HyGI: Neural Global Illumination for Dynamic Hybrid Gaussian–Mesh Scenes

Anonymous authors
Paper under double-blind review

Abstract

Relighting dynamic scenes composed of 2D Gaussians and triangle meshes requires computing global illumination, i.e., light transport between these hybrid assets, to produce plausible shading, shadows, and indirect illumination effects.

Existing relightable Gaussian methods primarily address relighting individual Gaussian assets, rather than modeling dynamic light transport between Gaussians and triangle meshes. Although a few real-time global illumination techniques target such hybrid scenes, they often introduce severe visual artifacts due to the heuristic approximations required by vanilla real-time renderer. Instead, we present HyGI, a neural global illumination renderer for hybrid Gaussian–mesh scenes with dynamic objects, lights, and views. Given a scene's Gaussian and mesh assets with physically based materials, HyGI takes an inexpensive stochastic G-buffer and light-space shadow maps as scene and shadow cues, then aggregates this spatially coherent information for efficient relighting with a kernel-predicting layer guided by Gaussian footprint descriptors, and further end-to-end predicts direct shading, visibility, and indirect illumination with light encoders and separate decoders. Our method targets interactive light transport between Gaussian and mesh assets and successfully produces the high visual quality of physically-based rendering at interactive rates. Across twelve scenes, HyGI improves valid-surface PSNR by 1.8–9.0 dB over the real-time hybrid renderer 3DGS-GI and by 2.6–11.2 dB over relightable Gaussian methods on static scenes, at 29–59 ms per frame.

1 Introduction

Gaussian splatting (Kerbl et al., 2023; Huang et al., 2024) has made it practical to reconstruct real objects as high-quality, real-time renderable assets, and such assets are increasingly placed into conventional triangle-mesh environments (Guédon & Lepetit, 2024; Xiao et al., 2026). Relighting such hybrid scenes requires computing global illumination, i.e., light transport between the Gaussian and mesh assets: a Gaussian object should cast shadows onto a mesh floor and receive light bounced from a mesh wall, and because objects, lights and the camera move, these effects change every frame and must be computed at interactive rates.

Relightable Gaussian methods address parts of this problem. Inverse rendering recovers physically based materials for individual Gaussian assets (Gao et al., 2024; Liang et al., 2024; Gu et al., 2025; Xu et al., 2026), and end-to-end methods such as GS^3 (Bi et al., 2024) and RNG (Fan et al., 2025) relight a captured scene with high quality, but each model is fitted to a single static configuration, usually one object, under a single light type. Neither models light transport between Gaussians and meshes in a changing scene. Path tracing over Gaussians (Moenne-Loccoz et al., 2024; Zhu et al., 2026) evaluates this transport in full, but remains an offline process. Real-time methods either precompute radiance transfer, which assumes static geometry and distant light (Guo et al., 2024), or, in the few that target hybrid Gaussian–mesh scenes, such as 3DGS-GI (Hu et al., 2026), rely on heuristic approximations such as radiance caches and denoisers, which often bring undesirable blurry and ghosting artifacts.

We take a learning-based route and treat global illumination in hybrid scenes as a prediction problem. Instead of tracing rays through Gaussians at runtime, we rasterize an inexpensive stochastic G-buffer of the whole hybrid scene and light-space shadow maps as scene and shadow cues, and train a

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.

network to map these noisy inputs to path-traced light transport. The network is supervised by a hybrid path tracer, which we obtain by extending PTIR-GS (Zhu et al., 2026) to scenes that mix Gaussians and meshes.

We call the resulting pipeline HyGI. It takes a scene's Gaussian assets with physically based materials, together with its meshes. We mainly focus on 2D Gaussian (surfel) assets, which we reconstruct with geometry field splatting (Jiang et al., 2025a), giving the path tracer better-defined ray hits; since the network reads only rasterized attributes, it applies to volumetric Gaussians as well. Each frame, HyGI rasterizes the scene with a given number of stochastic samples per pixel, aggregates the noisy surface attributes and shadow cues with kernel-predicting networks (Bako et al., 2017) guided by Gaussian footprint descriptors, and predicts direct shading, visibility and indirect illumination with a light encoder and separate decoders. A model is trained per scene, and at test time objects, lights and the camera move freely within the trained range.

We evaluate HyGI on twelve scenes, from environment-lit Gaussian objects to dynamic rooms that mix Gaussian objects and meshes. It achieves the highest valid-surface PSNR in every scene, 1.8–9.0 dB above 3DGS-GI. On the five static scenes it also leads the relightable Gaussian methods RNG and GS^3, by 3.1–9.7 dB and 2.6–11.2 dB, although these reconstruct the scene from images rather than receiving its assets. HyGI runs at 29–59 ms per frame, slower than 3DGS-GI and GS^3 and comparable to RNG, and the supplementary video shows it on dynamic scenes.

In summary, our contributions are:

- A learned global-illumination pipeline for hybrid Gaussian–mesh scenes with moving objects, lights and camera, supervised by a path tracer that we extend to mixed Gaussian and mesh geometry.
- A stochastic Gaussian G-buffer with learned, kernel-predicting filtering guided by Gaussian footprint descriptors, which replaces sorted blending and extra samples with a small, fixed number of samples per pixel.
- An evaluation on hybrid and all-Gaussian scenes showing higher valid-surface accuracy than the existing real-time global-illumination techniques for Gaussian–mesh scenes in all twelve scenes at interactive frame rates, together with comparisons to relightable Gaussian methods on static scenes and ablations of each component.

2 Related Work

Gaussian representations and relighting. 3D Gaussian splatting (Kerbl et al., 2023) renders volumetric Gaussians in real time, and surfel variants (Huang et al., 2024; Jiang et al., 2025a) add explicit normals and well-defined surfaces, which we rely on (Section 3.2). Inverse rendering decomposes such scenes into geometry, materials and lighting, first with rasterized shading and baked indirect light (Gao et al., 2024; Liang et al., 2024; Jiang et al., 2024) and later with ray-traced interreflections (Chen et al., 2025; Sun et al., 2025; Gu et al., 2025; Jiang et al., 2025b), while diffusion priors predict normals and materials (Ye et al., 2024; Zeng et al., 2024; Xu et al., 2026). These methods produce the relightable assets our pipeline takes as input. End-to-end relightable Gaussians such as GS^3 (Bi et al., 2024) and RNG (Fan et al., 2025) instead learn relighting directly from images under varying light (Zeng et al., 2023); they reach high quality on the captured scene, but each model is fitted to one static configuration and light type, so objects cannot be moved and transport between separately reconstructed assets is out of scope.

Ray tracing and global illumination with Gaussians. Gaussians can be ray traced with hardware acceleration (Moenne-Loccoz et al., 2024; Wu et al., 2025), and PTIR-GS (Zhu et al., 2026) builds a Monte Carlo path tracer on this machinery, which we extend to mixed Gaussian and mesh scenes to render our references offline. Precomputed radiance transfer relights Gaussians in real time but assumes static geometry and distant light (Guo et al., 2024), and hybrid representations combine meshes and Gaussians for rendering and editing without relighting (Guédon & Lepetit, 2024; Xiao et al., 2026). Closest to our setting, 3DGS-GI (Hu et al., 2026) computes real-time global illumination for dynamic Gaussian and mesh scenes with stochastic ray tracing, radiance caches and denoising; HyGI addresses the same scenes with a network trained against a path-traced reference instead of a hand-designed estimator.

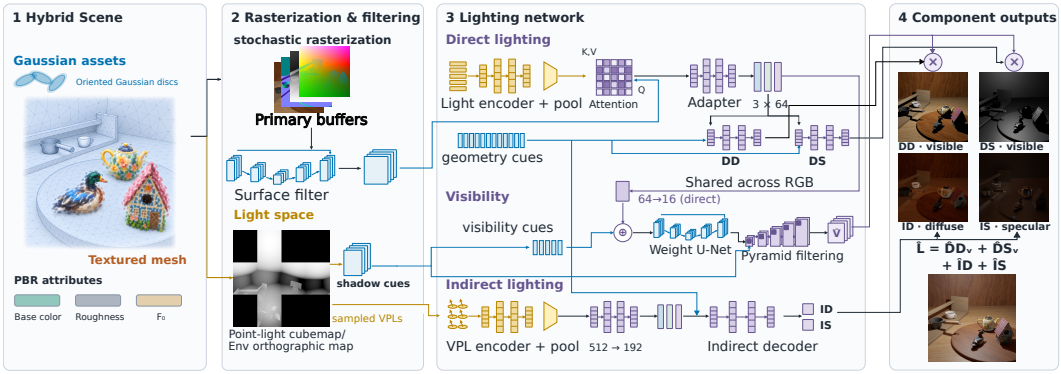

Figure 1: Overview of HyGI. Mesh and Gaussian assets with physically based materials are rasterized stochastically into a camera G-buffer and a light-space map (a cube map for a point light, an orthographic map for an environment map), and kernel-predicting networks filter the surface attributes and shadow cues. The lighting network predicts the unshadowed diffuse (DD) and specular (DS) direct shading from the encoded light and explicit geometry, the visibility $\hat{V}$ by learned filtering of the shadow-map occlusion, and the indirect components (ID, IS) from virtual point lights; the relit image is $\hat{L} = \hat{U}\hat{V} + \hat{I}$ (Equation (1)).

