

GeoSplatting: Towards Geometry Guided Gaussian Splatting for Physically-based Inverse Rendering

Kai Ye^{1,*}, Chong Gao^{2,*}, Guanbin Li², Wenzheng Chen^{3,4,†}, Baoquan Chen^{1,4,†}

¹School of Intelligence Science and Technology, Peking University

²School of Computer Science and Engineering, Sun Yat-sen University

³Wangxuan Institute of Computer Technology, Peking University

⁴State Key Laboratory of General Artificial Intelligence, Peking University

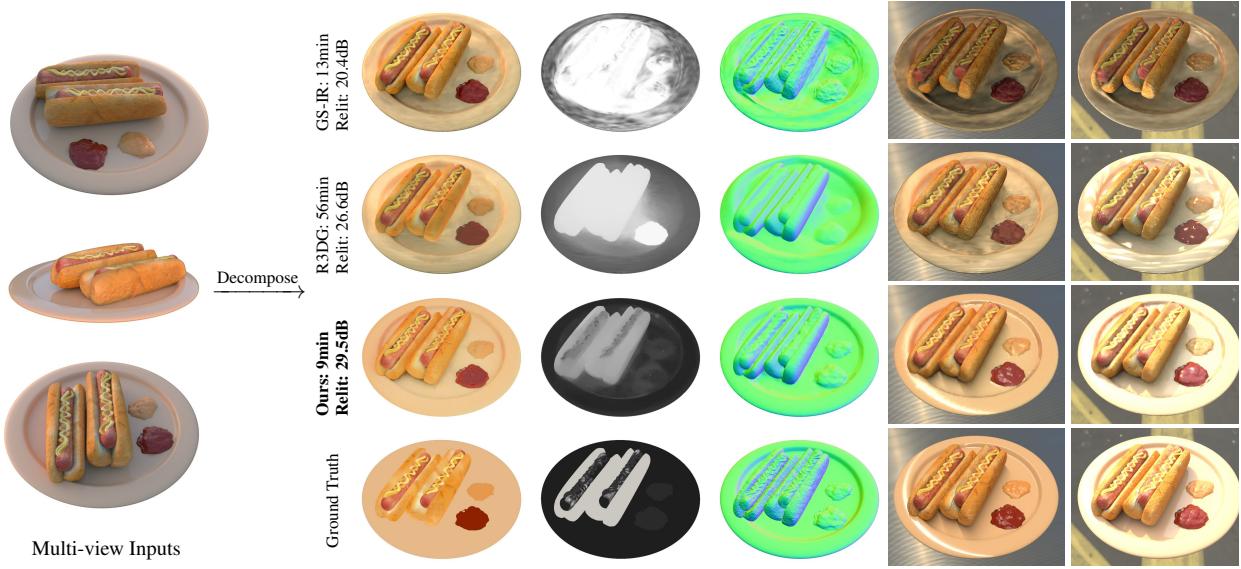


Figure 1. **Decomposition and Relighting Results.** Existing 3DGS-based inverse rendering methods, such as Relightable 3DGS (R3DG) [8] and GS-IR [24], typically rely on approximated normal directions, which can lead to inaccuracies in light transport modeling and subsequently result in noisy material decomposition and flawed relighting results. In contrast, we propose GeoSplatting, a novel approach that augments 3DGS with explicit geometry guidance to improve normal estimation and light transport modeling, resulting in superior decomposition and relighting quality, while also delivering excellent training efficiency.

Abstract

Recent 3D Gaussian Splatting (3DGS) representations [18] have demonstrated remarkable performance in novel view synthesis; further, material-lighting disentanglement on 3DGS warrants relighting capabilities and its adaptability to broader applications. While the general approach to the latter operation lies in integrating differentiable physically-based rendering (PBR) techniques to jointly recover BRDF materials and environment lighting, achieving a precise disentanglement remains an inherently difficult task due to the challenge of accurately modeling light transport. Existing approaches typically approximate Gaussian points'

normals, which constitute an implicit geometric constraint. However, they usually suffer from inaccuracies in normal estimation that subsequently degrade light transport, resulting in noisy material decomposition and flawed relighting results. To address this, we propose GeoSplatting, a novel approach that augments 3DGS with explicit geometry guidance for precise light transport modeling. By differentiably constructing a surface-grounded 3DGS from an optimizable mesh, our approach leverages well-defined mesh normals and the opaque mesh surface, and additionally facilitates the use of mesh-based ray tracing techniques for efficient, occlusion-aware light transport calculations. This enhancement ensures precise material decomposition while preserving the efficiency and high-quality rendering capabilities of 3DGS. Comprehensive evaluations across di-

*Equal contribution. †Equal advisory.

verse datasets demonstrate the effectiveness of GeoSplatting, highlighting its superior efficiency and state-of-the-art inverse rendering performance. The project page can be found at <https://pku-vcl-geometry.github.io/GeoSplatting/>.

1. Introduction

3D reconstruction, fueled by powerful representations [17, 28], has achieved significant progress in recent years. While the reconstructed objects exhibit detailed geometry and photorealistic appearance, enabling striking renderings from novel viewpoints [1, 38], the lack of material and lighting decomposition limits their adaptability to varying lighting conditions. Consequently, traditional 3D reconstruction methods fall short of producing versatile 3D assets suitable for downstream applications such as gaming, filmmaking, and AR/VR. To address this limitation, several approaches grounded in Physically-Based Rendering (PBR) theory [14] extend beyond traditional appearance recovery to material-lighting disentanglement, framing this problem as physically-based inverse rendering [2, 8, 12, 45].

Existing approaches tackle inverse rendering by combining differentiable PBR techniques with powerful 3D representations, enabling joint estimation of material and lighting through gradient descent. Earlier works rely on explicit 3D representations, such as triangular meshes, to seamlessly integrate with differentiable PBR pipelines [10, 31]. However, these approaches can suffer from less robust optimization, as mesh-based differentiable rendering methods typically generate gradients only at triangle edges [22]. In contrast, the success of NeRF and NeuS in novel view synthesis has inspired efforts to combine PBR techniques with neural fields, leading to more robust inverse rendering. Nevertheless, due to the heavy ray sampling involved in their volumetric rendering process, these methods are computationally expensive and require hours to converge [25, 40].

