Z. Qian, S. Wang, M. Mihajlovic, A. Geiger, and S. Tang, “3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting,” *arXiv preprint arXiv:2312.09228*, 2023.
- [20] Z. Lu, X. Guo, L. Hui, T. Chen, M. Yang, X. Tang, F. Zhu, and Y. Dai, “3d geometry-aware deformable gaussian splatting for dynamic view synthesis,” *arXiv preprint arXiv:2404.06270*, 2024.
- [21] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, “4d gaussian splatting for real-time dynamic scene rendering,” *arXiv preprint arXiv:2310.08528*, 2023.
- [22] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, “Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes,” *arXiv preprint arXiv:2312.07920*, 2023.
- [23] J. Lin, Z. Li, X. Tang, J. Liu, S. Liu, J. Liu, Y. Lu, X. Wu, S. Xu, Y. Yan *et al.*, “Vastgaussian: Vast 3d gaussians for large scene reconstruction,” *arXiv preprint arXiv:2402.17427*, 2024.
- [24] Y. Liu, H. Guan, C. Luo, L. Fan, J. Peng, and Z. Zhang, “Citygaussian: Real-time high-quality large-scale scene rendering with gaussians,” *arXiv preprint arXiv:2404.01133*, 2024.
- [25] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, “Gaussian splatting slam,” *arXiv preprint arXiv:2312.06741*, 2023.
- [26] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, “Gsslam: Dense visual slam with 3d gaussian splatting,” *arXiv preprint arXiv:2311.11700*, 2023.
- [27] M. Li, S. Liu, and H. Zhou, “Sgs-slam: Semantic gaussian splatting for neural dense slam,” *arXiv preprint arXiv:2402.03246*, 2024.
- [28] Y. Chen, Z. Chen, C. Zhang, F. Wang, X. Yang, Y. Wang, Z. Cai, L. Yang, H. Liu, and G. Lin, “Gaussianeditor: Swift and controllable 3d editing with gaussian splatting,” *arXiv preprint arXiv:2311.14521*, 2023.
- [29] J. Fang, J. Wang, X. Zhang, L. Xie, and Q. Tian, “Gaussianeditor: Editing 3d gaussians delicately with text instructions,” *arXiv preprint arXiv:2311.16037*, 2023.
- [30] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html>
- [31] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 11895–11907. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html>
- [32] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: <https://openreview.net/forum?id=PxTIG12RRHS>
- [33] A. Guédon and V. Lepetit, “Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering,” *arXiv preprint arXiv:2311.12775*, 2023.
- [34] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, “2d gaussian splatting for geometrically accurate radiance fields,” *arXiv preprint arXiv:2403.17888*, 2024.

- [35] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, “Mip-splatting: Alias-free 3d gaussian splatting,” *arXiv preprint arXiv:2311.16493*, 2023.
- [36] L. Huang, J. Bai, J. Guo, Y. Li, and Y. Guo, “On the error analysis of 3d gaussian splatting and an optimal projection strategy,” 2024.
- [37] L. Radl, M. Steiner, M. Parger, A. Weinrauch, B. Kerbl, and M. Steinberger, “Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering,” *arXiv preprint arXiv:2402.00525*, 2024.
- [38] S. R. Bulò, L. Porzi, and P. Kortschieder, “Revising densification in gaussian splatting,” *arXiv preprint arXiv:2404.06109*, 2024.
- [39] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, “Scaffold-gs: Structured 3d gaussians for view-adaptive rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 654–20 664.
- [40] S. Roth and M. J. Black, “Specular flow and the recovery of surface structure,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2. IEEE, 2006, pp. 1869–1876.
- [41] K. Han, K.-Y. K. Wong, D. Schnieders, and M. Liu, “Mirror surface reconstruction under an uncalibrated camera,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1772–1780.
- [42] J. Zeng, C. Bao, R. Chen, Z. Dong, G. Zhang, H. Bao, and Z. Cui, “Mirror-nerf: Learning neural radiance fields for mirrors with whitted-style ray tracing,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 4606–4615.
- [43] A. Dave, Y. Zhao, and A. Veeraraghavan, “Pandora: Polarization-aided neural decomposition of radiance,” in *European Conference on Computer Vision*. Springer, 2022, pp. 538–556.
- [44] A. Kadambi, V. Taamazyan, B. Shi, and R. Raskar, “Polarized 3d: High-quality depth sensing with polarization cues,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3370–3378.
- [45] T. Whelan, M. Goesele, S. J. Lovegrove, J. Straub, S. Green, R. Szeliski, S. Butterfield, S. Verma, R. A. Newcombe, M. Goesele *et al.*, “Reconstructing scenes with mirror and glass surfaces.” *ACM Trans. Graph.*, vol. 37, no. 4, p. 102, 2018.
- [46] M. Aittala, T. Aila, and J. Lehtinen, “Reflectance modeling by neural texture synthesis,” *ACM Transactions on Graphics (ToG)*, vol. 35, no. 4, pp. 1–13, 2016.
- [47] J. T. Barron and J. Malik, “Shape, illumination, and reflectance from shading,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1670–1687, 2014.
- [48] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, “Mitsuba 2: A retargetable forward and inverse renderer,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–17, 2019.
- [49] W. Chang, V. Sivarad, D. Nowrouzezahrai, T. Hachisuka, R. Ramamoorthi, and T.-M. Li, “Parameter-space restir for differentiable and inverse rendering,” in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–10.
- [50] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, “Ref-nerf: Structured view-dependent appearance for neural radiance fields,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 5481–5490.
- [51] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, “Tensorf: Tensorial radiance fields,” in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.
- [52] T. Wu, J.-M. Sun, Y.-K. Lai, and L. Gao, “De-nerf: Decoupled neural radiance fields for view-consistent appearance editing and high-frequency environmental relighting,” in *ACM SIGGRAPH 2023 conference proceedings*, 2023, pp. 1–11.
- [53] L. Ma, V. Agrawal, H. Turki, C. Kim, C. Gao, P. Sander, M. Zollhöfer, and C. Richardt, “Specnerf: Gaussian directional encoding for specular reflections,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 21 188–21 198.
- [54] P. Debevec, “A median cut algorithm for light probe sampling,” in *ACM SIGGRAPH 2008 classes*, 2008, pp. 1–3.
- [55] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [56] R. L. Cook and K. E. Torrance, “A reflectance model for computer graphics,” *ACM Transactions on Graphics (ToG)*, vol. 1, no. 1, pp. 7–24, 1982.
- [57] J. T. Kajiya, “The rendering equation,” in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, pp. 143–150.
- [58] M. Oquab, T. Darzet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, “Dinov2: Learning robust visual features without supervision,” 2023.
- [59] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 159–12 168.
- [60] Z. Kuang, Y. Zhang, H.-X. Yu, S. Agarwala, E. Wu, J. Wu *et al.*, “Stanford-orb: A real-world 3d object inverse rendering benchmark,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [61] J. Zhang, Y. Yao, S. Li, J. Liu, T. Fang, D. McKinnon, Y. Tsin, and L. Quan, “Neilf++: Inter-reflectable light fields for geometry and material estimation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3601–3610.
- [62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
- [63] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, “Objaverse: A universe of annotated 3d objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 142–13 153.