VolS-GS: Relightable Gaussian Splatting with Volumetric Subsurface Scattering

Anonymous authors
Paper under double-blind review

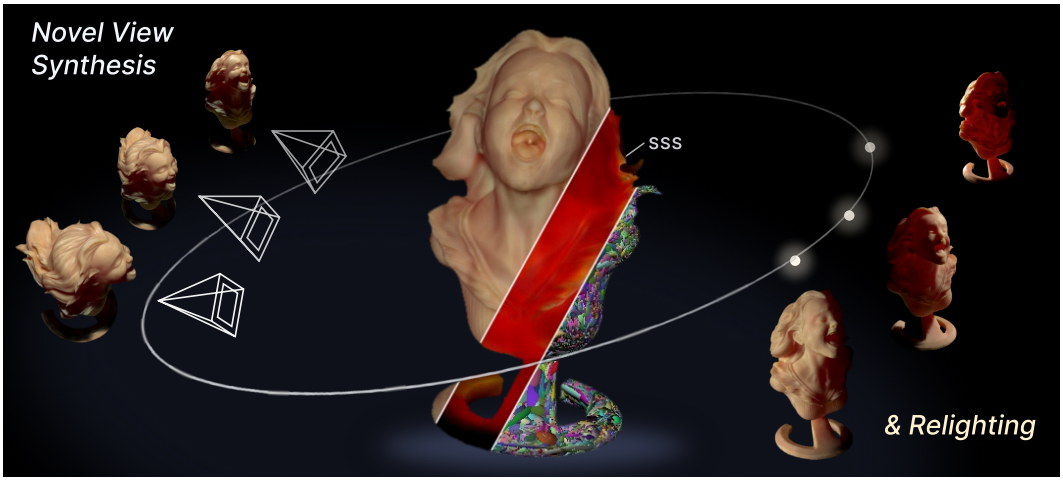


Figure 1: **VolS-GS.** proposes a high-quality relighting pipeline with three different modalities. **Center:** a rendered image can be decomposed into normal, diffuse, specular, subsurface scattering(SSS) and shadow. **Left:** VolS-GS faithfully synthesizes novel view renders. **Right:** relighting.

Abstract

We present VolS-GS, a relightable Gaussian splatting framework that reconstructs objects from one-light-at-a-time (OLAT) captures and renders them under novel lighting and viewpoints. Relightable Gaussian Splatting methods typically model appearance independently at each primitive, which makes non-local effects difficult to represent. This limitation is particularly apparent for subsurface scattering, where light entering the object at one location can emerge at another. Rather than modeling this effect solely with a neural network or a local kernel at each primitive, we use the spatial support of the Gaussian scene as the domain of a differentiable finite-volume transport solver, so that light can propagate through the object's interior. A small network predicts scattering and absorption coefficients for each Gaussian, and the solve redistributes incident light through the resulting field. The coefficients are fit to images rather than measured, so the solve supplies a transport-shaped path for aggregating per- primitive appearance, not a measurement of the material. To keep the learned shadow and specular terms from taking over the other components, our shadow term is predicted from visibility together with the transmittances and the scattering the solve produces, and a regularizer suppresses specular highlights in regions the shadow term predicts to be unlit. Experiments on three OLAT benchmarks show that VolS-GS consistently improves relighting quality on held-out lights and views.

1 Introduction

Relighting a captured object requires predicting its appearance under lighting conditions that were not observed during capture. One-light-at-a-time (OLAT) capture provides a controlled setting for

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.

this task by recording each image under a known point light, and the model must render the object from new viewpoints and under new light positions. Neural radiance fields made relighting practical by learning appearance as a function of position and illumination (Zeng et al., 2023; Jin et al., 2023), and 3D Gaussian splatting (Kerbl et al., 2023) replaced the implicit field with an explicit set of spatial primitives that renders far faster. Existing Gaussian approaches (Bi et al., 2024; Dihlmann et al., 2024; Zheng et al., 2026) give every primitive its own appearance model, evaluated from that primitive's own attributes.

Non-local transport is not beyond the representation. Radiosity-GS (Jiang et al., 2025) transports light between primitives with a differentiable radiosity solve, and Ref-Gaussian (Yao et al., 2025) traces inter-reflections among them. Both route light *around* the object, through the space outside it. Subsurface scattering routes light *through* it. The light enters at one surface point, travels inside the medium, and leaves at another. No Gaussian method solves for that interior path. SSS-GS (Dihlmann et al., 2024) conditions a scattering term on the light direction and SSD-GS (Zheng et al., 2026) evaluates a closed-form diffusion kernel at each primitive, so in both the subsurface response remains a function of one primitive's own attributes. A BSSRDF couples light entering the object to radiance emerging elsewhere, which introduces a surface integral and makes direct evaluation expensive. Classical diffusion models reduce this coupling to a local kernel under assumptions such as a homogeneous medium and a simple boundary (Jensen et al., 2001; Donner & Jensen, 2005; Habel et al., 2013). A per-primitive evaluation of such a kernel therefore has no interior path (Figure 2).

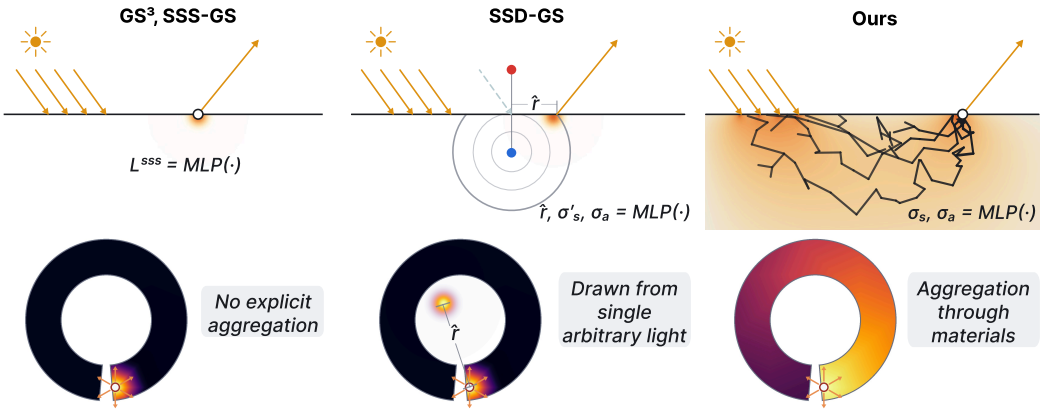


Figure 2: **Per-primitive kernels versus transport through the medium.** Top: where each method computes its subsurface response. Bottom: a C-shaped medium lit at the opening. The dipole depends on Euclidean distance, so its support crosses the gap, while our solve reaches the far tip only along the medium.

The Gaussian representation already provides the spatial support on which subsurface scattering can be computed. We turn that into an occupancy field on a coarse grid and solve on the occupied cells. Each Gaussian then reads the solved field back at its own position. The grid carries no parameters of its own, so the domain is the Gaussians themselves rather than an extracted mesh. VolS-GS learns scattering and absorption properties over this domain and uses them in a finite-volume transport solve. The solve produces the multiply scattered field, while a separate single-scattering term adds the once-scattered light integrated along the view chord, which the angle-averaged fluence cannot represent. An adjoint formulation makes the solver differentiable without unrolling its iterations, so the medium is learned end to end from images alone. Shadow modeling shares transmittance cues with the transport solve, and a regularizer discourages the specular branch from absorbing transported light.

In summary, we present VolS-GS, a relightable Gaussian splatting method, with the following contributions. (1) A differentiable volumetric subsurface scattering defined on the Gaussian scene's occupancy, allowing light to propagate across primitives. (2) A shadow term that reads the same transport field and a regularizer on the specular branch, which together keep the learned shadow and the learned specular from trading against each other. A full discussion of related work is deferred to Appendix A.

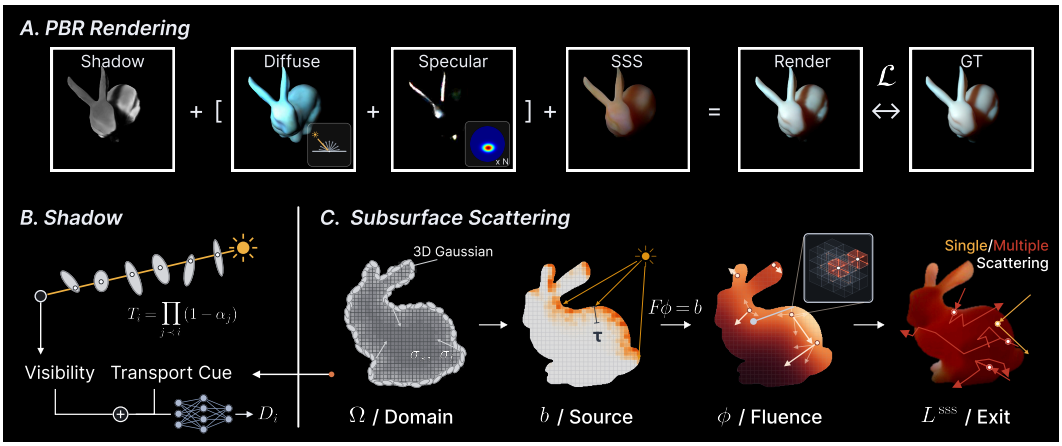


Figure 3: **Overview of VolS-GS. (A)** A render is the sum of a shadowed surface response and a subsurface term, compared against the captured frame. **(B)** The shadow is predicted from two cues gathered along the ray to the light, the fraction the Gaussians block and the fraction the medium transmits. **(C)** The subsurface term is propagated rather than predicted per primitive. The Gaussians define a domain Ω, the light entering it forms a source b, one linear solve $F\phi = b$ spreads that source through the medium, and each Gaussian reads the result back as outgoing radiance, so light can enter in one place and leave in another.

2 Method

We use the Gaussian cloud itself as the spatial support of a transport computation, solving for one unknown field, the fluence, on the cells the Gaussians occupy. Its source is the light that has scattered once inside the medium, and its solution is read back at each primitive as outgoing radiance (Figure 3).

2.1 Preliminaries

3D Gaussian splatting (Kerbl et al., 2023) represents a scene as a set of anisotropic Gaussians $\{g_i\}_{i=1}^{N}$, each with position $\mathbf{x}_i$, rotation and scale $(\mathbf{q}_i, \mathbf{s}_i)$ defining its covariance, opacity o_i, and a color c_i. Rendering projects the Gaussians to screen space and alpha-composites them front to back,

$$C(\mathbf{p}) \;=\; \sum_{i \in \mathcal{S}(\mathbf{p})} c_i \, \alpha_i \prod_{j<i} (1-\alpha_j), \tag{1}$$

where α_i combines o_i with the projected footprint and $\mathcal{S}(\mathbf{p})$ is the depth-sorted set of Gaussians covering pixel $\mathbf{p}$. Our method changes only the computation of the per-Gaussian color c_i, leaving the rasterizer untouched.

