LumiMotion: Improving Gaussian Relighting with Scene Dynamics

Joanna Kaleta[1,2]*, Piotr Wójcik[3,4,5]*, Kacper Marzol[6]
Tomasz Trzciński[1,7], Kacper Kania[1], Marek Kowalski[8]

[1]Warsaw University of Technology [2]Sano Centre for Computational Medicine
[3]Inst. for Biomedical Informatics, Univ. Hospital of Cologne [4]University of Cologne
[5]CMMC Cologne [6]Jagiellonian University [7]IDEAS Research Institute [8]Microsoft

Abstract

In 3D reconstruction, the problem of inverse rendering, namely recovering the illumination of the scene and the material properties, is fundamental. Existing Gaussian Splatting-based methods primarily target static scenes and often assume simplified or moderate lighting to avoid entangling shadows with surface appearance. This limits their ability to accurately separate lighting effects from material properties, particularly in real-world conditions. We address this limitation by leveraging dynamic elements—regions of the scene that undergo motion—as a supervisory signal for inverse rendering. Motion reveals the same surfaces under varying lighting conditions, providing stronger cues for disentangling material and illumination. This thesis is supported by our experimental results which show we improve LPIPS by 23% for albedo estimation and by 15% for scene relighting relative to next-best baseline. To this end, we introduce ***LumiMotion****, the first Gaussian-based approach that leverages dynamics for inverse rendering and operates in arbitrary dynamic scenes. Our method learns a dynamic 2D Gaussian Splatting representation that employs a set of novel constraints which encourage the dynamic regions of the scene to deform, while keeping static regions stable. As we demonstrate, this separation is crucial for correct optimization of the albedo. Finally, we release a new synthetic benchmark comprising five scenes under four lighting conditions, each in both static and dynamic variants, for the first time enabling systematic evaluation of inverse rendering methods in dynamic environments and challenging lighting. Link to project page in footnote[1].*

1. Introduction

Inverse rendering - the task of recovering geometry, material properties, and illumination from a set of images of a scene - is a fundamental challenge in computer vision and graphics [1]. Although methods like Neural Radiance Fields [32] and Gaussian Splatting [22] allow for accurately

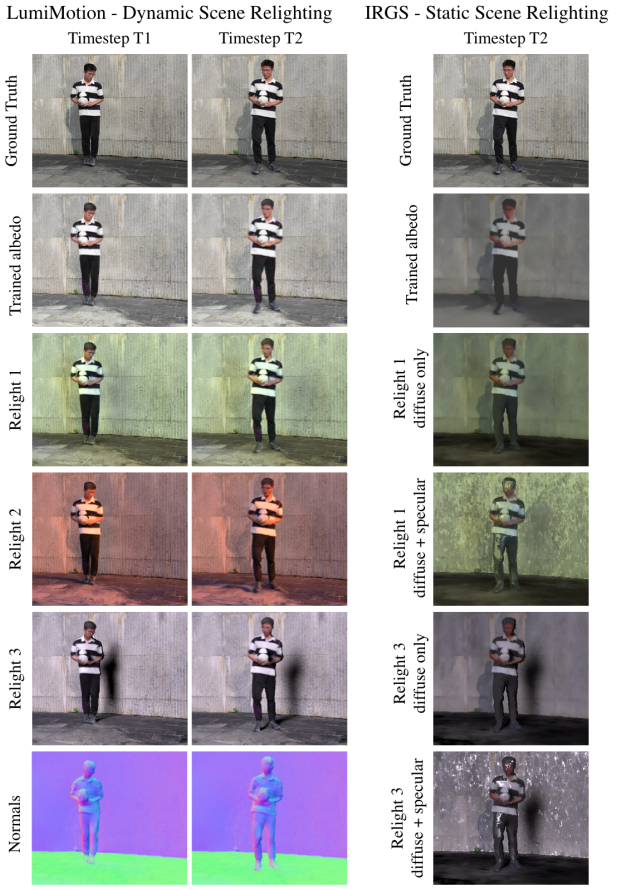

Figure 1. **Qualitative results for LumiMotion and IRGS [12] on a real-world dataset.** Note that thanks to modeling the scene dynamics **LumiMotion** can successfully remove shadows from the albedo, leading to improved relighting. IRGS struggles to correctly remove shadows from the albedo, and the specular component is not properly optimized. To allow for comparison of **LumiMotion** and IRGS in absence of the latter's specularity estimation errors we show some of IRGS results with diffuse component only.

*Equal contribution
[1]https://joaxkal.github.io/LumiMotion/

representing scene geometry, they represent the color of the reconstructed scene as it was observed, including shadows and other illumination-dependent effects. This conflation of lighting and material means the scene cannot be rendered under a different illumination (it cannot be relit). This limits the applicability of those reconstructions in fields such as gaming or film making, which require control over illumination. Recent works have introduced physically-motivated decomposition methods to address this limitation [7, 10, 12], though, as shown in Fig. 1, such approaches may struggle in real-world scenes with significant direct illumination.

We hypothesize that the reason existing methods struggle is that they operate on static scenes, where the content is not moving. This lack of motion means that it is difficult to discern between the intrinsic color of objects and the observed color, which is a function of the incident light. For example, when viewing images of static scenes, it may be difficult to determine whether a certain part of the scene is darker because a shadow is cast on it or because the material itself is dark. Thus, we propose **LumiMotion**, the first Gaussian-based method to perform inverse rendering of arbitrary dynamic scenes. Based solely on a video sequence of a scene with dynamic elements, our approach reconstructs the scene's geometry, as well as its material properties and illumination. The lighting conditions are assumed to be static, not known a priori and are inferred as part of the training. We leverage the scene dynamics to improve the quality of the estimated illumination and albedo, thereby enhancing the ability to render those scenes under novel illumination conditions.

LumiMotion operates in two stages. In the first stage, we jointly learn the static scene geometry and a neural network that models the dynamics of the scene including changes to shape and color of objects in the scene. The geometry is modeled with 2D Gaussian Splats [15] that represent the surfaces in the scene with a collection of flat disks. A key benefit of this approach over 3D Gaussian Splatting or Neural Radiance Fields is that the 2D Gaussians have normal vectors associated with them and those are necessary for our second stage. In the second stage we freeze both the geometry and the neural network and proceed to inverse rendering. At this point we move away from rendering the Gaussian color directly and instead we compute the color of each Gaussian as a function of its material and the incident light that is computed via ray tracing. This allows for jointly optimizing the material (described jointly by albedo and roughness) and the illumination.

Once we have those parameters, we can relight the scene with novel illumination or use the estimated illumination to relight other scenes. We demonstrate those abilities qualitatively on real data and quantitatively on a new synthetic dataset we created, where the ground truth illumination and material are known. Please see Fig. 1 for a teaser of the results.

