GS-ID: Illumination Decomposition on Gaussian Splatting via Adaptive Light Aggregation and Diffusion-Guided Material Priors

Kang Du[1] Zhihao Liang[2] Yulin Shen[1] Zeyu Wang[1,3]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]South China University of Technology
[3]The Hong Kong University of Science and Technology

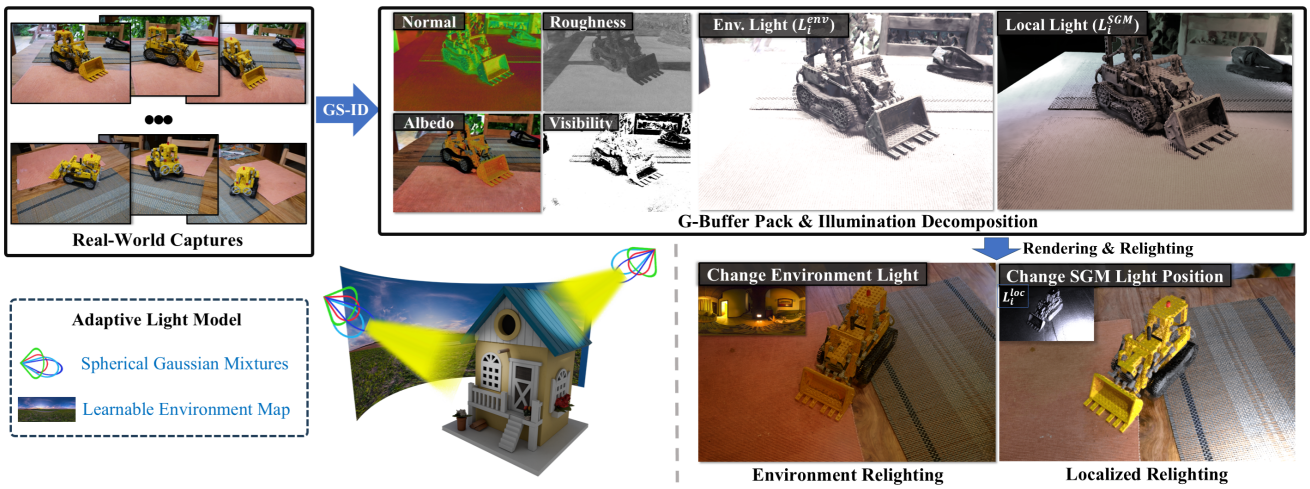

Figure 1. GS-ID pipeline and relighting results. GS-ID introduces a novel lighting model based on adaptively optimized spherical Gaussian mixtures, enabling precise and editable control of local lighting. Combined with per-splat shadow-aware vectors and diffusion-guided material priors, GS-ID achieves state-of-the-art illumination decomposition on 3D Gaussian Splatting.

Abstract

Gaussian Splatting (GS) has emerged as an effective representation for photorealistic rendering, but the underlying geometry, material, and lighting remain entangled, hindering scene editing. Existing GS-based methods struggle to disentangle these components under non-Lambertian conditions, especially in the presence of specularities and shadows. We propose ***GS-ID****, an end-to-end framework for illumination decomposition that integrates adaptive light aggregation with diffusion-based material priors. In addition to a learnable environment map for ambient illumination, we model spatially-varying local lighting using anisotropic spherical Gaussian mixtures (SGMs) that are jointly optimized with scene content. To better capture cast shadows, we associate each splat with a learnable unit vector that encodes shadow directions from multiple light sources, further improving material and lighting estimation. By combining SGMs with intrinsic priors from diffusion models, GS-ID significantly reduces ambiguity in light-material-geometry interactions and achieves state-of-the-art performance on inverse rendering and relighting benchmarks. Experiments also demonstrate the effectiveness of GS-ID for downstream applications such as relighting and scene composition.*

1. Introduction

3D Gaussian Splatting (3DGS) [14] has emerged as a promising 3D representation, offering explicit scene modeling with real-time differentiable rendering capabilities. However, its practical adoption by downstream applications faces a critical limitation: the inherent entanglement of geometry, material, and illumination components during multiview reconstruction, which prevents further editing

of individual components. Illumination decomposition on 3DGS has significant value as it can facilitate versatile GS editing using disentangled components, including changing the light and material for scene composition.

There are three critical challenges in achieving effective illumination decomposition on 3DGS. 1) Insufficient light modeling: Existing methods struggle to represent complex lighting environments, often failing to balance global ambient lighting with localized high-frequency effects. 2) Entangled shadows: Shadows arising from intricate light-geometry interactions obscure material estimation and are difficult to isolate. 3) Ill-posed intrinsic estimation: Without priors, decomposing the coupled outputs of 3DGS into geometry, material, and lighting remains highly ambiguous.

Recent methods [4, 5, 10, 11, 16, 29, 35] model illumination using global lighting representations, such as learnable environment maps or neural light fields. However, they typically assume distant light sources, neglecting near-field effects that are critical for capturing localized specular highlights and accurate material properties. Emerging solutions like VMINer [7] and GS^3 [3] incorporate photometric prior knowledge (e.g., predefined light sources, intensity distributions, and angular profiles) into near-field illumination modeling. However, strong parametric assumptions about light configurations restrict their practical applicability, making them dependent on synthetic scenes with calibrated lighting setups. Shadow modeling is another bottleneck. MII [37] employs a learned visibility MLP, but struggles with multiple discrete light sources. Prior-free decomposition under such conditions remains ill-posed, especially in in-the-wild captures. To resolve geometry-light ambiguity, DN-Splatter [25] uses pre-trained models to provide pseudo-normal supervision, but it relies on additional sensors (e.g., LiDAR), limiting its applicability.

