

GANG: Geometrically-Aligned Neural Gaussians for Efficient and Realistic Relighting

Deqi Li, Shi-Sheng Huang, Hongbo Fu *Member, IEEE*, Hua Huang, *Senior Member, IEEE*

Abstract—Efficient and realistic relighting of complex scenes with unknown illumination remains a crucial but challenging task. Recent advancements in 3D Gaussian Splatting (3DGS) have shown impressive object-level relighting. However, they still struggle with complex real-world scenes, mainly due to the challenges of accurately decoupling intricate geometry, materials, and lighting using *concise* 3D Gaussian primitives. In this paper, we propose a new Geometrically-Aligned Neural Gaussian Splatting (GANG) method, which performs *efficient* physically based rendering (PBR) directly on anchor-based relightable neural Gaussians. Our key idea is to regularize the decoded neural Gaussians *geometrically aligned* with the latent signed distance field (SDF) surface spawned from anchors using a differentiable implicit indicator function (IIF) solver. It brings effective geometric association to accurate decoupling of materials and lighting for efficient and realistic relighting of complex scenes. Furthermore, we propose a *locally consistent* geometry regularization to guide more concise neural Gaussian learning with a hybrid lighting model, which combines position-learnable spherical Gaussians (SGs) and an environment map, allowing accurate modeling of both local and global illumination. Experimental results on public datasets demonstrate that GANG consistently outperforms previous PBR methods in material decomposition and relighting quality, while representing complex scenes with concise anchors. To the best of our knowledge, GANG is a new state-of-the-art 3DGS method for realistic relighting, enabling efficient rendering and flexible editing materials and illumination, especially for complex scenes.

Index Terms—Neural Gaussian splatting; Adaptive geometric alignment; Hybrid lighting model; Realistic relighting



1 INTRODUCTION

Realistic relighting is an important topic in the computer vision and computer graphics communities. It has been widely applied in virtual reality (VR) and augmented reality (AR) to deliver immersive experiences [1], [2]. The key task of image relighting is to infer the intrinsic properties (geometry, physical materials, and lighting) of a scene captured by one or more images. However, accurately decomposing these properties and achieving high-fidelity relighting remains challenging, especially in complex scenes with intricate geometry, materials, and unknown illumination.

Recently, some methods have attempted to alleviate this ill-posed property decomposition problem using neural radiance fields (NeRF) [3], employing techniques such as tensor decomposition [4], illumination decomposition [5], [6], and semantic priors [7]. However, NeRF-based approaches require significant computational complexity, causing non-negligible latency for relighting. Taking advantage of efficient 3D Gaussian splatting (3DGS) [8], recent methods have explored intrinsic decomposition in Gaussian splatting by assigning additional learnable parameters to each Gaussian primitive for relighting. [9], [10], [11], [12]. However, this per-primitive strategy inevitably introduces redundant Gaussians, leading to increased storage overhead. More importantly, in complex scene relighting, these methods often disregard the principle of local material consistency,

whereby real-world objects typically exhibit smoothly varying or homogeneous properties within local regions. Independent optimization of individual Gaussians ignores this inherent regularity, exacerbating the entanglement between material representation and illumination estimation, and ultimately degrading relighting quality in complex scenes.

For complex scenes, anchor-based methods [13], [14] have shown potential to balance reconstruction quality and rendering efficiency. However, in these approaches, the neural Gaussians are often decoded independently, lacking an effective geometric association with the anchors. This induces geometric uncertainty with disordered spatial organization, leading to non-compact reconstruction for the scene. Such uncertainty further degrades the accuracy of geometry, material, and lighting decomposition, often producing floaters and other artifacts in relighting tasks.

In this paper, we propose a novel Geometrically-Aligned Neural Gaussian Splatting, called GANG, which can accurately decouple the geometry, materials, and lighting for complex scenes using a *concise* anchor-based neural Gaussian representation, thus achieving efficient and realistic relighting. Inspired by recent anchor-based neural Gaussians [13], [14], our GANG adopts a similar anchor-based structure but predicts the anchor feature into relightable 3D Gaussian primitives using a lightweight network, thus leading to a new anchor-based relightable neural Gaussian representation. More importantly, we propose to regularize the neural Gaussians to be *geometrically aligned* with the latent SDF surface that is spawned by the anchors using a differentiable implicit indicator function (IIF) solver. It introduces an effective geometric association between anchors and their predicted neural Gaussians. Besides, we perform Physically Based Rendering (PBR) directly on the

- Deqi Li, Shi-Sheng Huang, Hua Huang are with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China. E-mail: dqli@mail.bnu.edu.cn, huangss@bnu.edu.cn, huahuang@bnu.edu.cn. Hongbo Fu with the School of Creative Media, City University of Hong Kong. E-mail: hongbofu@ust.hk.

Hua Huang is the corresponding author.

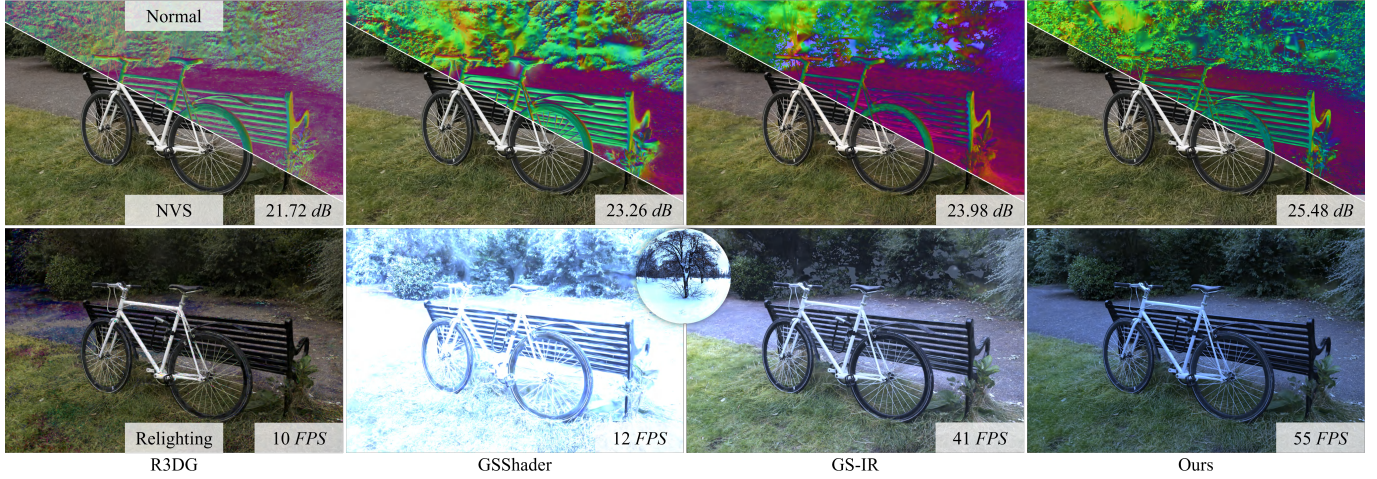


Fig. 1. Efficient and realistic relighting of complex scenes by Geometrically-Aligned Neural Gaussians (GANG), which more accurately decomposes materials from unknown illumination scenes, and more efficiently performs more realistic relighting than state-of-the-art PBR-based methods.

neural Gaussians, preserving the realistic rendering quality and efficient rendering speed simultaneously. Moreover, we further regularize the neural Gaussians spawned from the same anchor as a local plane with locally consistent geometry and material properties, where local geometric structures can be naturally perceived with the stellate neural representation. To simulate unknown illumination, we propose a hybrid lighting representation that combines position-learnable spherical Gaussians (SGs) with a cube-map environment map, and build up a highly accurate decomposition of geometry, materials, and lighting for the anchor-based relightable neural Gaussian learning. Benefiting from these components, our GANG achieves precise scene decomposition, efficient and realistic relighting, while enabling highly freeform material and lighting editing in a very efficient manner.

To evaluate the effectiveness of the proposed GANG, we conducted extensive experiments comparing it with the baseline of novel view synthesis (NVS) (3DGS [8], 2DGS [15], OCTree-GS [14]) and state-of-the-art PBR-based relightable Gaussian methods (R3DGS [11], Gaussian-Shader [10], GS-IR [9]) on three public datasets (Mip-NeRF 360 [16], Deep T&T [17], [18], and TensorIR Synthetic [4]). The experimental results show that our GANG can achieve superior performance in novel view synthesis, material decomposition, and relighting compared to previous PBR-based methods. Specifically, our GANG achieves 1.5dB higher relighting PSNR and 2dB better albedo decomposition than R3DGS and GS-IR, along with a more 8dB relighting gain over GSShader. For NVS task, our GANG also achieves better rendering quality with about 3dB PSNR accuracy than all PBR-based methods. Furthermore, GANG demonstrates outstanding storage and rendering efficiency. It only requires half the training time of R3DGS, but achieves a rendering speed of $5\times$ faster and a storage cost of $1/25$.

In summary, GANG establishes a novel Gaussian-based relightable rendering method that delivers efficient and realistic rendering, especially aimed at complex scenes.

1) We design an adaptive geometry alignment optimization strategy that associates the anchor in the implicit

space with the geometric distribution of predicted neural Gaussians, enabling compact and concise scene geometry representation.

2) We propose a local consistency optimization strategy that refines local geometry by constraining surface normals and distances to camera, while enforcing consistent Gaussian material properties to achieve smooth distributions.

3) We construct a hybrid lighting representation that combines the learnable spherical Gaussians with a learnable environment map, enabling accurate modeling of both local and global illumination under unknown lighting conditions and improving the accuracy of material decomposition.

2 RELATED WORK

2.1 Scene Representation

A faithful 3D reconstruction of a scene is fundamental for its relighting. Recently, implicit neural radiance fields (NeRF) and its variants [3], [19], [20], [21] have achieved significant success but require substantial computational resources. Although several variants have reduced the training time from days to hours by introducing voxel grids [22], tri-planes [23], or hash encoding [24], the expensive sample queries for volume rendering still result in noticeable rendering latency.