Table 1. **Overview of the representative relightable methods and datasets.** Existing methods often target static scenes or focus on human avatars, assume known lights or require unnatural lighting setup. In contrast, **LumiMotion** uniquely performs inverse rendering of generic dynamic scenes under unknown natural lighting. We also show that **none** of the available datasets with known GT lighting meet the requirements for the tackled setup, which we address with our newly released dataset.

Relightable methods				
Method Name	Supports Dynamics	Object-Agnostic	Unknown Train Light	Natural Train Light
GS^3 [2]	✗	✓	✗	✗
R-3DGS [10], GI-GS [7], IR-GS [12]	✗	✓	✓	✓
TensorIR [19], NeRFactor [51]	✗	✓	✓	✓
Relightable Neural Actor [29]	✓	✗	✗	✓
Gaussian Codec Avatars [35] & [38]	✓	✗	✗	✗
IntrinsicAvatar [37]	✓	✗	✓	✓
Relightable [...] Neural Avatars [48]	✓	✗	✓	✓
LumiMotion	✓	✓	✓	✓

Relightable datasets				
Dataset Name	Domain	Scene Type	Train Light Type	Static-Dynamic Reference
OLAT [27]	generic	static	OLAT	✗
TensoIR [19]	generic	static	natural	✗
Synthetic4Relight [52]	generic	static	natural	✗
Stanford-ORB [23]	generic	static	natural	✗
RANA [17]	avatars	dynamic	natural	✗
Codec Avatar Studio [31]	avatars	both	multi-light	✗
LumiMotion	generic	both	natural	✓

Our contributions can be summarized as follows:

- First Gaussian-based approach for inverse rendering in arbitrary dynamic scenes, achieving better separation of material and illumination by leveraging scene dynamics as a supervisory signal.
- A novel set of constraints on the deformation network that allows for better separation of static and dynamic parts of the scene and for improved modeling of scene geometry in time.
- A new synthetic dataset allowing for comparing the performance of inverse rendering approaches for static and dynamic scenes.

2. Related Work

Novel View Synthesis. Novel view synthesis aims to generate images of a scene from novel viewpoints using a limited set of input observations. Neural Radiance Fields (NeRF) [32] marked a breakthrough in novel view synthesis, offering high quality of view-dependent renderings. Despite their strengths, NeRFs suffer from slow training and rendering. Several approaches were made to overcome this limitation [5, 11, 33]. Recently, 3D Gaussian Splatting [22] proposed representing scenes with a gaussian point cloud rendered with a tile-based rasterizer, achieving state-of-the-art results with lower computational cost. A series of subsequent studies tackled a range of challenges, including editability [8, 36, 41], realistic modeling of conditioned appearance [21, 42], modeling of motion and scene dynam-

ics [16, 28, 40, 44] or improved geometry reconstruction [9, 13, 15]. Utilizing disc-like, flat Gaussian primitives in 2DGS [15] improved surface reconstruction quality while well-defined ray-splat intersection provided a straightforward foundation for extending the method to various tasks.

Inverse Rendering. Inverse rendering seeks to decompose a scene into its geometry, material properties, and lighting effects based on input images. A key challenge lies in the inherent ambiguity between observed photometric effects and the true material and lighting parameters, often resulting in multiple plausible solutions to the rendering equation. At the same time, modeling physical conditions is crucial for relighting optimized scenes. To this end, several methods have incorporated spatially-varying BRDF parameters into neural representations [3, 6, 43, 45, 49]. For example, NERD leverages multiview supervision under varying illumination and employs a path-traced differentiable renderer. TensoIR [19] utilizes TensorRF [5] tri-plane representation for efficient computation of visibility and indirect lighting by ray-tracing.

More recent methods explore inverse rendering using Gaussian Splatting, aiming to optimize material properties for each Gaussian. Some approaches focus solely on modeling reflective properties [18, 46, 47], while others utilize the full rendering equation, which requires accurate visibility estimation. In GS³ [2], occlusions are handled via shadow splatting, allowing for fast relighting but limited to OLAT-type training data. R3DG [10] employs ray tracing for visibility estimation and baking, computing shading individually per Gaussian. GI-GS [7], IRGS [12], and GS-IR [25] adopt a deferred shading approach: they first rasterize maps into a G-buffer, then apply the full rendering equation for shading. Notably, IRGS [12] leverages 2DGS with a differentiable ray tracer and addresses the computational overhead via stratified relighting in each iteration. Note that all of the above models focus on static scenes and thus do not leverage information from the scene motion. Other methods [14, 24, 35, 38, 48, 53], although designed to handle relightable dynamics, are constrained by human-pose priors and typically require either known training light conditions or access to large datasets with diverse lighting. In contrast to these approaches, our method learns lighting from a scene without the need to observe it under multiple illuminations and with no assumption on object category. Tab. 1 summarizes the differences between the discussed methods and available datasets.

3. Method

LumiMotion operates in a two-stage setup. In the first stage, we perform a 3D reconstruction of the scene. This process combines creation of Gaussians with learning a deformation network that models the scene's dynamics. In the second stage, we take the geometry and deformation learned in the first stage, which are now frozen, and jointly optimize the illumination and material parameters of the scene. An overview of our method is presented in Fig. 2.

3.1. Preliminaries

2D Gaussian Splatting (2DGS). 2DGS [15] is particularly well-suited for view synthesis and relighting, thanks to its ability to produce smooth and accurate surface normals. 2DGS represents a scene as a collection of flat 2D Gaussians embedded in 3D space. Each Gaussian is defined by a central point $\boldsymbol{\mu} \in \mathbb{R}^3$, two tangential vectors $\mathbf{t}_u, \mathbf{t}_v \in \mathbb{R}^3$ which define the normal, and two scaling factors s_u, s_v that control the spread in the local tangent plane. A 2D Gaussian is parameterized as follows:

$$G(u, v) = \boldsymbol{\mu} + s_u \mathbf{t}_u u + s_v \mathbf{t}_v v. \tag{1}$$

For rendering, each disk is projected to screen space via a homogeneous transformation. An explicit ray-splat intersection computes the local (u, v) coordinates for each pixel, and the projected value is evaluated using a screen-space filter. Gaussians are alpha-composited front-to-back based on depth ordering to accumulate color per pixel. For more implementation details, we refer the reader to [15].