To address these challenges, we propose GS-ID, a novel end-to-end framework for illumination decomposition on 3DGS. GS-ID integrates an adaptive lighting model, a deshadowing module, and pretrained diffusion-based priors for geometry and material. A customized CUDA-based optimization with deferred rendering further accelerates the decomposition process. Inspired by production lighting pipelines in tools like Unity [26] and Unreal Engine [6], which combine ambient lighting with localized sources, we design a lighting model that explicitly separates ambient and localized illumination. GS-ID represents ambient light via a learnable environment map and models high-frequency localized lighting using spatially-varying spherical Gaussian mixtures (SGMs). These SGMs are initialized on a 3D grid and adaptively aggregated during optimization to capture complex lighting effects (Section 3.1). To handle shadow-induced errors in material estimation, we introduce a deshadowing module that learns per-splat visibility vectors, enabling the network to disentangle shadows caused by multiple unknown light sources (Section 3.2). Finally, to resolve ambiguity in joint light-material optimization, we introduce pretrained diffusion priors. For geometry, we incorporate normal priors [34] to stabilize reconstruction. For material, we guide decomposition using albedo and roughness maps from a pretrained diffusion model (Section 3.3.1).

This paper makes the following contributions:

- We introduce GS-ID, an end-to-end framework for illumination decomposition, which leverages material priors from pretrained diffusion models to improve joint light and material optimization on 3DGS.
- We propose a novel lighting model supporting adaptive optimization of local and ambient illumination under unknown conditions with a CUDA implementation.
- We develop a GS-based visibility-aware deshadowing model for efficient shadow approximation caused by multiple light sources, improving material estimation quality.

2. Related Work

Geometry Reconstruction. Regarding surface reconstruction, some methods [20, 21, 23, 28, 31, 32] use an MLP to model an implicit field representing the target surface. After training, surfaces are extracted using isosurface extraction algorithms [12, 17]. Recently, several methods [9, 33] have achieved fast and high-quality geometry reconstruction of complex scenes based on 3DGS [14]. However, these methods often treat the appearance of a functional of view directions, neglecting the modeling of physical materials and light transport on the surface.

Intrinsic Decomposition. To decompose the intrinsics from observations, some monocular methods [15, 34, 39, 40] learn from labeled datasets and estimate intrinsics directly from single images. However, these methods lack multi-view consistency and struggle to tackle out-of-distribution cases. In contrast, other methods [2, 10, 11, 16, 18, 24, 36, 38] construct a 3D consistent intrinsic field from multiple observations. While obtaining impressive results, current methods almost only consider environmental illumination, making precise editing challenging.

Lighting Models. Accurate illumination estimation from multi-view observations remains challenging due to inverse rendering ambiguities. Current methodologies primarily follow two paradigms: 1) For unknown illumination conditions, approaches utilize either implicit neural representations (e.g., MLP-based coordinate networks like NeRF variants [4, 5]) or explicit parametric models (e.g., HDR environment maps in [11, 16]). While neural light fields ([30, 35]) achieve differentiable surface reconstruction with inter-reflection modeling through MLPs, they trade interpretability for flexibility - excelling at view synthesis but lacking spatially aware editing controls. Explicit directional lighting models enable intuitive environ-

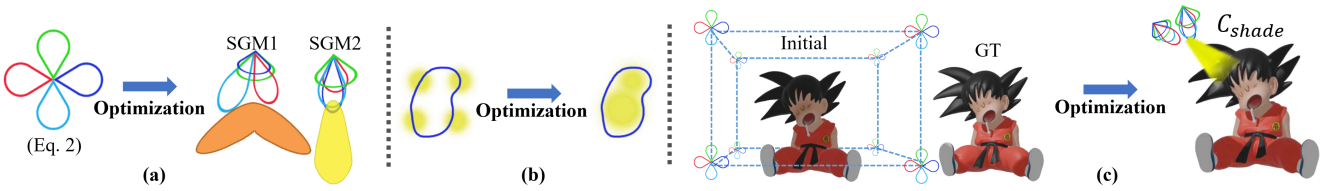

Figure 2. Illustration of the SGM optimization process. (a) A single SGM can adaptively represent different anisotropic illuminations after optimization. (b) Multiple SGMs can adaptively aggregate to represent complex and irregular highlight regions. (b) Illustration of the overall adaptive lighting optimization. The detailed optimization process is available in the supplemental material.

mental map replacement yet fail to resolve localized light interactions. 2) Under known illumination constraints (e.g., calibrated flash images), hybrid decomposition frameworks like VMINER [7] leverage controlled captures with/without auxiliary lights to separate near/far-field components within fixed viewpoints. This photometric prior integration addresses partial information scenarios but restricts dynamic scene adaptation.

3. Methodology

GS-ID uses an adaptive lighting model, a deshadowing model, and diffusion-guided normal and material priors to decompose intrinsic properties from illumination on 3DGS. We use deferred shading with the G-buffer pack for more efficient rendering. Figure 3 shows the complete GS-ID pipeline. To initialize a reasonable 3DGS scene before optimizing the lighting model, we only use normal priors in the first 20k iterations for reconstruction, and use all normal and material priors afterward for joint material and lighting optimization.

3.1. Adaptive Lighting Model

Effectively modeling complex lighting conditions in a scene is the first step toward illumination decomposition. We propose an adaptive lighting model using a set of SGMs to model high-frequency lighting components originating from spatially localized discrete emitters and a learnable environment map to model ambient illumination. The initial SGMs are uniformly placed in a fixed range of $[-3, 3]^3$. Our optimization process aggregates them to model the complex local lighting adaptively.

Using this adaptive lighting model, we formulate incident radiance as: $L_i = L_i^{\text{SGM}} + L_i^{\text{env}}$, where L_i^{SGM} accounts for high-frequency effects from discrete emitters and L_i^{env} is ambient lighting from distant sources. The rendering equation can be written as:

$$\begin{aligned} L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = &\int_\Omega L_i^{\text{env}}(\boldsymbol{x}, \boldsymbol{\omega}_i) f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)(\boldsymbol{\omega}_i \cdot \boldsymbol{n}) \mathrm{d}\boldsymbol{\omega}_i + \\ &\int_\Omega L_i^{\text{SGM}}(\boldsymbol{x}, \boldsymbol{\omega}_i) f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)(\boldsymbol{\omega}_i \cdot \boldsymbol{n}) \mathrm{d}\boldsymbol{\omega}_i \\ \approx &L_o^{\text{env}}(\boldsymbol{x}, \boldsymbol{\omega}_o) + L_o^{\text{SGM}}(\boldsymbol{x}, \boldsymbol{\omega}_o) * V, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ represents a point in the 3D space and $\boldsymbol{n}$ is the surface normal. f_r is the bidirectional reflectance distribution function (BRDF) for physically based rendering [19]. $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ refer to the incident and outgoing directions, respectively. V represents a modulating weight caused by shadow, which is modeled in detail in Section 3.2.