Efficient Gaussian rasterization. Pruning and compaction reduce the number of Gaussians (Fang & Wang, 2024; Fan et al., 2024; Lee et al., 2024; Mallick et al., 2024; Bagdasarian et al., 2025) and are orthogonal to our approach. Order-independent methods avoid per-pixel sorting (Hou et al., 2025), and stochastic methods, following stochastic transparency (Enderton et al., 2010), keep one randomly accepted opaque sample per pixel sample (Kheradmand et al., 2025; Rijsdijk et al., 2026), reducing noise with more samples or temporal accumulation. We apply this idea to the G-buffer of a hybrid scene with a small, fixed number of samples per pixel and filter the remaining noise with a learned network.

Learned rendering of light transport. Neural networks approximate light transport through precomputed or cached representations (Ren et al., 2013; Rainer et al., 2022; Hadadan et al., 2021; Müller et al., 2021) and by shading G-buffers, also in dynamic scenes (Nalbach et al., 2017; Diolatzis et al., 2022; Zheng et al., 2023; 2024; Su et al., 2024). LightFormer (Ren et al., 2024) gathers encoded lights per pixel by attention, and NeLiF (Sheng et al., 2025) decodes shading from G-buffers and shadow cues.

3 Method

3.1 Overview

A scene consists of Gaussian surfel assets with physically based materials, triangle meshes, one light and a camera, and objects, light and camera change from frame to frame. We predict the outgoing radiance at every visible surface point $\boldsymbol{x}$ as

$$\hat{L}(\boldsymbol{x}) = \hat{V}(\boldsymbol{x})\,\hat{U}(\boldsymbol{x}) + \hat{I}(\boldsymbol{x}), \tag{1}$$

where $\hat{U}$ is the unshadowed direct radiance with diffuse (DD) and specular (DS) parts, $\hat{V}$ the direct visibility, and $\hat{I}$ the indirect radiance with diffuse (ID) and specular (IS) parts. The network sees only inexpensive rasterized inputs (Section 3.4) and is supervised with the same components from a hybrid path tracer (Section 3.3); we train one model per scene (Figure 1).

3.2 Hybrid scene representation

HyGI takes Gaussian assets of either kind, since the network only reads the rasterized G-buffer (Section 3.4); the scenes of series (a) in Section 4 use volumetric Gaussians (Kerbl et al., 2023). We build our own assets from Gaussian surfels, flat oriented disks (Huang et al., 2024) that carry

an explicit normal and a well-defined ray intersection, which the path tracer needs. We reconstruct them with GFSGS (Jiang et al., 2025a), which splats a geometry field for opaque surfaces and recovers solids more precisely than 2DGS (Huang et al., 2024), supervised with the scanned mesh when one is available and with predicted normals (Ye et al., 2024) otherwise (Section 5). Its opacity (Equation (2)) is flat-topped: a saturated surfel stays at 0.99 out to about half a standard deviation from its center and falls below 0.01 at two, whereas the oG opacity of 2DGS decays along the Gaussian tail, so a ray either hits a surfel or passes it. Materials (base color, roughness, metallic) come from diffusion-based prediction (Xu et al., 2026) or a fit under the path tracer (Zhu et al., 2026); meshes keep their own textures and materials (Appendix A).

3.3 Hybrid path-traced reference

We extend the Monte Carlo path tracer PTIR-GS (Zhu et al., 2026) to GFSGS surfels and to scenes that mix surfels with textured meshes. A ray $\boldsymbol{o} + t\boldsymbol{\omega}$ meets surfel i (center $\boldsymbol{\mu}_i$, normal $\boldsymbol{n}_i$, tangents $\boldsymbol{t}_u, \boldsymbol{t}_v$, scales σ_u, σ_v, amplitude g_i) in its plane, where we evaluate the GFSGS opacity:

$$t_i = \frac{\boldsymbol{n}_i \cdot (\boldsymbol{\mu}_i - \boldsymbol{o})}{\boldsymbol{n}_i \cdot \boldsymbol{\omega}}, \quad \rho_i = \left(\frac{(\boldsymbol{x}_i - \boldsymbol{\mu}_i)\cdot \boldsymbol{t}_u}{\sigma_u}\right)^2 + \left(\frac{(\boldsymbol{x}_i - \boldsymbol{\mu}_i)\cdot \boldsymbol{t}_v}{\sigma_v}\right)^2, \quad \alpha_i = \mathrm{GF}\big(g_i\, e^{-\rho_i/2}\big), \tag{2}$$

with $\boldsymbol{x}_i = \boldsymbol{o} + t_i\boldsymbol{\omega}$ and $\mathrm{GF}(x) = 1 - \exp(-0.03279\, x^{3.4})$, capped at 0.99 (Jiang et al., 2025a). Hits are composited front to back in order of t_i. For hybrid scenes, the tracer composites only the surfels in front of the closest mesh hit t_m, giving Gaussian coverage $O_G = 1 - \prod_{t_i < t_m}(1 - \alpha_i)$, and continues the path from the Gaussians with probability O_G and from the mesh otherwise:

$$\mathbb{E}[L] = O_G\, L(\boldsymbol{x}_G) + (1 - O_G)\, L(\boldsymbol{x}_m), \tag{3}$$

so shadows and inter-reflections cross between surfels and meshes within one path. The reference stores the four components, the unshadowed direct light and region masks for Gaussian objects and mesh receivers; Appendix B gives the estimators and our changes to PTIR-GS.

3.4 Stochastic hybrid G-buffer

Instead of sorting and blending Gaussians, we follow stochastic transparency (Enderton et al., 2010; Kheradmand et al., 2025) and let each surfel occlude a sample fully or not at all. Each pixel takes N samples, as hardware multisamples when $N > 1$; for each sample ray we evaluate the same α_i as the path tracer (Equation (2)) and accept the surfel if a random uniform number ζ satisfies $\zeta < \alpha_i$. Volumetric Gaussians go through the same stochastic sampling. Meshes are rasterized with ordinary depth testing, an accepted surfel behind the mesh hit is discarded, and the nearest accepted surfel wins. Because acceptances are independent and the winner is the first accepted surfel along the ray,

$$P(\text{winner} = i) = \alpha_i \prod_{j:\, t_j < t_i} (1 - \alpha_j), \tag{4}$$

which is the weight of i in ray-ordered compositing; averaging the N samples gives the coverage and the hit-conditioned attributes (Appendix B).

Consistency with the reference. To faithfully reproduce a Gaussian path-tracing result, the rasterizer must converge to the surface the path tracer sees. We therefore order fragments by the exact ray–surfel hit depth rather than the surfel center, which keeps Equation (4) in ray order (Radl et al., 2024), and we drop the screen-space low-pass floor of native Gaussian rasterization.

Shadow cues and descriptors. For a point light, we rasterize a cube map with the same scheme and, at each receiver of the filtered G-buffer, compute an eight-channel shadow cue (C8) from the cube-map depth and coverage (Appendix B); an environment map uses one orthographic map along a representative direction instead. Because a stochastic sample hides the shape of the surfel that produced it, we also store Gaussian footprint descriptors: the surfel's screen-space conic and the Mahalanobis distance ρ of the sample, averaged over Gaussian hits, and the fraction f_G of samples that hit a Gaussian.

3.5 NETWORK

Input filtering. The network combines kernel-predicting denoisers (Bako et al., 2017), the light and shadow encoders of LightFormer (Ren et al., 2024), and the shading decoders and learned shadow filtering of NeLiF (Sheng et al., 2025); Appendix C gives layer sizes. Two kernel-predicting U-Nets predict a 5×5 kernel per pixel: one filters the G-buffer, guided by its coverage, the descriptors and f_G, and the other the shadow cues recomputed at the filtered receivers. Both have no attribute targets and learn only through the lighting losses (Equation (6)), so they produce the inputs the decoders find most useful rather than clean attributes.