Rendering Equation. In Stage 2 of our pipeline, we relight the reconstructed scene under a given illumination. In computer graphics, the appearance of the surface is governed by the classical rendering equation [20], which models the interaction between the properties of light and material:

$$\begin{aligned} L_o(\mathbf{x}, \omega_o) = \int_\Omega f_r(\omega_i, \omega_o, \mathbf{x}) \big(V(\omega_i, \mathbf{x}) L_{\text{env}}(\omega_i) \\ + L_{\text{ind}}(\mathbf{x}, \omega_i) \big) (\omega_i \cdot \mathbf{n}) \, \mathrm{d}\omega_i, \end{aligned} \tag{2}$$

where $L_o(\mathbf{x}, \omega_o)$ denotes the outgoing radiance at point $\mathbf{x}$ in direction ω_o, $f_r(\omega_i, \omega_o, \mathbf{x})$ represents the bidirectional reflectance distribution function (BRDF), $L_{\text{env}}(\omega_i)$ is the incoming environment radiance from direction ω_i, $\mathbf{n}$ is the surface normal at $\mathbf{x}$, $L_{\text{ind}}(\mathbf{x}, \omega_i)$ is the indirect illumination term, $V(\omega_i, \mathbf{x})$ is the visibility of environment light from the point $\mathbf{x}$ in the direction ω_i, and Ω denotes the hemisphere oriented around $\mathbf{n}$. The BRDF describes the amount of light reflected from direction ω_i towards ω_o for the material at position $\mathbf{x}$.

Most inverse rendering tasks aim to decompose and reconstruct scene components by estimating material properties (f_r) and environment illumination (L_{env}) from observed images. Following the simplified Disney BRDF model [4], we parametrize bidirectional reflectance distribution function with albedo ρ and roughness α. While those values change throughout the scene and thus depend on the position $\mathbf{x}$ we omit the position here for brevity. The final BRDF which combines diffuse and specular terms is:

$$f_r(\omega_i, \omega_o) = \frac{\rho}{\pi} + \frac{D(\rho; \alpha) F(\omega_o, \mathbf{h}; \rho) G(\omega_i, \omega_o, \mathbf{h}; \alpha)}{4(\omega_i \cdot \mathbf{n})(\omega_o \cdot \mathbf{n})}, \tag{3}$$

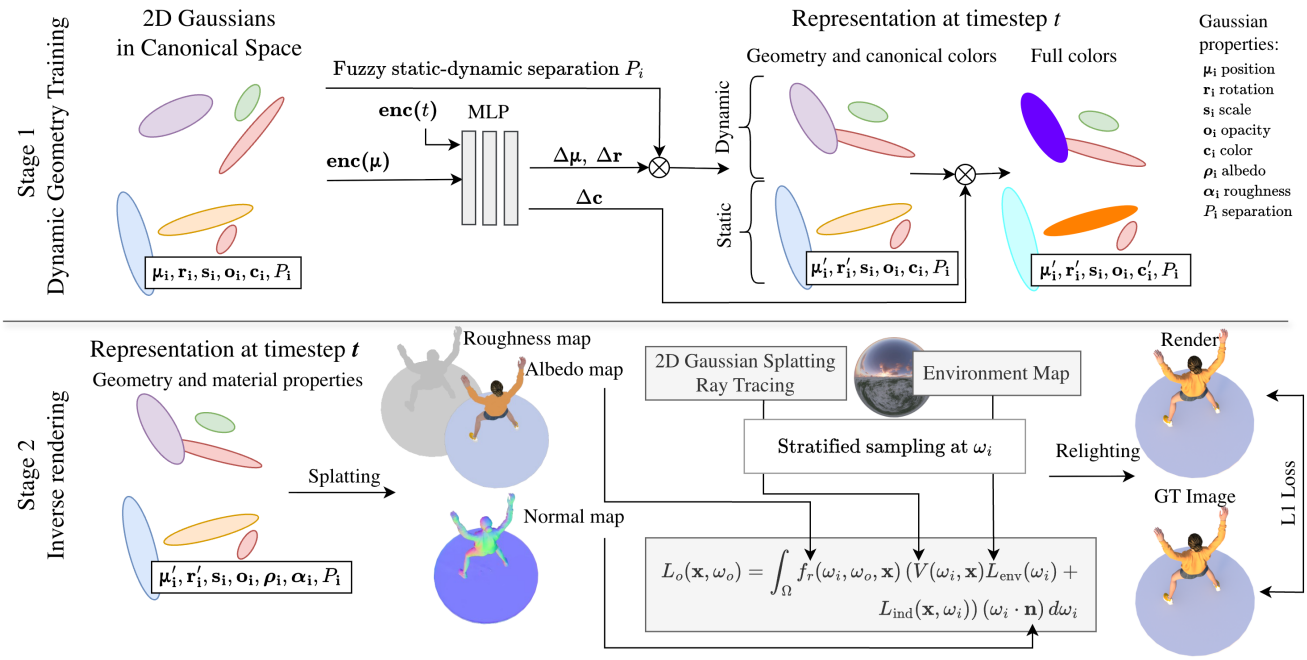

Figure 2. **Method overview.** In Stage 1, a dynamic representation of 2D Gaussians, tailored for the relighting task, is trained. The canonical color from Stage 1 serves as a starting point for albedo and is further optimized in Stage 2, together with roughness and the unknown environment map. In Stage 2, we rasterize the albedo, roughness and normal maps, then perform stratified sampling from the environment map L_{env} and apply ray tracing to compute per-pixel visibility V and indirect light L_{ind}. L1 Loss is computed between the rendered pixels and the ground truth pixels.

where D, F, and G denote the normal distribution function, the Fresnel term, and the geometry term, respectively, and $\mathbf{h} = \frac{\omega_i+\omega_o}{\|\omega_i+\omega_o\|}$, for details see [7]. We further assume that elements of the scene do not emit light.

3.2. Stage 1: Dynamic Geometry Learning for Relighting

We base our method on 2D Gaussian Splatting (2DGS), which provides accurate surface normals that are critical for separating illumination from materials. To capture dynamic scene changes over time, following classical dynamic scene modeling approaches [40, 44], we use a multilayer perceptron (MLP) to predict Gaussian transformations.

Given a timestep $t \in [0,1]$ and the canonical Gaussian position $\boldsymbol{\mu} = (x,y,z)$, the MLP predicts changes in position $\Delta\boldsymbol{\mu} \in \mathbb{R}^3$, rotation $\Delta\mathbf{r} \in \mathbb{R}^3$, and color $\Delta\mathbf{c} \in \mathbb{R}^3$:

$$(\Delta\boldsymbol{\mu}, \Delta\mathbf{r}, \Delta\mathbf{c}) = \text{MLP}(\text{enc}(t), \text{enc}(\boldsymbol{\mu})), \tag{4}$$

where enc$(\cdot)$ denotes positional encoding [32]. The timestep t denotes a moment in time for the dynamic scene being modeled. Note that we choose not to model changes of opacity or scale of the Gaussians as this would allow the MLP to make objects appear and disappear instead of moving them through space. This, in turn, would go against our main goal, which is to recover illumination and materials with the use of motion in the scene.

Static-dynamic fuzzy separation. To compute the actual Gaussian position and rotation at time t, we could simply add the predicted deltas to the canonical values. However, we notice that accurately modeling element dynamics is crucial for correct albedo estimation and can be flawed in textureless areas. For example, moving shadows on a flat surface can be explained either by color changes or by moving/disappearing Gaussians, or a combination of both. Importantly, a moving Gaussian representing a moving shadow cannot be assigned a stable albedo color in Stage 2. Thus, we aim to separate static and dynamic components explicitly in Stage 1.