3.1.1. SGM-Based Local Lighting

Local light sources are often accountable for various lighting effects like highlights. We model complex local lighting using SGMs [27], where each SGM comprises n_{SG} individual SGs. The k-th SG in an SGM is parameterized by primary emission direction $\boldsymbol{b}_k \in \mathbb{S}^2$, sharpness $\lambda_k \in \mathbb{R}^+$, amplitude $\mu_k \in \mathbb{R}^+$, and mixture weights $\boldsymbol{w}_k \in \mathbb{R}^3$ controlling RGB chromaticity. An SGM contains multiple directional lobes represented by the SGs along their primary emission directions, producing a "spotlight-like" distribution (Figure 2a). Multiple SGMs can be adaptively placed in the 3D space and aggregated to model complex light sources that cause irregular highlights. As shown in Figure 2b, the SGMs can be jointly optimized because of their differentiable parameters:

$$\text{SGM}(\boldsymbol{\omega}_o; \boldsymbol{b}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{k=1}^{n_{\text{SG}}} \mu_k e^{\lambda_k (\boldsymbol{\omega}_o \cdot \boldsymbol{b}_k - 1)} \cdot \boldsymbol{w}_k. \tag{2}$$

The output radiance L_o^{SGM} at the surface point $\boldsymbol{x}$ integrates incident illumination through the Cook-Torrance BRDF f_r:

$$\begin{aligned} L_o^{\text{SGM}}(\boldsymbol{x}, \boldsymbol{\omega}_o) &= \int_\Omega f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i^{\text{SGM}}(\boldsymbol{x}, \boldsymbol{\omega}_i)(\boldsymbol{n} \cdot \boldsymbol{\omega}_i) \mathrm{d}\boldsymbol{\omega}_i \\ &\approx \sum_j^{N_{\text{light}}} \frac{f_r^{(j)} \cdot \text{SGM}(\boldsymbol{\omega}_o^{(j)})(\boldsymbol{n} \cdot \boldsymbol{\omega}_i^{(j)}) * V_j}{|\boldsymbol{p}_j - \boldsymbol{x}|^2}, \end{aligned} \tag{3}$$

where N_{light} SGMs are initialized on a 3D grid and strategically optimized. All SGs in an SGM use the same spatial position $\boldsymbol{p}_j$ while maintaining their own SG parameters $(\boldsymbol{b}_k, \lambda_k, \mu_k)$. The modulating weight caused by shadow, V_j, is estimated using a shadow-aware unit vector associated with each splat and each lighting direction, as discussed in Section 3.2 and illustrated in Figure 2. $|\boldsymbol{p}_j - \boldsymbol{x}|$ is the distance between the surface and the j-th light source, modeling illumination decay over distance. To enhance computational efficiency, we apply progressive pruning of low-energy SGMs with $|\boldsymbol{w}_k| < \tau$, as discussed in Section 4.1.

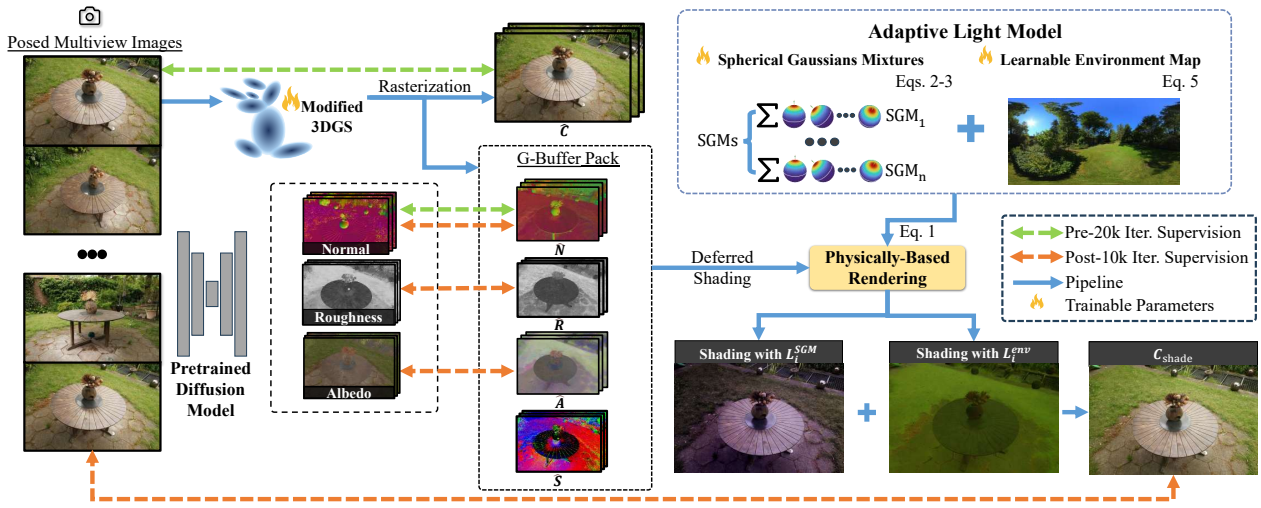

Figure 3. Overview of the GS-ID pipeline. We first reconstruct a coarse 3D Gaussian Splatting (3DGS) scene using normal priors from a diffusion model. Material priors are then incorporated to jointly optimize illumination and intrinsic properties, which are stored as G-Buffer maps for efficient deferred shading. Illumination is represented by an adaptive lighting model composed of spherical Gaussian mixtures (SGMs) and a learnable environment map.

