TRON: TRACING RAYS TO ORCHESTRATE A NEURAL RENDERER FOR 3D GAUSSIAN RECONSTRUCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce TRON, a rendering framework that combines 3D Gaussian ray tracing with neural rendering to enable realistic and controllable rendering of real-world 3D scenes under novel lighting, dynamic object motion, object insertion, and material editing. Prior approaches that rely solely on physically based rendering (PBR) of Gaussian representations struggle to achieve realistic relighting due to imperfections in reconstructed geometry, material estimates, and light transport estimation. At the same time, neural rendering methods often lack an explicit scene representation, limiting their ability to support interactive editing with fine-grained manipulation. TRON bridges these two paradigms. We use intrinsic decomposition priors from a learned inverse rendering model to regularize the material properties of a Gaussian field, and repurpose a ray tracer to provide radiometric guidance rather than final pixels. By treating this output as a structured 3D scaffold, we empower a lightweight neural renderer to bridge the domain gap between shading-model constrained estimates and photorealistic output. Our key insight is that the combination of explicit 3D knowledge with robust material priors provides speed and controllability, while neural rendering enables the synthesis of photorealistic images. To support real-world scenarios, we train our neural renderer with a multi-stage strategy consisting of large-scale pretraining and targeted fine-tuning on a newly constructed dataset of 2.1M rendered synthetic and real-world frames from 3D reconstructions. TRON outperforms Gaussian-based relighting methods in realism, and prior neural renderers in editability and speed. To the best of our knowledge, TRON is the first method to enable practical interactive applications in captured 3D environments, offering realistic appearance under dynamic geometric, lighting and material conditions.

Keywords: Neural Rendering, 3D Gaussian Ray Tracing, Inverse Rendering, Relighting

1 INTRODUCTION

Recent advances have propelled multi-view reconstructions based on 3D Gaussian Splats (Kerbl et al., 2023) toward interactive use cases, enabling elasto-dynamic simulation (Modi et al., 2024; Xie et al., 2023), mechanical property prediction (Dagli et al., 2025), and painting (Pandey et al., 2025) directly with 3D Gaussians. Despite promising demonstrations, broad adoption in interactive settings remains limited, as such applications require rendering under dynamic lighting and object motion. A key limitation of Gaussian representations is their reliance on *baked shading*, where lighting effects (e.g., shadows and reflections) are encoded in a fixed scene radiance. Motivated to lift these barriers, we introduce TRON, a novel framework that extends Gaussian ray tracing (Moenne-Loccoz et al., 2024) with explicit modeling of lighting effects coupled with a neural renderer, enabling realistic rendering under dynamic lighting, moving objects, and varying materials.

Our rendering solution considers three key requirements: (i) direct 3D control with dynamic materials, illumination and object poses, (ii) interactive response to these dynamic edits (no pre-computation), and (iii) realistic rendering consistent with these controls. Recent approaches extending 3DGS to physically-based rendering (PBR) can enable some editing (Fan et al., 2025; Du et al., 2024), but offer limited support of dynamic object motion, and often struggle to produce realistic results due to imperfect geometry, materials, and light transport computation in this under-constrained intrinsic decomposition problem. At the same time, emerging neural rendering approaches (Liang et al.,

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

2025a) show photorealistic results, but lack explicit control over 3D properties (He et al., 2025) and suffer from inconsistencies across camera trajectories due to their generative nature. Furthermore, these approaches remain computationally demanding and are often designed for offline use.

Our method combines the strength of explicit Gaussian representations and a neural renderer. To alleviate the ill-posed nature of intrinsic decomposition, we leverage existing priors (Liang et al., 2025a) and extend 3D Gaussian ray tracing (Moenne-Loccoz et al., 2024) to dynamic PBR shading under novel HDR environment map lighting. Our pipeline is the first to render *irradiance under dynamic geometry and illumination in real-time* for a Gaussian-based representation. This allows us to design a neural renderer conditioned on both PBR shading and irradiance, enabling dynamic control and improved realism over relightable Gaussian baselines. To mitigate the lack of paired real-world data, we propose a multi-stage training curriculum, including (i) large-scale pretraining on synthetic data with faithful illumination and material controls, (ii) domain adaptation to real-world scenes with imperfections, and (iii) multi-frame temporal consistency fine-tuning. To support interactive use, we adopt efficient single-step architectures (Wu et al., 2025a; Zhang et al., 2026).

To evaluate our framework, we consider the task of novel-view rendering under modified relighting conditions in both synthetic and real-world settings. Our quantitative evaluation include visual error metrics to assess reconstruction quality and runtime costs, demonstrating clear improvements in both aspects. Beyond relighting, we showcase challenging real-world applications, including rendering *dynamic shadows in a physically simulated Gaussian scene*, material editing and harmonized object insertion. In summary, our contributions are as follows:

- A novel rendering framework that combines Gaussian ray tracing and a neural renderer for controllable, multi-view consistent, interactive rendering under dynamic conditions in real-world scenes.
- A multi-stage neural renderer training strategy that balances controllability and realism by combining large-scale synthetic pretraining, adaptation to real-world data with imperfect reconstructions, and temporal consistency optimization.
- Improved fidelity on real-world scenes in a range of interactive applications, including relighting, material editing, object insertion and removal.

2 Related Work

Reconstruction methods utilizing volumetric radiance fields (Mildenhall et al., 2020; Huang et al., 2024; Moenne-Loccoz et al., 2024; Wu et al., 2025b) synthesize photorealistic novel views, but entangle materials and illumination into baked radiance, precluding relighting. Within this line of works, contemporary relightable Gaussian based methods, therefore optimize physically based material properties instead (Gao et al., 2024; Jiang et al., 2023), approximating visibility through baked probes (Liang et al., 2023b), single-bounce global illumination (Chen et al., 2025a), or diffusion priors (Du et al., 2024). Given that inverse rendering is ill-posed, all rely on fragile regularizations, and often suffer from shadows bleeding into base color, thus degrading when relighting far from capture conditions, i.e. under highly contrastive light that casts long shadows. Coverage is also narrow, as many works are confined to isolated objects rather than full scenes (Gu et al., 2025), single specular objects (Keyang et al., 2024), one-light-at-a-time (Bi et al., 2024), or outdoor scenes (Kaleta et al., 2025). Critically, all adopt shading models incapable of true photorealism, and most bake shadows or require re-optimization with dynamic geometry.

A parallel line of works relights in image or video space using large generative models (Zeng et al., 2024b; Liang et al., 2025a; He et al., 2025; Zhang et al., 2025a), achieving realism without explicit 3D control, but with potential view inconsistency, and at high cost, where single-step distillation (Wu et al., 2025a; Zhang et al., 2026) provides a promising mitigation to the latter.

Recently, hybrid methods attempt to bridge both lines of works. GR3EN (Xing et al., 2026) uses a neural renderer to bake a new relit appearance into the original 3D reconstruction, inhibiting online dynamic lighting or scene motion. Another recent work (Careaga & Aksoy, 2025) relights photographs by combining explicit path-tracing with a finetuned neural renderer in a 2.5D setting. To our knowledge, application of neural rendering for dynamic relighting of Gaussians has not been explored, prohibiting responsive relighting with faithful shadows under motion. An extended related works section describing the evolution of works related to our setting is provided in Appendix B.

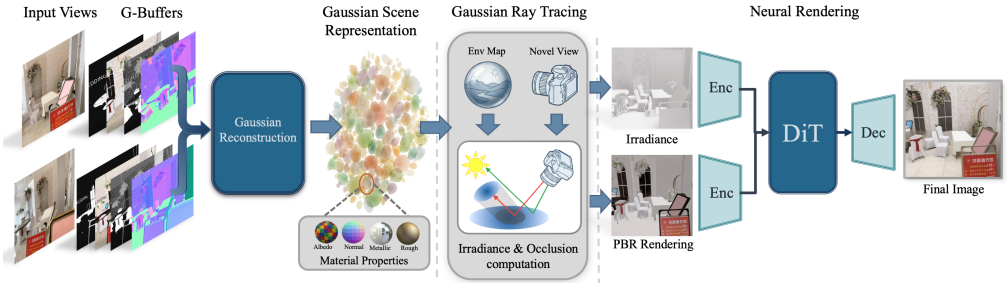


Figure 1: **TRON Architecture** During training, we (1) apply an intrinsic decomposition model on all input views to extract G-buffers diffusion priors and then (2) reconstruct a scene from input views as a set of 2D Gaussians, and (3) lift and bake the G-buffer priors in 3D. At inference time, given an envmap and camera view we render a pair of buffers: PBR shaded and and irradiance, then the Neural Renderer maps these pairs of buffers to a high quality RGB output image.

3 METHOD

Given multi-view images and camera parameters $\{(I_i, C_i)\}_{i=1}^{N}$, our goal is to infer a 3D representation that can be rendered under any novel global environment map E or object motion. Our framework (Fig.1) optimizes a Gaussian particle representation (§3.1) guided by existing priors (§3.2) and extends Gaussian ray tracing to PBR and irradiance (§3.3), improving realism with a neural renderer that synthesizes the final image (§4). Our formulation supports dynamic rendering (no precomputation) under dynamic lighting, Gaussian object motion, insertion and removal, and PBR material editing.

3.1 GAUSSIAN SCENE REPRESENTATION

We optimize a 3D scene $\mathcal{G}$ consisting of M material-augmented Gaussians, defined as follows:

$$\mathcal{G} = \left\{ A_j = (\boldsymbol{\mu}_j, \mathbf{R}_j, \mathbf{s}_j, \sigma_j, \boldsymbol{\beta}_j); \quad B_j = (\mathbf{b}_j, m_j, r_j) \right\}_{j=1}^{M}, \tag{1}$$

where A_j denotes standard 3DGS parameters (Kerbl et al., 2023; Moenne-Loccoz et al., 2024) (position $\boldsymbol{\mu}_j$, opacity σ_j, view-dependent color $\boldsymbol{\beta}_j$ represented with spherical harmonics of degree 3) and B_j encodes our additional intrinsic material properties. To encourage better surfaces, we follow 2D Gaussian splatting(Huang et al., 2024) and parametrize each Gaussian with a rotation matrix $\mathbf{R}_j \in SO(3)$ and 2D scaling $\mathbf{s}_j \in \mathbb{R}_+^2$. B_j contains per-particle base color $\mathbf{b}_j \in \mathbb{R}^3$, as well as scalar metallicity m_j, and roughness r_j values. We leverage rich priors to disentangle these properties from the traditionally entangled appearance in A_j.

3.2 PRIOR-INFORMED RECONSTRUCTION

To mitigate the ill-posed nature of material estimation from renderings alone, we leverage a learned intrinsic image decomposition prior (Liang et al., 2025a) to predict per-view material maps $\phi(I) = (I_n, I_b, I_r, I_m)$, containing normals I_n, base color I_b, roughness I_r and metallicity I_m. These are collectively referred to as the G-Buffers, and define the augmented multi-view input $\{(I_i, \phi(I_i), C_i)\}_{i=1}^{N}$. These predicted values provide a valuable signal, but can be inconsistent across views, making optimization more challenging. To stabilize optimization, we propose a two-stage procedure. In the first stage, we optimize the geometric and baked appearance $\{A_j\}$, supervised by images I with an additional loss term between geometric normal and predicted I_n. This stage enforces alignment between geometry and appearance. In the second stage, we fix $\{A_j\}$ and optimize the materials $\{B_j\}$ using the remaining signals (I_b, I_r, I_m). See D.1 for full optimization details.

3.3 RAY-TRACED SHADING AND IRRADIANCE

Even with strong learned priors, reconstructed geometry and materials remain imperfect. As a result, directly using physically based rendering as the final output often produces artifacts. However, the reconstructed Gaussian scene still provides valuable 3D information about visibility, surface orientation, material composition, and light transport.

Given the material-augmented Gaussian scene $\mathcal{G}$ and an environment map E, TRON extends Gaussian Ray Tracing (Moenne-Loccoz et al., 2024; Wu et al., 2025b) to render two guidance buffers: an

approximate PBR image $\hat{I}_{\text{pbr}}$ and a dynamic irradiance image $\hat{I}_{\text{irr}}$. The PBR buffer captures local material response under the target illumination, while the irradiance buffer captures incoming light modulated by scene visibility, including shadows caused by dynamic geometry. These signals provide complementary information: $\hat{I}_{\text{pbr}}$ encodes material-dependent appearance, whereas $\hat{I}_{\text{irr}}$ exposes illumination and occlusion cues that are difficult for an image-space model to infer reliably.

Efficiently computing these buffers is challenging. Unlike mesh rendering, a primary ray may intersect many Gaussian primitives, making direct shading prohibitively expensive. We first address this with a deferred rendering strategy (Deering et al., 1988): Gaussian contributions are first accumulated along each primary ray to form a representative surface point and G-buffer, and shading is then evaluated once per pixel rather than once per primitive. For efficient PBR shading, we use the split-sum approximation (Karis, 2013; Munkberg et al., 2022), which preintegrates environment lighting for material-dependent reflectance. Since this approximation does not fully account for visibility effects such as cast shadows, we additionally estimate irradiance by tracing a small number of secondary rays. We sample secondary directions using multiple importance sampling (Veach, 1998), biasing samples toward high-contribution directions determined by both the material properties and the environment map. Finally, we exploit the order-invariance of transmittance to trace secondary rays out of order, enabling efficient execution on modern ray-tracing hardware. Additional implementation details are provided in Appendix D.2.

4 Neural Renderer

Our Gaussian ray tracer produces two image buffers per view: a physically based rendering, $\hat{I}_{\text{pbr}}$, that captures direct light–material interactions, and an irradiance buffer, $\hat{I}_{\text{irr}}$, that captures the incoming light at the surface including shadows. By construction, these buffers are approximations of a real photograph: (i) they omit higher–order rendering effects, e.g., subsurface scattering, transparent and complex materials and (ii) they inherit the limitations of Gaussian reconstruction, which typically covers only the foreground of a scene at high density and leaves the periphery sparse, blurry, or missing. For those reasons, closing this gap with a classical forward renderer is impractical. Instead, we propose a *neural renderer* $\mathcal{R}_\theta$ conditioned on $\hat{I}_{\text{pbr}}$ and $\hat{I}_{\text{irr}}$ which produces a single realistic RGB image $\hat{I}_{\text{rgb}} = \mathcal{R}_\theta(\hat{I}_{\text{pbr}}, \hat{I}_{\text{irr}})$. We treat $\mathcal{R}_\theta$ as a strongly conditioned generative model: it must (a) be expressive enough to hallucinate the missing rendering effects and inpaint reconstruction artifacts, and (b) be efficient enough to be queried at interactive rates in a streaming setting. We first describe the architecture and training objective, and then our paired data curation strategy.