To achieve such separation, we introduce an auxiliary per-Gaussian variable P that indicates whether the Gaussian belongs to the static or dynamic group. We sample P using a Binary Concrete distribution [30], a continuous relaxation of a Bernoulli distribution that concentrates most mass near 0 or 1. The relaxed variable $\tilde{P}$ is defined as:

$$\tilde{P} = \text{sigmoid}\left(\frac{1}{T}\left(\log(|P|) + \log(U) - \log(1-U)\right)\right),$$
$$U \sim \text{Uniform}(0,1). \tag{5}$$

where T is the temperature hyperparameter. We set $T = 0.5$, encouraging a **near-binary separation**. During inference, we fix $U = 0.5$ to deterministically obtain $\tilde{P}$.

The final Gaussian position $\boldsymbol{\mu}'$ and rotation $\mathbf{r}'$ at time t

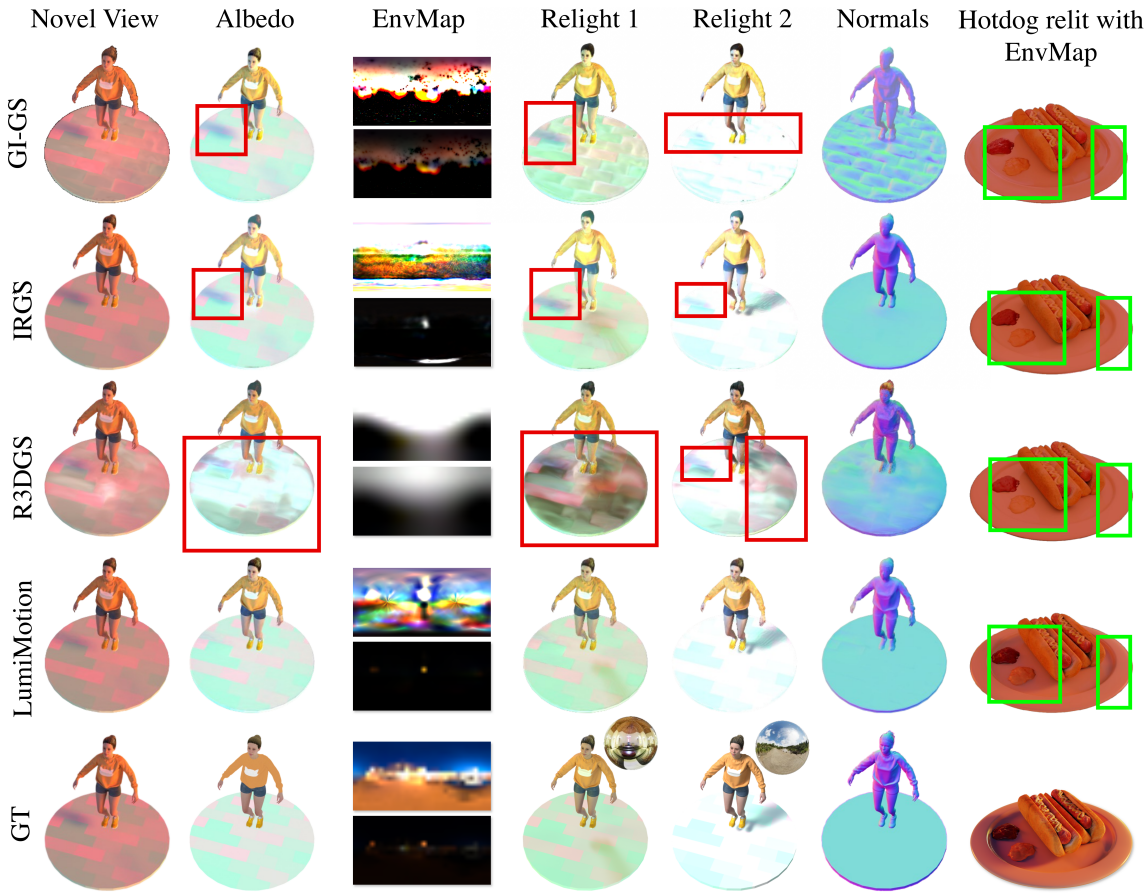

Figure 3. **Qualitative comparison of novel view synthesis, relighting, components and estimated lighting.** All renders are shown from a novel viewpoint. Our method not only achieves stronger relighting capabilities and effectively removes shadows from albedo, but also estimates lighting more accurately. The presented environment map (top shows the map truncated to $[0, 1]$ while bottom shows it scaled so that the maximum value is 1) captures the main light direction more accurately and relights new item more realistically than environment maps estimated from static baselines. **Please zoom in for details.**

are then computed as:

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \tilde{P}\Delta\boldsymbol{\mu}, \quad \mathbf{r}' = \mathbf{r} + \tilde{P}\Delta\mathbf{r}. \tag{6}$$

This formulation along with the regularization on P explained below ensures that dynamic changes are applied selectively, leading to more accurate modeling of scene geometry in time, which is essential for precise relighting and material decomposition in the second stage.

Modeling Temporal Color Variation. Since we focus on dynamic scenes under static lighting, we allow Gaussian colors to vary over time. We observe that significant color changes typically arise from two sources: (i) moving shadows affecting both static and dynamic elements, and (ii) changes in incident illumination on dynamic elements due to motion. To model these effects, we define the color at time t as:

$$\mathbf{c}' = \mathbf{c}(1 - \Delta\mathbf{c}), \tag{7}$$

where $\mathbf{c}$ is Gaussian canonical color. We use multiplicative change to mimic how light affects surfaces (see Eq. (2)), while $(1 - \Delta\mathbf{c})$ allows for applying regularization on excessive color variation. Such formulation captures effects like moving shadows and illumination changes on dynamic elements. The canonical color $\mathbf{c}$ approximates a pseudo-albedo that serves as the initial estimate for material decomposition in Stage 2.

Training Loss. Following 2DGS [15], we apply the reconstruction loss $\mathcal{L}_c$, along with normal consistency loss $\mathcal{L}_n$, which aligns the rendered normal map with the underlying surface geometry, and depth distortion loss $\mathcal{L}_d$, which encourages tight spatial concentration of the Gaussians.