To ensure that SGM optimization leads to an accurate model of local lighting rather than ambient lighting, we introduce two regularization terms considering the position and value of each SGM light source:

$$\mathcal{L}_{\text{pos}} = \sum_{j}^{N_{\text{light}}} \max(d_{\text{min}}^{(j)} - d_{\text{max}}, 0), \quad \mathcal{L}_{\text{val}} = \sum_{j}^{N_{\text{light}}} \sum_{k}^{n_{\text{SG}}} \|\boldsymbol{w}_{jk}\|_2$$
$$\mathcal{L}_{\text{light}} = \lambda_{\text{pos}} \mathcal{L}_{\text{pos}} + \lambda_{\text{val}} \mathcal{L}_{\text{val}}. \tag{4}$$

Here $d_{\text{min}}^{(j)} = \min_{\boldsymbol{x}} |\boldsymbol{p}_j - \boldsymbol{x}|$ represents the minimum distance between the j-th light position $\boldsymbol{p}_j$ and the surface position $\boldsymbol{x}$ in the scene. The 3D position $\boldsymbol{x}$ in world coordinates is derived by back-projecting the depth value $\hat{D}$ from the depth buffer. The parameter d_{max} is a hyperparameter with a default value of 3. The weight $\boldsymbol{w}_{jk}$ is responsible for controlling the contribution of the k-th SG in the j-th SGM. We set both λ_{pos} and λ_{val} to 1×10^{-6} by default. These two regularization terms help direct the SGMs toward the surface so that SGMs can better model complex local lighting.

Our SGM-based lighting model can represent spatially varying illumination through an adaptive mechanism. Compared to existing methods, as illustrated in Figure 1, our approach can create lighting effects like localized highlight regions and easily support illumination editing.

3.1.2. Ambient Lighting

In addition to local lighting, ambient lighting L_o^{env} can be reformulated into its diffuse ($L_{o\text{-diff}}^{\text{env}}$) and specular ($L_{o\text{-spec}}^{\text{env}}$) components. We adopt an image-based lighting model and the split-sum approximation [13] to handle the intractable integral. L_o^{env} can be represented as:

$$L_o^{\text{env}}(\boldsymbol{x}, \boldsymbol{\omega}_o) = L_{o\text{-diff}}^{\text{env}} + L_{o\text{-spec}}^{\text{env}},$$
$$L_{o\text{-diff}}^{\text{env}} \approx K_{\text{diff}}^{\text{env}} I_{\text{diff}}^{\text{env}}, \quad K_{\text{diff}}^{\text{env}} = (1 - M)\frac{\boldsymbol{A}}{\pi},$$
$$L_{o\text{-spec}}^{\text{env}} \approx \underbrace{\int_{\Omega} \frac{DFG}{4(\boldsymbol{n} \cdot \boldsymbol{\omega}_o)} \, \mathrm{d}\boldsymbol{l}}_{\text{Environment BRDF } (K_{\text{spec}}^{\text{env}})} \cdot \underbrace{\int_{\Omega} DL_i(\boldsymbol{l})(\boldsymbol{l} \cdot \boldsymbol{n}) \, \mathrm{d}\boldsymbol{l}}_{\text{Pre-Fil. Env. Map } (I_{\text{spec}}^{\text{env}})}. \tag{5}$$

$K_{\text{spec}}^{\text{env}}$ can be quickly accessed in precomputed lookup tables. $I_{\text{diff}}^{\text{env}}$ and $I_{\text{spec}}^{\text{env}}$ are embedded within a learnable environment map.

3.2. Deshadowing Model

During reconstruction, cast shadows are often baked into the albedo, leading to inaccurate material estimation. Existing methods like ray tracing or offline baking [16] are non-differentiable and computationally expensive, limiting scalability. To address this during training, we assign each 3DGS primitive a learnable unit vector $\boldsymbol{s} \in \mathbb{S}^2$ that captures the dominant shadow direction under multiple lights. These vectors are alpha-blended [14] into a screen-space shadow field and integrated into the G-buffer, enabling efficient, differentiable shadow prediction without explicit ray tracing (see Figure 4a). This deshadowing mechanism is used only during training. At inference, standard shadow mapping techniques are applied for relighting (see Figure 4b).

The visibility of incident light from direction $\boldsymbol{\omega}_i^{(j)}$ is estimated as:

$$V_j = \sigma\left(\alpha \cdot \hat{\boldsymbol{S}} \cdot \boldsymbol{\omega}_i^{(j)} + \beta\right), \tag{6}$$

where $\hat{\boldsymbol{S}}$ is a screen-space shadow vector field derived from alpha-blending the per-primitive vectors $\boldsymbol{s}$ in the G-Buffer.

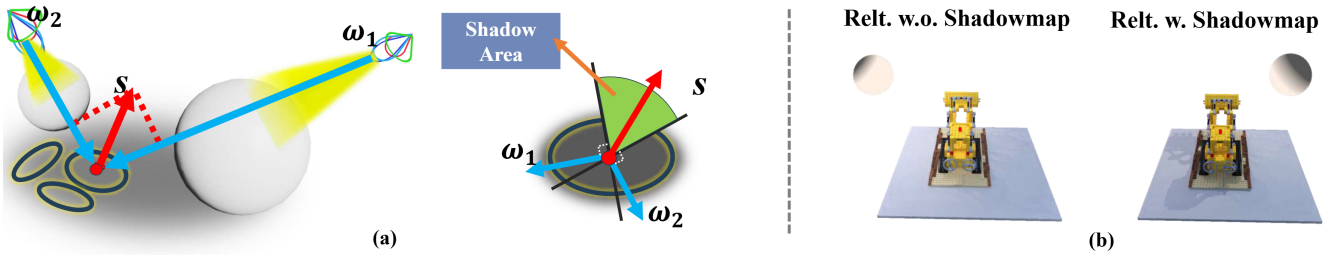

Figure 4. Illustration of our proposed deshadowing model. The lighting directions are denoted by $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$. For each Gaussian splat, we optimize a unit vector $\boldsymbol{s}$ to represent the dominant direction in which a light casts a shadow on it.

Here, $\sigma(\cdot)$ denotes the sigmoid function. The hyperparameters α and β modulate the sharpness and baseline level of shadow effects, respectively, with default values $\alpha = 8$ and $\beta = 10^{-3}$. This differentiable formulation facilitates shadow-material disentanglement during training, as visualized in Figure 7 and validated through ablation in Figure 9.