4.1 Architecture

Backbone. Pretrained text-to-image diffusion models contain strong generative priors that have been shown to transfer well to image-to-image translation tasks (Zhang et al., 2026). We adopt the same recipe and build on Cosmos 0.6B (NVIDIA et al., 2025), a text-to-image latent diffusion model that combines a 0.6 billion parameter DiT (NVIDIA et al., 2025) with a 0.14B-parameter WAN-2.1 causal video autoencoder (Wan et al., 2025) as its tokenizer. The autoencoder $(\mathcal{E}, \mathcal{D})$ consumes either an image or a short video and emits a temporally compressed latent that the DiT $\mathcal{F}_\theta$ denoises in latent space. To adapt the model to our neural rendering task, we finetune the DiT from the public Cosmos weights; the encoder and decoder are kept frozen.

Input encoding. Because the two input buffers carry very different statistics $\hat{I}_{\text{pbr}}$ is an PBR rendering that includes high-frequency details, textures and reflections, while I_{irr} is a smooth low-frequency irradiance map, we encode the two buffers separately and average the result to create our input latent:

$$z = \tfrac{1}{2}\big(\mathcal{E}(\hat{I}_{\text{pbr}}) + \mathcal{E}(\hat{I}_{\text{irr}})\big). \tag{2}$$

Compared to channel-wise concatenation, averaging keeps the latent dimensionality of the original Cosmos backbone unchanged, allowing us to reuse all DiT weights. Figures 8 and 9 present an ablation study of this design choice.

Single-step deterministic denoising. In standard latent diffusion, the backbone $\mathcal{F}_\theta$ is queried for many timesteps along a stochastic noise schedule and conditioned on a text prompt. For real-time online rendering this is prohibitively expensive. Following the single-step training paradigm of (Zhang et al., 2026), we repurpose $\mathcal{F}_\theta$ into a deterministic image-to-image operator: we feed the clean latent

z from Eq. (2) directly into the DiT, fix the diffusion timestep to a constant value τ (we use $\tau = 250$ on a 1000-step schedule), and replace the text-conditioning tokens with the model's null embedding. The decoder then produces the predicted RGB image:

$$I_{\text{rgb}} \;=\; \mathcal{D}\Big(\mathcal{F}_\theta\big(z;\,\tau,\,\varnothing\big)\Big). \tag{3}$$

This single forward pass yields a stable, deterministic mapping from input buffers to RGB, eliminates iterative sampling, and improves frame-to-frame structural consistency in the streaming setting.

Temporal context. A purely per-frame renderer can flicker on moving cameras because the network is free to choose a different realization of the missing content each frame. We therefore extends $\mathcal{R}_\theta$ to operate on short clips. Let K be the clip length; following the WAN-2.1 tokenizer, which compresses time by a factor of 4, we use $K = 1{+}4n$ ($K{=}5$ in practice). We stack the K frames of each buffer along the temporal axis and run a single spatio-temporal forward pass:

$$z_{t:t+K-1} \;=\; \tfrac{1}{2}\big(\mathcal{E}(I_{\text{pbr},\,t:t+K-1}) + \mathcal{E}(I_{\text{irr},\,t:t+K-1})\big), \tag{4}$$

which is then mapped by the DiT and decoded as in Eq. (3). The DiT activates its native interleaved spatial and temporal attention layers, so it can use neighbouring frames as context for ambiguous regions while keeping the per-frame structure of $\hat{I}_{\text{pbr}}$ intact. For longer trajectories we apply $\mathcal{R}_\theta$ in a sliding window of K frames with an overlap of $K-1$ (a stride of one) and resolve overlapping predictions by averaging or by discarding non-anchor frames.

Training objective. We supervise $\mathcal{R}_\theta$ against the ground-truth RGB frames $I^\star$ with a pixel and a perceptual term, applied independently to every frame of the clip:

$$\mathcal{L} \;=\; \lambda_2 \left\|\hat{I}_{\text{rgb}} - I^\star\right\|_2^2 \;+\; \lambda_p\, \mathcal{L}_{\text{LPIPS}}\big(\hat{I}_{\text{rgb}},\, I^\star\big). \tag{5}$$

The ℓ_2 term anchors the predicted radiance to the correct mean and matches global tone, exposure, and colour balance, but on its own tends to produce blurry predictions wherever the input-to-RGB mapping is multi-modal (e.g. specular highlights or missing regions). The LPIPS term (Zhang et al., 2018), a weighted distance in the feature space of a frozen VGG-16 (Simonyan & Zisserman, 2015) model, counters this by allowing the network to commit to plausible high-frequency content instead of averaging over it, and is the main driver of sharp textures and crisp edges. We use $\lambda_2{=}\lambda_p{=}1.0$ throughout, and the same loss is applied in both the image and video stages; only the DiT $\mathcal{F}_\theta$ receives gradient updates.

4.2 Paired Data Curation

Training $\mathcal{R}_\theta$ requires paired samples $\{(\hat{I}_{\text{pbr}},\, \hat{I}_{\text{irr}},\, I^\star)\}$ in which the two input buffers come from the Gaussian ray tracer and the target $I^\star$ is a high-quality photoreal frame of the same view. Such triples are not available off-the-shelf, so we build a curated mixture of synthetic and real data.

Synthetic data. For full ground-truth control we render 975 procedurally varied scenes through a path tracer, each from a fixed 93-frame camera trajectory and under 20 distinct illumination settings (10 high-contrast and 10 low-contrast environment maps), and pair every rendered frame with the matching PBR and irradiance buffers produced by the Gaussian Ray Tracer on the same scene. This yields 1.81 M paired samples spanning a wide range of materials, geometries, and light transports.

Real data. To close the synthetic–real gap we also curate 955 real captures from the DL3DV (Ling et al., 2024) benchmark, reconstructed with our Gaussian representation and optimized for material (Section 3.2) and envmap (Appendix D.3). Each scene contributes on average $\sim$300 frames at its native illumination, for a total of around 290 K paired samples. Real captures introduce reconstruction artifacts (sparsity at the periphery, view-dependent specularities, dynamic exposure) that are absent from the synthetic split and are essential for adapting the renderer to in-the-wild content.

Three-stage curriculum. We train the DiT $\mathcal{F}_\theta$ in three stages, with the encoders and decoder held frozen throughout. *Stage 1:* we use single-frame samples ($K{=}1$) drawn exclusively from the synthetic split and optimize the loss in Eq. (5). This stage exposes $\mathcal{F}_\theta$ to a wide range of materials and illuminations under clean ground truth and adapts the diffusion prior to the buffer statistics. *Stage 2:* we keep $K{=}1$ but switch the data source to the real split. This stage narrows the domain gap and teaches $\mathcal{F}_\theta$ to handle the reconstruction artifacts characteristic of in-the-wild captures. *Stage 3:* we move to $K{=}5$ fixed-length clips on the real split, which exercises the DiT's temporal attention layers under the same per-frame supervision. This stage teaches $\mathcal{F}_\theta$ to leverage temporal context

for flicker-free streaming without drifting from the per-frame solution learned in Stages 1–2. The combined training set comprises approximately 2.1 M paired samples.

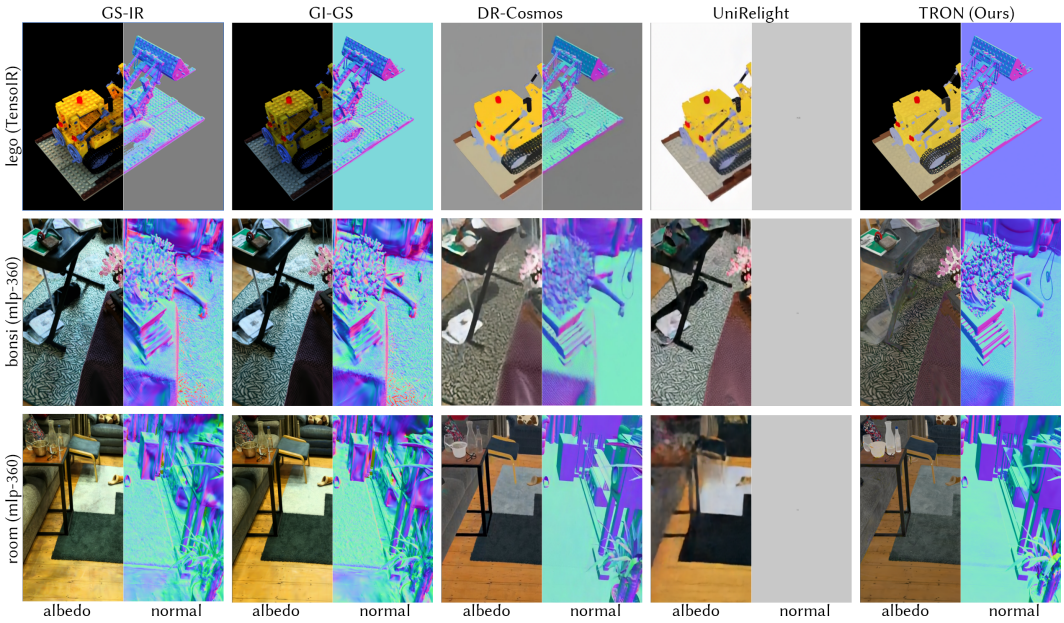


Figure 2: **Qualitative Material Decompositions Comparison**. As inverse-rendering is an ill-posed problem, previous Gaussian based methods (Chen et al., 2025a; Liang et al., 2023b) that rely on direct optimization typically suffer from shadows baked into albedo. Neural rendering methods (Liang et al., 2025a; He et al., 2025) either sacrifice spatial resolution or suffer from multi-view inconsistencies. By baking priors as a volumetric feature field, TRON can consolidate inconsistencies from high resolution priors.

Method	TensorIR				TRON-Synth Objects				TRON-Synth Scenes			
	Albedo			Normals	Albedo			Normals	Albedo			Normals
	PSNR↑	SSIM↑	LPIPS↓	MAE↓	PSNR↑	SSIM↑	LPIPS↓	MAE↓	PSNR↑	SSIM↑	LPIPS↓	MAE↓
GS-IR (Liang et al., 2023b)	30.06	0.928	0.107	5.32	19.85	0.831	0.305	10.85	19.82	0.841	0.316	11.69
GI-GS (Chen et al., 2025a)	29.64	0.927	0.107	5.25	19.10	0.797	0.316	10.17	18.84	0.814	0.338	11.22
GaussianShader (Jiang et al., 2023)	23.55	0.832	0.168	14.90	9.23	0.356	0.533	34.15	8.54	0.376	0.540	37.37
GS-ID (*paper*) (Du et al., 2024)	33.49	0.952	0.079	4.602	–	–	–	–	–	–	–	–
DR-Cosmos (Liang et al., 2025a) *(2D prior)*	33.22	0.959	0.073	5.34	**30.99**	**0.940**	**0.156**	7.19	**31.11**	**0.940**	**0.131**	5.59
UniRelight (He et al., 2025)	32.69	0.947	0.089	–	27.36	0.854	0.301	–	27.98	0.858	0.273	–
Ours	**34.10**	**0.964**	**0.069**	**4.37**	29.65	0.933	0.250	4.97	31.09	0.925	0.261	3.58

Table 1: **Quantitative Material Decompositions Comparison**. By lifting priors to 3D and baking them into the Gaussian field, TRON outperforms all Gaussians variants, while remaining on par with the priors of DR-Cosmos (Liang et al., 2025a) despite consolidating high frequency multiview inconsistencies.

5 Experiments

We evaluate TRON on a mix of synthetic and real-world datasets. Our main task objective is to relight a scene under *novel* illumination. In our quantitative evaluation (§5.2), we focus on fidelity of material decomposition, realism of relighting, and rendering speed. Our qualitative results support these findings and highlight the novel capabilities of our method, difficult to achieve with other techniques such as: harmonization, precise control of shadows under dynamic scene motion, material editing and high quality interactive preview.

We further extend our evaluation with additional ablation studies and comparisons within Appendix A.

5.1 Experimental Setup

Benchmark. To evaluate material decomposition and relighting, we extend the benchmark of previous works (Liang et al., 2025a) and generate a synthetic test set: *TRON-Synth*, using the same path tracing pipeline as in our synthetic training data generation (See Appendix E) but with a different

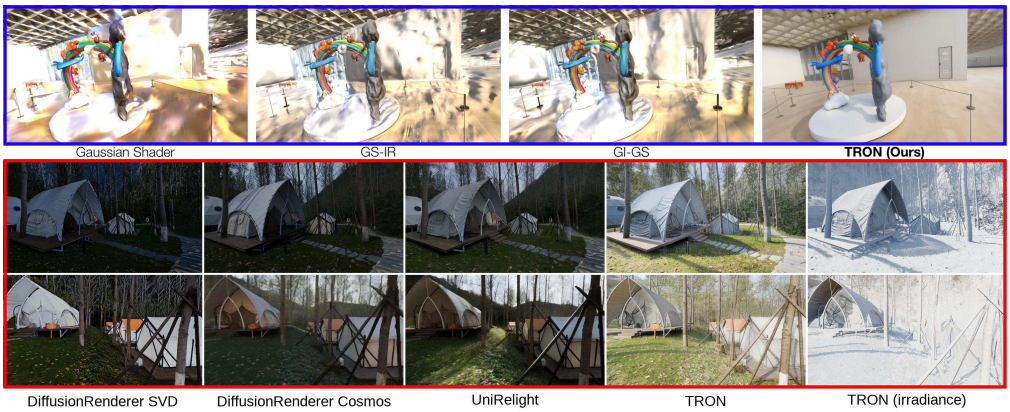


Figure 3: **Qualitative Relighting Comparisons**. Top: Gaussian Relighting baselines. To achieve interactive rendering, previous Gaussian relighting methods apply split-sum (Karis, 2013), which convolves lighting and does not model occlusions. TRON separates PBR and irradiance channels to avoid these pitfalls. Bottom: Neural Rendering baselines. Contemporary neural methods rely on priors embedded in their weights to hallucinate realistic shadows, while TRON respects explicit 3D structure (last column).