To handle floating Gaussians, we apply a binary cross-entropy loss $\mathcal{L}_o$ with respect to the foreground mask $\mathcal{M}$:

$$\mathcal{L}_o = -\mathcal{M}\log\mathcal{O} - (1 - \mathcal{M})\log(1 - \mathcal{O}), \tag{8}$$

where $\mathcal{O}$ is per-pixel accumulated gaussian opacity. In addition to the above, we introduce losses specific to dynamic

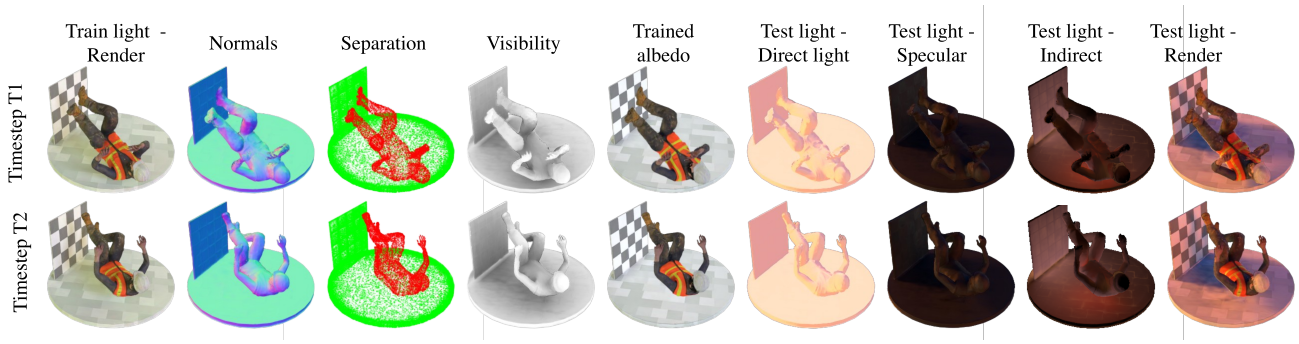

Figure 4. **Qualitative evaluation of components for a dynamic scene for different timesteps. LumiMotion** maintains consistent normals across timesteps, accurately separates static elements, and produces temporally consistent relighting.

modeling.

Static-Dynamic Separation Loss $\mathcal{L}_P$. To encourage Gaussians to remain static whenever possible, we apply an $L1$ penalty on the dynamic assignment variable P:

$$\mathcal{L}_P = \frac{1}{N}\sum_{i=1}^{N}|P_i|, \tag{9}$$

where P_i is the predicted probability of Gaussian i is dynamic, and N is the number of Gaussians. Minimizing $\mathcal{L}_P$ directly encourages P_i to be close to 0, favoring static representations and reducing unnecessary dynamic modeling.

Color and Position Change Regularization. To additionally discourage the model from predicting unnecessary position and color changes, we apply $L2$ regularization on both the predicted color and position deltas, defined as

$$\mathcal{L}_{\Delta c} = \frac{1}{N}\sum_{i=1}^{N}\|\Delta \mathbf{c}_i\|^2, \quad \mathcal{L}_{\Delta \mu} = \frac{1}{N}\sum_{i=1}^{N}\|\Delta \boldsymbol{\mu}_i\|^2, \tag{10}$$

penalizing excessive color variation and spatial movement of Gaussians.

Overall Loss. The total loss for the first stage training is then:

$$\begin{aligned}\mathcal{L}^1 = \mathcal{L}_c + \lambda_n\mathcal{L}_n + \lambda_d\mathcal{L}_d + \lambda_o\mathcal{L}_o + \lambda_P\mathcal{L}_P \\ +\lambda_{\Delta c}\mathcal{L}_{\Delta c} + \lambda_{\Delta \mu}\mathcal{L}_{\Delta \mu},\end{aligned} \tag{11}$$

where $\lambda_n, \lambda_d, \lambda_o, \lambda_P, \lambda_{\Delta c}, \lambda_{\Delta \mu}$ are weighting coefficients. Please see supplementary materials for their weights.

3.3. Stage 2: Inverse Rendering

In the second stage, we perform inverse rendering to decompose the scene into material properties and environment lighting. To model materials, each 2D Gaussian is assigned a diffuse albedo ρ that is initialized with the canonical color $\mathbf{c}$ from the first stage, and roughness α. These properties remain constant for each timestep t. In Stage 2, color changes arise solely from the rendering equation, which determines

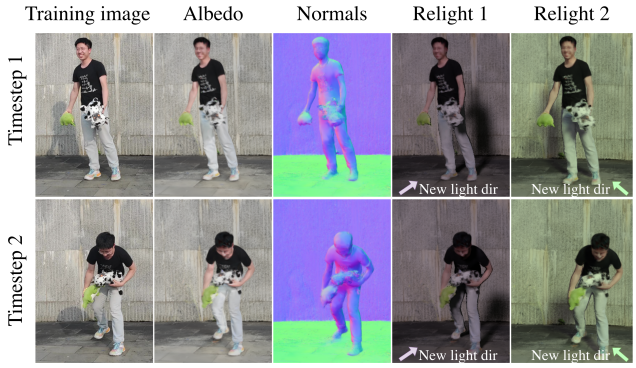

Figure 5. **LumiMotion on real-world ENeRF scene.** The sharp shadow cast by the dynamic actor is largely removed from albedo. Under novel lighting, the actor appears realistic, casting shadows in appropriate locations.

light–surface interaction and thus the $\Delta\mathbf{c}$ output of the MLP is not used. Environment lighting L_{env} is modeled using an image where each pixel corresponds to light intensity and color from a direction ω_i. During Stage 2 we jointly optimize ρ and α for each Gaussian as well as a single L_{env} for the whole scene. Further details about optimized parameter set and gradient flow are available in supplementary.

When rendering the scene, similarly to [7, 12], we apply the rendering equation after rasterization rather than per-Gaussian. This approach allows shading effects such as shadows to appear at each pixel, rather than being limited to per-Gaussian granularity. To obtain per-pixel material values we alpha-blend the albedo and roughness attributes across Gaussians during rasterization.

The incident radiance L_i at surface point $\mathbf{x}$ along direction ω_i is represented by the sum $V(\omega_i, \mathbf{x})L_{\text{env}}(\omega_i) + L_{\text{ind}}(\omega_i, \mathbf{x})$, where the visibility term $V(\omega_i, \mathbf{x}) \in \{0, 1\}$ is obtained via 2D Gaussian ray tracing from $\mathbf{x}$ in direction ω_i. A low value of $V(\omega_i, \mathbf{x})$ indicates that the light from $L_{env}(\omega_i)$ is occluded before reaching point $\mathbf{x}$. We compute indirect term L_{ind} similarly to [12] where indirect light values are traced: during training, RGB values used in ray

Table 2. **Quantitative results for albedo estimation and relighting.** Static methods use only one timestep with multiple views of the scene. **LumiMotion** uses a dynamic scene for training while testing is done on the same views and the same single timestep as in the static setting. Results are grouped by train-test light setting. Although training on dynamic scenes is **inherently more challenging**, **LumiMotion** significantly outperforms albedo estimation tasks on all metrics across all baselines. Notably, for both tasks, we achieve a substantial improvement in LPIPS which is a metric that corresponds well to perceptual quality.

