

# Path-Traced Inverse Rendering with Global Illumination in 3D Gaussian Fields

JUNKE ZHU, University of Science and Technology of China, China

HAO ZHANG, Peking University, China

YUTIAN ZHU, University of Science and Technology of China, China

ANG LI, CHENXIAO HU, MENG GAI, and FEI ZHU, Peking University, China

ZHANGJIN HUANG\*, University of Science and Technology of China, China

SHENG LI\*, Peking University, China



Fig. 1. Our splatting-free framework ensures consistency between forward path-traced rendering and backward optimization. We demonstrate recovered 3D Gaussian assets under physically based multi-bounce global illumination in a heterogeneous scene, where all objects except the light sources, walls, and metallic box are modeled as 3D Gaussian primitives. This allows the recovered Gaussian materials to be reused for path-traced rendering and relighting.

Ray tracing enables 3D Gaussian fields to serve as a representation for physically based light transport. Faithful inverse rendering requires forward rendering and backward optimization to be defined within a consistent light-transport pipeline. Existing Gaussian inverse-rendering methods typically rely on splatting-derived G-buffers and screen-space optimization. The local affine approximation of perspective projection can introduce inconsistencies with path tracing, while simplified rendering formulations often neglect or approximate indirect illumination and visibility. Therefore, we propose a

\*Sheng Li and Zhangjin Huang are both corresponding authors.

Authors' Contact Information: Junke Zhu, junkezhu@mail.ustc.edu.cn, University of Science and Technology of China, China; Hao Zhang, zhanghao25@stu.pku.edu.cn, Peking University, China; Yutian Zhu, zhuyutian@mail.ustc.edu.cn, University of Science and Technology of China, China; Ang Li, psnalaalansp@gmail.com; Chenxiao Hu, hineven@pku.edu.cn; Meng Gai, gaimeng@pku.edu.cn; Fei Zhu, feizhu@pku.edu.cn, Peking University, China; Zhangjin Huang, zhuang@ustc.edu.cn, University of Science and Technology of China, China; Sheng Li, lisheng@pku.edu.cn, Peking University, China.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

© 2026 Copyright held by the owner/author(s).

ACM 1557-7368/2026/12-ART235

<https://doi.org/10.1145/3842545>

splatting-free path-traced inverse-rendering framework for 3D Gaussian fields that unifies forward rendering and backward optimization within the same recursive multi-bounce light-transport pipeline. We formulate light transport over overlapping Gaussian primitives directly in path space, enabling Monte Carlo path tracing and pathwise gradient replay over the same recursively sampled transport paths. The framework jointly optimizes materials and environment light under the full rendering equation, with ray-traced visibility and global illumination explicitly evaluated. Extensive experiments demonstrate competitive material estimation and improved path-traced rendering quality, producing more plausible shadows, reflections, and relighting under global illumination. The code is available at [https://junkzhu.github.io/project\\_pages/PTIR](https://junkzhu.github.io/project_pages/PTIR).

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

Additional Key Words and Phrases: 3D Gaussian, Inverse rendering, Ray tracing, Global illumination, Relighting, Material recovery

## ACM Reference Format:

Junke Zhu, Hao Zhang, Yutian Zhu, Ang Li, Chenxiao Hu, Meng Gai, Fei Zhu, Zhangjin Huang, and Sheng Li. 2026. Path-Traced Inverse Rendering with Global Illumination in 3D Gaussian Fields. *ACM Trans. Graph.* 45, 6, Article 235 (December 2026), 22 pages. <https://doi.org/10.1145/3842545>

## 1 INTRODUCTION

3D Gaussian Splatting (3DGS) has achieved remarkable success in novel view synthesis [Kerbl et al. 2023], becoming an important representation for 3D scene perception and understanding. Most 3DGS methods rely on rasterization, projecting Gaussian primitives onto the image plane and compositing them through ordered alpha blending. This formulation is highly effective for image-space reconstruction, but physically based inverse rendering requires recovered materials to remain valid under light transport involving ray-space visibility and indirect illumination. Ray tracing provides a physically based rendering paradigm for 3D Gaussian fields by treating Gaussians as scene primitives in light transport simulation, naturally supporting shadows, reflections, and indirect illumination that remain challenging to model in splatting-based pipelines [Hu et al. 2026; Moenne-Loccoz et al. 2024; Xie et al. 2024].

As illustrated in Fig. 2, existing inverse-rendering methods for Gaussian representations typically estimate physically motivated properties from splatting-derived G-buffers optimized with screen-space shading objectives. However, splatting sorts Gaussians according to their center depths [Gao et al. 2024], whereas ray tracing determines interactions from their peak responses along each ray. In addition to the depth inconsistency, projecting a 3D Gaussian into a 2D splat through a local affine approximation can introduce additional errors. Consequently, naively integrating materials inverted with splatting-based methods into a path-traced rendering pipeline can lead to severe artifacts, revealing that the recovered properties remain tied to screen-space optimization rather than path-space transport. Moreover, these methods typically rely on simplified rendering equations. Without multi-bounce path tracing, indirect illumination is often neglected, while visibility effects such as shadows are approximated using precomputed or rasterization-based terms.

We therefore introduce a splatting-free inverse-rendering framework based on a path-space equivalent interaction model for 3D Gaussian fields. Under this model, forward Monte Carlo path tracing and backward optimization are performed over the same ray-traced Gaussian interactions without relying on splatting. We construct equivalent surface interactions in ray space, evaluate visibility through ray tracing, and perform multi-bounce path tracing under the full rendering equation. Gradients are propagated by replaying the same interactions using path replay backpropagation (PRB) [Vicini et al. 2021]. Multiple importance sampling and a compact Spherical-Gaussian environment parameterization are used for variance reduction and stable lighting optimization. Experiments demonstrate competitive material estimation quality together with improved consistency in path-traced rendering, relighting, and appearance editing. The main contributions of our work are as follows:

- We propose a splatting-free path-traced inverse rendering framework for 3D Gaussian fields, where forward rendering and backward optimization are defined within the same ray-tracing pipeline, avoiding the mismatch between splatting-based material estimation and path-traced rendering.
- We formulate path-space light transport for 3D Gaussian fields under the full rendering equation, where equivalent surface

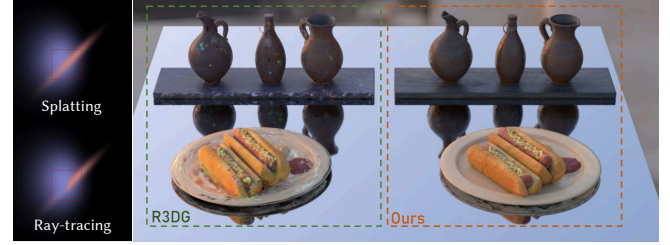


Fig. 2. Path-traced evaluation of inverse-rendered 3D Gaussian assets. (Left) Splatting orders Gaussians by their center depths, whereas ray tracing determines effective depths from peak-response locations along each ray. (Right) R3DG [Gao et al. 2024] optimizes materials within a splatting-based pipeline, resulting in inconsistencies under path-traced rendering, while our framework produces more consistent results.

interactions define shading states for material and illumination optimization through recursive multi-bounce transport.

- We experimentally demonstrate that material and illumination can be recovered in 3D Gaussian fields within a ray-traced light-transport pipeline, achieving competitive inverse-rendering quality while enabling physically based relighting and appearance editing under global illumination.

## 2 RELATED WORK

### 2.1 Ray Tracing of Gaussian Primitives

3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] represents radiance fields with anisotropic 3D Gaussian primitives and enables real-time novel-view synthesis through tile-based splatting. However, its screen-space compositing and spherical-harmonic view-dependent appearance limit effects involving incoherent light paths. HTGS [Hahlbohm et al. 2025] performs perspective-correct Gaussian evaluation along per-pixel viewing rays and introduces hybrid transparency to reduce depth-ordering artifacts. 3DGUT [Wu et al. 2025] employs an unscented-transform formulation for nonlinear camera models and supports hybrid rendering with traced secondary rays. These methods retain rasterized primary visibility, whereas fully ray-traced formulations process primary and recursively generated rays within a unified ray-tracing pipeline. For fully ray-traced rendering, 3D Gaussian Ray Tracing (3DGRT) [Moenne-Loccoz et al. 2024] approximates Gaussians with icosahedral mesh proxies and constructs a scene-level BVH for hardware-accelerated differentiable ray tracing. Similarly, Byrski et al. [2025] use ellipsoidal proxies for accurate ray-primitive intersections. Several works [Hu et al. 2026; Sun et al. 2025b] improve efficiency with probabilistic ray-intersection tests that reduce ray payload overhead and avoid explicit sorting. Another line of research [Condor et al. 2025; Zhou et al. 2024] reformulates Gaussian light transport for volumetric rendering. EnvGS [Xie et al. 2024] models environmental reflections by ray tracing Gaussians visible only through indirect paths. Beyond 3D Gaussians, 2D Gaussian Splatting (2DGS) [Huang et al. 2024] improves geometry with anisotropic 2D primitives, while IRGS [Gu et al. 2025] introduces 2D Gaussian ray tracing for inter-reflections. Poirier-Ginter et al. [2025] use ray tracing for physically based specular reflections.



## 2.2 Inverse Rendering of 3D Gaussians

Reconstructing scene geometry, materials, and illumination from multi-view images via inverse rendering remains highly challenging. Recent advances in 3DGS have inspired several works to integrate traditional shading models into the 3D Gaussian framework to disentangle radiance into explicit material and lighting components [Guo et al. 2024; Jiang et al. 2024; Sun et al. 2025a; Wu et al. 2024]. For relighting, Gao et al. [2024] bake occlusion and indirect illumination via point-based ray tracing, while Liang et al. [2024] adopt a split-sum approximation of the rendering equation and store pre-computed occlusion and indirect terms in volumes. In contrast, Bi et al. [2024] and Fan et al. [2025] bypass explicit lighting models and instead regress global illumination effects using neural networks, which is particularly effective for challenging structures such as hair. Gu et al. [2025] proposed a 2D Gaussian-based ray-tracing approach to efficiently sample incident radiance and compute inter-reflections. Jiang et al. [2025] extended classical radiosity to Gaussian surfels and formulated a differentiable solution for global light transport; however, due to the low-frequency nature of radiosity, their method cannot faithfully capture high-frequency illumination effects. Poirier-Ginter et al. [2025] optimize materials in 3D Gaussian scenes using path tracing driven by multi-view G-buffers. Yang et al. [2026] recover diffuse albedo through single-bounce optimization using a reconstructed radiance cache and apply multi-bounce path tracing after scene editing.

Environmental illumination is a key component of inverse rendering and has been represented by image-based lighting [Debevec 1998], spherical harmonics [Ramamoorthi and Hanrahan 2001], spherical Gaussians [Wang et al. 2009], and Haar wavelets [Ng et al. 2003]. Existing methods either explicitly optimize lighting representations or implicitly encode illumination in neural fields [Du et al. 2024; Ling et al. 2024; Yao et al. 2022; Zhang et al. 2023]. Several works approximate the rendering equation through closed-form SG integration [Zhang et al. 2021, 2022], whereas TensoIR [Jin et al. 2023] evaluates environment illumination by ray sampling, which naturally aligns with our ray-traced formulation.

Our method follows a different formulation from these approaches. Existing Gaussian inverse-rendering methods typically remain coupled with splatting-based material buffers, screen-space visibility, precomputed transport terms, or use ray tracing only as an auxiliary component for selected effects. In contrast, we consider a splatting-free, perspective-correct path-tracing setting, where material recovery and relighting are performed under multi-bounce global illumination without precomputed light-transport approximations.

Concurrent and independent work, SRT-GS [Xu et al. 2026], introduces a sorting-free stochastic estimator for differentiable ray-traced 3DGS reconstruction and extends it to a neural relightable representation. It propagates gradients through Monte Carlo-sampled Gaussian interactions, while our framework uses path replay to optimize explicit material parameters through BRDF evaluations along recursively sampled multi-bounce transport paths. Its stochastic estimator could potentially accelerate the ray-Gaussian evaluation component of our pipeline.

## 3 METHOD

We first introduce the necessary background definitions (Sec. 3.1). We then present path-space light transport for 3D Gaussian fields with path-traced global illumination (Sec. 3.2). Next, we describe the learnable Spherical-Gaussian environment light (Sec. 3.3). Finally, we formulate inverse rendering by replaying the same path-space interactions for gradient propagation (Sec. 3.4).

### 3.1 Preliminary

**3.1.1 3D Gaussian Primitives.** Each Gaussian primitive is further associated with material attributes, including shading normal  $\mathbf{n} \in \mathbb{R}^3$ , albedo  $\mathbf{a} \in \mathbb{R}^3$ , roughness  $r \in \mathbb{R}$ , and metallic  $m \in \mathbb{R}$ , in addition to the vanilla 3D Gaussian parameters [Kerbl et al. 2023].

For each Gaussian primitive intersected by a ray, we first identify its maximum-response point along the ray  $r(t) = \mathbf{o} + t\mathbf{d}$ . This point corresponds to the peak of the Gaussian response along the ray and admits a closed-form solution [Moenne-Loccoz et al. 2024]:

$$t_{\text{peak}}^{(i)} = -\frac{\mathbf{d}^\top \Sigma_i^{-1}(\mathbf{o} - \boldsymbol{\mu}_i)}{\mathbf{d}^\top \Sigma_i^{-1} \mathbf{d}}, \quad \mathbf{p}_{\text{peak}}^{(i)} = r\left(t_{\text{peak}}^{(i)}\right), \quad (1)$$

where  $\boldsymbol{\mu}_i$  denotes the center of the  $i$ -th Gaussian primitive and  $\Sigma_i$  its covariance matrix encoding anisotropic shape. Since Eq. (1) is analytic, both  $t_{\text{peak}}^{(i)}$  and  $\mathbf{p}_{\text{peak}}^{(i)}$  are differentiable with respect to the ray parameters and Gaussian parameters, enabling gradient propagation through the response query. At this point, we evaluate both the Gaussian response and the associated local material attributes of the primitive. The effective opacity is defined as

$$\alpha_i = \sigma_i \rho\left(\mathbf{p}_{\text{peak}}^{(i)}\right), \quad (2)$$

where  $\sigma_i$  denotes the opacity parameter of the  $i$ -th Gaussian primitive, and  $\rho$  is the Gaussian density evaluated at its maximum-response point. This quantity is later used to determine interaction validity and to weight material aggregation.

**3.1.2 Rendering Equation.** Consistent with prior Gaussian inverse rendering works [Gao et al. 2024; Gu et al. 2025; Sun et al. 2025a], we focus on opaque, surface-dominated scenes. Given an equivalent surface interaction at  $\mathbf{x}$  and an outgoing direction  $\boldsymbol{\omega}_o$ , the outgoing radiance is described by the classical surface rendering equation [Kajiya 1986]:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = L_e(\mathbf{x}, \boldsymbol{\omega}_o) + \int_{\Omega} f_r(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\mathbf{x}, \boldsymbol{\omega}_i) (\mathbf{n} \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \quad (3)$$

where  $L_o$ ,  $L_i$ , and  $L_e$  denote the outgoing, incident, and emitted radiance, respectively;  $\boldsymbol{\omega}_i$  and  $\boldsymbol{\omega}_o$  denote the incident and outgoing directions; and  $\Omega$  denotes the upper hemisphere defined by the shading normal  $\mathbf{n}$ . Since we assume non-emissive Gaussians,  $L_e = 0$  at Gaussian surface interactions. The function  $f_r$  denotes the bidirectional reflectance distribution function (BRDF). We adopt a simplified Disney BRDF [Burley 2012], parameterized by diffuse albedo  $\mathbf{a}$ , roughness  $r$ , and metallic value  $m$ . The BRDF in Eq. (3) combines a Lambertian diffuse term  $f_d = \frac{\mathbf{a}}{\pi}$  and a Cook-Torrance specular term  $f_s$ :

$$f_s(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{DFG}{4(\mathbf{n} \cdot \boldsymbol{\omega}_i)(\mathbf{n} \cdot \boldsymbol{\omega}_o)}, \quad (4)$$

where  $D$ ,  $F$ , and  $G$  denote the GGX normal distribution function, the Schlick Fresnel term, and the Schlick-GGX geometry term, respectively.