3.3. Diffusion-Guided Priors

3.3.1. 3DGS Reconstruction with Normal Priors

We observe that 3DGS reconstruction sometimes mistakenly interprets glossy regions as holes, as shown in Figure 8. To address this issue, we incorporate priors from a monocular geometric estimator to improve the output geometric structures. Specifically, we leverage a pretrained diffusion model RGB↔X [34] to provide normal supervision. The supervision loss $\mathcal{L}_{\text{base}}$ is defined as:

$$\begin{aligned} \mathcal{L}_{\text{base}} &= \mathcal{L}_{\text{color}} + \lambda_{\boldsymbol{n}} \mathcal{L}_{\boldsymbol{n}}, \\ \mathcal{L}_{\boldsymbol{n}} &= \sum_{\hat{\boldsymbol{n}} \subset \hat{\boldsymbol{N}}} \mathbb{1} \left(1 - \hat{\boldsymbol{n}}^T \boldsymbol{n}\right), \end{aligned} \tag{7}$$

where $\mathcal{L}_{\text{color}}$ is an RGB reconstruction loss that combines the L1 loss with D-SSIM from 3DGS. We use RGB↔X to estimate the normal vector $\boldsymbol{n}$, treating it as a pseudo ground truth (pseudo-GT) normal, to supervise our rendered normal $\hat{\boldsymbol{n}}$ in the G-buffer pack. We propose a depth-masked weighting scheme to avoid unreliable priors from distant areas. We assign a weight of 0 to pixels with depth values exceeding a default threshold of 0.8 and a weight of 1 otherwise, which we denote as $\mathbb{1}$ in Equation (7). The hyperparameter $\lambda_{\boldsymbol{n}}$ is set to a default value of 0.05. By incorporating normal priors, we improve the accuracy of normal estimation and address geometry reconstruction challenges in textureless regions, providing robust normal estimation.

3.3.2. Light Optimization with Material Priors

To reduce the inherent ambiguity in joint light optimization, we leverage pseudo-GT material attributes derived from RGB↔ X [34]:

$$\mathcal{L}_{\text{material}} = \lambda_R L_2(R, \widehat{R}) + \lambda_{\boldsymbol{A}} L_2(\boldsymbol{A}, \widehat{\boldsymbol{A}}), \tag{8}$$

where $\widehat{R}$ and $\widehat{\boldsymbol{A}}$ denote estimated roughness and RGB albedo maps from the G-buffer pack, while R and $\boldsymbol{A}$ represent their corresponding pseudo-GT values. We omit the metallic term as it shows a weak impact on lighting effects. The hyperparameters λ_R and $\lambda_{\boldsymbol{A}}$ are set to default values of 0.1 and 1.0, respectively. Combining Equations (4), (7) and (8), the total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{light}} + \mathcal{L}_{\text{material}}, \tag{9}$$

where $\mathcal{L}_{\text{light}}$ and $\mathcal{L}_{\text{material}}$ take effect after 10k iterations after an initial scene is reconstructed.

4. Experiments

4.1. Implementation Details

We implement GS-ID and conduct experiments on an NVIDIA RTX 4090 GPU. The pipeline begins by adapting the original 3DGS method [14] for 20k iterations to reconstruct robust scene geometry, assisted by normal priors. For illumination decomposition, we initialize $N_{\text{light}} = M^3$ localized spherical Gaussian mixtures (SGMs) uniformly distributed within the scene's axis-aligned bounding box (AABB), where M controls the spatial resolution. We empirically set $M = 3$ to balance expressiveness and computational efficiency. Each SGM contains $n_{\text{SG}} = 16$ spherical Gaussians. The position $\boldsymbol{p}_j$ of each SGM is defined as:

$$\boldsymbol{p}_j = \boldsymbol{c}_{\text{min}} + \left(\frac{j_x}{M-1}, \frac{j_y}{M-1}, \frac{j_z}{M-1}\right) \odot (\boldsymbol{c}_{\text{max}} - \boldsymbol{c}_{\text{min}}), \tag{10}$$

where $\boldsymbol{c}_{\text{min}}$ and $\boldsymbol{c}_{\text{max}}$ are the AABB bounds (default: $[-3, 3]$), and j_x, j_y, j_z index the grid.

Other hyperparameters, including the shadow vector blending parameters $(\alpha, \beta) = (8, 10^{-3})$, are selected empirically to ensure stable gradient behavior. Crucially, our framework includes light regularization terms—both value- and position-based—that consistently guide the optimization to convergence across different initializations and hyperparameter choices, reducing the sensitivity to manual tuning. To improve training efficiency, we employ an energy-aware pruning strategy that adaptively disables low-activation light sources during optimization:

$$\boldsymbol{w}_{\text{max}} = \max_{i,j,k} \boldsymbol{w}_{ijk}^{(t)}, \quad \tau^{(t)} = \boldsymbol{w}_{\text{max}} + \ln(\delta), \tag{11}$$

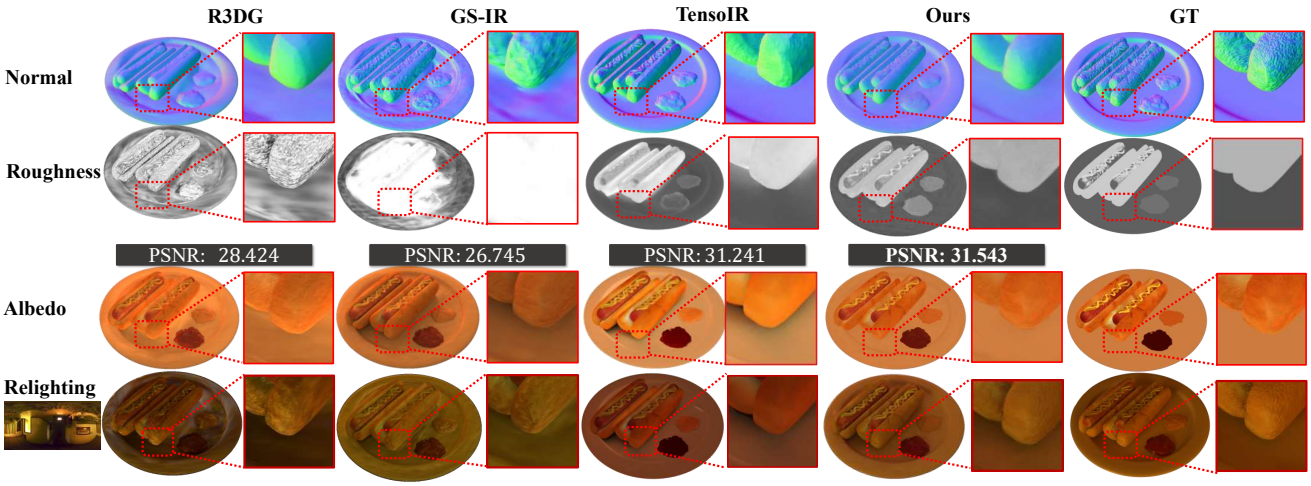

Figure 5. Qualitative evaluation on the TensoIR dataset. GS-ID disentangles shadows effectively and yields more accurate albedo, normal, and roughness maps. Albedo and roughness are evaluated using the same protocol as TensoIR.