More recently, 3D Gaussian Splatting (3DGS) [18] has emerged as a powerful 3D representation, offering both high-quality novel-view synthesis and impressive efficiency. However, the lack of explicit geometry boundaries and well-defined surface normals prevents vanilla 3DGS from accurately modeling light transport, thereby hindering effective material-lighting disentanglement. Motivated by geometry enhancement techniques for 3DGS [3, 6, 9, 11, 26, 39, 43], several approaches have attempted to extend its use to inverse rendering tasks. A common strategy involves squashing 3DGS points into surfels, from which surface normals can be approximated via implicit geometric constraints, such as depth-normal regularization [8, 12, 24, 35]. Nevertheless, accurately modeling light transport requires both precise normal directions and opaque surfaces, which define light propagation directions and light-surface intersections, respectively. As a result, ex-

isting 3DGS-based inverse methods—relying on approximated normals and the inherently semi-transparent nature of Gaussian points—typically struggle with light transport modeling, leading to noisy material decompositions and flawed relighting.

To address these challenges in 3DGS-based inverse rendering frameworks, we propose GeoSplatting, a principled solution that augments 3DGS with explicit geometry guidance. Unlike existing 3DGS-based inverse rendering methods that rely on implicit geometric constraints to iteratively approximate Gaussian point normals, GeoSplatting differentiably constructs surface-aligned Gaussian points from an optimizable, explicit mesh. Consequently, GeoSplatting inherently leverages the well-defined mesh normals and the opaque mesh surface for more accurate light transport modeling, resulting in superior material-lighting disentanglement and improved relighting quality.

Specifically, we incorporate isosurface techniques [34] to utilize an optimizable triangular mesh as the geometry guidance. By introducing a Mesh-to-Gaussian adaptor (MGadapter) that differentiably constructs structured Gaussian points grounded on the mesh geometry, we can leverage the explicit mesh normals and the opaque mesh surface to enhance 3DGS with accurate light transport modeling. The fully differentiable nature of MGadapter allows for end-to-end optimization of the geometry guidance during training, ensuring consistency between the mesh and 3DGS. Such consistency further facilitates the use of mesh-based ray tracing techniques for efficient light transport calculations, delivering excellent optimization efficiency while effectively accounting for shadow effects and inter-reflection.

We conduct extensive experiments to validate the effectiveness of our method, demonstrating that GeoSplatting significantly outperforms previous 3DGS-based inverse rendering baselines in terms of geometry accuracy and material-lighting disentanglement while delivering exceptional efficiency compared to both 3DGS-based and implicit-field-based inverse rendering methods.

2. Related Work

2.1. Geometry Modeling for Inverse Rendering

Implicit Field Representations. Neural Radiance Fields (NeRF) [28] pioneered in utilizing implicit neural networks and differentiable volume rendering [7] for high-quality novel view synthesis. Building on this, several works have extended NeRF by modeling implicit surfaces, linking density values in volumetric rendering with Signed Distance Functions (SDF) [32, 38, 41]. While these representations offer well-defined normal directions through SDF gradients, making them promising candidates for inverse rendering tasks [13, 25, 40], their reliance on dense sampling can lead to inefficiencies during both training and inference.

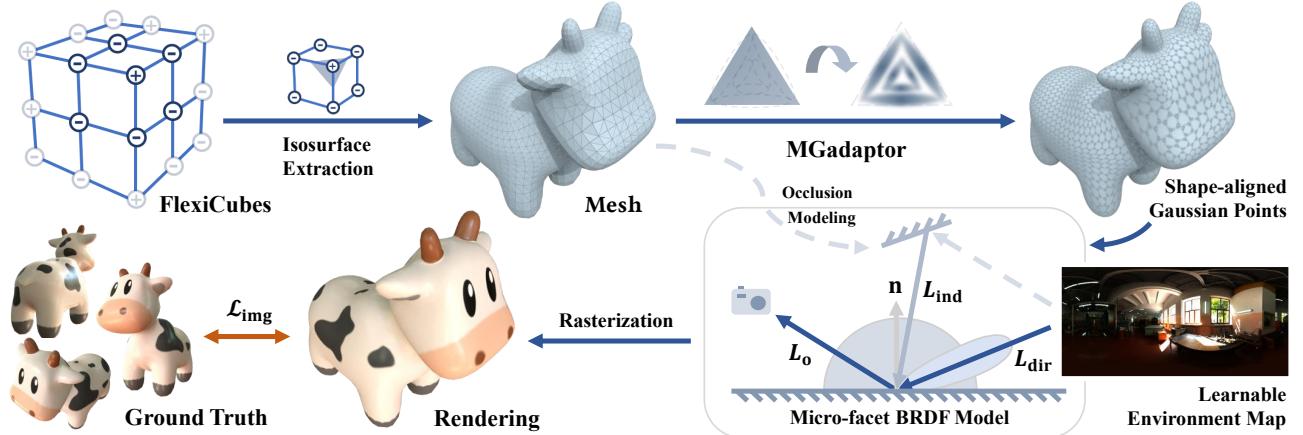


Figure 2. **Pipeline.** GeoSplatting first extracts an intermediate mesh from the scalar field, upon which Gaussian points are sampled and rendered using PBR equations. Finally, they are composited into images through the Gaussian rasterization pipeline. **The entire process is fully differentiable and can be trained end-to-end.**

Mesh Representations. Explicit representations [33, 34] extract triangular meshes from SDF or scalar fields, providing well-defined normal directions and making them well-suited for the Physically-Based Rendering (PBR) framework [10, 31]. However, mesh-based differentiable rendering [4, 5, 22] often produces sparse gradients, primarily concentrated around triangle edges. Consequently, mesh optimization tends to be more challenging than with implicit fields, particularly when handling complex geometries (*e.g.*, thin structures) and high-frequency lighting-material interactions (*e.g.*, specular surfaces).