Method	TRON-Synth Objects			TRON-Synth Scenes		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
GaussianShader (Jiang et al., 2023)	12.93	0.723	0.385	14.12	0.799	0.420
GS-IR (Liang et al., 2023b)	18.50	0.812	0.307	17.84	0.836	0.336
GI-GS (Chen et al., 2025a)	19.07	0.803	0.318	19.86	0.853	0.328
UniRelight (He et al., 2025)	19.47	0.839	0.261	21.95	0.877	0.258
DR-SVD (Liang et al., 2025a)	20.17	0.862	0.279	22.50	0.899	0.234
DR-Cosmos (Liang et al., 2025a)	18.92	0.851	0.248	22.04	0.895	0.220
TRON - Ray Tracer only (PBR)	15.76	0.807	0.296	17.83	0.866	0.269
TRON - Neural Renderer, Phase 1	20.95	0.860	0.313	21.48	0.884	0.294

Table 2: **Novel Relighting Comparison.** Computed on 40 TRON-Synth scenes, averaged over 5 per-scene environment maps, all test views. This benchmark isolates *phase 1* of our neural-renderer curriculum, which is trained purely on synthetic data. View along qualitative Fig 10 & Fig 11.



Figure 4: **Glossy Surfaces:** TRON can add reflections absent in PBR guidance inputs through the generative capabilities of the neural renderer.

set of 3D objects, environment maps and render settings. Our test dataset contains 15 object-centric scenes and 25 multi-object scenes, each rendered with a path tracer under 5 sampled illuminations, and paired with ground-truth material decompositions. For completeness with prior art, we also evaluate on the TensorIR (Jin et al., 2023) dataset, consisting of 4 shapes with their materials, each relit with 5 different envmaps. For evaluation on real data we denote *TRON-DL3DV-33* as a dataset consisting of 33 scenes from the 2nd partition of DL3DV-10K (Ling et al., 2024). These scenes were never used during training, and each one is paired with 3 different rotated environment maps from a curated collection of envmaps consisting of different contrast levels.

Baselines. We evaluate TRON against both Gaussian-based relighting methods and neural image/video models. Relightable NeRF variants are excluded, as they are not well-suited for dynamic scene motion or local edits, our target application area. We select GaussianShader (Jiang et al., 2023), GS-IR (Liang et al., 2023b) and GI-GS (Chen et al., 2025a) to represent relightable reconstruction methods, given their ability to model occlusions, extend to scene scale, and public code available. In addition, published results from GS-ID (Du et al., 2024) are included, which to the best of our knowledge, is the only Gaussian based method using material priors for relighting, but has no public codebase. Contrary to previous benchmarks conducted by these works, which focus on inverting the original scene illumination, here our focus is on measuring the accuracy of novel relighting. Furthermore, we compare against DiffusionRenderer (Liang et al., 2025a) and UniRelight (He et al., 2025), which represent neural rendering and relighting, respectively.

5.2 Evaluation and Comparisons

Material Decomposition. We use synthetic datasets to evaluate the fidelity of our material decomposition in Tb.1, reporting PSNR, SSIM and LPIPS (Zhang et al., 2018) for rendered albedo (scaled



Figure 5: **Results Gallery.** Top & middle: show TRON results for Novel Relighting on real-world captures. Bottom: editing global light intensity (left) and rendering shadows of a dynamic harmonized object (right).



Figure 6: **Harmonization Comparisons**: Our method dynamically computes the highlight and shadow conditions of the scene for a harmonized object.

according to common practice, see Appendix F.3) and Mean Angular Error (MAE) for rendered normals. By utilizing robust priors (Liang et al., 2025a), TRON outperforms Gaussian relighting baselines that optimize for material decomposition by matching original scene appearance, often resulting in artifacts like shadows baked into albedo (Fig.2) and incorrect normals. Conversely, our method bakes priors into an existing high-quality 3D scaffold, avoids resolution limitations and consolidates any multi-view inconsistencies exhibited by neural approaches, explaining our improved performance over this class of approaches. In contrast to previous neural techniques, our baked materials are 3D consistent and can be rendered in real-time from any view under dynamic scene changes.

Relighting Synthetic Scenes. As no comprehensive benchmark with ground truth illumination exists for *full* scenes, we evaluate in synthetic settings. To that end, we suggest TRON-Synth (§5.1), which also includes high contrast envmaps that cast long shadows, a property uncommon in previous relighting benchmarks. Tb. 2 reports novel relighting metrics against all baselines. The synthetic benchmark has ground truth for relit views so reconstruction metrics may apply directly. Importantly, the neural renderer improves on the ray-traced buffer it consumes proving the neural renderer is not bounded by the approximation error of the shading model, and can mimic the full high quality path-tracer used to generate TRON-Synth. Here we deliberately measure the neural renderer immediately post phase 1, as afterwards we adapt it to the real domain (§4.2), which is our target setting. As Tb. 2 shows, our method is on par with the neural variants. Note that reconstruction metrics weakly reward shadow accuracy, as cast shadows span few pixels per view, even though our method is the only one to correctly generate the long shadows. We show these differences in complementary qualitative comparisons of Fig 10 and Fig 11.

Relighting Real World Scenes. To assess photorealism, we devise an explainable agent-based metric that uses a VLM to perform pairwise comparisons between frames relit by our method and the baselines with approximate metrics on TRON-DL3DV-33. See Appendix F.4 for prompt and examples. For each scene and relighting condition, we sample 5 views, evaluating the win rate of our method against every method across all 495 pairs per method with 3 different seeds for standard deviation. Our results are summarized in Tb. 3. As a sanity check, we also

compare our method against the original frames, and validate real frames win 89% of the time. Our method significantly outperforms Gaussian-based baselines, achieving win rate of 97%-99%. While these methods can achieve competitive PSNR on original scene reconstruction with optimized materials, their entangled materials (Fig.2) result in degraded appearance under novel lighting (Fig.3) of in-the-wild captured scenes. For neural inverse rendering techniques, trained to generate highly realistic output, we observe mixed results with win rates of 47%-64%. This performance of more computationally heavy neural techniques comes at a cost of lower controllability: i.e., with strong shadow mismatches for different rendered trajectories of the same scene. See Fig.7.

V.S. Baseline	Ours Win Rate	FPS ↑	First-frame Latency ↓
GaussShader	99.73 ± 0.12	21.06	47.5 ms
GI-GS	97.71 ± 0.42	3.64	275 ms
GS-IR	97.31 ± 0.31	15.87	63 ms
DR-SVD	64.92 ± 0.62	0.54	44.7 s
DR-Cosmos	50.03 ± 0.65	0.96	59.7 s
UniRelight	47.27 ± 1.41	0.13	450.0 s
TRON - Fast Preview *(Ray Traced PBR only)*	89.36 ± 1.49	**51.81**	**19.3 ms**
TRON - Full Quality *(Neu. Renderer enabled)*	–	1.6	625 ms
GT (real photo)	10.84 ± 0.62	–	–

Table 3: **Relighting A/B Photorealism Comparison**: TRON Full Quality realism win-rate (%) vs. all baselines and preview mode, as judged by a GPT 5.1 agent on TRON-DL3DV-33. The last column measures the elapsed time for feedback, e.g. latency from the moment rendering begins until the first frame appears.

5.3 Additional Applications.

Combining a 3D grounded representation (§3.1) and an informed neural renderer (§4) unlocks a variety of additional applications: **(i) Harmonization:** objects can be inserted into the scene by stacking the Gaussians of an object with the those of another scene. The neural renderer relights both at the same time, harmonizing the object into the scene (Fig. 6). **(ii) Material Editing:** by manipulating the G-buffer values of existing Gaussians, the PBR and irradiance control signals can be quickly re-rendered, letting the neural renderer visualize different appearances immediately. **(iii) Dynamic Shadows:** rendering shadows of moving objects under changing lighting conditions (Fig. 5) is naturally achieved as irradiance buffers are rendered from geometry on the fly, then processed by the neural renderer. **(iv) Dynamic Lights:** by rotating the envmap, shadows change their location faithfully according to scene geometry, by re-rendering the PBR and irradiance buffers.

Interactive application results are best viewed with accompanying video and project page. TRON is unique in its ability to render shadows for moving objects and under changing illumination, with immediate feedback: Gaussian methods rely on baked visibility or simplified ambient occlusion, and neural methods have no persistent 3D representation in which objects can be moved. Our setup allows for quick preview (Tb. 3) through real-time iterations with the ray tracer PBR mode and neural renderer disabled (FPS 51.81, see §3.3), or slower, higher quality interactive mode with the neural renderer image-mode enabled (FPS 1.6, see §4.1). For full quality videos, once camera trajectories are decided, we can render temporally stable offline videos by processing a sliding window of K frames at once, and averaging same-frame from overlapping windows (see §4.1).

6 Conclusion & Limitations

We presented TRON, a novel rendering pipeline combining Gaussian ray tracing with neural rendering to achieve a fast, controllable and high quality renderings of real-world captures. Our proposed neural renderer is able to add photorealistic novel illuminations on top of existing reconstructions, paving the way for numerous interactive applications. Despite the benefits of combining neural rendering with photorealistic synthesis, constraining the model to preserve the physical attributes of the input scans remains an open challenge. For example, the model may introduce global tone shifts or alter the original scene illumination. In addition, our model relies on input reconstructions, degrading when training view coverage is sparse and signals become noisy. Specifically, background regions are under-determined and are prone to temporal artifacts. Lastly, our control signals do not fully express all phenomena, including glass materials, subsurface scattering, and local lights. As a result, the neural renderer may introduce details that are not explicitly controllable, such as the inter-reflections shown in Fig. 4.

Our framework opens exciting avenues for future work, particularly in enforcing better object identity preservation within the neural renderer, further achieving real time performance and accurately modeling complex, 3D-aware phenomena like refractions and better control of near field illumination.

AI USE STATEMENT

Large models are used in this work in two distinct capacities. First, a vision-language model is a component of the evaluation itself: the explainable agent-based photorealism metric reported in Tb. 3 is computed by prompting GPT 5.1 to perform pairwise comparisons between frames relit by our method and by each baseline, with the prompt, the permutation and seeding protocol, and representative examples given in Appendix F.4. Second, AI assistance was used while preparing several of the diagrams and figures in this paper. The method, its implementation, the experimental design and all claims are the authors' own. Every reported number was produced by the authors' own runs and verified by them, and the authors take full responsibility for the content of this paper.

ETHICS STATEMENT

This work renders and relights static reconstructions of scenes, and does not involve human subjects, user studies, or personally identifying data. The real-world captures we evaluate on come from the publicly released DL3DV-10K and Mip-NeRF 360 datasets and are used under their respective licenses. The remaining benchmark scenes are synthetic and were generated by us. Photorealistic relighting of captured scenes carries the same dual-use concern as other generative rendering methods, in that it can be used to produce imagery that appears to have been captured under conditions that never occurred. Our pipeline mitigates this only partially: because shading is driven by an explicit 3D representation and an explicit environment map, the illumination applied to a scene is recorded rather than implicit. As we note in §6, inter-reflections and other effects outside the split-sum model are synthesised by the neural renderer and are therefore plausible rather than physically measured, and should not be read as evidence about the captured scene.

REPRODUCIBILITY STATEMENT

The full architecture, training curriculum and losses are specified in §4 and §3.3, with the ray tracer and reconstruction stages given as explicit algorithms in Appendix D.1 and Appendix D.3. Complete hyperparameters, hardware and training times are reported in the *Hyperparameters, Hardware & Training Time* section of the Appendix, and the construction of the training corpus and of the evaluation benchmarks, including the per-scene environment maps, is described in Appendix E. The evaluation protocols behind every table - the albedo scaling used for material decomposition, the agent-based photorealism comparison, and the VBench criteria - are given in Appendix F.3, Appendix F.4 and Appendix F.5 respectively. Our implementation is prepared for public release, and we will additionally release the synthetic benchmark so that other methods can be evaluated against the baselines reported here.

REFERENCES

Hadi Alzayer, Philipp Henzler, Jonathan T. Barron, Jia-Bin Huang, Pratul P. Srinivasan, and Dor Verbin. Generative multiview relighting for 3d reconstruction under extreme illumination variation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 10933–10942, June 2025.

Haiyang Bai, Jiaqi Zhu, Songru Jiang, Wei Huang, Tao Lu, Yuanqi Li, Jie Guo, Runze Fu, Yanwen Guo, and Lijun Chen. Gare: Relightable 3d gaussian splatting for outdoor scenes from unconstrained photo collections. *ICCV*, 2025.

Sai Praveen Bangaru, Lifan Wu, Tzu-Mao Li, Jacob Munkberg, Gilbert Bernstein, Jonathan Ragan-Kelley, Frédo Durand, Aaron Lefohn, and Yong He. SLANG.D: Fast, Modular and Differentiable Shader Programming. *ACM Trans. Graph.*, 42(6), 2023.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

Zoubin Bi, Yixin Zeng, Chong Zeng, Fan Pei, Xiang Feng, Kun Zhou, and Hongzhi Wu. Gs3: Efficient relighting with triple gaussian splatting. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687576. URL https://doi.org/10.1145/3680528.3687576.

Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stadhouderskade 592, Amsterdam, 2024. URL http://www.blender.org.

Christophe Bolduc, Yannick Hold-Geoffroy, Zhixin Shu, and Jean-François Lalonde. Gaslight: Gaussian splats for spatially-varying lighting in hdr. *ArXiv*, 2025.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

Chris Careaga and Yağız Aksoy. Physically controllable relighting of photographs. In *Proc. SIGGRAPH*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Hongze Chen, Zehong Lin, and Jun Zhang. Gi-gs: Global illumination decomposition on gaussian splatting for inverse rendering. In *ICLR*, 2025a.

Yihang Chen, Qianyi Wu, Mengyao Li, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Fast feedforward 3d gaussian splatting compression. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Hoon-Gyu Chung, Seokjun Choi, and Seung-Hwan Baek. Differentiable inverse rendering with interpretable basis brdfs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982.

Rishit Dagli, Donglai Xiang, Vismay Modi, Charles Loop, Clement Fuji Tsang, Anka He Chen, Anita Hu, Gavriel State, David IW Levin, and Maria Shugrina. Vomp: Predicting volumetric mechanical property fields. *arXiv preprint arXiv:2510.22975*, 2025.

Michael Deering, Stephanie Winner, Bic Schediwy, Chris Duffy, and Neil Hunt. The triangle processor and normal vector shader: a vlsi system for high performance graphics. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, pp. 21–30, New York, NY, USA, 1988. Association for Computing Machinery. ISBN 0897912756. doi: 10.1145/54852.378468. URL https://doi.org/10.1145/54852.378468.

Isaac Deutsch, Nicolas Moënne-Loccoz, Gavriel State, and Zan Gojcic. Ppisp: Physically-plausible compensation and control of photometric variations in radiance field reconstruction, 2026. URL https://arxiv.org/abs/2601.18336.

Jan-Niklas Dihlmann, Arjun Majumdar, Andreas Engelhardt, Raphael Braun, and Hendrik P.A. Lensch. Subsurface scattering for gaussian splatting. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 121765–121789. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/dc72529d604962a86b7730806b6113fa-Paper-Conference.pdf.