Diffusion-based light transport. Where scattering dominates absorption, light loses its directionality after a few events and multiple scattering is well approximated by diffusion of the fluence ϕ, the direction-averaged light density — the zeroth angular moment of the radiative transfer equation, kept to first order (P1) (Ishimaru, 1978; Stam, 1995; Jensen et al., 2001). The approximation is weakest at the boundary and for the first, still-directional event, which we therefore keep separate (Section 2.4).

2.2 Relighting formulation

Under a point light with incident direction $\boldsymbol{\omega}_l$, view direction $\boldsymbol{\omega}_o$, and distance d_i to Gaussian g_i, the outgoing radiance of the primitive is the sum of shadowed direct reflection and subsurface scattering,

$$c_i \;=\; L_i^{\mathrm{d}} + L_i^{\mathrm{s}} + L_i^{\mathrm{sss}}, \qquad L_i^{\mathrm{d}} \;=\; \frac{D_i}{d_i^2}\, \mathbf{a}_i \, \langle \mathbf{n}_i, \boldsymbol{\omega}_l \rangle, \qquad L_i^{\mathrm{s}} \;=\; \frac{D_i}{d_i^2}\, f_i^{\mathrm{s}}(\boldsymbol{\omega}_l, \boldsymbol{\omega}_o), \tag{2}$$

where $1/d_i^2$ is the point light's inverse-square falloff and the learned decay $D_i \in [0,1]$ is the additional attenuation from shadowing. The diffuse term is Lambertian with per-Gaussian albedo $\mathbf{a}_i$ and a learned shading normal $\mathbf{n}_i$, and the specular lobe f_i^{s} is a mixture of anisotropic spherical Gaussians (Xu et al., 2013) whose amplitudes are conditioned on the light direction, an expressive basis for stretched and anisotropic highlights (Bi et al., 2024) in place of an explicit microfacet BRDF (Walter et al., 2007). The decay multiplies the two direct terms only. Light that leaves through the subsurface channel has already been attenuated inside the solve, by the march that builds its source and by absorption along the path, so applying D_i to it as well would attenuate the same travel twice; Section 3.3 measures the alternative. D_i is predicted from a geometric visibility V_i splatted from the light's viewpoint, together with transport cues (Section 2.5), and the colored Gaussians are composited by Eq. 1. Section 2.4 defines L_i^{sss}.

2.3 Gaussian-Supported Transport Domain

Transport Domain. We lay a 64^3 grid over an axis-aligned box around the cloud and rasterize the spatial support of the Gaussians into a fractional occupancy field $O(\mathbf{x}) \in [0,1]$ on it, taking the occupied cells as the transport domain Ω. The occupancy is kept fractional inside the domain rather than rounded to a binary mask, because the boundary term of the finite-volume operator introduced below scales with interface area, which a binary mask overcounts (Johansen & Colella, 1998). A low occupancy cutoff decides which cells enter Ω (Appendix C). Gaussians fit the surface, so rasterizing them leaves a hollow shell, and a shell has no interior for light to cross. The interior is therefore filled by a generalized winding number over a subsample of them (Barill et al., 2018), which separates inside from outside without a watertight mesh. The per-Gaussian coefficients are then splatted onto the same grid with a kernel spanning several cells. Each interior cell takes the coefficients of the nearest surface cell, propagated inward on a coarser grid, the construction used for extension velocities in level-set methods (Adalsteinsson & Sethian, 1999) (Appendix C).

Gradients flow back through this splatting, and the domain itself adds no learnable parameters: it is read off the Gaussians at every rebuild. Following the classical two-part decomposition, the multiple-scattering solve is driven by the singly scattered light (Stam, 1995; Arbree et al., 2011). For each active cell $\mathbf{c}$ we march toward the light through O and accumulate $\tau(\mathbf{c}) = \int O\, ds$, the path length weighted by occupancy rather than the geometric distance, so that empty space between disconnected parts of the object contributes nothing to the extinction, and set

$$E(\mathbf{c}) = \frac{e^{-\sigma_t(\mathbf{c})\,\tau(\mathbf{c})}}{d(\mathbf{c})^2}, \qquad b(\mathbf{c}) = \sigma_s(\mathbf{c})\,E(\mathbf{c})\,O(\mathbf{c}). \tag{3}$$

Here E is the light that survives the march to the cell, and b is the density of first-scattering events there, which acts as the source term of the solve.

Medium Coefficients. A small MLP maps each Gaussian's encoded position, its per-primitive latent, and the encoded light and view directions to per-channel scattering and absorption coefficients,

$$\big(\boldsymbol{\sigma}_s(g_i),\, \boldsymbol{\sigma}_a(g_i)\big) \;=\; \mathrm{MLP}_{\mathrm{mat}}\big(\gamma(\mathbf{x}_i),\, \mathbf{z}_i,\, \gamma(\boldsymbol{\omega}_l),\, \gamma(\boldsymbol{\omega}_o)\big) \;\in\; \mathbb{R}^3 \times \mathbb{R}^3, \tag{4}$$

where g_i is the i-th Gaussian, γ a frequency encoding and $\mathbf{z}_i$ its per-Gaussian latent (Appendix C). Extinction is $\sigma_t = \sigma_s + \sigma_a$ and the diffusion coefficient it implies is $\kappa = 1/(3\sigma_t)$, both per channel. We refer to σ_s, σ_a and σ_t collectively as the medium coefficients.

2.4 Volumetric Subsurface Scattering

Source and Diffusion Solve. We solve for the fluence ϕ, one scalar per cell and per color channel on Ω, which obeys

$$-\nabla\cdot\big(\kappa(\mathbf{x})\,\nabla\phi\big) + \sigma_a(\mathbf{x})\,\phi = b(\mathbf{x}) \quad \text{on } \Omega, \qquad \kappa\,\partial_n\phi + \tfrac{1}{2A(\eta)}\,\phi = 0 \quad \text{on } \partial\Omega, \tag{5}$$

The second equation is a Robin condition stating that a fraction of the light reaching the surface leaves and the rest is internally reflected, where $A(\eta)$ is the classical internal-reflection factor, equal to one at $\eta = 1$ (Jensen et al., 2001; Arbree et al., 2011). Integrating Eq. 5 over one cell and splitting

its faces into those shared with another active cell and those that meet air gives the standard finite-volume stencil (Patankar, 1980), per cell,

$$\underbrace{\frac{1}{h^2}\sum_{d\in\mathcal{N}(c)}\kappa_{cd}\,\alpha_{cd}\,(\phi_c-\phi_d)}_{D}\;+\;\underbrace{\sigma_a(c)\,O(c)\,\phi_c}_{M}\;+\;\underbrace{\frac{|\nabla O|_c}{2A(\eta)\,h}\,\phi_c}_{S}\;=\;b(c), \tag{6}$$

with h the cell size, ϕ_c the fluence in cell c, $\mathcal{N}(c)$ its six neighbours, $\kappa_{cd}=2\kappa_c\kappa_d/(\kappa_c+\kappa_d)$ the finite-volume face conductance of two half cells and $\alpha_{cd}=\min(O_c,O_d)$ the open fraction of that face. The factor $|\nabla O|_c$ in the third block is the interface area the cell presents to the outside, which is why a fractional occupancy matters: a binary mask puts the whole face area on every boundary cell. D transports fluence between neighbouring cells and is driven by the difference $\phi_c-\phi_d$, M absorbs it, and S lets it escape through the faces that meet air. Stacking the rows gives $F\phi=b$ with $F=D+M+S$, where D discretizes the divergence term of Eq. 5, M its absorption term, and S its Robin boundary condition.

We solve this system with preconditioned conjugate gradients rather than forming F^{-1}, and we obtain gradients without storing the iterations. The operator does not depend on the light, so a scene lit by several sources is the sum of per-light solves, which is how Appendix D.3 relights under an environment map without retraining. Differentiating $F\phi=b$ gives $F\,\partial\phi=\partial b-(\partial F)\,\phi$, so the sensitivity of ϕ is itself a solve against the same operator. Because F is symmetric no transpose is needed, and solving $F\lambda=\partial\mathcal{L}/\partial\phi$ once yields the gradients with respect to b and to the diagonal of $M+S$ (Propositions 1 and 2). The backward pass therefore costs one extra solve and its memory does not grow with the iteration count. Differentiating a transport solve through its adjoint rather than through its iterations is the standard treatment in differentiable rendering (Nimier-David et al., 2019; 2020; Vicini et al., 2021). Here the operator is assembled on the Gaussians' own occupancy. The face conductances are held at stop-gradient, so the medium is learned through the terms it enters directly.

Exit Radiance. Multiply scattered radiance leaves the boundary as

$$L_{\mathrm{ms}}(g_i)=\frac{1+1/A(\eta)}{4\pi}\;F_t(\cos\theta_o)\;\phi\big(\mathbf{x}_i-\delta\,\mathbf{n}_i\big), \tag{7}$$

with F_t the Fresnel transmittance and the fluence sampled one cell inward ($\delta=h$), since the splats sit on the voxelized boundary where interpolation would mix interior values with the zeros outside. We set the exit index to $\eta=1$ on every scene, so $A(\eta)$ and F_t reduce to identity and the boundary scale is absorbed by the calibration below.

Diffusion is angle-averaged and does not represent the once-scattered light near the surface, which the classical decomposition adds separately (Hanrahan & Krueger, 1993; Jensen et al., 2001). Along the view ray through g_i, in material depth u,

$$L_1(g_i)=\frac{1}{4\pi}\int_0^{U_i}Q\big(\mathbf{x}(u)\big)\;e^{-\sigma_t u}\,du,\qquad Q=\sigma_s\,E, \tag{8}$$

where U_i is the material chord, the part of the view ray that lies inside the medium, marched on the same occupancy field as τ so that the two path lengths are measured consistently. Q is the same source the solve is driven from, so the term costs one 32-sample quadrature and no new state. The channel a Gaussian finally emits is

$$L_i^{\mathrm{sss}}\;=\;\mathbf{k}_i^{\mathrm{sss}}\odot g_{\mathrm{cal}}\,[L_1(g_i)+L_{\mathrm{ms}}(g_i)]\,, \tag{9}$$

where $\mathbf{k}_i^{\mathrm{sss}}$ is a learned per-primitive, per-channel amplitude and g_{cal} one learned scalar per scene, both calibrating solver units to rasterizer radiometry. Where the medium is thin the solve's support collapses onto the primitive's own neighbourhood and the channel carries almost nothing, so the model falls back to local shading rather than colliding with it (Remark 2).

The coefficients are learned from images and conditioned on the light and view directions, so they are not optical constants of the material.

2.5 Shadow and Specular Disambiguation

Both the specular basis and the shadow decay are learned, and a highlight that a shadow cancels leaves no trace in the rendered image, so the two can trade against each other freely. The specular component can therefore explain smooth transported illumination whenever the shadow term suppresses the direct component, which is exactly the appearance the solve of Section 2.4 should own.