2.2. 3DGS-based Inverse Rendering

More recently, 3D Gaussian Splatting (3DGS) [17] has emerged as a powerful representation for novel view synthesis, offering significantly superior rendering efficiency compared to NeRF and its variants. However, 3DGS models the scene with Gaussian points which often misalign with the ground truth surface and are prone to artifacts like floaters. To address these issues, some research extends the vanilla 3DGS with geometry enhancement techniques, either by introducing additional regularizations to ensure Gaussian points adhere closely to the surface and remain disk-shaped [3, 6, 9, 11], or by jointly training 3DGS with implicit SDF fields [26, 39, 43].

Inspired by these geometry enhancement techniques, several methods are proposed to further extend the vanilla 3DGS for inverse rendering tasks. For example, R3DG [8] learns additional normal attributes and regularizes normal directions using rendered depth maps. Gaussian-Shader [12] utilizes the shortest axis direction, while GIR [35] employs eigendecomposition to determine surface orientations. These methods also incorporate the Disney BRDF [27] with split-sum simplification [16] and consider indirect lighting through ray tracing or residual terms. However, the approximated normals are less accurate and can be spatially inconsistent, therefore leading to poor de-

composition results, especially for reflective surfaces. In contrast, we propose a unified framework that combines explicit mesh representation with 3DGS, achieving shape consistency between meshes and Gaussian points. This innovation enables explicit modeling of normal directions and enhances Gaussian primitives to better capture light transport, resulting in more effective material and lighting decomposition for inverse rendering tasks.

2.3. Rendering Equation Evaluation

While most inverse rendering approaches rely on the physically-based rendering equation to model material-lighting disentanglement, directly solving the rendering equation, which involves a complex and computationally expensive integral, remains impractical. Instead, many approaches incorporate approximation techniques to accelerate computations. Among these, the split-sum approximation [16] stands out for achieving both high efficiency and excellent render quality and has been utilized in numerous inverse rendering approaches [12, 24, 31]. However, its lack of self-occlusion modeling often leads to inaccurate decomposition, especially under strong shadow effects. In contrast, methods that incorporate Spherical Gaussian (SG) with ray tracing can effectively handle shadow effects and inter-reflection [2, 8, 13, 44, 46], but struggle with reflective surface modeling due to the low-frequency nature of SG.

To address both the need for self-occlusion awareness and reflective surface modeling, our approach employs Monte Carlo sampling to evaluate the rendering equation in a ray tracing manner, building on prior works [10, 25]. Rather than directly conducting ray tracing on 3D Gaussians [8], our approach utilizes the explicit mesh extracted from our isosurface to serve as an effective proxy, benefiting from the more efficient BVH-accelerated ray tracing.

3. Methodology

GeoSplatting effectively bridges 3DGS and mesh represen-

tations by differentiably constructing a set of structured Gaussian points grounded on the mesh surface, leveraging the accurate geometry and normals of the mesh, along with the efficiency and superior rendering capabilities of 3DGS. As illustrated in Fig. 2, it begins by utilizing FlexiCubes [34] to extract a triangular mesh from the learnable iso-value grids. We then introduce a MGAdapter that generates 3DGS points on the mesh surface in a differentiable manner (Sec. 3.1). Next, we present an efficient 3DGS PBR framework, which computes the rendering equation [15] to produce per-Gaussian PBR colors (Sec. 3.2). A BVH-accelerated ray tracing is performed on the extracted mesh to evaluate light transport (Sec. 3.3). Once the Gaussian points are shaded, the 3DGS rasterization algorithm is used to generate the final rendering results. Additionally, we provided implementation details in Sec. 3.4.

3.1. Geometry Guided Gaussian Points Generation

Background. The vanilla 3DGS [18] represents a Gaussian ellipsoid by a full 3D covariance matrix Σ and its center position μ : $G(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)}$, where \mathbf{x} is the location of a 3D point. To ensure a valid positive semi-definite covariance matrix, Σ is decomposed into the scaling matrix \mathbf{S} and the rotation matrix \mathbf{R} that characterizes the geometric shape of a 3D Gaussian. Beyond μ , \mathbf{S} and \mathbf{R} , each Gaussian point maintains additional learnable parameters including Through periodic densification and culling heuristics, 3DGS adaptively controls the distribution of Gaussian points during optimization. While capable of high-fidelity rendering, 3D Gaussians inherently lack deterministic geometry boundaries and explicit normal directions. This significantly complicates light transport computation, thereby hindering their direct application to physically-based inverse rendering tasks.

Method. Our goal is to introduce explicit geometric guidance to 3D Gaussian Splatting. To achieve this, we leverage the isosurface technique FlexiCubes [34] for geometry guidance and propose MGAdapter, which generates Gaussian points guided by the mesh geometry. Starting with a scalar field $\zeta : \mathbb{R}^3 \rightarrow \mathbb{R}$ which maps spatial locations to scalar values, isosurfaces can be represented by discretizing ζ as learnable values stored at grid vertices. We then apply the differentiable isosurface extraction technique, *i.e.*, FlexiCubes, to obtain a triangular mesh M from ζ . With this intermediate mesh M serving as explicit guidance, we propose MGAdapter, formulated as the function \mathcal{T} , to generate Gaussian points attributes $\{(\mu_i, \mathbf{S}_i, \mathbf{R}_i, \mathbf{n}_i)\}$ from each triangles P . Specifically, the MGAdapter produces K Gaussian points for each triangle P :

$$\{(\mu_i, \mathbf{S}_i, \mathbf{R}_i, \mathbf{n}_i) | i = 1, \dots, K\} = \mathcal{T}(P) . \quad (1)$$

\mathcal{T} computes the shape attributes of each generated Gaussian point, including its position $\mu_i \in \mathbb{R}^3$, scale $\mathbf{S}_i \in \mathbb{R}^{3 \times 3}$,

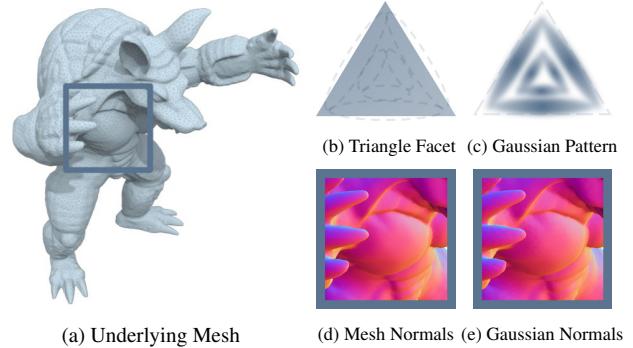


Figure 3. **MGAdapter Overview.** (a) Given the underlying mesh, (b) MGAdapter is applied to each triangle facet, (c) following a predefined pattern to draw six Gaussian points. The pattern is pre-defined in barycentric coordinate space to maintain consistency between (d) the mesh normals and (e) the Gaussian normals without additional optimization, inherently enabling accurate modeling of Gaussian normals.