Kang Du, Zhihao Liang, and Zeyu Wang. Gs-id: Illumination decomposition on gaussian splatting via diffusion prior and parametric light source optimization, 2024. URL https://arxiv.org/abs/2408.08524.

Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10786–10796, 2021.

Yotam Erel, Rishabh Dabral, Vladislav Golyanik, Amit H. Bermano, and Christian Theobalt. Practilight: Practical light control using foundational diffusion models. *ACM Transactions on Graphics (TOG)*, 44(6):1–11, 2025.

Jiahui Fan, Fujun Luan, Jian Yang, Milos Hasan, and Beibei Wang. Rng: Relightable neural gaussians. *Proceedings of CVPR 2025*, 2025.

Frédéric Fortier-Chouinard, Zitian Zhang, Louis-Etienne Messier, Mathieu Garon, Anand Bhattad, and Jean-François Lalonde. Spotlight: Shadow-guided object relighting via diffusion. In *International Conference on 3D Vision (3DV)*. IEEE, March 2026.

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*, pp. 73–89. Springer, 2024.

Shrisudhan Govindarajan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Radiant foam: Real-time differentiable ray tracing. *arXiv:2502.01157*, 2025.

Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing. In *CVPR*, 2025.

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

Antoine Guédon, Diego Gomez, Nissim Maruani, Bingchen Gong, George Drettakis, and Maks Ovsjanikov. Milo: Mesh-in-the-loop gaussian splatting for detailed and efficient surface reconstruction. *ACM Transactions on Graphics*, 2025. URL https://anttwo.github.io/milo/.

Kyu Beom Han, Jaeyoon Kim, Woo Jae Kim, Jinhwan Seo, and Sung-eui Yoon. Radiometrically consistent gaussian surfels for inverse rendering. *arXiv preprint arXiv:2603.01491*, 2026.

Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022.

Kai He, Ruofan Liang, Jacob Munkberg, Jon Hasselgren, Nandita Vijaykumar, Alexander Keller, Sanja Fidler, Igor Gilitschenski, Zan Gojcic, and Zian Wang. Unirelight: Learning joint decomposition and synthesis for video relighting. *arXiv preprint arXiv:2506.15673*, 2025.

Yumeng He, Yunbo Wang, and Xiaokang Yang. Metags: A meta-learned gaussian-phong model for out-of-distribution 3d scene relighting. *arXiv preprint arXiv:2405.20791*, 2024.

Jan Held, Renaud Vandeghen, Sanghyun Son, Daniel Rebain, Matheus Gadelha, Yi Zhou, Ming C. Lin, Marc Van Droogenbroeck, and Andrea Tagliasacchi. Triangle splatting+: Differentiable rendering with opaque triangles. *arXiv*, 2025.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.

Kaiwen Jiang, Jia-Mu Sun, Zilu Li, Dan Wang, Tzu-Mao Li, and Ravi Ramamoorthi. Differentiable light transport with gaussian surfels via adapted radiosity for efficient relighting and geometry reconstruction. *ACM Transactions on Graphics (TOG)*, 44(6), December 2025.

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023.

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

Haian Jin, Yuan Li, Fujun Luan, Yuanbo Xiangli, Sai Bi, Kai Zhang, Zexiang Xu, Jin Sun, and Noah Snavely. Neural gaffer: Relighting any object via diffusion. In *Advances in Neural Information Processing Systems*, 2024.

James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986. ISSN 0097-8930. doi: 10.1145/15886.15902. URL https://doi.org/10.1145/15886.15902.

Joanna Kaleta, Kacper Kania, Tomasz Trzciński, and Marek Kowalski. LumiGauss: Relightable Gaussian Splatting in the Wild. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Brian Karis. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3): 1, 2013.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Ye Keyang, Hou Qiming, and Zhou Kun. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH Conference Proceedings, Denver, CO, United States, July 28 - August 1, 2024*, 2024.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. Spotlight Presentation.

Peter Kocsis, Vincent Sitzmann, and Matthias Niessner. Intrinsic image diffusion for indoor single-view material estimation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Shuichang Lai, Letian Huang, Jie Guo, Kai Cheng, Bowen Pan, Xiaoxiao Long, Jiangjing Lyu, Chengfei Lv, and Yanwen Guo. Glossygs: Inverse rendering of glossy objects with 3d gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2025. doi: 10.1109/TVCG.2025.3547063.

Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. Envidr: Implicit differentiable renderer with neural environment lighting. *arXiv preprint arXiv:2303.13022*, 2023a.

Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025a.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. *arXiv preprint arXiv:2311.16473*, 2023b.

Zhihao Liang, Hongdong Li, Kui Jia, Kailing Guo, and Qi Zhang. GUS-IR: Gaussian Splatting With Unified Shading for Inverse Rendering . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 47(10):8364–8378, October 2025b. ISSN 1939-3539. doi: 10.1109/TPAMI.2025.3578416. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2025.3578416.

Chih-Hao Lin, Jia-Bin Huang, Zhengqin Li, Zhao Dong, Christian Richardt, Tuotuo Li, Michael Zollhöfer, Johannes Kopf, Shenlong Wang, and Changil Kim. IRIS: Inverse rendering of indoor scenes from low dynamic range images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Yehonathan Litman, Fernando De la Torre, and Shubham Tulsiani. Lightswitch: Multi-view relighting with material-guided diffusion. In *ICCV*, 2025a.

Yehonathan Litman, Or Patashnik, Kangle Deng, Aviral Agrawal, Rushikesh Zawar, Fernando De la Torre, and Shubham Tulsiani. Materialfusion: Enhancing inverse rendering with material diffusion priors. In *3DV*, 2025b.

Linjie Lyu, Valentin Deschaintre, Yannick Hold-Geoffroy, Miloš Hašan, Jae Shin Yoon, Thomas Leimküehler, Christian Theobalt, and Iliyan Georgiev. Intrinsicedit: Precise generative image manipulation in intrinsic space. *ACM Transactions on Graphics*, 44(4), 2025.

Nadav Magar, Amir Hertz, Eric Tabellion, Yael Pritch, Alex Rav-Acha, Ariel Shamir, and Yedid Hoshen. Lightlab: Controlling light sources in images with diffusion models. 2025. doi: 10.1145/3721238.3730696.

Alexander Mai, Dor Verbin, Falko Kuester, and Sara Fridovich-Keil. Neural microfacet fields for inverse rendering. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 408–418, 2023. URL https://api.semanticscholar.org/CorpusID:257900923.

Alexander Mai, Peter Hedman, George Kopanas, Dor Verbin, David Futschik, Qiangeng Xu, Falko Kuester, Jon Barron, and Yinda Zhang. Ever: Exact volumetric ellipsoid rendering for real-time view synthesis, 2024. URL https://arxiv.org/abs/2410.01804.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Vismay Modi, Nicholas Sharp, Or Perel, Shinjiro Sueda, and David IW Levin. Simplicits: Mesh-free, geometry-agnostic elastic simulation. *ACM Transactions on Graphics (TOG)*, 43(4):1–11, 2024.

Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024.

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

Patrick Noras, Jun Myeong Choi, Didier Stricker, Pieter Peers, and Roni Sengupta. Gains: Gaussian-based inverse rendering from sparse multi-view captures, 2025. URL https://arxiv.org/abs/2512.09925.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Karran Pandey, Anita Hu, Clement Fuji Tsang, Or Perel, Karan Singh, and Maria Shugrina. Painting with 3d gaussian splat brushes. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2016. ISBN 0128006455.

Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei Efros, and George Drettakis. Multi-view relighting using a geometry-aware network. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, 38(4), July 2019. URL http://www-sop.inria.fr/reves/Basilic/2019/PGZED19.

Yohan Poirier-Ginter, Alban Gauthier, Julien Philip, Jean-François Lalonde, and George Drettakis. A Diffusion Approach to Radiance Field Relighting using Multi-Illumination Synthesis. *Computer Graphics Forum*, 2024. ISSN 1467-8659. doi: 10.1111/cgf.15147.

Yohan Poirier-Ginter, Jeffrey Hu, Jean-François Lalonde, and George Drettakis. Editable Physically-based Reflections in Raytraced Gaussian Radiance Fields. In *SIGGRAPH Asia 2025 - 18th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia*, Hong Kong, Hong Kong SAR China, December 2025. doi: 10.1145/3757377.3763971. URL https://inria.hal.science/hal-05306634.

Christian Reiser, Stephan Garbin, Pratul P. Srinivasan, Dor Verbin, Richard Szeliski, Ben Mildenhall, Jonathan T. Barron, Peter Hedman, and Andreas Geiger. Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis. *SIGGRAPH*, 2024.

Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jian Zhang, Bin Zhou, Errui Ding, and Jingdong Wang. Gir: 3d gaussian inverse rendering for relightable scene factorization. *IEEE Transactions on Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Troy James Sobotka. *AgX Tonemapper*, 2024. URL https://github.com/sobotka/AgX.

Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. Svg-ir: Spatially-varying gaussian splatting for inverse rendering. *Proceedings of CVPR 2025*, 2025.

Eric Veach. *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162.

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. *arXiv preprint arXiv: 2503.01774*, 2025a.

Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Tong Wu, Jiali Sun, Yu-Kun Lai, Yuewen Ma, Leif Kobbelt, and Lin Gao. Deferredgs: Decoupled and relightable gaussian splatting with deferred shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47:6307–6319, 2024. URL https://api.semanticscholar.org/CorpusID:269148633.

Zirui Wu, Jianteng Chen, Laijian Li, Shaoteng Wu, Zhikai Zhu, Kang Xu, Martin R. Oswald, and Jie Song. 3d gaussian inverse rendering with approximated global illumination, 2025c. URL https://arxiv.org/abs/2504.01358.

Chen Xi, Peng Sida, Yang Dongchen, Liu Yuan, Pan Bowen, Lv Chengfei, and Zhou. Xiaowei. Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination. *arxiv: 2404.11593*, 2024.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint arXiv:2311.12198*, 2023.

Xiaoyan Xing, Konrad Groh, Sezer Karagolu, Theo Gevers, and Anand Bhattad. Luminet: Latent intrinsics meets diffusion models for indoor scene relighting. *arXiv preprint arXiv:2412.00177*, 2024.

Xiaoyan Xing, Philipp Henzler, Junhwa Hur, Runze Li, Jonathan T. Barron, Pratul P. Srinivasan, and Dor Verbin. Gr3en: Generative relighting for 3d environments. *ArXiv*, abs/2601.16272, 2026. URL https://api.semanticscholar.org/CorpusID:284982082.

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 (ECCV)*, 2022.

Kai Ye, Chong Gao, Guanbin Li, Wenzheng Chen, and Baoquan Chen. Geosplatting: Towards geometry guided gaussian splatting for physically-based inverse rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28991–29000, 2025.

Keyang Ye, Qiming Hou, and Kun Zhou. Progressive radiance distillation for inverse rendering with gaussian splatting, 2024. URL https://arxiv.org/abs/2408.07595.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024.

Chong Zeng, Yue Dong, Pieter Peers, Youkang Kong, Hongzhi Wu, and Xin Tong. Dilightnet: Fine-grained lighting control for diffusion-based image generation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024a.

Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. RGB↔X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024b. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657445. URL https://doi.org/10.1145/3641519.3657445.

Fan Zhang, Shulin Tian, Ziqi Huang, Yu Qiao, and Ziwei Liu. Evaluation agent: Efficient and promptable evaluation framework for visual generative models. *arXiv preprint arXiv:2412.09645*, 2024.

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)*, 2023a.

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 10728–10745, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1cce374fec7829c24a97ae0e91cd0999-Paper-Conference.pdf.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Tianyuan Zhang, Zhengfei Kuang, Haian Jin, Zexiang Xu, Sai Bi, Hao Tan, He Zhang, Yiwei Hu, Milos Hasan, William T. Freeman, Kai Zhang, and Fujun Luan. RelitLRM: Generative relightable radiance for large reconstruction models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=3Oli4u6q3p.

Wenyuan Zhang, Jimin Tang, Weiqi Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. MaterialRefGS: Reflective gaussian splatting with multi-view consistent material inference. In *Advances in Neural Information Processing Systems*, 2025c.

X. Zhang et al. Su-rgs: Relightable 3d gaussian splatting from sparse views under unconstrained illuminations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2025d.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor. *ACM Transactions on Graphics (TOG)*, 40:1 – 18, 2021b. URL https://api.semanticscholar.org/CorpusID:235313848.

Youjia Zhang, Teng Xu, Junqing Yu, Yuteng Ye, Yanqing Jing, Junle Wang, Jingyi Yu, and Wei Yang. Nemf: Inverse volume rendering with neural microflake field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22919–22929, 2023b.

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

Yuxuan Zhang, Katarína Tóthová, Zian Wang, Kangxue Yin, Haithem Turki, Riccardo de Lutio, Yen-Yu Chang, Or Litany, Sanja Fidler, and Zan Gojcic. Diffusionharmonizer: Bridging neural reconstruction and photorealistic simulation with online diffusion enhancer. *arXiv preprint arXiv:2602.24096*, 2026. URL https://arxiv.org/abs/2602.24096.

Yang Zhou, Songyin Wu, and Ling-Qi Yan. Unified gaussian primitives for scene representation and rendering. 2024.

Yongyang Zhou, Fanglue Zhang, Zichen Wang, and Lei Zhang. Rtr-gs: 3d gaussian splatting for inverse rendering with radiance transfer and reflection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 6888–6897, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755197. URL https://doi.org/10.1145/3746027.3755197.

Zuo-Liang Zhu, Jian Yang, and Beibei Wang. Gaussian splatting with discretized sdf for relightable assets. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2025a.

Zuo-Liang Zhu, Jian Yang, and Beibei Wang. Gs-ror$\hat{2}$: Bidirectional-guided 3dgs and sdf for reflective object relighting and reconstruction. *ACM Transactions on Graphics (TOG)*, 2025b. doi: 10.1145/3759248.