Shadowing has two stages. We first estimate geometric visibility from the light by rendering the Gaussian scene with the rasterizer of Eq. 1.

$$V_i \;=\; \frac{\sum_{\mathbf{p}} G_i(\mathbf{p})\, T_i(\mathbf{p})}{\sum_{\mathbf{p}} G_i(\mathbf{p})}, \qquad T_i(\mathbf{p}) \;=\; \prod_{j \prec i} \big(1 - \alpha_j(\mathbf{p})\big), \tag{10}$$

where $\mathbf{p}$ ranges over the light-view pixels in the footprint G_i of g_i and $j \prec i$ are the Gaussians closer to the light. $V_i \in [0,1]$ is a transmittance, one when nothing lies between g_i and the light and zero when it is fully occluded. A small network then predicts the attenuation $D_i \in [0,1]$ from this visibility together with transmittance cues taken from the transport field,

$$D_i \;=\; \sigma\Big(g_\theta\big(\gamma(\mathbf{x}_i),\, \gamma(\boldsymbol{\omega}_l),\, \mathbf{z}_i,\, \mathbf{h}_i\big)\Big), \qquad \mathbf{h}_i \;=\; \big[\, V_i,\;\; e^{-\bar{\sigma}_{t,i}\,\tau_i},\;\; e^{-\bar{\sigma}_{t,i}\,U_i},\;\; r_i,\;\; \chi_i \,\big], \tag{11}$$

with $\mathbf{z}_i$ the same per-Gaussian latent as in Eq. 4 and the hint vector $\mathbf{h}_i$ gradient-stopped, for the same reason as the face conductances: the shadow branch reads the transport field, and letting it push gradients back would let it reshape the medium to make shadows cheaper, fitting σ_t to the shadow residual rather than to transport. τ_i and U_i are the material path lengths toward the light and along the view chord, marched on the same occupancy grid as the transport, and $\bar{\sigma}_{t,i}$ is the primitive's mean extinction.

Both the shadow decay and the transport solve therefore depend on the same extinction field σ_t. The last two entries, the channel's luminance relative to the undecayed direct term and its red chroma, indicate the share of the local light that is transported rather than reflected.

We break the ambiguity with a shadow-aware prior on the rendered images,

$$\mathcal{L}_{\text{leak}} \;=\; \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} \tilde{S}(\mathbf{p})\, \big(1 - \mathrm{sg}[\tilde{D}(\mathbf{p})]\big), \tag{12}$$

where $\mathcal{P}$ is the set of pixels the object covers, $\tilde{S}$ and $\tilde{D}$ are the per-Gaussian specular radiance and decay after alpha compositing along the ray, and $\mathrm{sg}[\cdot]$ stops the gradient, so the penalty acts on the specular term only and cannot be avoided by brightening shadows.

2.6 Training

All components are trained jointly from scratch. The objective is

$$\mathcal{L}_{\text{total}} \;=\; \mathcal{L}_{\text{photo}} \;+\; \lambda_\alpha\, \mathcal{L}_\alpha \;+\; \lambda_{\text{leak}}\, \mathcal{L}_{\text{leak}} \;+\; \lambda_{\text{med}}\, \mathcal{L}_{\text{med}}, \tag{13}$$

where $\mathcal{L}_{\text{photo}}$ combines ℓ_1 and D-SSIM as in 3DGS (Kerbl et al., 2023), $\lambda_{\text{med}} = 0.05$, $\mathcal{L}_\alpha$ is an ℓ_1 loss between the accumulated opacity and the capture's alpha channel (λ_α=0.2), $\mathcal{L}_{\text{leak}}$ is the regularizer of Eq. 12 (λ_{leak}=0.05), and $\mathcal{L}_{\text{med}}$ regularizes the learned medium by smoothing $(\log \sigma_t, \text{albedo})$ across neighboring Gaussians and constraining Gaussian scale relative to the transport grid. The scale term is what keeps the two representations compatible. A splat carries one fluence value, read at one point, and one chord length, so a splat spanning many cells sits across regions with different ϕ and is described by none of them. We utilize a progressive training strategy for modeling SSS after constructing a fundamental geometry and appearance following SSD-GS (Zheng et al., 2026). The subsurface channel is switched on after a 9k-iteration warm-up, during which the solve is skipped and only the surface terms and the geometry are fit. After it, every step runs one forward and one adjoint solve, and the transport domain is rebuilt whenever densification changes the Gaussian count and in any case every 300 renders.

3 EXPERIMENTS

3.1 SETUP

Datasets. We evaluate on three OLAT datasets covering subsurface-dominant, opaque, and anisotropic material regimes. The **SSS-GS** benchmark (Dihlmann et al., 2024), built for subsurface scattering, provides five synthetic translucent objects and three real light-stage captures. The **NRHints** set (Zeng et al., 2023) contributes seven real captured objects, most of which are not strongly translucent. The **GS^3** synthetic set (Bi et al., 2024) adds six scenes spanning metal, fur, and mixed materials.

Baselines. The methods that appear in qualitative comparisons are retrained from public implementations under matched resolution, splits, and training budgets. More details are listed in Appendix C. These are GS^3 (Bi et al., 2024), a general relighting system, SSD-GS (Zheng et al., 2026), a recent relightable Gaussian baseline with an explicit subsurface component that is the local-kernel counterpoint to our solve, RNG (Fan et al., 2025), and SSS-GS (Dihlmann et al., 2024), which introduces a dedicated subsurface scattering component. The per-scene tables in Appendix D.1 also list 3DGS (Kerbl et al., 2023) and GI-GS (Chen et al., 2025) with the numbers reported by Zheng et al. (2026) as additional reference points; both fall well below the retrained baselines. We use a fixed hyperparameter configuration across all scenes, following each baseline's official codebase.

Implementation. All timings are measured on one A6000 in steady state (median over frames 20 onward of a 120-frame pass on an otherwise idle card, warm-up discarded; the per-scene transfer basis is precomputed once and excluded). Under this protocol VolS-GS renders the five SSS-GS synthetic scenes at 64.5 ± 4.9 FPS. On the SSS-GS synthetic scenes, training takes under 3.5 hours and under 8 GiB of GPU memory per scene. Per-scene frame times, primitive counts and the rest of the implementation are in Appendix C.

Table 1: **Main results.** Left: the eight subsurface scenes, per scene, every method at the 60k protocol iteration. Right: the remaining two datasets as test-split means over their 7 and 6 scenes (per-scene numbers trained at 100k in Appendix D.1). Best red, second orange.

	Method	BUNNY	DRAGON	CANDLE	SOAP	STATUE	CRYSTAL	JOYFULYELL	PHYSALIS	NRHints dataset mean	GS^3 dataset mean
PSNR↑	GS^3	34.7109	36.1753	38.3707	38.3713	35.3291	35.2767	31.5467	28.9163	31.94	31.11
	SSS-GS	36.0035	34.7397	38.3432	36.4006	35.5485	34.6201	31.4973	28.9331	25.16	28.23
	RNG	35.3300	34.6134	40.3420	38.4962	29.2148	33.6849	30.0986	28.6492	30.74	28.64
	SSD-GS	38.9147	37.7832	41.9214	38.2959	35.9034	35.2096	31.8938	29.5474	32.05	31.01
	VolS-GS (Ours)	40.1453	38.0579	43.6366	41.0098	38.3515	35.6278	32.1856	30.2210	32.37	31.59
SSIM↑	GS^3	0.9811	0.9769	0.9882	0.9852	0.9815	0.9813	0.9665	0.9535	0.9500	0.9676
	SSS-GS	0.9797	0.9596	0.9864	0.9762	0.9765	0.9776	0.9603	0.9461	0.8499	0.9452
	RNG	0.9859	0.9715	0.9940	0.9896	0.9647	0.9743	0.9582	0.9483	0.9272	0.9560
	SSD-GS	0.9907	0.9828	0.9942	0.9884	0.9845	0.9808	0.9676	0.9585	0.9504	0.9672
	VolS-GS (Ours)	0.9920	0.9835	0.9960	0.9935	0.9899	0.9820	0.9690	0.9590	0.9513	0.9702
LPIPS↓	GS^3	0.0239	0.0251	0.0167	0.0243	0.0207	0.0196	0.0419	0.0479	0.0795	0.0364
	SSS-GS	0.0323	0.0539	0.0222	0.0286	0.0368	0.0231	0.0468	0.0542	0.1589	0.0649
	RNG	0.0177	0.0312	0.0080	0.0116	0.0371	0.0291	0.0492	0.0551	0.0998	0.0465
	SSD-GS	0.0129	0.0199	0.0077	0.0186	0.0185	0.0205	0.0411	0.0470	0.0802	0.0362
	VolS-GS (Ours)	0.0112	0.0188	0.0060	0.0077	0.0130	0.0189	0.0385	0.0444	0.0779	0.0330

3.2 MAIN RESULTS

Quantitative Results. Table 1 shows that VolS-GS achieves the best average performance across all three datasets. All per-scene metrics are provided in Appendix D.1. The improvement is most consistent on the SSS-GS benchmark, where subsurface scattering is the dominant effect. The gains also carry over to NRHints and GS^3, which contain less translucent and more diverse materials.

Qualitative Results. Figure 4 shows relighting results across translucent and opaque objects. GS^3 and SSS-GS reproduce the overall appearance but their predictions become less accurate around shadows and fine appearance details. SSS-GS in particular does not optimise stably on four of the seven NRHints scenes, and we report and show those rows as measured. RNG also captures the overall illumination but tends to lose high-frequency details, particularly on LEGO and CAT. SSD-GS

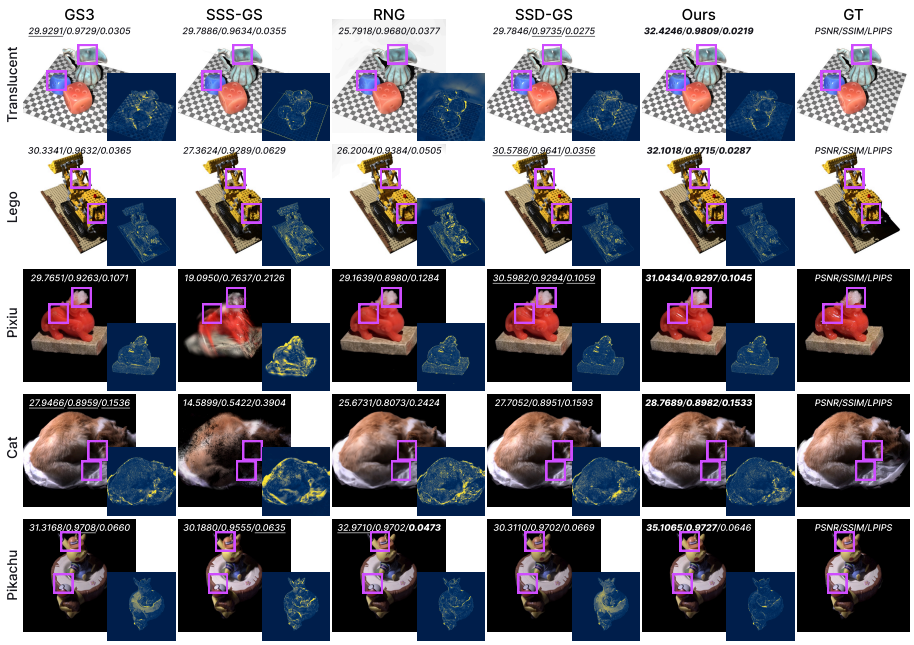


Figure 4: Qualitative relighting comparison under novel lighting across different scenes. TRANSLUCENT and LEGO are synthetic scenes from GS^3, while PIXIU, CAT, and PIKACHU are real captures from NRHints. Per-frame PSNR/SSIM/LPIPS are printed on each panel, **best** in bold and second underlined within a row. The inset is the error map $|\text{render} - \text{GT}|$ averaged over color channels and clipped at 0.15, on one ramp shared by every panel.

produces competitive results but still struggles with intricate shadow patterns and complex subsurface effects.