and rotation $\mathbf{R}_i \in \mathbb{R}^{3 \times 3}$, along with the normal $\mathbf{n}_i \in \mathbb{R}^3$. Note that the shape parameters of the Gaussian points are fully determined by each triangle P . We empirically set $K = 6$ and divide the triangle into 6 sub-regions, placing one Gaussian point in each region (Fig. 3). The position μ_i and normal \mathbf{n}_i are computed via barycentric interpolation, while the scale \mathbf{S}_i and rotation \mathbf{R}_i are decided by the orientation and shape of the triangle P . The opacity is set to 1, as the mesh surface is opaque. This MGAdapter sampling process is fully differentiable, allowing gradients to be effectively propagated to the isosurface ζ . We illustrate the MGAdapter in Fig. 3 and provide the details in Appendix B.

Discussion. Rather than learning approximated normals and supervising with normal-depth consistency loss [8, 12], our surface-grounded Gaussian points leverage the well-defined mesh normals, resulting in improved normal accuracy. This leads to significantly better inverse rendering performance compared to previous 3DGS-based methods that rely on approximated normals. Furthermore, the MGAdapter is specifically designed to produce highly structured Gaussian points that maintain shape consistency relative to the triangular mesh. This shape alignment naturally enables the intermediate mesh M to serve as a proxy for self-occlusion evaluation, facilitating effective light transport modeling for occlusion-aware decomposition, which we explain later in Sec. 3.3.

3.2. Physically-based Gaussian Rendering

Background. The vanilla 3DGS assigns each Gaussian point a k -order spherical harmonic parameter to represent view-dependent rendering effects, which lacks physical interpretability. In contrast, our GeoSplatting represents high-order lighting effects through the physically-based rendering equation [15] (Eq. 2), where the BRDF material is for-

mulated by GGX microfacet model [37] (Eq. 3).

$$\mathbf{L}_o(\mathbf{x}, \omega_o) = \int_{\mathcal{H}^2} \mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o) \mathbf{L}_i(\mathbf{x}, \omega_i) |\mathbf{n} \cdot \omega_i| d\omega_i , \quad (2)$$

$$\mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o) = (1 - m) \frac{\mathbf{a}}{\pi} + \frac{DFG}{4|\mathbf{n} \cdot \omega_i||\mathbf{n} \cdot \omega_o|} . \quad (3)$$

In Eq. 2, the outgoing radiance $\mathbf{L}_o(\mathbf{x}, \omega_o)$ in the direction ω_o is computed as the integral of the BRDF function $\mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o)$, the incoming light $\mathbf{L}_i(\mathbf{x}, \omega_i)$, and the cosine term $|\mathbf{n} \cdot \omega_i|$, which accounts for the angle between the surface normal \mathbf{n} and the incoming light direction ω_i , over the hemisphere \mathcal{H}^2 . In Eq. 3, the GGX model defines the BRDF function $\mathbf{f}_r(\mathbf{x}, \omega_i, \omega_o)$ as two components: the diffuse term $(1 - m) \frac{\mathbf{a}}{\pi}$ and the specular term $\frac{DFG}{4|\mathbf{n} \cdot \omega_i||\mathbf{n} \cdot \omega_o|}$, which are all determined by PBR material attributes, including albedo $\mathbf{a} \in [0, 1]^3$, metalness $m \in [0, 1]$ and roughness $\rho \in [0, 1]$.

Method. Our goal is to assign PBR attributes to Gaussian points and then evaluate the rendering equation to produce PBR colors. Following prior works [10, 31], we introduce multi-resolution hash grids [30] \mathcal{E}_d and \mathcal{E}_s for PBR attribute querying. Specifically, given a Gaussian point with position μ_i , its PBR attributes are generated as follows:

$$\mathbf{a}_i = \mathcal{E}_d(\mu_i) , \quad (\rho_i, m_i) = \mathcal{E}_s(\mu_i) . \quad (4)$$

Discussion. Once PBR attributes are obtained, the BRDF function \mathbf{f}_r is determined. We then differentiably solve Eq. 2 with Monte Carlo sampling to obtain PBR colors for each Gaussian point, which are subsequently processed using standard 3DGS rasterization and rendered into final images via alpha compositing. Subsequently, a backpropagation process is applied to progressively optimize the PBR attributes through photometric supervision.

3.3. Light Transport Modeling

Background. During the evaluation of Eq. 2, accurately modeling the incident lighting $\mathbf{L}_i(\mathbf{x}, \omega_i)$ remains challenging, as it comprises not only the environment illumination but also the indirect illumination from multi-bounce inter-reflection. Effective light transport modeling is required to capture higher-order effects, such as shadowing and inter-reflections. Instead of relying on computationally expensive path tracing techniques for multi-bounce light transport modeling [21, 42], existing approaches often approximate the lighting with a single-bounce model, disentangling $\mathbf{L}_i(\mathbf{x}, \omega_i)$ into two components: the direct term $\mathbf{L}_{dir}(\omega_i)$, representing the environmental illumination, and the indirect term $\mathbf{L}_{ind}(\mathbf{x}, \omega_i)$, accounting for inter-reflection. A scalar factor $O(\mathbf{x}, \omega_i)$ is introduced to model self-occlusion, weighting the ratio between these two lighting components:

$$\mathbf{L}_i(\mathbf{x}, \omega_i) = (1 - O(\mathbf{x}, \omega_i)) \mathbf{L}_{dir}(\omega_i) + O(\mathbf{x}, \omega_i) \mathbf{L}_{ind}(\mathbf{x}, \omega_i) . \quad (5)$$

