GLINT: Modeling Scene-Scale Transparency via Gaussian Radiance Transport

Youngju Na[1,2,*] Jaeseong Yun[2] Soohyun Ryu[2] Hyunsu Kim[2] Sung-Eui Yoon[1] Suyong Yeon[2]

[1]KAIST [2]NAVER LABS

https://youngju-na.github.io/GLINT

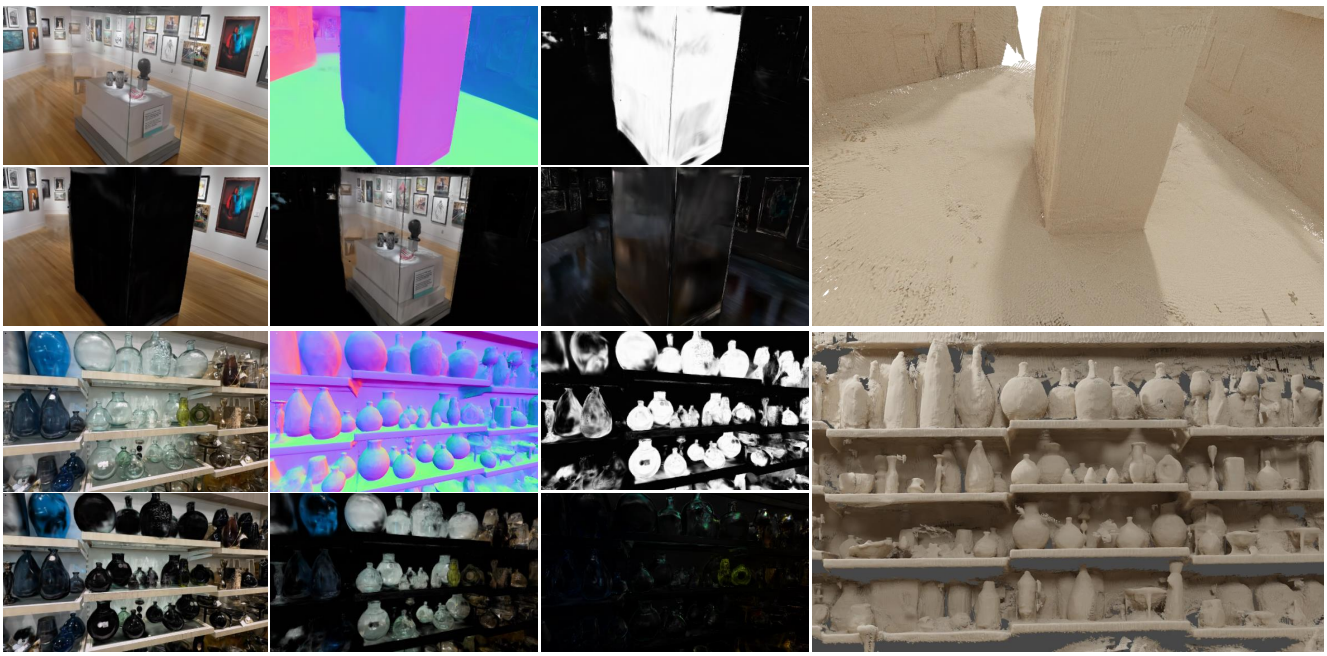


Figure 1. Our framework GLINT performs decomposed Gaussian radiance transport to reconstruct transparent surfaces with physically consistent geometry and appearance. (Left) The first row shows a rendered image, normal map, and transparency map. The second row visualizes the radiance contributions of the interface, transmission, and reflection components. (Right) Reconstructed Mesh.

Abstract

While 3D Gaussian splatting has emerged as a powerful paradigm, it fundamentally fails to model transparency such as glass panels. The core challenge lies in decoupling the intertwined radiance contributions from transparent interfaces and the transmitted geometry observed through the glass. We present GLINT, a framework that models scene-scale transparency through explicit decomposed Gaussian representation. GLINT reconstructs the primary interface and models reflected and transmitted radiance separately, enabling consistent radiance transport. During optimization, GLINT bootstraps transparency localization from geometry-separation cues induced by the decomposition, together with geometry and material priors from a pre-trained video relighting model. Extensive experiments demonstrate consistent improvements over prior methods for reconstructing complex transparent scenes.

