ELM-GS: ENVIRONMENT LIGHTING MODULATION ON GAUSSIAN SPLATTING FOR EFFICIENT INVERSE RENDERING

Anonymous authors
Paper under double-blind review

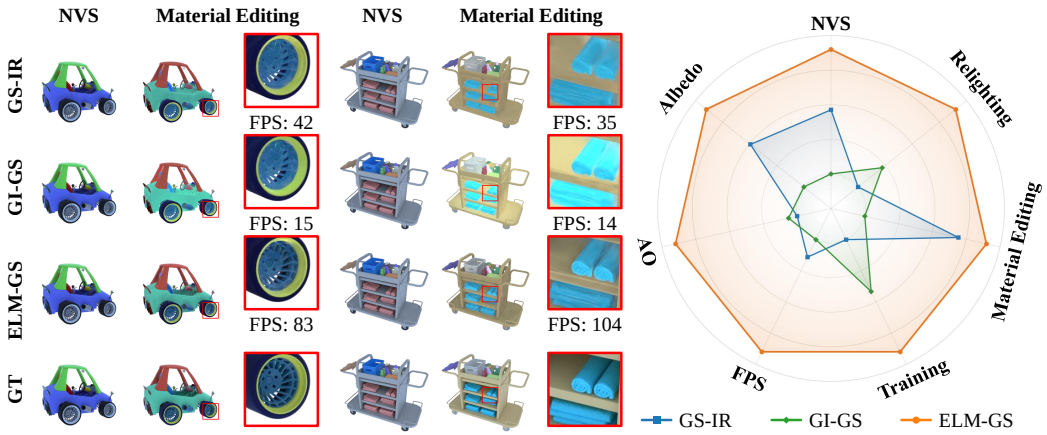

Figure 1: Comparison of AO-based 3DGS inverse rendering methods on novel view synthesis, intrinsic decomposition, and scene editing. Our proposed ELM-GS achieves superior performance across all tasks.

ABSTRACT

Inverse rendering aims to recover intrinsic scene properties from images, enabling applications such as scene editing. Recent Gaussian Splatting (GS)-based methods achieve high-quality Physically-based Rendering (PBR) but often incur substantial computational costs due to expensive ray tracing. Ambient occlusion (AO)-based alternatives reduce this cost, yet occlusion estimation overhead and albedo—shading entanglement still limit their efficiency and editability. Inspired by asset-production pipelines that precompute AO as texture maps for efficient lighting modulation, we propose **ELM-GS**, an efficient 3DGS framework for inverse rendering with environment lighting modulation. ELM-GS introduces a lighting-aware scene representation that decodes learnable features into primitive-wise AO and environment compensation (EC) terms, which respectively attenuate and enhance local environment illumination. Both modulation terms can be splatted into G-buffer textures for efficient PBR. To improve physical plausibility, we derive multi-view occlusion priors via Screen Space Ambient Occlusion to explicitly supervise AO learning. We further construct **Mat-Edit**, a synthetic benchmark with pixel-level material annotations to support fine-grained material editing, together with ground-truth AO and material-edited images for comprehensively evaluating the performance of inverse rendering. Experiments show that ELM-GS achieves state-of-the-art efficiency in inverse rendering and novel view synthesis while outperforming existing AO-based 3DGS methods in intrinsic decomposition and scene editing. The code will be made publicly available.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Inverse rendering aims to recover scene geometry, materials, and illumination from multi-view images, enabling downstream applications such as relighting and material editing. However, ambiguities among these intrinsic factors make the problem inherently ill-posed. Early methods Zhang et al. (2021); Boss et al. (2021); Xu et al. (2023); Wang et al. (2024); Wu et al. (2025a) combine Neural Radiance Fields (NeRF) Mildenhall et al. (2020) with Physically-based Rendering (PBR), but suffer from the high computational cost of volume rendering. More recently, 3D Gaussian Splatting (3DGS) Kerbl et al. (2023) has been extended to inverse rendering Gao et al. (2024); Shi et al. (2025); Bai et al. (2025), substantially improving optimization and rendering efficiency. Nevertheless, evaluating environment illumination requires sampling multiple incident directions and estimating visibility, typically through expensive ray tracing. To reduce this cost, recent approaches Liang et al. (2024); Chen et al. (2025); Liang et al. (2025) incorporate ambient occlusion (AO) using precomputed occlusion volumes or runtime path tracing. Despite further efficiency gains, these methods still incur substantial overhead in training and rendering, while simplified illumination modeling may cause residual shading effects to be baked into the albedo estimates. Therefore, existing AO-based 3DGS inverse rendering methods face challenges in both computational efficiency and accurate intrinsic decomposition, which can further degrade downstream relighting and material editing.

In practical asset-production pipelines, a widely adopted strategy is to precompute AO and store it as artist-refined texture maps like material properties Kontkanen & Laine (2005); McGraw & Sowers (2008). During rendering, it modulates diffuse lighting to approximate complex ray tracing while preserving local occlusion effects. Motivated by this insight, we investigate how the modulation term can be better designed to improve AO-based 3DGS inverse rendering. We first consider whether representing AO in the same manner as material parameters and rendering it into G-buffer textures through a single splatting pass supports both efficient and accurate estimation. Building upon this, we further explore extending the modulation term beyond attenuation to also enhance illumination, aiming to improve intrinsic decomposition and thereby benefit downstream scene editing.

In this paper, we propose ELM-GS, a 3DGS-based inverse rendering framework that enables efficient Physically-based Rendering through environment lighting modulation. We first develop a lighting-aware scene representation that decodes each learnable lighting feature into directional visibilities, from which AO and bent normal are derived to capture local occlusion in diffuse lighting. An environment compensation (EC) term is also decoded from the same feature to reduce albedo–shading entanglement. Based on this representation, material properties, AO, and other G-buffer textures are efficiently rendered through differentiable tile rasterizer and subsequently processed by deferred shading for ELM-based rendering. To provide explicit supervision for AO estimation, we further derive multi-view occlusion priors from rendered position and normal maps using Screen Space Ambient Occlusion (SSAO). During optimization, these geometry-derived AO maps serve as pseudo ground truth to supervise occlusion estimation alongside conventional shading, material, and lighting losses, encouraging physically plausible AO predictions. To facilitate fine-grained material editing and more comprehensive evaluation of inverse rendering and its downstream applications, we further construct the Mat-Edit synthetic benchmark. Mat-Edit provides pixel-level material annotations that can be associated with Gaussian primitives, together with ground-truth AO maps and material-edited images. As shown in Figure 1, compared with existing AO-based 3DGS inverse rendering methods, our approach achieves improved performance across novel view synthesis, intrinsic decomposition, and scene editing, while substantially enhancing both optimization and rendering efficiency. In summary, the contributions of this work are as follows:

- We propose ELM-GS, an efficient 3DGS-based inverse rendering framework that introduces environment lighting modulation into deferred shading. Its lighting-aware scene representation jointly models ambient occlusion and environment compensation, enabling efficient PBR while improving the performance of inverse rendering.
- We introduce geometry-derived multi-view occlusion priors based on SSAO to provide explicit supervision for AO learning, encouraging more physically plausible occlusion estimation without substantially increasing optimization overhead.
- We construct Mat-Edit, a synthetic benchmark with pixel-level material annotations, ground-truth AO maps, and material-edited images, supporting fine-grained material editing and more comprehensive evaluation of inverse rendering methods.

2 Related Works

Novel View Synthesis. Novel view synthesis aims to render photorealistic images from unseen viewpoints given multi-view observations. Neural Radiance Fields (NeRF) Mildenhall et al. (2020) and its variants Barron et al. (2021); Müller et al. (2022); Yang et al. (2025) achieve high-quality scene representation but suffer from inefficient rendering, while 3D Gaussian Splatting (3DGS) Kerbl et al. (2023) and subsequent advances Lu et al. (2024); Zhou & Ni (2025); Tang et al. (2026) enable high-fidelity reconstruction and real-time rendering with explicit Gaussian primitives. However, conventional 3DGS encodes appearance with spherical harmonics, entangling material properties and illumination and thus limiting downstream tasks such as scene editing.

3DGS-based Inverse Rendering. Inverse rendering recovers geometry, material properties, and illumination from multi-view images, but remains challenging due to its ill-posed nature. Recent 3DGS-based methods Wu et al. (2025b); Zhou et al. (2025); Ye et al. (2025); Sun et al. (2025); Bo et al. (2026); Niu et al. (2026) address this issue by incorporating Physically-based Rendering (PBR), typically computing illumination under image-based lighting by sampling incident radiance along multiple directions using computationally expensive ray tracing. To reduce this overhead, several works Liang et al. (2024); Chen et al. (2025); Liang et al. (2025) introduce ambient occlusion (AO) to approximate shading effects, but both precomputed occlusion volumes and runtime AO estimation introduce additional optimization and rendering overhead. Moreover, the simplified PBR modeling can entangle unresolved shading effects with albedo estimates, degrading intrinsic decomposition as well as downstream scene editing.

3 Preliminaries

3D Gaussian Splatting. 3DGS Kerbl et al. (2023) represents a scene using a set $\boldsymbol{\mathcal{G}}$ with N Gaussian primitives, each parameterized by a position $\boldsymbol{\mu}_i$, covariance $\boldsymbol{\Sigma}_i$, spherical harmonics coefficients $\boldsymbol{SH}_i$, and opacity α_i. The geometric shape G_i of primitive $\mathcal{G}_i$ is defined as:

$$G_i(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right), \tag{1}$$

where $\boldsymbol{x}$ denotes a coordinate in 3D space. For view v, each primitive $\mathcal{G}_i$ is projected onto the image plane as an ellipse G_i^v. These Gaussians are then composited in front-to-back order via alpha blending to compute the rendered color $\mathcal{R}_v(\boldsymbol{u})$ at pixel $\boldsymbol{u}$:

$$\mathcal{R}_v(\boldsymbol{u}) = \sum_{i=1}^{N} w_i^v(\boldsymbol{u})\boldsymbol{C}_i^v, \qquad w_i^v(\boldsymbol{u}) = \alpha_i G_i^v(\boldsymbol{u}) \prod_{j=1}^{i-1}\big(1-\alpha_j G_j^v(\boldsymbol{u})\big), \tag{2}$$

where $\boldsymbol{C}_i^v$ is the view-dependent color evaluated from $\boldsymbol{SH}_i$, and $w_i^v(\boldsymbol{u})$ is the blending weight.

Physically-based Rendering. Physically-based Rendering (PBR) Kajiya (1986) computes outgoing radiance L_o in direction $\boldsymbol{\omega}_o$ at surface point $\boldsymbol{x}$ with normal $\boldsymbol{n}_{\boldsymbol{x}}$ via the following equation:

$$L_o(\boldsymbol{\omega}_o, \boldsymbol{x}) = \int_\Omega f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i, \boldsymbol{x}) L_i(\boldsymbol{\omega}_i, \boldsymbol{x})(\boldsymbol{\omega}_i \cdot \boldsymbol{n}_{\boldsymbol{x}})\, d\boldsymbol{\omega}_i, \tag{3}$$

where Ω is the normal-oriented upper hemisphere, L_i is the incident radiance in direction $\boldsymbol{\omega}_i$ and f_r is the bidirectional reflectance distribution function (BRDF). Under image-based lighting, L_i can be decomposed into direct radiance L_i^{dir} from the environment map and indirect radiance L_i^{ind} from surface inter-reflections, yielding corresponding outgoing components L^{dir} and L^{ind}:

$$L_i(\boldsymbol{\omega}_i, \boldsymbol{x}) = V_{\boldsymbol{x}}^{\boldsymbol{\omega}_i} L_i^{dir}(\boldsymbol{\omega}_i) + L_i^{ind}(\boldsymbol{\omega}_i, \boldsymbol{x}), \qquad L_o(\boldsymbol{\omega}_o, \boldsymbol{x}) = L^{dir}(\boldsymbol{\omega}_o, \boldsymbol{x}) + L^{ind}(\boldsymbol{\omega}_o, \boldsymbol{x}), \tag{4}$$

where $V_{\boldsymbol{x}}^{\boldsymbol{\omega}_i}$ is the visibility of point $\boldsymbol{x}$ in direction $\boldsymbol{\omega}_i$. According to the Cook-Torrance BRDF model Cook & Torrance (1982), f_r is decomposed into diffuse f_d and specular f_s terms, which contribute to the diffuse L_d^{dir} and specular L_s^{dir} components of direct radiance, respectively:

$$f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i, \boldsymbol{x}) = \underbrace{(1-m_{\boldsymbol{x}})\frac{\boldsymbol{a}_{\boldsymbol{x}}}{\pi}}_{\text{diffuse } f_d(\boldsymbol{x})} + \underbrace{\frac{DFG}{4(\boldsymbol{\omega}_i \cdot \boldsymbol{n}_{\boldsymbol{x}})(\boldsymbol{\omega}_o \cdot \boldsymbol{n}_{\boldsymbol{x}})}}_{\text{specular } f_s(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i, \boldsymbol{x})}, \qquad L^{dir}(\boldsymbol{\omega}_o, \boldsymbol{x}) = L_d^{dir}(\boldsymbol{x}) + L_s^{dir}(\boldsymbol{\omega}_o, \boldsymbol{x}), \tag{5}$$