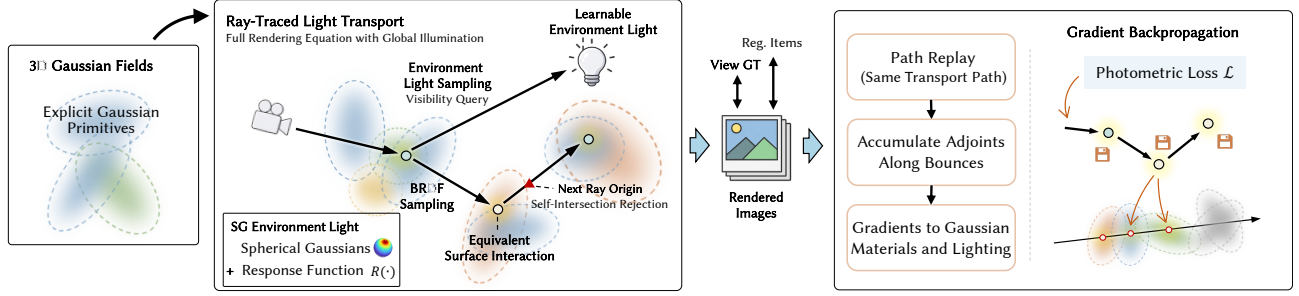


Fig. 3. Our framework jointly defines forward rendering and backward optimization for 3D Gaussian fields within a unified ray-tracing pipeline. Given a 3D Gaussian scene, we construct ray-traced equivalent surface interactions, evaluate visibility and multi-bounce path-traced light transport under the full rendering equation, and support optimization of environment illumination using a compact Spherical-Gaussian representation. Multiple importance sampling reduces the variance of incident radiance estimation. Gradients are propagated through path replay backpropagation via equivalent interactions, ensuring consistency between backward optimization and the forward ray-traced transport paths.

Applying Eq. (3) to 3D Gaussian fields requires defining both surface interactions and light transport in path space. Instead of querying an explicit surface, each scattering event is constructed from the responses of overlapping Gaussians for BRDF evaluation. The rendering integral is then estimated through stochastic sampling and ray-traced visibility, while recursive ray propagation accounts for multi-bounce indirect illumination. This formulation provides a common physical basis for both path-traced rendering and the subsequent inverse optimization.

### 3.2 Ray-Traced Light Transport in 3D Gaussian Fields

Path tracing naturally models global illumination through multi-bounce light transport. Ray tracing provides a unified way to evaluate surface interactions, visibility, and secondary rays in ray space, which remains challenging for purely rasterization-based pipelines.

We trace rays directly through 3D Gaussian fields in a perspective-correct manner, rather than relying on the affine projection used in splatting. Our formulation targets solid objects, where Gaussian primitives are interpreted as surface-oriented elements rather than volumetric media such as smoke, as considered by the VPPT integrator of Condor et al. [2025]. Under this assumption, the Gaussian field forms a thin-shell surface representation, allowing ray-surface interactions to be modeled without treating the field as a participating medium or sampling volumetric scattering events along the ray.

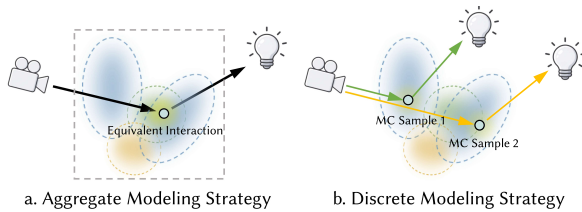


Fig. 4. Different modeling strategies for ray-Gaussian interactions. (a) The aggregate strategy treats overlapping Gaussians as a single equivalent interaction. (b) The discrete strategy samples one primitive according to the opacity responses and treats it as the interaction.

3D Gaussians are semi-explicit primitives: a local surface is typically represented by multiple overlapping Gaussians, so a ray interaction cannot always be assigned to a single primitive. In this work, we adopt an aggregated-surface interpretation (Fig. 4(a)), widely used and empirically supported in splatting-based methods, in which nearby contributing Gaussians jointly define an equivalent interaction state. Ray spawning and shading are then performed from this representative state, rather than from a stochastically selected primitive. A discrete micro-surface interpretation (Fig. 4(b)) is also compatible with our framework by replacing the aggregation with stochastic selection of a contributing Gaussian primitive, but we found it more sensitive to local geometric inconsistencies in practice (Fig. 13). Path replay can propagate gradients through the selected interaction, so stochastic selection does not introduce additional nondifferentiability.

**3.2.1 Path-Space Equivalent Surface Interaction.** Under the above assumption, our goal is to define a stable path-space representative interaction for the local aggregated surface encountered by the ray. This interaction provides a common path-space state for BRDF evaluation, ray spawning, and gradient replay.

**Interaction Validity.** Monte Carlo path tracing inherently introduces stochastic variance in the transport estimator. In 3D Gaussian fields, however, a ray may additionally accumulate weak responses from Gaussian tails or unstable overlaps that do not correspond to stable local surface interactions. If such unreliable responses are directly treated as valid hits, they can lead to unstable interaction depth and material estimates, and this instability further propagates through multi-bounce transport.

Therefore, we enforce a geometric validity criterion based on the accumulated contribution weight along the ray, i.e., the predicted opacity. A candidate path-space interaction is valid only when this weight exceeds a geometry threshold:

$$\chi_{\text{geom}} = \mathbb{I}\left(\sum_{i=1}^N T_i \alpha_i > \tau_g\right), \quad (5)$$

where  $\chi_{\text{geom}} \in \{0, 1\}$  denotes a binary indicator of geometric validity,  $T_i$  is the accumulated transmittance up to the  $i$ -th Gaussian along the ray,  $\alpha_i$  is the effective opacity of the  $i$ -th Gaussian evaluated at

the ray interaction location, and  $\tau_g$  is set to 0.5 by default. Since the Gaussian geometry is constructed using the same response-based accumulation mechanism during training and is further regularized by a mask entropy loss that encourages foreground opacity toward one, this criterion does not introduce noticeable discontinuities during inverse rendering in practice.

**Interaction State.** Conditioned on a valid path-space interaction  $\mathbf{p}$ , we then define an equivalent interaction state  $\mathbf{s}_p$  that represents the aggregate contribution of the Gaussian primitives. Since this state is intended to represent the local surface configuration rather than the accumulated response magnitude itself, we use normalized contribution weights to aggregate the contributing primitives. Accordingly, the representative interaction depth is defined as

$$\mathcal{D} = \mathbb{E}[d \mid \chi_{\text{geom}} = 1] = \sum_{i=1}^N w_i t_{\text{peak}}^{(i)}, \quad w_i = \frac{T_i \alpha_i}{\sum_{j=1}^N T_j \alpha_j}. \quad (6)$$

Under the valid-hit condition,  $w_i t_{\text{peak}}^{(i)}$  defines a conditional distribution over the contributing Gaussian primitives. Accordingly,  $\mathcal{D}$  serves as the representative depth of the aggregated interaction.

Using the same normalized contribution weights, we further aggregate the local material attributes to obtain the effective interaction state. Specifically, the effective shading normal  $\mathcal{N} \in \mathbb{R}^3$ , albedo  $\mathcal{A} \in \mathbb{R}^3$ , roughness  $\mathcal{R} \in \mathbb{R}$ , and metallic  $\mathcal{M} \in \mathbb{R}$  are defined as

$$\{\mathcal{N}, \mathcal{A}, \mathcal{R}, \mathcal{M}\} = \sum_{i=1}^N w_i \{\mathbf{n}_i, \mathbf{a}_i, r_i, m_i\}, \quad w_i = \frac{T_i \alpha_i}{\sum_{j=1}^N T_j \alpha_j}. \quad (7)$$

The aggregated normal  $\mathcal{N}$  is normalized before BRDF evaluation. The contribution-weighted attributes define the path-space shading state of our equivalent-surface model, under which the induced light-transport integral is evaluated using Monte Carlo path tracing.

**3.2.2 Ray Spawning and Visibility Test.** We use multiple importance sampling (MIS) [Veach 1997] to estimate incident radiance at each equivalent surface interaction, which reduces variance by combining environment-light sampling and BRDF importance sampling. At the  $i$ -th bounce, the incident radiance term in Eq. (8) can be compactly expressed as

$$\hat{L}_i = \underbrace{w_{\text{env}}^{(i)} \frac{V(\mathbf{p}_i, \boldsymbol{\omega}_{\text{env}}^{(i)}) L_{\text{env}}(\boldsymbol{\ell}, \boldsymbol{\omega}_{\text{env}}^{(i)})}{p_{\text{env}}(\boldsymbol{\omega}_{\text{env}}^{(i)})}}_{\text{Environment-sampled Query}} + \underbrace{w_{\text{brdf}}^{(i)} \frac{\hat{L}_{i+1}(\mathbf{r}_{i+1})}{p_{\text{brdf}}(\boldsymbol{\omega}_{\text{brdf}}^{(i)} \mid \mathbf{s}_{\mathbf{p}_i})}}_{\text{BRDF-sampled Query}}, \quad (8)$$

where  $w_{\text{env}}^{(i)}$  and  $w_{\text{brdf}}^{(i)}$  are the MIS weights,  $V(\cdot)$  denotes the visibility estimated by the shadow-ray test, and  $\hat{L}_{i+1}(\mathbf{r}_{i+1})$  is the radiance carried by the next-bounce ray  $\mathbf{r}_{i+1}$ .

**Shadow Ray Test:** We estimate shadow-ray visibility  $V(\cdot)$  using a stochastic transmittance test in the any-hit shader, following Hu et al. [2026]. This yields a reliable estimate of transmittance while allowing early termination in high-opacity regions for improved efficiency.

**Next-Ray Event Determination:** The BRDF-sampled term in Eq. (8) is evaluated by tracing the spawned ray  $\mathbf{r}_{i+1}$  and determining the next transport event. For continuation rays, the stochastic transmittance test used for shadow rays must be coupled with a geometric validity check, since weak Gaussian-tail responses should not be treated as valid secondary surface interactions. Accordingly, we augment stochastic transmittance continuation with an additional geometric validity criterion:

$$\chi_{\text{pass}} = \mathbb{I}(\chi_{\text{geom}} = 0) \mathbb{I}(\xi < T_{\text{ray}}), \quad \xi \sim \mathcal{U}(0, 1), \quad (9)$$

where  $\chi_{\text{pass}} \in \{0, 1\}$  indicates whether the spawned ray passes through the Gaussian field when no valid subsequent interaction is formed,  $\chi_{\text{geom}}$  is the geometric validity indicator defined in Eq. (5), and  $T_{\text{ray}}$  denotes the accumulated transmittance along the ray.

**Self-Intersection Rejection:** A direct consequence of representing a surface by overlapping Gaussian primitives is that rays spawned from an equivalent interaction may immediately encounter residual responses from primitives that already contributed to the same local surface. Such responses do not correspond to a new transport event, but to repeated intersections with the same aggregated thin-shell surface, and can therefore introduce spurious self-occlusion for both shadow rays and continuation rays.

A fixed origin offset treats self-intersection as a purely numerical issue and does not generalize well across models, since the appropriate offset magnitude depends on object scale and local Gaussian support thickness. We therefore introduce a backface-aware origin-offset strategy that classifies local back-facing responses along the spawned ray as self-intersections and advances the ray origin past the corresponding Gaussian peak:

$$\hat{\mathbf{o}} = \mathbf{o} + \chi_{\text{self}} (t_{\text{peak}} + \epsilon) \mathbf{d}, \quad \chi_{\text{self}} = \mathbb{I}(t_{\text{peak}} > 0) \mathbb{I}(\mathbf{d}^\top \mathbf{n} > 0). \quad (10)$$

Here  $\hat{\mathbf{o}}$  denotes the updated ray origin,  $t_{\text{peak}}$  is defined in Eq. (1),  $\epsilon$  is a small positive constant for numerical robustness, and  $\mathbf{n}$  denotes the shading normal of the candidate Gaussian primitive. The condition  $\mathbf{d}^\top \mathbf{n} > 0$  identifies a back-facing response relative to the spawned ray, which is treated as a repeated hit of the current aggregated surface rather than a valid occluding interaction.

### 3.3 Learnable Environment Light

Incident directions sampled from the environment illumination directly participate in BRDF evaluation and can affect gradient variance. In addition to conventional 2D environment maps or cube maps, we optionally parameterize environment illumination using a compact set of  $N_{\text{sg}}$  Spherical Gaussians (SGs). This continuous and differentiable representation is compatible with Monte Carlo ray sampling and multiple importance sampling (MIS), while its compact parameter space yields smoother optimization and improved stability under low-SPP training.

The SG lobe directions  $\{\boldsymbol{\eta}_i\}_{i=1}^{N_{\text{sg}}}$  are initialized uniformly over the unit sphere via Fibonacci sampling [González 2009], where  $\boldsymbol{\eta}_i \in \mathbb{S}^2$  denotes the unit direction of the  $i$ -th SG lobe. The radiance along direction  $\mathbf{v}$  is modeled as

$$L(\mathbf{v}) = \sum_{i=1}^{N_{\text{sg}}} c_i \exp(\lambda_i (\boldsymbol{\eta}_i \cdot \mathbf{v} - 1)), \quad c_i = R(\mathbf{x}_i; \gamma), \quad (11)$$

where  $\mathbf{x}_i \in \mathbb{R}^3$  denotes the unconstrained RGB amplitude parameter of the  $i$ -th SG lobe,  $\lambda_i$  controls its sharpness, and  $R(\mathbf{x}; \gamma) = \exp(\text{softplus}(\mathbf{x})/\gamma) - 1$  maps raw parameters to non-negative HDR radiance values. This compact parameterization preserves high-dynamic-range illumination, avoids invalid negative radiance, and stabilizes environment-light optimization in our ray-traced inverse rendering framework.

### 3.4 Inverse Rendering

**3.4.1 Problem Formulation.** Building upon the path-tracing framework defined above, we formulate physically based rendering with



multiple importance sampling (MIS) over 3D Gaussian fields. A camera ray is denoted as  $\mathbf{r}(\mathbf{o}, \mathbf{d})$ , where  $\mathbf{o}$  and  $\mathbf{d}$  denote the ray origin and direction, respectively. Given a camera ray  $\mathbf{r}$ , the rendered radiance is expressed as

$$\mathcal{T}(\mathbf{r}) = \sum_{i=0}^K \beta_i f_{\text{brdf}}(\mathbf{s}_{p_i}) L_i(\boldsymbol{\ell}, \mathbf{d}_i), \quad \mathbf{s}_{p_i} = \Phi\left(\mathbf{r}_i, \sum_{j=1}^{N_i} w_j^{(i)} \boldsymbol{\Theta}_j\right), \quad (12)$$

where  $\beta_i$  is the path throughput up to the  $i$ -th bounce,  $f_{\text{brdf}}(\mathbf{s}_{p_i})$  is the BRDF evaluated at the equivalent interaction state  $\mathbf{s}_{p_i}$ , and  $L_i(\boldsymbol{\ell}, \mathbf{d}_i)$  denotes the incident radiance arriving from direction  $\mathbf{d}_i$  under lighting  $\boldsymbol{\ell}$ . Each path-space interaction  $p_i$  is associated with a local shading state  $\mathbf{s}_{p_i}$  formed by aggregating the attributes of the Gaussians contributing to that interaction, where  $\Phi(\cdot)$  denotes the mapping from the current ray and aggregated Gaussian attributes to the local shading state.