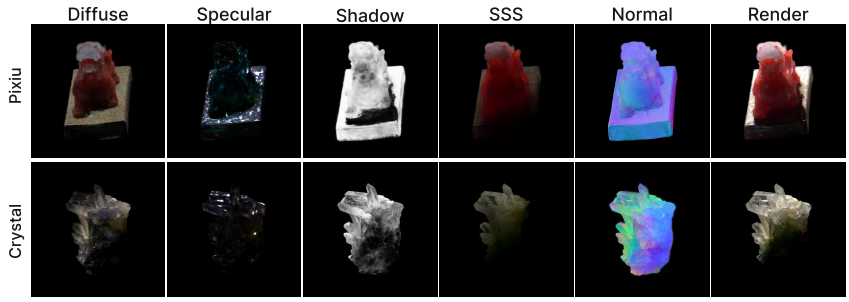


Figure 5: **Material decomposition.** Our own channels on two real captures, PIXIU (top) and CRYSTAL (bottom), one row per scene. The specular column is amplified only where it is already bright ($\times 2.5$ above its own 95th percentile), identically in both rows.

Material Decomposition. Figure 5 shows the learned decomposition into diffuse, specular, shadow, SSS, and normal components. The SSS channel captures the broad glow visible in backlit regions while the specular and shadow branches remain more localized. More detailed investigations across SSD-GS and SSS-GS are in Appendix H.

3.3 ABLATIONS

We ablate the main components of VolS-GS to understand their individual roles. We consider four variants: (1) $-$Volumetric SSS, which removes the volumetric scattering component; (2) $-$transport cues on shadow, which prevents the shadow branch from using transport-derived cues; (3) shadow

Table 2: **Ablations** on Bunny and Statue where SSS and shadow are prevalent. Each row retrains the final configuration with one component changed.

	BUNNY			STATUE		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Full model	**40.15**	**0.9920**	**0.0112**	**38.35**	**0.9899**	**0.0130**
− Volumetric SSS	34.67	0.9812	0.0283	36.19	0.9852	0.0173
− Transport cues on shadow	39.17	0.9910	0.0126	35.09	0.9818	0.0211
Shadow term multiplied onto SSS	39.02	0.9901	0.0132	37.66	0.9885	0.0144
− Shadow term	36.30	0.9859	0.0179	34.41	0.9794	0.0230

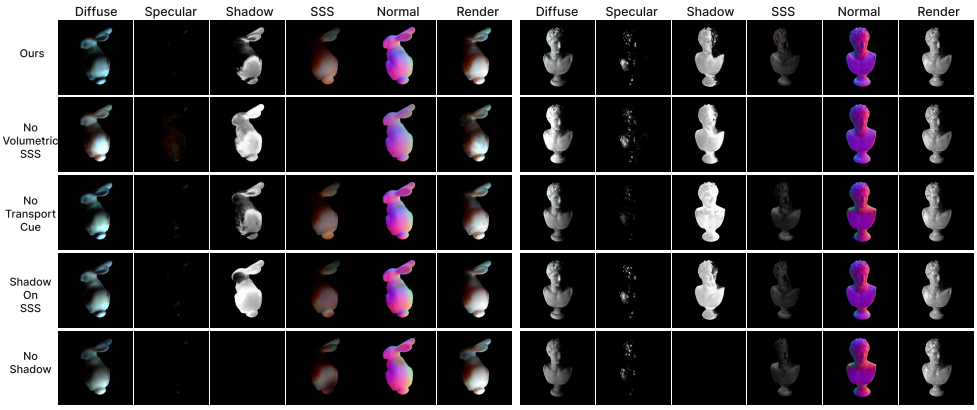


Figure 6: **Ablation study.** Material decomposition on BUNNY (left) and STATUE (right) from the SSS-GS dataset. Rows are the full model and the four single-change variants of Table 2.

term multiplied onto SSS, which applies the learned shadow to the SSS component as well; and (4) −shadow term, which removes the learned shadow component.

Figure 6 and Table 2 give the visual and the metric comparison of these variants. Removing the volumetric SSS channel (1) changes the rendered appearance of BUNNY and STATUE wherever transported light contributes. Removing the transport cues (2) changes the shadow decomposition, most visibly on STATUE, where the predicted shadow no longer follows the same attenuation pattern. Multiplying the shadow onto the SSS channel (3) changes the render only slightly but rewrites the shadow channel, degrading its interpretability, and the cost on held-out lights is visible in Table 2. Removing the shadow term (4) flattens the shading. The specular leakage regularizer contributes to cleaner decomposition, whose investigation is reported in Appendix F.

4 CONCLUSION

We presented VolS-GS, a relightable Gaussian representation that augments local shading with a learned volumetric subsurface scattering, enabling light propagation across the spatial support of the scene. The shadow term reads cues from the same transport and a prior keeps the specular lobe out of unlit pixels, so the components stay separable and the learned decomposition is interpretable. Across three OLAT benchmarks, VolS-GS achieves high visual quality while reproducing the light scattered inside the object, and the resulting reconstructions relight faithfully under novel lights and views. We recommend that readers refer to the supplementary video.

Limitations and future work. Since our solve for volumetric subsurface scattering is based on a diffusion approximation, it may not suffice where the radiance is still directional, and it carries no path for light exchanged between surfaces on concave regions rather than through the medium (Figure 12). We leave directional subsurface scattering in a discrete volume regime and unified representation for diverse non-local effects as interesting future work.

AI USE STATEMENT

In the preparation of this work, we used LLMs to polish the text, including grammar checking and phrase refinement, to polish figure illustrations, and to help debug code. LLMs did not contribute to any of the scientific development.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The method and the assumptions it rests on are stated in the main paper. Its complete proofs, implementation settings, training schedules, per-scene results for every method and the evaluation protocol are all described in Appendix.

The codebase including VolS-GS training and render with the same protocol will be released.

REFERENCES

David Adalsteinsson and James A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2–22, 1999.

Adam Arbree, Bruce Walter, and Kavita Bala. Heterogeneous subsurface scattering using the finite element method. *IEEE Transactions on Visualization and Computer Graphics*, 17(7):956–969, 2011.

Gavin Barill, Neil G. Dickson, Ryan Schmidt, David I. W. Levin, and Alec Jacobson. Fast winding numbers for soups and clouds. *ACM Trans. Graph.*, 37(4):43:1–43:12, 2018.

Zoubin Bi, Yixin Zeng, Chong Zeng, Fan Pei, Xiang Feng, Kun Zhou, and Hongzhi Wu. GS^3: Efficient relighting with triple gaussian splatting. In *ACM SIGGRAPH Asia Conference Proceedings*, 2024.

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/CVF International Conference on Computer Vision (ICCV)*, 2021.

J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.

Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Towards learning-based inverse subsurface scattering. In *IEEE International Conference on Computational Photography (ICCP)*, 2020.

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.

Per H. Christensen and Brent Burley. Approximate reflectance profiles for efficient subsurface scattering. Technical Report 15-04, Pixar Animation Studios, 2015.

Xi Deng, Fujun Luan, Bruce Walter, Kavita Bala, and Steve Marschner. Reconstructing translucent objects using differentiable rendering. In *ACM SIGGRAPH Conference Proceedings*, 2022.

Eugene d'Eon and Geoffrey Irving. A quantized-diffusion model for rendering translucent materials. *ACM Trans. Graph.*, 30(4), 2011.

Jan-Niklas Dihlmann, Arjun Majumdar, Andreas Engelhardt, Raphael Braun, and Hendrik P.A. Lensch. Subsurface scattering for gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.*, 24(3), 2005.

Jiahui Fan, Fujun Luan, Jian Yang, Miloš Hašan, and Beibei Wang. RNG: Relightable neural gaussians. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

Jeppe Revall Frisvad, Toshiya Hachisuka, and Thomas Kim Kjeldsen. Directional dipole model for subsurface scattering. *ACM Transactions on Graphics*, 34(1), 2014.

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. *arXiv preprint arXiv:2311.16043*, 2023.

Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics*, 32(6), 2013.

Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. IRGS: Inter-reflective gaussian splatting with 2D gaussian ray tracing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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*, 2024.

Ralf Habel, Per H. Christensen, and Wojciech Jarosz. Photon beam diffusion: A hybrid monte carlo method for subsurface scattering. *Computer Graphics Forum (Proceedings of EGSR)*, 32 (4), 2013.

Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *SIGGRAPH*, 1993.

Akira Ishimaru. *Wave Propagation and Scattering in Random Media*. Academic Press, 1978.

Henrik Wann Jensen and Juan Buhler. A rapid hierarchical rendering technique for translucent materials. In *ACM SIGGRAPH*, 2002.

Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of SIGGRAPH*, 2001.

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 (Proc. SIGGRAPH Asia)*, 2025. arXiv:2509.18497.

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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. TensoIR: Tensorial inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

Hans Johansen and Phillip Colella. A cartesian grid embedded boundary method for poisson's equation on irregular domains. *Journal of Computational Physics*, 147(1):60–85, 1998.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 2023.

Zhiyi Kuang, Yanchao Yang, Siyan Dong, Jiayue Ma, Hongbo Fu, and Youyi Zheng. OLAT Gaussians for generic relightable appearance acquisition. In *SIGGRAPH Asia 2024 Conference Papers*, 2024.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. GS-IR: 3D gaussian splatting for inverse rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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 *European Conference on Computer Vision (ECCV)*, 2020.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics*, 38(6), 2019.

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. Radiative backpropagation: An adjoint method for lightning-fast differentiable rendering. *ACM Transactions on Graphics*, 39(4), 2020.

Suhas V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing, 1980.

Christophe Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum*, 13(3), 1994.

Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics*, 21(3), 2002.

Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Jos Stam. Multiple scattering as a diffusion process. In *Eurographics Workshop on Rendering*, 1995.

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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Delio Vicini, Vladlen Koltun, and Wenzel Jakob. A learned shape-adaptive subsurface scattering model. *ACM Transactions on Graphics*, 38(4), 2019.