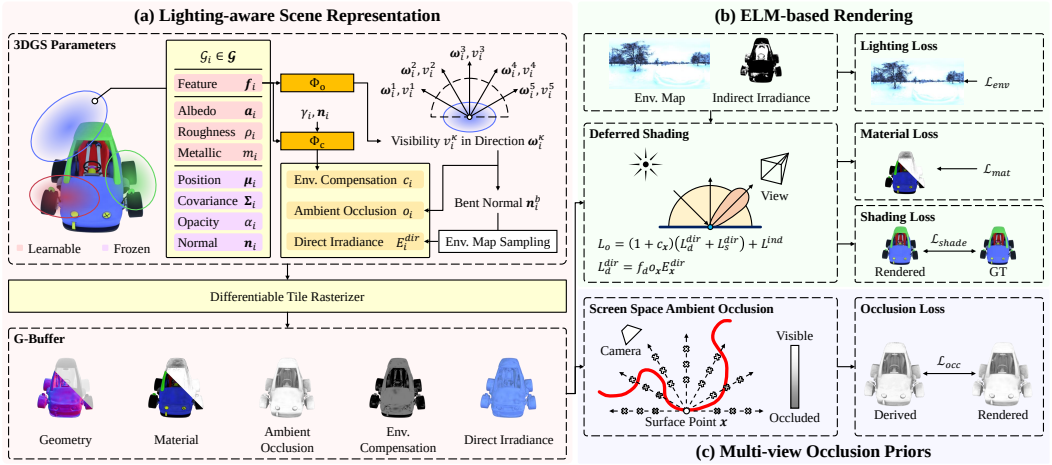

Figure 2: Overview of the proposed ELM-GS framework. Our **lighting-aware scene representation** (a) decodes primitive-wise lighting features into ambient occlusion and environment compensation, which are then rendered into G-buffer textures in a single splatting pass. These terms then modulate environment lighting in **ELM-based rendering** (b), with environment compensation further capturing local illumination variations to alleviate albedo–shading entanglement. During optimization, **multi-view occlusion priors** (c) derived via Screen Space Ambient Occlusion further supervise the learning of AO terms, promoting physically plausible estimations.

where $\boldsymbol{a}_{\boldsymbol{x}}$ and $m_{\boldsymbol{x}}$ denote albedo and metallic, F is the Fresnel term, D and G denote the microfacet distribution function and geometric shadowing factor associated with roughness $\rho_{\boldsymbol{x}}$, respectively.

Ambient Occlusion. To accelerate diffuse lighting evaluation, ambient occlusion (AO) approximates the visibility-weighted lighting integral using a cubemap lookup followed by scalar modulation. Specifically, AO is defined as the cosine-weighted average visibility over the hemisphere:

$$o_{\boldsymbol{x}} = \frac{1}{\pi} \int_{\Omega} V_{\boldsymbol{x}}^{\boldsymbol{\omega}_i} (\boldsymbol{\omega}_i \cdot \boldsymbol{n}_{\boldsymbol{x}}) \, d\boldsymbol{\omega}_i, \tag{6}$$

where $o_{\boldsymbol{x}}$ denotes the AO value at a surface point $\boldsymbol{x}$. The diffuse direct component of the outgoing radiance can then be approximated as:

$$L_d^{dir}(\boldsymbol{x}) = f_d(\boldsymbol{x}) \int_{\Omega} V_{\boldsymbol{x}}^{\boldsymbol{\omega}_i} L_i^{dir}(\boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \boldsymbol{n}_{\boldsymbol{x}}) \, d\boldsymbol{\omega}_i \approx f_d(\boldsymbol{x}) o_{\boldsymbol{x}} E_{\boldsymbol{x}}^{dir}, \tag{7}$$

where $E_{\boldsymbol{x}}^{dir} = \mathrm{Env}_d(\boldsymbol{n}_{\boldsymbol{x}})$ is the unoccluded environment irradiance, obtained from a pre-integrated diffuse cubemap Env_d via normal $\boldsymbol{n}_{\boldsymbol{x}}$.

4 Method

4.1 Overview

AO-based 3DGS inverse rendering typically relies on geometry-dependent baking or path tracing for occlusion estimation, introducing additional computational overhead. Moreover, simplified illumination modeling may lead to compensatory solutions in which local shading effects are baked into albedo estimates. Inspired by asset-production pipelines that use precomputed AO texture maps to efficiently modulate environment lighting, we propose ELM-GS with a lighting-aware scene representation that learns lighting modulation as primitive-wise attributes. As illustrated in Figure 2, ELM-GS extends conventional AO-based attenuation with environment compensation (EC) to account for local illumination variations and reduce albedo–shading entanglement. Both learned modulation terms can be rendered as G-buffer textures in a single splatting pass to enable efficient deferred shading, while multi-view occlusion priors derived using Screen Space Ambient Occlusion further guide AO optimization toward more physically plausible occlusion effects. With these

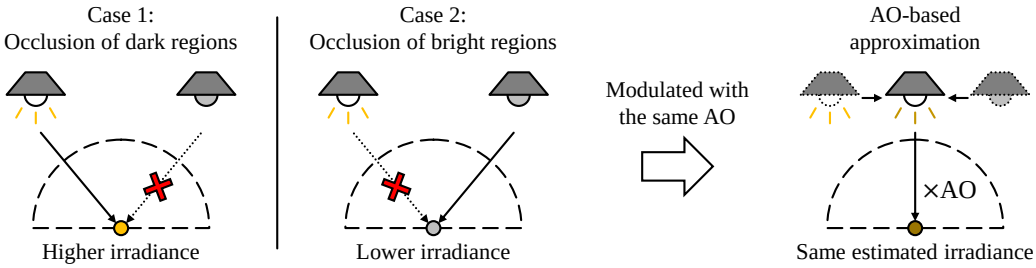

Figure 3: AO-based lighting ambiguity. Occluding dark and bright regions results in different received irradiance, yet identical AO values and surface normals yield the same irradiance estimate.

components, our method improves both the efficiency and effectiveness of AO-based 3DGS inverse rendering, enabling real-time PBR-based novel view synthesis while achieving superior performance on downstream relighting and material editing.

4.2 Lighting-aware Scene Representation

To efficiently modulate environment lighting, we encode AO and EC using primitive-wise lighting features, which can be rendered into G-buffer textures in a single splatting pass. For each Gaussian $\mathcal{G}_i$, a multilayer perceptron (MLP) Φ_o predicts K directional visibilities from its feature $\boldsymbol{f}_i$, while another MLP Φ_c predicts view-dependent EC:

$$\{v_i^\kappa\}_{\kappa=1}^K = \Phi_o(\boldsymbol{f}_i), \qquad o_i = \frac{1}{K}\sum_{\kappa=1}^{K} v_i^\kappa, \qquad c_i = \Phi_c(\boldsymbol{f}_i, \boldsymbol{\gamma}_i, \boldsymbol{n}_i), \tag{8}$$

where $\boldsymbol{\gamma}_i$ is the viewing direction and $\boldsymbol{n}_i$ is the primitive normal. Each v_i^κ represents visibility along direction $\boldsymbol{\omega}_i^\kappa$, sampled over the upper hemisphere at $\boldsymbol{\mu}_i$ using a Vogel disk pattern, as detailed in the appendix. The predicted visibilities also define a bent normal $\boldsymbol{n}_i^b$ that directs diffuse lighting queries toward unoccluded regions. The sampled diffuse direct irradiance E_i^{dir} is defined as:

$$\boldsymbol{n}_i^b = \frac{\sum_{\kappa=1}^{K} v_i^\kappa \boldsymbol{\omega}_i^\kappa}{\sum_{\kappa=1}^{K} v_i^\kappa}, \qquad E_i^{dir} = \mathrm{Env}_d(\boldsymbol{n}_i^b), \tag{9}$$

where Env_d is the pre-integrated diffuse environment cubemap.

4.3 ELM-based Rendering

Conventional AO modulation approximates directional visibility with a single scalar, failing to distinguish occlusion of dark and bright environment regions. As shown in Figure 3, similar AO values may lead to similar lighting approximations despite substantially different directional illumination. This ambiguity can be further exacerbated by low-resolution environment maps, where each texel conflates lighting from more spatially distinct surface points. A similar limitation exists in specular lighting, which relies on a single reflection-direction lookup from the pre-filtered environment map. Consequently, the more expressive albedo may compensate for discrepancies between the approximated lighting and observed appearance, making intrinsic decomposition more difficult.

To account for local illumination variations inadequately captured by conventional AO, EC modulates the diffuse and specular environment contributions, providing an additional optimization path to facilitate intrinsic decomposition. Following prior work Liang et al. (2024); Chen et al. (2025), we compute the outgoing radiance via deferred shading with environment lighting modulation (ELM):

$$L_o(\boldsymbol{\omega}_o, \boldsymbol{x}) = (1 + c_{\boldsymbol{x}})\left(L_d^{dir}(\boldsymbol{x}) + L_s^{dir}(\boldsymbol{\omega}_o, \boldsymbol{x})\right) + L^{ind}(\boldsymbol{\omega}_o, \boldsymbol{x}), \tag{10}$$

$$L_d^{dir}(\boldsymbol{x}) \approx f_d(\boldsymbol{x}) o_{\boldsymbol{x}} E_{\boldsymbol{x}}^{dir}, \qquad L_s^{dir}(\boldsymbol{\omega}_o, \boldsymbol{x}) \approx \underbrace{\int_\Omega \frac{DFG}{4(\boldsymbol{\omega}_o \cdot \boldsymbol{n}_{\boldsymbol{x}})}\, d\boldsymbol{\omega}_i}_{\text{BRDF integral } R} \underbrace{\int_\Omega DL_i(\boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \boldsymbol{n}_{\boldsymbol{x}})\, d\boldsymbol{\omega}_i}_{\text{Pre-filtered environment map } I_s}, \tag{11}$$

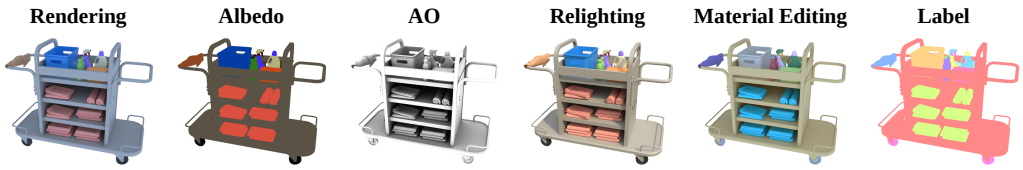

Figure 4: An example view of the Cleaning Cart scene from our proposed Mat-Edit dataset.

where R and I_s are obtained from a precomputed BRDF lookup table and a pre-filtered environment map, respectively, following the split-sum approximation Karis & Games (2013). By combining AO-based attenuation with EC-based compensation, ELM provides a simple yet effective solution for improving intrinsic decomposition while retaining efficient deferred rendering.

4.4 Multi-view Occlusion Priors

Although modulation terms can be learned without direct supervision like other components, the learned AO may absorb material and shading effects, resulting in physically implausible estimates. We therefore derive geometry-based multi-view occlusion priors using Screen Space Ambient Occlusion (SSAO) to explicitly supervise AO learning without expensive global occlusion computation.