Light encoding. An MLP embeds the light records (one for a point light, 2000 importance samples for an environment map, divided by their mean radiance s) and averages them into a light token; the attention that weights the tokens of several lights reduces to a projection for our single light. With per-pixel embeddings of the light direction and half vector, the token gives 64 features per color channel, and a convolutional encoder turns the filtered shadow cue into a shadow feature.

Direct shading. Diffuse and specular MLPs, shared across color channels, map these features and 12 geometric values in the light's frame to log-space shading $\hat{y}_{\mathrm{DD}}, \hat{y}_{\mathrm{DS}}$, which give with the filtered materials the unshadowed direct radiance

$$\hat{U} = s\left[(\hat{\boldsymbol{\rho}} + \epsilon) \odot (e^{\hat{y}_{\mathrm{DD}}} - 1) + \hat{\boldsymbol{f}}_0 \odot (e^{\hat{y}_{\mathrm{DS}}} - 1)\right], \tag{5}$$

with diffuse albedo $\hat{\boldsymbol{\rho}}$, specular reflectance $\hat{\boldsymbol{f}}_0$ and $\epsilon = 10^{-3}$. An optional zero-initialized convolutional head adds a spatial residual to $\log(1 + \hat{U}/s)$; it is trained only by the image-level terms of Equation (6).

Visibility. Rather than decoding visibility, a U-Net predicts per-pixel filter weights from the compressed direct features, the filtered shadow cues and the shadow feature, and a partitioning pyramid applies them to the hard shadow-map occlusion; $\hat{V}$ is one minus the result. An optional head like that for $\hat{U}$ corrects $\hat{V}$ in the environment-lit scenes (Section 4).

Indirect illumination. 500 virtual point lights per frame (cube-map texels for a point light, filtered surface pixels for an environment map) are embedded and averaged like the light records, and a decoder shared across color channels predicts $\hat{y}_{\mathrm{ID}}, \hat{y}_{\mathrm{IS}}$ from these features and the local geometry; $\hat{I}$ is composed as in Equation (5). Series (a), whose reference has direct light only, has no indirect branch.

3.6 TRAINING

Targets live in the same log space: $y_c = \log(1 + L_c/(s\, m_c))$ for $c \in \{\mathrm{DD}, \mathrm{DS}, \mathrm{ID}, \mathrm{IS}\}$, where $L_{\mathrm{DD}}, L_{\mathrm{DS}}$ are the *unshadowed* reference components and m_c is the reference diffuse albedo or $\boldsymbol{f}_0$ (plus ϵ). The loss is

$$\mathcal{L} = \sum_c R\big(|\hat{y}_c - y_c|\big) + R\big(0.3\,(y_{\mathrm{DD}} + y_{\mathrm{DS}})\,|\hat{V} - V|\big) + \lambda\,\big(1 - \mathrm{SSIM}_M\big(\ell(\hat{L}), \ell(L)\big)\big), \tag{6}$$

where $V = \mathrm{clip}(L_{\mathrm{vis}}/L_{\mathrm{unsh}}, 0, 1)$ is the reference visibility, weighted by the reference brightness, and R averages over valid surface pixels, separately within Gaussian objects and the rest and then across the two. The last term, an SSIM (Wang et al., 2004) averaged over valid surface pixels, compares the composed prediction $\hat{L} = \hat{U}\hat{V} + \hat{I}$ with the reference radiance L in the log domain $\ell(x) = \log(1 + x)/8$, with $\lambda = 0.2$ (0 for the environment-lit scenes of series (a) and the Cornell scenes). We train each model for 50k iterations with Adam (Kingma & Ba, 2015) (learning rate 3×10^{-5}, batch 4 at 512^2, mixed precision) and keep an exponential moving average of the weights (decay 0.999).

Table 1: Display PSNR (dB) on held-out frames over valid surface (V), Gaussian objects (O) and receivers (R); best per scene and region in bold, "–" where a method does not apply. Protocol in Section 4.2.

	Ours			3DGS-GI			RNG			GS^3		
Scene	V	O	R	V	O	R	V	O	R	V	O	R
(a) Environment light, direct-only reference												
snowy	**26.63**	**22.14**	**27.76**	19.42	17.22	19.80	–	–	–	–	–	–
envmap12	**28.70**	**25.92**	**29.22**	20.30	19.78	20.37	–	–	–	–	–	–
overcast	**28.95**	**26.79**	**29.32**	19.93	19.88	19.93	–	–	–	–	–	–
(b) Moving object and point light												
Cornell teapot	**36.68**	**29.37**	**37.44**	32.14	26.19	32.63	–	–	–	–	–	–
Cornell birdhouse	**36.52**	**30.13**	**37.03**	30.74	20.03	32.68	–	–	–	–	–	–
(c) Static geometry, moving point light												
Coffee sparse	**32.31**	**25.52**	**34.01**	25.91	21.62	26.59	25.18	23.59	25.35	21.27	18.36	21.64
Coffee dense	**28.76**	**26.16**	**30.66**	21.52	20.67	21.95	25.64	25.69	25.62	25.32	24.29	25.85
desk_chair	**28.95**	25.53	**30.02**	21.60	19.96	21.98	25.19	**25.96**	25.07	26.35	25.51	26.52
stone_plane	**27.75**	**23.19**	**30.58**	22.92	19.85	24.26	18.08	18.33	18.02	16.57	16.52	16.58
Coffee room	**29.81**	**24.96**	**30.85**	23.78	20.33	24.36	24.08	23.00	24.21	24.09	24.45	24.05
(d) Dynamic scenes												
gallery (env.)	**27.82**	**24.17**	**28.60**	26.05	21.93	27.02	–	–	–	–	–	–
indoor (point)	**28.68**	**26.31**	**29.28**	23.70	22.99	23.84	–	–	–	–	–	–

4 EXPERIMENTS

4.1 SETUP

Scenes. We use four series of scenes (Appendix E): *(a)* the environment-lit R3DG scene on which 3DGS-GI is evaluated, with a direct-only reference (Gao et al., 2024); *(b)* a Gaussian object moving in a mesh Cornell box under a moving point light; *(c)* static geometry under a moving point light and camera, the setting relightable Gaussians support, in all-Gaussian (Coffee sparse and dense) and hybrid scenes (desk_chair, stone_plane, Coffee room); and *(d)* dynamic scenes in which Gaussian objects, the light and the camera move in mesh rooms (gallery under an environment map, indoor under a point light). Each model is trained on one scene and evaluated on held-out frames.

Baselines and metrics. 3DGS-GI (Hu et al., 2026) computes real-time global illumination for given Gaussian and mesh assets and applies to all series. RNG (Fan et al., 2025) and GS^3 (Bi et al., 2024) reconstruct relightable Gaussians from multi-view images under varying light and apply to series (c), where the comparison is between complete systems, since our method and 3DGS-GI receive the assets. We report PSNR, SSIM and LPIPS (Zhang et al., 2018) on tone-mapped images, each frame exposed from its reference, over valid surface pixels, and separately over Gaussian objects and receivers (Appendix E).

4.2 COMPARISON

Table 1 compares all methods; for ours we report, per scene, the better of the raw and exponentially averaged weights. 3DGS-GI accumulates radiance over frames; we report it after 1 or 32 accumulated frames at each test state, whichever gives it higher valid PSNR (Table 12). Unmodified 3DGS-GI casts no shadows from environment light, so for the environment-lit scenes we add environment importance sampling to it and report that version.

HyGI attains the highest PSNR on valid surface and on receivers in every scene, and on Gaussian objects in all but desk_chair, where RNG leads by 0.4 dB (Table 1 and Figure 2); on valid surface it leads 3DGS-GI by 1.8–9.0 dB, RNG by 3.1–9.7 dB and GS^3 by 2.6–11.2 dB. It also has the best valid-surface SSIM and LPIPS in most scenes (Appendix F); the exceptions are the LPIPS of 3DGS-GI with environment importance sampling on envmap12 and overcast, and the SSIM of RNG on Coffee dense and GS^3 on desk_chair. This accuracy comes at a cost in speed (Table 2): at

Table 2: Time per frame (ms) at 512^2 on an RTX 3090 for series (c) and (d). Our method is timed for a camera move and a light move (for the gallery, a rotation of the environment map); the costs of 3DGS-GI, RNG and GS^3 do not depend on what moves.

	Ours				
Scene	Camera	Light	3DGS-GI	RNG	GS^3
Coffee sparse	38.7	57.0	6.8	36.1	12.0
Coffee dense	39.5	50.7	7.3	31.7	6.4
desk_chair	32.0	41.2	10.1	26.0	13.0
stone_plane	30.9	41.5	6.6	32.3	22.4
Coffee room	28.6	40.9	4.8	26.2	12.3
gallery (env.)	40.7	58.5	7.3	–	–
indoor (point)	35.3	59.1	13.2	–	–

29–59 ms per frame, HyGI runs at interactive rates but is 3–9× slower than 3DGS-GI and 1.4–8× slower than GS^3, and up to 1.6× slower than RNG, which it matches for camera moves.