	Method	Roughness MSE ↓	Normal MAE ↓	Albedo PSNR ↑	Albedo SSIM ↑	Albedo LPIPS ↓	Novel View Synthesis PSNR ↑	Novel View Synthesis SSIM ↑	Novel View Synthesis LPIPS ↓	Relighting PSNR ↑	Relighting SSIM ↑	Relighting LPIPS ↓	Training Time
NeRF	InvRender [38]	0.008	5.074	27.34	0.933	0.100	27.37	0.934	0.089	23.97	0.901	0.101	15 hrs
	NVDiffrec† [18]	0.010	6.078	29.17	0.908	0.115	30.70	0.962	0.052	19.88	0.879	0.102	1.2 hrs
	TensoIR [11]	0.013	4.100	29.28	0.950	0.085	35.09	0.976	0.040	28.58	0.944	0.081	3.9 hrs
3DGS	GSshader† [10]	0.065	6.647	18.59	0.876	0.092	19.99	0.891	0.089	22.42	0.872	0.103	35 min
	RelightGS† [8]	0.016	6.078	29.47	0.930	0.107	37.57	0.983	0.020	24.41	0.890	0.100	41 min
	GS-IR† [16]	0.027	4.947	30.29	0.941	0.084	35.33	0.974	0.027	24.37	0.885	0.096	26 min
	Ours†	0.007	4.602	33.49	0.952	0.079	39.13	0.984	0.020	28.69	0.947	0.075	40 min

Table 1. Quantitative evaluation on the TensoIR Synthetic dataset. Real-time methods are labeled with †.

Method	Albedo PSNR↑	Albedo SSIM↑	Albedo LIPPS↓	Novel View Synthesis PSNR↑	Novel View Synthesis SSIM↑	Novel View Synthesis LIPPS↓
GSshader† [10]	25.375	0.912	0.071	36.248	0.967	0.043
GS-IR† [16]	25.471	0.926	0.068	36.858	0.971	0.041
RelightGS† [8]	27.223	0.953	0.057	40.228	**0.990**	**0.012**
Ours (3^3 SGMs)†	**29.133**	**0.954**	**0.056**	**40.412**	0.988	**0.012**

Table 2. Quantitative comparison on the ADT dataset.

where $i \in [1, N_{\text{light}}]$, $j \in [1, n_{\text{SG}}]$, k denotes the RGB channels, and δ=10^{-3} by default. SGMs with all weights below threshold τ are progressively pruned throughout training.

The memory complexity of screen-space lighting evaluation, $O(HWN_{\text{light}}n_{\text{SG}})$, is addressed via customized CUDA kernels using chunked processing and G-Buffer–aware computation to reduce peak GPU memory. After light initialization and pruning, joint optimization of geometry, lighting, and material proceeds for an additional 10k iterations under the supervision of pretrained diffusion priors. The full pipeline achieves real-time rendering at 60 fps with approximately 1.2× GPU memory usage and 1.5× longer training time compared to vanilla 3DGS.

4.2. Datasets and Metrics

We evaluate GS-ID on three complementary datasets to comprehensively assess illumination decomposition performance across both synthetic and real-world scenarios. Our primary benchmark is the TensoIR Synthetic dataset [11], which provides four object-centric scenes with ground-truth albedo and roughness maps. We further enhance this dataset by rendering additional screen-space roughness maps to supplement its existing intrinsic ground truth. To validate real-world performance, we adopt two sources: **(1)** A curated relighting dataset of four high-fidelity scanned objects from the Aria Digital Twin (ADT) repository [22], which provides high-resolution geometry and material under varied lighting; **(2)** Nine unbounded real-world scenes from Mip-NeRF 360 [1], covering diverse indoor and outdoor environments with complex, spatially-varying illumination. For evaluation, we use a combination of standard novel view synthesis (NVS) metrics—PSNR, SSIM, and LPIPS—as well as mean angular error (MAE) of surface normals to measure the fidelity of geometric reconstruction. To ensure a fair comparison, we slightly adjust the render-