1. Introduction

Photorealistic 3D reconstruction with accurate geometry is a central goal in vision and graphics. 3D Gaussian Splatting (3DGS) [19] has recently advanced this goal with real-time rendering and high visual fidelity. The core challenge in modeling transparent surfaces with 3DGS arises from its monolithic α-blending formulation, which inherently conflates geometry and appearance across multiple radiance paths. Consequently, 3DGS bakes these optical phenomena into entangled Gaussian primitives, achieving visual plau-

*Work done during an internship at NAVER LABS.

sibility while compromising geometric accuracy and interpretability. The problem is especially pronounced in large-scale scenes containing thin transparent structures, such as architectural glass, display cases, or windows, as in Fig. 1. In these scenarios, a single pixel captures a superposition of radiance from reflected and transmitted components originating from distinct physical locations [36]. To render photorealistic novel view images, the Gaussians associated with glass must have negligibly low opacity or be pruned during optimization to reveal background objects. Conversely, physically grounded glass geometry demands high opacity with sharp boundaries. This ambiguity forces standard α-blending into a compromise, often producing ghost-like reflections or missing transparent geometry.

Existing approaches often rely on object-centric assumptions and segmentation masks [21, 24], or are restricted to primary first-surface reconstruction, struggling to recover transmissive radiance contributions. Such limitations prevent a physically faithful reconstruction, which is crucial for downstream applications including robotic manipulation, digital-twin construction, and scene understanding.

In this paper, we present **GLINT**, *a Gaussian Light INverse-rendering framework for scene-scale Transparency reconstruction.* Specifically, we model a scene with a decomposed Gaussian representation that explicitly partitions primitives into interface, transmission, and reflection components, enabling a formulation for both transparent and opaque regions. Our hybrid Gaussian radiance transport unifies rasterization and ray tracing to reconstruct multi-path radiance in a physically grounded manner. In addition, we incorporate geometric and material priors from a pre-trained video diffusion relighting model [26], which complements our decomposition and provides optimization stability. As illustrated in Fig. 1, our framework successfully reconstructs complex transparent scenes, producing accurate geometry, reliable transparency maps, and an interpretable decomposition of radiance. Our key contributions are as follows:

- We propose a decomposed Gaussian representation that enables modeling of both transparent and opaque regions through explicit separation of first-visible interface, transmission, and reflection components.
- We introduce a hybrid transparency-aware rendering scheme that unifies rasterization and ray tracing under a physically grounded radiance formulation for consistent multi-path radiance transport.
- We introduce 3D-FRONT-T, a first synthetic benchmark for scene-scale transparency that enables quantitative evaluation of both appearance and geometry. Using this dataset and the real-world DL3DV-10K [27] dataset, we achieve state-of-the-art reconstruction performance in both photometric and geometric evaluation.

2. Related Work

Our study lies at the intersection of neural scene representations and physically based inverse rendering, which aims to reconstruct accurate geometry and appearance with scene-scale transparency. We review relevant research including: advances in 3DGS, radiance decomposition for non-Lambertian surfaces, and transparent scene reconstruction.

2.1. Gaussian Splatting

3DGS [19] represents scenes using explicit 3D Gaussian primitives, achieving real-time novel view synthesis with high visual fidelity. Since its introduction, numerous extensions have sought to improve its geometric accuracy and photometric realism. Several works reduce reliance on strong SfM initialization [9, 20], introduce alias-free splatting [44], or enhance geometric fidelity [5, 6, 12, 15]. For instance, adapting Gaussians into planar-constrained structures [5, 12] or surfels [6, 15] enables a more faithful representation of fine-grained surface details.