RNG and GS^3 relight simple, object-centred scenes well, and on desk_chair RNG even exceeds our object PSNR. They degrade on scenes in which a few Gaussian objects sit sparsely on a Gaussian surface with many training lights between them, such as Coffee sparse and stone_plane, where GS^3 fails (Figure 2, third row).

4.3 Dynamic scenes

The supplementary videos show four sequences in which objects, the light or environment and the camera move, next to path-traced references; our frames go through standard temporal anti-aliasing (Appendix D). Because every frame is inferred from its own inputs, our results follow the reference under motion, whereas 3DGS-GI, although it accumulates at least as many frames per video frame as fit into one of ours, leaves a shadow trail behind a moving object and lags a rotating sun. Over the gallery and Coffee room sequences, our valid-surface PSNR is 27.7 and 33.3 dB, against 23.6 and 26.4 dB for 3DGS-GI.

5 Ablation Studies

Geometry of the assets. As an example, we reconstruct the teapot (120 views) with four geometry variants and fit materials to each with the path tracer, so that only the geometry differs: (A) 2DGS, (B) GFSGS, (C) GFSGS with normals predicted by StableNormal (Ye et al., 2024), and (D) GFSGS supervised with depth and normals from the scanned mesh. Under a point light, the cast shadow depends on geometry alone (Figure 3): it breaks up near the contact for 2DGS, becomes coherent with GFSGS, and is cleanest when GFSGS is supervised by the scanned mesh, while predicted normals soften the shadow of thin parts. We therefore supervise GFSGS with the scanned mesh when one is available and with StableNormal normals otherwise; predicted normals may deviate from the true ones, but they keep the ray-traced shadows consistent.

Input sampling and filtering. Table 3 varies the number of samples per pixel of the stochastic G-buffer and removes the learned filtering and the Gaussian descriptors, on the two hybrid scenes with constant light power. To measure the noise that the stochastic input leaves in the output, we render each test state with 32 sampling seeds and report the per-pixel standard deviation σ across these renders. Filtering matters most when samples are scarce: at one sample per pixel it raises valid PSNR by 1.60 and 1.88 dB and object PSNR by 3.21 and 2.45 dB on desk_chair and stone_plane, against 0.24 and 0.47 dB valid at eight samples, and it lowers the sampling noise on the Gaussian objects from $\sigma = 12.29$ to 6.23 and from 14.70 to 8.98. Gaussian descriptors add 0.50 and 0.04 dB at eight samples. With filtering and descriptors, four samples per pixel come within 0.2 dB of eight while saving 6–7 ms per light move, and one sample costs 0.6–0.8 dB and saves 8–10 ms.

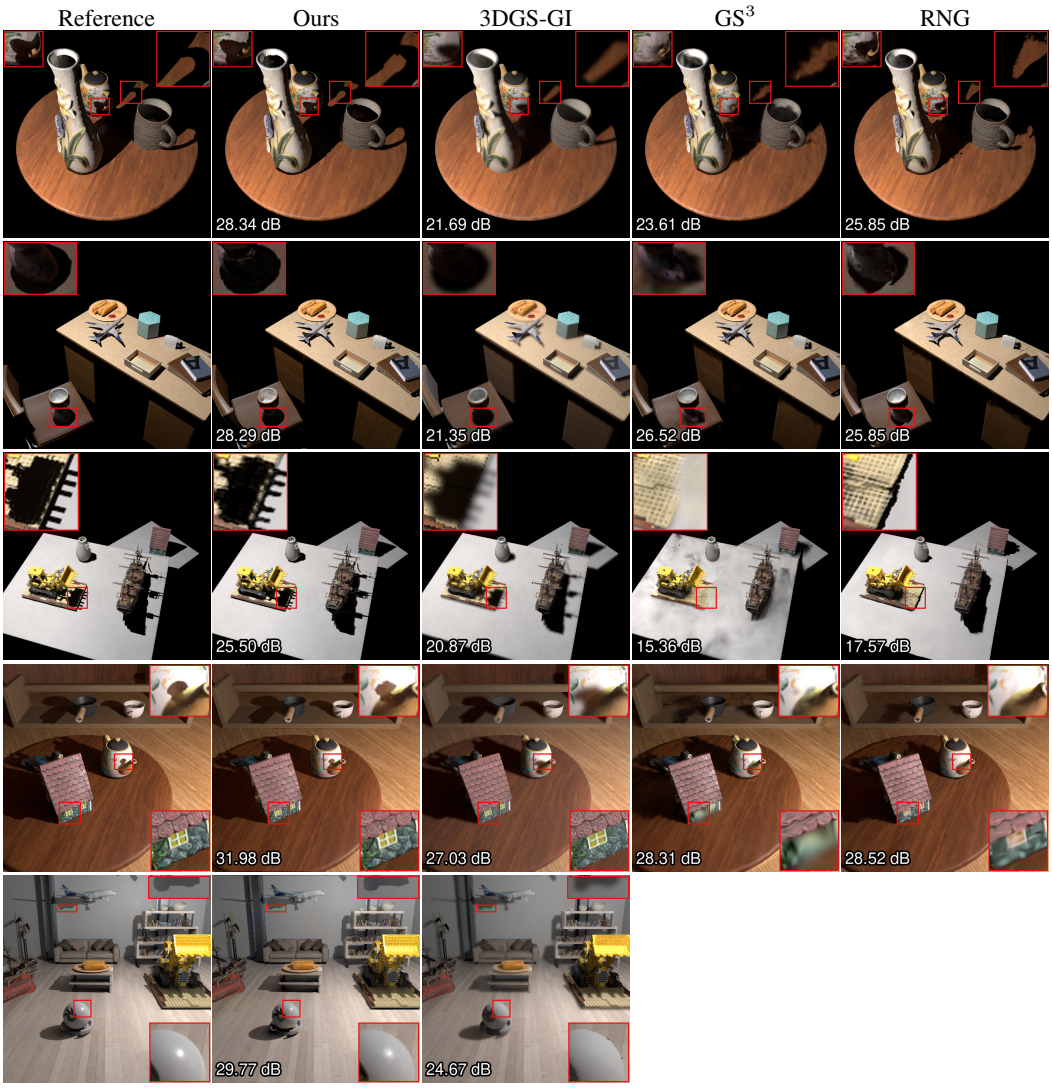

Figure 2: Held-out frames of Coffee dense, desk_chair, stone_plane, Coffee room and indoor (top to bottom), each displayed at its own exposure, with the valid-surface PSNR of the frame at that exposure; GS^3 and RNG do not apply to indoor.

6 LIMITATIONS

Light types. Our models support a single point light or an environment map. The formulation extends to several lights, since the light encoder fuses multiple lights, but the cost of the light-space rasterization and of the light fusion grows linearly with the number of lights; supporting many or mixed light sources at interactive rates is left to future work.

Light-space inputs. The light-space depth and coverage maps are rasterized stochastically and are therefore noisy. Because the shadow test against these maps is binary, a filtering network placed before it receives no useful gradient, and how to exploit Gaussian descriptors to denoise light-space depth remains open.

Per-scene training. Each model is trained for one scene, so a new scene requires new reference renderings and training rather than working out of the box. Because the network reads local G-buffer attributes and Gaussian descriptors rather than scene coordinates, we expect training across many assets to generalize at the object level.



(A) 2DGS (B) GFSGS (C) + pred. normals (D) + scanned mesh

Figure 3: Asset geometry variants under a point light (teapot, path traced, shared exposure); (C) and (D) add supervision to GFSGS. Insets magnify the handle, the base contact and the cast shadow next to it.