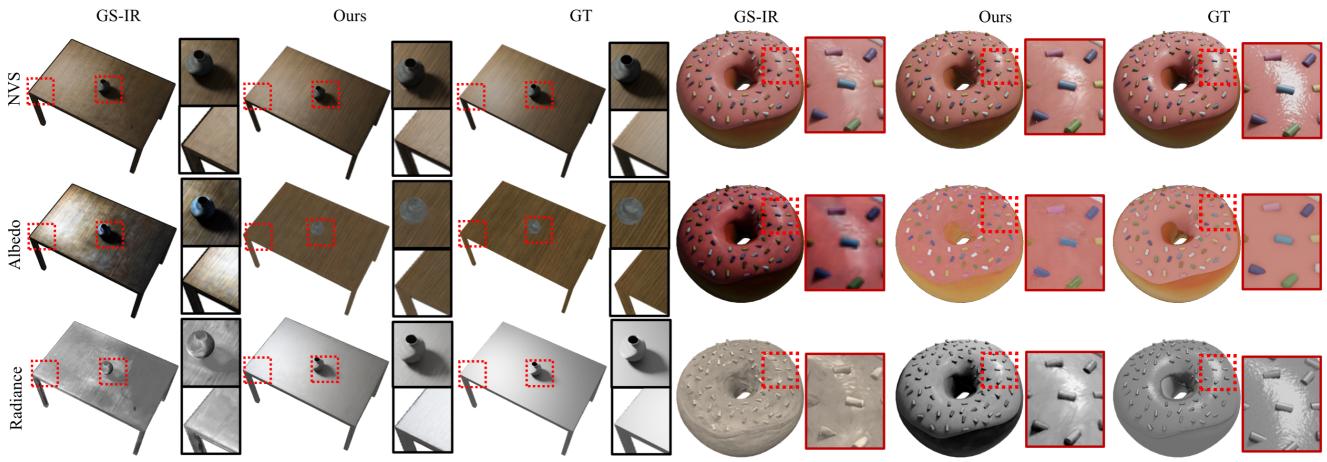

Figure 6. Qualitative comparison on the ADT dataset. GS-ID separates different illumination components and removes specular highlights and shadow artifacts from albedo estimation, significantly outperforming existing solutions under unknown lighting conditions.

Figure 7. Relighting results on the Mip-NeRF 360 dataset. Our adaptive lighting model supports both environment relighting and SGM relighting with shadow maps. Remarkably, relighting is achieved solely by modifying light parameters, with no need for retraining.

ing equation in Equation (1) to:

$$L_o^{\text{radiance}} = \frac{L_o^{\text{env}}(\boldsymbol{x}, \boldsymbol{\omega}_o)}{\hat{\boldsymbol{A}}} + L_o^{\text{SGM}}(\boldsymbol{x}, \boldsymbol{\omega}_o) \cdot V, \tag{12}$$

where $\hat{\boldsymbol{A}}$ denotes the albedo from the G-Buffer and V represents the visibility term. This adjustment aligns our prediction with radiance-only baselines for consistent evaluation. Additional results on more scenes and evaluation details are provided in the supplementary material.

4.3. Comparative Analysis

Illumination Decomposition Analysis on TensoIR. Quantitative results on the TensoIR dataset (Table 1) demonstrate the comprehensive advantages of our method. We achieve a PSNR of 39.13 dB for novel view synthesis (NVS), surpassing the second-best method R3DG by 1.56 dB, and reduce the surface normal estimation error by 24.2% (4.602 vs. 6.078 MAE). While TensoIR achieves slightly lower normal error due to its spatially continuous MLP branch, our GS-based design enables more efficient joint optimization, leading to improved albedo, roughness, and overall relighting performance. Furthermore, our method reduces training time by 87% and significantly lowers memory consumption. It also outperforms all GS-based baselines in both normal estimation and relighting quality.

Illumination Decomposition Analysis on ADT. Our experiments on the ADT dataset focus explicitly on evaluating material (albedo) and illumination (radiant intensity) decomposition accuracy. As shown in Figure 6, our framework robustly separates diffuse reflectance from high-frequency lighting effects, achieving cleaner albedo maps and physically consistent radiance fields compared to baselines—a direct outcome of our adaptive light model. Existing methods, by contrast, exhibit residual highlights contaminating their radiance predictions or oversmoothed/altered textures degrading their estimated albedo (e.g, specu-

lar surfaces). Quantitatively (Table 2), we outperform alternatives across both tasks, with particularly significant margins (+12% PSNR) for NVS due to faithful disentanglement of these components.

Illumination Decomposition Analysis on Real-World Dataset. On the unbounded real-world Mip-NeRF 360 dataset, our framework demonstrates robust performance under challenging natural illumination. As shown in Figure 7, our method supports both far-field and near-field relighting with high visual fidelity.

4.4. Ablation Study

We conduct an ablation study to evaluate the effectiveness of the introduced diffusion priors, the proposed lighting model, and our CUDA deferred shading scheme.

Analysis of Diffusion Priors. To validate the efficacy of diffusion-derived priors, we categorize them into two distinct components: normal priors and material priors. The quantitative comparisons in Table 3 show that both priors improve the results of NVS and illumination decomposition. Figure 8 shows that normal prior elimination leads to holistic misestimation of specular areas. Our extensive validation confirms that the proposed hyperparameter configuration achieves optimal performance.

Analysis of Lighting Model. GS-ID employs an adaptive lighting model that combines parametric SGMs with an environment map to capture complex illumination. We conduct ablations by disabling SGMs and removing the EnvMap. As shown in Table 3, combining both components improves albedo and NVS accuracy, with SGMs offering a larger PSNR gain for NVS (38.18 dB vs. 36.67 dB). Figure 9 shows SGMs better capture high-frequency specularities, while the absence of a shadow field hinders shadow removal, reducing albedo accuracy by 3.213 dB.

Analysis of Shading Schemes. Forward shading with PBR requires substantial computational resources, as it computes the rendering equation for each primitive. This issue is exacerbated when rendering with Gaussian Splatting, where millions of GS points are typically present in a scene. We compared the training times between forward and deferred shading across various scenes from the Mip-NeRF 360 dataset [1]. The results (Table 4) demonstrate that deferred shading accelerates training, with faster scaling with the number of points, achieving up to a 4 times speedup. Furthermore, our CUDA implementation of differentiable rendering enables saving over 40% storage space.

Discussion and Limitations. Our work has achieved good performance in illumination decomposition, but some directions are worth further exploration. We found that natural materials and geometry are often isotropic, while our current 3DGS representation is anisotropic, creating extra Gaussian shapes. We hope to develop an isotropic representation to save storage space in the future.

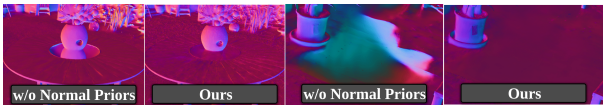

Figure 8. Effects of normal priors. Normal priors prevent areas with specular reflections from being misestimated as holes, significantly improving normal estimation accuracy.

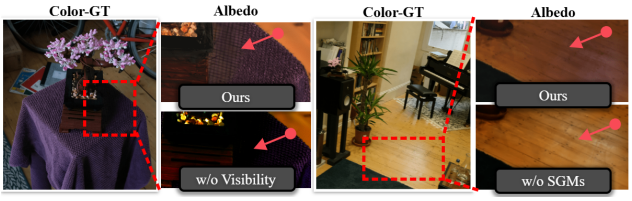

Figure 9. Effects of lighting components. Left: Shadow field estimation removes shadow artifacts from albedo estimation. Right: Adaptive SGMs enable accurate specular highlight reconstruction in albedo estimation.