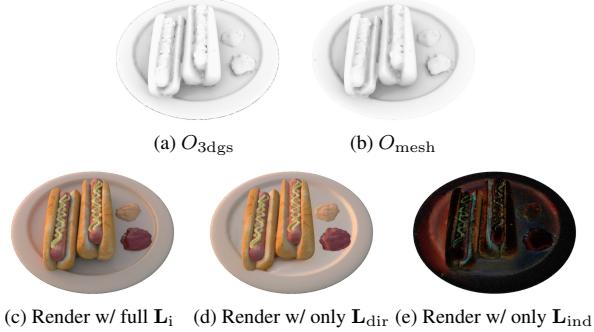


Figure 4. **Self-Occlusion Evaluation.** We visualize the occlusion term evaluated on (a) generated Gaussian points (O_{3dgs}) and (b) the underlying mesh (O_{mesh}), respectively. The Gaussian-mesh shape alignment ensures consistency between O_{3dgs} and O_{mesh} , enabling the mesh to serve as an effective proxy for accelerated self-occlusion evaluation. We additionally illustrate (c) full PBR, (d) PBR without self-occlusion, and (e) Inter-reflection, to demonstrate the effectiveness of our self-occlusion modeling.

Here, the direct term $\mathbf{L}_{dir}(\omega_i)$ is usually represented with a $H \times W \times 3$ environment map which can be queried by latitude and longitude coordinates, while the indirect term is modeled as a set of Spherical Harmonic coefficients.

Method. Our goal is to achieve effective light transport modeling. The key to this lies in estimating the occlusion term $O(\mathbf{x}, \omega_i)$ both accurately and efficiently. Prior 3DGS-based inverse rendering approaches [8, 29] typically model occlusion term as a contiguous form $O_{3dgs}(\mathbf{x}, \omega_i) \in [0, 1]$ and estimate it by accumulating Gaussian opacities along the ray $\mathbf{r}(t) = \mathbf{x} + t\omega_i$, which is computationally expensive and leads to low efficiency. In contrast, by leveraging explicit geometry guidance, we replace $O_{3dgs}(\mathbf{x}, \omega_i) \in [0, 1]$ with a binary term $O_{mesh}(\mathbf{x}, \omega_i) \in \{0, 1\}$, which is estimated on the deterministic mesh surface. This approach enables mesh-based ray tracing, accelerated by BVH, for efficient occlusion evaluation. As illustrated in Fig 4, this replacement does not introduce noticeable errors because our MGadapter ensures shape alignment between the 3DGS and the mesh, thereby maintaining consistency between the approximated $O_{mesh}(\mathbf{x}, \omega_i)$ and the ground truth $O_{3dgs}(\mathbf{x}, \omega_i)$. A more detailed analysis of the error bound can be found in Appendix B.3.

Discussion. With explicit geometry guidance, we propose ray tracing techniques tailored to our hybrid representations for self-occlusion evaluation. Compared to existing methods which also rely on ray tracing for occlusion-aware light transport modeling [8, 13, 25], our solution significantly reduces optimization time, delivering exceptional efficiency.

3.4. Implementation Details

Loss Functions. The pipeline of GeoSplatting is fully differentiable and can be trained end-to-end with photometric

supervision. Thanks to the geometry guidance, we do not require regularization terms, such as dist loss [1] or pseudo depth normal loss [8, 12], to provide additional geometric constraints. However, similar to NVdiffrecmc, GeoSplatting also relies on an object mask loss, as optimizing explicit surface is more challenging. Specifically, the photometric loss is computed as: $\mathcal{L}_{\text{img}} = \mathcal{L}_1 + \lambda_{\text{ssim}} \mathcal{L}_{\text{SSIM}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}$, where $\mathcal{L}_{\text{mask}}$ is an MSE loss between the rendered alpha map and the ground truth object mask. Furthermore, we add an entropy loss to constrain the shape, following DMTet and FlexiCubes [33, 34]. To achieve better decomposition, we apply smoothness regularization on \mathbf{k}_d and \mathbf{k}_s along with a light regularization, following NVdiffrecmc and R3DG [8, 10]. The final loss \mathcal{L} is a combination of the photometric loss and the regularization losses: $\mathcal{L} = \mathcal{L}_{\text{img}} + \lambda_{\text{entropy}} \mathcal{L}_{\text{entropy}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}} + \lambda_{\text{light}} \mathcal{L}_{\text{light}}$. More details are provided in Appendix D.

Split-Sum Warm-up Directly evaluating the integral in Eq. 2 is inefficient and may lead to optimization instability, especially during the initial stages of training when the geometric shape used to model light transport is still noisy. To mitigate this issue, we follow prior works [25] to adopt the split-sum approximation [16] technique during the early training stage. The split-sum technique assumes no self-occlusion, approximating Eq. 2 as follows:

$$\begin{aligned} \mathbf{L}_o(\omega_o) \approx & \int_{\mathcal{H}^2} f_r(\omega_i, \omega_o) |\mathbf{n} \cdot \omega_i| d\omega_i \\ & \cdot \int_{\mathcal{H}^2} \mathbf{L}_{\text{dir}}(\omega_i) D |\mathbf{n} \cdot \omega_i| d\omega_i . \end{aligned} \quad (6)$$

While Eq. 6 enables fast pre-computation, significantly enhancing evaluation speed, its lack of self-occlusion modeling can lead to suboptimal decomposition results (*e.g.*, baking shadow into albedo). Therefore, after this warm-up stage, we transition to our Monte Carlo sampling process for a more accurate evaluation of Eq. 2, incorporating shadow effects and inter-reflection.

Appearance Refinement. GeoSplatting conducts the per-Gaussian shading, *i.e.*, first computes PBR colors on Gaussian points and then rasterizes the per-Gaussian colors into pixels. Since each Gaussian point only delivers a single PBR color, the rendering detail of a surface region is constrained by its Gaussian point density, making it difficult to accurately model high-frequency variations in appearance (*e.g.*, strong reflections). This issue is particularly pronounced for our highly structured Gaussian points. To address it, we consider the deferred shading techniques which conduct pixel-wise PBR for appearance refinement. Similar to existing works [19, 23], we allow GeoSplatting to switch to deferred shading and slightly adjust displacement of Gaussian points at the end of the training, as illustrated in Fig. 5. An ablation study on this refinement can be found in Sec. 4.4 and further details are provided in Appendix E.