Table 3: Ablation of the stochastic input on desk_chair and stone_plane (constant light power): display PSNR (dB) over valid surface (V), Gaussian object (O) and receiver (R), standard deviation σ on the Gaussian objects across 32 sampling seeds (display values $\times 255$), and time per frame (ms) for a camera move / a light move at 512^2 on an RTX 3090; "full" adds Gaussian descriptors to the filtering; 50k iterations, EMA weights.

		desk_chair					stone_plane				
spp	Input	V	O	R	σ	ms	V	O	R	σ	ms
8	raw	27.17	23.53	28.35	5.74	28.4 / 37.6	26.06	21.89	28.39	8.18	27.4 / 38.0
8	+ filtering	27.41	23.99	28.47	4.24	31.9 / 41.3	26.53	22.45	28.76	6.63	30.9 / 41.5
8	full	27.90	24.41	29.00	4.08	32.0 / 41.2	26.57	22.63	28.64	6.16	30.9 / 41.5
4	raw	27.11	22.93	28.60	7.49	24.8 / 31.1	25.71	21.22	28.43	10.60	24.3 / 31.7
4	full	27.78	24.06	29.00	4.87	28.2 / 34.5	26.40	22.21	28.74	7.18	27.8 / 35.3
1	raw	25.66	20.26	28.23	12.29	23.2 / 28.2	23.94	18.77	27.78	14.70	23.4 / 30.1
1	full	27.26	23.47	28.52	6.23	26.5 / 31.6	25.82	21.23	28.68	8.98	26.8 / 33.6

Temporal stability in dynamic scenes. Our ablation shows that the stochastic inputs leave noise in the predicted image. Because the network takes no history, it is immune to ghosting; however, when an object, the light or the camera moves, the stochastic rasterization changes its realization, which causes temporal instability. Bringing temporal stability natively into the pipeline is left to future work.

Resolution. Cost grows with the pixel count: above 512^2, frames are not interactive on current hardware, and training is limited by GPU memory.

7 Conclusion

We presented HyGI, a neural renderer for global illumination in dynamic scenes that combine Gaussian assets with triangle meshes. It rasterizes an inexpensive stochastic G-buffer and light-space shadow maps, filters them with kernel-predicting networks guided by Gaussian footprint descriptors, and predicts direct shading, visibility and indirect light, supervised by a path tracer that we extend to Gaussian surfels and meshes. Across twelve scenes, from environment-lit Gaussian objects to rooms in which Gaussian objects, meshes, lights and the camera move, HyGI is more accurate on valid surface than the real-time hybrid renderer 3DGS-GI and than relightable Gaussian methods, at 29–59 ms per frame. It is, however, slower than 3DGS-GI, supports a single point light or environment map, and is trained per scene (Section 6); faster inference, many and mixed light sources, and one model across scenes are the next steps.

ETHICS STATEMENT

This work uses synthetic scenes and reconstructed assets and involves no human subjects or personal data. The captured objects come from the Digital Twin Catalog (Dong et al., 2025), the environment-lit scenes from the official release of R3DG (Gao et al., 2024), the mesh scenes and objects from BlenderKit (BlenderKit, 2026), and the environment maps from Poly Haven (Poly Haven, 2026), each used under its license. Relighting makes composited images more convincing and could be misused to fabricate imagery; our method relights only explicitly given, reconstructed assets and does not synthesize new content.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B specify the reference renderer, the stochastic G-buffer and the loss, Appendix C gives the layer sizes of the network, and Section 3.6 lists all training hyperparameters. Appendix E describes every scene with its splits, sample counts and denoising, together with the metrics, the per-frame exposure rule and the evaluation regions. Videos of the dynamic scenes are in the supplementary material. We will release the code and the reference data upon publication.

AI USE STATEMENT

The research problem, the overall pipeline and the choice of data are ours. We used large language model coding assistants to implement parts of the system under our direction, including the reference renderer extensions, the data processing (selecting, converting and packing captured and rendered data), and the training and evaluation code. They also designed some experiments and gave feedback on others, for example the measurement of sampling noise across seeds, carried out diagnostic experiments, and helped analyze their results. For this paper, we also used them to identify relevant literature and verify bibliographic entries, to suggest the paper structure, to draft and edit text, to format tables and references, and to create figures: the comparison and ablation figures were generated by AI-written scripts, and the pipeline figure started from an AI-generated template that we edited. No data was generated by an AI model. We tested all AI-written code and checked its outputs against independent measurements, took every reported number from the corresponding result files, and checked every citation against its publisher or arXiv record.

REFERENCES

Attila T. Áfra. Intel® Open Image Denoise, 2026. https://www.openimagedenoise.org.

Pontus Andersson, Jim Nilsson, Peter Shirley, and Tomas Akenine-Möller. Visualizing errors in rendered high dynamic range images. In *Eurographics 2021 – Short Papers*, 2021. doi: 10.2312/egs.20211015.

Milena T. Bagdasarian, Paul Knoll, Yi-Hsin Li, Florian Barthel, Anna Hilsmann, Peter Eisert, and Wieland Morgenstern. 3dgs.zip: A survey on 3d gaussian splatting compression methods. *Comput. Graph. Forum*, 44(2), 2025. doi: 10.1111/CGF.70078. URL https://doi.org/10.1111/cgf.70078.

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony DeRose, and Fabrice Rousselle. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Transactions on Graphics*, 36(4):1–14, 2017. doi: 10.1145/3072959.3073708.

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.

BlenderKit. BlenderKit: 3d model library, 2026. https://www.blenderkit.com.

Hongze Chen, Zehong Lin, and Jun Zhang. GI-GS: Global illumination decomposition on Gaussian splatting for inverse rendering. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=hJIEtJlvhL.

Stavros Diolatzis, Julien Philip, and George Drettakis. Active exploration for neural global illumination of variable scenes. *ACM Transactions on Graphics*, 41(5):1–18, 2022. doi: 10.1145/3522735.

Zhao Dong, Ka Chen, Zhaoyang Lv, Hong-Xing Yu, Yunzhi Zhang, Cheng Zhang, Yufeng Zhu, Stephen Tian, Zhengqin Li, Geordie Moffatt, Sean Christofferson, James Fort, Xiaqing Pan, Mingfei Yan, Jiajun Wu, Carl Yuheng Ren, and Richard Newcombe. Digital twin catalog: A large-scale photorealistic 3D object digital twin dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 753–763, 2025. doi: 10.1109/CVPR52734.2025.00079.

Eric Enderton, Erik Sintorn, Peter Shirley, and David P. Luebke. Stochastic transparency. In Daniel G. Aliaga, Manuel M. Oliveira, Amitabh Varshney, and Chris Wyman (eds.), *Proceedings of the 2010 Symposium on Interactive 3D Graphics, SI3D 2010, February 19-21, 2010, Washington, DC, USA*, pp. 157–164. ACM, 2010. doi: 10.1145/1730804.1730830. URL https://doi.org/10.1145/1730804.1730830.

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

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ FPS. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/fd881d3b625437354d4421818f81058f-Abstract-Conference.html.

Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXVII*, volume 15135 of *Lecture Notes in Computer Science*, pp. 165–181. Springer, 2024. doi: 10.1007/978-3-031-72980-5_10. URL https://doi.org/10.1007/978-3-031-72980-5_10.

Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3D Gaussians: Realistic point cloud relighting with BRDF decomposition and ray tracing. In *Computer Vision – ECCV 2024*, Lecture Notes in Computer Science, pp. 73–89. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72995-9_5.

Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. IRGS: Inter-reflective Gaussian splatting with 2D Gaussian ray tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10943–10952, 2025. doi: 10.1109/CVPR52734.2025.01022.

Antoine Guédon and Vincent Lepetit. Gaussian frosting: Editable complex radiance fields with real-time rendering. In *Computer Vision – ECCV 2024*, Lecture Notes in Computer Science, pp. 413–430. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73650-6_24.

Yijia Guo, Yuanxi Bai, Liwen Hu, Ziyi Guo, Mianzhi Liu, Yu Cai, Tiejun Huang, and Lei Ma. PRTGS: Precomputed radiance transfer of Gaussian splats for real-time high-quality relighting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pp. 5112–5120, 2024. doi: 10.1145/3664647.3680893.

Saeed Hadadan, Shuhong Chen, and Matthias Zwicker. Neural radiosity. *ACM Transactions on Graphics*, 40(6):1–11, 2021. doi: 10.1145/3478513.3480569.

Qiqi Hou, Randall Rauwendaal, Zifeng Li, Hoang Le, Farzad Farhadzadeh, Fatih Porikli, Alexei Bourd, and Amir Said. Sort-free gaussian splatting via weighted sum rendering. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=y8uPsxR8PN.

Chenxiao Hu, Meng Gai, Guoping Wang, and Sheng Li. Dynamic global illumination for interactive Gaussian splatting scenes in real time. *IEEE Transactions on Visualization and Computer Graphics*, 32(7):6919–6931, 2026. doi: 10.1109/TVCG.2026.3689199.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11. ACM, 2024. doi: 10.1145/3641519.3657428.

Kaiwen Jiang, Venkataram Sivaram, Cheng Peng, and Ravi Ramamoorthi. Geometry field splatting with Gaussian surfels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5752–5762, 2025a. doi: 10.1109/CVPR52734.2025.00540.

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*, 44(6):1–25, 2025b. doi: 10.1145/3763305.

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. GaussianShader: 3D Gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5322–5332, 2024. doi: 10.1109/CVPR52733.2024.00509.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433.