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. Path replay backpropagation: Differentiating light paths using constant memory and linear time. *ACM Transactions on Graphics*, 40(4), 2021.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Eurographics Symposium on Rendering*, 2007.

Jiaping Wang, Shuang Zhao, Xin Tong, Stephen Lin, Zhouchen Lin, Yue Dong, Baining Guo, and Heung-Yeung Shum. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Transactions on Graphics*, 27(1):9:1–9:18, 2008.

Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Transactions on Graphics*, 32(6), 2013.

Yuxuan Yao, Zixuan Zeng, Chun Gu, Xiatian Zhu, and Li Zhang. Reflective Gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2025.

Hong-Xing Yu, Michelle Guo, Alireza Fathi, Yen-Yu Chang, Eric Ryan Chan, Ruohan Gao, Thomas Funkhouser, and Jiajun Wu. Learning object-centric neural scattering functions for free-viewpoint relighting and scene composition. *Transactions on Machine Learning Research*, 2023.

Youwen Yuan and Xi Zhao. GTSR: Subsurface scattering awared 3D Gaussians for translucent surface reconstruction. *arXiv preprint arXiv:2603.22036*, 2026.

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 Conference Proceedings*, 2023.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021a.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. NeRFactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics*, 40(6), 2021b.

Iris Zheng, Guojun Tang, Alexander Doronin, Paul Teal, and Fang-Lue Zhang. SSD-GS: Scattering and shadow decomposition for relightable 3D gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2026.

A Related work

A.1 Relightable neural and Gaussian representations

Relighting a captured object means re-rendering it under light it was never photographed under. On neural fields (Mildenhall et al., 2020) one line recovers an explicit reflectance model, pairing reflectance with visibility (Srinivasan et al., 2021), fitting spherical-Gaussian BRDFs (Zhang et al., 2021a), separating shape from material under unknown illumination (Boss et al., 2021; Zhang et al., 2021b), or reparameterising the view-dependent term for specular surfaces (Verbin et al., 2022). Another keeps appearance implicit and conditions it on the light, the setting we work in, learned from point-lit images (Zeng et al., 2023) or carried on a tensorial field (Jin et al., 2023).

3D Gaussian splatting (Kerbl et al., 2023) moved relighting onto primitives that rasterize in real time. Relightable 3D Gaussians (Gao et al., 2023) and GaussianShader (Jiang et al., 2024) give each Gaussian BRDF parameters under environment lighting. In the one-light-at-a-time (OLAT) setting, OLAT Gaussians (Kuang et al., 2024) condition a network on the light and hint it from a proxy mesh, GS^3 (Bi et al., 2024) adds a Lambertian lobe with an anisotropic specular basis and a learned residual, RNG (Fan et al., 2025) drops the explicit model altogether, and SSD-GS (Zheng et al., 2026) contributes an MLP-refined shadow field and a subsurface term. We keep this family's pipeline and protocol. GS^3's residual absorbs transported light along with everything else its explicit terms omit, so no subsurface component can be read out of it.

A.2 Non-local light transport in Gaussian scenes

Inverse rendering on Gaussian primitives recovers geometry, material and lighting jointly: GS-IR (Liang et al., 2024) bakes occlusion into a per-Gaussian representation to make the rendering equation tractable, and IRGS (Gu et al., 2025) evaluates incident radiance on the fly with differentiable 2D Gaussian ray tracing so that secondary bounces are not approximated away. Several recent Gaussian methods explicitly model light transport between primitives. PRTGS (Guo et al., 2024) precomputes the radiance transfer of Gaussian splats so that soft shadows and indirect illumination can be evaluated in real time under low-frequency environment lighting. Ref-Gaussian (Yao et al., 2025) adds physically based deferred shading and Gaussian-grounded inter-reflection for glossy objects. Radiosity-GS (Jiang et al., 2025) makes a radiosity solve differentiable on Gaussian surfels and evaluates it under the same point-light capture we use. These methods transport light between surfaces, whereas subsurface scattering transports light through the interior of the object.

A.3 Subsurface scattering

A BSSRDF integrates over every point at which light could have entered. Analytic kernels assume a homogeneous, semi-infinite medium under uniform illumination, collapsing that integral into a function of the entry-to-exit distance (Hanrahan & Krueger, 1993; Jensen et al., 2001; Donner & Jensen, 2005; d'Eon & Irving, 2011; Habel et al., 2013). Locality is a property of the evaluation, not of these models: they also integrate the entry surface hierarchically (Jensen & Buhler, 2002), restore the dropped directional dependence (Frisvad et al., 2014), and fit profiles for the same integral (Christensen & Burley, 2015).

For spatially varying media the field turned to numerical solvers, from the diffusion equation on blobs and grids (Stam, 1995) to finite differences on a PolyGrid (Wang et al., 2008) and a finite-element treatment with a diffusive-source boundary (Arbree et al., 2011). Ours is that equation, discretized on the Gaussian scene's own occupancy. A separate line recovers the medium from images instead of assuming it (Gkioulekas et al., 2013; Che et al., 2020; Deng et al., 2022).

A newer line learns subsurface appearance instead of solving for it, relaxing the planar boundary (Vicini et al., 2019), rendering object scattering functions in real time (Yu et al., 2023), fitting scattering-aware Gaussians (Yuan & Zhao, 2026), or pairing explicit shading with an implicit residual (Dihlmann et al., 2024), which also contributes our benchmark. SSD-GS (Zheng et al., 2026) evaluates a dipole at each Gaussian. In VolS-GS the coefficients are still learned for each primitive. Where the light goes is decided by the solve.

B THEORETICAL PROPERTIES

This appendix states the properties referenced in Sections 2.3 and 2.4. Throughout, $\sigma_t = \sigma_s + \sigma_a$ and $\kappa = 1/(3\sigma_t)$ are as defined in Eq. 4, $\alpha = \sigma_s/\sigma_t \in (0, 1)$ is the single-scattering albedo, and $\mathbf{J} = -\kappa\nabla\phi$ denotes the P_1 flux vector, not the scalar source Q of Eq. 8. Phase is isotropic. An anisotropic phase reduces at the diffusion level to $\sigma_s' = (1 - g)\sigma_s$ (Stam, 1995; Ishimaru, 1978), so g is not separately identifiable from the reduced coefficient our network already predicts; we therefore fix $g = 0$.

Proposition 1 (Well-posedness and stability of the discrete solve)**.** *Let $F = D+M+S$ be the finite-volume operator of Eq. 5: D the divergence part with harmonic-mean face conductances $\kappa_f > 0$, $M = \mathrm{diag}(\sigma_a \Delta V)$ with $\sigma_a > 0$, and S the Robin boundary part with nonnegative diagonal entries. Then F is symmetric positive definite, the discrete solution $\phi = F^{-1}b$ exists and is unique, and $\|\phi\| \le \|b\|/\lambda_{\min}(M)$.*

Proof. D is a weighted graph Laplacian on the active cells ($\mathbf{v}^\top D\mathbf{v} = \sum_f \kappa_f (v_i - v_j)^2/\Delta x^2 \ge 0$) and S is diagonal and nonnegative. M is diagonal with strictly positive entries because the albedo map bounds $\alpha \le 0.999$, hence $\sigma_a \ge (1 - 0.999)\,\sigma_{\min} > 0$. The sum is therefore symmetric positive definite, the CG iteration is well defined and convergent, and $\lambda_{\min}(F) \ge \lambda_{\min}(M)$ bounds the solve. □

Proposition 2 (Adjoint gradients for the full operator)**.** *For a scalar loss $\mathcal{L}(\phi)$ with $\phi = F^{-1}b$, the parameter gradients are $\partial_b \mathcal{L} = F^{-1}\nabla_\phi \mathcal{L}$ and, for any parameter θ entering F, $\partial_\theta \mathcal{L} = -\left(F^{-1}\nabla_\phi \mathcal{L}\right)^\top (\partial_\theta F)\,\phi$. Since F is symmetric, the adjoint system is the same operator, so back-propagation through the solve costs one additional CG solve.*

Proof. Differentiating $F\phi = b$ gives $d\phi = F^{-1}(db - (dF)\phi)$, and setting $\boldsymbol{\lambda} = F^{-\top}\nabla_\phi \mathcal{L} = F^{-1}\nabla_\phi \mathcal{L}$ (symmetry) gives both expressions. This is the standard adjoint-state method. □

Lemma 1 (Reciprocity in heterogeneous media)**.** *The variable-coefficient diffusion operator $\mathcal{A}\phi = -\nabla\cdot(\kappa(\mathbf{x})\nabla\phi) + \sigma_a(\mathbf{x})\phi$ with the Robin condition of Eq. 5 is self-adjoint; its Green's function satisfies $G(\mathbf{x}, \mathbf{x}') = G(\mathbf{x}', \mathbf{x})$ for arbitrary $\kappa > 0$, $\sigma_a \ge 0$.*

Proof. For ϕ, ψ satisfying the Robin condition, $\langle \mathcal{A}\phi, \psi\rangle - \langle \phi, \mathcal{A}\psi\rangle = \oint \kappa\,(\phi\,\partial_n\psi - \psi\,\partial_n\phi)\,dS = \oint \frac{1}{2A}(\phi\psi - \psi\phi)\,dS = 0$, using Green's identity and then the boundary condition to substitute $\kappa\,\partial_n\phi = -\phi/(2A)$. The zeroth-order term is symmetric trivially, and symmetry of G follows in the standard way. Our discretization preserves it exactly (Proposition 1: $F = F^\top$). □

Remark 1 (Superposition over lights)**.** *The diffusion equation is linear in the incident source and b is linear in the incident field, so several lights superpose, and the learned components preserve this at the level of the operator: b is assembled per light, the solve is applied to their sum, and the visibility decay multiplies each light's own ballistic term. Two qualifications. The split into single-scattered (L_1) and diffused (L_{ms}) radiance is not strictly disjoint, because the source we assemble is not albedo-reduced (Section 2.4); this does not affect superposition, since both terms are built per light. And the material coefficients are conditioned on direction (Section 2.3), so superposition is exact for the transport given the medium, not for the medium itself.*

Remark 2 (Degeneration as $\sigma_s \to 0$)**.** *As $\sigma_s \to 0$ the source $b = \sigma_s EO$ of Eq. 3 vanishes linearly, so $\phi \to 0$ and $L_{\mathrm{ms}} \to 0$, while the operator $F = D + M + S$ stays symmetric positive definite throughout: the channel switches itself off continuously rather than the solve degenerating. On opaque materials the surface terms of Eq. 2 are therefore left unperturbed.*

Remark 3 (Exit coefficient)**.** *The exit factor $(1 + 1/A(\eta))/(4\pi)$ in Eq. 7 follows from substituting the Robin relation $\kappa\,\partial_n\phi = -\phi/(2A)$ into the first-order angular expansion $L(\boldsymbol{\omega}) = \frac{1}{4\pi}(\phi + 3\,\mathbf{J}\cdot\boldsymbol{\omega})$ with $\mathbf{J} = -\kappa\nabla\phi$, evaluated at the boundary for outgoing directions and averaged; it is the $\Gamma_s = 0$ specialization of the render query of Arbree et al. (2011). At $\eta = 1$, $A = 1$ and the factor is $1/(2\pi)$. The exit carries no further factor of σ_s: it already enters through the source, and ϕ is linear in it, so applying it again would make the exitant radiance quadratic in a chromatic coefficient.*

Algorithm 1 One transport evaluation in VolS-GS. Steps 1–2 run when densification changed the Gaussian count and in any case every 300 iterations.