Our goal is to jointly recover material attributes  $\boldsymbol{\Theta}$  and lighting  $\boldsymbol{\ell}$  by minimizing the discrepancy between rendered images and ground-truth images  $I_{\text{gt}}$ :

$$\min_{\boldsymbol{\Theta}, \boldsymbol{\ell}} \mathbb{E}_{\mathbf{r} \sim \pi(\mathbf{r})} [\mathcal{L}(\mathcal{T}(\mathbf{r}; \boldsymbol{\Theta}, \boldsymbol{\ell}), I_{\text{gt}}(\mathbf{r}))], \quad (13)$$

where  $\pi(\mathbf{r})$  denotes the camera-ray sampling distribution over image pixels, and  $p(\mathbf{r}) = (\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_K)$  denotes the multi-bounce light transport path induced by the camera ray  $\mathbf{r}$ .

**3.4.2 Gradient Propagation over Equivalent Interactions.** Direct differentiation of Eq. (13) with respect to the learnable parameters is impractical for Monte Carlo path tracing: it requires retaining a large intermediate computation graph across many sampled paths, while Gaussian aggregation further requires tracking the contributing primitives and their interaction states, leading to substantial memory overhead. Existing inverse rendering methods in 3D Gaussian fields [Gao et al. 2024] typically address this by caching intermediate screen-space states via splatting (e.g., G-buffers). In a fully ray-traced pipeline, however, explicitly caching per-path states becomes increasingly expensive as the sample count and the number of bounces increase.

We instantiate path replay backpropagation (PRB) [Vicini et al. 2021] over the same path-space equivalent interactions used by the forward estimator. During replay, equivalent interactions are reconstructed from replayed rays and random numbers, yielding a pathwise gradient conditioned on the replayed sampling decisions. The replayed contribution uses the same weighted estimator form as the forward pass, with PDFs, MIS weights, visibility indicators, and validity decisions evaluated on the replayed path and held fixed during differentiation. Gradients propagate only through smooth pathwise terms, including BRDF evaluation, equivalent interaction attributes, incident radiance, and ray-direction reparameterization. The resulting pathwise replay gradient can be expressed as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Theta}_j} = \sum_i \frac{\partial \mathcal{L}}{\partial \mathcal{T}_i} \left( \underbrace{\frac{\partial \mathcal{T}_i}{\partial f_{\text{brdf}}} \frac{\partial f_{\text{brdf}}}{\partial \mathbf{s}_{p_i}}}_{\text{BRDF gradient}} + \underbrace{\frac{\partial \mathcal{T}_i}{\partial L_i} \frac{\partial L_i}{\partial \mathbf{d}_i} \frac{\partial \mathbf{d}_i}{\partial \mathbf{s}_{p_i}}}_{\text{Ray-induced gradient}} \right) \frac{\partial \mathbf{s}_{p_i}}{\partial \boldsymbol{\Theta}_j}. \quad (14)$$

Since the local BRDF value enters the path throughput multiplicatively at the current interaction, its derivative is accumulated through the replayed throughput along the sampled path. For the BRDF parameterization in this work, albedo  $a$  only affects the local BRDF value but not the sampled direction and therefore receives

gradients only through the BRDF branch, while roughness  $r$  affects both the local BRDF value and the reparameterized outgoing direction, and therefore additionally receives gradients through the ray-induced branch.

Since each equivalent interaction is explicitly aggregated from contributing 3D Gaussian primitives, the contribution weight of each participating primitive is known. For local material attributes, this allows gradients of the equivalent interaction state to be directly redistributed to the contributing Gaussians:

$$\frac{\partial \mathbf{s}_{p_i}}{\partial \mathbf{g}_j} = w_j^{(i)} \mathbf{I}, \quad \bar{\mathbf{g}}_j += w_j^{(i)} \bar{\mathbf{s}}_{p_i}, \quad (15)$$

where  $\mathbf{g}_j$  denotes the local material-attribute vector of the  $j$ -th Gaussian primitive, and  $w_j^{(i)}$  is its contribution weight at the  $i$ -th replayed interaction. This weight-sharing between primal aggregation and gradient redistribution is what keeps the backward pass aligned with the forward ray-traced estimator.

The gradient of the environment light parameters can be expressed as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\ell}} = \sum_i \frac{\partial \mathcal{L}}{\partial \mathcal{T}_i} \frac{\partial \mathcal{T}_i}{\partial L_i} \frac{\partial L_i}{\partial \boldsymbol{\ell}} = \sum_i \frac{\partial \mathcal{L}}{\partial \mathcal{T}_i} \beta_i f_{\text{brdf}} \frac{\partial L_i}{\partial \boldsymbol{\ell}}, \quad (16)$$

where the gradient flows from the rendering loss to the environment parameters through the path throughput and the incident radiance evaluated along replayed paths.

**3.4.3 Training Objective.** We optimize the material attributes  $\boldsymbol{\Theta}$  and the light parameters  $\boldsymbol{\ell}$  by minimizing an objective composed of an appearance term  $\mathcal{L}_{\text{view}}$  and regularization terms  $\mathcal{L}_{\text{reg}}$ . Following Kerbl et al. [2023], the appearance term combines an  $\ell_1$  term with a D-SSIM term:

$$\mathcal{L}_{\text{view}} = (1 - \lambda_s) \mathcal{L}_1 + \lambda_s \mathcal{L}_{\text{D-SSIM}}. \quad (17)$$

For regularization  $\mathcal{L}_{\text{reg}}$ , we apply an edge-aware total variation penalty to the predicted material attribute maps. Let  $\mathcal{I}$  denote the set of regularized attribute maps. The smoothness term is defined as

$$\mathcal{L}_{\text{tv}} = \sum_{I \in \mathcal{I}} \sum_{\mathbf{p}} \|\nabla I(\mathbf{p})\|_2^2 \exp(-\|\nabla I_{\text{gt}}(\mathbf{p})\|), \quad (18)$$

where  $I_{\text{gt}}$  is the reference RGB image used to compute the edge-aware weights.

Jointly estimating material properties and environment light can be ambiguous. Following Du et al. [2024], we introduce an early-stage material prior to alleviate this coupling without requiring per-scene parameter tuning. Specifically, we employ a monocular diffusion-based estimator [Zeng et al. 2024] to provide material priors and regularize the predicted material maps during the early optimization stage:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{tv}} \mathcal{L}_{\text{tv}} + \lambda_{\text{pri}} \sum_{I \in \mathcal{I}} \mathcal{L}_2(I, I_{\text{pri}}), \quad (19)$$

where  $I_{\text{pri}}$  denotes the corresponding values predicted by the monocular prior. The prior weight  $\lambda_{\text{pri}}$  is initialized to 0.5, linearly decayed to 0.005 over the first 400 iterations, and kept fixed thereafter. The prior is assigned a relatively small weight and mainly provides weak guidance for material initialization during early optimization, with the RGB reconstruction loss remaining the primary supervision.

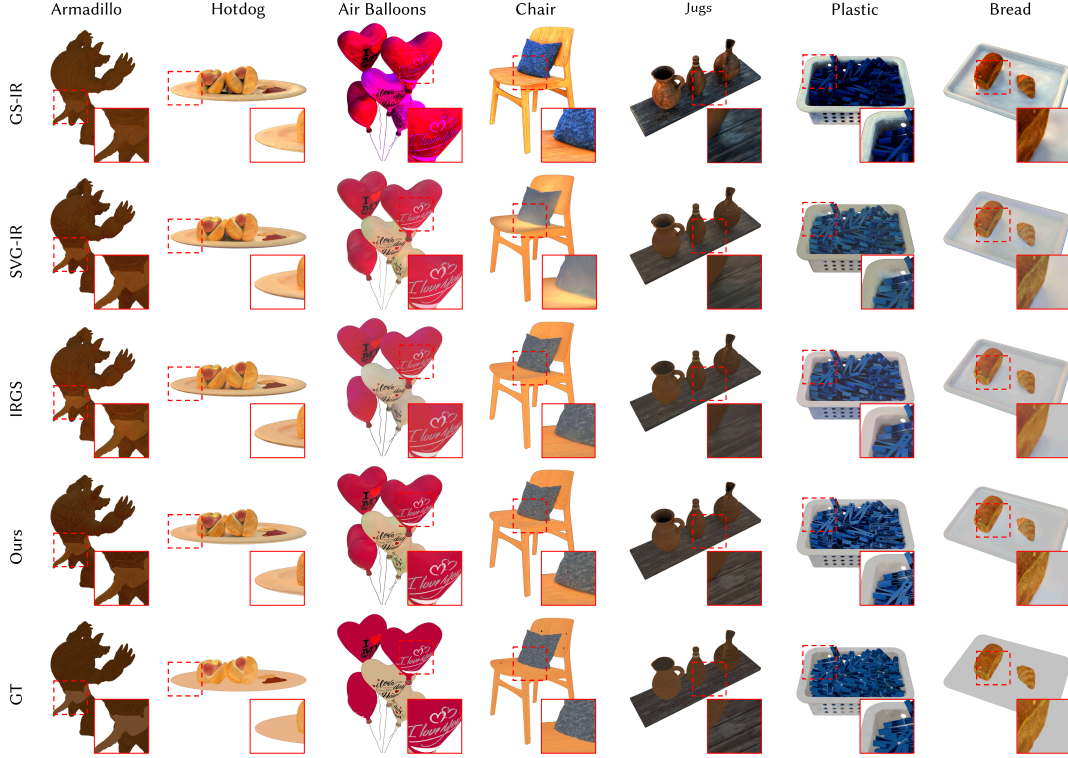


Fig. 5. Albedo recovery results on benchmark datasets, demonstrating that our framework achieves superior performance on albedo recovery.

## 4 EXPERIMENTS AND ABLATIONS

### 4.1 Implementation Details

We implement our method in an NVIDIA OptiX-based ray-tracing framework [Parker et al. 2010] and run all experiments on an NVIDIA RTX 3090 GPU. We adopt a two-stage optimization pipeline. In the first stage, Gaussian normals are supervised by depth-derived pseudo-normals using an angular normal-consistency loss, together with a normal prior from the same diffusion-based estimator described in Sec. 3.4.3. Gaussian densification via splitting and cloning is enabled in this stage. In the inverse-rendering stage, we fix the geometry and optimize the model for 800 iterations. Optimization is performed with 32 samples per pixel (SPP) and a maximum of 2 bounces, while final rendering uses 256 SPP. The D-SSIM mixing weight is set to  $\lambda_s = 0.2$ , and the TV regularization weight is set to  $\lambda_{tv} = 0.095$ . For the spherical Gaussian representation, we set  $N_{sg} = 24$  and  $\gamma = 0.3$ , and use a learning rate of 0.035 for both material and light parameters. A single filtering pass with a fixed  $3 \times 3$  normalized binomial kernel is applied to the rendered image during optimization to suppress Monte Carlo noise.

### 4.2 Results

We compare our method with state-of-the-art splatting-based inverse-rendering approaches [Gao et al. 2024; Gu et al. 2025; Liang et al. 2024; Sun et al. 2025a]. IRGS [Gu et al. 2025] and SVG-IR [Sun et al. 2025a] incorporate indirect illumination through screen-space approximations. To align with these baselines, we fix the metallic

Table 1. Quantitative evaluation on TensorIR dataset.

| Method                     | Albedo<br>PSNR↑ | NVS<br>PSNR↑ | Relight<br>PSNR↑ | Roughness<br>MSE↓ | Normal<br>MAE↓ | Training<br>Hours↓ |
|----------------------------|-----------------|--------------|------------------|-------------------|----------------|--------------------|
| TensorIR [Jin et al. 2023] | 29.27           | 35.09        | 28.58            | 0.013             | 4.100          | 4.0                |
| GS-IR [Liang et al. 2024]  | 29.94           | 35.33        | 24.37            | 0.027             | 4.948          | 0.5                |
| R3DG [Gao et al. 2024]     | 29.27           | 33.35        | 27.37            | 0.016             | 5.927          | 1.1                |
| SVG-IR [Sun et al. 2025a]  | 30.48           | 36.71        | 31.10            | 0.033             | 4.358          | 1.1                |
| IRGS [Gu et al. 2025]      | 30.62           | 35.43        | 29.91            | 0.054             | 4.209          | 0.9                |
| Ours w/o Prior             | 31.04           | 36.50        | 31.31            | 0.023             | 4.028          | 0.4                |
| Ours                       | 32.12           | 36.17        | 31.84            | 0.015             |                |                    |

Table 2. Quantitative evaluation on Synthetic4Relight dataset.

| Method                    | Albedo<br>PSNR↑ | NVS<br>PSNR↑ | Relight<br>PSNR↑ | Roughness<br>MSE↓ | Training<br>Hours↓ |
|---------------------------|-----------------|--------------|------------------|-------------------|--------------------|
| GS-IR [Liang et al. 2024] | 19.48           | 33.95        | 25.40            | 0.011             | 0.4                |
| R3DG [Gao et al. 2024]    | 28.65           | 34.10        | 33.12            | 0.010             | 0.9                |
| SVG-IR [Sun et al. 2025a] | 29.06           | 36.10        | 32.59            | 0.009             | 1.0                |
| IRGS [Gu et al. 2025]     | 30.50           | 34.42        | 34.35            | 0.008             | 0.7                |
| Ours w/o Prior            | 30.80           | 36.84        | 33.82            | 0.008             | 0.6                |
| Ours                      | 31.09           | 36.66        | 34.59            | 0.008             |                    |

parameter to zero and use a  $128 \times 64$  2D environment map for benchmark evaluations. Given sufficient optimization, this higher-capacity representation offers flexibility for fitting complex illumination and can achieve stronger benchmark performance. Separately, for our

Table 3. Quantitative evaluation on RT4Relight dataset.

| Method                    | Albedo<br>PSNR↑ | NVS<br>PSNR↑ | Relight<br>PSNR↑ | Roughness<br>MSE↓ | Training<br>Hours↓ |
|---------------------------|-----------------|--------------|------------------|-------------------|--------------------|
| GS-IR [Liang et al. 2024] | 23.89           | 33.87        | 24.91            | 0.083             | 0.5                |
| R3DG [Gao et al. 2024]    | 25.80           | 34.61        | 24.60            | 0.025             | 0.9                |
| SVG-IR [Sun et al. 2025a] | 26.71           | 36.63        | 27.77            | 0.022             | 1.4                |
| IRGS [Gu et al. 2025]     | 29.28           | 36.35        | 29.56            | 0.065             | 0.7                |
| Ours w/o Prior            | 29.34           | 36.49        | 29.81            | 0.018             | 0.6                |
| Ours                      | 29.57           | 35.99        | 30.76            | 0.016             |                    |

standalone real-world experiments, we adopt the compact spherical Gaussian illumination representation, which enables faster convergence and smoother optimization gradients.

**Results on Benchmark Datasets.** We evaluate on TensoIR [Jin et al. 2023], Synthetic4Relight [Zhang et al. 2022], and RT4Relight. These synthetic datasets provide ground-truth materials and relighting images, enabling quantitative evaluation of material decomposition and relighting quality. Quantitative results in Tabs. 1, 2, and 3 show that our method achieves competitive and stable performance across most metrics while maintaining competitive training efficiency. Training time is averaged across all scenes. Qualitative albedo and roughness results are shown in Figs. 5 and 6.