Given a certain view, we obtain the camera-space normal and position maps $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\mathcal{P}}$ through differentiable tile rasterizer. For each pixel $\boldsymbol{p}$, we sample M hemispherical directions using a Vogel pattern oriented by $\boldsymbol{\mathcal{N}}(\boldsymbol{p})$, with T points along each direction. Each sample $\boldsymbol{s}_{\boldsymbol{p}}^{m,t}$ is projected onto the image plane at $\boldsymbol{p}^{m,t}$, where the corresponding surface position $\boldsymbol{x}_{\boldsymbol{p}}^{m,t} = \boldsymbol{\mathcal{P}}(\boldsymbol{p}^{m,t})$ is retrieved to compute the visibility $V_{\boldsymbol{p}}^{m,t}$ of $\boldsymbol{s}_{\boldsymbol{p}}^{m,t}$:

$$V_{\boldsymbol{p}}^{m,t} = \mathbb{I}\big(s_{\boldsymbol{p}}^{m,t,z} < x_{\boldsymbol{p}}^{m,t,z}\big)\,, \qquad \boldsymbol{\Psi}_{\boldsymbol{p}}^{m} = \big\{t \mid V_{\boldsymbol{p}}^{m,t} = 0 \wedge t \in \{1,\dots,T\}\big\}\,, \tag{12}$$

where $s_{\boldsymbol{p}}^{m,t,z}$ and $x_{\boldsymbol{p}}^{m,t,z}$ are the camera-space z-coordinates of the sample $\boldsymbol{s}_{\boldsymbol{p}}^{m,t}$ and the retrieved surface position $\boldsymbol{x}_{\boldsymbol{p}}^{m,t}$, respectively, $\mathbb{I}$ is the indicator function, and $\boldsymbol{\Psi}_{\boldsymbol{p}}^{m}$ contains the occluded sample indices. To emphasize local occlusion, directional visibility $V_{\boldsymbol{p}}^{m}$ is computed with distance attenuation based on the nearest occluded sample, and the AO value $\mathcal{O}(\boldsymbol{p})$ at pixel $\boldsymbol{p}$ is defined as:

$$V_{\boldsymbol{p}}^{m} = \begin{cases} 1, & \boldsymbol{\Psi}_{\boldsymbol{p}}^{m} = \emptyset, \\ 1 - \exp\!\left(-\beta \frac{\min(\boldsymbol{\Psi}_{\boldsymbol{p}}^{m})}{T}\right), & \text{otherwise}, \end{cases} \qquad \mathcal{O}(\boldsymbol{p}) = \frac{\zeta}{M} \sum_{m=1}^{M} V_{\boldsymbol{p}}^{m}, \tag{13}$$

where β controls distance attenuation and ζ scales the AO value. Before optimization, we precompute AO maps for all training views via SSAO as supervision. This preprocessing is substantially cheaper than constructing global occlusion volumes, enabling more efficient inverse rendering.

4.5 Optimization

Our method starts from the model pretrained in the initial stage of GS-IR Liang et al. (2024), and performs optimization in the decomposition stage with the decomposition loss $\mathcal{L}_d$ and occlusion loss $\mathcal{L}_{occ}$. Specifically, the overall loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_d + \lambda \mathcal{L}_{occ}, \qquad \mathcal{L}_{occ} = \left\| \mathcal{O} - \hat{\mathcal{O}} \right\|_1, \tag{14}$$

where $\mathcal{O}$ and $\hat{\mathcal{O}}$ denote the SSAO-derived and rendered AO maps, respectively, $\mathcal{L}_d$ follows previous methods Liang et al. (2024); Chen et al. (2025), and λ is the weight of the occlusion loss.

4.6 Primitive-Wise Material Editing

Given pixel-level material annotations, we use FlashSplat Shen et al. (2024) to associate semantic material labels with Gaussian primitives. The material parameters of selected primitives, including albedo, roughness, and metallic, can then be replaced with target values for fine-grained material editing. Since scene geometry and lighting representations remain unchanged, edited scenes can be directly rendered under different material configurations without retraining. Details of the editing procedure are provided in the appendix.

Table 1: Quantitative comparison on novel view synthesis (NVS), albedo estimation and relighting quality on the TensoIR dataset. The best, second-best, and third-best results are highlighted.

Type	Method	NVS			Albedo		Relighting		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
NeRF	NeRFactor	24.68	0.922	0.120	25.13	0.940	23.38	0.908	0.131
	InvRender	27.37	0.934	0.089	27.34	0.933	23.97	0.901	0.101
	TensoIR	35.09	0.976	0.040	29.27	0.950	28.58	0.944	0.081
2DGS+RT	RadioGS	37.86	0.980	0.027	31.05	0.952	32.09	0.953	0.048
3DGS+AO	GS-IR*	34.95	0.964	0.042	30.11	0.922	23.97	0.874	0.109
	GI-GS*	34.35	0.961	0.044	30.39	0.923	23.99	0.877	0.109
	Ours	35.71	0.968	0.040	31.07	0.929	26.03	0.896	0.097

Table 2: Quantitative comparison on novel view synthesis (NVS), intrinsic decomposition and scene editing on the Mat-Edit dataset.

Method	NVS			Albedo		AO		Material Editing			Relighting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	WMSE↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RadioGS	40.42	0.993	0.009	25.43	0.940	—	—	—	—	—	28.34	0.959	0.046
GS-IR*	35.70	0.982	0.030	23.65	0.912	15.70	0.029	23.06	0.935	0.075	21.70	0.884	0.104
GI-GS*	34.68	0.978	0.037	22.62	0.900	16.07	0.025	20.96	0.921	0.084	22.48	0.890	0.099
Ours	36.66	0.983	0.029	24.50	0.913	20.89	0.007	23.69	0.937	0.074	24.85	0.917	0.088

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate ELM-GS on both synthetic and real-world datasets. The synthetic datasets include ten object-centric scenes from TensoIR Jin et al. (2023) and our Mat-Edit. For real-world evaluation, we use four indoor and five outdoor scenes from Mip-NeRF 360 Barron et al. (2022) and two outdoor scenes from Tanks & Temples Knapitsch et al. (2017).

Mat-Edit Benchmark. Existing inverse rendering datasets primarily focus on novel view synthesis, albedo estimation, and relighting, with limited support for quantitative evaluation of shading effects and material editing. To address this limitation, we construct Mat-Edit, a synthetic benchmark comprising six object-centric scenes rendered in Blender from multiple viewpoints with known camera parameters. As illustrated in Figure 4, Mat-Edit provides pixel-level material annotations, ground-truth AO maps, and material-edited reference images. For each annotated material, we additionally provide two sets of material parameters, including albedo, roughness, and metallic values, corresponding to the original and edited configurations. These annotations enable fine-grained material editing, with paired parameters and reference images supporting controlled evaluation.

Baselines. We compare ELM-GS with NeRF-based methods NeRFactor Zhang et al. (2021), InvRender Zhang et al. (2022), and TensoIR Jin et al. (2023), the ray-tracing (RT)-based 2DGS method RadioGS Han et al. (2026), and AO-based 3DGS methods GS-IR Liang et al. (2024) and GI-GS Chen et al. (2025). For public datasets, methods marked with * are retrained using their official codes, while the remaining results are taken from Chen et al. (2025) and Han et al. (2026).

Implementation Details. Since GS-IR, GI-GS, and ELM-GS share the initial stage, we initialize their decomposition stages with the same pretrained model obtained from GS-IR for a fair comparison. All experiments are conducted on a single NVIDIA RTX 4090 GPU with 24 GB of memory.

Metrics. We measure rendering quality using PSNR, SSIM Wang et al. (2004), and LPIPS Zhang et al. (2018), and efficiency using frames per second (FPS) and training time (in seconds). AO quality is additionally evaluated using weighted mean squared error (WMSE) to balance the contributions of occluded and unoccluded regions according to the average AO value. The detailed computation is provided in the appendix.

Table 3: Quantitative comparison on novel view synthesis on the real-world datasets.

Method	Mip-NeRF 360 Indoor			Mip-NeRF 360 Outdoor			Tanks & Temples		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
GS-IR*	28.11	0.882	0.225	23.57	0.679	0.285	22.06	0.775	0.248
GI-GS*	28.52	0.888	0.210	23.95	0.689	0.280	22.54	0.794	0.227
Ours	28.96	0.897	0.207	23.99	0.695	0.278	22.56	0.795	0.231

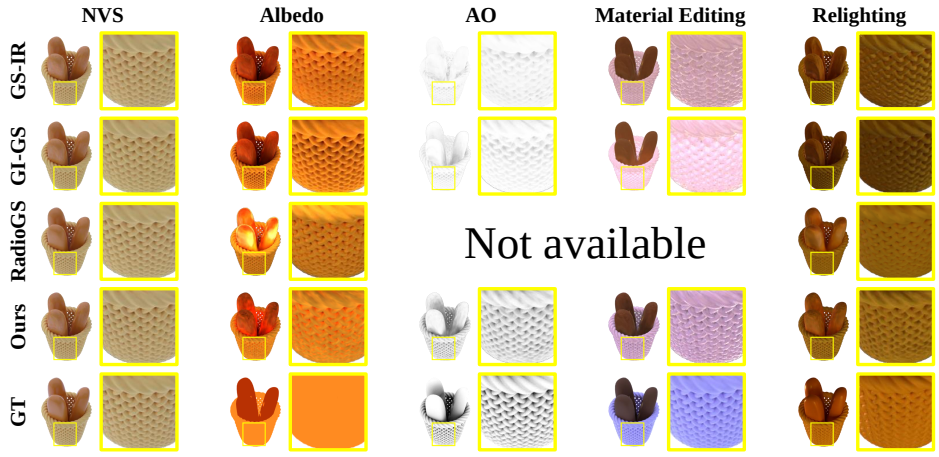

Figure 5: Qualitative comparison on the Mat-Edit synthetic dataset.

5.2 Comparison with State-of-the-Art Methods

Novel View Synthesis. As reported in Tables 1 and 2, ELM-GS achieves the best novel view synthesis performance among AO-based methods on object-centric synthetic scenes and narrows the gap to RT-based approaches. The qualitative results in Figure 5 further show improved rendering fidelity. On larger real-world scenes, where RT-based methods encounter CUDA out-of-memory errors, ELM-GS consistently outperforms existing AO-based counterparts, as shown in Table 3.

Intrinsic Decomposition. Comparing with existing AO-based methods, our ELM-GS achieves more accurate albedo and AO estimation, as shown in Tables 1 and 2. Figure 5 further shows albedo with fewer baked-in shading effects and more faithful local occlusion, alleviating the underestimation observed in competing methods. These results indicate improved intrinsic decomposition, enabling ELM-GS to better support a broader range of downstream applications.

Scene Editing. ELM-GS also improves downstream relighting and material editing, as reported in Tables 1 and 2. Qualitative comparisons in Figure 5 show that our method better preserves geometric details with improved sense of depth in the edited results, reducing the flattened appearance observed in existing AO-based methods. These improvements are consistent with the more accurate intrinsic decomposition achieved by ELM-GS.

Computational Efficiency. Beyond reconstruction and editing quality, ELM-GS substantially improves both optimization and rendering efficiency. On Mat-Edit, ELM-GS achieves more than 2.5× rendering speedup and reduces decomposition-stage optimization time by approximately 30% compared with the most efficient AO-based baseline, as reported in Table 4. Moreover, ELM-GS remains applicable to larger real-world scenes and achieves real-time rendering where RT-based approaches exceed GPU memory capacity, demonstrating favorable scalability with scene complexity.

5.3 Ablation Study

Multi-view Occlusion Prior. Without explicit supervision, learned AO may absorb appearance variations unrelated to geometric occlusion. As illustrated by the "w/o SSAO" setting in Table 5,

Table 4: Quantitative comparison on computational efficiency.

Method	TenoIR		Mat-Edit		Mip-NeRF 360		Tanks & Temples	
	FPS↑	Training↓	FPS↑	Training↓	FPS↑	Training↓	FPS↑	Training↓
RadioGS	3.4	2938	4.0	3140	CUDA out of memory		CUDA out of memory	
GS-IR	22.5	714	36.5	677	16.0	4174	23.3	2600
GI-GS	14.6	683	14.7	740	8.9	2164	17.6	1259
Ours	96.6	497	98.1	495	47.9	1170	66.5	826

Table 5: Ablation study of ELM-GS. Performance is measured in PSNR across all tasks.

Method	TenoIR				Mat-Edit					
	NVS	Albedo	Relighting	FPS	NVS	Albedo	AO	Material Editing	Relighting	FPS
baseline	34.95	30.11	23.97	22.5	35.70	23.65	15.70	23.06	21.70	27.7
SSAO only	34.36	30.15	26.14	0.6	35.21	24.20	21.85	23.34	25.45	0.6
w/o LAO	35.12	30.97	26.24	0.6	35.84	24.39	21.85	23.65	25.33	0.6
w/o LEC	34.85	30.62	25.88	100.7	36.06	24.35	20.87	23.48	25.12	96.8
w/o SSAO	35.99	30.50	25.90	88.7	36.88	24.23	16.88	23.52	23.50	90.3
Full	35.71	31.07	26.03	96.6	36.66	24.50	20.89	23.69	24.85	98.1

removing SSAO-derived supervision substantially degrades decomposition and editing quality relative to the "Full" setting. To further evaluate the priors themselves, we directly adopt SSAO-derived AO maps without EC in the "SSAO only" setting, which yields accurate occlusion estimates and improves both tasks. These results validate the effectiveness of multi-view occlusion priors as explicit guidance for physically plausible AO learning, complementing our efficient rendering formulation.