More recently, 3DGS [8] has gained extensive attention, enabling efficient training and rendering, inspiring a wide range of subsequent methods and applications [25]. To improve the quality of geometric reconstruction, subsequent work has introduced various improvements. These include normal priors [26], opacity fields [27], optimized pruning strategies [28], unite SDF [29], introduced constraints [30], and flattened 3D ellipsoids to 2D ellipses [15]. However, these methods bear the high overhead of storing million of Gaussian primitives for real scene. Although recent anchor-based representations have demonstrated significant storage reduction by decoding Gaussian properties from latent features via tiny MLPs [13], [14]. These methods still lack an effective geometric association between anchors and their spawned neural Gaussians for compact Gaussian learning.

In contrast, our GANG adopts a similar anchor-based structure but uses a differentiable IIF solver to introduce

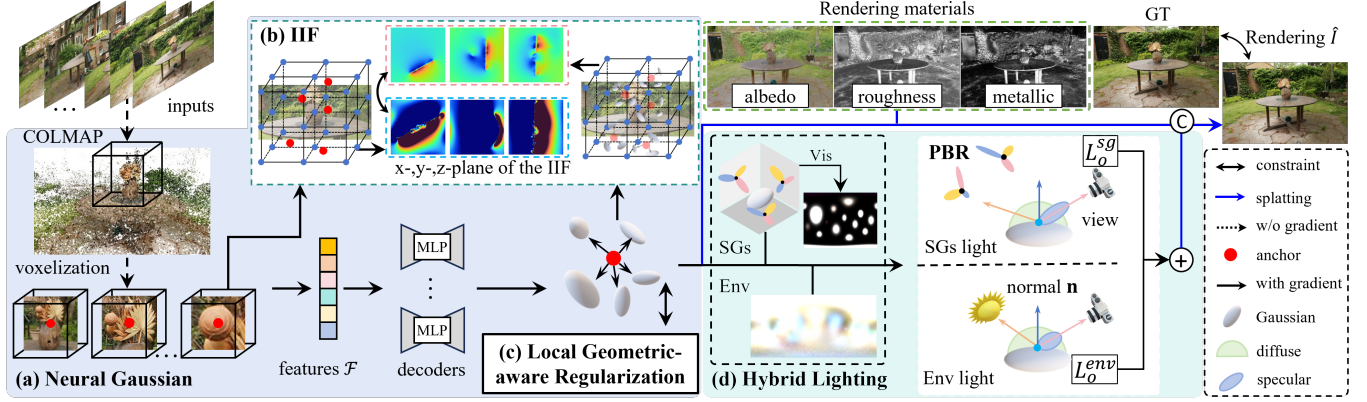


Fig. 2. The pipeline of GANG. Given a set of complex scene images in an uncontrolled environment, we propose to reconstruct the complex scene using anchor-based relightable neural Gaussians, where the anchors (red dot) and neural Gaussians (a), (Sec. 3.1)) are geometrically aligned via a IIF solver (Sec. 3.2). A hybrid lighting model is proposed to accurately simulate environmental lighting both globally and locally (Sec. 3.3)(c). Benefiting from the stellate representation, we treat neural Gaussians spawned from the same anchor as forming a local plane and optimize them with locally consistent geometric and material regularization (Sec. 3.4). Furthermore, we employ a two-stage learning strategy (Sec. 3.5) to progressively refine the representation, enabling efficient and realistic relighting.

an effective geometric association between anchors and their predicted neural Gaussians, leading to a geometrically aligned relightable anchor-based neural Gaussian representation. Moreover, we propose using local consistency to ensure smooth transitions among primitives, which further improve the compactness of neural Gaussian reconstruction.

2.2 Neural Relighting

Decoupling materials and lighting from multi-view images is challenging due to their intrinsic coupling complexity. Previous methods relied on controllable lighting conditions to simplify decomposition problems, such as fixing the lighting and focusing on rotating objects [31], using a flashlight from a mobile phone [32], or using known or calibrated lighting [33]. However, these approaches are restricted to small objects with special acquisition conditions, limiting their applicability to diverse real-world scenes. To address these limitations, some neural representations have been proposed to model lighting and view-dependent material properties, such as single-scattering participating media lighting model [34], spherical Gaussians (SG) [35], learnable environment maps [36], incident lights [37] and surface intersection fields [38]. However, these methods require large computational resources and hours of training.

Recently, with the rise of 3DGS, various methods have explored efficient relighting with different lighting models. GS³ [39] decomposes materials under known lighting conditions, generating realistic shadows. GS-Phong [40] applies the Blinn-Phong model [41] to decompose scenes into materials. PRTGS [42] employs precomputed radiance transfer to improve real-time rendering under dynamic lighting. GlossyGS [43] integrates material priors and normal map prefiltering to accurately reconstruct high-fidelity geometry and materials of glossy objects. Other methods use the bidirectional reflection distribution function (BRDF) to model the scene, such as R3DG [11], GSShader [10], GS-IR [9], and GS-ID [12]. However, the additional learnable material parameters for each Gaussian greatly increase the storage overhead. The main limitation of these methods is

their focus on relighting or shading small objects only. The lack of relighting design for complex scenes makes these methods ineffective in relighting scenes. Furthermore, the latest Gaussian ray-tracers [44], [45], [46] simulate advanced secondary lighting effects for remarkable rendering quality. However, these methods often cost huge computations for ray tracing even at the object level [45], which would not be suitable for efficient relightable rendering for complex scenes such as ours.

Our GANG directly performs PBR directly on anchor-based neural Gaussian primitives, avoiding G-buffer rendering [9], [12] and thereby preserving both realism and efficiency. Moreover, our hybrid lighting model combines SGs and a learnable environment map to effectively simulate global and local lighting in unknown scenes.

3 METHOD

Given multi-view images of a complex scene captured in an uncontrolled environment with unknown illumination, our approach aims to reconstruct the scene using concise geometrically aligned neural Gaussians (GANG), while accurately decoupling the materials and lighting of the scene on the level of 3D Gaussian primitives, thus enabling efficient and realistic relighting. Fig. 2 shows the main pipeline of our approach.

3.1 Relightable Neural Gaussian

Inspired by Scaffold-GS [13] and Octree-GS [14], we propose a new anchor-based relightable Gaussian representation by forcing the neural Gaussians geometrically aligned with the latent SDF surface from anchors. We perform PBR directly on the neural Gaussians (with PBR color rendering and image splatting), thus enabling efficient and realistic relighting simultaneously.

Anchor-based Relightable Neural Gaussians. Given a set of unorganized RGB images $\{I_i | i = 1, \dots, N\}$ as input with image number N , we first use COLMAP [47] to generate a sparse point cloud P_{col} and estimate the camera

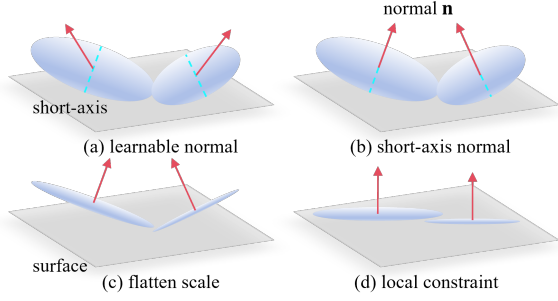


Fig. 3. An illustration of the normal definition. The learnable normal in R3DG [11] lacks a clear physical interpretation (a). The shortest-axis normal provides a well-defined direction but still preserves the Gaussian volume effect (b). The introduced flattening loss (c) helps reduce this volume influence, while the proposed local constraints (d) enforce consistent orientations of the constrained local normals, producing both a smooth and planar effect on the representation of local geometry.

parameters. P_{col} is divided into voxels following Octree-GS [14], and the anchors V are defined as a set of learnable intrinsic properties, with positions initialized to the centroid of the voxels. For each anchor $v \in V$, the intrinsic properties are $v = \{x_v, l_v, r_v, f_v, o_v\}$, where $x_v, l_v \in \mathbb{R}^3$ are the position and factor of scale, $r_v \in \mathbb{R}^4$ is the rotation represented by a set of quaternions, $f_v \in \mathbb{R}^{32}$ is a context feature, and $o_v \in \mathbb{R}^{K \times 3}$ is an offset.

Then the multi-head decoder F regresses each anchor v to a set of K relightable Gaussian primitives $\{\mathbf{g}_k | k = 1, \dots, K\}$. Specifically, each anchor is represented with a view-dependent latent feature $\mathcal{F} = \{f_v, \Delta_v, d_v\}$. Here, $\Delta_v = \|v - V_c\|_2$ is the relative viewing distance, and $d_v = \text{normalize}(v - V_c)$ is the direction of v to the camera position V_c . Each neural Gaussian is decoded from the latent feature $\mathcal{F}$ to obtain a set of relightable parameters with geometry, appearance, and material as $\mathbf{g}_k = \{\mu_k, s_k, q_k, \alpha_k, c_k, A_k, \rho_k, M_k\}$. Among them, the position $\mu_k \in \mathbb{R}^3$, the scale $s_k \in \mathbb{R}^3$, and the quaternion rotation $q_k \in \mathbb{R}^4$ associated with the covariance, are formulated as

$$\begin{aligned} \{\mu_0, \dots, \mu_{K-1}\} &= x_v + \{o_{v,1}, \dots, o_{v,K-1}\} \cdot l_v, \\ \{s_0, \dots, s_{K-1}\} &= \text{sigmoid}(F_s(\mathcal{F})) \cdot l_v, \\ \{q_0, \dots, q_{K-1}\} &= \text{normalize}(F_q(\mathcal{F}) \cdot r_v). \end{aligned} \quad (1)$$

The other parameters $\phi \in \{\alpha_k, c_k, A_k, \rho_k, M_k\}$ are directly decoded by the MLP-based decoders in F , formulated as $\phi = \mathcal{A}_\phi(F_\phi(\mathcal{F}))$, where $\mathcal{A}_\phi$ is an activation function of the parameter ϕ , F_ϕ is the decoder of ϕ , $c_k, A_k \in \mathbb{R}^3$ are color and albedo, and $\alpha_k, \rho_k, M_k \in \mathbb{R}^1$ are opacity, roughness, and metallic, respectively.