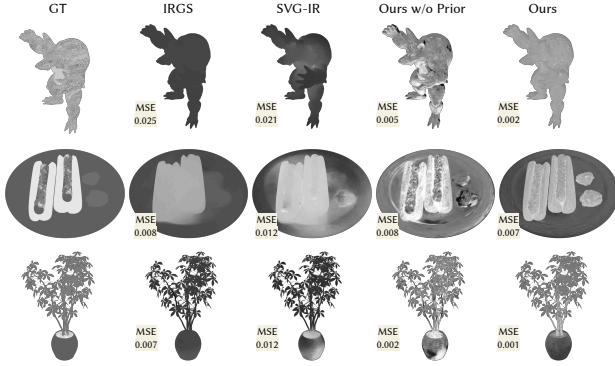


Fig. 6. Roughness recovery results. Our recovered roughness is closer to the ground truth, while the diffusion prior helps regularize early optimization and suppress degenerate low-roughness artifacts.

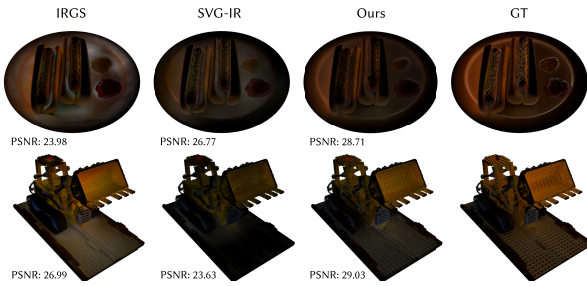


Fig. 7. Indirect-only rendering results from inverse-rendered scenes. Our result is obtained through path tracing under the full rendering equation, yielding indirect-light estimates that are closer to the ground truth.

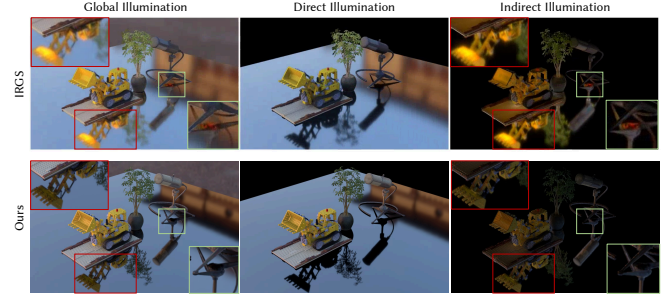


Fig. 8. Multi-bounce light transport result. Compared with IRGS [Gu et al. 2025], our path-traced framework explicitly traces repeated light bounces under the full rendering equation, enabling more consistent direct illumination, reflections, and indirect radiance.

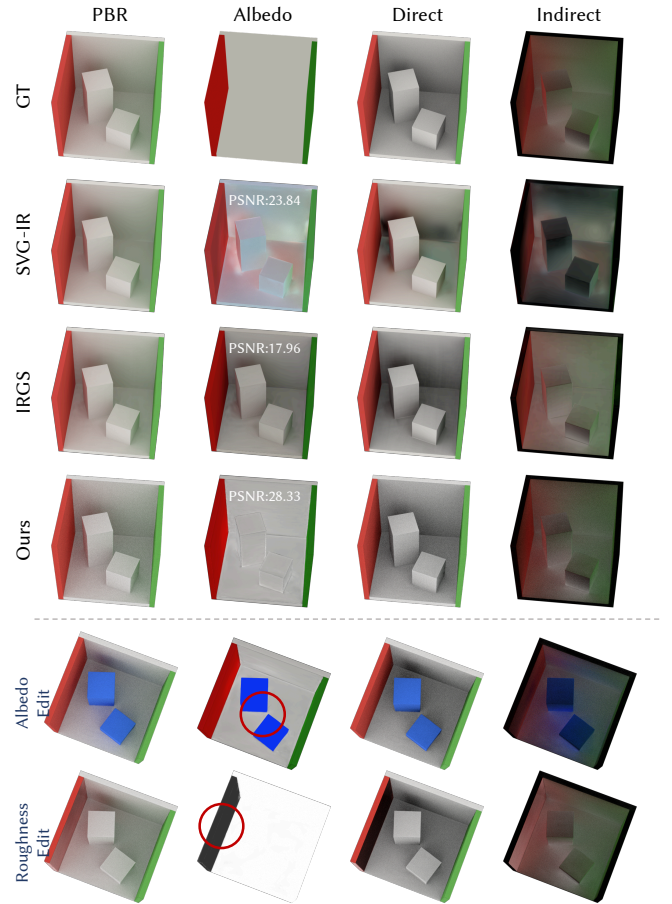


Fig. 9. Cornell box inverse rendering and material editing with a maximum of 4 bounces. Material edits in our framework, including albedo and roughness changes, affect not only the directly visible appearance but also global illumination effects such as indirect radiance and color bleeding.





Fig. 10. Relighting comparison on benchmark datasets. Our framework achieves more natural relighting with soft shadows and realistic reflections.



Fig. 11. Relighting results from our method on Stanford-ORB [Kuang et al. 2023], DTU [Aanæs et al. 2016], and Mip-NeRF 360 [Barron et al. 2022] scenes. We demonstrate relighting with environment maps and near-field light sources, producing consistent illumination and soft shadows under the target illumination.



**Results on Global Illumination.** We compare our method with IRGS [Gu et al. 2025] and SVG-IR [Sun et al. 2025a] in terms of indirect illumination (Fig. 7). When the recovered objects are rendered in a forward path-tracing pipeline, our method produces more plausible indirect illumination (Fig. 8). IRGS models indirect illumination using spherical-harmonic values attached to Gaussians, which can lead to inaccurate indirect-light intensity when evaluated under multi-bounce global illumination. Experiments on the Cornell-box scene (Fig. 9) suggest that multi-bounce global illumination helps disentangle surface albedo from color bleeding. Moreover, due to the consistency between forward rendering and backward optimization, material edits propagate to the resulting global illumination.

**Relighting Results on Synthetic and Real Datasets.** We first compare relighting on synthetic benchmarks (Fig. 10). For real objects without ground-truth relighting, we show single-object results on Stanford-ORB and DTU, and scene-level results on Mip-NeRF 360 scenes (Fig. 11) under environment maps and near-field lights. Our path-traced framework naturally accounts for visibility under the target illumination. Relighting is evaluated through Monte Carlo path tracing without relying on screen-space visibility or cached transport. This enables realistic soft shadows and illumination changes, especially under near-field lights, supporting physically consistent relighting and appearance editing.

### 4.3 Ablation Study

We conduct ablation studies on the TensorIR dataset without prior regularization to validate the effectiveness of key components (Tab. 4) and analyze the effects of different implementation settings (Tab. 5).

**Global Illumination.** Modeling global illumination with the full rendering equation via multi-bounce transport improves material recovery and relighting quality over direct illumination alone, highlighting the importance of indirect light transport for inverse rendering (Fig. 12(a) and the 1-bounce setting in Tab. 5).

Table 4. Ablation study of key components on TensorIR dataset (w/o Prior), demonstrating the effectiveness of each component.

| Method                                          | Albedo PSNR↑ | NVS PSNR↑    | Relight PSNR↑ |
|-------------------------------------------------|--------------|--------------|---------------|
| w/o Geom. Criterion                             | 30.96        | 36.19        | 31.13         |
| w/o Backface-aware Offset, $\epsilon_o = 0.025$ | 30.89        | 36.16        | 31.13         |
| w/o Backface-aware Offset, $\epsilon_o = 0.05$  | 30.94        | 36.38        | 31.24         |
| Discrete Interaction                            | 30.71        | 36.15        | 30.65         |
| Full                                            | <b>31.04</b> | <b>36.50</b> | <b>31.31</b>  |

**Geometric Validity Criterion.** The geometric-validity criterion for ray continuation in Eq. (9) introduces a stricter condition for secondary interactions during multi-bounce transport, improving their reliability and avoiding light-leaking artifacts (Fig. 12(b)).

**Self-Intersection Rejection.** We evaluate the backface-aware origin-offset rule in Eq. (10). It suppresses spurious self-occlusion by rejecting local back-facing responses from overlapping Gaussians that represent the same aggregated surface. Compared with a fixed ray-origin offset, our rule reduces self-occlusion artifacts without relying on a manually selected offset magnitude (Fig. 12(c)).

**Prior Regularization.** The diffusion prior provides a coarse perceptual anchor during the early stage of path-traced inverse rendering,

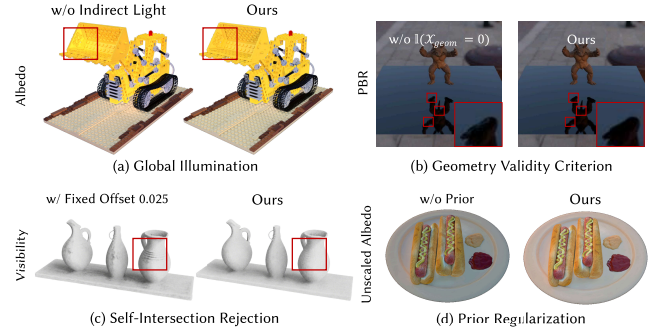


Fig. 12. Ablation studies on various components of our framework.

guiding the material parameters toward a stable and spatially consistent solution. In particular, it yields more plausible albedo intensity before scale alignment and reduces the sensitivity of material optimization to scene-specific hyperparameter tuning (Figs. 12(d) and 6).

**Discrete Interaction Modeling.** Discrete modeling can also be optimized through differentiable path tracing, but in our implementation, it exhibits greater sensitivity to geometric consistency, as each interaction depends on a single sampled Gaussian rather than an aggregated surface state (Fig. 13). A possible explanation is the mismatch between the aggregate interaction model used for geometry reconstruction and the discrete interaction model adopted during inverse rendering. Xu et al. [2026] introduce a sorting-free stochastic reconstruction estimator that may produce geometry better aligned with the discrete interaction model used during inversion.

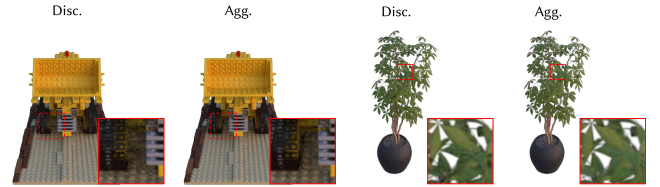


Fig. 13. Ablation study of ray-Gaussian interaction models during material optimization with 32 SPP. Aggregate interaction modeling typically yields more reliable results in our implementation.

**SPP and Bounces.** We ablate the number of samples per pixel (SPP) and the maximum number of bounces during inversion in Tab. 5. Increasing either setting generally improves inversion quality while increasing optimization time, with higher SPP additionally raising peak GPU memory usage. The slightly higher Albedo PSNR with the 1-bounce setting is mainly due to the Armadillo scene (+0.67 dB), where the missing indirect illumination can be partially absorbed into the recovered albedo scale and subsequently favored by scale alignment during evaluation. The maximum number of bounces has little effect on peak GPU memory because path traversal is performed iteratively, updating a fixed-size per-path payload in place rather than allocating depth-indexed buffers. For comparison, the average peak GPU memory usage is 12.81 GB for IRGS [Gu et al. 2025] and 22.65 GB for SVG-IR [Sun et al. 2025a] under their

default settings. Our default implementation settings (32 SPP and 2 Bounces) provide a favorable quality-time trade-off.

Table 5. Ablation of SPP and bounce count on the TensorIR dataset (w/o Prior). Each factor is varied independently from the default optimization setting (32 SPP and a maximum of 2 bounces), with the other fixed. The default setting provides a favorable quality-time trade-off.

| Setting   | Albedo<br>PSNR↑ | NVS<br>PSNR↑ | Relight<br>PSNR↑ | Avg. Peak GPU mem.<br>GB↓ | Training<br>Hours↓ |
|-----------|-----------------|--------------|------------------|---------------------------|--------------------|
| 1 SPP     | 28.12           | 34.89        | 30.00            | <b>1.51</b>               | <b>0.04</b>        |
| 16 SPP    | 30.53           | 36.45        | 31.30            | 3.91                      | 0.21               |
| 64 SPP    | <b>31.45</b>    | <b>36.63</b> | <b>31.75</b>     | 8.83                      | 0.77               |
| 1 Bounce  | <b>31.18</b>    | 34.91        | 31.17            | <b>5.67</b>               | <b>0.32</b>        |
| 3 Bounces | 31.06           | 36.52        | 31.32            | 5.70                      | 0.42               |
| 4 Bounces | 31.06           | <b>36.53</b> | 31.36            | 5.71                      | 0.44               |
| 9 Bounces | 31.07           | <b>36.53</b> | <b>31.44</b>     | 5.71                      | 0.47               |
| Default   | 31.04           | 36.50        | 31.31            | 5.70                      | 0.39               |

## 5 DISCUSSION AND LIMITATIONS

*Computational Overhead.* Our goal is not real-time novel-view synthesis, but physically consistent inverse rendering of Gaussian materials for path-traced rendering, relighting, and appearance editing. This requires optimizing materials under the same ray-traced transport used for final rendering. Path replay backpropagation avoids storing intermediate path states, but adds a replay pass that repeats Gaussian-ray intersections, shadow tests, and MIS evaluations. With 32 SPP for both primal and replay passes, the first-interaction cost is about 32× higher than 3DGRT [Moenne-Loccoz et al. 2024]. Runtime, training time, and peak GPU memory are reported in Tab. 6. For comparison, under a 256-SPP setting on the TensorIR dataset, our method takes 3.85 s per frame on average, compared with 1.24 s for IRGS [Gu et al. 2025], requiring approximately 3.1× the rendering time. Reducing this cost remains an important future direction; adaptive sampling, transport caching, and neural radiosity-style approximations may help amortize multi-bounce transport.

Table 6. Runtime and memory statistics across different Gaussian-count scales under 32 SPP and a maximum of 2 bounces. Training time is estimated for 800 optimization iterations, where the 32 samples per pixel are packed and processed as one ray chunk.

| Gaussian-count scale     | Res. | Inversion    | Peak GPU mem. | Train time | Post-opt. render |
|--------------------------|------|--------------|---------------|------------|------------------|
| 158K (TensorIR)          | 800  | 1.66 s/iter  | 5.70 GB       | 0.39 h     | 0.59 s/frame     |
| 179K (Synthetic4Relight) | 800  | 2.63 s/iter  | 5.66 GB       | 0.60 h     | 0.95 s/frame     |
| 106K (Stanford-ORB)      | 1200 | 8.64 s/iter  | 9.29 GB       | 2.01 h     | 3.07 s/frame     |
| 2.5M (Mip-NeRF 360)      | 1237 | 24.92 s/iter | 13.68 GB      | 5.65 h     | 9.04 s/frame     |

*Opaque Surface Scenes.* Our formulation targets opaque, surface-dominated scenes. Instead of treating each Gaussian as an independent volumetric medium, we aggregate overlapping Gaussian responses into an equivalent surface interaction and perform PBR-based recursive multi-bounce light transport over this representation. Accordingly, our light-transport formulation and self-intersection

rejection mechanism assume surface-like interactions and are not intended for genuinely volumetric or transmissive phenomena, such as smoke, hair, fur, or refractive materials.