Learnable Ambient Occlusion. View-wise SSAO evaluation incurs substantial rendering overhead and introduces view-dependent AO variations at each surface point. As illustrated by the "w/o LAO" setting in Table 5, replacing learned AO with SSAO estimation reduces rendering speed and NVS performance. In contrast, by decoding primitive-wise AO from lighting features, our method preserves comparable occlusion accuracy while maintaining view consistency and avoiding costly evaluation during rendering, enabling real-time PBR-based NVS.

Learnable Environment Compensation. EC compensates for unmodeled local illumination variations in the simplified PBR pipeline. As recorded by the"w/o LEC" settings in Table 5, removing EC substantially degrades NVS performance. EC also improves intrinsic decomposition by reducing baked-in shading in the albedo estimates, further benefiting downstream editing. These results confirm that EC provides a simple yet effective solution for improving the performance of AO-based inverse rendering.

6 Conclusion

In this paper, we presented ELM-GS, an efficient 3DGS framework for inverse rendering that addresses the computational overhead of occlusion estimation and albedo–shading entanglement arising from simplified AO-based illumination modeling. ELM-GS introduces environment lighting modulation through learnable AO and EC, which model local occlusion and complementary illumination variations, respectively, while enabling efficient deferred shading. We further derive geometry-based multi-view occlusion priors using SSAO to support physically plausible AO learning. In addition, we introduce Mat-Edit, a synthetic benchmark with pixel-level material annotations that enable fine-grained material editing for 3DGS-based methods. With ground-truth AO maps and material-edited reference images, Mat-Edit further enables a more comprehensive evaluation of inverse rendering methods across intrinsic decomposition and downstream scene editing. Extensive experiments demonstrate that ELM-GS improves novel view synthesis, intrinsic decomposition, and scene editing over existing AO-based methods while substantially reducing optimization and rendering costs, enabling real-time PBR-based rendering even on complex real-world scenes.

AI USE STATEMENT

After completing the initial draft, we use AI tools solely to polish wording and improve language clarity. All scientific content and ideas are developed by the authors.

REFERENCES

Haiyang Bai, Jiaqi Zhu, Songru Jiang, Wei Huang, Tao Lu, Yuanqi Li, Jie Guo, Runze Fu, Yanwen Guo, and Lijun Chen. GaRe: Relightable 3D Gaussian Splatting for Outdoor Scenes from Unconstrained Photo Collections. In *Proceedings of the International Conference on Computer Vision*, pp. 26456–26465, 2025.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A Multiscale Representation for Anti-aliasing Neural Radiance Fields. In *Proceedings of the International Conference on Computer Vision*, pp. 5855–5864, 2021.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

Jinyang Bo, Fan Dou, Wenrui Quan, Shangxun Liu, Yang Xu, Yuhe Zhang, Kang Li, and Guohua Geng. GHPT: Real-Time Relightable Gaussian Splatting using Hybrid Path Tracing. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 25990–25999, 2026.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. NeRD: Neural Reflectance Decomposition from Image Collections. In *Proceedings of the International Conference on Computer Vision*, pp. 12684–12694, 2021.

Hongze Chen, Zehong Lin, and Jun Zhang. GI-GS: Global Illumination Decomposition on Gaussian Splatting for Inverse Rendering. In *Proceedings of the International Conference on Learning Representations*, pp. 41825–41854, 2025.

Robert L Cook and Kenneth E. Torrance. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.

Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3D Gaussians: Realistic Point Cloud Relighting with BRDF Decomposition and Ray Tracing. In *Proceedings of the European Conference on Computer Vision*, pp. 73–89, 2024.

Kyu Beom Han, Jaeyoon Kim, Woo Jae Kim, Jinhwan Seo, and Sung-Eui Yoon. RadioGS: Radiometrically Consistent Gaussian Surfels for Inverse Rendering. In *International Conference on Learning Representations*, pp. 1–33, 2026.

Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. TensoIR: Tensorial Inverse Rendering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2023.

James T Kajiya. The Rendering Equation. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pp. 143–150, 1986.

Brian Karis and Epic Games. Real Shading in Unreal Engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-time Radiance Field Rendering. *ACM Transactions on Graphics*, (4):1–14, 2023.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-scale Scene Reconstruction. *ACM Transactions on Graphics*, 36(4):1–13, 2017.

Janne Kontkanen and Samuli Laine. Ambient Occlusion Fields. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, pp. 41–48, 2005.

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

Zhihao Liang, Hongdong Li, Kui Jia, Kailing Guo, and Qi Zhang. GUS-IR: Gaussian Splatting with Unified Shading for Inverse Rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):8364–8378, 2025.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D Gaussians for View-adaptive Rendering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024.

Tim McGraw and Brian Sowers. Hardware Accelerated Per-texel Ambient Occlusion Mapping. In *Proceedings of the International Symposium on Visual Computing*, pp. 1115–1124, 2008.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision*, pp. 405–421, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022.

Jiahao Niu, Rongjia Zheng, Wenju Xu, Wei-Shi Zheng, and Qing Zhang. SGS-Intrinsic: Semantic-Invariant Gaussian Splatting for Sparse-View Indoor Inverse Rendering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 26021–26030, 2026.

Qiuhong Shen, Xingyi Yang, and Xinchao Wang. FlashSplat: 2D to 3D Gaussian Splatting Segmentation Solved Optimally. In *Proceedings of the European Conference on Computer Vision*, pp. 456–472, 2024.

Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jian Zhang, Bin Zhou, Errui Ding, and Jingdong Wang. GIR: 3D Gaussian Inverse Rendering for Relightable Scene Factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–12, 2025.

Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. SVG-IR: Spatially-varying Gaussian Splatting for Inverse Rendering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 16143–16152, 2025.

Zhenyu Tang, Chaoran Feng, Xinhua Cheng, Wangbo Yu, Junwu Zhang, Yuan Liu, Xiao-Xiao Long, Wenping Wang, and Li Yuan. NeuralGS: Bridging Neural Fields and 3D Gaussian Splatting for Compact 3D Representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9493–9501, 2026.

Haoyuan Wang, Wenbo Hu, Lei Zhu, and Rynson WH Lau. Inverse Rendering of Glossy Objects via the Neural Plenoptic Function and Radiance Fields. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 19999–20008, 2024.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Sean Wu, Shamik Basu, Tim Broedermann, Luc Van Gool, and Christos Sakaridis. PBR-NeRF: Inverse Rendering with Physics-based Neural Fields. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 10974–10984, 2025a.

Tong Wu, Jia-Mu Sun, Yu-Kun Lai, Yuewen Ma, Leif Kobbelt, and Lin Gao. DeferredGS: Decoupled and Relightable Gaussian Splatting with Deferred Shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6307–6319, 2025b.

Yingyan Xu, Gaspard Zoss, Prashanth Chandran, Markus Gross, Derek Bradley, and Paulo Gotardo. ReNeRF: Relightable Neural Radiance Fields with Nearfield Lighting. In *Proceedings of the International Conference on Computer Vision*, pp. 22581–22591, 2023.

Peiqi Yang, Zhangkai Ni, Hanli Wang, Wenhan Yang, Shiqi Wang, and Sam Kwong. Shell-guided Compression of Voxel Radiance Fields. *IEEE Transactions on Image Processing*, 34:1179–1191, 2025.

Kai Ye, Chong Gao, Guanbin Li, Wenzheng Chen, and Baoquan Chen. GeoSplatting: Towards Geometry Guided Gaussian Splatting for Physically-based Inverse Rendering. In *Proceedings of the International Conference on Computer Vision*, pp. 28991–29000, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. NeRFactor: Neural Factorization of Shape and Reflectance under an Unknown Illumination. *ACM Transactions on Graphics*, 40(6):1–18, 2021.

Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling Indirect Illumination for Inverse Rendering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 18643–18652, 2022.

Hongbi Zhou and Zhangkai Ni. Perceptual-GS: Scene-adaptive Perceptual Densification for Gaussian Splatting. In *Proceedings of the International Conference on Machine Learning*, pp. 1–26, 2025.

Yongyang Zhou, Fanglue Zhang, Zichen Wang, and Lei Zhang. RTR-GS: 3D Gaussian Splatting for Inverse Rendering with Radiance Transfer and Reflection. In *Proceedings of the ACM International Conference on Multimedia*, pp. 6888–6897, 2025.

Table 6: Source assets used in Mat-Edit. Asset names link to their original Blendkit pages.

Asset	Creator	License
Hotel Cleaning Cart	Free poly	CC0
Janitor Cart With Bag	sotilrac Asmat	Royalty Free
Marketcar Tuning	Vic Matthew Noda	Royalty Free
Golf Cart	Mike Radjabov	Royalty Free
Simple Old Horse-Drawn Carriage	Joachim Bornemann	Royalty Free

APPENDIX

A DETAIL OF HEMISPHERE SAMPLING USING A VOGEL DISK PATTERN

We use the same Vogel disk sampling scheme to generate hemisphere directions for primitive-wise directional visibility prediction and Screen Space Ambient Occlusion (SSAO) computation. Given a spatial point $\boldsymbol{x}$ and its normal $\boldsymbol{n}$, we construct a local orthonormal frame $(\boldsymbol{t}, \boldsymbol{b}, \boldsymbol{n})$, where $\boldsymbol{t}$ and $\boldsymbol{b}$ span the tangent plane. For primitive-wise visibility, $\boldsymbol{x} = \boldsymbol{\mu}_i$ and $\boldsymbol{n} = \boldsymbol{n}_i$ are the center and normal of primitive $\mathcal{G}_i$, respectively. For SSAO, $\boldsymbol{x} = \boldsymbol{\mathcal{P}}(\boldsymbol{p})$ and $\boldsymbol{n} = \boldsymbol{\mathcal{N}}(\boldsymbol{p})$ are the position and normal map values at pixel $\boldsymbol{p}$, respectively.

To generate L directions, we first distribute L samples over the unit disk using a Vogel disk pattern:

$$r_j = \sqrt{\frac{j - \frac{1}{2}}{L}}, \qquad \phi_j = \varphi_g(j-1), \qquad \varphi_g = \pi\left(3 - \sqrt{5}\right), \quad j = 1, \ldots, L, \tag{15}$$

where r_j, ϕ_j, and φ_g denote the radial coordinate, azimuth angle, and golden angle, respectively. The number of directions L is set to K for primitive-wise visibility and M for SSAO. Each disk sample is then mapped to the upper unit hemisphere to obtain a local direction $\hat{\boldsymbol{\omega}}^j$, defined as:

$$\hat{\boldsymbol{\omega}}^j = \left(r_j \cos\phi_j,\, r_j \sin\phi_j,\, \sqrt{1 - r_j^2}\right). \tag{16}$$

This mapping produces a cosine-weighted distribution of directions over the hemisphere. The corresponding direction $\boldsymbol{\omega}^j$ at $\boldsymbol{x}$ is obtained by transforming the local direction into the coordinate system of $\boldsymbol{x}$ and $\boldsymbol{n}$:

$$\boldsymbol{\omega}^j = \hat{\omega}_x^j \boldsymbol{t} + \hat{\omega}_y^j \boldsymbol{b} + \hat{\omega}_z^j \boldsymbol{n}, \tag{17}$$

where $\hat{\omega}_x^j$, $\hat{\omega}_y^j$, and $\hat{\omega}_z^j$ denote the components of $\hat{\boldsymbol{\omega}}^j$ in the local frame.

B DETAIL OF THE MAT-EDIT SYNTHETIC BENCHMARK

In this paper, we introduce the Mat-Edit synthetic benchmark for comprehensively evaluating inverse rendering performance. The dataset contains 6 scenes, 5 of which are built from assets obtained from Blendkit. We use Blender 5.0.0 to render each scene from multiple viewpoints and save the corresponding camera parameters.

Each scene includes 100 training views and 200 test views, with each view providing camera parameters, a rendered image, albedo and AO maps, material editing and relighting results, and a pixel-level material label map. For each material label, we also provide two sets of parameters for the corresponding material, specifying its configurations before and after editing. Figure 6 provides an overview of Mat-Edit, including all collected scenes and the corresponding components of a representative frame from each scene. The 5 Blendkit assets used in Mat-Edit are listed in Table 6, together with their creators and licenses.