APPENDIX

A Additional Results **19**

A.1 Ablation Studies 19

A.2 Additional Comparisons 22

B Extended Related Works **25**

C Preliminaries **26**

C.1 3D Gaussian Ray Tracing 26

C.2 Physically Based Rendering 26

D Implementation Details **26**

D.1 Prior-Informed Reconstruction. 26

D.2 Gaussian Scene Rendering 27

D.3 Envmap Optimization 31

D.4 Hyperparameters, Hardware & Training Time 31

E Dataset Construction **32**

E.1 Data Pipeline 32

E.2 Synthetic Benchmark Assets 32

F Evaluation Details **33**

F.1 Inference Mode 33

F.2 Baseline Details 33

F.3 Albedo Scaling 34

F.4 A/B Photorealism Additional Details 34

F.5 VBench Metrics 36

A Additional Results

A.1 Ablation Studies

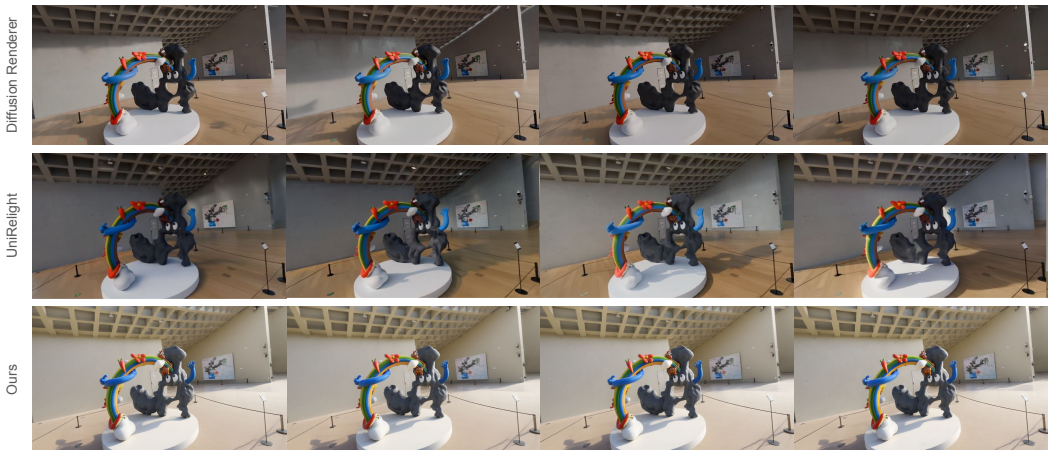


Figure 7: **Random seed sensitivity comparison**: We evaluate the sensitivity of each method to random initialization by running each model four times on the same input image with different random seeds. This test measures whether a model consistently follows the input conditioning or instead hallucinates lighting and shadow details. UniRelight (He et al., 2025) and Diffusion Renderer (Liang et al., 2025a) are both multi-step diffusion models, making their outputs sensitive to the initial noise. In practice, we observe large variations across seeds: the cast shadow on the floor and the shadow on the background wall can change substantially in shape and appearance. This indicates that these methods perform a significant amount of generative reconstruction when producing shadows. By contrast, our method remains consistent across random seeds, owing to its single-step image-to-image design and strong input conditioning.

Opponent (compared v.s. TRON Full Quality)	TRON Win Rate
TRON - Fast Preview *(Ray Traced PBR only)*	89.36 ± 1.49
TRON - Neural Renderer, phase 1 ckpt (32K iters)	94.95 ± 0.61
TRON - Neural Renderer, phase 2 ckpt (64K iters)	82.90 ± 0.71
TRON - Full Quality, PBR guidance only (irrad. zeroed)	81.55 ± 0.51
TRON - Full Quality, irrad. only (PBR zeroed)	98.65 ± 0.23
GT (real photo)	10.84 ± 0.62

Table 4: **Ablations - A/B photorealism test of Full Pipeline vs. variants**: presenting the win-rate (%) of the full quality TRON against each ablation variant, using the same procedure of Tb. 3.

Variant (vs. TRON PBR output)	Variant Win Rate
NR, phase 1 ckpt (32K iters)	53.40 ± 1.17
NR, phase 2 ckpt (64K iters)	60.47 ± 0.62
NR, full (phase 3, 96K iters)	**89.36 ± 1.49**
NR, PBR only (irrad. zeroed)	80.13 ± 0.31
NR, irrad. only (PBR zeroed)	23.18 ± 0.49

Table 5: **Ablations - A/B photorealism test of Ray Tracer only vs. variants**: presenting the win-rate (%) the ray tracer's PBR output against each TRON ablation variant, isolating how much each neural renderer component contibutes to added realism. Test conducted using protocol of Tb. 3.

Tb. 4 presents an ablation on the photorealism criteria, of the full quality TRON pipeline, trained with all phases of the curriculum. Following Tb. 3, the first section of the table highlights the improvements of the combined neural renderer and ray tracer over the ray tracer alone. The section ablates the neural renderer training curriculum, showing that *without* real-domain adaptation (phase 1 ckpt) and temporal consistency amortization (phase 2 ckpt), the quality of realism is reduced. The third section highlights why both neural renderer input conditions are required. Without PBR and irradiance, shadow and texture awareness is reduced, which consequentially hurts photorealism (see also Fig. 8 & Fig. 9 for complementary qualitative results). The last line is a control experiment that verifies the photorealism evaluation protocol by measuring the TRON full pipeline against GT images, where as expected, the vast majority of the time GT images are judged as more photoreal.

Tb. 5 has an additional ablation from a different angle: rather than measuring the full pipeline against each variant, we examine how much each variant improves over the non-neural ray tracer. Since phase 1 of the neural renderer is trained on synthetic data, it contributes little to photorealism. The advantage emerges after domain adaptation (phase 2 ckpt) and temporal consistency amortization is introduced (phase 3 ckpt). For uni-signal neural renderer variants, the irradiance channel carries no texture and is the only variant that loses to the tracer given that most material information is lost.

See Fig. 8 & Fig. 9 for a sweep ablation on weighting the neural renderer latents, showing how the different guidance signals affect the final outcome.

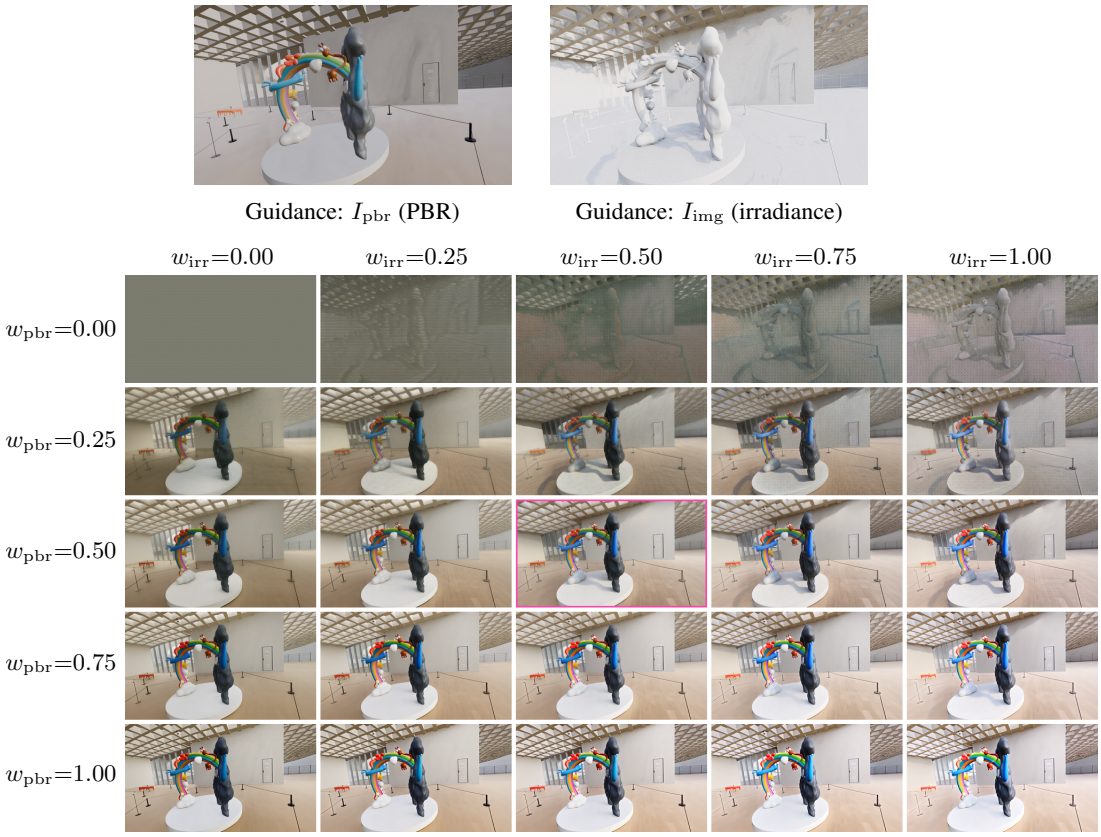


Figure 8: **Latent-fusion weight sweep.** The Neural Renderer fuses the two input modalities by $I_{\text{fused}} = w_{\text{pbr}}\, I_{\text{pbr}} + w_{\text{irr}}\, I_{\text{img}}$, where I_{pbr} and I_{img} are the standardized VAE latents of the rendered PBR and irradiance images shown at the top. The centre cell (outlined in magenta) is the default fusion settings $(w_{\text{pbr}}, w_{\text{irr}}) = (0.5, 0.5)$. The two axes carry distinct shading roles: increasing w_{irr} (left$\rightarrow$right) progressively reintroduces cast shadows into the rendering, while increasing w_{pbr} (top$\rightarrow$bottom) transitions from diffusive to specular highlights.

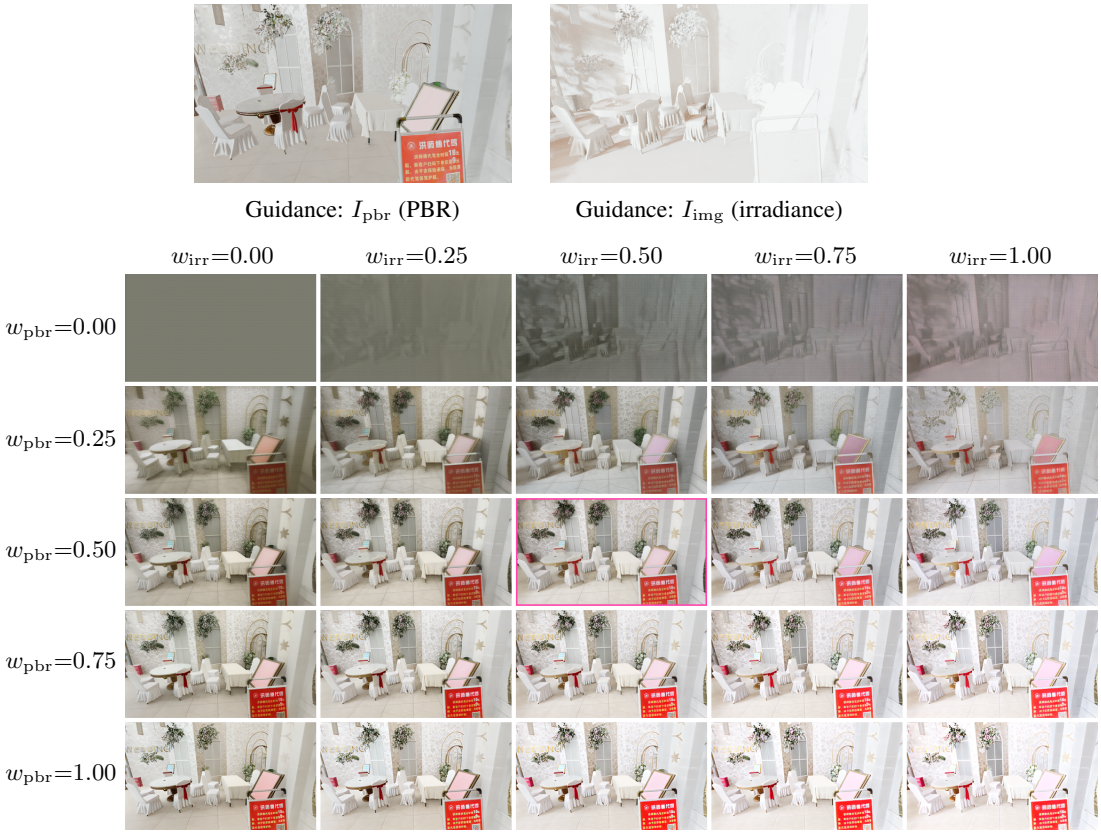


Figure 9: **Additional latent-fusion weight sweep.** Same setup as Fig. 8: rows vary $w_{\text{pbr}} \in [0, 1]$ top to bottom, columns vary $w_{\text{irr}} \in [0, 1]$ left to right, and the magenta-outlined centre cell is default fusion weights $(0.5,\ 0.5)$. The two guidance inputs I_{pbr} and I_{img} are shown at the top. The same dichotomy holds: shadows naturally form with low w_{irr}, while adhering closer to the guidance images as the irradiance weight increases.

A.2 Additional Comparisons

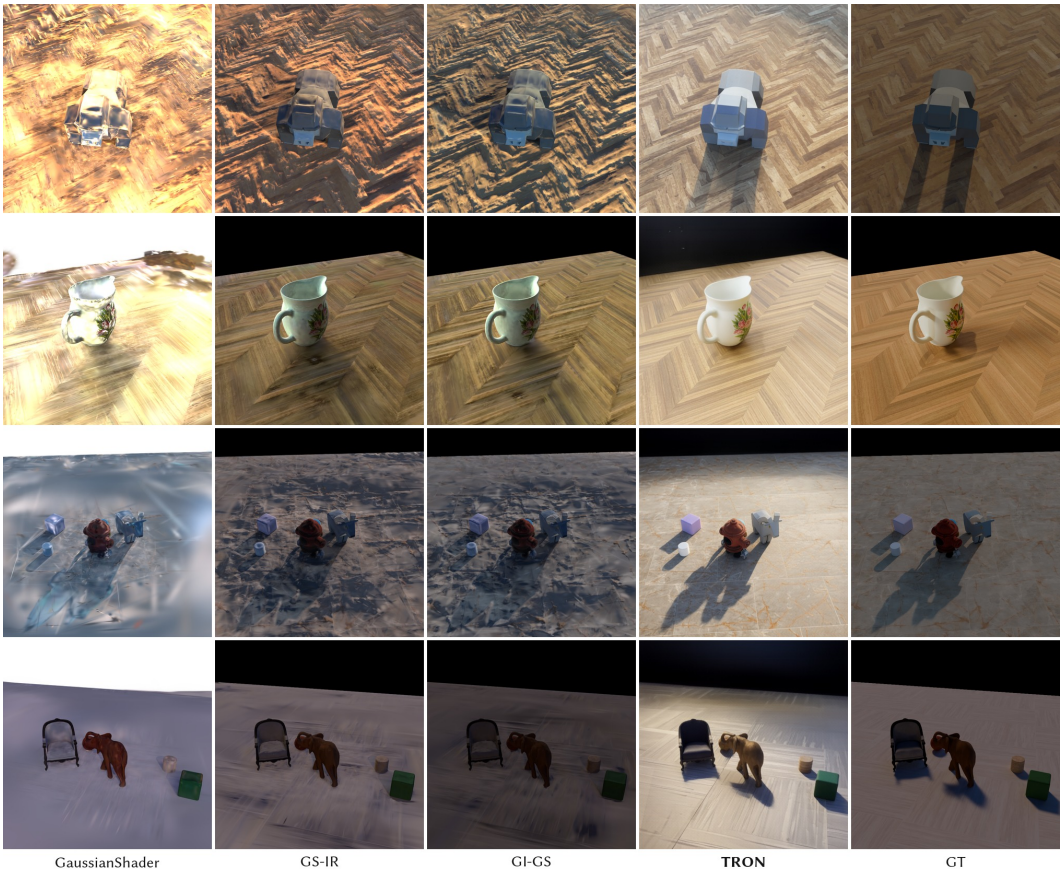


Figure 10: **Comparison to Gaussian based methods on the Relighting Synthetic Benchmark.** Here we qualitatively measure the final checkpoint of the neural renderer. All methods were presented different illumination conditions during reconstruction. Top example: baselines exhibit baked shadows, in incorrect location and direction. Bottom row: long shadows due to novel relighting are not modeled by baselines, compared to TRON. Evidentially, TRON is the only method that respects the correct physical position of the shadows due to incorporating the irradiance channel condition.