The anchor point v and the neural Gaussian $\mathbf{g}_k$ share the same normal definition. The normal $\mathbf{n}_k$ of $\mathbf{g}_k$ is jointly determined by the scale s_k , the rotation vector q_k , and the viewing direction V_c , representing the rotation axis corresponding to the smallest scale that is aligned with the viewing direction. It is formulated as

$$\begin{aligned} \mathbf{n}_{d,k} &= Q_{i,k}(\arg\min_{i \in \{0,1,2\}} s_{i,k}), \\ \mathbf{n}_k &= \begin{cases} \mathbf{n}_{d,k}, & \text{if } \mathbf{n}_{d,k} \cdot V_n \geq 0, \\ -\mathbf{n}_{d,k}, & \text{else } \mathbf{n}_{d,k} \cdot -V_n < 0. \end{cases} \end{aligned} \quad (2)$$

Here, $Q_{i,k}$ is the rotation matrix constructed from the rota-

tion vector q_k , i denotes the index of the axis corresponding to the smallest scale in s_k , and $V_n = \text{normalized}(\mu_k - V_c)$ is the unit vector from the primitive position μ_k to the camera position V_c . In Fig. 3, we illustrate the comparison of normals under different definitions and constraints.

PBR Color Rendering. For each relightable Gaussian primitive $\mathbf{g}$, we use the PBR technology to calculate its PBR color c' . Specifically, for $\mathbf{g}$ located in μ with normal $\mathbf{n}$, given a light intensity $L_i(w_i, \mu)$ along the incident direction w_i , the PBR color c' of $\mathbf{g}$ along the view direction w_o can be formulated as

$$c'(\mu, w_i) = \int_{\Omega} L_i(\mu, w_i) f_r(w_i, w_o) (w_i \cdot \mathbf{n}) dw_i, \quad (3)$$

where Ω is the upper hemisphere. Each neural Gaussian is considered to be a Cook-Torrance microfacet [48]. The bidirectional reflection distribution function (BRDF) is used to model the reflection intensity as

$$f_r(w_i, w_o) = (1 - M) \frac{A}{\pi} + \frac{DFG}{4(\mathbf{n} \cdot w_i)(\mathbf{n} \cdot w_o)}, \quad (4)$$

where $A, M, \rho \in [0, 1]$ are albedo, metallic and roughness, $\mathbf{n}$ is normal, D is the normalized distribution function, F is the Fresnel function, and G is the geometry function. Eq. 3 can be rewritten as $c' = c'_d + c'_s$, where c'_d is a diffuse term and c'_s is a specular term,

$$\begin{aligned} c'_d &= (1 - M) \frac{A}{\pi} \int_{\Omega} L_i(\mu, w_i) (w_i \cdot \mathbf{n}) dw_i, \\ c'_s &= \int_{\Omega} \frac{DFG}{4(\mathbf{n} \cdot w_i)(\mathbf{n} \cdot w_o)} L_i(\mu, w_i) (w_i \cdot \mathbf{n}) dw_i. \end{aligned} \quad (5)$$

Image Splatting. Once the PBR color for each relightable Gaussian primitive is calculated, we follow the projection process to splat 3D Gaussian primitives on the 2D image plane. Same as 3DGS [8], a tile-based rasterizer is used to splat the color and materials, which efficiently sorts 2D primitives and employs α -blending in N -ordered points.

3.2 Implicit Geometric Alignment

To spawn compact neural Gaussians, each anchor is best suited to fit the surface of the scene, and spawns neural Gaussians along the surface. However, the lack of real-scene surface priors and weak anchor-Gaussian geometric association through offset as done by previous works [13], [14] often lead to discretely spawning neural Gaussians from the same anchor with excessive distance and anisotropy. To alleviate this issue, We introduce the Implicit Indicator Function (IIF) [49] to achieve latent-space surface alignment between the anchor and the neural Gaussian. By encouraging the consistency of their distributions, this association allows for adaptive geometric correction and leads to more accurate and consistent geometric alignment.

Specifically, we use position and normal of anchor and neural Gaussian to describe the geometry of the scene. By scaling the scene into a unit space and sampling a grid 128^3 , the vertices are treated as probes to describe the varying numbers of anchors and neural Gaussians on a unified scale. Following IIF [49], we first obtain the uniformly discretized point field $\mathcal{P}$ and normal field $\mathcal{N}$. It is achieved by rasterizing the points and normals of each Gaussian or anchor onto

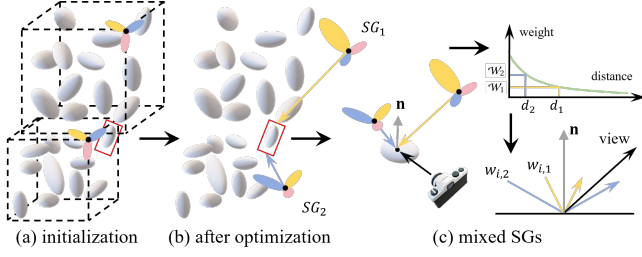


Fig. 4. An illustration of the multiple spherical Gaussian mixtures. The initial positions of the SGs are set at the upper vertex of the voxel (a). After optimization, the positions and properties of the SGs are adaptively optimized to simulate direct lighting (b) through a distance-weighted spherical Gaussian mixture (c).

a uniformly sampled voxel grid via differentiable inverse weighted trilinear interpolation. The spectral method [50] is applied to reconstruct the latent function from these discretized fields. For neural Gaussian, the normalized IIF $\mathcal{I}_n$ is formulated as

$$\mathcal{I}_n = \frac{m}{\text{abs}(\mathcal{I}'_n|_{x=0})} (\mathcal{I}'_n - \frac{1}{|\{\mu\}|} \sum_{c \in \{\mu\}} \mathcal{I}'_n|_{x=c}), \quad (6)$$

$$\mathcal{I}'_n = \text{IFFT}(\tilde{g}_{\sigma,r}(\mathcal{P}_n) \odot \frac{i\mathcal{P}_n \cdot \hat{\mathcal{N}}_n}{-2\pi\|\mathcal{P}_n\|^2}).$$

where $\mathcal{I}'_n$ is the unnormalized indicator function, $\tilde{g}_{\sigma,r}(\cdot)$ represents a Gaussian smoothing kernel with bandwidth σ and grid resolution r in the spectral domain, $\odot$ is the element-wise product, $\hat{\mathcal{N}}_n$ is the spectral feature of $\mathcal{N}_n$ obtained via the Fast Fourier transform (FFT). The IIF of the anchor $\mathcal{I}_a$ is calculated in the same way.

Finally, the adaptive geometric alignment between anchor and neural Gaussian is achieved by minimizing the difference between $\mathcal{I}_a$ and $\mathcal{I}_n$. It is formulated as

$$\mathcal{L}_{\text{IIF}} = \frac{1}{N} \sum_{i=1}^N (\mathcal{I}_n^{(i)} - \mathcal{I}_a^{(i)})^2, \quad (7)$$

where N is the total number of grid points.

3.3 Hybrid Lighting Rendering

To accurately simulate the real illumination of the scene, we use a hybrid lighting representation to model incident lighting L_i including global ambient lighting L_i^{env} and local direct lighting L_i^{sg} .

Global Ambient Lighting. Global ambient lighting is represented as a learnable cubemap, allowing the model to capture spatially varying illumination and efficiently optimize lighting during training. Following GS-IR [9], we use the split-sum approximation [51] to handle the integral in Eq. 3, allowing Eq. 5 to be rewritten in a tractable form as

$$c_d^{\text{env}} \approx (1 - M) \frac{A}{\pi} I_d^{\text{cub}}, \quad l = \langle w_i \cdot \mathbf{n} \rangle, \quad (8)$$

$$c_s^{\text{env}} \approx \underbrace{\int_{\Omega} \frac{DFG}{4\langle \mathbf{n} \cdot w_i \rangle \langle \mathbf{n} \cdot w_o \rangle} l dw_i}_{\text{Env BRDF } K_{\text{env}}} \underbrace{\int_{\Omega} DL_i(\mu, w_i) l dw_i}_{\text{Pre-Filtered Env Map } I_s^{\text{cub}}}$$

where K_{env} can be precomputed and stored in a lookup table, while I_d^{cub} and I_s^{cub} denote the embedded diffuse and

specular ambient components within the learnable environment cubemap.

Multiple Spherical Gaussians. We use multiple spherical Gaussians (SGs) [35], [52], [53] to simulate local direct illumination with additional learnable positions x_{sg} , through a distance-weighted spherical Gaussian mixture. As shown in Fig. 4 (a), we apply the K-means algorithm to partition the neural Gaussian into K_s voxels, where the number of SGs $K_s = K^2$ is adaptively determined based on the number of layers K in the structured grids of the neural Gaussian. The initial position of an SG is defined as the upper vertex of the voxel farthest from the clustered Gaussians. The local direct lighting L_i^{sg} takes a mixture spherical function

$$L_i^{\text{sg}}(w_i) = \sum_{j=1}^{K_s} SG(w_i; \xi_j, \lambda_j, \delta_j), \quad (9)$$

$$SG(\mathcal{V}; \xi, \lambda, \delta) = \delta e^{\lambda(\mathcal{V} \cdot \xi - 1)},$$

where $\mathcal{V} \in \mathcal{S}^2$ is the input, $\xi_j \in \mathcal{R}^3$, $\lambda \in \mathcal{R}$, $\delta_j \in \mathcal{R}^3$ denotes the lobe axis, the amplitude, and the sharpness of j -th SG, respectively. According to Eq. 5, the PBR color of SG is rewritten as

$$c_d^{\text{sg}} = (1 - M) \frac{A}{\pi} \sum_{j=1}^{K_s} \mathcal{W}_j SG(w_{i,j}), \quad c_s^{\text{sg}} = \sum_{j=1}^{K_s} \mathcal{W}_j SG_{s,j}, \quad (10)$$

where $SG(w_{i,j})$ is the diffuse term with direction $w_{i,j}$ from the j -th SG to the neural Gaussian, and $SG_{s,j}$ is the specular term. As shown in Fig. 4 (c), $\mathcal{W}_j = e^{-d}$ is a distance weight, where d is the distance from x_{sg} to μ . The overall rendering color of $\mathbf{g}$ in the hybrid lighting model is formulated as

$$c'(\mu, w_i) = c_d^{\text{env}} + c_s^{\text{env}} + c_d^{\text{sg}} + c_s^{\text{sg}}. \quad (11)$$

3.4 Local Geometric-Aware Regularization

A faithful Gaussian distribution is crucial to achieve high-quality relighting. However, unordered Gaussian primitives often make it challenging to reconstruct a compact 3D Gaussian representation of a complex scene, especially for neural Gaussian representations [13], [14] like ours. As shown in Fig. 5, we further propose geometric-aware optimization by employing local geometric and material consistency within neighbor primitives to mitigate this issue and make the local transition smoother.