(a) Forward Shading

(b) Deferred Shading

Figure 5. **Appearance Refinement.** (a) The rendering details of forward shading are limited by the spatial density of Gaussian points. (b) In contrast, deferred shading conducts pixel-wise PBR, delivering much more detailed results for reflective objects.

4. Experiments

We perform extensive experiments to verify the effectiveness of our inverse rendering method. We first present material decomposition and relighting results on three different datasets in Sec. 4.1. Subsequently, we report normal quality in Sec. 4.2, real-world results in Sec. 4.3, and ablation studies in Sec. 4.4. Our method achieves state-of-the-art inverse rendering performance while demonstrating exceptional efficiency, completing training within 10-15 minutes on a single NVIDIA GTX 4090.

4.1. Inverse Rendering Performance

Datasets & Metrics. We evaluate material decomposition and relighting performance on the Synthetic4Relight dataset [46], the TensoIR Synthetic dataset [13], and the Shiny Blender dataset [36], which cover various challenging scenes with complex non-Lambertian materials, indirect lighting effects, and strong reflection. Following [8], we assess relighting and albedo quality using PSNR, SSIM, and LPIPS metrics, along with roughness MSE, as shown in Table 1. For the Shiny Blender dataset consisting of reflective objects, we follow prior works [12, 25, 36] to evaluate novel view synthesis (NVS) performance.

Performance & Discussion. We compare our method with mesh-based (NVdiffrecmc), implicit-field-based (TensoIR, NeRO), and 3DGS-based (R3DG, GS-Shader, GS-IR) relightable approaches. As shown in Table 1 and Fig. 6, our method achieves state-of-the-art performance in relighting and material assessment, demonstrating superior material and lighting disentanglement, while also delivering exceptional training efficiency.

4.2. Normal Quality

Datasets & Metrics. We conduct normal quality comparison on the TensoIR Synthetic dataset [13] and the Shiny Blender dataset to validate effectiveness of our geometry guidance, covering both diffuse objects and specular objects. Following prior works [13, 36, 46], the normal accuracy is evaluated in terms of Mean Angular Error (MAE).

Dataset		Synthetic4Relight										TensoIR Synthetic										Shiny Blender		
Method	Training minutes↓	Relighting			Albedo			Roughness				Relighting	Albedo	Synthetic			NVS	SSIM ↑	LPIPS ↓					
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓			PSNR ↑	SSIM ↑	LPIPS ↓								
NVdiffe	72	28.89	0.953	0.061	28.66	0.941	0.066	0.026	24.64	0.916	0.078	25.84	0.922	0.096	28.70	0.945	0.119							
NVdiffecmc	82	30.23	0.946	0.080	29.14	0.945	0.067	0.010	26.51	0.923	0.087	27.71	0.931	0.083	28.03	0.932	0.125							
TensoIR	~270	29.94	0.948	0.082	29.60	0.934	0.073	0.015	28.51	0.945	0.059	28.35	0.953	0.059	27.89	0.907	0.156							
NeRO	~800	30.15	0.961	0.052	26.69	0.936	0.067	0.006	26.31	0.933	0.071	25.98	0.908	0.100	29.84	0.962	0.072							
GS-IR	20	23.81	0.902	0.086	26.66	0.936	0.085	0.025	24.35	0.884	0.091	26.80	0.892	0.116	27.01	0.892	0.133							
GS-Shader	63	22.32	0.924	0.084	N/A	N/A	N/A	0.050	22.42	0.872	0.103	N/A	N/A	N/A	30.64	0.952	0.081							
R3DG	~110	31.00	0.964	0.050	28.31	0.951	0.058	0.013	28.55	0.927	0.072	28.74	0.945	0.072	28.83	0.942	0.098							
Ours	14	34.10	0.971	0.037	29.90	0.949	0.062	0.004	29.95	0.943	0.052	29.41	0.941	0.067	31.14	0.955	0.080							

Table 1. **Quantitative Results of Inverse Rendering.** Our method achieves state-of-the-art performance in relighting quality and material recovery for general cases from the Synthetic4Relight and TensoIR Synthetic datasets, along with NVS for reflective cases from the Shiny Blender dataset. Note that GS-Shader does not provide disentangled albedo but rather a diffuse color merged with lighting, so we leave it as N/A. Also, note that we apply the albedo scaling introduced in [45] to perform a fair comparison.

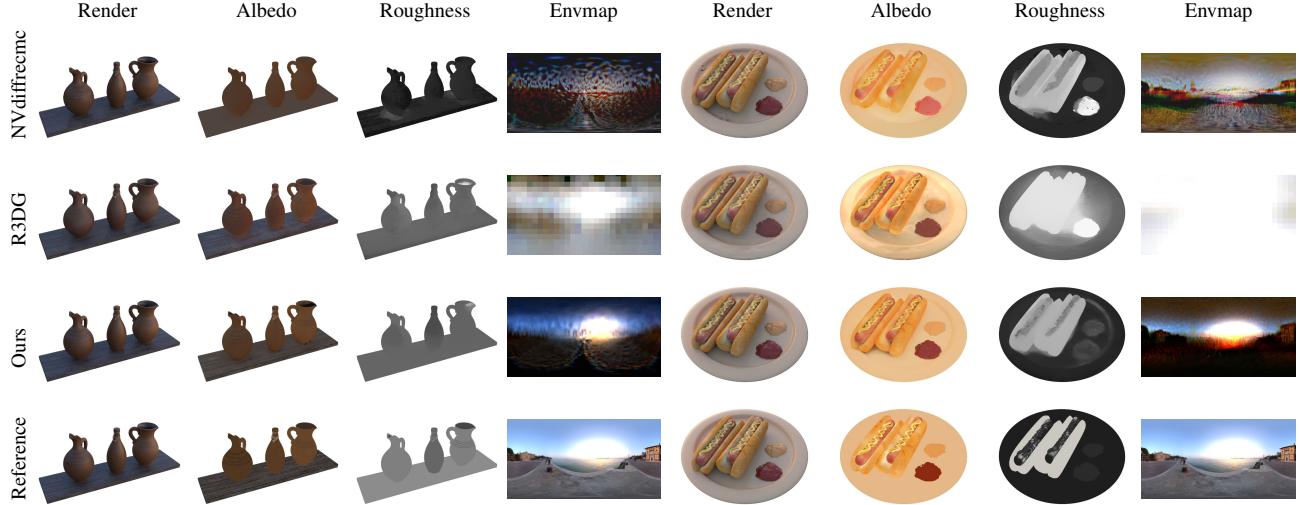


Figure 6. **Qualitative Results from the Synthetic4Relight dataset.** Compared to the baselines, our method achieves superior inverse rendering performance in albedo, roughness, and environment lighting recovery.