C DETAIL OF MATERIAL EDITING

To enable fine-grained material editing, we leverage FlashSplat to assign material labels to Gaussian primitives using the multi-view material label maps provided in our Mat-Edit benchmark. For each

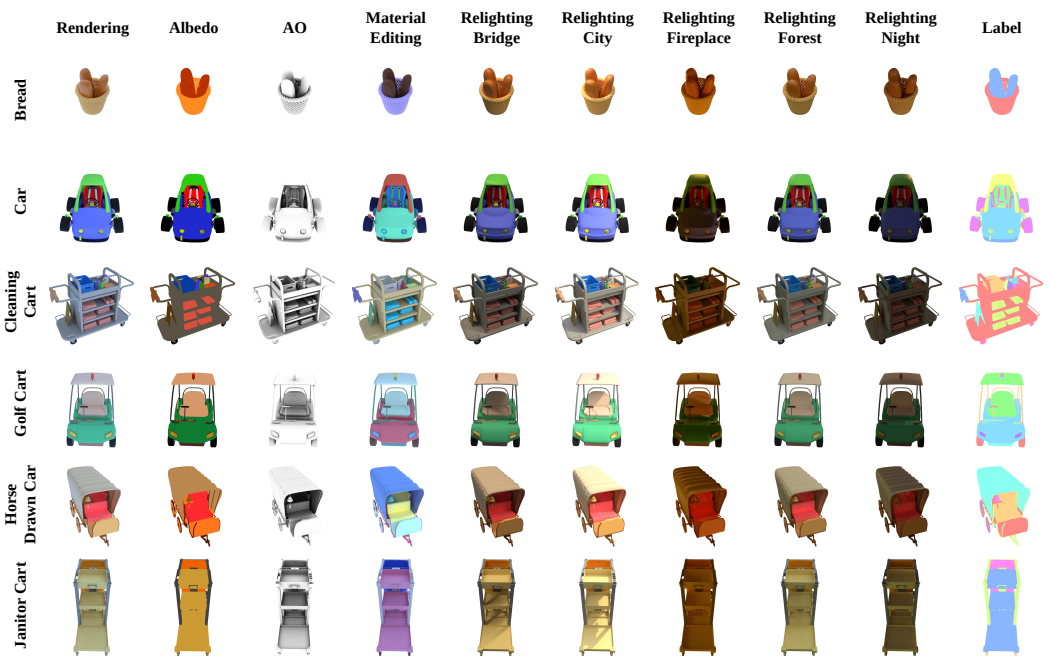

Figure 6: Overview of the Mat-Edit synthetic benchmark.

primitive $\mathcal{G}_i$, the score S_i^c for material class c is computed by accumulating its rendering weights over pixels with the corresponding label:

$$S_i^c = \sum_v \sum_{\boldsymbol{p}} w_i^v(\boldsymbol{p})\, \mathbb{I}(L_v(\boldsymbol{p}) = c)\,, \tag{18}$$

where $L_v(\boldsymbol{p})$ denotes the material label at pixel $\boldsymbol{p}$ in view v, $w_i^v(\boldsymbol{p})$ is the rendering weight, and $\mathbb{I}(\cdot)$ is the indicator function. We then assign each Gaussian the material label ℓ_i with the highest score:

$$\ell_i = \arg\max_c S_i^c. \tag{19}$$

The assigned label links each primitive to the corresponding material record in our dataset, from which the target parameters are retrieved for editing. By replacing the learned material parameters of the selected Gaussians with these target values, the edited scene can be rendered using the same model without retraining.

D Detail of the Computation of Weighted Mean Squared Error

The severe imbalance between unoccluded and occluded pixels in AO maps may cause evaluations to be dominated by unoccluded regions, reducing their sensitivity to occlusion estimation errors. We therefore report a weighted mean squared error (WMSE) alongside PSNR, using the mean ground truth AO value $\bar{a}$ to mitigate the influence of this imbalance on evaluation. The WMSE value at pixel $\boldsymbol{p}$ is defined as:

$$w_{\boldsymbol{p}} = \bar{a}(1 - a_{\boldsymbol{p}}) + (1 - \bar{a})a_{\boldsymbol{p}}, \qquad \text{WMSE} = \frac{\sum_{\boldsymbol{p}} w_{\boldsymbol{p}} \left(a_{\boldsymbol{p}} - \hat{a}_{\boldsymbol{p}}\right)^2}{\sum_{\boldsymbol{p}} w_{\boldsymbol{p}}}, \tag{20}$$

where $\boldsymbol{p}$ indexes pixels in the image, $a_{\boldsymbol{p}}$ and $\hat{a}_{\boldsymbol{p}}$ are the ground-truth and predicted AO values, respectively, and $w_{\boldsymbol{p}}$ is the pixel weight.

E Additional Analysis on Training Efficiency

In this section, we further analyze the training efficiency of ELM-GS. Tables 7, 8, and 9 report per-scene training time of different optimization stages on synthetic and real-world datasets. Following

Table 7: Quantitative comparison on training efficiency on the TensoIR dataset. The best, second-best, and third-best results are highlighted.

Stage	Method	Armadillo	Ficus	Hotdog	Lego	Average
Stage 1	GS-IR	1459	1440	1529	1507	1484
	RadioGS	2260	2087	2371	3029	2437
Stage 2	GS-IR	86	115	167	189	139
	Ours	179	180	181	184	181
Stage 3	GS-IR	573	614	565	546	575
	GI-GS	726	686	611	710	683
	RadioGS	2544	2920	2961	3327	2938
	Ours	301	315	325	322	316
Decomposition	GS-IR	659	729	732	735	714
	GI-GS	726	686	611	710	683
	RadioGS	2544	2920	2961	3327	2938
	Ours	480	495	506	506	497
Total	GS-IR	2118	2169	2261	2242	2198
	GI-GS	2185	2126	2140	2217	2167
	RadioGS	4804	5007	5332	6356	5375
	Ours	1939	1935	2035	2013	1981

the optimization procedure, we divide the training process into three stages, where stage 1 initializes scene geometry, stage 2 constructs occlusion priors, and stage 3 optimizes PBR parameters. Since stage 1 is shared across different AO-based methods, we only report the corresponding time of GS-IR and RadioGS. Moreover, stage 2 is only present in methods that precompute occlusion information, including GS-IR and ELM-GS. We also report the combined time for stages 2 and 3 as "Decomposition", alongside the total training time, to provide an overview of the training cost.

For occlusion prior construction in stage 2, the main computational cost lies in visibility estimation. GS-IR constructs occlusion volumes by sequentially baking visibility at individual grid points through repeated rendering, while ELM-GS directly generates multi-view occlusion priors using SSAO with parallel sampling across pixels. As shown in Tables 7, 8, and 9, ELM-GS consistently requires less time for occlusion prior generation, with a more pronounced advantage in large-scale scenes.

ELM-GS further improves training efficiency in stage 3 through its accelerated rendering pipeline. Specifically, the modulation terms required for deferred shading are obtained in a single splatting pass, reducing the rendering cost at each iteration and thus the time required for PBR parameter optimization.

Together, efficient prior construction in stage 2 and accelerated rendering in stage 3 reduce the combined cost of decomposition. Although ELM-GS requires an additional prior construction stage similar to GS-IR, efficient SSAO-based prior generation keeps this overhead low, while faster rendering in stage 3 further reduces the total decomposition time.

F Additional Qualitative Comparisons

In this section, we provide additional qualitative comparisons to demonstrate the superior inverse rendering performance of our method among AO-based 3DGS approaches. Since all methods share the same 3DGS model obtained from the initial stage, they exhibit similar geometric structures, with the main differences reflected in the rendered pixel colors. To facilitate a more intuitive comparison, we additionally visualize normalized error maps for results with available ground truth. Specifically, the pixel-wise error is computed as the mean squared error between the predicted and ground truth RGB values across the three color channels.

Novel View Synthesis. The gains of ELM-GS in novel view synthesis mainly stem from the proposed environment compensation terms. As shown in Figure 7 and 8, our method achieves more ac-

Table 8: Quantitative comparison on training efficiency on the Mat-Edit dataset.

Stage	Method	Bread	Car	Cleaning Cart	Golf Cart	Horse Drawn Car	Janitor Cart	Average
Stage 1	GS-IR	665	783	731	808	785	1301	846
	RadioGS	1984	2424	2711	2594	2515	2809	2506
Stage 2	GS-IR	306	323	251	268	235	197	263
	Ours	203	183	181	173	171	173	181
Stage 3	GS-IR	396	391	441	429	415	410	414
	GI-GS	826	693	763	768	707	684	740
	RadioGS	2458	4163	2786	3108	2561	3763	3140
	Ours	322	320	322	310	302	311	315
Decomposition	GS-IR	702	714	692	697	650	607	677
	GI-GS	826	693	763	768	707	684	740
	RadioGS	2458	4163	2786	3108	2561	3763	3140
	Ours	525	503	503	483	473	484	495
Total	GS-IR	1367	1497	1423	1505	1435	1908	1523
	GI-GS	1491	1476	1494	1576	1492	1985	1586
	RadioGS	4442	6587	5497	5702	5076	6572	5646
	Ours	1190	1286	1234	1291	1258	1785	1341

Table 9: Quantitative comparison on training efficiency on the real-world datasets.

Stage	Method	Bicycle	Bonsai	Counter	Kitchen	Room	Stump	Garden	Flowers	Treehill	Train	Truck
Stage 1	GS-IR	2182	1742	1880	2063	1922	2143	2057	1602	1685	1064	1224
Stage 2	GS-IR	4733	3001	937	1843	922	3558	2349	3097	3888	1240	2268
	Ours	85	96	96	108	115	58	74	79	47	93	94
Stage 3	GS-IR	1575	1631	1467	1524	1681	1380	1434	1246	1297	800	892
	GI-GS	2262	2199	2344	2480	2322	1922	2000	2002	1949	1220	1298
	Ours	1302	955	1008	1066	1042	1123	1179	1035	1063	675	790
Decomposition	GS-IR	6308	4632	2404	3367	2603	4938	3783	4343	5185	2040	3160
	GI-GS	2262	2199	2344	2480	2322	1922	2000	2002	1949	1220	1298
	Ours	1387	1051	1104	1174	1157	1181	1253	1114	1110	768	884
Total	GS-IR	8490	6374	4284	5430	4525	7081	5840	5945	6870	3104	4384
	GI-GS	4444	3941	4224	4543	4244	4065	4057	3604	3634	2284	2522
	Ours	3569	2793	2984	3237	3079	3324	3310	2716	2795	1832	2108

curate rendering around object boundaries, where local illumination variations are more pronounced, indicating its ability to mitigate the limitations of simplified AO-based illumination modeling. It also effectively reduces geometry- and illumination-induced deviations on smooth surfaces. Figure 9 further presents qualitative results on real-world scenes, demonstrating the robustness of our method across varying scene scales.

Albedo Estimation. ELM-GS achieves substantial improvements in albedo estimation by more effectively disentangling albedo from shading effects. As shown in Figure 10 and Figure 11, our method produces albedo maps with substantially reduced residual shading and occlusion compared with the baselines. For example, in the lego scene (the fourth row of Figure 10), ELM-GS estimates similar albedo values for the occluded lower surface and the unoccluded regions of the same material. In the bread scene (the first row of Figure 11), our method effectively removes the shading inside the cavities. These improvements further suggest the potential of our method for downstream tasks that rely on accurate intrinsic decomposition.