Shadow Sensitivity. We conduct a shadow sensitivity test on random seeds and present its results within Fig. 7. In this test we use a different random seed per frame to test for the consistency of each method. Neural methods like Diffusion-Renderer and UniRelight are highly sensitive to the input seed, and accordingly although they appear real, their shadow shape and location change greatly between seeds and clips. TRON relies only on irradiance guidance to place its shadows and thus maintains consistent shadows (as the neural renderer is a feed-forward network that *does not* rely on noise).

VBench. In Tb. 6, we validate our method on the Video Generative Models benchmark of VBench (Zhang et al., 2024), with 5 criteria that allow evaluation directly on rendered videos without an accompanying prompt. We compare smooth trajectories, using the first 100 frames per relit variation of a scene in TRON-DL3DV-33, resized to 1920×1080, rendered by our method and neural baselines, totalling in comparison over 99 unique clips. Our method shows competitive performance with these offline, full denoising pipelines, while exhibiting interactive speed, controllability, and opening up new dynamic applications for captured 3D worlds. We also point out differences in how our method and Liang et al. (2025a); He et al. (2025) handle shadows: the group of neural renderers relies on prior knowledge embedded in their weights, whereas ours is guided by irradiance signals grounded in a 3D world. Consequently, these methods have limited control over shadows, possibly hallucinating unseen occluders (Fig. 6) or silhouettes not faithful to subtle scene geometry (Fig. 11).



Figure 11: **Comparison to Neural based methods on the Relighting Synthetic Benchmark.** Qualitatively measuring the final checkpoint of the neural renderer, complementary to Table 2. By adhering to irradiance cues rendered according to the geometry of the scene, TRON models the correct silhouttes, faithful to the scene geometry. In comparison, other neural baselines condition on priors embedded in their weights, resulting in shadows that do not adhere to the intricate geometries of the scene.

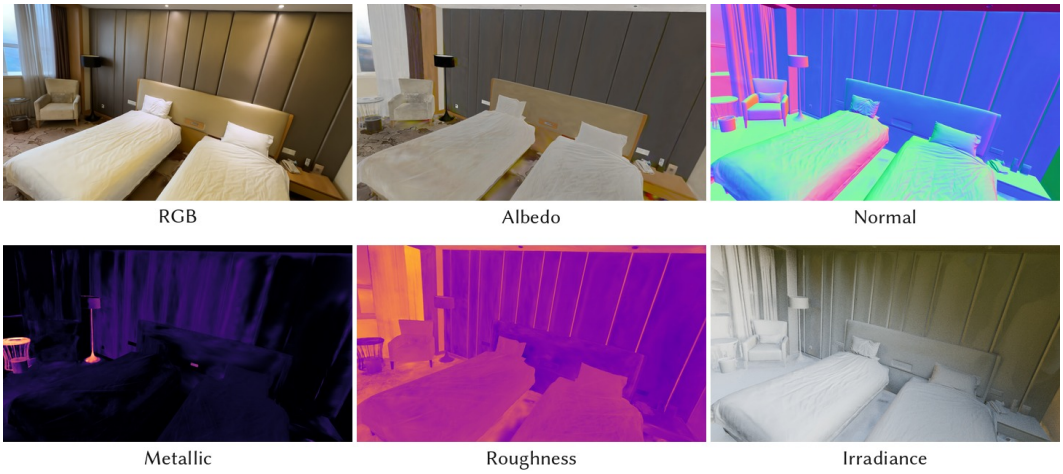


Figure 12: **Additional material decomposition results.** Showing G-buffer representation baked into the Gaussian field, each channel rendered individually.



Figure 13: **Comparisons and Ablation of Tracer v.s. Full Pipeline on mip-360 (Barron et al., 2022).** Left to right, columns 1-3: With shadows and highlights baked into the albedo, artifacts emerge during novel relighting. Column 4: Even with a clean albedo map, the shading model is limited in expressivity. Right column: the full pipeline benefits from the realism the neural renderer introduces.

Method	Metrics				
	Motion Smoothness	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality
DR-SVD	0.9820	0.8658	0.9088	0.4891	0.6263
DR-Cosmos	0.9839	0.8617	0.9141	0.4707	0.5895
UniRelight	**0.9865**	**0.8738**	0.9160	0.4680	0.5522
TRON (Ours)	0.9787	0.8737	**0.9190**	**0.4959**	**0.6956**

Table 6: **Quantitative Video-Quality Assessment:** measured via VBench (Zhang et al., 2024) on relit clips from TRON-DL3DV-33. Higher is better.

Additional Relighting & Material Decomposition Comparisons. Fig. 13 presents an additional qualitative comparison between Gaussian baselines, and TRON with and without the neural renderer on the mip-360 (Barron et al., 2022) dataset. Additional relighting qualitative comparisons over TRON-Synth are provided in Fig. 10 and Fig. 11. Fig.12 presents additional material decomposition results on the DL3DV-10K dataset.

B Extended Related Works

The *Lagrangian* nature of particle-based primitives, such as Gaussians (Kerbl et al., 2023; Mai et al., 2024; Huang et al., 2024; Wu et al., 2025b; Moenne-Loccoz et al., 2024) and triangles (Govindarajan et al., 2025; Held et al., 2025), makes them especially appealing for simulating digital worlds. Lighting and material decompositing techniques targeting implicit representations (Boss et al., 2021; Verbin et al., 2022; Zhang et al., 2021b; Mai et al., 2023; Jin et al., 2023; Zhang et al., 2023b; Yao et al., 2022) and signed distance functions (Zhang et al., 2021a; Lin et al., 2025; Liang et al., 2023a; Zhang et al., 2023a; 2022) remain difficult to apply to interactive editing. While mesh extraction methods offer a solution, they require repeated conversions (Munkberg et al., 2022; Hasselgren et al., 2022; Wang et al., 2023), risk losing delicate details, or retain baked lighting (Guédon & Lepetit, 2024; Yu et al., 2024; Reiser et al., 2024; Guédon et al., 2025). Thus, we focus on extending dynamic lighting to Gaussian-based representations.

Relightable Gaussians. A growing body of work addresses this limitation by optimizing Gaussian scene representations with partial physically based rendering (PBR) properties (Gao et al., 2024; Liang et al., 2023b; Wu et al., 2025c; Jiang et al., 2023; Chen et al., 2025a; Shi et al., 2025; Liang et al., 2025b; Ye et al., 2024), emphasizing materials (Chung et al., 2025; Zhang et al., 2025c) or inter-reflections (Keyang et al., 2024; Poirier-Ginter et al., 2025; Gu et al., 2025; Han et al., 2026). Many approaches resort to simplified domains like one-light-at-a-time (OLAT) setups (Bi et al., 2024; Fan et al., 2025; Dihlmann et al., 2024; He et al., 2024), single specular objects (Jiang et al., 2023; Zhu et al., 2025b; Zhou et al., 2025; Lai et al., 2025; Gu et al., 2025; Wu et al., 2024; Zhu et al., 2025a; Ye et al., 2025; Sun et al., 2025), or specific outdoor scenes (Bai et al., 2025; Kaleta et al., 2025; Zhang et al., 2025d). While recent methods successfully leverage material priors from generative models (Zeng et al., 2024b; Eftekhar et al., 2021; Ke et al., 2024; Noras et al., 2025) for decomposition (Du et al., 2024; Chen et al., 2025a), dynamic illumination remains a bottleneck: by employing baked shadows or uniformly sampled shadow rays, these techniques fail to model long silhouettes and require reoptimization for dynamic geometry. Jiang et al. (Jiang et al., 2025) and Zhou et al. (Zhou et al., 2024) set an elegant stage for differentiable global illumination, but are efficiency-limited to single objects or slow offline optimization, respectively. Orthogonal to our work, Poirier-Ginter et al. (Poirier-Ginter et al., 2025) model indirect reflections but omit shadows and relighting. A universal limitation of this line of work is the reliance on constrained shading models, inherently precluding true photorealism. In contrast, our approach narrows this domain gap using a neural renderer while natively supporting dynamic shadows, respecting the material composition of the scene.

Image-Space Neural Rendering. A parallel line of work tries to achieve relighting through learning, for example by applying large generative models in image space (Magar et al., 2025; Erel et al., 2025; Kocsis et al., 2024; Xi et al., 2024; Fortier-Chouinard et al., 2026; Xing et al., 2024; Zeng et al., 2024a; Jin et al., 2024; Zhang et al., 2025a), with some methods leveraging multi-view images (Zhang et al., 2025b; Philip et al., 2019; Litman et al., 2025a), or videos (He et al., 2025; Xing et al., 2026). By training paired forward and inverse renderers, RGB↔X approaches (Liang et al., 2025a; Zeng et al., 2024b; Lyu et al., 2025) map rendered images to G-buffers and vice versa, achieving impressive intrinsic decomposition and relighting. While the generative nature of these approaches enables them to hallucinate realism, it limits their controllability and comes at a high computational cost. DIFIX3D (Wu et al., 2025a) and DiffusionHarmonizer (Zhang et al., 2026) present efficient pipelines for single-step inference by distilling pretrained diffusion models. Inspired by these approaches, we inject explicit 3D knowledge as conditions for neural rendering.

Hybrid Approaches. Some multi-view reconstruction approaches independently relight each input image and lift to 3D (Alzayer et al., 2025; Bolduc et al., 2025; Poirier-Ginter et al., 2024), or predict per-view materials (Litman et al., 2025b) and optimize a mesh reconstruction (Hasselgren et al., 2022). While promising, these works are limited in scene scale and suffer from hallucinated shadows and artifacts from inconsistent view predictions.

C PRELIMINARIES

C.1 3D GAUSSIAN RAY TRACING

3DGRT (Moenne-Loccoz et al., 2024) models scene geometry and appearance using a set of semi-transparent *radiating* Gaussian particles. The spatial influence of each Gaussian at a point $\mathbf{x} \in \mathbf{R}^3$ is defined by the kernel function:

$$G_i(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right), \tag{6}$$

where the mean position μ_i and the anisotropic covariance matrix Σ_i are optimizable, allowing the underlying Gaussian distribution to flexibly take the approximated form of a 2D surflet or 3D volume. To evaluate the radiance along a given ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, 3DGRT relies on an approximation of the Volumetric Rendering Equation:

$$L(\mathbf{o},\mathbf{d}) \approx \sum_{i=1}^{N} \mathbf{c}_i(\mathbf{d})\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{7}$$

Here, opacity $\alpha_i = \sigma_i G_i(\mathbf{x})$ is linearly approximated at $\mathbf{x}(t)$, the point along ray $\mathbf{r}$ maximizing the Gaussian response $G_i(\mathbf{x})$.

C.2 PHYSICALLY BASED RENDERING

The rendering equation (Kajiya, 1986) formulates the relationship between the outgoing radiance and incident light:

$$L_o(\mathbf{x},\omega_o) = \int_\Omega f_r(\mathbf{x},\omega_o,\omega_i)L_i(\mathbf{x},\omega_i)|\mathbf{n}\cdot\omega_i|\,d\omega_i \tag{8}$$

L_o is the outgoing radiance at surface point $\mathbf{x}$ in direction ω_o. The right hand side is an integral of the product of the incident radiance, $L_i(\omega_i)$ (which includes a visibility term) from direction ω_i and the BSDF, $f_r(\mathbf{x},\omega_o,\omega_i)$, which models the surface reflectance. The integration domain is the hemisphere, Ω, around the surface normal$\mathbf{n}$. Since this equation is infinitely recursive and hence intrectable, it is common practice to evaluate it with Monte Carlo estimators (Veach, 1998; Pharr et al., 2016), or simplify it with approximations like Split-Sum (Karis, 2013).

We leverage a Cook–Torrance microfacet BRDF (Cook & Torrance, 1982),

$$f(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) \;=\; \frac{D\,G\,F}{4\,(\boldsymbol{\omega}_o\cdot\mathbf{n})\,(\boldsymbol{\omega}_i\cdot\mathbf{n})}, \tag{9}$$

whose specular term carries the reflection and highlight cues that drive the Neural Renderer towards photorealistic relighting. For efficiency, we preintegrate the specular component with split-sum (Karis, 2013; Munkberg et al., 2022):

$$L(\boldsymbol{\omega}_o) \;\approx\; \underbrace{\int_\Omega L_i(\boldsymbol{\omega}_i)\,D(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)\,(\boldsymbol{\omega}_i\cdot\mathbf{n})\,d\boldsymbol{\omega}_i}_{\text{prefiltered envmap}} \cdot \underbrace{\int_\Omega f(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)\,(\boldsymbol{\omega}_i\cdot\mathbf{n})\,d\boldsymbol{\omega}_i}_{\text{BRDF LUT}}. \tag{10}$$

The first integral collapses to a roughness-mipped query of the envmap; the second is precomputed once into a 2D lookup table over $(\mathbf{n}\cdot\boldsymbol{\omega}_o,\, r)$.

D IMPLEMENTATION DETAILS

D.1 PRIOR-INFORMED RECONSTRUCTION.