Env Lights	Method	Albedo			Relight		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Dam Wall	R-3DGS	20.744 ± 0.661	0.900 ± 0.013	0.128 ± 0.031	21.220 ± 1.843	0.915 ± 0.016	0.112 ± 0.028
↓	GI-GS	20.943 ± 1.747	0.906 ± 0.014	0.105 ± 0.023	18.434 ± 1.681	0.868 ± 0.023	0.139 ± 0.032
Harbour Sunset	IR-GS	22.888 ± 1.559	0.936 ± 0.013	0.076 ± 0.023	26.177 ± 1.606	0.953 ± 0.011	0.064 ± 0.018
	LumiMotion	27.268 ± 1.568	0.952 ± 0.007	0.069 ± 0.024	26.037 ± 0.579	0.928 ± 0.007	0.060 ± 0.012
Chapel Day	R-3DGS	22.463 ± 2.001	0.927 ± 0.017	0.096 ± 0.038	22.282 ± 2.806	0.943 ± 0.012	0.081 ± 0.030
↓	GI-GS	24.733 ± 2.862	0.955 ± 0.014	0.056 ± 0.016	22.673 ± 1.513	0.880 ± 0.015	0.125 ± 0.022
Golden Bay	IR-GS	23.769 ± 2.732	0.956 ± 0.015	0.053 ± 0.017	28.157 ± 1.978	0.966 ± 0.009	0.046 ± 0.020
	LumiMotion	30.838 ± 1.798	0.973 ± 0.007	0.036± 0.014	28.563 ± 0.478	0.939 ± 0.011	0.041 ± 0.007
Golden Bay	R-3DGS	19.945 ± 1.124	0.899± 0.018	0.133 ± 0.041	19.563 ± 1.874	0.918 ± 0.013	0.118 ± 0.033
↓	GI-GS	21.295 ± 2.930	0.932 ± 0.020	0.087 ± 0.025	17.636 ± 2.293	0.823 ± 0.029	0.132 ± 0.030
Dam Wall	IR-GS	20.910 ± 1.379	0.937 ± 0.013	0.082 ± 0.024	25.009 ± 1.615	0.955 ± 0.011	0.060 ± 0.014
	LumiMotion	27.929 ± 1.932	0.959± 0.010	0.058 ± 0.019	25.405 ± 0.690	0.936 ± 0.013	0.048 ± 0.009

tracing correspond to the colors $\mathbf{c}$ from the first stage, while for inference under novel lighting, we use colors evaluated from the rendering equation. We employ Monte Carlo integration with uniform stratified sampling selecting N_r ray directions over the hemisphere to efficiently evaluate the rendering equation. The final RGB output of the rendering function can be written as:

$$\mathbf{c}_{\text{pbr}}(\omega_o) = \frac{2\pi}{N_r} \sum_{i=1}^{N_r} f_r(\omega_i, \omega_o, \mathbf{x}) L_i(\omega_i, \mathbf{x})(\omega_i \cdot \mathbf{n}) \quad (12)$$

where L_i denotes incident radiance. Following [12], we also randomly sample N pixels per iteration to reduce computation time.

Stage 2 combines three losses: (1) L_c loss from Stage 1, (2) $L1$ loss for Stage 2 renders against GT pixels, (3) $L2$ regularization with small weight λ_{env} that penalizes high values in the lower region of L_{env}. L_c is computed between GT images and pure Gaussian splatting renders and it constrains the fine-tuned Gaussian parameters, preventing them from deviating excessively from their Stage 1 values. $L1$ is the only loss used to supervise the $\mathbf{c}_{\text{pbr}}$.

4. Experiments and Results

Datasets. To thoroughly evaluate our method under controlled relighting and shadow-casting conditions, we introduce a novel synthetic benchmark consisting of **20 variations**: five distinct scenes, each rendered under four different lighting environments ('Harbour Sunset', 'Dam Wall', 'Golden Bay', and 'Chapel Day'). The scenes differ in object types, motion patterns, and levels of surface specularity. For every scene, both static and dynamic versions are provided to enable fair comparison across temporal settings. For dynamic version we have D-NeRF [34]-like captures: one view per one timestep. The static dataset captures a single timestep from the same camera poses as the dynamic version. This setup enables training both static and dynamic models on comparable data, and allows direct evaluation using identical camera views and time steps. Test set consists of novel views. Additional details about the dataset are provided in the **Supplementary Material**.

To qualitatively assess performance in real-world conditions, we use two scenes from ENeRF dataset [26], an outdoor dataset which captures dynamic people casting prominent shadows from 18 cameras. Its multi-view nature supports training and evaluation of static and dynamic models.

Experimental setup. For synthetic data, we evaluate key aspects of our method: albedo estimation and relighting quality. We report PSNR, SSIM [39], and LPIPS [50] as evaluation metrics. For each scene in our dataset, we conduct experiments under three configurations, selecting one light for training and another for testing. Detailed training parameters and extended evaluation including videos are provided in the Supplementary Material. All experiments are conducted on an NVIDIA RTX 3090, with training for both stages taking approximately 1.2 hours per synthetic scene.

Results. Quantitative comparisons against state-of-the-art methods are summarized in Tab. 2, demonstrating our method's strong performance across the evaluation metrics. We achieve strong performance in albedo estimation, surpassing **all baselines across all metrics** by a large margin, which demonstrates the enhanced capability of our method in removing light-related artifacts. For relighting, our metrics surpass the baselines in most cases. With regard

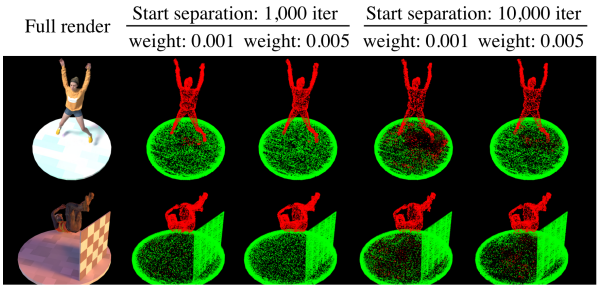

Figure 6. **Influence of separation loss parameters.** The dynamic objects in the scene remain correctly classified as dynamic across tested separation weights and initialization times. However, shadows on static parts of the scene (e.g. ground) can be misclassified as dynamic Gaussians (here mostly visible for start 10,000 iter). Increasing the separation weight or starting the separation earlier helps to reduce such misclassifications. **Zoom in for details.**