Ambient Occlusion Estimation. The multi-view occlusion priors efficiently obtained through SSAO enable ELM-GS to produce more physically plausible ambient occlusion. As shown in Figure 12, our method recovers finer occlusion details that are largely overlooked by the baselines. Figure 13 further compares the predicted AO maps with the ground truth on the Mat-Edit dataset,

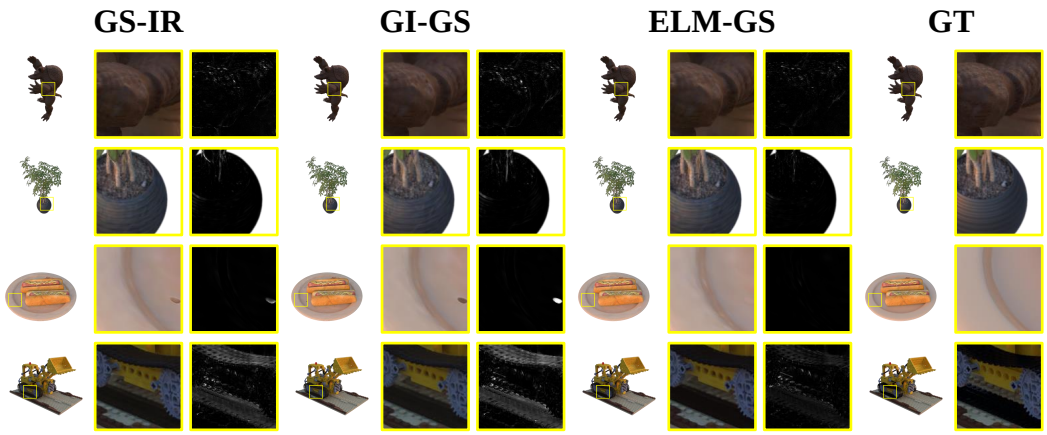

Figure 7: Qualitative comparison of novel view synthesis results on the TensoIR synthetic dataset.

where ELM-GS achieves substantially improved AO estimation quality. On real-world scenes, our method consistently preserves richer local occlusion cues, as illustrated in Figure 9. For example, the occlusion between grass blades and the surface patterns on rubber gloves are clearly captured in the predicted AO maps. These results demonstrate the superior AO estimation capability of ELM-GS alongside its significantly improved efficiency, enhancing depth perception in rendered scenes and benefiting downstream tasks.

Material Editing. Unlike AO-based baselines, which tend to lose substantial occlusion cues after material replacement and consequently weaken the sense of depth in the rendered scene, ELM-GS achieves better albedo—shading disentanglement and preserves geometric details even after replacing the learned material. As shown in Figure 14, our method more clearly distinguishes seams between objects with similar materials and shows evident advantages in heavily occluded regions. In addition, the overall appearance of the rendered results is closer to the ground truth, further indicating a reduced coupling between material properties and environmental illumination.

Relighting. Similar to material editing, the improved AO estimation also benefits the relighting performance of ELM-GS. As shown in Figure 15 and Figure 16, our method preserves clearer geometric structures under novel illumination conditions, resulting in an improved sense of depth in the rendered scenes. Together with the material editing results, these downstream evaluations further demonstrate the effectiveness of ELM-GS in disentangling material, illumination, and occlusion, highlighting its overall capability for inverse rendering.

G Visualization of Ablation Studies

Analysis on Learnable Ambient Occlusion. As shown in Figure 17, the "SSAO Only" and "w/o LAO" variants achieve favorable intrinsic decomposition and scene editing quality by directly using SSAO-derived occlusion priors for rendering, although introducing a higher computational cost. These results support the effectiveness of the priors as explicit supervision for AO estimation. Building on this guidance, learnable AO enables the full model to avoid explicit SSAO computation while rendering AO of comparable quality. Moreover, it promotes multi-view consistency and adaptively compensates for occlusion effects not captured by the priors through image supervision, improving intrinsic decomposition and subsequent material editing. When SSAO supervision is removed entirely, as in the "w/o SSAO" variant, AO estimation becomes less physically plausible, substantially degrading scene editing quality.

However, jointly learning AO and EC can introduce spot-like artifacts into EC and the estimated albedo. These artifacts become apparent during relighting, when EC is omitted under the novel illumination. Despite this limitation, learnable AO substantially accelerates rendering while im-

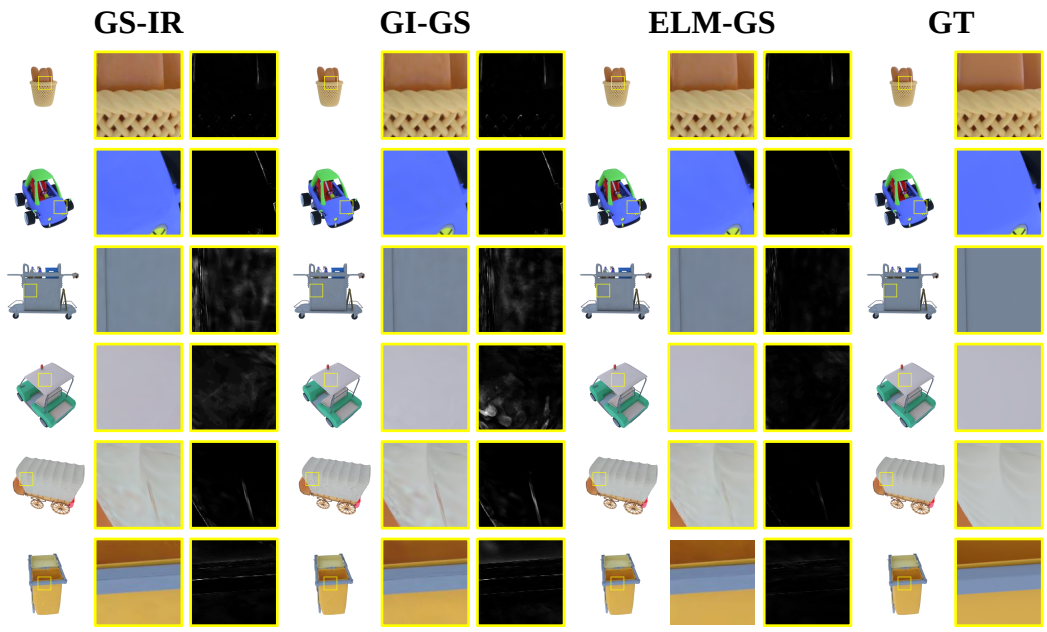

Figure 8: Qualitative comparison of novel view synthesis results on the Mat-Edit synthetic dataset.

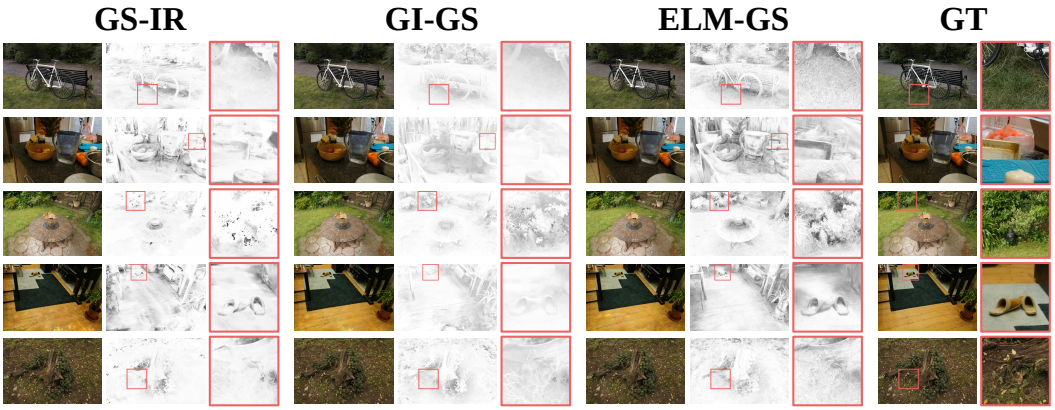

Figure 9: Qualitative comparison of novel view synthesis and ambient occlusion estimation results on the Mip-NeRF 360 real-world dataset.

proving the intrinsic decomposition and material editing performance, demonstrating its practical effectiveness.

Analysis on Learnable Environment Compensation. Environment compensation improves intrinsic decomposition under simplified AO-based illumination modeling. As shown in Figure 17, removing this term during both training and evaluation in the "w/o LEC" variant leads to noticeable color discrepancies in the estimated albedo across regions of the same material subject to different local illumination conditions. We visualize the albedo in sRGB space to make these discrepancies more apparent. After replacing the estimated material, the "w/o LEC" variant loses occlusion cues within the car interior, resulting in a flatter appearance. These results demonstrate that EC provides a simple yet effective solution for improving both intrinsic decomposition and scene editing in AO-based inverse rendering methods.

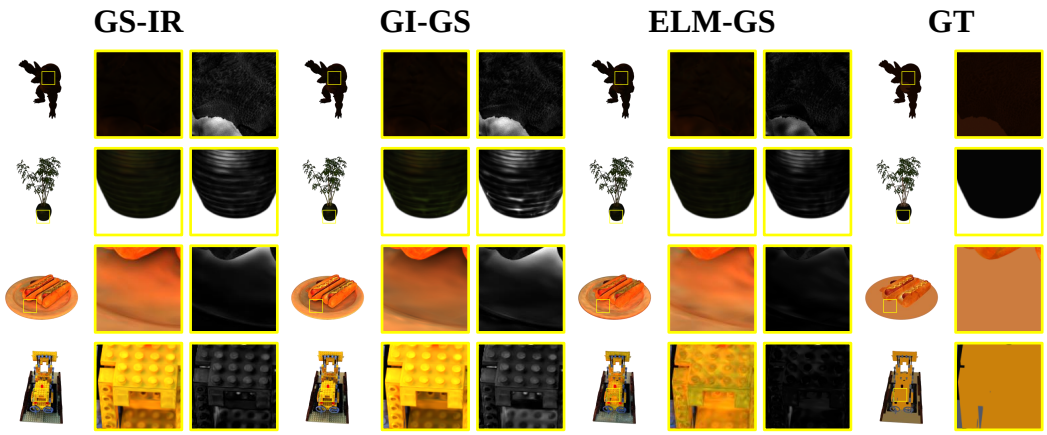

Figure 10: Qualitative comparison of albedo estimation results on the TensoIR synthetic dataset.

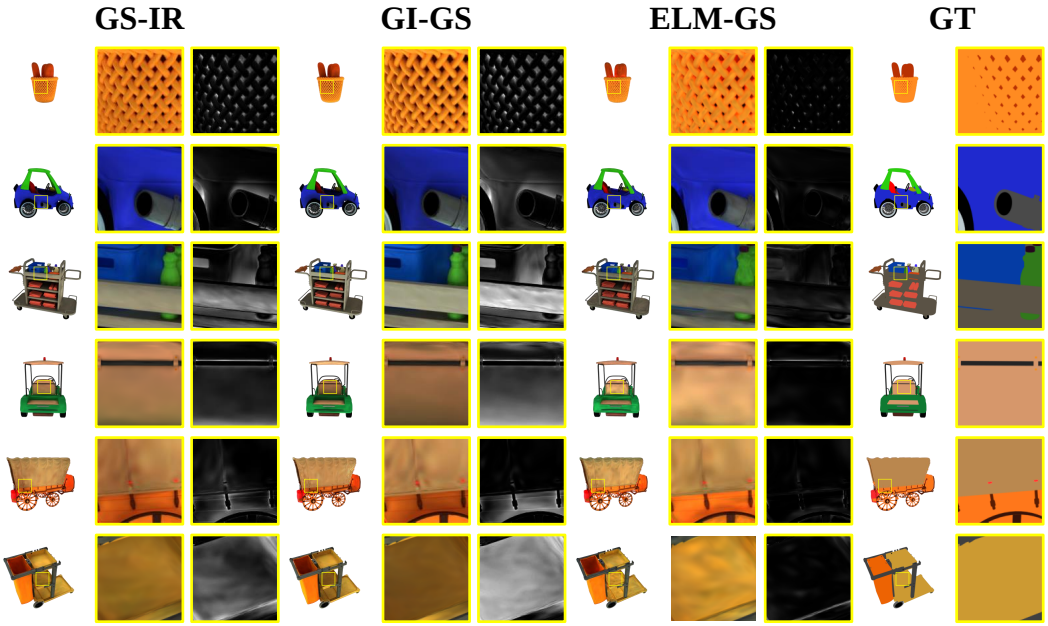

Figure 11: Qualitative comparison of albedo estimation results on the Mat-Edit synthetic dataset.

H MORE RESULTS

In this section, we further present diverse scene editing results on the Mat-Edit benchmark. For each scene, we evaluate five materials under five environment maps, including one edited material from our dataset and four randomly generated ones. Figures 18, 19, 20, 21, and 22 show the corresponding editing results of ELM-GS. These results demonstrate that the proposed benchmark supports a broader range of material and illumination variations, enabling more diverse scene editing configurations. Meanwhile, ELM-GS consistently preserves local occlusion cues and geometric structures after editing, maintaining a clear sense of depth across different materials and lighting conditions.