Require: Gaussians $\{(\boldsymbol{\mu}_i, \Sigma_i, \alpha_i, \mathbf{z}_i)\}$, light position $\mathbf{p}_\ell$, camera
Ensure: per-Gaussian subsurface radiance L_i^{sss}

1: $\Omega \leftarrow \{\text{cells whose splatted occupancy exceeds the threshold}\}$ ▷ 64^3 grid over the scene bounds
2: splat $(\boldsymbol{\kappa}, \boldsymbol{\sigma}_a, \boldsymbol{\sigma}_s)$ to cell centers; fill interior cells from the nearest surface cell
3: **for all** $c \in \Omega$ **do** ▷ source, Eq. 3
4: $\quad \tau_c \leftarrow$ occupancy integrated over 32 steps from c toward $\mathbf{p}_\ell$
5: $\quad b_c \leftarrow \sigma_{s,c}\, O_c \exp(-\sigma_{t,c}\tau_c) \,/\, d_c^2$
6: **end for**
7: assemble $F = D + M + S$ ▷ face conductances, absorption, Robin boundary
8: $\phi \leftarrow \mathrm{CG}(F, b)$ ▷ warm-started from the transfer basis at inference
9: $\phi_i \leftarrow$ trilinear sample of ϕ at $\boldsymbol{\mu}_i$ ▷ pull back to primitives
10: $L_i^{\text{sss}} \leftarrow g_{\text{cal}} \left(\mathrm{exit}(\phi_i) + \mathrm{singleScatter}(\boldsymbol{\mu}_i, \mathbf{p}_\ell) \right)$

backward: solve $F\lambda = \partial\mathcal{L}/\partial\phi$ with the same operator; λ gives the gradients of b and of the diagonal of $M + S$. Face conductances are stop-gradient, and the solver iterations are never unrolled.

C Implementation details

Per-frame cost. Per frame the transport adds a source march, one finite-volume solve and the 32-sample single-scatter quadrature. The operator has one row per active grid cell, between 5.1×10^4 and 1.4×10^5 cells across the 21 scenes (median 7.4×10^4 of the 64^3 total). At inference a precomputed transfer basis (Sloan et al., 2002) over a Fibonacci light shell warm-starts the solve, so eight preconditioned conjugate-gradient iterations suffice. Measured end to end on four scenes, test PSNR differs from the converged solve by -0.009 to $+0.000$ dB, and every number this paper reports is produced with that setting. The fixed-topology parts of the pipeline are replayed as CUDA graphs. Because the grid is 64^3 for every scene, frame time follows the primitive count rather than the pixel count, so a sixteen-fold increase in pixels at a matched primitive count (Bunny and CupFabric, both near 82k) adds only a few milliseconds while the densest scene (Drums, 550k) costs 47.07 ms. Training runs on a single GPU with the same schedule as the baselines, with the calibration gain trained for the first 20k iterations and then frozen.

Table 3: **Render cost.** A single run on one A6000, reported per dataset and per resolution.

dataset	res.	scenes	FPS
SSS-GS synthetic	256^2	5	64.5 ± 4.9
SSS-GS real	512^2	3	66.0 ± 10.2
NRHints real	512^2	4	46.0 ± 2.5
	1024^2	3	54.8 ± 11.7
GS^3 synthetic	512^2	6	32.9 ± 11.3

	Bunny (82k)		Statue (108k)		Drums (550k)	
stage	ms	%	ms	%	ms	%
visibility rasterization	3.48	23.5	3.64	22.4	8.18	17.4
shading, pre-solve	0.96	6.5	1.25	7.7	7.20	15.3
transport solve	8.19	55.3	8.92	54.7	23.81	50.6
composition, post-solve	0.38	2.5	0.49	3.0	2.41	5.1
primary rasterization	1.80	12.2	1.98	12.2	5.46	11.6
frame	14.81	100	16.29	100	47.07	100

Transport implementation. Algorithm 1 states the steps. The grid is rebuilt whenever densification changes the Gaussian count and in any case every 300 iterations, and the interface-area estimate entering the Robin term is box-smoothed over 3^3 cells and renormalised over the active cells, the standard treatment for a surface term on a staircase grid (Brackbill et al., 1992). Without it the solve

is not grid convergent: ϕ at 64^3 and at 96^3 agrees to a correlation of 0.08, and to 0.97 with it, while the source assembled from the same fields already agrees to 0.96 at both resolutions, which places the discrepancy in the operator rather than in the splatting or the march. Training runs 20 conjugate-gradient iterations preconditioned by a three-level geometric V-cycle (Stam, 1995) to a tolerance of 10^{-3}, and both the forward and the adjoint solve are truncated, so the gradient is the implicit-function estimator evaluated at an approximate solution. The source attenuation is factorised as $\exp(-\sigma_t(c)\,\tau(c))$ rather than integrated along the ray. The march accumulates occupancy only, so τ is a geometric quantity that survives until the grid is rebuilt, while σ_t is refreshed at every step; a path-integrated extinction would have to be remarched for every cell and light at every iteration. The factorisation is exact when the medium is uniform along the path and approximate otherwise.

Material head and shading. The material head is a small MLP over the positional encoding, the per-Gaussian latent, and the frequency-encoded light and view directions; outputs map to $[\sigma_{\min}, \sigma_{\max}] = [10^{-3}, 25]$ per channel through a scaled sigmoid. We set the exit index to $\eta = 1.0$ on every scene, including the real captures, so the Fresnel factors reduce to identity and the residual boundary scale is absorbed by the calibration above. The specular lobe is a mixture of 8 anisotropic spherical Gaussians with scene-global lobe frames and widths and per-Gaussian amplitudes and specular color, evaluated in the Gaussian's local frame with a Schlick Fresnel factor (Schlick, 1994). The decay MLP of Eq. 11 has three hidden layers of 32 units.

Training. Losses, schedules, densification, and optimizer settings are inherited unchanged from the base relighting system, plus the scale-consistency penalty (weight 0.05) that discourages single splats from spanning many transport cells. A single configuration is used for all scenes. Retrained baselines follow the same per-dataset schedule. Every number in the paper, ours and every baseline's, is read at the protocol iteration for its dataset (60k or 100k), which each table states.

D Additional qualitative results

D.1 Per-scene results

Tables 4 and 5 give the per-scene numbers behind the dataset means of Table 1; the SSS-GS family is already per scene in Table 1. Figure 7 shows five further test frames beside the tables, chosen to cover all three dataset families rather than to favour any one of them.

Table 4: **Real captures, per scene** (NRHints). The 3DGS and GI-GS rows are quoted from Zheng et al. (2026). Every other row is our own retraining to the 100k protocol iteration. Best red, second orange among retrained rows.

	Method	CAT	CATSMALL	CUPFABRIC	FISH	FURSCENE	PIKACHU	PIXIU
PSNR↑	3DGS	14.5326	22.5727	25.0488	22.8411	18.4838	19.6310	18.5501
	GI-GS	13.9988	22.3222	24.3821	22.7500	17.8520	19.1867	18.1064
	GS^3	27.7597	34.6288	36.3707	31.1387	30.5467	32.6480	30.4776
	SSS-GS	16.6533	31.4044	32.9904	22.8345	20.4048	28.4206	23.4290
	RNG	26.7680	34.2488	37.1390	29.2007	27.8608	31.1881	28.7688
	SSD-GS	27.9527	34.6987	36.9159	31.0531	30.6964	32.1040	30.9603
	VolS-GS (Ours)	28.1084	35.0647	37.0219	31.1809	30.8567	32.9575	31.3762
SSIM↑	3DGS	0.6962	0.8896	0.9430	0.8312	0.7869	0.9000	0.8298
	GI-GS	0.3162	0.8750	0.9178	0.7437	0.5811	0.8724	0.6117
	GS^3	0.9059	0.9766	0.9818	0.9254	0.9503	0.9678	0.9424
	SSS-GS	0.6475	0.9464	0.9625	0.8065	0.7868	0.9477	0.8516
	RNG	0.8464	0.9677	0.9827	0.8933	0.9163	0.9657	0.9180
	SSD-GS	0.9069	0.9760	0.9825	0.9246	0.9511	0.9668	0.9446
	VolS-GS (Ours)	0.9084	0.9772	0.9823	0.9254	0.9522	0.9689	0.9447
LPIPS↓	3DGS	0.3033	0.1183	0.0868	0.1814	0.2101	0.1103	0.1721
	GI-GS	0.3518	0.1339	0.1116	0.2063	0.2492	0.1347	0.3089
	GS^3	0.1314	0.0650	0.0525	0.0872	0.0707	0.0673	0.0821
	SSS-GS	0.3400	0.0881	0.0688	0.1809	0.2167	0.0744	0.1434
	RNG	0.1992	0.0632	0.0371	0.1292	0.1099	0.0521	0.1082
	SSD-GS	0.1313	0.0679	0.0519	0.0879	0.0737	0.0680	0.0807
	VolS-GS (Ours)	0.1297	0.0630	0.0516	0.0858	0.0701	0.0658	0.0790

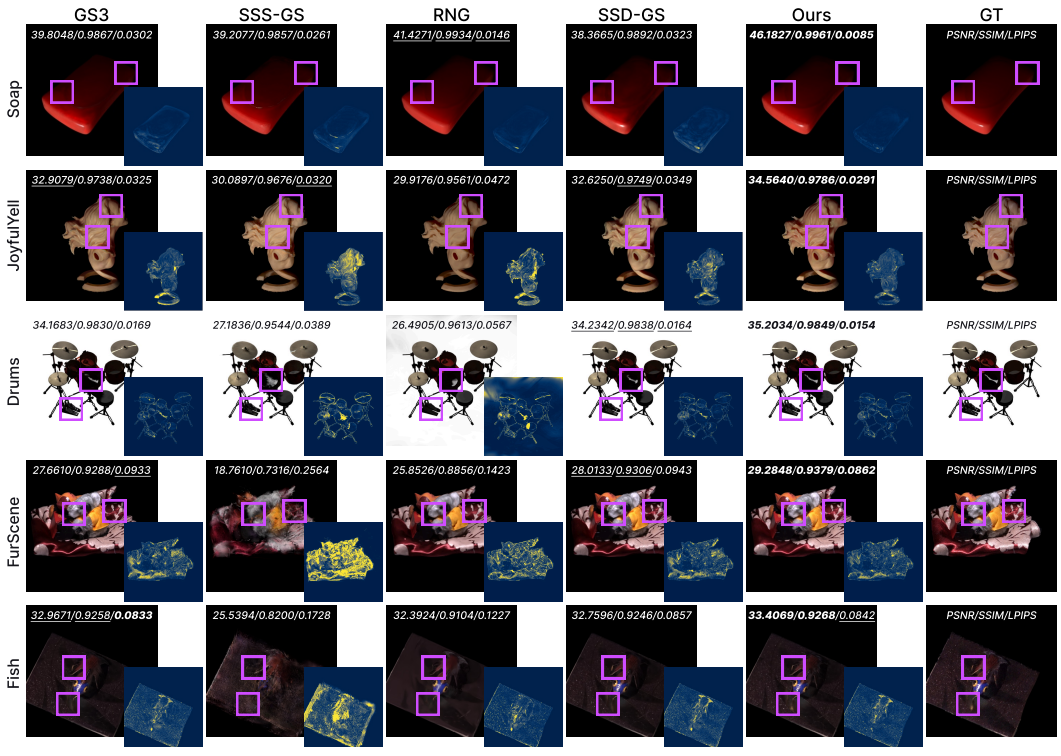


Figure 7: **Additional qualitative comparisons.** Five further test frames, one per row: SOAP and JOYFULYELL from the subsurface set, DRUMS from the GS^3 synthetic set, FURSCENE and FISH real captures. Per-frame PSNR/SSIM/LPIPS are printed on each panel, **best** bold and second underlined; the inset is |render − GT| clipped at 0.15 on one shared ramp.