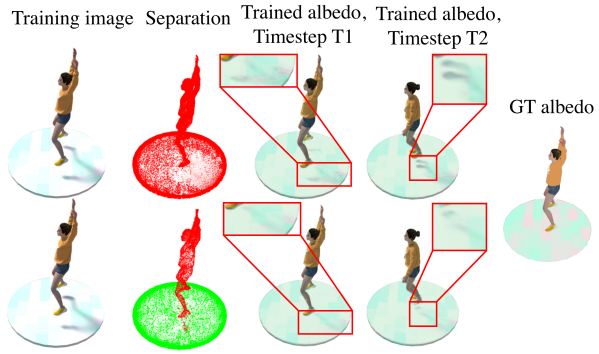

Figure 7. **Ablation study of static-dynamic separation.** The demonstrated scene is characterized with strong directional light and strong shadows. Without the static-dynamic separation (top), the Gaussians reconstruct a strong shadow that follows the object motion, which degrades albedo optimization in the second stage. When separation is enabled, the shadows are modeled by $\Delta\mathbf{c}$ instead of $\Delta\boldsymbol{\mu}$ which enables correct albedo optimization. **Please zoom in for details.**

to LPIPS—the metric that best corresponds to perceptual quality—**LumiMotion** performs best in all cases, with an average improvement of 15%. Relighting task is inherently more difficult for dynamic scenes - it depends strongly on properly estimated normals, which are more challenging to obtain due to the added constraint of temporal consistency. It is notable that our approach performs competitively on the relighting task, highlighting its capability even under the complexities of a dynamic setup. We present qualitative results in Fig. 3 and highlight improvements in relighting fidelity, albedo consistency and environment map reconstruction. Please note that, thanks to dynamics, we can better estimate the direction of incoming light, and also prevent the model from baking shadows in the base color. **We provide extended results in the Supplementary Material.**

We evaluate our approach on two clips from the **challenging real-world** ENeRF dataset, where sharp shadows are cast by moving actors. This setup enables us to assess our method's robustness in handling real-world lighting variations and dynamic geometry. Since it is a multiview setup, we can compare our method with IRGS which is the most recent baseline. See Fig. 1 where, relative to IRGS, we remove the majority of shadows from the albedo from the first scene. Additionally, unlike IRGS our method does not exhibit artifacts when rendering with specular component. The second scene is presented in Fig. 5.

Table 3. **Ablation study on Golden Bay → Dam Wall configuration,** characterized by strong directional train light. The additional components improve albedo and relight reconstruction quality.

Method	Albedo			Relight		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
w/o Stage 2	23.935	0.943	0.072	23.154	0.926	0.060
w/o $\Delta\mathbf{c}$	23.983	0.937	0.083	23.142	0.926	0.061
w/o P	26.768	0.954	0.064	25.079	0.933	0.052
full model	**27.929**	**0.959**	**0.058**	**25.405**	**0.936**	**0.048**

In Fig. 4, we demonstrate that our method produces coherent renderings across timesteps, with smooth normals, consistent separation of dynamic elements, and shadows estimated in accordance with the moving geometry.

Finally, we perform an ablation study on key components of our pipeline (Tab. 3). Fig. 6 illustrates the influence of separation parameters, showing how varying the separation weight and initialization affects the identification of dynamic elements. We show that Gaussians that move to simulate lighting effects, such as shadows, negatively impact albedo optimization. Fig. 7 highlights the benefits of our separation strategy in addressing this issue.

5. Conclusions and Limitations

We proposed a two-stage inverse rendering framework that leverages dynamics as a supervisory signal to separate illumination from material properties. We further introduced a benchmark with scenes under various lighting and motion conditions, enabling systematic evaluation of our approach. **LumiMotion** achieves improved relighting performance and more accurate material estimation compared to static-only baselines, proving that incorporating dynamic content is beneficial for these tasks. **Limitations.** We observe that accurate normal estimation and the quality of the learned deformations are critical to performance. In our framework, reconstruction quality remains limited, as temporally consistent and physically accurate motion and normal estimation in complex dynamic scenes is still an open challenge. Our simple separation strategy can generate artifacts when handling intricate dynamics, and more accurate supervision, such as optical flow, may be crucial. Additionally, the framework is sensitive to inaccurate camera pose estimation, sparse camera setups or inaccurate initialization.

Acknowledgements. This paper received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857533. The research is supported by Sano project carried out within the International Research Agendas programme of the Foundation for Polish Science, co-financed by the European Union under the European Regional Development Fund. The research was created within the project of the Minister of Science and Higher Education "Support for the activity of Centers of Excellence established in Poland under Horizon 2020" on the basis of the contract number MEiN/2023/DIR/3796. The work of J. Kaleta was supported by National Science Centre, Poland (grant no. 2022/47/O/ST6/01407). The work of K. Marzol was supported by the project *Effective Rendering of 3D Objects Using Gaussian Splatting in an Augmented Reality Environment* (FENG.02.02-IP.05-0114/23), carried out under the First Team programme of the Foundation for Polish Science and co-financed by the European Union through the European Funds for Smart Economy 2021–2027 (FENG). The work of P. Wójcik was supported by the German Research Foundations (DFG) funded Collaborative Research Center 1310 *Predictability in evolution* project C03. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017221.

References

[1] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering intrinsic scene characteristics. *Comput. vis. syst*, 2 (3-26):2, 1978. 1

[2] Zoubin Bi, Yixin Zeng, Chong Zeng, Fan Pei, Xiang Feng, Kun Zhou, and Hongzhi Wu. Gs^3: Efficient relighting with triple gaussian splatting. In *SIGGRAPH Asia 2024 Conference Papers, SA 2024, Tokyo, Japan, December 3-6, 2024*, pages 12:1–12:12. ACM, 2024. 2, 3

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 12664–12674. IEEE, 2021. 3

[4] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm siggraph*, pages 1–7. vol. 2012, 2012. 3

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022. 2, 3

[6] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21557–21568, 2021. 3

[7] Hongze Chen, Zehong Lin, and Jun Zhang. GI-GS: global illumination decomposition on gaussian splatting for inverse rendering. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 2, 3, 4, 6

[8] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21476–21485, 2023. 2

[9] Jaehoon Choi, Yonghan Lee, Hyungtae Lee, Heesung Kwon, and Dinesh Manocha. Meshgs: Adaptive mesh-aligned gaussian splatting for high-quality rendering. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 3310–3326, 2024. 3

[10] Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *European Conference on Computer Vision*, pages 73–89. Springer, 2024. 2, 3

[11] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14346–14355, 2021. 2

[12] Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10943–10952, 2025. 1, 2, 3, 6, 7

[13] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 3

[14] Yu Hong, Yize Wu, Zhehao Shen, Chengcheng Guo, Yuheng Jiang, Yingliang Zhang, Qiang Hu, Jingyi Yu, and Lan Xu. Beam: Bridging physically-based rendering and gaussian modeling for relightable volumetric video. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 7968–7977, 2025. 3

[15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 27 July 2024-1 August 2024*, page 32. ACM, 2024. 2, 3, 5

[16] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. SC-GS: sparse-controlled gaussian splatting for editable dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 4220–4230. IEEE, 2024. 3

[17] Umar Iqbal, Akin Caliskan, Koki Nagano, Sameh Khamis, Pavlo Molchanov, and Jan Kautz. RANA: relightable articulated neural avatars. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 23085–23096. IEEE, 2023. 2