Method	Normal	Albedo			Novel View Synthesis		
	MAE↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o normal	5.187	32.140	0.941	0.089	37.962	0.978	0.028
w/o material	4.867	29.585	0.935	0.096	38.189	0.975	0.027
w/o all priors	5.286	29.101	0.921	0.099	37.950	0.962	0.030
w/o SGMs	5.916	26.725	0.911	0.096	36.670	0.953	0.031
w/o EnvMap	4.701	32.320	0.942	0.090	38.180	0.975	0.028
w/o shadow field	4.751	30.280	0.941	0.091	37.580	0.974	0.029
single SGM	4.616	32.191	0.950	0.089	38.670	0.982	0.025
forward	4.660	32.688	0.951	0.088	39.015	0.980	0.024
Ours	**4.602**	**33.493**	**0.952**	**0.079**	**39.130**	**0.984**	**0.020**

Table 3. Ablation study on the TensoIR Synthetic dataset. The results show the effects of priors from diffusion models, our proposed deshadowing model, and the adaptive lighting model.

Scene	Points	Training Time↓ (w. CUDA)		GPU Memory↓ (Def. Shading)	
		Fwd. Shading	Def. Shading	w/o CUDA	w. CUDA
room	1.23M	82min	62min	28.2G	18.5G
kitchen	1.31M	88min	66min	29.3G	18.8G
garden	3.55M	299min	68min	36.8G	21.4G
bicycle	3.84M	305min	70min	38.9G	21.9G

Table 4. Comparison of shading schemes on the Mip-NeRF 360 [1] dataset. Deferred rendering can achieve up to a 4× acceleration in complex scenes, with consistent speedup. Our CUDA implementation can save over 40% GPU memory.

5. Conclusion

We propose GS-ID, an end-to-end framework for illumination decomposition on 3D Gaussian Splatting. It integrates an adaptive lighting model, a deshadowing module, and geometry/material priors extracted from pretrained diffusion models. GS-ID further incorporates a customized CUDA-based optimization pipeline and deferred shading to improve convergence and efficiency. Experiments show that GS-ID surpasses existing methods in both illumination and intrinsic decomposition. Moreover, it enables high-quality, user-controllable light editing and scene composition, supporting diverse downstream applications.

Acknowledgements. We sincerely appreciate the Meituan Academy of Robotics' constructive feedback throughout the project.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 6, 8

[2] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural Reflectance Fields for Appearance Acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 2

[3] Zoubin Bi, Yixin Zeng, Chong Zeng, Fan Pei, Xiang Feng, Kun Zhou, and Hongzhi Wu. Gs3: Efficient relighting with triple gaussian splatting. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–12, 2024. 2

[4] 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 IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 2

[5] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural Representations for Videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 2

[6] Epic Games, Inc. The Most Powerful Real-Time 3D Creation Tool - Unreal Engine. https://www.unrealengine.com/, 2024. Accessed Sep 12, 2024. 2

[7] Fan Fei, Jiajun Tang, Ping Tan, and Boxin Shi. VMINer: Versatile Multi-view Inverse Rendering with Near-and Far-field Light Sources. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11800–11809, 2024. 2, 3

[8] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3D Gaussian: Real-time Point Cloud Relighting with BRDF Decomposition and Ray Tracing. *arXiv:2311.16043*, 2023. 6

[9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *SIGGRAPH (Conference Paper Track)*, page 32. ACM, 2024. 2

[10] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. GaussianShader: 3D Gaussian Splatting With Shading Functions for Reflective Surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024. 2, 6

[11] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023. 2, 6

[12] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual Contouring of Hermite Data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 339–346, 2002. 2

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

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 4, 5

[15] Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. Intrinsic Image Diffusion for Indoor Single-view Material Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5198–5208, 2024. 2

[16] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. *CoRR*, abs/2311.16473, 2023. 2, 4, 6

[17] William E Lorensen and Harvey E Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353, 1998. 2

[18] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 2, 6

[19] Fred E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Appl. Opt.*, 4(7):767–775, 1965. 3

[20] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable Volumetric Rendering: Learning Implicit 3D Representations Without 3D Supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3504–3515, 2020. 2

[21] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[22] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20133–20143, 2023. 6

[23] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[24] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2

[25] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 2

[26] Unity Technologies. Unity Real-Time Development Platform — 3D, 2D, VR & AR Engine. https://unity.com/, 2024. Accessed Sep 12, 2024. 2

[27] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10, 2009. 3

[28] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[29] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning Indoor Inverse Rendering With 3D Spatially-Varying Lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12538–12547, 2021. 2

[30] Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf: Neural Incident Light Field for Physically-Based Material Estimation. In *European Conference on Computer Vision*, pages 700–716. Springer, 2022. 2

[31] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 2

[32] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume Rendering of Neural Implicit Surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2

[33] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian Opacity Fields: Efficient and Compact Surface Reconstruction in Unbounded Scenes. *CoRR*, abs/2404.10772, 2024. 2

[34] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Milos Hasan. RGB↔X: Image Decomposition and Synthesis Using Material- and Lighting-Aware Diffusion Models. In *SIGGRAPH (Conference Paper Track)*, page 75. ACM, 2024. 2, 5

[35] Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf++: Inter-Reflectable Light Fields for Geometry and Material Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3601–3610, 2023. 2

[36] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021. 2

[37] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 2

[38] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling Indirect Illumination for Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022. 2, 6

[39] Jingsen Zhu, Fujun Luan, Yuchi Huo, Zihao Lin, Zhihua Zhong, Dianbing Xi, Rui Wang, Hujun Bao, Jiaxiang Zheng, and Rui Tang. Learning-based Inverse Rendering of Complex Indoor Scenes with Differentiable Monte Carlo Raytracing. In *SIGGRAPH Asia*, pages 6:1–6:8. ACM, 2022. 2

[40] Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker. IRISformer: Dense Vision Transformers for Single-Image Inverse Rendering in Indoor Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2822–2831, 2022. 2