Accurate surface modeling is crucial not only for geometry reconstruction but for capturing secondary lighting effects, such as reflection and refraction, which are sensitive to surface geometry. However, existing approaches commonly rely on spherical harmonics to approximate view-dependent radiance, which limits their ability to capture complex non-Lambertian effects in an interpretable manner.

2.2. Radiance Decomposition in Gaussian Splatting

To move beyond simple view-dependent appearance and model complex light–material interactions, several works decompose radiance into more interpretable components.

One line of research integrates physically-based material properties directly into the Gaussian primitives. For instance, GaussianShader [17] and R3DG [10] incorporate BRDF shading functions for each Gaussian, enabling per-primitive separation into diffuse and specular components.

A different strategy focuses on explicitly modeling the source of reflections rather than just the surface properties. DeferredGS [37] and 3DGS-DR [42] employ deferred rendering pipelines, which first buffer geometric properties and then compute view-dependent shading and reflections in a second pass. Taking this concept further, EnvGS [39] introduces a separate set of environment Gaussians to explicitly model the surrounding scene, then uses a ray tracing pipeline to query this environment representation to render strong reflections.

While these approaches greatly improve the realism and interpretability, their formulations remain restricted to opaque reflected light paths and provide limited capacity for modeling transmitted radiance. Consequently, they struggle with transparent surfaces that exhibit both reflection and transmission, a common case in real-world settings such as glass facades, display cases, and windows.

2.3. Transparent Scene Reconstruction

Transparency poses a fundamental challenge for multi-view reconstruction because the observed radiance is a mixture of transmitted background and glass-reflected environment components, violating the single-surface visibility assumption in conventional methods [38]. For optically thin glass, light transport remains nearly linear with negligible refraction [11], yet radiance contributions from multiple depths overlap within each pixel, resulting in perceptually complex observations. This contrasts with refractive transparency, where pronounced refraction at material interfaces produces explicit optical phenomena that require volumetric modeling of the refractive medium [2, 25, 33].

Under the Gaussian splatting paradigm, TransparentGS [16] introduces transparent Gaussian primitives and light-field probes to efficiently handle object-centered refraction, while TSGS [24] targets optically thin transparency using first-surface rasterization for accurate geometry reconstruction. Despite these advances, both approaches remain object-centric and require segmentation or assume a single surface, making them difficult to scale to complex multi-depth transparency in real-world scenes.

In this paper, we address these limitations by modeling scene-scale optically thin transparency through a decomposed Gaussian representation and a physically consistent radiance transport formulation that jointly reconstructs geometry and appearance without requiring masks.

3. Method

3.1. Preliminaries: 2D Gaussian Rendering

Our method builds upon Gaussian splatting, specifically adopting 2D Gaussian Splatting (2DGS) [15], which represents scenes using anisotropic 2D Gaussians defined on local tangent planes. This resolves geometric inaccuracies in 3DGS [19], which relies on volumetric 3D primitives.

While 3DGS defines primitives with a 3D covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$, 2DGS constrains each Gaussian's covariance to lie on a local tangent plane. Each 2D Gaussian primitive G_i is parameterized by its 3D mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, a 2D covariance $\mathbf{\Sigma}_i^{2D}$ defined on the tangent plane [15], Spherical Harmonics (SH) coefficients $\mathbf{c}_i$ for view-dependent color, and an opacity o_i.