Performance & Discussion. As shown in Table 2 and Fig. 7, our method achieves significantly improved normal quality over 3DGS-based baselines which only deliver approximated normals, demonstrating the effectiveness of our geometry guidance. Compared to approaches based on implicit-field, which are carefully optimized for several hours, our method also achieves competitive normal quality, while delivering exceptional training efficiency.

Method	Training Time minutes↓	TensoIR Synthetic		Shiny Blender	
		MAE ↓	MAE ↓	MAE ↓	MAE ↓
Mesh-based	NVdiffe	72	4.97	9.38	
	NVdiffecmc	82	4.81	9.76	
Implicit Field	TensoIR	~270	4.10	4.42	
	NeRO	~800	5.14	2.48	
3DGS-based	GS-IR	20	5.41	4.42	
	GS-Shader	63	6.79	7.03	
	R3DG	~110	5.45	7.04	
	Ours	14	4.08	2.15	

Table 2. **Quantitative Results of Normal Estimation.** Our method achieves state-of-the-art normal quality while delivering exceptional training efficiency.

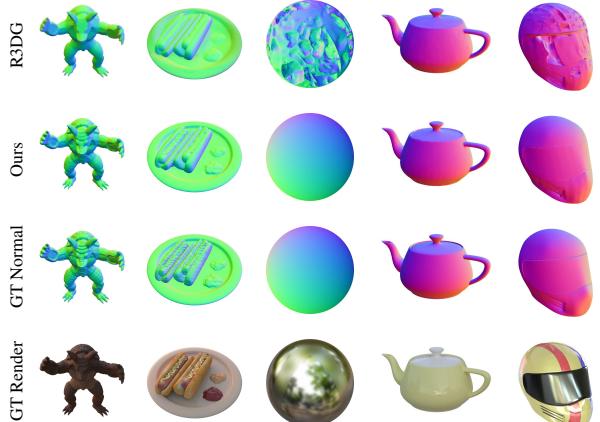


Figure 7. **Qualitative Comparison of Normal Estimation.** Our approach delivers excellent normal quality across various cases, including shadow effects, inter-reflection, and specular surfaces.

4.3. Real-World Results

In Fig. 8, we provide qualitative results of real-world cases from the Stanford-ORB dataset [20], to further validate the effectiveness of our method. More results and discussions for real-world applications can be found in Appendix F.

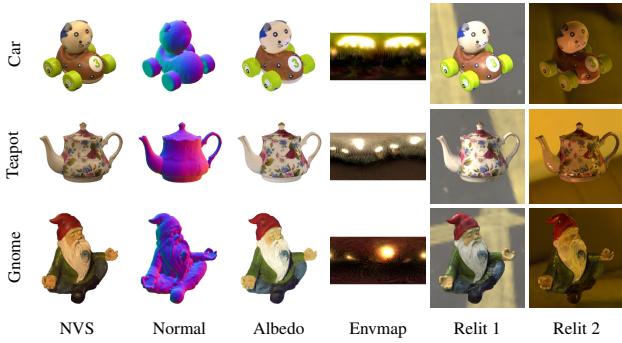


Figure 8. **Qualitative Results from the Stanford-ORB Dataset.** We provide qualitative results including NVS, surface normal, albedo, environment map, and relighting results.

4.4. Ablation Studies

To validate the effectiveness of the aforementioned techniques proposed in this paper, we conduct ablation studies on the TensoIR Synthetic dataset. We compare inverse rendering performance in terms of NVS, relighting, albedo estimation, and normal accuracy, as shown in Table 3.

Description	NVS	Relighting	Albedo	Normal
	PSNR \uparrow	PSNR \uparrow	PSNR \uparrow	MAE \downarrow
w/o shape alignment	35.95	26.39	26.72	8.29
w/o appearance refinement	35.07	27.95	29.31	4.42
w/o occlusion modeling	35.87	27.36	27.80	6.17
w/o indirect light	36.01	28.92	29.18	4.97
full modeling	36.45	29.95	29.41	4.08

Table 3. **Ablation Studies.** We conduct extensive quantitative comparisons to validate the effectiveness of GeoSplatting.

Shape Alignment. Firstly, we conduct an ablation study on our MGadapter by no longer generating shape-consistent Gaussian points from the barycentric pattern. Instead, we randomly sample K Gaussian points on each face and allow them to automatically learn scale and rotation attributes. The quantitative results demonstrate that the shape alignment between Gaussian points and the underlying mesh guidance is crucial to GeoSplatting, as both normal modeling and light transport modeling depend on it.

Appearance Refinement. Next, as confirmed by the quantitative results, appearance refinement effectively enhances rendering quality but has minimal impact on decomposition results and normal quality. This is because the appearance refinement is applied at the end of the training stage, once material and geometry recovery has already converged.

Self-Occlusion Modeling. Lastly, we validate the effectiveness of self-occlusion modeling. By setting $O_{3dgs} = 0$, we evaluate the performance of GeoSplatting without occlusion modeling. Similarly, by setting $L_{ind} = 0$, we assess the performance of GeoSplatting without indirect light. As illustrated in Table 3 and Fig. 9, the occlusion modeling has

little impact on NVS performance but can be crucial for accurate decomposition and relighting.

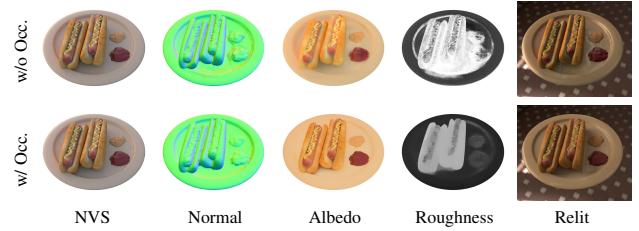


Figure 9. **Qualitative Ablation Studies on Self-Occlusion Modeling.** Despite comparable NVS quality, the absence of occlusion modeling leads to significantly degraded performance, in terms of normal accuracy, material estimation, and relighting quality.