Shakiba Kheradmand, Delio Vicini, George Kopanas, Dmitry Lagun, Kwang Moo Yi, Mark J. Matthews, and Andrea Tagliasacchi. Stochasticsplats: Stochastic rasterization for sorting-free 3d gaussian splatting. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 26326–26335. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02443. URL https://doi.org/10.1109/ICCV51701.2025.02443.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 21719–21728. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02052. URL https://doi.org/10.1109/CVPR52733.2024.02052.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. GS-IR: 3D Gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21644–21653, 2024. doi: 10.1109/CVPR52733.2024.02045.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De la Torre. Taming 3dgs: High-quality radiance fields with limited resources. In Takeo Igarashi, Ariel Shamir, and Hao (Richard) Zhang (eds.), *SIGGRAPH Asia 2024 Conference Papers, SA 2024, Tokyo, Japan, December 3-6, 2024*, pp. 2:1–2:11. ACM, 2024. doi: 10.1145/3680528.3687694. URL https://doi.org/10.1145/3680528.3687694.

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*, 43(6):232:1–232:19, 2024. doi: 10.1145/3687934.

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics*, 40(4):1–16, 2021. doi: 10.1145/3450626.3459812.

Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. Deep shading: Convolutional neural networks for screen space shading. *Computer Graphics Forum*, 36(4):65–78, 2017. doi: 10.1111/cgf.13225.

Poly Haven. Poly Haven: Hdris, 2026. https://polyhaven.com.

Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering. *ACM Trans. Graph.*, 43(4):64:1–64:17, 2024. doi: 10.1145/3658187. URL https://doi.org/10.1145/3658187.

Gilles Rainer, Adrien Bousseau, Tobias Ritschel, and George Drettakis. Neural precomputed radiance transfer. *Computer Graphics Forum*, 41(2):365–378, 2022. doi: 10.1111/cgf.14480.

Haocheng Ren, Yuchi Huo, Yifan Peng, Hongtao Sheng, Weidong Xue, Hongxiang Huang, Jingzhen Lan, Rui Wang, and Hujun Bao. LightFormer: Light-oriented global neural rendering in dynamic scene. *ACM Transactions on Graphics*, 43(4):1–14, 2024. doi: 10.1145/3658229.

Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. Global illumination with radiance regression functions. *ACM Transactions on Graphics*, 32(4):1–12, 2013. doi: 10.1145/2461912.2462009.

Joris Rijsdijk, Christoph Peters, Michael Weinmann, and Ricardo Marroquim. Gaussian point splatting. *ACM Trans. Graph.*, 45(4):45:1–45:11, 2026. doi: 10.1145/3811272. URL https://doi.org/10.1145/3811272.

Hongtao Sheng, Yuchi Huo, Chuankun Zheng, Guangzhi Han, Yifan Peng, Shi Li, Bin Zang, Hao Zhu, Rui Tang, Yiming Wu, Rui Wang, and Hujun Bao. NeLiF: Neural lighting function generation for real-time indoor rendering. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, pp. 1–11, 2025. doi: 10.1145/3757377.3763958.

Rui Su, Honghao Dong, Jierui Ren, Haojie Jin, Yisong Chen, Guoping Wang, and Sheng Li. Dynamic neural radiosity with multi-grid decomposition. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, pp. 1–12, 2024. doi: 10.1145/3680528.3687685.

Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. SVG-IR: Spatially-varying Gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16143–16152, 2025. doi: 10.1109/CVPR52734.2025.01505.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3DGUT: Enabling distorted cameras and secondary rays in Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26036–26046, 2025. doi: 10.1109/CVPR52734.2025.02425.

Zeyu Xiao, Mingyang Sun, Yimin Cong, Lintao Wang, Dongliang Kou, Zhenyi Wu, Dingkang Yang, Peng Zhai, Zeyu Wang, and Lihua Zhang. UniMGS: Unifying mesh and 3D Gaussian splatting with single-pass rasterization and proxy-based deformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11023–11031, 2026. doi: 10.1609/aaai.v40i13.38080.

Jieting Xu, Rengan Xie, Zijian Huang, Zehui Jin, Rui Wang, and Yuchi Huo. GS-PI: An optimization-decoupled appearance decomposition approach for generating PBR Gaussian assets. *arXiv preprint arXiv:2609.19907*, 2026.

Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. StableNormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics*, 43(6):1–18, 2024. doi: 10.1145/3687971.

Chong Zeng, Guojun Chen, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. Relighting neural radiance fields with shadow and highlight hints. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11. ACM, 2023. doi: 10.1145/3588432.3591482.

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*, pp. 1–11, 2024. doi: 10.1145/3641519.3657445.

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

Chuankun Zheng, Yuchi Huo, Shaohua Mo, Zhihua Zhong, Zhizhen Wu, Wei Hua, Rui Wang, and Hujun Bao. NeLT: Object-oriented neural light transfer. *ACM Transactions on Graphics*, 42(5): 1–16, 2023. doi: 10.1145/3596491.

Chuankun Zheng, Yuchi Huo, Hongxiang Huang, Hongtao Sheng, Junrong Huang, Rui Tang, Hao Zhu, Rui Wang, and Hujun Bao. Neural global illumination via superposed deformable feature fields. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, pp. 1–11, 2024. doi: 10.1145/3680528.3687680.

Junke Zhu, Hao Zhang, Yutian Zhu, Ang Li, Chenxiao Hu, Meng Gai, Fei Zhu, Zhangjin Huang, and Sheng Li. PTIR-GS: Path-traced inverse rendering with global illumination in 3D Gaussian fields, 2026.

A Implementation Details

Reference renderer. We extend the OptiX implementation of PTIR-GS (Zhu et al., 2026). Each surfel is bounded by a two-triangle proxy and accepted only after its exact plane hit and opacity (Equation (2)) are checked; textured meshes are traced in the same scene.

Input generation. A headless OpenGL rasterizer produces all network inputs: a surfel pass tests each hardware multisample, a mesh pass fills the same samples, and a resolve pass computes Equation (8) and the descriptors. It renders six 512^2 cube faces for a point light or one orthographic map for an environment map, and generates both the training data and the inputs at inference.

Interactive runtime. The rasterizer shares its buffers with CUDA, and the network runs as TensorRT engines in half precision, with the indirect branch in single precision. A camera move rasterizes, filters and runs the network, and a light move also rebuilds the light-space maps and the virtual point lights (Table 2); the network takes about 19 ms and the filters about 2 ms at 512^2 on an RTX 3090.

Screen-space low-pass floor. Removing the screen-space low-pass floor (Section 3.4) improves display PSNR on Coffee dense by 0.62–0.71 dB on valid surface and by 0.75–0.85 dB on the Gaussian objects, for two model variants with a single change each. With the floor, silhouette coverage converges to about twice its reference value as the sample count grows, a bias that no amount of sampling removes.

On the silhouette pixels of two Cornell frames, the converged Gaussian coverage is 0.94–0.95 with the floor and 0.45–0.50 without it, against 0.46–0.50 in the reference.

B Method Details

Reference estimator. For a point light of intensity I at $\boldsymbol{\ell}$, direct light is estimated at the primary hit $\boldsymbol{x}$ by next-event estimation,

$$L_{\mathrm{dir}} = f(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_l)\, \cos\theta_l\, \frac{I}{\|\boldsymbol{\ell} - \boldsymbol{x}\|^2}\, V, \qquad V = \mathbb{1}[\text{no mesh occluder}] \prod_j (1 - \alpha_j), \tag{7}$$

where the product runs over the surfels along the shadow ray; this deterministic transmittance is the expectation of a stochastic visibility test. The unshadowed direct light uses the same sample with $V = 1$. Each term is split into diffuse and specular parts by the BRDF lobe fraction

$\varphi = \min(1, f_{\text{spec}} \cos\theta / f \cos\theta)$ at the primary hit, giving DD and DS, and, using the first continuation direction, ID and IS for all light after one or more bounces; the four components sum to the full radiance. Paths have at most three bounces; sample counts per scene are listed in Table 6. Where Table 6 lists it, the reference is denoised with Open Image Denoise (OIDN) (Áfra, 2026). Relative to the released PTIR-GS, we add the analytic surfel primitive with the geometry-field opacity, the hybrid mesh query of Equation (3), deterministic surfel shadow transmittance, and the four-component decomposition.