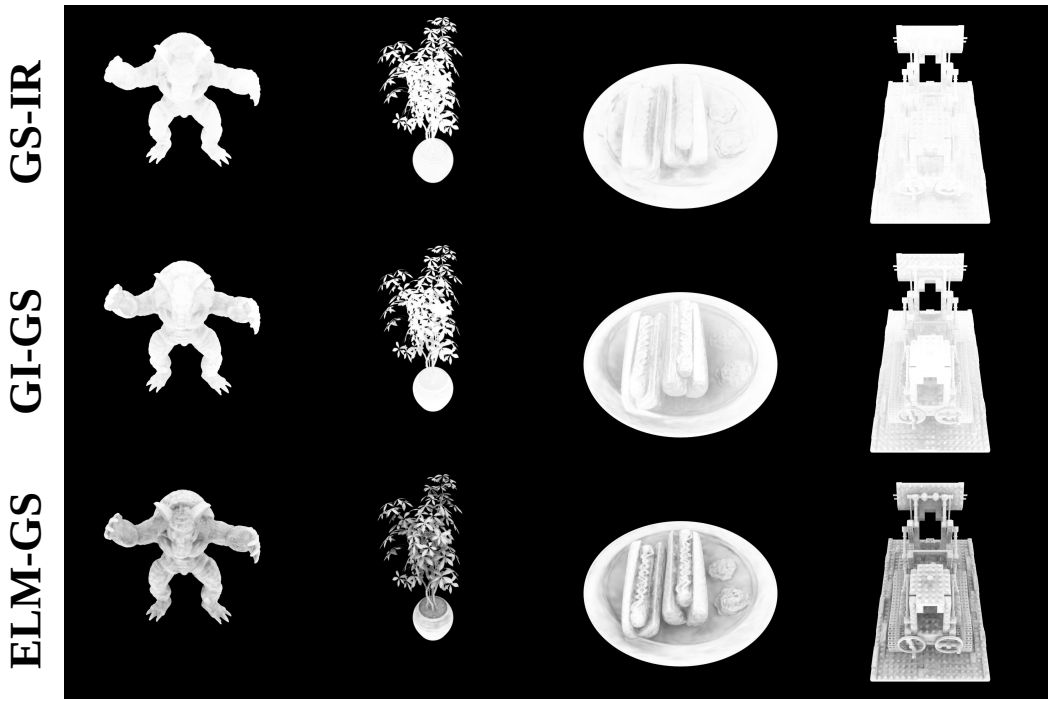

Figure 12: Qualitative comparison of ambient occlusion estimation results on the TensoIR synthetic dataset.

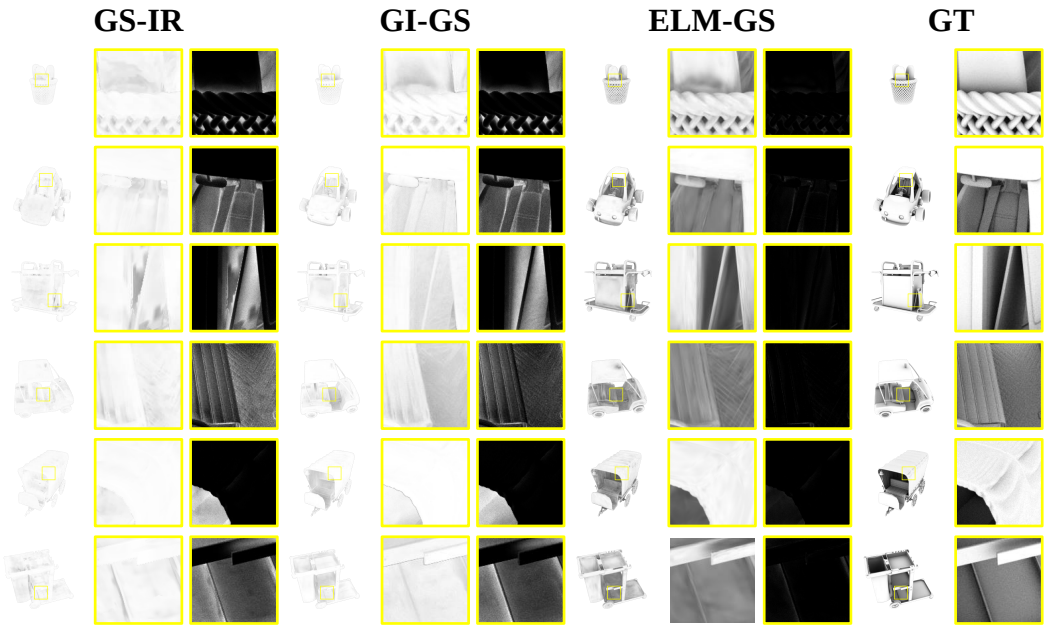

Figure 13: Qualitative comparison of ambient occlusion estimation results on the Mat-Edit synthetic dataset.

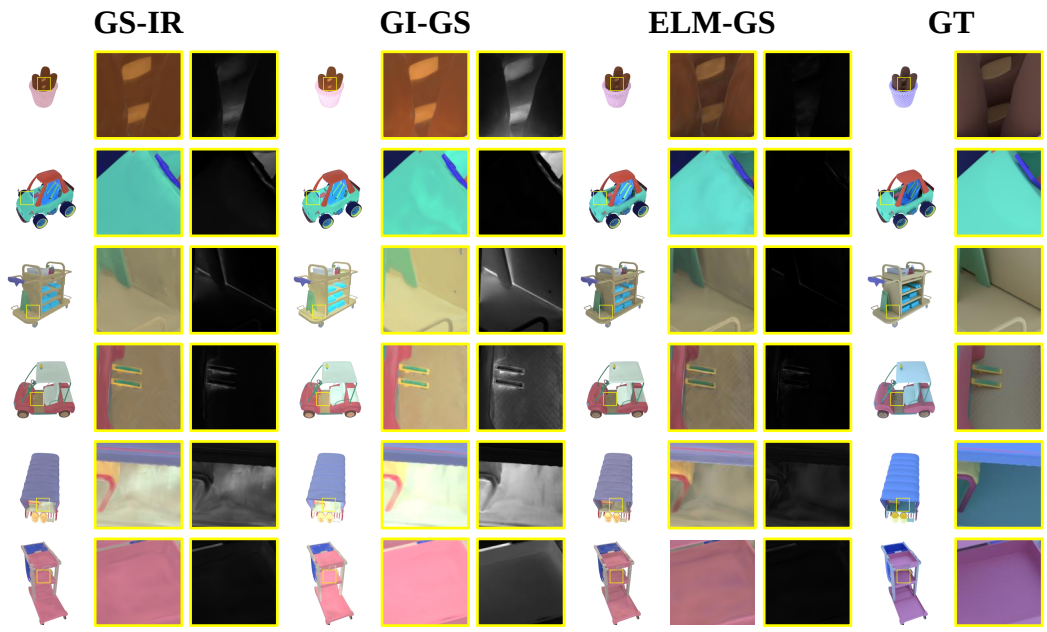

Figure 14: Qualitative comparison of material editing results on the Mat-Edit synthetic dataset.

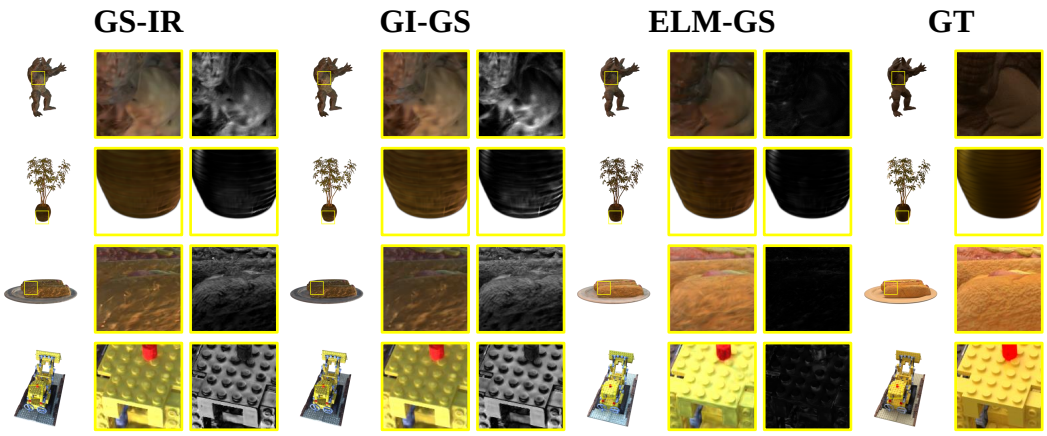

Figure 15: Qualitative comparison of relighting results on the TensoIR synthetic dataset.

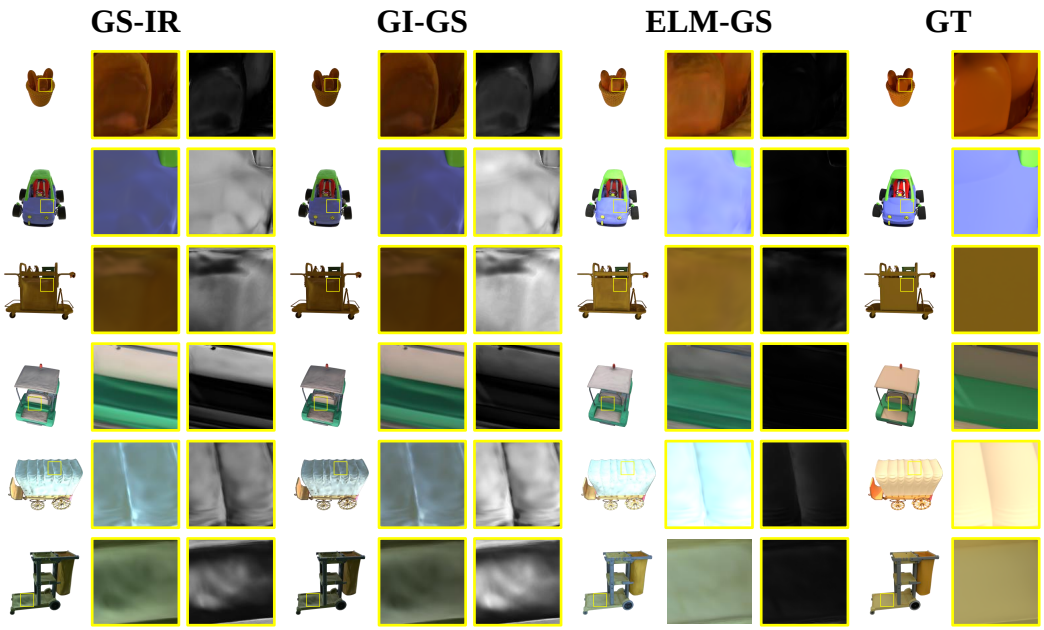

Figure 16: Qualitative comparison of relighting results on the Mat-Edit synthetic dataset.

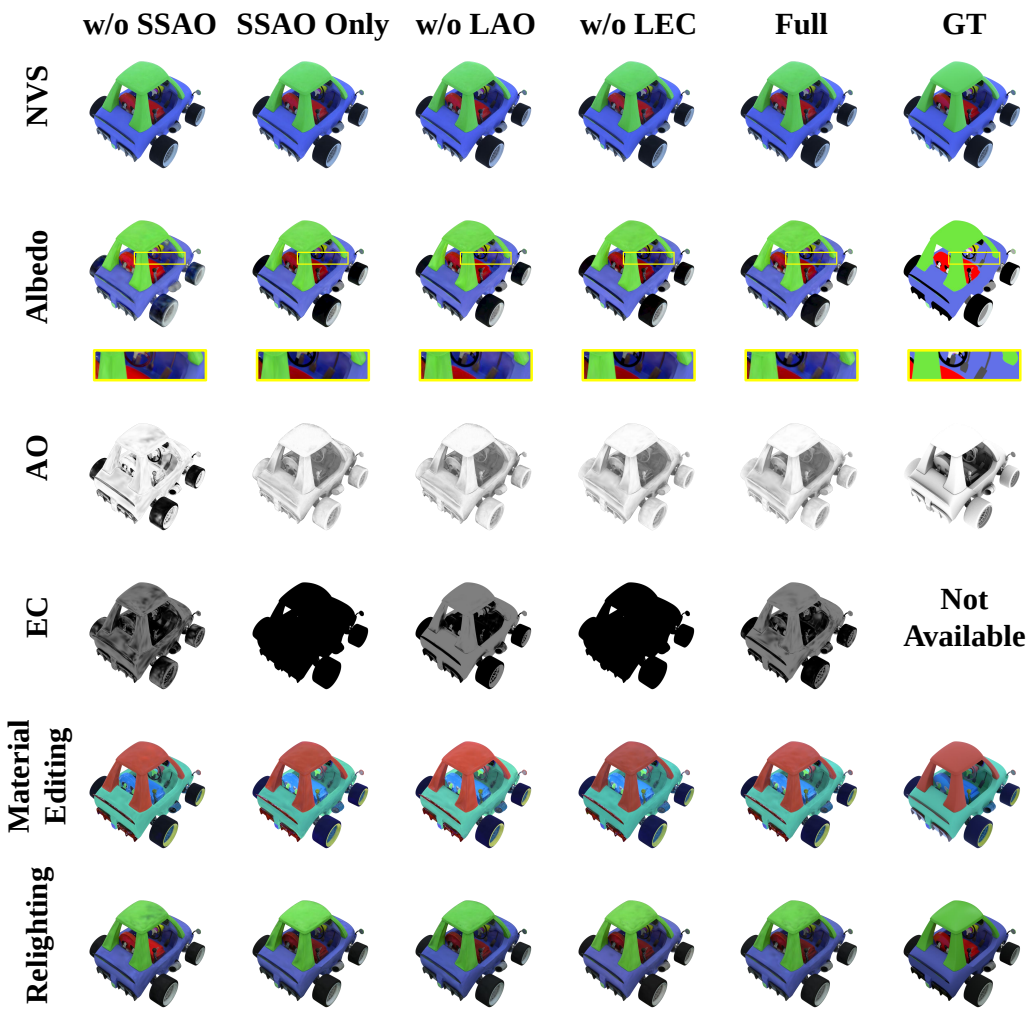

Figure 17: Visualization of ablation studies.