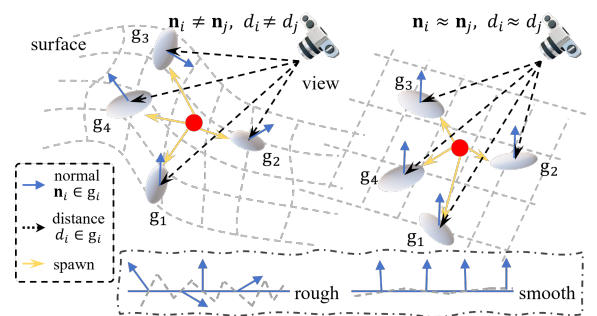


Fig. 5. Local geometry regularization in the geometry structure of neural Gaussian primitives (colored gray) spawned from an anchor (colored red). Using such regularization, the unordered Gaussian primitives (left) can be effectively adjusted (right) to a smoother local plane (down).

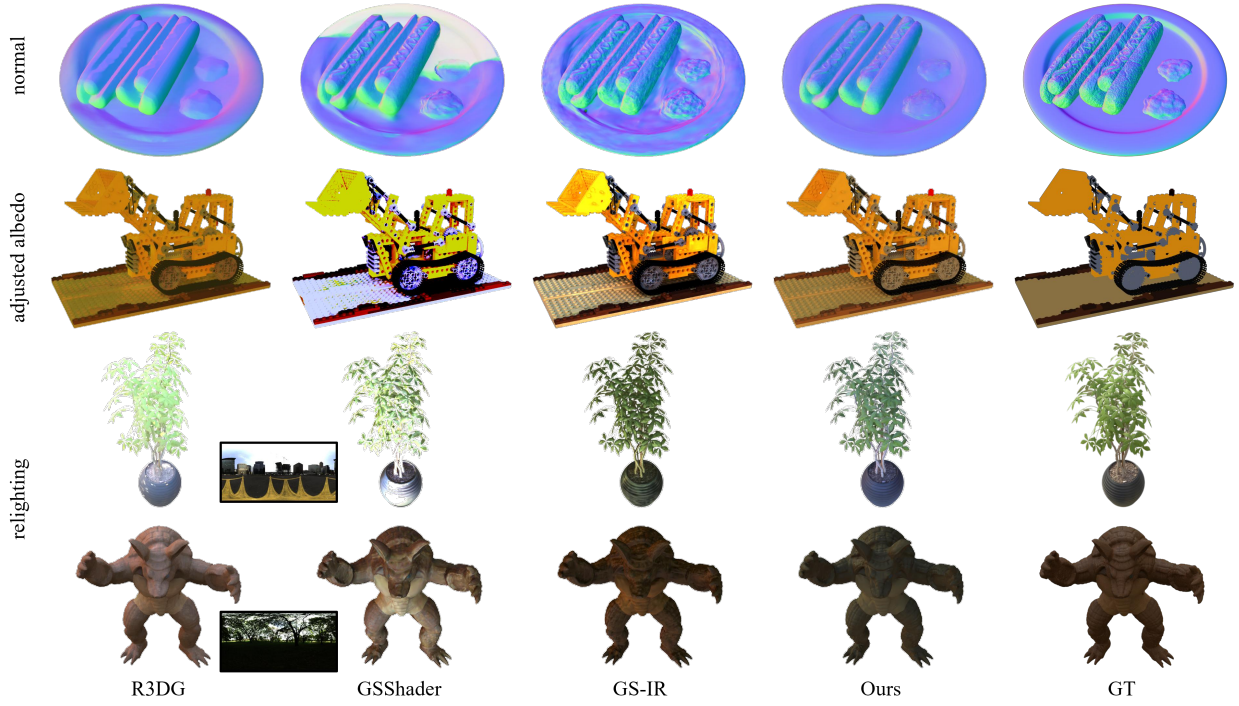


Fig. 6. Qualitative comparison of the material decomposition and relighting on the TensorIR synthetic dataset [4]. The albedo of GSShader [10] is the diffuse term because it establishes the diffuse and specular reflections for each Gaussian, rather than the BRDF parameters. The material decomposition by GS-IR [9] is not smooth enough and exhibits artifacts under relighting. In contrast, our method accurately decomposes materials, is robust to lighting changes, and restores smooth relighting.

Local Geometric Regularization. Unlike GeoGS [30] and ST-4DGS [54], which rely on extra KNN queries to establish neighborhood relations, our method inherently encodes local structure through the stellate construction between anchor and neural Gaussians. Specifically, we treat the local geometric structure of neural Gaussians spawned from the same anchor as a local plane. This assumption is satisfied by constraining the local geometric consistency $\mathcal{L}_{nl}$, which considers both the distances to the viewpoint and the normal directions, and can be formulated as

$$\mathcal{L}_{nl} = \underbrace{\frac{w_n}{N} \sum_{j \in N} \left(\|\mathbf{n}_{0,j}^T \mu_{0,j} - \frac{1}{K-1} \sum_{i \in (1,K)} \mathbf{n}_{i,j}^T \mu_{i,j}\| \right)}_{\text{normal term}} + \underbrace{\frac{w_d}{N \cdot K} \sum_{j \in N} \sum_{i \in K} \|\mu_{i,j} - V_c\|_2}_{\text{distance term}} \quad (12)$$

where $\mathbf{n}_{i,j}$ is the normal of i -th neural Gaussian spawned from j -th anchor, $w_n = 0.05$, and $w_d = 0.01$.

Local Material Regularization. Considering that material should remain consistent within the local plane of real-world scenes, we design a local material regularization to optimize the continuities in the Gaussian material distribution. The local material loss $\mathcal{L}_{ml}$ can be formulated as

$$\mathcal{L}_{ml} = \frac{1}{N \cdot K} \sum_{j \in N} \sum_{i \in K} \|m_{i,j} - \bar{m}_j\|^2, \quad (13)$$

where N is the anchor number, $m_{i,j}$ is one property of materials in the i -th neural Gaussian spawned from the j -th

anchor, $\bar{m}_j$ is the mean of the local material ($m \in \{A, \rho, M\}$).

3.5 Optimization

During Optimization, we adopt a two-stage optimization strategy to learn neural Gaussians. In the first stage, scene geometry is optimized while material and lighting parameters are kept frozen. The loss is defined as

$$\mathcal{L}_1 = \mathcal{L}_c + \lambda_n \mathcal{L}_n + \lambda_{nl} \mathcal{L}_{nl} + \lambda_f \mathcal{L}_f + \lambda_{iif} \mathcal{L}_{iif}, \quad (14)$$

where $\mathcal{L}_c$ is the photometric loss as 3DGS [8], and $\mathcal{L}_n$ is the global pseudo-normal loss, following 2DGS [15]. It computes normals from the rendered depth and compares them against the rendered normals. $\mathcal{L}_f = \|\min(s1, s2, s3)\|_1$ is the flattening loss of scale as NeuSG [55]. Combined with the definition of the normal direction in the shortest axis, this flattening loss helps flatten the local Gaussian distribution and alleviates the impact of the Gaussian volume.

In the second stage, we unfroze the material and lighting parameters and performed PBR rendering. The total loss is formulated as

$$\mathcal{L}_T = \mathcal{L}_c + \lambda_n \mathcal{L}_n + \lambda_{nl} \mathcal{L}_{nl} + \lambda_f \mathcal{L}_f + \lambda_{ml} \mathcal{L}_{ml} + \lambda_{tv} \mathcal{L}_{TV}, \quad (15)$$

where $\mathcal{L}_{TV}$ is the total variation (TV) loss in the rendering materials and the learnable environment image.

4 EXPERIMENTS

4.1 Experimental Setup

Dataset. We evaluate the efficiency of the proposed GANG on two publicly available datasets of complex real-world

TABLE 1

Quantitative comparison of material decomposition, novel view synthesis, and relighting on the TensorIR synthetic dataset [4]. The top block of the table is the comparison PBR methods, and the bottom block is the ablation comparisons of the core components. *local* represents local regularization of geometry and materials. The colors represent first-, second- and third-best. Bold numbers indicate ablation performance equal to or exceeding the full system.