The scene is rendered using standard alpha-compositing. Unlike 3DGS, which evaluates volumetric density through a projected 2D footprint, 2DGS performs perspective-correct splatting. For each pixel $\mathbf{p}$, the contribution of a primitive is computed by explicitly intersecting the camera ray with the primitive's tangent plane, yielding a local coordinate $\mathbf{u}_i(\mathbf{p}) = (u, v)$. The 2D Gaussian is evaluated as:

$$G_i(\mathbf{u}_i) = \exp\left[-\tfrac{1}{2}(u^2 + v^2)\right]. \tag{1}$$

The opacity $\alpha_i(\mathbf{p})$ at which primitive i contributes to pixel $\mathbf{p}$ is given by:

$$\alpha_i(\mathbf{p}) = o_i \cdot G_i(\mathbf{u}_i(\mathbf{p})). \tag{2}$$

With these opacities, the contribution of each primitive along the viewing ray is accumulated using front-to-back compositing. We first define the transmittance T_i, which measures how much light from primitive i remains unoccluded by all closer primitives $j < i$:

$$T_i = \prod_{j<i}(1 - \alpha_j(\mathbf{p})). \tag{3}$$

Finally, the pixel radiance is obtained by summing the colors of all primitives intersecting the ray:

$$\mathbf{L}(\mathbf{o}, \mathbf{d}) = \sum_{i\in\mathcal{S}(\mathbf{r})} T_i\, \alpha_i(\mathbf{p})\, \mathbf{c}_i(\mathbf{d}). \tag{4}$$

This perspective-correct formulation preserves the correct geometric ordering of splats and enables stable, view-consistent rendering.

Rendering techniques. In practice, the per-pixel contribution $\alpha_i(\mathbf{p})$ (Eq. (2)) can be computed in two ways.
Rasterization. The 2DGS [15] framework employs a tile-based rasterizer that identifies affected pixels, computes the perspective-correct ray–splat intersection $\mathbf{u}_i(\mathbf{p})$, evaluates the Gaussian (Eq. (1)), and composites results via Eq. (4). This intersection-based formulation maintains geometric consistency and avoids projection artifacts inherent to the affine projection approximation in [19].
Ray Tracing. Alternatively, Gaussian ray tracing [28, 39] is employed for secondary effects, as rasterization is impractical for querying unique paths per pixel. Following EnvGS [39], we represent each 2D Gaussian as a pair of triangles to enable hardware-accelerated BVH traversal, allowing efficient ray-primitive intersection tests.

3.2. Decomposed Gaussian Representation

Modeling transparency within standard Gaussian volume rendering is fundamentally limited by its monolithic α-blending, which entangles geometry and appearance in a single compositing stream. This creates an inherent *transparency-depth dilemma* [24], where appearance optimization conflicts with geometric accuracy. When transparent surfaces contribute minimal radiance, optimization drives their opacity toward zero, causing them to vanish under adaptive pruning [19]. Conversely, increasing opacity treats them as opaque occluders, blocking transmitted backgrounds and collapsing geometry into a single entangled field.

To resolve this, we introduce a decomposed Gaussian representation that explicitly partitions the scene's Gaussian primitives into three functional sets based on their optical