Algorithm 1 details the proposed two-stage prior-informed reconstruction procedure. For all reconstruction objectives, we employ the standard 3DGS loss function (Kerbl et al., 2023) (with $\lambda = 0.8$):

$$\mathcal{L}_{\text{rec}}(I,\hat{I}) = \lambda\,\|I-\hat{I}\|_1 + (1-\lambda)\big(1-\text{SSIM}(I,\hat{I})\big). \tag{11}$$

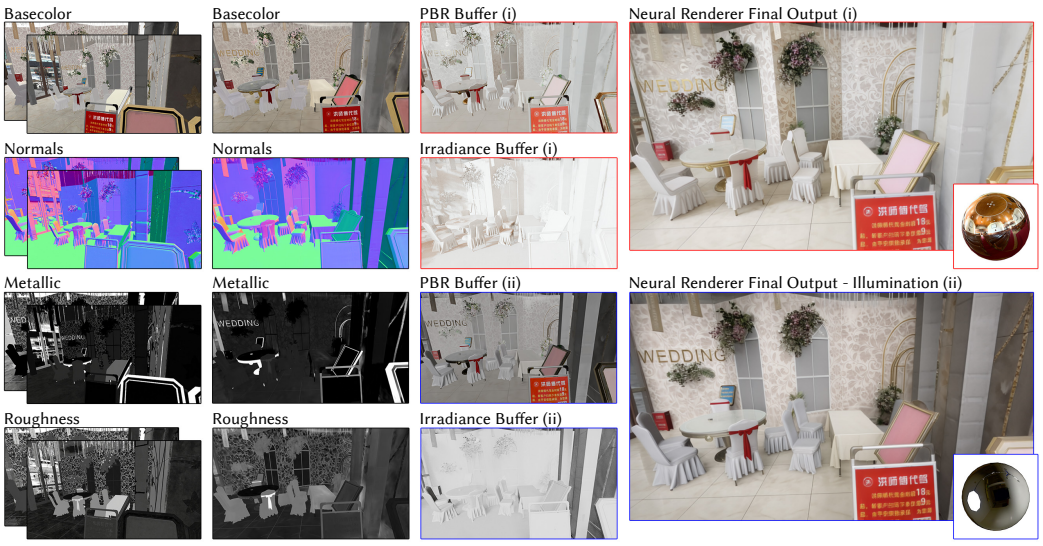


Figure 14: **Overall Pipeline:** (a) We extract per-frame material priors (b), and bake them into a 2D Gaussian representation as additional channels. (c) Given any novel lighting (envmap insets, rightmost), we can compute shading passes, PBR and irradiance (d), and use them to render a high-quality relit image with a Neural Renderer.

In Algorithm 1, $\mathcal{R}$ denotes the rendering operator defined according to the configurations listed in Table 7.

Contrary to previous works, our setting avoids using additional regularizations like depth-distortion and normal-depth consistency. Following common practice (Kerbl et al., 2023; Moenne-Loccoz et al., 2024), we use 3rd order Spherical Harmonics to model baked radiance in the reconstruction. Similar to concurrent works (Poirier-Ginter et al., 2025), we recognize that highly specular regions often suffer from degraded geometry, as very few views provide supervision for highlights. Still, rather than using lower order SH or replacing $\hat{I}_{rgb}$ with a non-directional diffusive prior, we found $\hat{I}_{rgb}$ to contain more crucial shade and texture cues to supervise the initial reconstruction.

Our prior prediction requirements are flexible. In this work we preprocess each view individually with an intrinsic decomposition model model (Liang et al., 2025a) for basecolor, metallic, roughness, normals and the model of (Zeng et al., 2024b) for irradiance. Flaws introduced during this preprocess phase are later mitigated when lifting to 3D and processing through our Neural Renderer. While priors can be extracted with video based versions of DiffusionRenderer (Liang et al., 2025a) for improved view consistency, we found that comes at the cost of lower prior resolution, crucial to high frequency channels such as normals.

D.2 Gaussian Scene Rendering

The rendering of an image pixel $\hat{I}_p$ for a camera with origin $\mathbf{o} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{R}^3$ is defined by tracing a ray $r(\tau) = \mathbf{o} + \tau\mathbf{d}, \tau \geq 0$ and accumulating contributions from Gaussian particles along the ray in a sorted order:

$$\hat{I}_p = \sum_{j=1}^{m} \mathbf{f}_j(\mathbf{d})\ \alpha_j \prod_{k=1}^{j-1}(1-\alpha_k). \tag{12}$$

Here α_j is the opacity of the jth depth-order Gaussian, computed from the maximal Gaussian kernel response along the ray and modulated by the learned particle opacity parameter σ_j. The function $\mathbf{f}_j(\mathbf{d})$ represents a general per-particle attribute to be rendered, potentially dependent on the viewing direction $\mathbf{d}$, such as view-dependent color, base color, surface normals, or other geometric or material properties. Table 7 summarizes the rendering quantities obtained by instantiating Eq. (12). Notably, our adoption of the 2D Gaussian formulation (Huang et al., 2024; Gu et al., 2025) yields an explicit and well-defined per-particle *geometric* normal, which we leverage throughout our method.

Rendered buffer	$\mathbf{f}_j(\mathbf{d})$	Description
RGB image	$\mathrm{SH}(\mathbf{d}; \boldsymbol{\beta}_j)$	View-dependent color represented using spherical harmonics coefficients
Base color	$\mathbf{b}_j \in \mathbb{R}^3$	Intrinsic diffuse (albedo) color
Roughness	$r_j \in \mathbb{R}$	Surface roughness parameter
Metallicity	$m_j \in \mathbb{R}$	Material metallicity parameter
Normal	$\mathbf{n}_j = \mathbf{R}_j \mathbf{e}_3$	Per-particle geometric normal derived from Gaussian orientation

Table 7: Instantiation of Eq. (12) for different rendered buffers. Each choice of $\mathbf{f}_j(\mathbf{d})$ corresponds to rendering a different scene attribute.

Deferred shading. We build upon 3DGRT (Moenne-Loccoz et al., 2024), and for efficiency we replace the forward integrator, which shades every particle independently with a deferred volumetric integrator (Deering et al., 1988). The sorted slab traversal is reused to gather a per-pixel G-buffer (position, normal, basecolor, roughness, metallic) by rendering each material channel through Eq.7 (Zhu et al., 2025b; Liang et al., 2025b). The shading-point depth is taken from a median cutoff: we accumulate transmittance along the ray and adopt the depth of the first particle past which it falls below 0.5. The shading normal defaults to the geometric normal. The geometric normal is derived directly from the rotation matrix R:

$$\mathbf{n}^{geo} = \tilde{\mathbf{n}}\mathbf{R}^\top \quad \text{where } \tilde{\mathbf{n}} = [0 \quad 0 \quad 1]^\top . \tag{13}$$

We evaluate the direct illumination integral using split-sum (Karis, 2013; Munkberg et al., 2022) as described in Eq. 10. The renderer produces linear HDR values, which are mapped to the unit range with AgX tonemapping (Sobotka, 2024) before being passed to the Neural Renderer. We denote the resulting image $\hat{I}_{pbr}$.

The split-sum approximation does not model occlusions, so we complement $\hat{I}_{pbr}$ with a ray-traced irradiance map, $\hat{I}_{irr}$, that integrates incident radiance over the visible upper hemisphere. At each shading point we trace K shadow rays via multiple importance sampling (Veach, 1998), combining cosine sampling and light importance sampling using a piecewise-constant 2D distribution sampling technique (Pharr et al., 2016), and query each ray for visibility against $\mathcal{G}$. Visibility is the complement of transmittance, and transmittance along a ray is order-invariant in the volumetric formulation (Gao et al., 2024), so we trace shadow rays in an out-of-order routine that avoids the per-hit insertion-sort otherwise required for compositing (this is implemented with any-hit shader). When a ray escapes the field, we fetch the envmap by differentiable bilinear texture lookup.

Algorithm 1 Prior-Informed Gaussian Reconstruction

Require: Input images and cameras $\{(I_i, C_i)\}_{i=1}^{N}$, intrinsic decomposition model ϕ
Ensure: Gaussian scene $\mathcal{G} = \{A_j; B_j\}_{j=1}^{m}$

Intrinsic decomposition
for $i = 1, \ldots, N$ **do**
    $(I_n^i, I_b^i, I_r^i, I_m^i) \leftarrow \phi(I_i)$

Stage 1: Baked appearance reconstruction
Initialize Gaussian geometry and appearance parameters $\{A_j\}_{j=1}^{m}$
for $t = 1, \ldots, T_1$ **do**
    Sample a training view (I_i, C_i)
    Render RGB image:

$$\hat{I}_i \leftarrow \mathcal{R}(\{A_j\}, C_i, \mathbf{f}^{\text{rgb}})$$

    Render normal map:

$$\hat{I}_n^i \leftarrow \mathcal{R}(\{A_j\}, C_i, \mathbf{f}^{\text{normal}})$$

    Compute loss:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{rec}}(\hat{I}_i, I_i) + \lambda_n \mathcal{L}_{\text{rec}}(\hat{I}_n^i, I_n^i)$$

    Update $\{A_j\}$ using gradient descent

Stage 2: Material optimization
Freeze $\{A_j\}$
Initialize material parameters $\{B_j\}_{j=1}^{m}$
for $t = 1, \ldots, T_2$ **do**
    Sample a training view (I_i, C_i)
    Render material channels:

$$(\hat{I}_b^i, \hat{I}_r^i, \hat{I}_m^i) \leftarrow \mathcal{R}(\{A_j\}, \{B_j\}, C_i, \mathbf{f}^{\text{mat}})$$

    Compute loss:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{rec}(\hat{I}_b^i, I_b^i) + \mathcal{L}_{rec}(\hat{I}_r^i, I_r^i) + \mathcal{L}_{rec}(\hat{I}_m^i, I_m^i)$$

    Update $\{B_j\}$ using gradient descent

Output
$\mathcal{G} \leftarrow \{A_j; B_j\}_{j=1}^{m}$
return $\mathcal{G}$

Algorithm 2 Stage 3: Envmap Optimization

Require: Input images $\{(I_i, C_i)\}_{i=1}^{N}$, pre-trained Gaussian scene $\mathcal{G} = \{A_j; B_j\}_{m=1}^{M}$, irradiance prior model ψ
Ensure: Optimized Environment Map E and global scale σ
// Pre-processing: Extract Irradiance Priors
for $i = 1, \ldots, N$ **do**
  $I_e^i \leftarrow \psi(I_i)$ ▷ Estimate pseudo-GT irradiance

Stage 3a: Environment Map Refinement
Freeze geometry $\{A_j\}$ and materials $\{B_j\}$; Initialize cubemap E
for $t = 1, \ldots, T_{3a}$ **do**
  Sample view (I_i, C_i)
  Render G-buffer: $\{\mathbf{p}, \mathbf{n}, \mathbf{m}\} \leftarrow \mathcal{R}_{\text{deferred}}(\mathcal{G}, C_i)$
  Sample K directions $\{\omega_k\}_{k=1}^{K} \sim p(\omega)$ using MIS on E and BRDF
  Compute visibility $v_k \leftarrow \text{Trace}(\mathcal{G}, \mathbf{p}, \omega_k)$ ▷ Any-hit occlusion test
  $\hat{I}^i_{\text{irr}} = \frac{1}{K} \sum_{k=1}^{K} W_k^{\text{MIS}} \frac{E(\omega_k) \cdot v_k \cdot (\mathbf{n} \cdot \omega_k)}{p(\omega_k)}$ ▷ Monte Carlo Estimator
  $\mathcal{L}_{\text{stage3a}} = \|\hat{I}^i_{\text{irr}} - I^i_e\|_1$
  $E \leftarrow E - \eta \nabla_E \mathcal{L}_{\text{stage3a}}$
  Update mipmap pyramid $\mathcal{M}(E)$ ▷ Synchronize for next iteration

Stage 3b: Global Intensity Alignment
Freeze $\{A_j, B_j, E\}$; Initialize $\sigma \leftarrow 1.0$
for $t = 1, \ldots, T_{3b}$ **do**
  Sample view (I_i, C_i)
  Compute mipmap pyramid $\mathcal{M}(\sigma \cdot E)$ ▷ Reflect current intensity scale
  Render specular component using split-sum:

$$\hat{I}_{\text{spec}} \leftarrow \mathcal{R}_{\text{split-sum}}(\mathcal{M}, \mathbf{n}, \mathbf{m}) \quad \text{where level } \ell \propto \text{roughness}$$

  Compute shadowed diffuse: $\hat{I}_{\text{diff}} \leftarrow \text{MC_Estimate}(\mathcal{M}, \mathbf{n}, \mathbf{m}, \text{vis} = v_k)$
  $\hat{I}^i_{\text{pbr}} = \hat{I}_{\text{diff}} + \hat{I}_{\text{spec}}$
  $\mathcal{L}_{\text{stage3b}} = \mathcal{L}_{\text{rec}}(\hat{I}^i_{\text{pbr}}, I_i)$
  $\sigma \leftarrow \sigma - \eta \nabla_\sigma \mathcal{L}_{\text{stage3b}}$
return $E_{\text{final}} \leftarrow \sigma \cdot E$

D.3 ENVMAP OPTIMIZATION

In order to train the neural renderer, we require paired of inputs $(\hat{I}_{\text{pbr}}, \hat{I}_{\text{irr}})$ and their matching high quality rgb image $I^{\star}$. For real world data, $I^{\star}$ is only available without $(\hat{I}_{\text{pbr}}, \hat{I}_{\text{irr}})$ that match the lighting conditions of the rgb image. In order to implement the real data curation described in Section 4.2, we apply the method described in Section 3.3 in a differentiable manner. We apply the pipeline of (Zeng et al., 2024b) to obtain a pseudo-gt irradiance estimation I_e, and supervise against our own computation of irradiance based on reconstructed scene geometry, optimizing the intensity values of the envmap in the process. To better facilitate high dynamic range, envmaps are queried with an exponential activation. For this data generation step only, we also mask out the sky from I_e and $\hat{I}_{\text{irr}}$ using an off-the-shelf segmentation model (Carion et al., 2025), as sky is emissive and Gaussian particles often exist in that region. The entire stage is described in Algorithm 2. Finally, in order to calibrate the envmap intensity scale, we run a small number of iterations to supervise a single scale parameter using rendered $\hat{I}_{\text{pbr}}$ and $I^{\star}$. Note that at inference time, we do not require the backwards pass and apply only the forward mode of Section 3.3.

D.4 HYPERPARAMETERS, HARDWARE & TRAINING TIME

Ray Tracer. The tracer part of the pipeline extends from Moenne-Loccoz et al. (Moenne-Loccoz et al., 2024).

We use the recent Slang (Bangaru et al., 2023) based implementation to obtain auto-differentiable kernels for the deferred integrator computing irradiance and bsdf. TRON uses the 2D kernel shape of (Moenne-Loccoz et al., 2024), embedded in a quad for fast ray-particle intersections, which gives us a natural definition of a geometric normal. We modify the integrator to skip Gaussians we hit from the back side, as we found these to yield inconsistent surface approximations.