	Normal MAE	Adjusted Albedo			Relighting			NVS			Time
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	
R3DG [11]	5.469	29.492	0.929	0.107	23.591	0.927	0.067	32.724	0.970	0.034	97 <i>m</i>
GSShader [10]	5.820	-	-	-	16.920	0.866	0.107	36.080	0.984	0.026	69 <i>m</i>
GS-IR [9]	5.313	29.887	0.921	0.101	24.134	0.918	0.073	35.277	0.963	0.044	23 <i>m</i>
GS-ID [12]	4.120	30.620	0.940	0.094	-	-	-	36.720	0.977	0.027	-
Ours	3.975	31.871	0.949	0.065	25.751	0.936	0.067	39.790	0.984	0.021	45 <i>m</i>
<i>w/o</i> IIF	4.061	31.726	0.949	0.066	25.540	0.931	0.069	39.496	0.983	0.022	38 <i>m</i>
<i>w/o</i> local	3.896	31.594	0.947	0.067	25.185	0.928	0.071	39.376	0.983	0.024	40 <i>m</i>
<i>w/o</i> $\mathcal{L}_{TV}$	3.991	31.625	0.948	0.067	25.448	0.931	0.072	39.072	0.983	0.022	43 <i>m</i>
<i>w/o</i> $\mathcal{L}_n$	4.242	31.751	0.947	0.067	24.975	0.931	0.072	38.944	0.982	0.023	41 <i>m</i>
<i>w/o</i> SG	4.031	31.548	0.948	0.067	24.814	0.927	0.070	39.693	0.983	0.021	42 <i>m</i>

scenes, including the Mip-NeRF 360 [16] and the Depth T&T composed from the Deep Blending dataset [17] and the Tanks&Templates dataset [18]. However, the absence of true values in real scenes on such datasets makes quantitative comparisons of relighting difficult. Therefore, we conduct quantitative evaluations of novel view synthesis (NVS) on real scenes and provide visual comparisons for relighting. An additional TensorIR synthetic dataset [4] is introduced for quantitative comparisons of NVS, material decomposition, and relighting. To prevent CUDA memory overflow, the Mip-NeRF 360 dataset is downsampled to 1/4 of its original resolution, while others remain unchanged. Real-scene datasets are split 7:1 into training and testing sets, and synthetic datasets follow the default 1:2 configuration.

Metrics. We evaluate rendering quality using PSNR, Structural Similarity Index (SSIM), and perceptual similarity (LPIPS) [56]. The Mean Absolute Error (MAE) is applied to evaluate the deviation between the rendered and ground-truth normal on the synthetic dataset. Furthermore, we also measure storage overhead (*Size (MB)*), training time (*Time (minute)*), and rendering efficiency (*FPS*).

Training Details. The training takes a total of 40K iterations, with 25K for the first stage and 15K for the second stage. Anchor densification concludes in the 18K iteration, and IIF is executed in the first stage after 10K, with 500 iterations executed every 4K. The weight in $\mathcal{L}_{IIF}$ is set to $\lambda_{iif} = 0.04$ before 20K and 0.01 thereafter. The pseudo-normal weight is set to $\lambda_n = 0.05$ after 5k during training. The local weight is $\lambda_{ml} = \lambda_{nl} = 0.01$, and the flatten weight is $\lambda_f = 20$. The TV weight is set to $\lambda_{tv} = 0.01$. All experiments are carried out on a single NVIDIA A800 GPU. Please refer to our Suppl for more details.

Comparison methods. We compare our proposed method with state-of-the-art PBR-based Gaussian methods, including R3DG [11], GSShader [10], and GS-IR [9]. In addition, we evaluate some baseline methods for novel view synthesis (NVS), such as 3DGS [8], 2DGS [15], and Octree-GS [14]. Some results from the original publications of GS-ID [12] are reproduced for comparison.

4.2 Evaluation on Synthetic Dataset

Although our method is primarily designed for complex scene relighting, it also demonstrates strong performance on the synthetic dataset. For fair comparison, the albedo results in Tab.1 and Fig.6 adopt the same adjustment strategy as TensorIR [4], R3DG [11], and GSShader [10]. The color scale ratio between the rendered and ground-truth albedo is computed across RGB channels and used to adjust the rendered albedo. However, ground-truth albedo is not available in real-world scenes. Therefore, unlike previous work [10], [11], the adjustment of the albedo scale is not applied during relighting to ensure fairness and consistency in the process.

As shown in Tab.1, GANG achieves state-of-the-art performance in material decomposition, relighting, and novel view synthesis (NVS), surpassing all existing PBR-based methods. In contrast, other methods show notable degradation relative to baseline approaches in NVS (Tab.2). In addition, GSShader experiences a significant decrease on relighting, while R3DG exhibits a decline on NVS. GANG achieves top performance, only matching R3DG in Relighting (LPIPS) and GSShader in NVS (SSIM), while also attaining a 0.2 improvement in MAE, over 3 dB gain in NVS, 1.2 dB increase in albedo prediction, and surpassing other methods by more than 1.5 dB in relighting. These results demonstrate the robustness and effectiveness of GANG across tasks. It is worth noting that local geometry-aware regularization causes a slight decline in normal prediction, likely because the approximate fitting of curved surfaces is insufficient to capture fine-grained surface details required for accurate normals. However, this component benefits other tasks, enhancing overall performance.

As shown in Fig. 6, the normals rendered by R3DG [11] with learnable parameters appear overly smooth, leading to the loss of fine relighting details. GSShader [10] produces distorted normals, and its relighting results are highly sensitive to specular highlights. GS-IR [9] generates normals with limited accuracy, and its albedo decomposition exhibits poor disentanglement, thereby introducing relighting artifacts. In contrast, our method renders normals that are both smooth



Fig. 7. Qualitative comparison of the novel view synthesis on the Mip-NeRF 360 [16] (bicycle, kitchen) and Deep T&T [17], [18] (train, drjohnson) datasets. Previous PBR-based methods often fail in view synthesis for real-world scenes, resulting in noticeable blurriness or artifacts. Our method not only preserves scene details but also accurately restores challenging areas with weak textures and the sky.

TABLE 2

Quantitative comparison of novel view synthesis on three public datasets. The colors represent first-, second- and third-best for the PBR-based methods. Compared to the baseline, the PBR-based methods exhibit significant quality degradation, while our approach maintains comparable performance.

	Mip-NeRF 360			Deep T&T			TensorIR Synthetic		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
3DGS [8]	29.76	0.891	0.122	26.92	0.881	0.206	41.87	0.989	0.016
2DGS [15]	28.91	0.874	0.155	26.39	0.869	0.237	40.57	0.986	0.022
Octree-GS [14]	29.92	0.890	0.122	27.22	0.885	0.206	41.22	0.988	0.019
R3DG [11]	23.93	0.754	0.256	12.96	0.486	0.646	32.72	0.970	0.034
GSShader [10]	26.70	0.840	0.178	20.54	0.777	0.347	36.08	0.984	0.026
GS-IR [9]	27.00	0.830	0.185	23.76	0.779	0.305	35.28	0.963	0.044
GS-ID [12]	27.19	0.829	0.230	25.92	0.852	0.243	36.72	0.977	0.027
Ours	29.82	0.888	0.129	26.69	0.879	0.223	39.79	0.984	0.021

and accurate, with well-disentangled albedo, resulting in robust relighting that faithfully captures illumination changes.

4.3 Evaluation on Real-world Datasets

In real-world scenes, we conduct both quantitative and qualitative evaluations for novel view synthesis, while rendered geometry and relighting are assessed only qualitatively due to the absence of ground-truth data.

4.3.1 Comparison on Novel View Synthesis

As reported in Tab.2, our GANG method is competitive with the baseline methods (3DGS [8], 2DGS [15], Octree-GS [14]), and achieves the best PSNR, LPIPS and SSIM in the PBR-based methods (R3DG [11], GSShader [10], GS-IR [9], GS-ID [12]). The quality of other PBR-based methods

shows a significant degradation, more than about 2dB in PSNR and 0.05 in SSIM. Specifically, on the Mip-NeRF 360 dataset [16], R3DG [11] experiences a drop of over 6 dB in PSNR, more than 1.3 decrease in SSIM, and an increase of over 0.1 in LPIPS. On the Deep T&T dataset [17], [18], the degradation of R3DG is even more pronounced, with PSNR dropping by over 14 dB, SSIM by more than 0.4, and LPIPS increasing by over 0.4. Similarly, GSShader exhibits significant degradation on the Deep T&T dataset [17], [18], with PSNR decreasing by over 6 dB, SSIM decreasing by more than 0.1, and LPIPS rising by over 0.4. Although our method shows a slight decrease in the quality of view synthesis compared to baseline (Octree-GS [14]), the degradation is minimal. On average, PSNR drops by only about 0.3 dB, SSIM decreases by 0.06, and LPIPS increases by 0.01. This