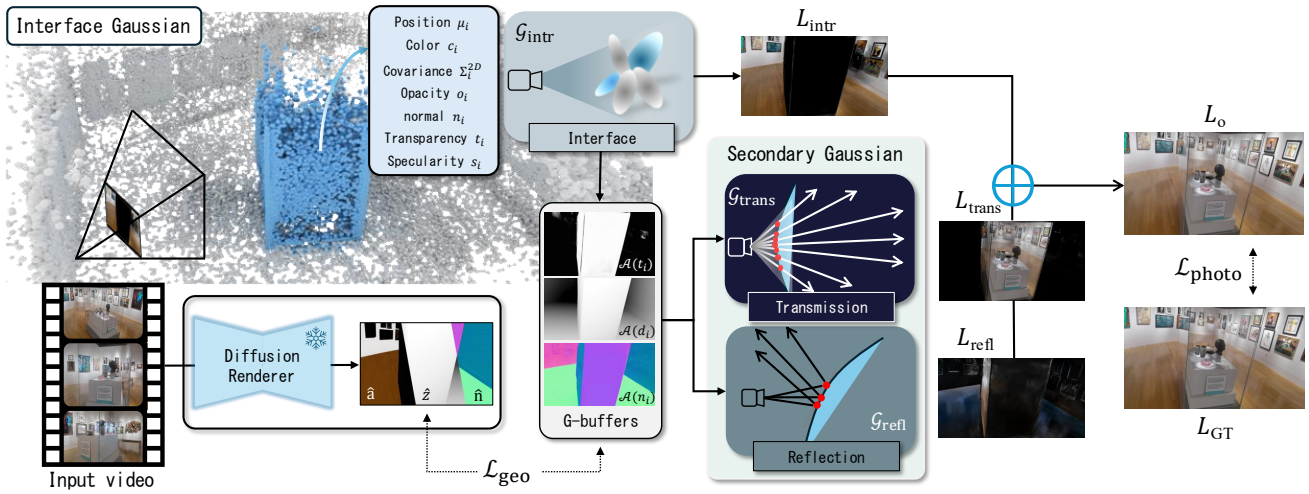


Figure 2. **Pipeline Overview.** The interface component $\mathcal{G}_{\text{intr}}$ captures primary first-surface, while secondary components $\mathcal{G}_{\text{trans}}$ and $\mathcal{G}_{\text{refl}}$ separately model transmission and reflection. The output color L_o is obtained through hybrid rendering under transparency-aware radiance transport and supervised by photometric loss $\mathcal{L}_{\text{photo}}$. DiffusionRenderer [26] provides priors that regularize the G-buffers via $\mathcal{L}_{\text{geo}}$.

roles: *Interface* ($\mathcal{G}_{\text{intr}}$), *Transmission* ($\mathcal{G}_{\text{trans}}$), and *Reflection* ($\mathcal{G}_{\text{refl}}$). The interface component $\mathcal{G}_{\text{intr}}$ captures the primary visible surface encountered by camera rays, encompassing both opaque and transparent material boundaries. It serves as the geometric interface, encoding per-point geometry and material attributes that govern how radiance is routed through subsequent components. The transmission component $\mathcal{G}_{\text{trans}}$ models background geometry visible through transparent surfaces, capturing transmitted radiance paths. Finally, the reflection component $\mathcal{G}_{\text{refl}}$ encodes environment radiance reflected at both opaque and transparent interfaces.

Hybrid rendering pipeline. With this decomposed representation, we employ a hybrid rendering approach that uses rasterization for primary surface visibility and ray tracing for secondary transmission and reflection paths. The interface component $\mathcal{G}_{\text{intr}}$ is rasterized to produce a G-buffer $\mathcal{B} = \{z, \mathbf{n}, t, s\}$ encoding depth, normal, transparency, and specularity, where each entry is obtained by applying the compositing operator $\mathcal{A}(\cdot)$ with weights $\mathcal{A}_i = T_i\alpha_i$ defined in Eq. 4. This G-buffer then guides ray-traced queries into $\mathcal{G}_{\text{trans}}$ and $\mathcal{G}_{\text{refl}}$ for secondary transmission and reflection, where the transparency $t \in [0, 1]$ gates the opaque–transparent split and the specularity $s \in [0, 1]$ controls the diffuse–specular balance. The overall rendering pipeline is illustrated in Fig. 2.

3.3. Transparency-Aware Radiance Transport

In this section, we formulate radiance transport through our decomposed Gaussian representation, where outgoing radiance is computed by querying and combining the radiance from interface, transmission, and reflection components based on local surface properties. Unlike prior radiance decomposition approaches [37, 39, 42, 46] which primarily address opaque reflection, our formulation adopts a BSDF-inspired decomposition [1] that splits outgoing radiance into reflection and transmission paths based on the surface properties encoded in the G-buffer.

The outgoing radiance, which we denote as L_o, is expressed as a transparency-gated interpolation between two transport branches:

$$L_o = (1 - t)\, L_{\text{opaque}} + t\, L_{\text{transparent}}, \tag{5}$$

where transparency t obtained from the G-buffer $\mathcal{B}$ determines whether radiance transport follows opaque or transparent paths.