Table 5: **GS^3 synthetic scenes, per scene** trained at 100K iterations. Notation as in Table 4.

	Method	TRANSLUCENT	ANISOMETAL	DRUMS	FURBALL	HOTDOG	LEGO
PSNR↑	3DGS	16.4899	17.1009	24.5093	20.1206	16.9535	15.9886
	GI-GS	16.0766	15.9567	24.6177	19.5295	16.8118	16.4229
	GS^3	29.7079	28.5958	32.9349	34.0920	32.2632	29.0746
	SSS-GS	28.2425	24.2328	28.4225	31.0156	30.8217	26.6436
	RNG	27.1813	26.1441	29.7962	33.0337	29.8325	25.8664
	SSD-GS	29.9792	28.2052	33.0411	34.1532	31.6721	28.9808
	VolS-GS (Ours)	31.6353	28.6080	33.2051	34.2224	31.7383	30.1361
SSIM↑	3DGS	0.8958	0.8849	0.9439	0.8951	0.8599	0.7904
	GI-GS	0.8586	0.8304	0.8941	0.8592	0.8282	0.7647
	GS^3	0.9710	0.9627	0.9838	0.9683	0.9724	0.9474
	SSS-GS	0.9562	0.9249	0.9619	0.9534	0.9616	0.9134
	RNG	0.9659	0.9509	0.9719	0.9587	0.9622	0.9267
	SSD-GS	0.9724	0.9606	0.9838	0.9703	0.9711	0.9447
	VolS-GS (Ours)	0.9783	0.9630	0.9839	0.9702	0.9719	0.9539
LPIPS↓	3DGS	0.0748	0.0704	0.0442	0.0865	0.1128	0.1416
	GI-GS	0.1155	0.1179	0.1242	0.1694	0.1638	0.1643
	GS^3	0.0319	0.0348	0.0160	0.0529	0.0340	0.0490
	SSS-GS	0.0521	0.0677	0.0421	0.0938	0.0472	0.0862
	RNG	0.0336	0.0412	0.0273	0.0730	0.0423	0.0618
	SSD-GS	0.0304	0.0363	0.0165	0.0466	0.0360	0.0515
	VolS-GS (Ours)	0.0246	0.0345	0.0155	0.0463	0.0335	0.0438

D.2 Relighting under a light sweep

A fixed test split cannot show whether appearance stays coherent as the light moves, only whether it is right at the frames that were held out. Figure 8 sweeps the light through a full revolution at the elevation the training lights actually occupy, so every sample sits inside the training distribution in elevation and differs only in azimuth. The cast shadow and the subsurface glow remain spatially consistent as the light moves, an additional qualitative check on the learned decomposition.

D.3 Environment map relighting

Our setting is one light at a time at a known position, and the source term of Eq. 3 is written for exactly that. Relighting under an environment map is a different problem: there the illumination is unknown and has to be estimated alongside the material. We do not evaluate on those benchmarks and we do not compare against methods built for them.

The decomposition itself is not tied to a single light, however. An environment map is a sum of directional sources, and the solve is linear in the source, so relighting under one is the same computation repeated: we importance-sample $K = 256$ directions from the map, place a distant light along each, and add the per-light renders with the map's own weights. Figure 9 shows the result on four maps, with the map rotated about the vertical axis between the two frames of each pair. Nothing is retrained.

E Ablations on the remaining scenes

Table 6: **Ablations on further scenes** (test PSNR↑). The same five rows as Table 2, whose Statue and Bunny PSNR columns are repeated here, extended to five further scenes. Every arm is read at the protocol iteration for its family, 60k on the SSS-GS scenes and 100k elsewhere.

	Statue	Bunny	JoyfulYell	Translucent	Drums	CupFabric	Pikachu
Full model	38.35	40.15	32.19	31.64	33.21	37.02	32.96
− Volumetric SSS	36.19	34.67	31.54	30.88	33.11	35.78	31.43
− Transport cues on shadow	35.09	39.17	32.08	31.01	32.74	36.41	32.19
Shadow term multiplied onto SSS	37.66	39.02	31.87	31.15	32.93	36.82	32.46
− Shadow term	34.41	36.30	29.24	28.42	32.47	34.07	28.81

Table 2 reports the single-change ablations on Bunny and Statue with all three metrics; Table 6 repeats their PSNR columns beside five further scenes, so the ordering can be read across every scene at once. Removing the shadow term is the largest single change on six of the seven scenes. Bunny is the exception and the clearest case for the channel: removing the subsurface term costs 5.48 dB there, the largest effect of any arm on any scene. What comes second follows the medium: the subsurface channel on JoyfulYell, Translucent, CupFabric and Pikachu, the transport cues on Statue and Drums. On Drums, whose translucent component is weak, removing the subsurface channel costs 0.10 dB, the least it costs on any scene, consistent with the degeneration of Remark 2. The one design choice that does not move the rendered image, the specular leakage regularizer, is in Appendix F.

F The specular leakage regularizer

Figure 10(a) shows the difference. This term does not earn PSNR and we do not present it as if it did. Switching it on moves test PSNR by between -0.30 and $+0.14$ dB across the seven scenes where both arms exist, a mean of -0.08 dB with a standard deviation of 0.14, and the sign changes from scene to scene.

The term earns its place by keeping the decomposition identifiable. Nothing in a photometric loss distinguishes a warm glow that the specular lobe paints onto a shadowed pixel from the same glow carried there by the medium, so the free lobe will absorb transported appearance whenever that is the cheaper fit. The regularizer penalises specular energy in pixels the light does not reach, the configuration in which a surface lobe cannot account for the radiance. On Bunny it collapses the

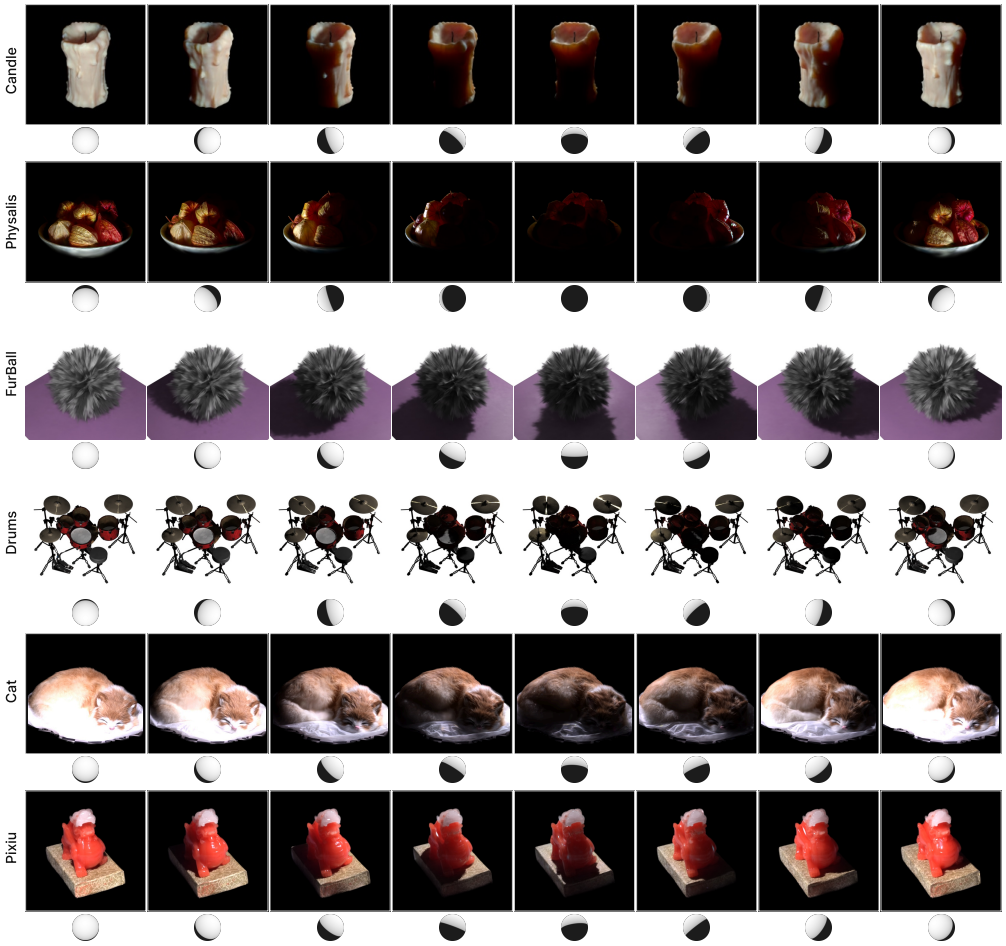


Figure 8: **Relighting under a continuous light sweep.** Eight unseen light positions per scene, one per column, with the disc below each column giving its light direction.