*Discontinuities and Non-differentiability Considerations.* Our path-traced inversion involves discrete decisions in geometry-validity evaluation and shadow-ray visibility testing. The hard geometry-validity threshold is applied after the Gaussian geometry and opacity have been reconstructed, and both remain fixed throughout the subsequent inversion. Following prior Gaussian inverse-rendering methods [Gu et al. 2025; Sun et al. 2025a], the reconstruction stage employs a mask entropy loss that encourages foreground opacity toward one, thereby reducing ambiguous responses near the threshold. Since the Gaussian geometry remains fixed during inversion, the validity decision for a given ray is independent of the optimized material and illumination parameters. We observed no noticeable threshold-induced artifacts in our experiments, although regions with uncertain geometry may be more sensitive to this transition. Shadow-ray visibility decisions likewise depend only on the fixed scene geometry. Path replay reproduces the forward visibility decisions and propagates gradients through the subsequent continuous shading evaluation. Accordingly, these discrete decisions do not obstruct gradient propagation with respect to the material and illumination parameters optimized in our setting.

*Alternative Interaction Models.* Our framework is compatible with different formulations of ray-Gaussian interactions. As described in Sec. 3.2, the aggregate model constructs an equivalent local shading state, whereas the discrete model replaces this aggregation step with Monte Carlo sampling of a single Gaussian primitive at each interaction. The two formulations share the same path-tracing and path-replay pipeline. During backward optimization, path replay reproduces the stochastic selection made in the forward pass and propagates gradients through the corresponding interaction. Therefore, the sampling decision need not be differentiated through and does not prevent gradient propagation through the subsequent shading evaluation. Due to nonlinear shading evaluation during path tracing, using a consistent interaction formulation for optimization and rendering is more appropriate. This modular formulation can also accommodate alternative Gaussian representations such as 2DGS [Huang et al. 2024]. Such representations can be incorporated by replacing the current ray-Gaussian evaluation and local interaction construction with their 2D counterparts while retaining the subsequent light-transport and optimization pipeline. Surface-aligned primitives may improve interaction consistency around thin structures and geometric boundaries, although gaps, overlaps, and grazing-angle intersections between planar primitives require careful handling during recursive light transport.

*Non-Smooth Gradients.* In our implementation, ray-traced Gaussian optimization may yield less smooth material gradients than splatting-based optimization, as neighboring rays can interact with different primitives and follow different transport paths under limited sampling. Splatting-based methods, by contrast, introduce implicit spatial smoothing through screen-space accumulation. This effect can be more pronounced in the absence of explicit smoothing regularization. Structure- or cluster-aware regularization over spatial or semantic neighborhoods could further mitigate this artifact.

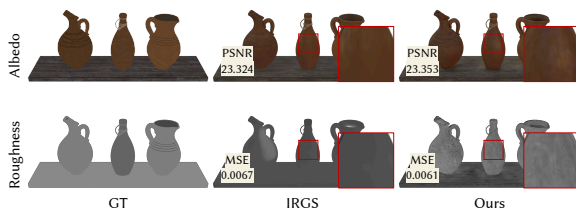


Fig. 14. Non-smooth material artifact. Although the quantitative metrics are comparable, ray-traced optimization may yield less smooth material maps than splatting-based methods under limited sampling.

## 6 CONCLUSION

We have explored inverse rendering in ray-traced 3D Gaussian scenes and introduced a splatting-free framework that defines forward rendering and backward optimization within the same ray-based light transport pipeline. This consistency allows material estimation and illumination optimization to be performed under a unified physical process, rather than relying on rasterization-derived buffers or screen-space approximations. Our results demonstrate the feasibility of using 3D Gaussian fields as optimizable scene representations for ray-traced inverse rendering, showing that Gaussian primitives can be integrated into physically based light transport beyond view-synthesis-oriented splatting.

By accounting for multi-bounce transport, our framework optimizes material and illumination parameters with the same path-traced process used for relighting, appearance editing, and physically based evaluation. This enables 3D Gaussian assets to be used in ray-traced pipelines, where visibility, shadows, reflections, and indirect illumination are evaluated within a unified path-space transport process. It supports scenes where Gaussians coexist with meshes, environment illumination, and analytic light sources. We hope this work encourages further exploration of efficient and robust ray-traced Gaussian representations, including adaptive sampling, transport caching, neural approximations, structure- or cluster-aware regularization to encourage coherent optimization within Gaussian groups, and extensions to a broader range of Gaussian representations.

## Acknowledgments

We thank the anonymous reviewers for valuable comments. We also thank NVIDIA for their contributions to ray tracing and EPFL for their work on path replay backpropagation. This work was supported in part by the National Key Research and Development Program of China (No. 2023YFF0905103).

## References

Henrik Aanaes, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjarholm Dahl. 2016. Large-Scale Data for Multiple-View Stereopsis. *International Journal of Computer Vision* (2016), 153–168.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *CVPR* (2022).

Zoubin Bi, Yixin Zeng, Chong Zeng, Fan Pei, Xiang Feng, Kun Zhou, and Hongzhi Wu. 2024. GS<sup>3</sup>: Efficient Relighting with Triple Gaussian Splatting. In *SIGGRAPH Asia 2024 Conference Papers*.

Brent Burley. 2012. Physically-Based Shading at Disney. In *ACM SIGGRAPH Course Notes*. 1–7.

Krzysztof Byrski, Marcin Mazur, Jacek Tabor, Tadeusz Dziarmaga, Marcin Kądziołka, Dawid Baran, and Przemysław Spurek. 2025. RaySplats: Ray Tracing based Gaussian Splatting. arXiv:2501.19196 [cs.CV] <https://arxiv.org/abs/2501.19196>

Jorge Condo, Sebastian Speierer, Lukas Bode, Aljaz Bozic, Simon Green, Piotr Didyk, and Adrian Jarabo. 2025. Don't Splat your Gaussians: Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media. *ACM Trans. Graph.* (Jan. 2025). doi:10.1145/3711853

Paul Debevec. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 189–198. doi:10.1145/280814.280864

Kang Du, Zhihao Liang, and Zeyu Wang. 2024. GS-ID: Illumination Decomposition on Gaussian Splatting via Diffusion Prior and Parametric Light Source Optimization. arXiv:2408.08524 [cs.CV] <https://arxiv.org/abs/2408.08524>

Jiahui Fan, Fujun Luan, Jian Yang, Milos Hasan, and Beibei Wang. 2025. RNG: Relightable Neural Gaussians. *Proceedings of CVPR 2025* (2025).

Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. 2024. Relightable 3D Gaussians: Realistic Point Cloud Relighting with BRDF Decomposition and Ray Tracing. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLV* (Milan, Italy). Springer-Verlag, Berlin, Heidelberg, 73–89. doi:10.1007/978-3-031-72995-9\_5

Álvaro González. 2009. Measurement of Areas on a Sphere Using Fibonacci and Latitude–Longitude Lattices. *Mathematical Geosciences* 42, 1 (Nov. 2009), 49–64. doi:10.1007/s11004-009-9257-x

Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. 2025. IRGS: Inter-Reflective Gaussian Splatting with 2D Gaussian Ray Tracing. In *CVPR*.

Yijia Guo, Yuanxi Bai, Liwen Hu, Ziyi Guo, Mianzhi Liu, Yu Cai, Tiejun Huang, and Lei Ma. 2024. PRTGS: Precomputed Radiance Transfer of Gaussian Splats for Real-Time High-Quality Relighting (MM '24). Association for Computing Machinery, New York, NY, USA, 5112–5120. doi:10.1145/3664647.3680893

Florian Hahlebohm, Fabian Friederichs, Tim Weyrich, Linus Franke, Moritz Kappel, Susana Castillo, Marc Stamminger, Martin Eisemann, and Marcus Magnor. 2025. Efficient Perspective-Correct 3D Gaussian Splatting Using Hybrid Transparency. *Computer Graphics Forum* 44, 2 (2025). doi:10.1111/cgf.70014

Chenxiao Hu, Meng Gai, Guoping Wang, and Sheng Li. 2026. Dynamic Global Illumination for Interactive Gaussian Splatting Scenes in Real Time. *IEEE Transactions on Visualization and Computer Graphics* 32, 7 (2026), 6919–6931. doi:10.1109/TVCG.2026.3689199

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery. doi:10.1145/3641519.3657428

Kaiwen Jiang, Jia-Mu Sun, Zilu Li, Dan Wang, Tzu-Mao Li, and Ravi Ramamoorthi. 2025. Differentiable Light Transport with Gaussian Surfels via Adapted Radiosity for Efficient Relighting and Geometry Reconstruction. *ACM Transactions on Graphics (TOG)* 44, 6 (December 2025).

Yingwenqi Jiang, Jia Dong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. 2024. GaussianShader: 3D gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5322–5332.

Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. 2023. TensorIR: Tensorial Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

James T. Kajiya. 1986. The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150. doi:10.1145/15886.15902

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023).

Zhengfei Kuang, Yunzhi Zhang, Hong-Xing Yu, Samir Agarwala, Shangzhe Wu, Jiajun Wu, et al. 2023. Stanford-ORB: a real-world 3D object inverse rendering benchmark. *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. 2024. GS-IR: 3D Gaussian Splatting for Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21644–21653.

Jingwang Ling, Ruihan Yu, Feng Xu, Chun Du, and Shuang Zhao. 2024. NeRF as a Non-Distant Environment Emitter in Physics-based Inverse Rendering. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 39, 12 pages. doi:10.1145/3641519.3657404

Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 2024. 3D Gaussian Ray Tracing: Fast Tracing of Particle Scenes. *ACM Transactions on Graphics* 43, 6, Article 232 (Nov. 2024), 19 pages. doi:10.1145/3687934



- Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. 2003. All-Frequency Shadows Using Non-Linear Wavelet Lighting Approximation. *ACM Transactions on Graphics* 22, 3 (July 2003), 376–381. doi:10.1145/882262.882280
- Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. 2010. OptiX: a general purpose ray tracing engine. *ACM Trans. Graph.* 29, 4, Article 66 (July 2010), 13 pages. doi:10.1145/1778765.1778803
- Yohan Poirier-Ginter, Jeffrey Hu, Jean-François Lalonde, and George Drettakis. 2025. Editable Physically-based Reflections in Raytraced Gaussian Radiance Fields. In *SIGGRAPH Asia 2025 - 18th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia*. Hong Kong, Hong Kong SAR China. doi:10.1145/3757377.3763971
- Ravi Ramamoorthi and Pat Hanrahan. 2001. An efficient representation for irradiance environment maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 497–500. doi:10.1145/383259.383317
- Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. 2025a. SVG-IR: Spatially-Varying Gaussian Splatting for Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16143–16152.
- Xin Sun, Iliyan Georgiev, Yun Fei, and Miloš Hašan. 2025b. Stochastic Ray Tracing of 3D Transparent Gaussians. *EGSR conference proceedings* (2025).
- Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph. D. Dissertation. Stanford University, Stanford, CA, USA. doi:10.5555/927297
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 108:1–108:14.
- Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. 2009. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph.* 28, 5 (Dec. 2009), 1–10. doi:10.1145/1618452.1618479
- Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 2025. 3DGUT: Enabling Distorted Cameras and Secondary Rays in Gaussian Splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2025).
- Tong Wu, Jia-Mu Sun, Yu-Kun Lai, Yuewen Ma, Leif Kobbelt, and Lin Gao. 2024. DeferredGS: Decoupled and Editable Gaussian Splatting with Deferred Shading. *arXiv preprint arXiv:2404.09412* (2024).
- Tao Xie, Xi Chen, Zhen Xu, Yiman Xie, Yudong Jin, Yujun Shen, Sida Peng, Hujun Bao, and Xiaowei Zhou. 2024. EnvGS: Modeling View-Dependent Appearance with Environment Gaussian. *arXiv preprint arXiv:2412.15215* (2024).
- Peiyu Xu, Xin Sun, Krishna Mullia, Raymond Fei, Iliyan Georgiev, and Shuang Zhao. 2026. Stochastic Ray Tracing for the Reconstruction of 3D Gaussian Splatting. *arXiv:2603.23637 [cs.CV]* <https://arxiv.org/abs/2603.23637>
- Xijie Yang, Mulin Yu, Changjian Jiang, Kerui Ren, Tao Lu, Jiangmiao Pang, Dahua Lin, Bo Dai, and Linning Xu. 2026. EAG-PT: Emission-Aware Gaussians and Path Tracing for Diffuse Indoor Scene Reconstruction and Editing. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26)*. Association for Computing Machinery, New York, NY, USA, Article 149, 12 pages. doi:10.1145/3799902.3811054
- Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2022. NeILF: Neural Incident Light Field for Material and Lighting Estimation. In *ECCV*. <https://arxiv.org/abs/2203.07182>
- Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. 2024. RGB $\leftrightarrow$ X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 75, 11 pages. doi:10.1145/3641519.3657445
- Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2023. NeILF++: Inter-reflectable Light Fields for Geometry and Material Estimation. *International Conference on Computer Vision (ICCV)* (2023).
- Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021. PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. 2022. Modeling Indirect Illumination for Inverse Rendering. In *CVPR*.
- Yang Zhou, Songyin Wu, and Ling-Qi Yan. 2024. Unified Gaussian Primitives for Scene Representation and Rendering. *arXiv:2406.09733 [cs.GR]* <https://arxiv.org/abs/2406.09733>

## Appendix

### A Physically Based Rendering Details

Our framework is built on physically based path tracing. For a non-emissive surface point  $\mathbf{x} \in \mathbb{R}^3$ , the outgoing radiance  $L_o \in \mathbb{R}^3$  along direction  $\omega_o$  is governed by the classical rendering equation [Kajiya 1986]:

$$L_o(\mathbf{x}, \omega_o) = \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) (\mathbf{n} \cdot \omega_i) d\omega_i, \quad (1)$$

where  $\omega_i \in \Omega$  denotes the incident direction,  $f_r(\cdot)$  is the bidirectional reflectance distribution function (BRDF),  $L_i \in \mathbb{R}^3$  is the incident radiance,  $\mathbf{n} \in \mathbb{R}^3$  is the surface normal, and  $\Omega$  denotes the upper hemisphere aligned with  $\mathbf{n}$ . The incident radiance  $L_i$  recursively encodes light transport from the entire scene, including direct illumination, visibility, and multi-bounce scattering.

We adopt a simplified Disney BRDF [Burley 2012] parameterized by albedo  $\mathbf{a} \in \mathbb{R}^3$ , roughness  $r \in \mathbb{R}$ , and metallic  $m \in \mathbb{R}$ . The metallic parameter controls the relative diffuse and specular contributions, and the BRDF is modeled as a combination of a diffuse term and a GGX microfacet specular term:

$$f_r(\omega_i, \omega_o) = (1 - m) \frac{\mathbf{a}}{\pi} (1 - F) + \frac{D(\mathbf{n}, \mathbf{h}) F(\omega_o, \mathbf{h}) G(\omega_i, \omega_o)}{4(\mathbf{n} \cdot \omega_i)(\mathbf{n} \cdot \omega_o)}, \quad (2)$$

$$\mathbf{h} = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|},$$

$$F_0 = (1 - m) \cdot 0.04 + m\mathbf{a},$$

$$F(\omega_o, \mathbf{h}) = F_0 + (1 - F_0) (1 - \max(\omega_o \cdot \mathbf{h}, 0))^5,$$

where  $\mathbf{h}$  is the half vector and  $F$  is the Fresnel term. The GGX normal distribution function and the Schlick-GGX approximation of the geometry term are defined as

$$D(\mathbf{n}, \mathbf{h}) = \frac{\alpha^2}{\pi [(\mathbf{n} \cdot \mathbf{h})^2 (\alpha^2 - 1) + 1]^2},$$

$$G(\omega_i, \omega_o) = G_1(\mathbf{n}, \omega_i) G_1(\mathbf{n}, \omega_o), \quad (3)$$

$$G_1(\mathbf{n}, \omega) = \frac{\mathbf{n} \cdot \omega}{(\mathbf{n} \cdot \omega)(1 - k) + k}, \quad k = \frac{\alpha}{2}, \quad \alpha = r^2.$$

In implementation, we evaluate the cosine-weighted BRDF term

$$\tilde{f}_r = f_r(\omega_i, \omega_o) \max(\mathbf{n} \cdot \omega_i, 0), \quad (4)$$

which directly corresponds to the BRDF-cosine factor in the integrand of Eq. (1).