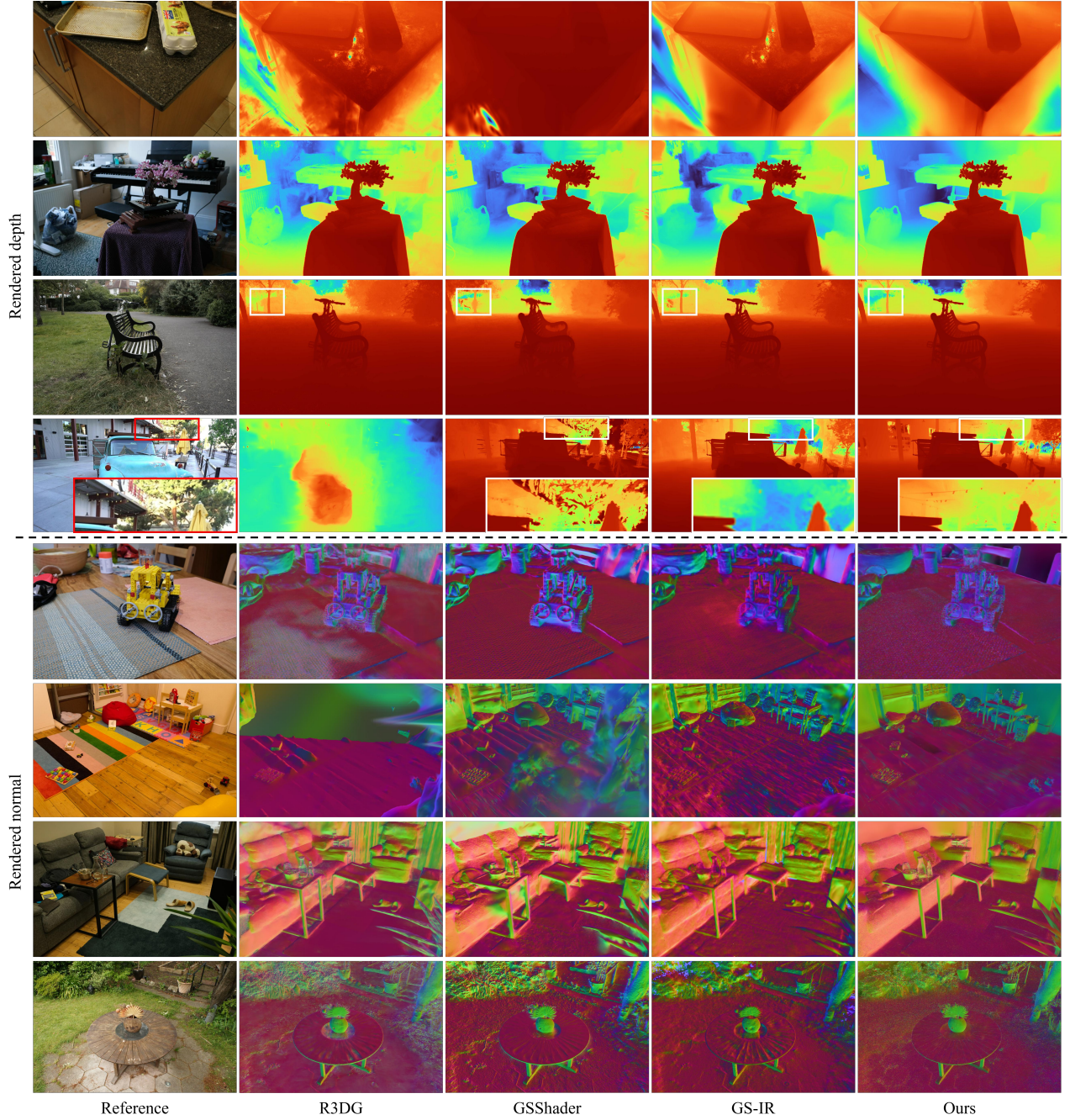


Fig. 8. Qualitative comparison of rendered depth and normal on real-world scenes. Our GANG preserves sharp object geometry and accurately renders thin structures (e.g., small lamps), while simultaneously maintaining high-quality rendering of both curved surfaces and planar regions.

minor drop is primarily due to the inevitable decomposition approximation errors in PBR-based methods, whereas our approach still consistently demonstrates significantly superior performance in complex scenes.

As shown in Fig.7, these PBR-based methods often fail to preserve fine details in view synthesis (e.g. kitchen) and struggle with weakly textured scenes (e.g. drJohnson). R3DG [11] produces severely degraded reconstructions in outdoor real-world scenes (e.g. train), while GSShader [10] generates a large number of floaters (e.g. bicycle and dr-johnson), significantly impairing view quality. GS-IR [9] also suffers from noticeable artifacts in the synthesized views. In

contrast, our method preserves fine texture details, delivers a clearer appearance with fewer artifacts, and further demonstrates state-of-the-art generalizability and stability across diverse real-world scenes.

4.3.2 Comparison on Geometric Reconstruction

To highlight differences between PBR-based methods, we assess geometric reconstruction quality through a comparative analysis of their rendered depth and normal. As shown in Fig.8, R3DG [11] produces an erroneous depth in the outdoor truck scene, while the rendered normals are severely degraded by floaters in the indoor playground scene. More generally, R3DG, GSShader [10], and GS-IR [9] all struggle to

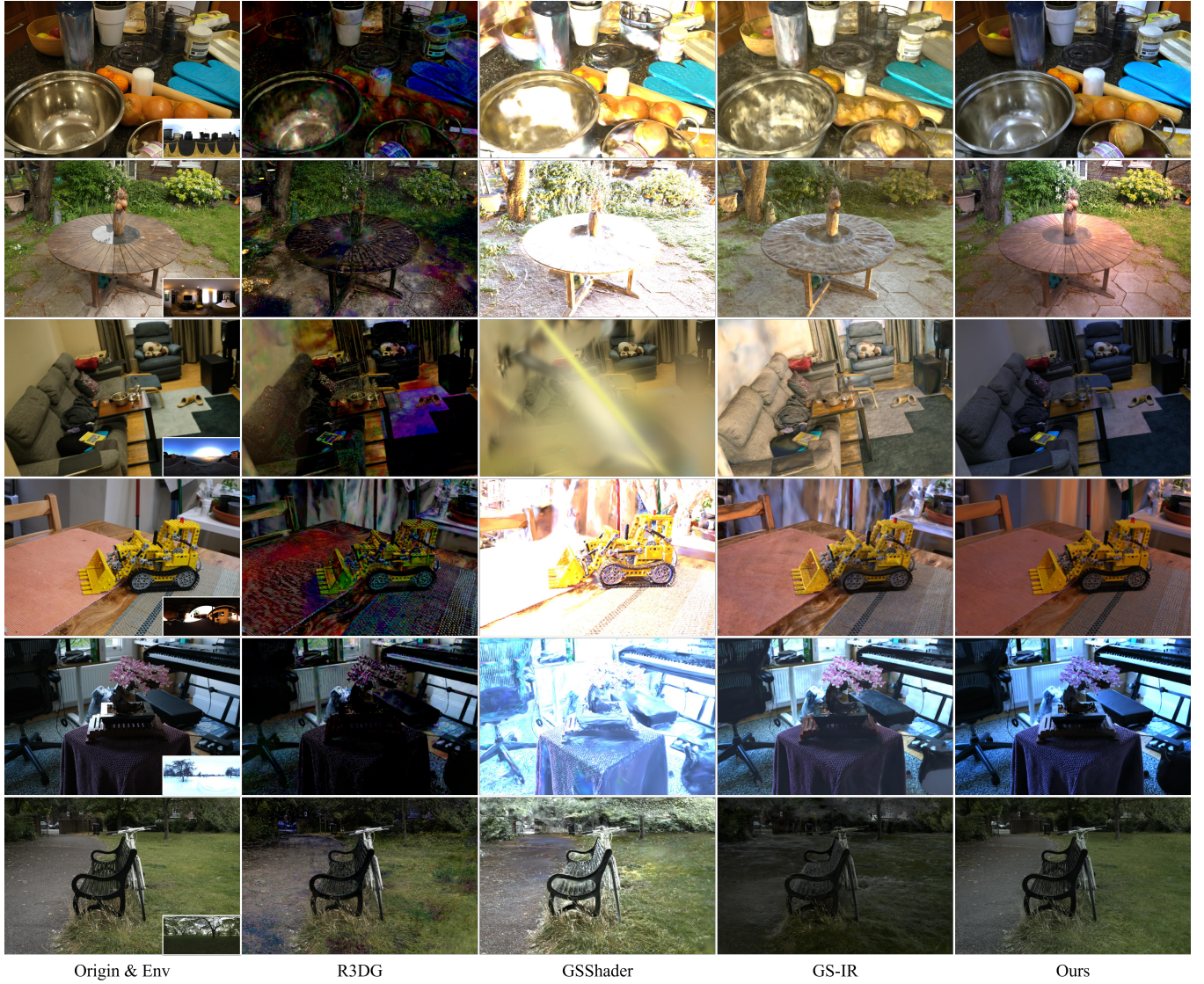


Fig. 9. Qualitative comparison of relighting on real-world scenes. Our method achieves realistic relighting, with rich texture details, clearer appearance, and fewer artifacts.

preserve geometric fidelity. Their depth maps often exhibit distortions, including inaccurate depth ordering and geometry in planar regions, while their normals lack sufficient accuracy, introducing artifacts that compromise relighting and view synthesis. These limitations are particularly pronounced on flat surfaces (e.g., tabletops and walls) and in curved objects, where severe artifacts become evident.

In contrast, our method accurately recovers sharp depth with correct ordering, and generates clean and well-defined normals, free of floaters or distortions. It maintains geometric consistency across both planar surfaces (e.g., tabletops and walls) and curved objects, enabling faithful reconstruction of fine structures and delivering reliable geometry for high-quality relighting and view synthesis.

4.3.3 Comparison on Relighting

As shown in Fig.9, we visualize relighting results for multiple real-world indoor and outdoor scenes under different environment maps. Overall, R3DG [11], GSShader [10], and GS-IR [9] all suffer substantial challenges in maintaining

photorealistic relighting, due to insufficient decoupling of geometry, material, and lighting.

Under the same experimental settings, R3DG [11] suffers from inaccurate geometric reconstruction and material decomposition, leading to relighting outputs that are misaligned with changes in the environment and exhibit pronounced color distortions (discussed in detail in the Suppl). GSShader [10] is highly sensitive to the intensity of environment map lighting, frequently producing overexposed relighting results that are visually inconsistent and fail to match human perceptual expectations, thus reducing reality. Additionally, its geometric reconstruction in the room scene is noticeably distorted, further compromising relighting quality. Similarly, GS-IR [9] is limited by inaccuracies in depth and normals, which result in pronounced artifacts in varying environment maps. These artifacts are particularly severe on planar surfaces, such as tabletops and walls, where accurate geometry is critical for achieving photorealistic relighting.

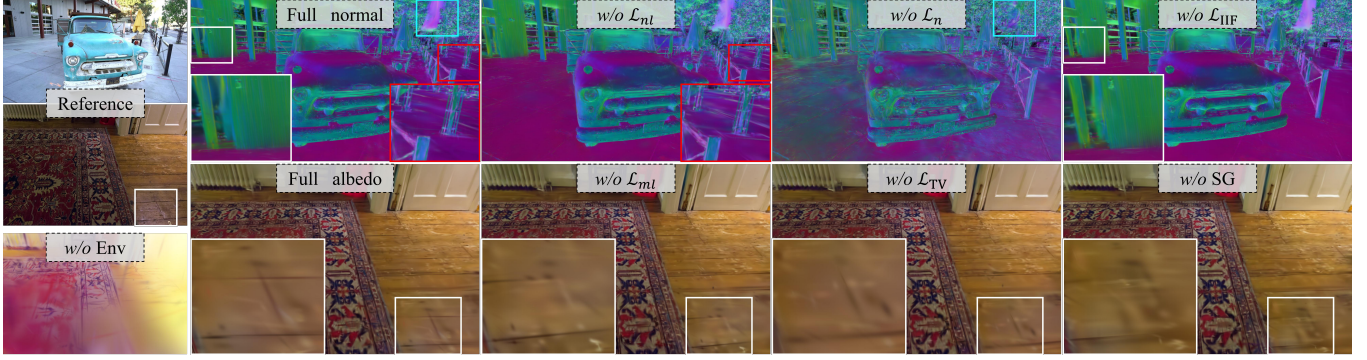


Fig. 10. Visual ablation of key components reveals their impact on normal and albedo fidelity.