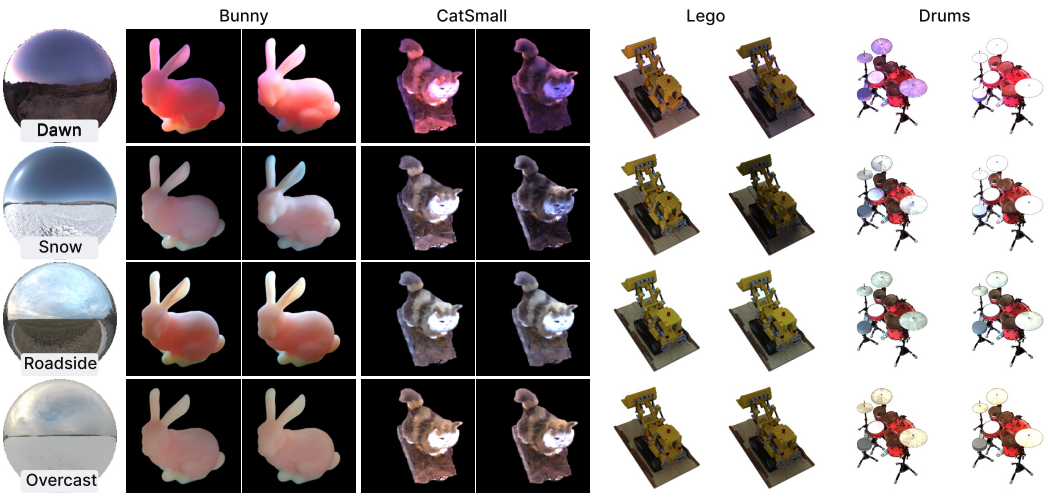
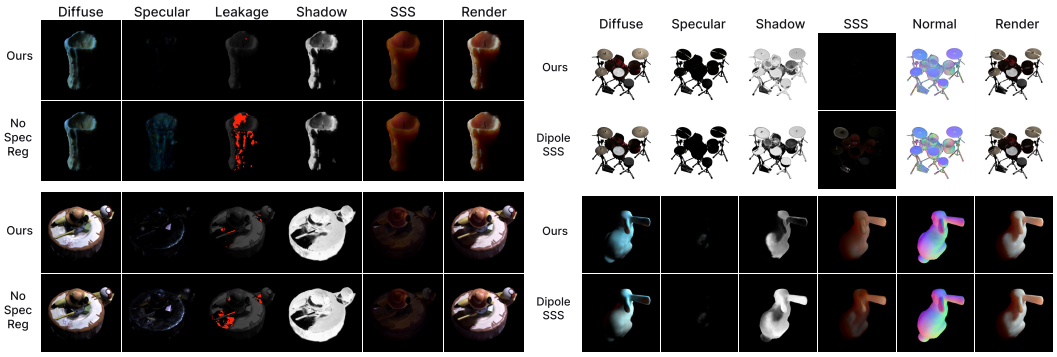


Figure 9: **Relighting under an environment map.** One row per map, shown as a mirror ball at the left; two frames of a full rotation of the map for each of four scenes. Nothing is retrained.



(a) Leakage removed by the regularizer

(b) Replacing the solve with the per-primitive kernel

Figure 10: **Two reads of the subsurface slot. (a)** CANDLE and PIKACHU with the specular leakage regularizer on (Ours) and off, same frame and channels. Without it the specular lobe paints a ghost of the object into shadowed regions and the diffuse term washes out. The *Leakage* column marks in red the pixels where the model's own decay says the light is blocked ($D_i < 0.3$) while the specular lobe still fires above 0.05. **(b)** DRUMS and BUNNY with our solve (Ours) and with SSD-GS's per-primitive dipole in the same slot, everything else held fixed. DRUMS shows the channel taking up surface residual, BUNNY the transported light misplaced. The SSS column of (b) is shown on its own black ground and at a $2.5\times$ gain so that both rows are legible.

specular footprint from 68% to 23% of the object at unchanged PSNR, the same image attributed differently.

Switching the regularizer off changes the full model by 0.18 dB, within the same range as the per-scene values above.

G REPLACING THE SOLVE WITH A PER-PRIMITIVE KERNEL

A diffusion profile is a kernel, not an outgoing radiance. The BSSRDF it belongs to integrates the profile against the light arriving over the surface, $L_o(\mathbf{x}_o) = \int_A \int_\Omega S(\mathbf{x}_i, \boldsymbol{\omega}_i, \mathbf{x}_o, \boldsymbol{\omega}_o)\, L_i \,\mathrm{d}\boldsymbol{\omega}\,\mathrm{d}A$, so evaluating it requires knowing where light entered. The per-primitive kernel of SSD-GS does not carry that integral. The profile is evaluated once, at a radius its network predicts, and scaled by the summed inverse-square falloff of the lights. Our own coefficients are learned as well. The difference is that the dipole spreads the profile along Euclidean distance from that radius, a route the medium need not occupy.

Thus we further investigate the effect of volumetric subsurface scattering against the dipole. Only the subsurface slot changes, with the transport cues, the specular regularizer, the exit term and the per-primitive amplitude held as in the full model. Table 7 reports the substitution, which costs accuracy on six of the seven scenes, by more than 2 dB on CUPFABRIC and PIKACHU. Figure 10(b) shows where the difference sits. On BUNNY our channel is the smoother of the two and stays in the region the decay leaves unlit, while the dipole spreads a comparable total over the whole body. DRUMS is the one scene the substitution wins on PSNR and SSIM. On an opaque object the dipole carries a channel-to-render ratio of 0.090 against our 0.001, so the slot is taking up residual that belongs to the surface terms. Our channel is not immune to that.

H DECOMPOSITION ACROSS METHODS

Figure 11 reads the same decomposition across the two baselines that also carry an explicit subsurface term, on PIXIU (left block) and STATUE (right block). On STATUE, SSD-GS's subsurface channel is empty (channel-to-render ratio 0.00 against our 0.52) while its rendered image still contains the backlit glow, so the glow is not explained by its explicit subsurface component. SSS-GS's physical channels are nearly dark on both scenes and its residual network carries more energy than its own final render (ratio 1.5 on STATUE), indicating substantial energy that is not exposed as the

Table 7: **The dipole kernel in place of the transport solve.** Only the subsurface slot changes: SSD-GS's own scattering module and its $\sum_\ell 1/d_\ell^2$ carrier replace our solve, while the transport cues on the shadow branch, the specular regularizer, the exit term and the per-primitive amplitude are left as in the full model.

	VolS-GS			Dipole SSS		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
BUNNY	**40.15**	**0.9920**	**0.0112**	39.52	0.9911	0.0119
STATUE	**38.35**	**0.9899**	**0.0130**	36.53	0.9862	0.0165
JOYFULYELL	**32.19**	**0.9690**	**0.0385**	31.68	0.9674	0.0408
TRANSLUCENT	**31.64**	**0.9783**	**0.0246**	31.13	0.9778	0.0251
CUPFABRIC	**37.02**	**0.9823**	**0.0516**	34.61	0.9744	0.0550
PIKACHU	**32.96**	**0.9689**	**0.0658**	30.73	0.9592	0.0694
DRUMS	33.21	0.9839	**0.0155**	**33.24**	**0.9844**	0.0158

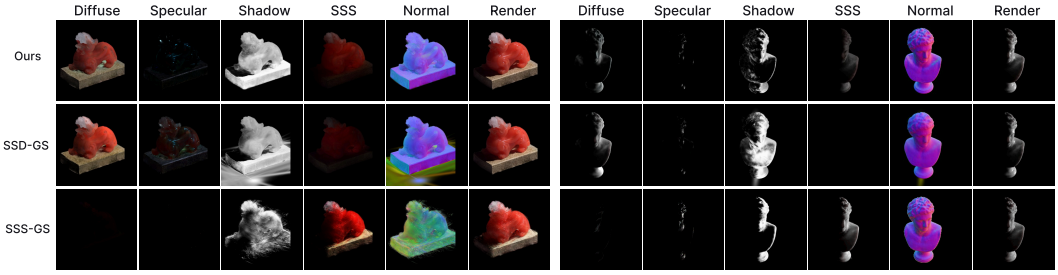


Figure 11: **Decomposition across methods.** PIXIU (left) and STATUE (right), each row one method's own decomposition on the same frame. For SSS-GS the SSS column shows its residual network's output, its only channel with appreciable energy.

method's explicit subsurface component; its SSS column in the figure therefore shows that residual. Our channel carries the backlit radiance itself, and the render matches the reference.

SSD-GS's empty channel on STATUE is the same failure that Figure 10(b) isolates. The module whose subsurface channel reads 0.00 here is the per-primitive dipole we substituted into our own slot there, and in that controlled setting the kernel either mis-places the transported light or fires where there is no medium to transport through. A kernel with no reading of the medium has no stable quantity to put in the channel, so the glow appears in the render but not in the decomposition.

I FAILURE CASES

The thinnest geometry fails first. On BUNNY the ear tip seen edge-on renders slightly dark, with less transmitted light than the capture (Figure 12). The single-scattering term of Eq. 8 exists for exactly this regime, where a lobe only a few mean free paths across is too thin for the diffusion approximation to be accurate, and it narrows the gap without closing it. Refining the transport grid from 64^3 to 96^3 moves the region only by 0.01 dB.

The second failure is on an opaque object. On HOTDOG the plate under the bun renders darker than the capture. The appearance there is a broad and nearly uniform highlight-like glow, which our decomposition does not target.

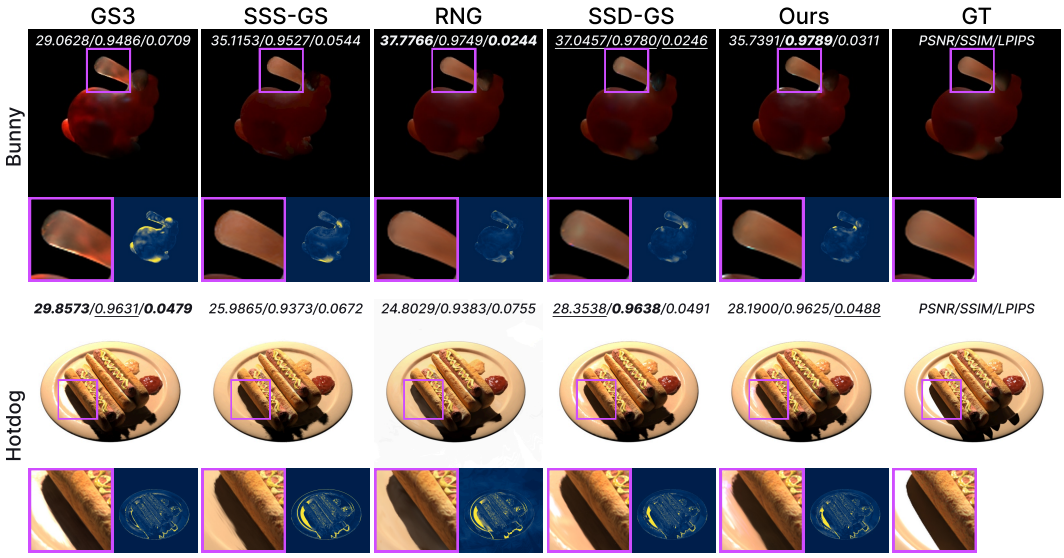


Figure 12: **Failure cases.** BUNNY with the ear folded back and seen edge-on, where the tip renders darker than the capture, and HOTDOG, where the plate under the bun renders darker. Under each render the magnified square marks the failing region at the same image coordinates in every column, beside the error map $|\text{render} - \text{GT}|$ averaged over color channels and clipped at 0.15 on one shared ramp. Per-frame PSNR/SSIM/LPIPS are printed on each panel, **best** in bold and second underlined within a row.