### B Inverse Rendering Algorithm Details

In Sec. 3.4 of the main paper, the gradient of the material parameters is decomposed into a local BRDF branch and a ray-induced branch:

$$\frac{\partial \mathcal{L}}{\partial \Theta_j} = \sum_i \frac{\partial \mathcal{L}}{\partial \mathcal{T}_i} \left( \frac{\partial \mathcal{T}_i}{\partial f_{\text{brdf}}} \frac{\partial f_{\text{brdf}}}{\partial s_{p_i}} + \frac{\partial \mathcal{T}_i}{\partial L_i} \frac{\partial L_i}{\partial d_i} \frac{\partial d_i}{\partial s_{p_i}} \right) \frac{\partial s_{p_i}}{\partial \Theta_j}. \quad (5)$$

Under multiple importance sampling (MIS), each interaction evaluates one environment-sampled query and one BRDF-sampled continuation query. We write the contribution of all replayed paths passing through the  $i$ -th interaction as

$$\mathcal{T}_i = \beta_i \left( f_i^{\text{env}} L_i + f_i^{\text{brdf}} f_{i+1}^{\text{env}} L_{i+1} + f_i^{\text{brdf}} f_{i+1}^{\text{brdf}} f_{i+2}^{\text{env}} L_{i+2} + \dots \right), \quad (6)$$

where  $\beta_i$  denotes the path throughput before the  $i$ -th interaction and  $L_i$  denotes the incident radiance along the sampled environment direction. The factors  $f_i^{\text{env}}$  and  $f_i^{\text{brdf}}$  are the evaluated scattering terms for the environment-sampling and BRDF-continuation branches,

including the BRDF value, cosine term, PDF correction, and MIS weight. During replay, PDF corrections and MIS weights are treated as fixed estimator weights and are not differentiated. With fixed geometry and path replay [Vicini et al. 2021], this treatment does not introduce additional non-differentiable operations into the replayed optimization.

Under this convention, gradients through the scattering factors below are propagated only through the evaluated BRDF-cosine terms. From Eq. (6), the derivatives with respect to the two local scattering factors are

$$\frac{\partial \mathcal{T}_i}{\partial f_i^{\text{env}}} = \beta_i L_i, \quad \frac{\partial \mathcal{T}_i}{\partial f_i^{\text{brdf}}} = \frac{\mathcal{T}_i - \beta_i f_i^{\text{env}} L_i}{f_i^{\text{brdf}}}. \quad (7)$$

These derivatives provide the adjoints propagated to the equivalent surface interaction through path replay. We denote the local cosine-weighted BRDF term by  $f_{\text{brdf}}$  and derive its material derivatives with respect to albedo, roughness, and metallic, with the BRDF normal detached. Omitting the interaction index  $i$ , we define

$$\text{denom} = 4(\mathbf{n} \cdot \omega_o)(\mathbf{n} \cdot \omega_i) + \epsilon, \quad \kappa = \frac{DG}{\text{denom}}, \quad c = (1 - \omega_o \cdot \mathbf{h})^5,$$

where  $\epsilon$  is a small constant for numerical stability. The derivatives with respect to albedo, roughness, and metallic are

$$\frac{\partial f_{\text{brdf}}}{\partial a_k} = (\mathbf{n} \cdot \omega_i) \left[ \kappa m (1 - c) \mathbf{e}_k + \frac{1 - m}{\pi} \left( (1 - F_k) \mathbf{e}_k - \mathbf{a} \frac{\partial F}{\partial a_k} \right) \right],$$

$$\frac{\partial f_{\text{brdf}}}{\partial r} = (\mathbf{n} \cdot \omega_i) F \frac{G \frac{\partial D}{\partial r} + D \frac{\partial G}{\partial r}}{\text{denom}}, \quad (8)$$

$$\frac{\partial f_{\text{brdf}}}{\partial m} = (\mathbf{n} \cdot \omega_i) \left[ \kappa (1 - c) (\mathbf{a} - 0.041) - \frac{\mathbf{a}}{\pi} \left( 1 - F + (1 - m) \frac{\partial F}{\partial m} \right) \right],$$

where  $a_k$  and  $F_k$  denote the  $k$ -th channels of  $\mathbf{a}$  and  $F$ , respectively;  $\mathbf{e}_k$  is the corresponding RGB basis vector,  $\mathbf{1}$  denotes the RGB all-ones vector.

Following Eq. (15) of the main paper, the resulting adjoints are propagated to each contributing Gaussian as

$$\begin{aligned} \bar{\mathbf{a}}_j &+= \mathbf{w}_j^{(i)} \boldsymbol{\eta}_i^\top \frac{\partial f_i}{\partial \mathbf{a}}, \\ \bar{\mathbf{r}}_j &+= \mathbf{w}_j^{(i)} \left( \boldsymbol{\eta}_i^\top \frac{\partial f_i}{\partial r} + \boldsymbol{\gamma}_i^\top \frac{\partial d_i}{\partial r} \right), \\ \bar{\mathbf{m}}_j &+= \mathbf{w}_j^{(i)} \boldsymbol{\eta}_i^\top \frac{\partial f_i}{\partial m}, \end{aligned} \quad (9)$$

where  $\boldsymbol{\eta}_i$  and  $\boldsymbol{\gamma}_i$  denote the adjoints of the cosine-weighted BRDF term and replayed ray direction, respectively.

For completeness, the normalized aggregation also induces an adjoint with respect to the unnormalized contribution  $q_j^{(i)}$ . With  $\mathbf{w}_j^{(i)} = q_j^{(i)} / Q_i$  and  $Q_i = \sum_{k \in C_i} q_k^{(i)}$ , the corresponding adjoint is

$$\bar{q}_j^{(i)} = \frac{\bar{s}_{p_i}^\top (\mathbf{g}_j - \mathbf{s}_{p_i})}{Q_i}. \quad (10)$$

Material attributes are updated using Eq. (9), while Gaussian opacity and geometry remain fixed throughout the inverse-rendering stage and therefore receive no gradient updates.

### C Geometry Optimization

Geometry plays an important role in inverse rendering, as accurate geometric information is essential for reliable surface interaction,

visibility estimation, and material recovery. Following existing methods [Gao et al. 2024; Gu et al. 2025], we adopt a lightweight strategy that uses pseudo normals estimated from rendered depth to supervise the optimized Gaussian normals.

Given the predicted distance map and the corresponding camera ray directions, we first reconstruct a pseudo surface and estimate a pseudo normal target  $\hat{\mathbf{N}}$ . The predicted normal map  $\mathbf{N}$  is then supervised by a cosine loss on valid foreground pixels, with a linearly increasing weight during training:

$$\mathcal{L}_n = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} (1 - \langle \mathbf{N}(\mathbf{u}), \hat{\mathbf{N}}(\mathbf{u}) \rangle), \quad \lambda_n(t) = \lambda_n \min\left(\frac{t}{T}, 1\right), \quad (11)$$

where  $\Omega$  denotes the set of valid foreground pixels,  $t$  is the current training iteration, and  $T$  is the total number of iterations. This warm-up avoids enforcing noisy pseudo-normal targets too strongly at the early stage of optimization.

Following GS-ID [Du et al. 2024], we further regularize the optimized normals using dataset-provided normal priors [Zeng et al. 2024] and an image-space smoothness term. The provided normal maps are loaded and transformed into world space to obtain  $\mathbf{N}^p$ . We use an  $\ell_2$  loss for the normal prior, and adopt the edge-aware total variation loss defined in Eq. (18) for normal smoothness, using the ground-truth RGB image as the edge reference:

$$\mathcal{L}_p = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \|\mathbf{N}(\mathbf{u}) - \mathbf{N}^p(\mathbf{u})\|_2^2, \quad \mathcal{L}_{tv} = \mathcal{L}_{TV}^{\text{edge}}(\mathbf{N}, \mathbf{I}), \quad (12)$$

where  $\mathbf{I}$  is the ground-truth RGB image. The foreground mask is used when computing both regularization terms. The normal prior is applied throughout training without warm-up, while the normal smoothness term is only used during the geometry and normal initialization stage and is disabled in the subsequent PBR inverse-rendering stage.

In addition, we adopt a mask binary cross-entropy loss to encourage foreground opacity toward 1 and background opacity toward 0:

$$\mathcal{L}_{\text{mask}} = -\frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} [M(\mathbf{p}) \log \hat{A}(\mathbf{p}) + (1 - M(\mathbf{p})) \log(1 - \hat{A}(\mathbf{p}))], \quad (13)$$

where  $M(\mathbf{p}) \in \{0, 1\}$  denotes the foreground mask and  $\hat{A}(\mathbf{p})$  is the rendered opacity at pixel  $\mathbf{p}$ . This suppresses floating Gaussians and keeps foreground opacity away from the hard-threshold boundary.

The overall normal-related objective is therefore:

$$\mathcal{L}_N = \lambda_n(t) \mathcal{L}_n + \lambda_p \mathcal{L}_p + \lambda_{tv} \mathcal{L}_{tv} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}. \quad (14)$$

We set  $\lambda_n = 0.01$ ,  $\lambda_{tv} = 0.01$ , and  $\lambda_{\text{mask}} = 0.1$  in all experiments. The prior term is enabled only for the Hotdog scene and the Synthetic4Relight scenes, with  $\lambda_p = 0.01$ ; it is disabled for all other scenes. Since normal supervision can also affect geometry-related Gaussian parameters through rendering weights and visibility, we reuse the densification strategy of vanilla 3D Gaussian Splatting [Kerbl et al. 2023], including its clone and split operations. The optimized geometry and normal results are visualized in Fig. 1 and Figs. 10–16.

Ray tracing determines Gaussian interactions according to their peak responses along each ray rather than center depths [Zhang et al. 2026], providing more accurate geometric cues for normal estimation. Moreover, the volumetric flexibility of 3D Gaussians can better preserve complex local geometric details in challenging regions. As shown in Fig. 2, our framework recovers finer normal details on the Lego scene.

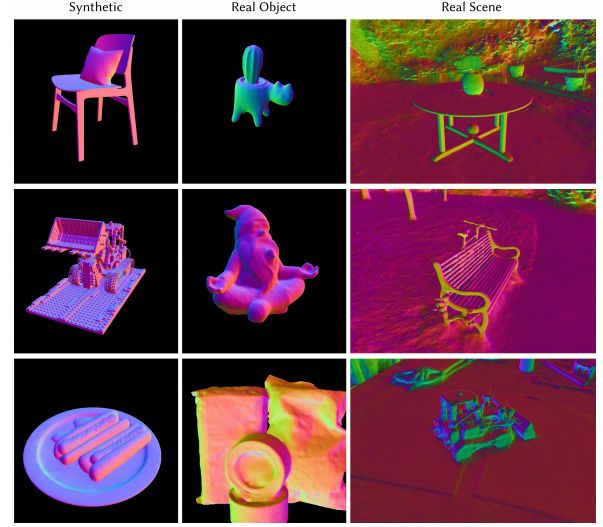


Fig. 1. Normal results on synthetic objects, real objects, and real scenes.

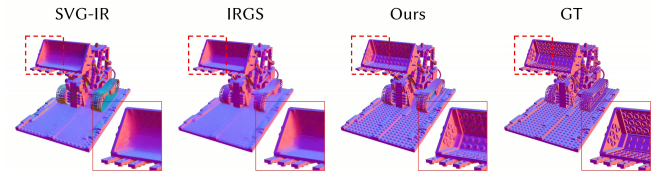


Fig. 2. Lego Normal Qualitative Comparisons(w/o Prior), showing that our framework recovers finer normal details.

## D Extensive Results

### D.1 Additional Cornell Box Results

Fig. 3 shows that increasing the maximum number of bounces captures more indirect illumination in the Cornell-box scene. The difference becomes minor beyond 4 bounces, indicating that 4-bounce transport is sufficient for this scene. Fig. 4 shows material editing results, where albedo and roughness changes affect both surface appearance and surrounding indirect illumination, demonstrating physically consistent responses under the same path-traced transport model.

### D.2 Heterogeneous Lighting and Geometry

Our framework resolves visibility and shadowing through ray tracing rather than precomputed visibility or heuristic approximations. This naturally supports different lighting configurations, including environment maps, point lights, and area lights, while allowing multiple geometric representations to be handled within the same light-transport pipeline. Different geometry types therefore share the same visibility evaluation and physically based light-transport formulation, without requiring representation-specific lighting approximations.

This flexibility is useful for mixed-material estimation in heterogeneous scenes. Regular structures such as walls, floors, and ceilings can be represented efficiently with triangle meshes, while objects



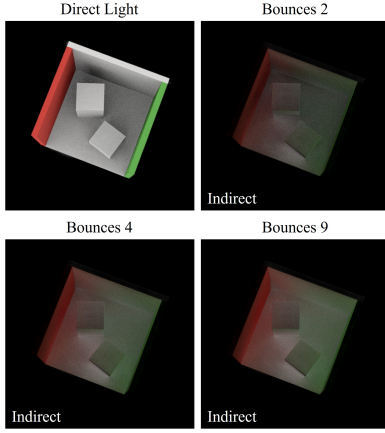


Fig. 3. Rendering results of the recovered Cornell box under different light transport settings. We compare direct illumination with path-traced rendering using up to 2, 4, and 9 bounces. Increasing the bounce count captures more indirect illumination.

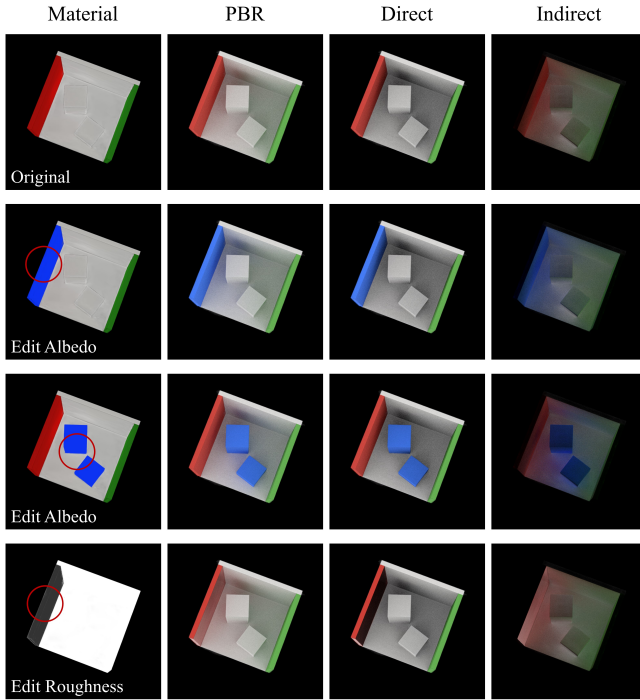


Fig. 4. Material editing results on the recovered Cornell-box scene. Editing albedo and roughness affects both direct appearance and indirect illumination under the same ray-traced transport pipeline.

with complex geometry and local detail can use 3D Gaussians. We demonstrate this capability in a scene containing both representations and multiple material types, illuminated by two area lights and an environment map. As shown in Fig. 7, our method jointly optimizes material properties for different geometry types and recovers consistent material attributes across the mixed scene.