TABLE 3

Quantitative comparison of efficiency analyses on three datasets. The methods in the top block are the baseline of the PBR-based methods in the second block. *Ours-C* is the result of the first stage.

	Mip-NeRF 360			Deep T&T			TensorIR Synthetic		
	Time	FPS	Size	Time	FPS	Size	Time	FPS	Size
3DGS [8]	25	111	571	14	153	479	11	273	31
2DGS [15]	25	91	378	14	131	280	7	243	14
Octree-GS [14]	25	168	112	18	245	67	12	373	8
Ours-C	35	171	105	28	228	71	23	311	10
R3GS [11]	188	10	3865	148	18	1728	97	32	287
GSShader [10]	128	12	302	87	20	123	69	40	19
GS-IR [9]	72	41	2451	51	42	1846	23	53	174
Ours	87	55	110	65	58	76	45	71	14

TABLE 4

Ablation study of components in NVS on the Deep T&T dataset [17], [18]. $\mathcal{L}_n$ is the depth-normal loss, $\mathcal{L}_{nl}$ and $\mathcal{L}_{ml}$ are local geometry and material losses, and *Env* is the learnable environment map.

	PSNR	SSIM	LPIPS		PSNR	SSIM	LPIPS
Full	26.69	0.879	0.223	w/o $\mathcal{L}_n$	26.52	0.869	0.236
w/o $\mathcal{L}_{IIF}$	26.43	0.875	0.225	w/o $\mathcal{L}_{ml}$	26.31	0.875	0.225
w/o $\mathcal{L}_{nl}$	26.26	0.875	0.228	w/o SG	26.58	0.877	0.223
w/o $\mathcal{L}_{TV}$	26.61	0.878	0.224	w/o Env	7.06	0.497	0.579

Our GANG maintains accurate geometric reconstruction and precise normals on both planar and curved surfaces. It effectively decouples material properties and lighting, enabling relighting results to faithfully respond to changes in environment maps with rich texture details, visually coherent illumination, and minimal artifacts, even in complex real-world indoor and outdoor scenes. Consequently, GANG consistently delivers high-quality photorealistic relighting, demonstrating its robustness and effectiveness on diverse real-world scenes.

4.4 Efficiency Analysis

As reported in the Tab. 3, we evaluate the efficiency of our two-stage approach in terms of storage and rendering speed. It shows that our first stage incurs only 10 minutes of extra training due to IIF and local regularization, while maintaining a rendering speed and model size comparable to the baselines. On the Mip-NeRF 360 [16] dataset, our method achieves the best efficiency, maintaining the same FPS as the baseline while reducing the model size by about

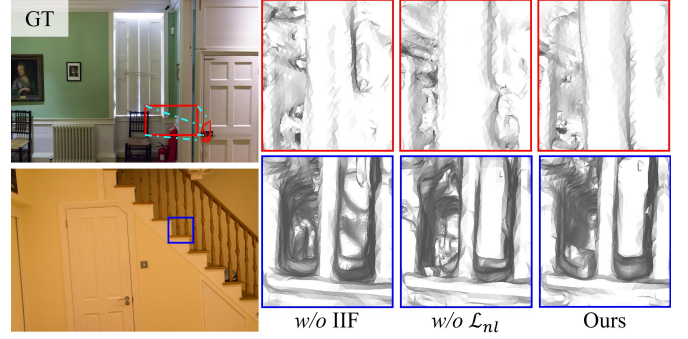


Fig. 11. Visual ablation of the core components in mesh. The mesh is reconstructed from the rendered depth with the Poisson method [57].

7MB. On Deep T&T [17], [18] and TensorIR synthetic [4], the model size grows slightly (4–2 MB) with a moderate decrease in FPS (10–15%). However, our method remains consistently more efficient than 3DGS and 2DGS, with only a slight disadvantage compared to Octree-GS.

In contrast, PBR-based methods generally impose substantial computational overhead and require significantly larger model sizes. However, our GANG offers an advantageous trade-off between rendering speed and memory footprint. Specifically, compared to R3DG [11], GANG cuts training time by half, reduces storage by 95%, and delivers 3× faster rendering. Against GSShader [10], it requires only about 80% of the training time, halves storage, and doubles rendering speed. Compared to GS-IR [9], GANG requires about 15 extra minutes of training due to additional 10K iterations, extra regularization, and SG-based hybrid lighting. GANG still achieves far better efficiency since the storage overhead is only about 1/20 and rendering speed is substantially higher. Overall, GANG achieves photorealistic rendering efficiently, striking a favorable balance among training time, rendering speed, and storage requirements, and clearly outperforming existing PBR-based methods.

4.5 Ablation Study

To validate the effectiveness of each component, we perform an ablation study of the proposed method on the Deep T&T dataset (Tab. 4) and the TensorIR synthetic dataset (Tab. 1), where each component is removed individually to evaluate its contribution. Furthermore, we visualize the impact of key



Fig. 12. Application of relighting and material editing. Relighting from a single moving global light with an unknown location (2nd row). By operating on Gaussian primitives, we can easily achieve controllable editing of BRDF parameters (4th and 5th columns).

components on geometry-material decomposition (Fig.10) and mesh reconstruction (Fig.11). Obviously, removing any component will lead to a decrease in the PSNR, indicating that each component contributes to the entire system and is conducive to the decomposition of scene content.

Geometry. All proposed geometric optimization strategies consistently demonstrate clear benefits to the overall quality of the reconstruction. Specifically, the Implicit Indicator Function (IIF) adaptively aligns the geometry with the underlying scene structure, thereby improving compactness and producing more faithful surface representations. Local regularization further refines geometric reconstruction by enforcing spatial coherence, effectively enhancing smoothness and stability in complex regions. As demonstrated in Fig. 11 and Fig. 10, the absence of any single component compromises the integrity of the reconstructed geometry, producing distortions and inconsistencies that highlight the essential role of each module in maintaining coherent surface structures. Moreover, the introduced pseudo-normal loss $\mathcal{L}_n$ plays a crucial role in enforcing global consistency between depth and normal, ensuring coherent scene geometry and improving overall fidelity.

Material. Disentanglement is reinforced by the proposed consistency and regularization losses, which ensure accurate separation of material properties. Specifically, removing the local material consistency loss $\mathcal{L}_{ml}$ compromises decoupling, producing visible albedo artifacts (Fig. 10) and resulting in a 0.38 dB PSNR drop (Tab. 4), demonstrating its importance for capturing fine material details. In addition, total variation (TV) regularization reduces residual albedo noise and enforces spatial smoothness, further stabilizing the decomposition and improving the realism of the reconstructed materials.

Hybrid Lighting. Spherical Gaussians (SGs) enhance the modeling of imprecise lighting in the global environment map, which is crucial for accurate material decomposition. Without SGs, material separation becomes insufficient (Tab. 1, Fig. 10) and rendering quality is significantly degraded (Tab. 4). In our system, most of the lighting energy is captured by the learnable environment map, representing the global diffuse component. Omitting the environment map and using fewer SGs to model scene illumination leads to substantial deterioration in both material decomposition and rendering fidelity, as evidenced by a PSNR drop of approximately 20 dB on the Deep T&T dataset.

4.6 Applications and Limitations

We demonstrate further applications of GANG in Fig.12. Its parameterized lighting enables precise and flexible control of both complex multi-light setups and single-light sources, supporting high-fidelity relighting. Leveraging SAGS [58] for the semantic segmentation of Gaussian primitives, GANG allows fine-grained material editing of material parameters and appearance at the primitive level, including selective modification of albedo, roughness, metallic, or color, without affecting other regions. These capabilities highlight the strength of GANG in interactive scene manipulation, providing controllable and photorealistic results for AR/VR, content creation, and visual effects.

Despite its advantages, our method has several limitations. We only model the global diffuse reflection and do not precisely model the specular reflection and mutual reflection. One solution is to introduce time-consuming ray tracing, such as IRGS [45], but it may not be suitable for complex scenes and breaks the balance between quality and efficiency. Extending GANG to support accurate light interactions in complex scenes, including soft shadows, specular highlights, and caustics, remains a promising direction. We leave these improvements for future work.

Please refer to the Suppl and Video for more results.

5 CONCLUSION

This paper presents Geometrically-Aligned Neural Gaussians (GANG) for complex scene relighting. By leveraging adaptive geometric alignment and efficient design choices, GANG delivers realistic and efficient relighting, enhancing the fidelity and compactness of Gaussian representations while enabling accurate material decomposition with minimal memory overhead. Its hybrid lighting model further facilitates precise decoupling of geometry, materials, and illumination under unknown lighting conditions. Furthermore, GANG supports flexible scene manipulation, including controllable lighting and material editing, while producing photorealistic rendering efficiently. Extensive experiments on synthetic and real-world datasets demonstrate its superiority over existing PBR-based methods in terms of relighting quality, novel view synthesis, and rendering efficiency. We envision that GANG provides a robust and versatile framework that can drive further research and practical applications in immersive AR/VR, content creation, and interactive scene editing.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China(62572060).