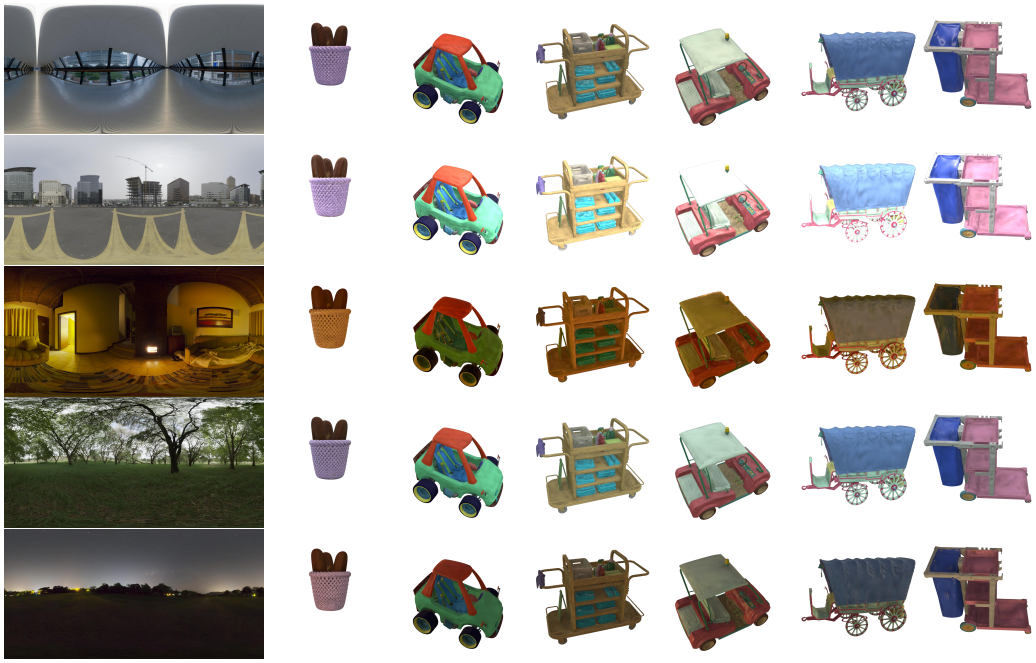

Figure 18: Material editing results under different environment maps of our ELM-GS. The materials in each scene are set to the edited configurations provided in the Mat-Edit benchmark.

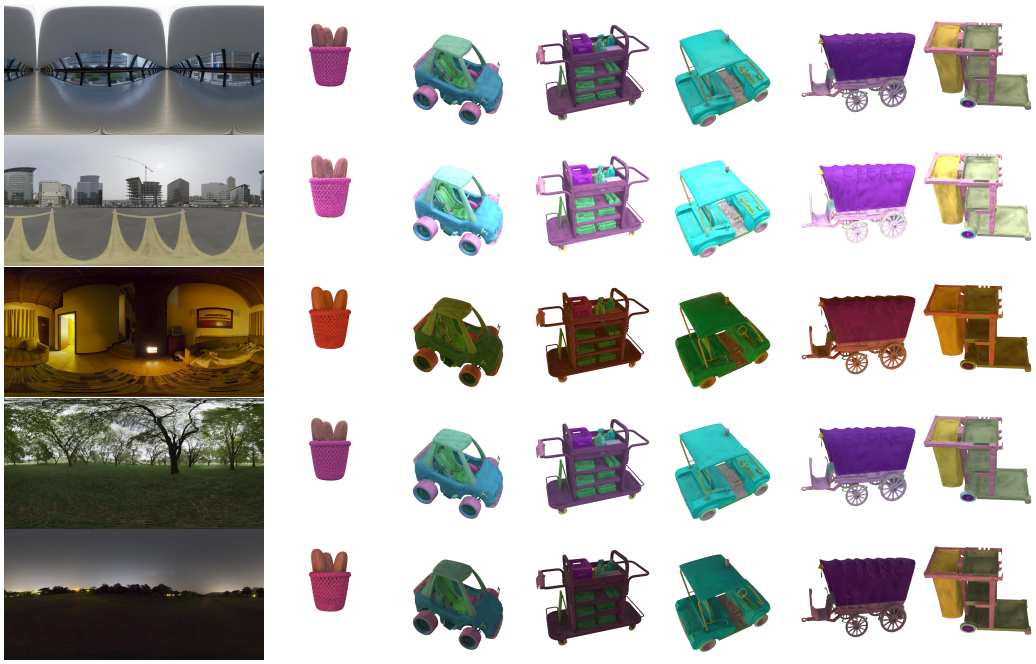

Figure 19: Material editing results under different environment maps of our ELM-GS. The materials in each scene are randomly generated.

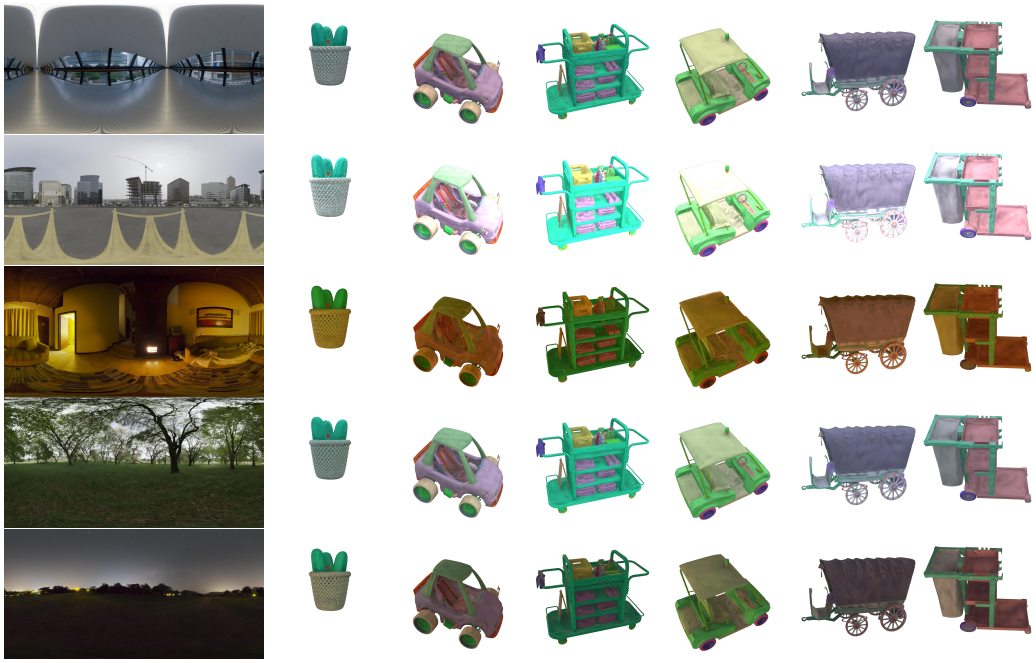

Figure 20: Material editing results under different environment maps of our ELM-GS. The materials in each scene are randomly generated.

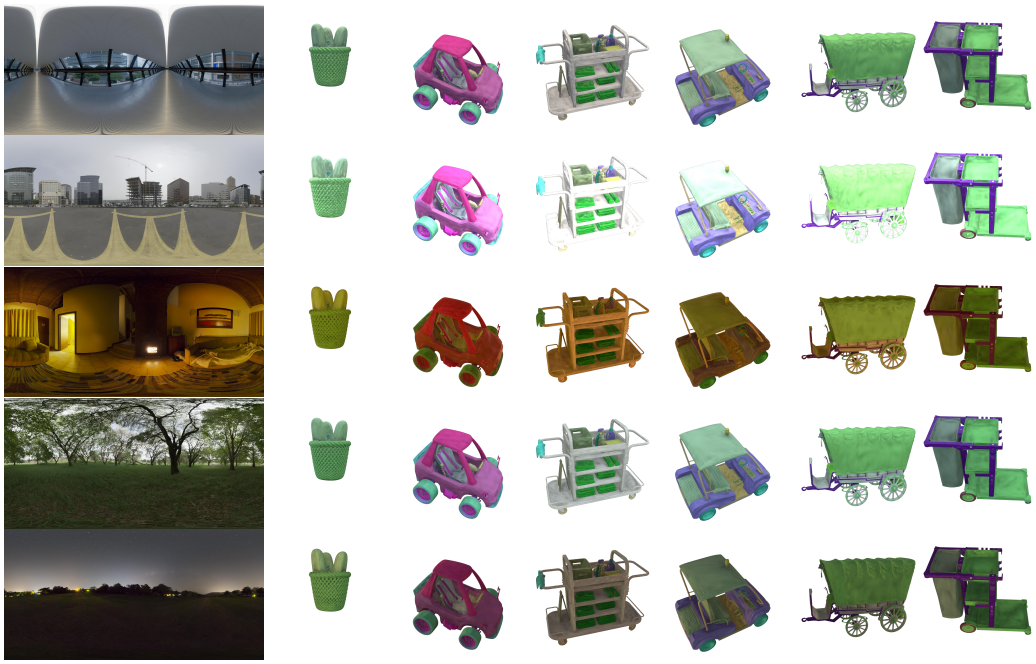

Figure 21: Material editing results under different environment maps of our ELM-GS. The materials in each scene are randomly generated.

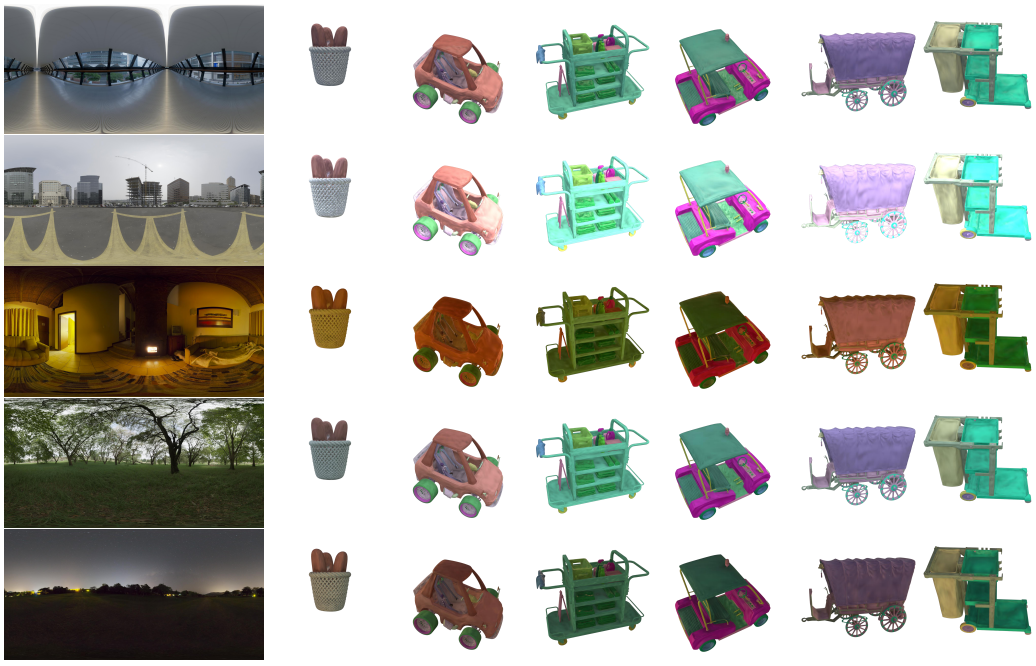

Figure 22: Material editing results under different environment maps of our ELM-GS. The materials in each scene are randomly generated.