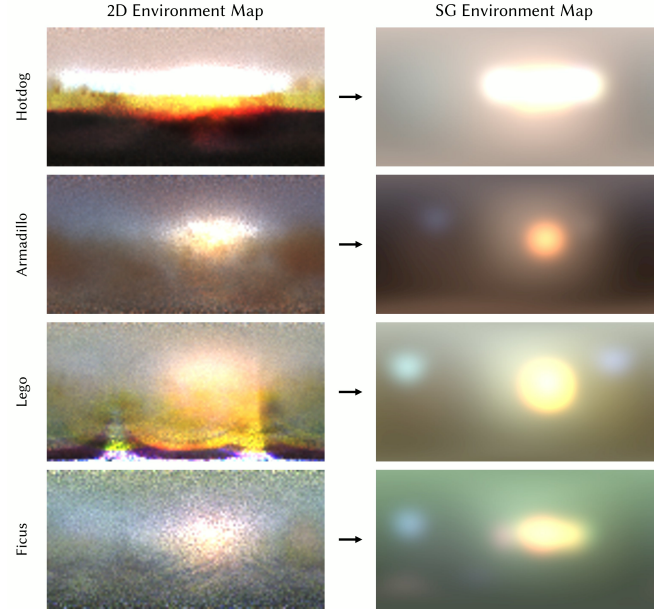


Fig. 5. Visualization of the recovered environment illumination. The compact Spherical-Gaussian representation accurately captures dominant lighting directions while providing a smoother and lower-noise representation in low-energy regions.

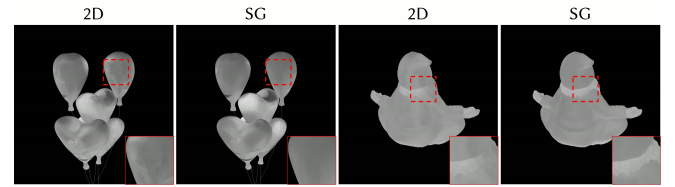


Fig. 6. Comparison of roughness estimation using the SG and 2D environment-map parameterizations. The SG parameterization generally produces smoother material estimates at the same optimization iteration.

### D.3 Environment Map

The Spherical Gaussian representation, with substantially fewer parameters, constrains the illumination space, facilitating faster convergence and smoother optimization gradients. Fig. 5 shows that the compact SG representation accurately captures dominant lighting directions while providing smoother estimates in low-energy regions. Fig. 6 further shows that the SG parameterization generally yields smoother material estimates.

For visualization and comparison, we also use the  $128 \times 64$  2D environment-map representation as existing methods. Fig. 8 compares the illumination recovered by our method with that of existing splatting-based inverse-rendering approaches.

### D.4 Extensive Qualitative Results

We provide additional qualitative results and comparisons in Figs. 9–16 to further demonstrate the effectiveness of our framework.

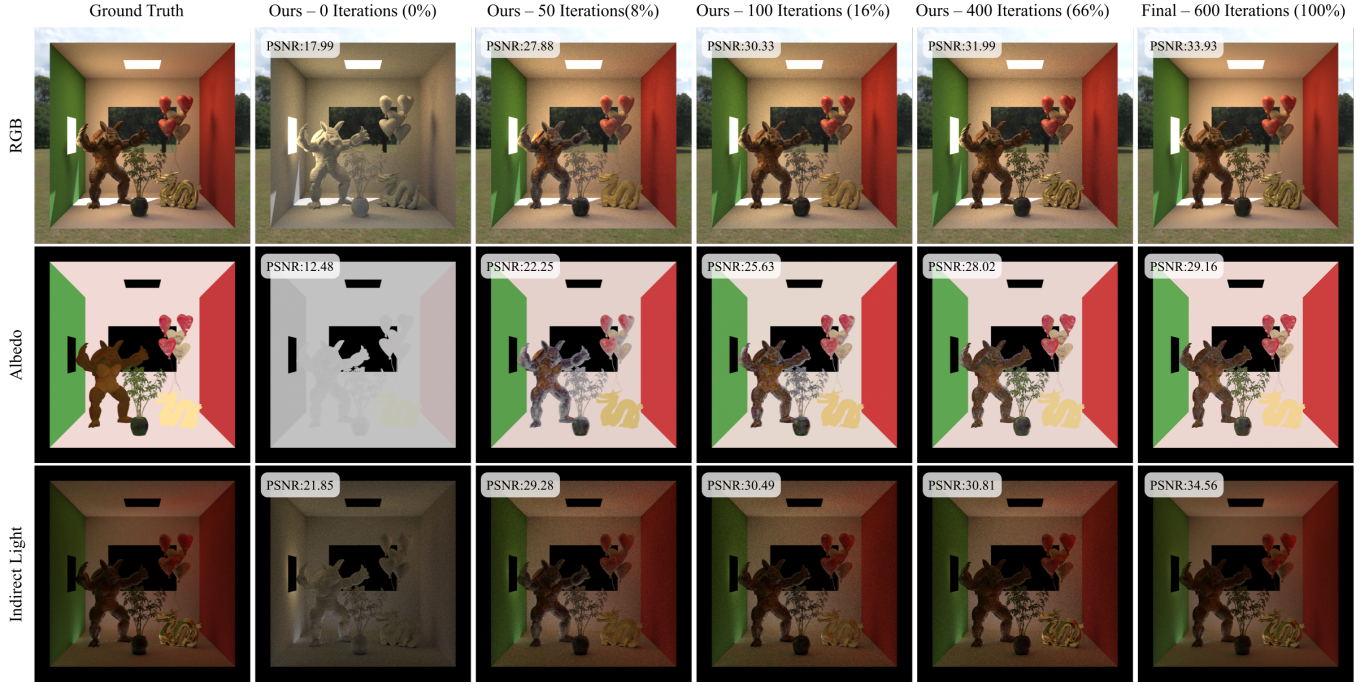


Fig. 7. Heterogeneous lighting and geometry. We construct a Cornell-box-style scene with mixed geometry: the walls and dragon are represented as triangle meshes, while the remaining objects are 3D Gaussian models. Such a hybrid representation is practical: regular structures such as walls can be modeled compactly with meshes, while objects with more complex local detail can be represented more flexibly using Gaussians. The scene is illuminated by two area lights and an environment map. Under fixed lighting, we perform single-view material estimation for 600 iterations. Our method recovers consistent material attributes across mesh and Gaussian representations under unified ray-traced light transport.

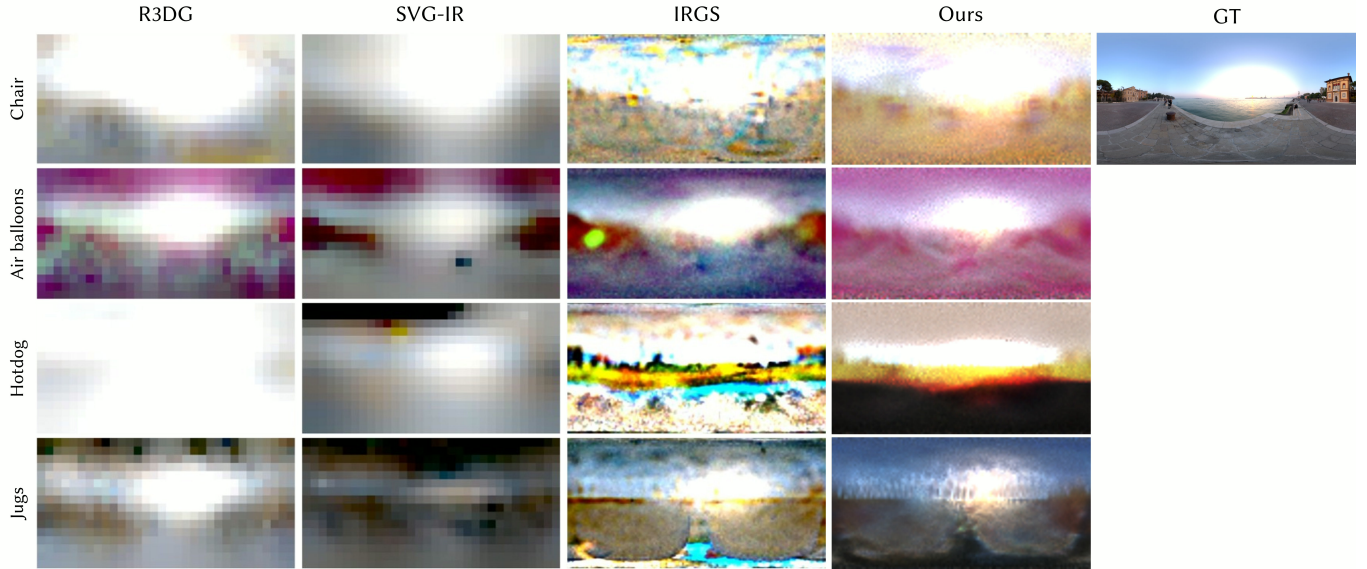


Fig. 8. Comparison of environment maps recovered by our method and existing inverse-rendering approaches on the Synthetic4Relight dataset. Our method accurately recovers the dominant light sources and their spatial distributions.

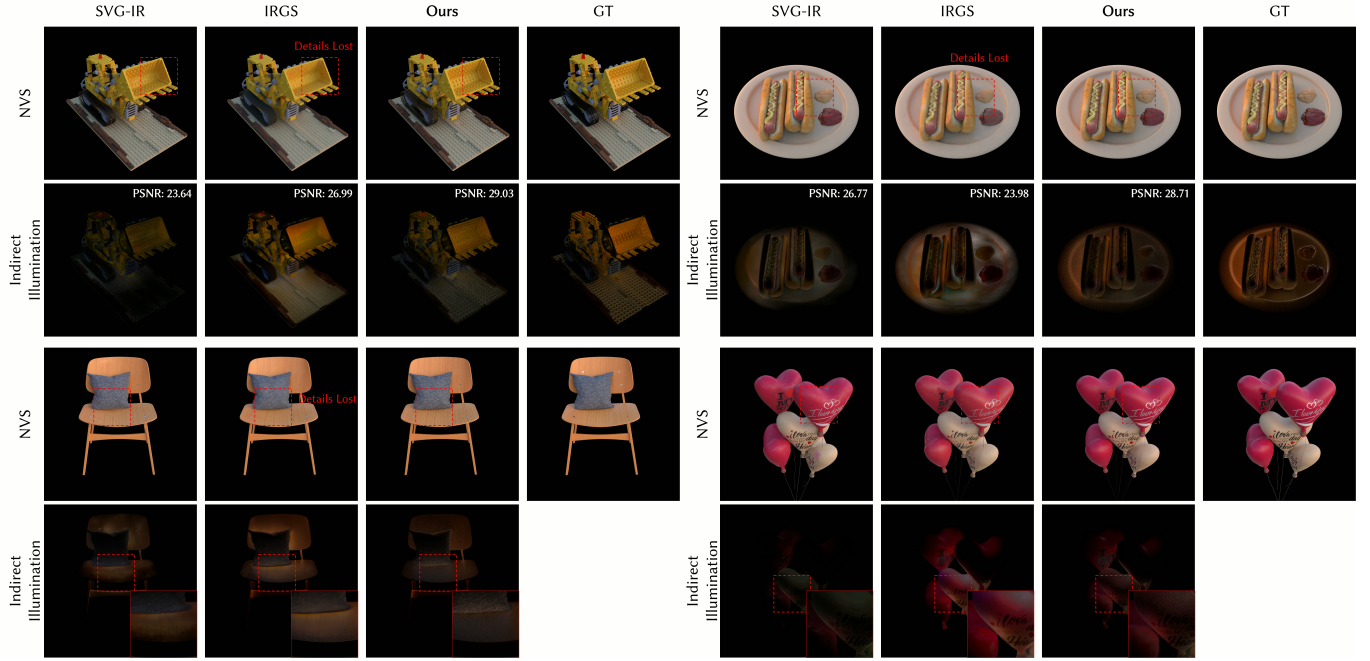


Fig. 9. Novel-view synthesis and indirect illumination results. Our framework outperforms IRGS [Gu et al. 2025] and SVG-IR [Sun et al. 2025] in both novel-view synthesis and indirect illumination modeling.

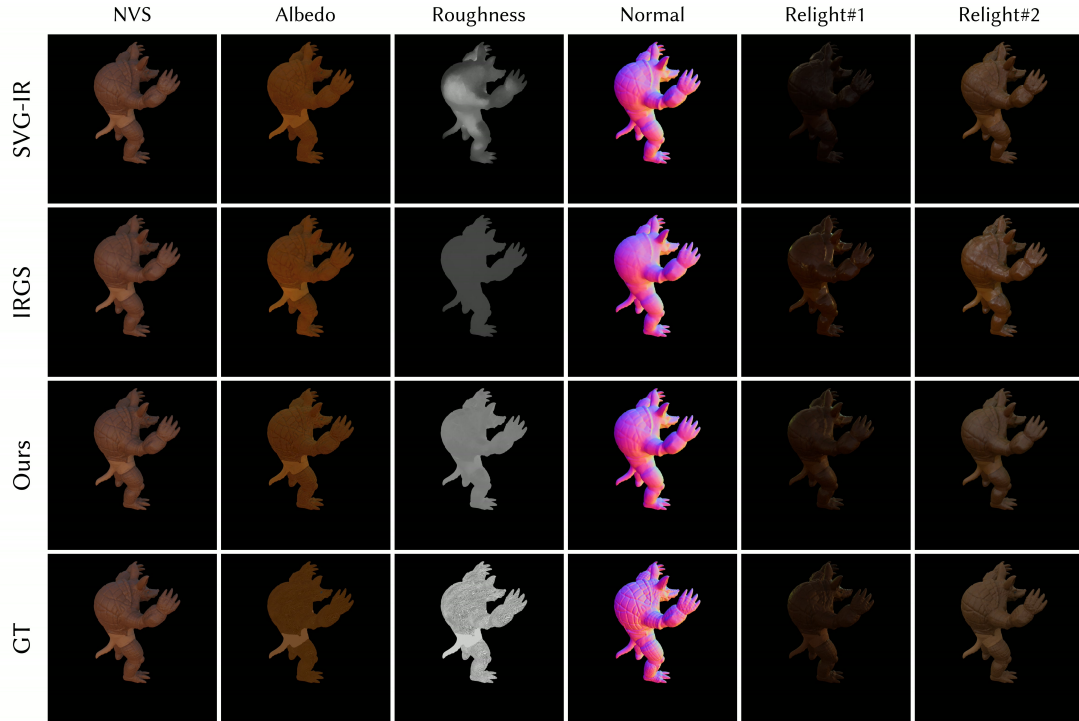


Fig. 10. Qualitative results on the Armadillo scene.