Opaque branch (L_{opaque}). For opaque surfaces, the outgoing radiance consists of the interface base color combined with reflected environment radiance, weighted by surface specularity. The transport is modeled as a physically-inspired blend between diffuse and specular components. The Fresnel reflectance is approximated using the Schlick formulation [32] with the outgoing direction $\omega_o = -\mathbf{d}$:

$$F(\omega_o) = F_0 + (1 - F_0)(1 - \max(0,\, \omega_o \cdot \mathbf{n}))^5, \tag{6}$$

where F_0 denotes the normal-incidence reflectance. A learnable per-pixel specularity s modulates the overall specular weight, yielding a Fresnel-weighted blending factor:

$$k_s = s + (1 - s)F(\omega_o), \tag{7}$$

and the outgoing opaque radiance is given by:

$$L_{\text{opaque}} = (1 - k_s)\, L_{\text{intr}} + k_s\, L_{\text{refl}}, \tag{8}$$

where L_{intr} is the base color rasterized from the interface set $\mathcal{G}_{\text{intr}}$, and $L_{\text{refl}}(\mathbf{x}, \omega_r) = \text{Trace}(\mathcal{G}_{\text{refl}}, \mathbf{x}, \omega_r)$ denotes radiance traced from the reflection component $\mathcal{G}_{\text{refl}}$ along the

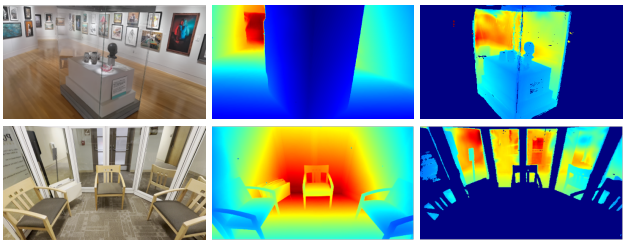


Figure 3. **Depth decomposition.** (Left) Rendered image, (Middle) interface depth, and (Right) transmission depth.

analytic reflection direction:

$$\omega_r = 2(\mathbf{n}\cdot\omega_o)\,\mathbf{n} - \omega_o. \tag{9}$$

This approximates the behavior of a diffuse–specular BRDF while omitting roughness and masking–shadowing terms for simplicity.

Transparent branch ($L_{\text{transparent}}$). For transparent surfaces, the outgoing radiance consists of transmitted radiance from the background and reflected radiance from the environment. We adopt a Fresnel-based decomposition inspired by the dielectric BSDF to balance these two contributions. The outgoing transparent radiance is formulated as:

$$L_{\text{transparent}} = (1 - k_s)\, L_{\text{trans}} + k_s\, L_{\text{refl}}, \tag{10}$$

where k_s is derived from Eq. 7, modulating the transmitted and reflected contributions. Under the optically thin assumption ($\omega_t \approx \omega_o$), refraction-induced bending is negligible, allowing the transmitted radiance to be approximated by $L_{\text{trans}} = \text{Trace}(\mathcal{G}_{\text{trans}}, \mathbf{x}, \omega_t)$.

By separating transmission and reflection into independently optimizable Gaussian components, this formulation mitigates the transparency–depth trade-off inherent in monolithic α-blending, which often leads to degraded visual fidelity or geometric accuracy.

3.4. Optimization

Our primary training objective minimizes the photometric error between the rendered outgoing radiance L_o (Eq. 5) and the ground-truth image L_{GT}:

$$\mathcal{L}_{\text{photo}} = \lambda_1\mathcal{L}_1 + \lambda_{\text{ssim}}\mathcal{L}_{\text{SSIM}} + \lambda_{\text{lpips}}\mathcal{L}_{\text{LPIPS}}, \tag{11}$$

where $\mathcal{L}_1$ measures pixel-wise reconstruction error, $\mathcal{L}_{\text{SSIM}}$ [35] enforces structural similarity, and $\mathcal{L}_{\text{LPIPS}}$ [45] captures perceptual fidelity.