REFERENCES

- [1] Z. Wang, J. Philion, S. Fidler, and J. Kautz, "Learning indoor inverse rendering with 3d spatially-varying lighting," in *IEEE ICCV*, 2021, pp. 12518–12527.
- [2] H. Kim, M. Jang, W. Yoon, J. Lee, D. Na, and S. Woo, "Switch-light: Co-design of physics-driven architecture and pre-training framework for human portrait relighting," in *IEEE CVPR*, 2024, pp. 25 096–25 106.
- [3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, p. 99–106, 2021.
- [4] H. Jin, I. Liu, P. Xu, X. Zhang, S. Han, S. Bi, X. Zhou, Z. Xu, and H. Su, "Tensor: Tensorial inverse rendering," in *IEEE CVPR*, 2023, pp. 165–174.
- [5] J. Ling, R. Yu, F. Xu, C. Du, and S. Zhao, "Nerf as a non-distant environment emitter in physics-based inverse rendering," in *ACM SIGGRAPH*, 2024.
- [6] Z. Wang, T. Shen, J. Gao, S. Huang, J. Munkberg, J. Hasselgren, Z. Gojcic, W. Chen, and S. Fidler, "Neural fields meet explicit geometric representations for inverse rendering of urban scenes," in *IEEE CVPR*, 2023, pp. 8370–8380.
- [7] Z. Li, L. Wang, M. Cheng, C. Pan, and J. Yang, "Multi-view inverse rendering for large-scale real-world indoor scenes," in *IEEE CVPR*, 2023, pp. 12 499–12 509.
- [8] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM TOG*, vol. 42, no. 4, 2023.
- [9] Z. Liang, Q. Zhang, Y. Feng, Y. Shan, and K. Jia, "Gs-ir: 3d gaussian splatting for inverse rendering," in *IEEE CVPR*, 2024, pp. 21 644–21 653.
- [10] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, and Y. Ma, "Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces," in *IEEE CVPR*, 2024, pp. 5322–5332.
- [11] J. Gao, C. Gu, Y. Lin, Z. Li, H. Zhu, X. Cao, L. Zhang, and Y. Yao, "Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing," in *ECCV*. Springer, 2025, pp. 73–89.
- [12] K. Du, Z. Liang, and Z. Wang, "Gs-id: Illumination decomposition on gaussian splatting via diffusion prior and parametric light source optimization," *arXiv preprint arXiv:2408.08524*, 2024.
- [13] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," in *IEEE CVPR*, 2024, pp. 20 654–20 664.
- [14] K. Ren, L. Jiang, T. Lu, M. Yu, L. Xu, Z. Ni, and B. Dai, "Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians," *arXiv preprint arXiv:2403.17898*, 2024.
- [15] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH*, 2024.
- [16] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *IEEE ICCV*, 2021, pp. 5835–5844.
- [17] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM TOG*, vol. 37, no. 6, 2018.
- [18] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: benchmarking large-scale scene reconstruction," *ACM TOG*, vol. 36, no. 4, 2017.
- [19] M. M. Johari, Y. Lepoittevin, and F. Fleuret, "Geonerf: Generalizing nerf with geometry priors," in *IEEE CVPR*, 2022, pp. 18 344–18 347.
- [20] G.-W. Yang, W.-Y. Zhou, H.-Y. Peng, D. Liang, T.-J. Mu, and S.-M. Hu, "Recursive-nerf: An efficient and dynamically growing nerf," *IEEE TVCG*, vol. 29, no. 12, pp. 5124–5136, 2023.
- [21] R. A. Rosu and S. Behnke, "Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices," in *IEEE CVPR*, 2023, pp. 8466–8475.
- [22] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *IEEE CVPR*, 2022, pp. 5491–5500.
- [23] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *IEEE CVPR*, 2023, pp. 12 479–12 488.
- [24] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM TOG*, vol. 41, no. 4, 2022.
- [25] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3d gaussian splatting as a new era: A survey," *IEEE TVCG*, vol. 31, no. 8, pp. 4429–4449, 2025.
- [26] P. Dai, J. Xu, W. Xie, X. Liu, H. Wang, and W. Xu, "High-quality surface reconstruction using gaussian surfels," in *ACM SIGGRAPH*, 2024.
- [27] Z. Yu, T. Sattler, and A. Geiger, "Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes," *ACM TOG*, vol. 43, no. 6, 2024.
- [28] D. Li, S.-S. Huang, and H. Huang, "Mpgs: Multi-plane gaussian splatting for compact scenes rendering," *IEEE TVCG*, vol. 31, no. 5, pp. 3256–3266, 2025.
- [29] M. Yu, T. Lu, L. Xu, L. Jiang, Y. Xiangli, and B. Dai, "Gsdf: 3dgs meets sdf for improved neural rendering and reconstruction," *NeurIPS*, pp. 129 507–129 530, 2024.
- [30] Y. Li, C. Lyu, Y. Di, G. Zhai, G. H. Lee, and F. Tombari, "Geogaussian: Geometry-aware gaussian splatting for scene rendering," in *ECCV*. Springer, 2025, pp. 441–457.
- [31] Y. Dong, G. Chen, P. Peers, J. Zhang, and X. Tong, "Appearance-from-motion: recovering spatially varying surface reflectance under unknown lighting," *ACM TOG*, vol. 33, no. 6, 2014.
- [32] S. Bi, Z. Xu, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi, "Deep reflectance volumes: Relightable reconstructions from multi-view photometric images," in *ECCV*. Springer, 2020, pp. 294–311.
- [33] C. Zeng, G. Chen, Y. Dong, P. Peers, H. Wu, and X. Tong, "Relighting neural radiance fields with shadow and highlight hints," in *ACM SIGGRAPH*, 2023, pp. 1–11.
- [34] M. Cabral, N. Bonneel, S. Lefebvre, and G. Drettakis, "Relighting photographs of tree canopies," *IEEE TVCG*, vol. 17, no. 10, pp. 1459–1474, 2011.
- [35] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, "Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting," in *IEEE CVPR*, 2021, pp. 5449–5458.
- [36] H.-G. Chung, S. Choi, and S.-H. Baek, "Differentiable point-based inverse rendering," in *IEEE CVPR*, 2024, pp. 4399–4408.
- [37] J. Zhang, Y. Yao, S. Li, J. Liu, T. Fang, D. McKinnon, Y. Tsin, and L. Quan, "Neilf++: Inter-reflectable light fields for geometry and material estimation," in *IEEE ICCV*, 2023, pp. 3578–3587.
- [38] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *IEEE CVPR*, 2021, pp. 7491–7500.
- [39] Z. Bi, Y. Zeng, C. Zeng, F. Pei, X. Feng, K. Zhou, and H. Wu, "Gs3: Efficient relighting with triple gaussian splatting," in *ACM SIGGRAPH Asia*, 2024.
- [40] Y. He, Y. Wang, and X. Yang, "Gs-phong: Meta-learned 3d gaussians for relightable novel view synthesis," *arXiv preprint arXiv:2405.20791*, 2024.
- [41] J. F. Blinn, "Models of light reflection for computer synthesized pictures," in *ACM SIGGRAPH*, 1977, pp. 192–198.
- [42] Y. Guo, Y. Bai, L. Hu, Z. Guo, M. Liu, Y. Cai, T. Huang, and L. Ma, "Prtgs: Precomputed radiance transfer of gaussian splats for real-time high-quality relighting," in *ACM MM*, 2024, p. 5112–5120.
- [43] S. Lai, L. Huang, J. Guo, K. Cheng, B. Pan, X. Long, J. Lyu, C. Lv, and Y. Guo, "Glossygs: Inverse rendering of glossy objects with 3d gaussian splatting," *IEEE TVCG*, vol. 31, no. 10, pp. 7478–7491, 2025.
- [44] N. Moenne-Loccoz, A. Mirzaei, O. Perel, R. de Lutio, J. Martinez Esturo, G. State, S. Fidler, N. Sharp, and Z. Gojcic, "3d gaussian ray tracing: Fast tracing of particle scenes," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–19, 2024.
- [45] C. Gu, X. Wei, Z. Zeng, Y. Yao, and L. Zhang, "Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing," in *IEEE CVPR*, 2025.
- [46] T. Xie, X. Chen, Z. Xu, Y. Xie, Y. Jin, Y. Shen, S. Peng, H. Bao, and X. Zhou, "Envgs: Modeling view-dependent appearance with environment gaussian," in *IEEE CVPR*, 2025.
- [47] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE CVPR*, 2016, pp. 4104–4113.

- [48] R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," *ACM TOG*, vol. 1, no. 1, p. 7–24, 1982.
- [49] S. Peng, C. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger, "Shape as points: A differentiable poisson solver," *NeurIPS*, vol. 34, pp. 13 032–13 044, 2021.
- [50] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods: fundamentals in single domains*. Springer, 2006.
- [51] B. Karis and E. Games, "Real shading in unreal engine 4," *Proc. Physically Based Shading Theory Practice*, vol. 4, no. 3, p. 1, 2013.
- [52] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-frequency rendering of dynamic, spatially-varying reflectance," in *ACM SIGGRAPH Asia*, 2009.
- [53] L.-Q. Yan, Y. Zhou, K. Xu, and R. Wang, "Accurate translucent material rendering under spherical gaussian lights," in *Computer Graphics Forum*, vol. 31, no. 7. Wiley, 2012, pp. 2267–2276.
- [54] D. Li, S.-S. Huang, Z. Lu, X. Duan, and H. Huang, "St-4dgs: Spatial-temporally consistent 4d gaussian splatting for efficient dynamic scene rendering," in *ACM SIGGRAPH*, 2024.
- [55] H. Chen, C. Li, and G. H. Lee, "Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance," *arXiv preprint arXiv:2312.00846*, 2023.
- [56] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE CVPR*, 2018, pp. 586–595.
- [57] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM TOG*, vol. 32, no. 3, pp. 1–13, 2013.
- [58] X. Hu, Y. Wang, L. Fan, J. Fan, J. Peng, Z. Lei, Q. Li, and Z. Zhang, "Semantic anything in 3d gaussians," *arXiv preprint arXiv:2401.17857*, 2024.