Per-pixel G-buffer estimate. With $h_s \in \{0, 1\}$ indicating that sample s hit a surfel or a mesh, g_s that it hit a surfel, and $\boldsymbol{a}_s$ its attributes, we store

$$c = \frac{1}{N} \sum_s h_s, \qquad f_G = \frac{1}{N} \sum_s g_s, \qquad \bar{\boldsymbol{a}} = \frac{\sum_s h_s \boldsymbol{a}_s}{\sum_s h_s}, \tag{8}$$

the coverage, the Gaussian fraction and the hit-conditioned attributes. In expectation, c is the opacity accumulated along the ray, and $\bar{\boldsymbol{a}}$ is a ratio estimator of the compositing-weighted attribute normalized by that opacity.

Shadow cue. The light's cube map (six 512^2 faces for a point light) stores the mean first-hit distance, the coverage c_L and the hit position per texel. At a receiver $\boldsymbol{p}$ with normal $\boldsymbol{n}$ from the filtered G-buffer, with $d = \|\boldsymbol{p} - \boldsymbol{\ell}\|$ and blocker distance d_b from the cube map,

$$\mathrm{C8} = \big[\, V,\ 10(d - d_b),\ 10(1 - d_b/d),\ \boldsymbol{n} \cdot \boldsymbol{\omega}_i,\ \boldsymbol{n} \cdot \boldsymbol{\omega}_o,\ \boldsymbol{p} \,\big], \tag{9}$$

where $V = 1 - b\, c_L$ combines a depth test $b \in \{0, 1\}$ against the cube map with the coverage.

C Network details

Table 4 lists the blocks of Section 3.5 with their size. We train with mixed precision; where half precision overflowed, the indirect branch (and, for the gallery, the shadow encoder) runs in single precision.

Table 4: Network blocks of the point-light models and their trainable parameters.

Block	Layers	Parameters
Surface filter	U-Net, 5 levels, width 24; 26 inputs $\to 5 \times 5$ kernel	182,641
Shadow-cue filter	same; 22 inputs $\to 5 \times 5$ kernel	181,778
Light encoder	MLP $9 \to 256^{\times 5}$, mean over records	265,728
Light attention	8 heads, 256-d; query MLP $13 \to 256$	266,752
Direction, half-vector encoders	$2\times$ MLP $3 \to 64^{\times 3} \to 32$, linear $32 \to 32$	23,424
Shadow encoder	convolutional U-Net on C8, linear $8 \to 8$	631,888
Direct adapter	linear $320 \to 3 \times 64$	61,632
DD, DS decoders	$2\times$ MLP $76 \to 64^{\times 3} \to 1$	26,626
$\hat{U}$ correction	3×3 convolutions $17 \to 16 \to 3$	2,899
Visibility weights	linear $64 \to 16$, linear $8 \to 5$, U-Net, 6 levels, 21 inputs	1,251,941
Partitioning pyramid	5 partitions, 5×5 kernels, 4×4 upsampling	–
Indirect encoder	MLP $9 \to 64^{\times 3} \to 128 \to 512$, mean, linear $512 \to 3 \times 64$	181,824
Indirect decoder	MLP $74 \to 64^{\times 3} \to 2$	13,250
Total		3,090,383

D Supplementary videos

The videos are rendered offline and played at 30 frames per second; they compare temporal behaviour, not speed (Table 2 gives the frame times). 3DGS-GI accumulates radiance over frames, so it renders K frames per video frame, at least the number that fits into one of our frames, with

each state held fixed, 32 warm-up frames and its temporal history never reset. Our method, RNG and GS^3 render every frame independently. Our frames go through standard temporal anti-aliasing (TAA) with motion vectors from the known camera and object transforms. The gallery video uses a 1024^2 shadow map instead of the 512^2 map the model was trained with; without retraining, this raises the valid-surface PSNR of the sequence from 28.16 to 28.67 dB and lowers the frame-to-frame jitter by 20%, for about 1 ms more per light move. The references are rendered with the path tracer of Section 3.3. Table 5 lists the sequences.

Table 5: Supplementary videos: scene and motion, 3DGS-GI frames per video frame (K), and valid-surface display PSNR (dB) against the reference over the sequence (valid / object / receiver where available).

Scene and motion	K	PSNR
gallery: drum kit and environment rotate once over 240 frames	8	ours 27.70 / 24.37 / 28.51; 3DGS-GI 23.60 / 21.76 / 23.96
Coffee room: camera and point light orbit the plate	6	ours 33.34 / 26.37 / 35.62; 3DGS-GI 26.39 / 22.02 / 27.31; RNG 30.37 / 26.13 / 31.26; GS^3 31.67 / 28.10 / 32.32
indoor: airplane sweeps back and forth	4	swept region: ours 33.66; 3DGS-GI 25.59, with a shadow trail that decays over about 20 frames
indoor: airplane, light and camera move	4	ours 32.20 / 28.82 / 33.03; 3DGS-GI 25.42 / 25.10 / 25.46

E DATASETS AND EVALUATION

Table 6: Scenes.

Scene	Geometry	Light	What moves	Train / test	Reference
snowy, envmap12, overcast	3D Gaussians (R3DG)	environment	camera, environment	500 / 50	R3DG renderer, 128 env. samples, direct only
Cornell teapot, birdhouse	surfels + mesh box	point	object, light	500 / 50	512 spp, no denoiser
Coffee sparse, dense	surfels	point	camera, light	900 / 100	512 spp, OIDN on indirect
desk_chair, stone_plane	surfels + mesh	point	camera, light	800 / 80	256 spp, OIDN on indirect
Coffee room	surfels + mesh	point	camera, light	1500 / 150	384 spp, OIDN on indirect
gallery	surfels + mesh	environment	objects, environment, camera	1500 / 150	384 spp, OIDN on all lobes
indoor	surfels + mesh	point (varying power)	object, light, camera	1500 / 150	384 spp, OIDN on indirect

Scenes. Series (a) has five objects represented as 3D Gaussians on a Gaussian ground plane, lit by three environment maps (snowy, envmap12, overcast), with the camera and the environment rotating between frames; as in the original setting of 3DGS-GI, the reference is rendered by the R3DG renderer (Gao et al., 2024) and contains direct light only. In series (b), a Gaussian teapot or birdhouse moves inside a mesh Cornell box with an open top under a point light that also moves, seen from a fixed camera. Series (c) contains two all-Gaussian Coffee scenes built from different assets, sparse and dense; desk_chair and stone_plane, which combine meshes and Gaussians; and a closed Coffee room with mesh furniture and Gaussian objects. Series (d) contains a gallery with an open ceiling under an environment map and an indoor room under a point light, in which Gaussian objects, the light and the camera move. Held-out frames have new camera, light and object configurations; since they informed model selection, they are validation rather than untouched test frames.

Baselines. We run 3DGS-GI as a Vulkan port of the original code and give it the same geometry and materials as our method. RNG and GS^3 reconstruct the scene from the training images and render new light positions; they are trained for 100k iterations and our models for 50k.

Metrics. Each frame uses its own exposure, 0.9 divided by the 99th percentile of that frame's reference over valid surface pixels, shared by all methods, followed by clamping and the sRGB transfer function; with a constant light power, a single exposure per scene would leave the frames with a distant light nearly black. PSNR pools the masked squared error over all frames; we never average across scenes.

F Additional Results

Table 7: Display SSIM (higher is better) for the scenes of Table 1; best per scene and region in bold.

	Ours			3DGS-GI			RNG			GS^3		
Scene	V	O	R	V	O	R	V	O	R	V	O	R
(a) Environment light, direct-only reference												
snowy	**0.923**	**0.753**	**0.944**	0.831	0.631	0.856	–	–	–	–	–	–
envmap12	**0.953**	**0.851**	**0.966**	0.920	0.733	0.944	–	–	–	–	–	–
overcast	**0.957**	**0.872**	**0.968**	0.922	0.743	0.945	–	–	–	–	–	–
(b) Moving object and point light												
Cornell teapot	**0.976**	**0.904**	**0.978**	0.970	0.900	0.972	–	–	–	–	–	–
Cornell birdhouse	**0.974**	**0.940**	**0.975**	0.963	0.645	0.974	–	–	–	–	–	–
(c) Static geometry, moving point light												
Coffee sparse	**0.920**	**0.840**	**0.927**	0.907	0.737	0.921	0.906	0.828	0.912	0.812	0.689	0.823
Coffee dense	0.878	0.823	0.903	0.830	0.764	0.859	**0.900**	**0.891**	**0.904**	0.879	0.835	0.897
desk_chair	0.910	0.871	**0.917**	0.827	0.702	0.850	0.897	0.897	0.897	**0.913**	**0.904**	0.914
stone_plane	**0.911**	**0.805**	**0.938**	0.858	0.701	0.899	0.782	0.673	0.810	0.714	0.638	0.733
Coffee room	**0.930**	**0.851**	**0.938**	0.890	0.765	0.903	0.877	0.822	0.883	0.874	0.839	0.878
(d) Dynamic scenes												
gallery (env.)	**0.916**	**0.809**	**0.929**	0.903	0.747	0.923	–	–	–	–	–	–
indoor (point)	**0.903**	**0.860**	**0.911**	0.855	0.761	0.872	–	–	–	–	–	–

Table 8: LPIPS (AlexNet, lower is better) for the scenes of Table 1; best per scene and region in bold.