For loss computation (Eq. 11), we use a weight of $\lambda = 0.2$ for SSIM and $1 - \lambda = 0.8$ for L1 reconstruction loss. For Tensor-IR and the synthetic-benchmark, we also apply a mask loss with $\lambda_o = 0.2$. We train each scene with 30K steps for the radiance field reconstruction phase, and 7K additional steps for lifting G-buffer channels. For envmap optimization during data generation, we apply 4K steps to learn the envmap through the irradiance prediction. For fair comparison, we do not leverage the recent additions of Markov Chain Monte Carlo (Kheradmand et al., 2024) and PPISP (Bilateral Grid Filtering) (Deutsch et al., 2026) which improve the reconstruction quality. When transitioning from volumetric radiance optimization to G-buffer optimization, we initialize Gaussian basecolor from the rgb. During shading, we use Multiple Importance Sampling with 2 spp for interactive preview, and 64 spp for high quality frames. Where learned envmaps are used, we fix them to resolution of 512×512, and use the logic of (Munkberg et al., 2022) to construct differentiable mipmaps efficiently.

Unless stated otherwise, our remaining hyperparameters follow the default configuration of (Moenne-Loccoz et al., 2024).

Neural Renderer. The neural renderer draws from the architecture of (Zhang et al., 2026), using Cosmos-Predict-0.6B as backbone. We train the model with an AdamW optimizer and learning rate of $2e^{-5}$, in 3-phases of 32K iteration each, as described in Section 4. In phase 1 of the training curriculum (Sec.4.2), we train the model on synthetic data of resolution 512×512. For phase 2 and 3 we resize input frames to 1024×576.

For the entire length of optimization, we keep the WAN 2.1 VAE frozen and train only the DiT throughout the entire training. Different to (Zhang et al., 2026), our architecture does not assume skip connections between the encoder and decoder due to domain differences in input and output. The single-step denoising setting follows (Zhang et al., 2026) and is prefixed to $t = 250$. In all phases, we train the model with L2 and LPIPS (Zhang et al., 2018) loss, weighted equally. After encoding each pair of inputs with the VAE, latent fusion is achieved by weighted average, where out settings are fixed to $w_{pbr} = 0.5$ and $w_{irr} = 0.5$. In our full pipeline, the temporal dimension compresses 5 frames to a latent of 2-dim.

For interactive preview, we feed single $\hat{I}_{pbr}$ and $\hat{I}_{irr}$ frames, encoded without temporal consistency. Our final results are conducted with temporal compression of 5 frames.

Hardware. Interactive applications which require a GUI were conducted on a machine equipped with an NVIDIA A6000 gpu. All comparisons reported in this paper, using the TRON and Gaussian baselines (Jiang et al., 2023; Liang et al., 2023b; Chen et al., 2025a) were conducted on a machine with an NVIDIA L40 gpu. For neural methods (Liang et al., 2025a; He et al., 2025), we used a single A100 gpu.

Data generation pipeline was run on a cluster of 22 x NVIDIA L40 gpus. The reconstruction and rendering of 955 scenes from DL3DV-10K (Ling et al., 2024) took 4 days end-to-end, including envmap optimization. The synthetic data generation, reconstruction and re-rendering lasted 72 hours. Each phase of the neural renderer was trained for 12 hours on a machine equipped with 8 x NVIDIA A100 gpus.

E Dataset Construction

E.1 Data Pipeline

TRON-DL3DV-33 Details. Our real-world evaluation set, TRON-DL3DV-33, consists of 33 scenes from the 2nd partition of DL3DV-10K (Ling et al., 2024). These scenes were never used during training of the neural renderer. Each scene is paired with 3 different rotated environment maps from a curated collection of envmaps of different contrast levels, rotated to match the scene. The entire dataset includes 99 video clips of varying illuminations. For Gaussian reconstruction methods, we augment the dataset with SFM initialization provided by (Chen et al., 2025b).

TRON-Synthetic Details As full-scene multi-illumination datasets with known environment maps are scarce, we render out a synthetic dataset, closely following prior work (Liang et al., 2025a). To construct TRON-Synth Benchmark, we collect 29 high-quality 3D models from PolyHaven and a diverse set of 33 medium and high-contrast HDR environment maps from multiple sources §E.2. TRON-Synth contains 15 single-object scenes and TRON-Synth 25 multi-object scenes, each rendered with a path tracer under 5 sampled illumination conditions. Objects are arranged on a textured PBR floor with collision checks to ensure physically plausible layouts. For each scene, we sample cameras using two protocols: (1) **Spiral cameras** - a trajectory of 290 viewpoints, covering pole to grazing views, which is used by all reconstruction-based baselines and TRON to reconstruct the original scene, and (2) **Oscillating video cameras** - smooth continuous camera trajectories of 93 frames, suited for both video model baselines and reconstruction-based baselines, thereof used by methods as test views. All images in the benchmark were rendered with resolution of 800 x 800.

E.2 Synthetic Benchmark Assets

E.2.1 High-Contrast Environment Maps

Sourced from Poly Haven:

1. `afrikaans_church_interior_4k`
2. `carpentry_shop_01_4k`
3. `construction_yard_4k`
4. `ferndale_studio_01_4k`
5. `ferndale_studio_06_4k`
6. `future_parking_4k`
7. `goegap_road_4k`
8. `monochrome_studio_01_4k`
9. `moon_lab_4k`
10. `qwantani_noon_puresky_4k`
11. `snowy_field_4k`
12. `story_studio_05_4k`
13. `studio_small_01_4k`

14. `studio_small_07_4k`
15. `teutonic_castle_moat_4k`
16. `thatch_chapel_4k`
17. `white_home_studio_4k`
18. `wooden_lounge_4k`
19. `yellow_field_4k`

E.2.2 Medium-Contrast Environment Maps

Sourced from HDRI-Hub:

1. `container_free_Env`
2. `Harbor_3_Free_Env`
3. `HDR_040_Field`
4. `HDR_041_Path`
5. `HDR_110_Tunnel_Env`
6. `HDR_111_Parking_Lot_2_Env`
7. `HDR_112_River_Road_2_Env`
8. `HDR_Free_City_Night_Lights_Ref`
9. `night_free_Env`
10. `Stonewall_Env`

Sourced from ambientCG:

11. `IndoorEnvironmentHDRI009_4K_HDR`
12. `IndoorEnvironmentHDRI020_4K_HDR`
13. `IndoorEnvironmentHDRI021_4K_HDR`
14. `NightEnvironmentHDRI010_4K_HDR`

F Evaluation Details

F.1 Inference Mode

To evaluate TRON on a new scene, we first obtain material priors per view (§3.2) and reconstruct the scene using our pipeline (§3.1). Given a novel HDR environment map, we use our Gaussian Ray Tracer (§3.3) to render $\hat{I}_{\text{pbr}}$ and $\hat{I}_{\text{irr}}$, which are then forwarded to the Neural Renderer (§4) to produce final RGB image. Given our extended Gaussian representation from Eq. (1), editing becomes easy through recombination of particles from other scenes with positions of their subsets shifted, or material properties modified.

During interactive applications, we apply the neural renderer in image preview mode, operating on single frames without temporal consistency. Mirroring the iterative workflows of Blender (Blender Online Community, 2024) and Cycles, this mode prioritizes immediate visual feedback over temporal consistency during scene drafting, while the complete framework is utilized for high fidelity finalization.

F.2 Baseline Details

Gaussian Methods. Evaluation of normals is conducted in camera space. For methods that render normals in world space (GI-GS, GS-IR), we normalize and rotate the normals of each pixel using the test view cameras. For GaussianShader, we found the normals consistency loss on synthetic-benchmark causing severe artifacts due to the depth derived reference normal being too noisy. For this dataset-benchmark only, we reduce the normal consistency loss weight to zero. We repeated the experiments of all baselines using their latest code released, and validated against results reported in the respective papers. Discrepancies in reported numbers are due to code changes since paper publications, and different evaluation protocols.

Neural Rendering Methods. For UniRelight and Diffusion Renderer, we project each envmap to camera space. As both methods do not require training data, for TRON-DL3DV-33 we feed the entire trajectory consisting of train and test views, and select the test frames as output.

F.3 Albedo Scaling

For basecolor (albedo) metrics, we apply per-channel scaling channel optimization restricted to foreground pixels, a practice introduced by NeRFactor (Zhang et al., 2021b) and adopted as standard by TensoIR (Jin et al., 2023) and subsequent Gaussian based inverse rendering works (Liang et al., 2023b) and (Chen et al., 2025a). Concretely, for each test view we solve a closed-form least-squares problem to find scale s that best aligns the albedo prediction of channel c of view i with the ground truth: $s^{\star}_{i,c} = \arg\min_{s \geq 0} \|s\, \hat{A}_{i,c} - A_{i,c}\|_2^2$. We repeat this convention for all baselines and datasets we evaluate on.

F.4 A/B Photorealism Additional Details

The A/B photorealism benchmark described in Tab. 3 was conducted using GPT 5.1. We collect 99 video clips by rendering TRON-DL3DV-33 with each method. In each round we sample one result from ours, and one from the baseline. The agent is presented with a set of instructions, instructed to choose which anonymous image fits the instructions better: A, B or tie. Images are permuted, and pairs are randomized per trial. We repeat this test with 3 different seeds and report standard deviation within the table.

Photorealism Evaluation Prompt

Role: You are evaluating two images, A and B, which are different rendering methods of the same 3D scene under the same lighting environment. Your task is to decide which looks more photorealistic.

Setup: Both images depict the same camera viewpoint and illumination (same HDR environment map). They differ only in the rendering method. Absolute image size is not a discriminator. Judge purely on rendering quality—not on content, framing, or composition.

Criteria (in priority order):

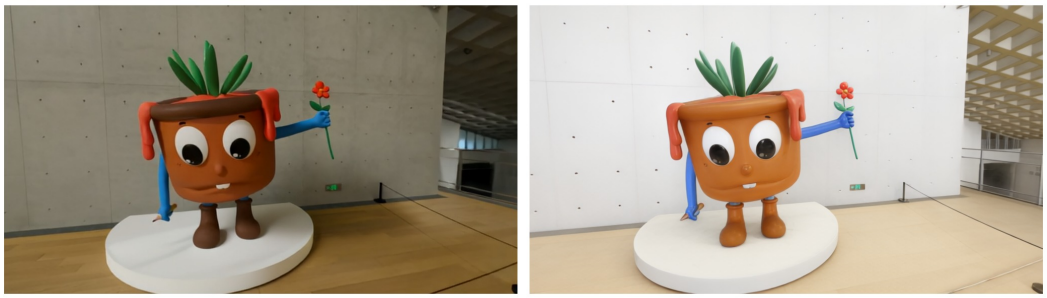
1. **Light transport:** Are specular highlights and reflections placed where geometry and lighting predict? Does indirect light look plausible?
2. **Shadow consistency:** Are shadow positions, softness, and falloff coherent with light sources? Are contact shadows present?
3. **Material plausibility:** Do surfaces read as the correct material? Avoid plastic-looking metals, chalky albedos, or oversaturated highlights.
4. **Geometric coherence:** Are silhouettes clean? No popping, ghosting, or isolated artifacts indicating instability.
5. **Color & exposure:** Are tints coherent? Does brightness fall off naturally? Is the white balance plausible?
6. **No neural artifacts:** Penalize blocky/checkerboard textures, hallucinated detail, smeared edges, or any "AI-look" that breaks physical plausibility.
7. **Daylight plausibility:**
   - Daytime scenes must **not** look uniformly dark. Globally dim renders where shadows should be present are penalized as failed rendering.
   - Conversely, washed-out daylight lacking strong shadows where expected is also penalized.

What to ignore: Camera-level effects (lens distortion, vignette) and aesthetic preference. The goal is realism, not beauty.

Anti-bias instruction: Position (A or B) must not influence your decision. Imagine labels swapped, would your answer change?

Output: **A single token.** Use “A” or “B”. Use “TIE” only when no meaningful difference exists.

UniRelight | **TRON**



Verdict: **TRON** judged more photorealistic than UniRelight*

Image B exhibits more physically plausible lighting and materials. The statue has appropriate glossy highlights and reflections consistent with the bright indoor environment, and the indirect light on the floor and pedestal is coherent. Shadows and contact areas are soft but present, matching the diffuse ambient lighting. Materials read convincingly as painted plastic with correct specular response and smoother shading. In Image A, the scene overall is underexposed relative to the apparent interior brightness, specular highlights on the character are muted and inconsistent, and the floor and pedestal lack realistic light interaction, giving a flatter, less physically based look. Therefore, B appears more photorealistic by the stated criteria.

TRON | **GT**



Verdict: **GT** judged more photorealistic than TRON*

Image B has more plausible light transport and shading: the indirect illumination and contrast between lit and shadowed areas look consistent with the fluorescent fixtures and window light, and contact shadows under chairs and objects are clearer and more grounded. Materials read more realistically — plastics and painted metal have appropriate specular response and roughness, while in A surfaces appear overly flat and washed out, with weaker highlights and almost no depth in the shadows. B also avoids the slightly smeared, over-smoothed look present in A, giving crisper geometry and more believable textures. Overall, B better matches real-world exposure, color balance, and shadow behavior for this indoor lighting setup.

Figure 15: **Agent Test Explainability.** Showing rationale for final verdict of agent score, as presented in A/B photorealism benchmark of Table 3. The proposed evaluation can be further validated with descriptions.

Per-frame FPS and first-frame latency were measured on a single NVIDIA A100 for UniRelight(He et al., 2025) and DiffusionRenderer(Liang et al., 2025a). TRON and Gaussian based methods (Chen et al., 2025a; Liang et al., 2023b; Jiang et al., 2023) were measured on NVIDIA L40. We also report FPS of 1.6 and latency of 625ms for our pipeline running on a desktop machine equipped with NVIDIA A6000.

F.5 VBench Metrics

We apply the VBench (Zhang et al., 2024) protocol on 99 differently illuminated clips from TRON-DL3DV-33. Content-only metrics operate purely on pixels and cross-frame features, evaluating the generated output without requiring a ground-truth reference. We measure the following dimensions of the benchmark, which do not require a textual prompt.

Motion Smoothness. Measures temporal coherence with AMT-S optical flow.

Subject Consistency. Evaluates whether the foreground subject remains coherent across the camera trajectory, using DINO across frames.

Background Consistency. Assesses whether background elements remain stable throughout the video, using CLIP across frames.

Aesthetic Quality. Estimates the overall perceived visual appeal of the frames using LAION over ViT-L/14.

Imaging Quality. Measures per-frame image quality independent of any reference images.