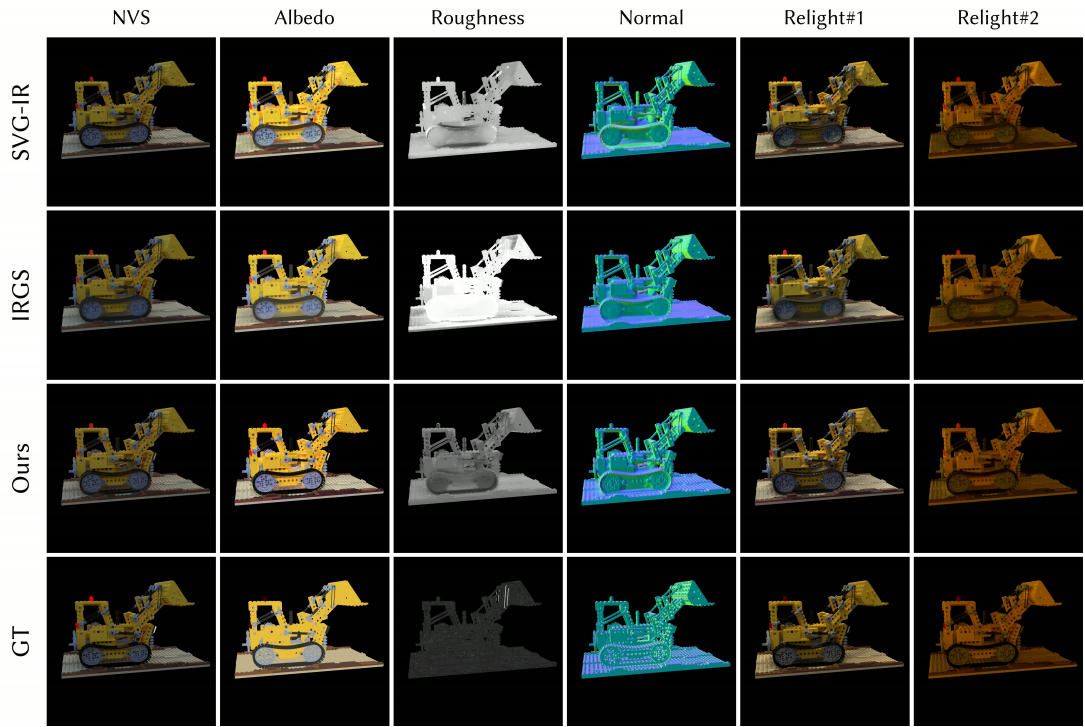


Fig. 11. Qualitative results on the Lego scene.

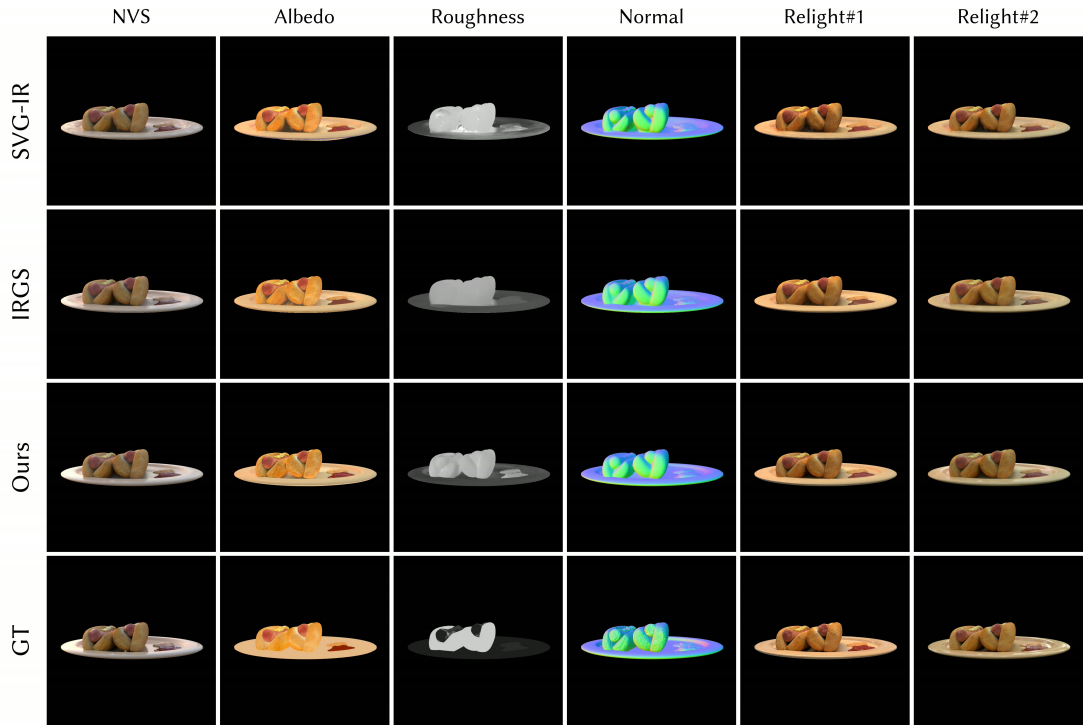


Fig. 12. Qualitative results on the Hotdog scene.



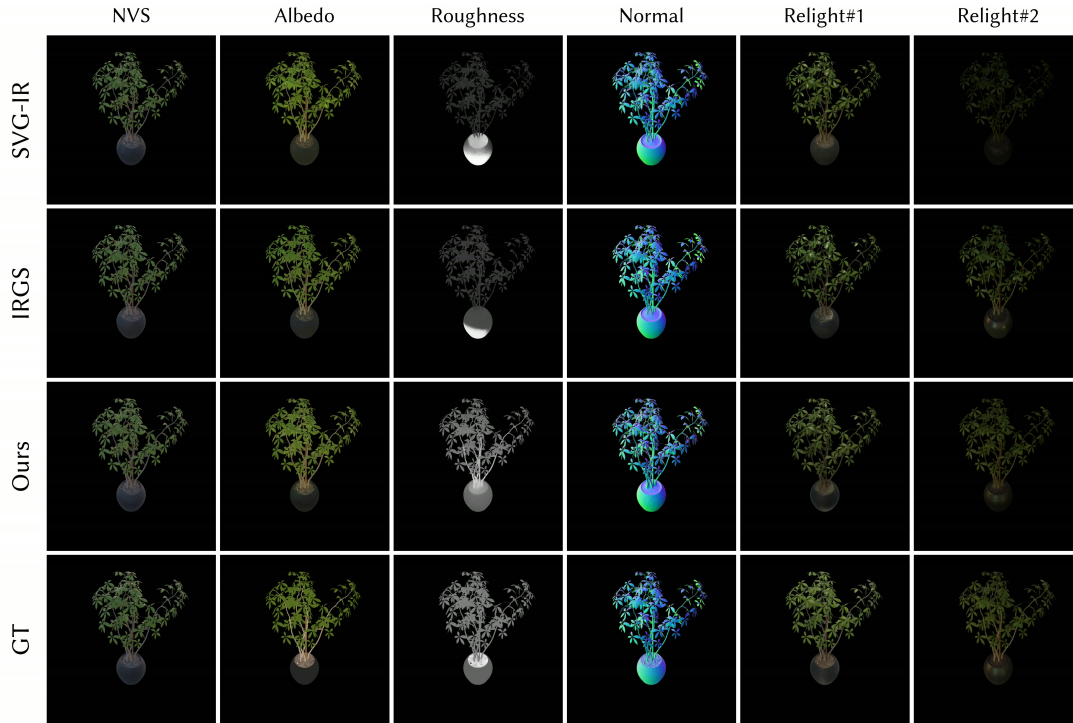


Fig. 13. Qualitative results on the Ficus scene.

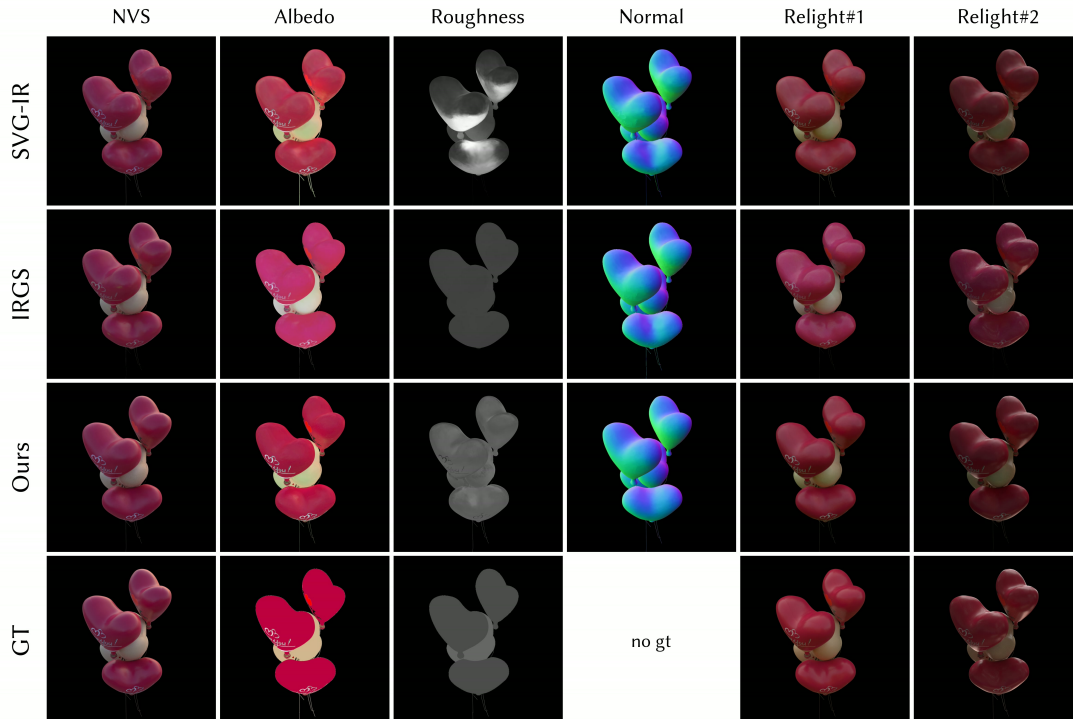


Fig. 14. Qualitative results on the Air Balloons scene.

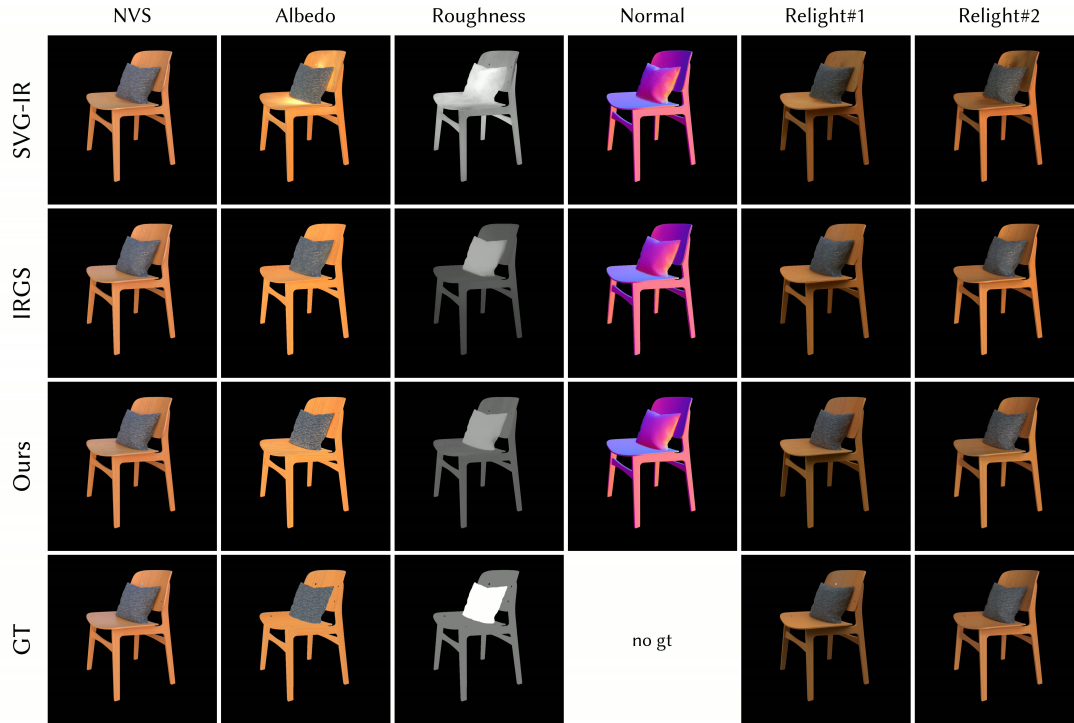


Fig. 15. Qualitative results on the Chair scene.

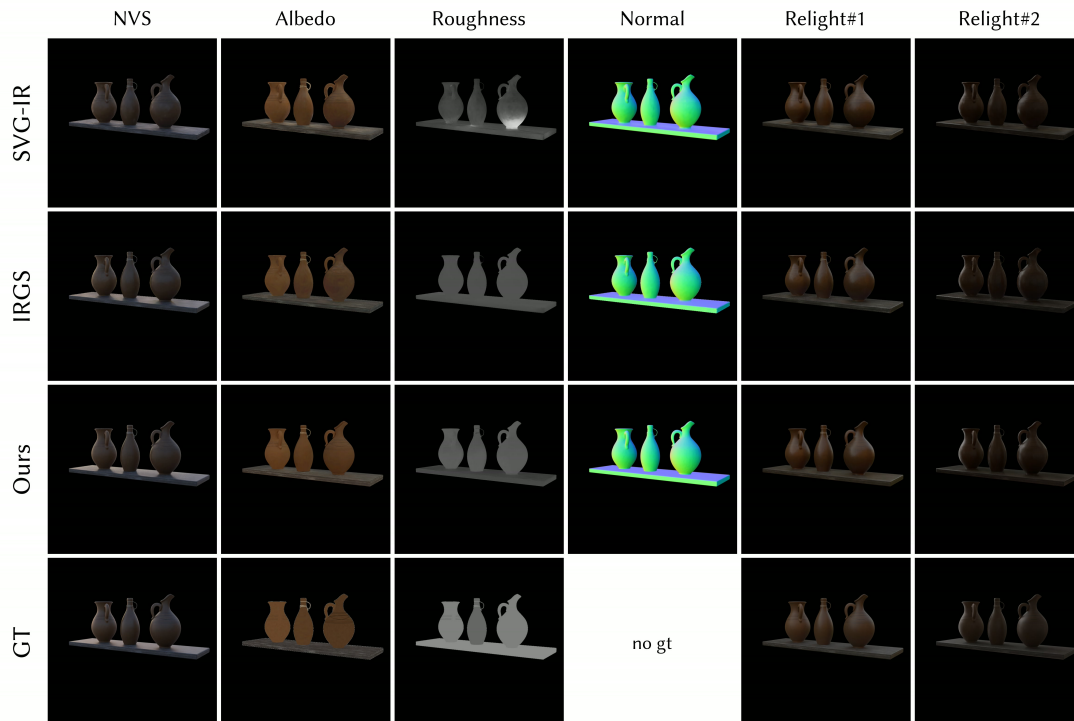


Fig. 16. Qualitative results on the Jugs scene.

## References

- Brent Burley. 2012. Physically-Based Shading at Disney. In *ACM SIGGRAPH Course Notes*. 1–7.
- Kang Du, Zhihao Liang, and Zeyu Wang. 2024. GS-ID: Illumination Decomposition on Gaussian Splatting via Diffusion Prior and Parametric Light Source Optimization. arXiv:2408.08524 [cs.CV] <https://arxiv.org/abs/2408.08524>
- Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. 2024. Relightable 3D Gaussians: Realistic Point Cloud Relighting with BRDF Decomposition and Ray Tracing. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLV* (Milan, Italy). Springer-Verlag, Berlin, Heidelberg, 73–89. doi:10.1007/978-3-031-72995-9\_5
- Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. 2025. IRGS: Inter-Reflective Gaussian Splatting with 2D Gaussian Ray Tracing. In *CVPR*.
- James T. Kajiya. 1986. The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150. doi:10.1145/15886.15902
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023).
- Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. 2025. SVG-IR: Spatially-Varying Gaussian Splatting for Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16143–16152.
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 108:1–108:14.
- Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. 2024. RGB $\leftrightarrow$ X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (*SIGGRAPH '24*). Association for Computing Machinery, New York, NY, USA, Article 75, 11 pages. doi:10.1145/3641519.3657445
- Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiao-Xiao Long, and Ping Tan. 2026. RaDe-GS: Rasterizing Depth in Gaussian Splatting. *ACM Trans. Graph.* 45, 2, Article 19 (March 2026), 14 pages. doi:10.1145/3789201