[18] 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. 3

[19] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 165–174. IEEE, 2023. 2, 3

[20] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, page 143–150, New York, NY, USA, 1986. Association for Computing Machinery. 3

[21] Joanna Kaleta, Kacper Kania, Tomasz Trzciński, and Marek Kowalski. Lumigauss: Relightable gaussian splatting in the wild. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE, 2025. 2

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. 1, 2

[23] Zhengfei Kuang, Yunzhi Zhang, Hong-Xing Yu, Samir Agarwala, Elliott Wu, Jiajun Wu, et al. Stanford-orb: a real-world 3d object inverse rendering benchmark. *Advances in Neural Information Processing Systems*, 36:46938–46957, 2023. 2

[24] Junxuan Li, Chen Cao, Gabriel Schwartz, Rawal Khirodkar, Christian Richardt, Tomas Simon, Yaser Sheikh, and Shunsuke Saito. Uravatar: Universal relightable gaussian codec avatars. In *SIGGRAPH Asia 2024 Conference Papers, SA 2024, Tokyo, Japan, December 3-6, 2024*, pages 128:1–128:11. ACM, 2024. 3

[25] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21644–21653, 2024. 3

[26] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia Conference Proceedings*, 2022. 7

[27] Isabella Liu, Linghao Chen, Ziyang Fu, Liwen Wu, Haian Jin, Zhong Li, Chin Ming Ryan Wong, Yi Xu, Ravi Ramamoorthi, Zexiang Xu, et al. Openillumination: A multi-illumination dataset for inverse rendering evaluation on real objects. *Advances in Neural Information Processing Systems*, 36:36951–36962, 2023. 2

[28] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic gaussians mesh: Consistent mesh reconstruction from monocular videos. *arXiv preprint arXiv:2404.12379*, 2024. 3

[29] Diogo Carbonera Luvizon, Vladislav Golyanik, Adam Kortylewski, Marc Habermann, and Christian Theobalt. Relightable neural actor with intrinsic decomposition and pose control. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LIX*, pages 465–483. Springer, 2024. 2

[30] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 4

[31] Julieta Martinez, Emily Kim, Javier Romero, Timur Bagautdinov, Shunsuke Saito, Shoou-I Yu, Stuart Anderson, Michael Zollhöfer, Te-Li Wang, Shaojie Bai, Chenghui Li, Shih-En Wei, Rohan Joshi, Wyatt Borsos, Tomas Simon, Jason Saragih, Paul Theodosis, Alexander Greene, Anjani Josyula, Silvio Mano Maeta, Andrew I. Jewett, Simon Venshtain, Christopher Heilman, Yueh-Tung Chen, Sidi Fu, Mohamed Ezzeldin A. Elshaer, Tingfang Du, Longhua Wu, Shen-Chi Chen, Kai Kang, Michael Wu, Youssef Emad, Steven Longay, Ashley Brewer, Hitesh Shah, James Booth, Taylor Koska, Kayla Haidle, Matt Andromalos, Joanna Hsu, Thomas Dauer, Peter Selednik, Tim Godisart, Scott Ardisson, Matthew Cipperly, Ben Humberston, Lon Farr, Bob Hansen, Peihong Guo, Dave Braun, Steven Krenn, He Wen, Lucas Evans, Natalia Fadeeva, Matthew Stewart, Gabriel Schwartz, Divam Gupta, Gyeongsik Moon, Kaiwen Guo, Yuan Dong, Yichen Xu, Takaaki Shiratori, Fabian Prada, Bernardo R. Pires, Bo Peng, Julia Buffalini, Autumn Trimble, Kevyn McPhail, Melissa Schoeller, and Yaser Sheikh. Codec Avatar Studio: Paired Human Captures for Complete, Driveable, and Generalizable Avatars. *NeurIPS Track on Datasets and Benchmarks*, 2024. 2

[32] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2022. 1, 2, 4

[33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2

[34] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021. 7

[35] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 130–141. IEEE, 2024. 2, 3

[36] Joanna Waczyńska, Piotr Borycki, Sławomir Konrad Tadeja, Jacek Tabor, and Przemysław Spurek. Games: Mesh-based adapting and modification of gaussian splatting. *ArXiv*, abs/2402.01459, 2024. 2

[37] Shaofei Wang, Bozidar Antic, Andreas Geiger, and Siyu Tang. Intrinsicavatar: Physically based inverse rendering of dynamic humans from monocular videos via explicit ray tracing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 1877–1888. IEEE, 2024. 2

[38] Shaofei Wang, Tomas Simon, Igor Santesteban, Timur Bagautdinov, Junxuan Li, Vasu Agrawal, Fabian Prada, Shoou-I Yu, Pace Nalbone, Matt Gramlich, et al. Relightable full-body gaussian codec avatars. In *Proceedings of the*

Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers, pages 1–12, 2025. 2, 3

[39] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 7

[40] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 3, 4

[41] Tong Wu, Jiali Sun, Yu-Kun Lai, Yuewen Ma, Leif Kobbelt, and Lin Gao. Deferredgs: Decoupled and editable gaussian splatting with deferred shading. *ArXiv*, abs/2404.09412, 2024. 2

[42] Tao Xie, Xi Chen, Zhen Xu, Yiman Xie, Yudong Jin, Yujun Shen, Sida Peng, Hujun Bao, and Xiaowei Zhou. Envgs: Modeling view-dependent appearance with environment gaussian. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5742–5751, 2025. 2

[43] Yingyan Xu, Gaspard Zoss, Prashanth Chandran, Markus Gross, Derek Bradley, and Paulo Gotardo. Renerf: Relightable neural radiance fields with nearfield lighting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22524–22534, 2023. 3

[44] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20331–20341, 2024. 3, 4

[45] 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. 3

[46] Yuxuan Yao, Zixuan Zeng, Chun Gu, Xiatian Zhu, and Li Zhang. Reflective gaussian splatting. *arXiv preprint*, 2024. 3

[47] Keyang Ye, Qiming Hou, and Kun Zhou. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 3

[48] Youyi Zhan, Tianjia Shao, He Wang, Yin Yang, and Kun Zhou. Interactive rendering of relightable and animatable gaussian avatars. *arXiv preprint arXiv:2407.10707*, 2024. 2, 3

[49] 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. *International Conference on Computer Vision (ICCV)*, 2023. 3

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[51] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6):237:1–237:18, 2021. 2

[52] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18622–18631. IEEE, 2022. 2

[53] Yiqun Zhao, Chenming Wu, Binbin Huang, Yihao Zhi, Chen Zhao, Jingdong Wang, and Shenghua Gao. Surfel-based gaussian inverse rendering for fast and relightable dynamic human reconstruction from monocular video. *arXiv preprint arXiv:2407.15212*, 2024. 3