	Ours			3DGS-GI			RNG			GS^3		
Scene	V	O	R	V	O	R	V	O	R	V	O	R
(a) Environment light, direct-only reference												
snowy	**0.184**	**0.109**	**0.194**	0.262	0.175	0.273	–	–	–	–	–	–
envmap12	0.161	**0.054**	0.174	**0.150**	0.098	**0.156**	–	–	–	–	–	–
overcast	0.159	**0.050**	0.172	**0.139**	0.093	**0.145**	–	–	–	–	–	–
(b) Moving object and point light												
Cornell teapot	**0.040**	**0.052**	**0.040**	0.075	0.071	0.075	–	–	–	–	–	–
Cornell birdhouse	**0.046**	**0.021**	**0.047**	0.073	0.125	0.071	–	–	–	–	–	–
(c) Static geometry, moving point light												
Coffee sparse	**0.091**	**0.064**	**0.093**	0.136	0.188	0.132	0.113	0.087	0.115	0.232	0.229	0.232
Coffee dense	**0.091**	0.083	**0.094**	0.203	0.228	0.192	0.105	**0.072**	0.119	0.127	0.105	0.137
desk_chair	**0.063**	**0.057**	**0.064**	0.195	0.242	0.187	0.095	0.072	0.099	0.091	0.073	0.094
stone_plane	**0.057**	**0.085**	**0.050**	0.192	0.210	0.188	0.218	0.214	0.219	0.299	0.262	0.308
Coffee room	**0.060**	**0.079**	**0.058**	0.124	0.171	0.119	0.144	0.121	0.147	0.160	0.110	0.165
(d) Dynamic scenes												
gallery (env.)	**0.104**	**0.109**	**0.103**	0.182	0.195	0.180	–	–	–	–	–	–
indoor (point)	**0.078**	**0.066**	**0.080**	0.151	0.156	0.150	–	–	–	–	–	–

Table 9: Log-domain PSNR (higher is better) on linear HDR radiance, $-10\log_{10}$ of the masked mean squared error between $\log(1+L)$ of prediction and reference with peak 1, for the scenes of Table 1; best per scene and region in bold.

	Ours			3DGS-GI			RNG			GS^3		
Scene	V	O	R	V	O	R	V	O	R	V	O	R
(a) Environment light, direct-only reference												
snowy	**37.48**	**33.29**	**38.48**	29.88	27.16	30.39	–	–	–	–	–	–
envmap12	**34.61**	**34.81**	**34.58**	27.94	28.95	27.83	–	–	–	–	–	–
overcast	**29.95**	**29.94**	**29.96**	22.51	23.35	22.41	–	–	–	–	–	–
(b) Moving object and point light												
Cornell teapot	**42.96**	**40.30**	**43.10**	40.29	36.43	40.53	–	–	–	–	–	–
Cornell birdhouse	**42.59**	**41.74**	**42.63**	39.77	32.67	40.42	–	–	–	–	–	–
(c) Static geometry, moving point light												
Coffee sparse	**29.76**	**21.96**	**32.23**	22.72	16.89	23.94	23.90	18.44	24.96	17.93	14.13	18.49
Coffee dense	**29.02**	**27.01**	**30.30**	19.09	19.97	18.76	25.68	24.74	26.15	25.71	24.04	26.69
desk_chair	**28.75**	23.80	**30.85**	20.46	17.54	21.29	23.28	23.11	23.31	23.71	**23.86**	23.69
stone_plane	**25.77**	**20.13**	**30.78**	21.21	16.96	23.62	16.21	15.88	16.30	13.81	14.30	13.70
Coffee room	**32.49**	24.61	**35.78**	26.83	20.84	28.42	27.64	23.09	28.56	27.31	**25.23**	27.59
(d) Dynamic scenes												
gallery (env.)	**30.03**	**26.12**	**30.92**	27.36	23.30	28.30	–	–	–	–	–	–
indoor (point)	**31.32**	**28.56**	**32.05**	24.66	24.74	24.64	–	–	–	–	–	–

Table 10: HDR-FLIP (Andersson et al., 2021) (lower is better) on linear HDR radiance, mean over each region, for the scenes of Table 1; best per scene and region in bold.

	Ours			3DGS-GI			RNG			GS^3		
Scene	V	O	R	V	O	R	V	O	R	V	O	R
(a) Environment light, direct-only reference												
snowy	**0.088**	**0.280**	**0.064**	0.164	0.453	0.127	–	–	–	–	–	–
envmap12	**0.049**	**0.197**	**0.030**	0.103	0.399	0.066	–	–	–	–	–	–
overcast	**0.039**	**0.163**	**0.024**	0.091	0.373	0.055	–	–	–	–	–	–
(b) Moving object and point light												
Cornell teapot	**0.076**	**0.184**	**0.072**	0.145	0.268	0.140	–	–	–	–	–	–
Cornell birdhouse	**0.078**	**0.163**	**0.075**	0.150	0.482	0.139	–	–	–	–	–	–
(c) Static geometry, moving point light												
Coffee sparse	**0.253**	0.405	**0.240**	0.374	0.439	0.369	0.290	**0.393**	0.281	0.362	0.508	0.350
Coffee dense	0.383	0.465	0.347	0.382	0.446	0.353	0.359	**0.374**	0.352	**0.352**	0.403	**0.330**
desk_chair	0.494	0.293	0.531	0.569	0.381	0.602	0.466	0.248	0.506	**0.432**	**0.242**	**0.467**
stone_plane	**0.349**	**0.418**	**0.331**	0.372	0.478	0.345	0.459	0.499	0.449	0.487	0.508	0.481
Coffee room	**0.194**	0.291	**0.184**	0.400	0.394	0.400	0.244	0.301	0.238	0.258	**0.285**	0.255
(d) Dynamic scenes												
gallery (env.)	**0.160**	**0.273**	**0.146**	0.177	0.312	0.160	–	–	–	–	–	–
indoor (point)	**0.153**	**0.218**	**0.142**	0.249	0.316	0.237	–	–	–	–	–	–

Table 11: Time per frame (ms) at 512^2 on an RTX 3090 for a camera move / a light move; for the gallery, our light move is a rotation of the environment map. RNG and GS^3 recompute their shadows in every frame, so both moves cost the same; "–" where a method does not apply.

	Ours		3DGS-GI		RNG		GS^3	
Scene	Cam.	Light	Cam.	Light	Cam.	Light	Cam.	Light
Cornell teapot	25.9	31.1	3.5	3.5	–	–	–	–
Cornell birdhouse	25.9	30.8	2.9	2.9	–	–	–	–
Coffee sparse	38.7	57.0	6.8	6.8	35.9	36.1	11.9	12.0
Coffee dense	39.5	50.7	7.3	7.3	31.7	31.7	6.4	6.4
desk_chair	32.0	41.2	10.1	10.3	26.0	26.0	13.1	13.0
stone_plane	30.9	41.5	6.6	6.7	32.2	32.3	22.1	22.4
Coffee room	28.6	40.9	4.8	4.8	26.2	26.2	12.3	12.3
indoor	35.3	59.1	13.2	13.3	–	–	–	–
gallery	40.7	58.5	6.9	–	–	–	–	–

Table 12: 3DGS-GI after 1 and 32 accumulated frames, display PSNR V / O / R; Table 1 shows the better of the two in valid PSNR.

	1 frame			32 frames		
Scene	V	O	R	V	O	R
snowy	16.91	16.55	16.96	19.42	17.22	19.80
envmap12	17.91	19.49	17.75	20.30	19.78	20.37
overcast	17.72	19.70	17.52	19.93	19.88	19.93
Cornell teapot	28.72	23.94	29.06	32.14	26.19	32.63
Cornell birdhouse	27.74	18.50	28.98	30.74	20.03	32.68
Coffee sparse	25.66	20.52	26.60	25.91	21.62	26.59
Coffee dense	21.23	20.05	21.85	21.52	20.67	21.95
desk_chair	21.30	19.20	21.81	21.60	19.96	21.98
stone_plane	22.56	19.08	24.21	22.92	19.85	24.26
Coffee room	23.71	20.15	24.32	23.78	20.33	24.36
gallery (env.)	23.36	20.56	23.88	26.05	21.93	27.02
indoor (point)	23.01	22.23	23.17	23.70	22.99	23.84