5. Discussion

Limitation. While GeoSplatting demonstrates state-of-the-art performance in NVS and relighting tasks, it still faces several challenges that motivate further research. First, its geometry guidance is derived from the isosurface. Although this significantly improves the geometry performance of 3DGs, it also requires masks during training. Moreover, constrained by the grid resolution of the isosurface, GeoSplatting struggles with thin structures and surfaces with complicated geometry. A promising direction for future work would be to explore how to apply adaptive resolution to accommodate detailed geometry, enabling its extension to scene-level tasks. More comprehensive discussions are provided in Appendix F & G.

Conclusion. We propose GeoSplatting, a novel hybrid representation that enhances 3DGs with explicit geometric guidance and differentiable PBR equations. GeoSplatting achieves superior efficiency and state-of-the-art inverse rendering performance across various datasets, which validates the effectiveness of its geometry guidance. Compared to prior 3DGs-based inverse rendering approaches, GeoSplatting significantly improves the normal quality and the accuracy of light transport modeling, delivering better decomposition results. Moreover, compared to state-of-the-art implicit-field-based approaches which require several hours to optimize, our method achieves competitive performance while converging within 10-15 minutes. We will release all the code to facilitate related research.

Acknowledgement

This work is supported in part by the National Key R&D Program of China under Grants 2022ZD0160801 and is

supported in part by the State Key Laboratory of General Artificial Intelligence. This work is also supported by the projects of Beijing Science and Technology Program (Z231100007423011) and National Natural Science Foundation of China (No. 62376012).

References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. [2](#), [6](#)
- [2] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. [2](#), [3](#)
- [3] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE transactions on visualization and computer graphics*, PP, 2024. [2](#), [3](#)
- [4] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaako Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019. [3](#)
- [5] Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khalis, Or Litany, and Sanja Fidler. DIB-R++: Learning to predict lighting and material with a hybrid differentiable renderer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [3](#)
- [6] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. [2](#), [3](#)
- [7] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, page 65–74, New York, NY, USA, 1988. Association for Computing Machinery. [2](#)
- [8] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv:2311.16043*, 2023. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#)
- [9] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. [2](#), [3](#)
- [10] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022. [2](#), [3](#), [5](#), [6](#)
- [11] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. [2](#), [3](#)
- [12] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023. [2](#), [3](#), [4](#), [6](#)
- [13] Haian Jin, Isabella Liu, Pejjia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023. [2](#), [3](#), [5](#), [6](#)
- [14] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986. [2](#)
- [15] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986. [4](#)
- [16] Brian Karis. Real shading in unreal engine 4. Technical report, Epic Games, 2013. [3](#), [6](#)
- [17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. [2](#), [3](#)
- [18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. [1](#), [2](#), [4](#)
- [19] Ye Keyang, Hou Qiming, and Zhou Kun. 3d gaussian splatting with deferred reflection. 2024. [6](#)
- [20] Zhengfei Kuang, Yunzhi Zhang, Hong-Xing Yu, Samir Agarwala, Elliott Wu, Jiajun Wu, et al. Stanford-orb: a real-world 3d object inverse rendering benchmark. *Advances in Neural Information Processing Systems*, 36, 2024. [7](#)
- [21] Eric P. Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Department of Computer Science, KU Leuven, Celestijnlaan 200A, 3001 Heverlee, Belgium, 1995. [5](#)
- [22] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. [2](#), [3](#)
- [23] Zhihao Liang, Hongdong Li, Kui Jia, Kailing Guo, and Qi Zhang. Gus-ir: Gaussian splatting with unified shading for inverse rendering. *arXiv preprint arXiv:2411.07478*, 2024. [6](#)
- [24] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21644–21653, 2024. [1](#), [2](#), [3](#)
- [25] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Transactions on Graphics (TOG)*, 42(4):1–22, 2023. [2](#), [3](#), [5](#), [6](#)
- [26] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. *ACM Transactions on Graphics (TOG)*, 43(6):1–12, 2024. [2](#), [3](#)
- [27] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez.

- Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*, New York, NY, USA, 2012. Association for Computing Machinery. 3
- [28] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, 2020. 2
- [29] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Ricardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics (ToG)*, 43(6):1–19, 2024. 5
- [30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 5
- [31] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8280–8290, 2022. 2, 3, 5
- [32] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2
- [33] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3, 6
- [34] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.*, 42(4), 2023. 2, 3, 4, 6
- [35] Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jingtuo Liu, Liangjun Zhang, Jian Zhang, Bin Zhou, Errui Ding, and Jingdong Wang. Gir: 3d gaussian inverse rendering for relightable scene factorization. *Arxiv*, 2023. 2, 3
- [36] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 6
- [37] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, page 195–206, Goslar, DEU, 2007. Eurographics Association. 5
- [38] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 2
- [39] Haodong Xiang, Xinghui Li, Xiansong Lai, Wanting Zhang, Zhichao Liao, Kai Cheng, and Xueping Liu. Gaussian-room: Improving 3d gaussian splatting with sdf guidance and monocular cues for indoor scene reconstruction, 2024. 2, 3
- [40] Ziyi Yang, Yanzhen Chen, Xinyu Gao, Yazhen Yuan, Yu Wu, Xiaowei Zhou, and Xiaogang Jin. Sire-ir: Inverse rendering for brdf reconstruction with shadow and illumination removal in high-illuminance scenes. *arXiv preprint arXiv:2310.13030*, 2023. 2
- [41] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2
- [42] Bohan Yu, Siqi Yang, Xuanning Cui, Siyan Dong, Baoquan Chen, and Boxin Shi. Milo: Multi-bounce inverse rendering for indoor scene with light-emitting objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10129–10142, 2023. 5
- [43] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf for improved rendering and reconstruction, 2024. 2, 3
- [44] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [45] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021. 2, 7
- [46] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 3, 6