While $\mathcal{L}_{\text{photo}}$ provides the main supervision signal, transparent scenes remain ill-posed because each pixel mixes reflected and transmitted radiance from different depths, making photometric cues alone insufficient. Prior works use dense monocular predictors [31, 40, 41] as auxiliary geometric regularizers [23, 24, 39]. Instead, we utilize the encoder of a pre-trained video relighting model [26] to obtain frame-consistent geometric and material priors that regularize the interface components.

Figure 4. **Obtained transparency maps.** GT images (first row) and learned transparency maps (second row).

Geometric regularization. Specifically, we use the predicted depth $\hat{z}$ and normal $\hat{\mathbf{n}}$ from the encoder [26] to regularize the interface component's geometric attributes in G-buffer:

$$\mathcal{L}_{\text{geo}} = \lambda_d\mathcal{L}_{\text{depth}}(z, \hat{z}) + \lambda_n\mathcal{L}_{\text{normal}}(\mathbf{n}, \hat{\mathbf{n}}), \tag{12}$$

where $\mathcal{L}_{\text{depth}}$ follows a scale-invariant formulation [43], and $\mathcal{L}_{\text{normal}} = 1 - \cos(\mathbf{n}, \hat{\mathbf{n}})$ penalizes angular deviation [24]. These priors stabilize the interface geometry, providing a robust scaffold for the secondary transmission and reflection components.

Bootstrapping transparency. A key advantage of our decomposed representation is its ability to bootstrap transparency localization without manual segmentation masks. While segmentation modules [31] can identify transparent objects (*i.e.*, glass, bottle) in isolation, they often fail or return noisy results for scene-scale transparency due to ambiguous boundaries and overlapping transmitted radiance.

Instead, our decomposed representation bootstraps transparency localization, leveraging signals that emerge during optimization. The principal cue is the inter-component depth difference $\Delta z = |z_{\text{intr}} - z_{\text{trans}}|$, where z_{intr} and z_{trans} denote the interface and transmitted depths respectively. This signal, arising where multiple depth layers coexist, reveals the spatial separation between the interface and transmitted geometry as in Fig. 3. As a complementary signal, we use the diffuse-albedo map $\hat{a}$ obtained from [26], where lower values help identify specular-dominant transport rather than diffuse transport. We construct a binary transparency mask by thresholding these bootstrapped cues:

$$M_{\text{trans}} = \mathbf{1}\big((\Delta z > \tau_d) \wedge (\hat{a} < \gamma_a)\big), \tag{13}$$

which supervises the predicted transparency buffer t via:

$$\mathcal{L}_{\text{trans}} = \lambda_t \| M_{\text{trans}} - t \|_1. \tag{14}$$

As shown in Fig. 4, this bootstrapped transparency localization effectively identifies glass regions across diverse scenes, exhibiting inherent transparency disentanglement due to our explicit decomposed representation.

Table 1. **Quantitative evaluation of geometry on the synthetic 3D-FRONT-T dataset.** We report normal metrics (MAE, and accuracy thresholds of 11.25°, 22.5°) and depth metrics (AbsRel, RMSE, $\delta < 1.25$), along with mesh metrics (CD, F1-score).

Method	Normal			Depth			Mesh	
	MAE↓	11.25° ↑	22.5° ↑	AbsRel↓	RMSE↓	$\delta < 1.25$ ↑	CD↓	F1↑
2DGS [15]	25.97	52.19	64.55	0.20	0.24	76.71	0.85	0.688
PGSR [5]	25.39	56.83	65.58	0.22	0.25	76.68	0.52	0.807
Ref-GS [46]	41.55	18.61	36.79	0.28	0.36	57.52	1.29	0.408
EnvGS [39]	14.37	68.22	80.23	0.13	0.16	86.10	0.87	0.640
TSGS [24]	9.89	86.29	92.24	0.08	0.12	95.56	0.52	0.798
GLINT (Ours)	7.96	86.37	92.28	0.04	0.07	98.32	0.34	0.836

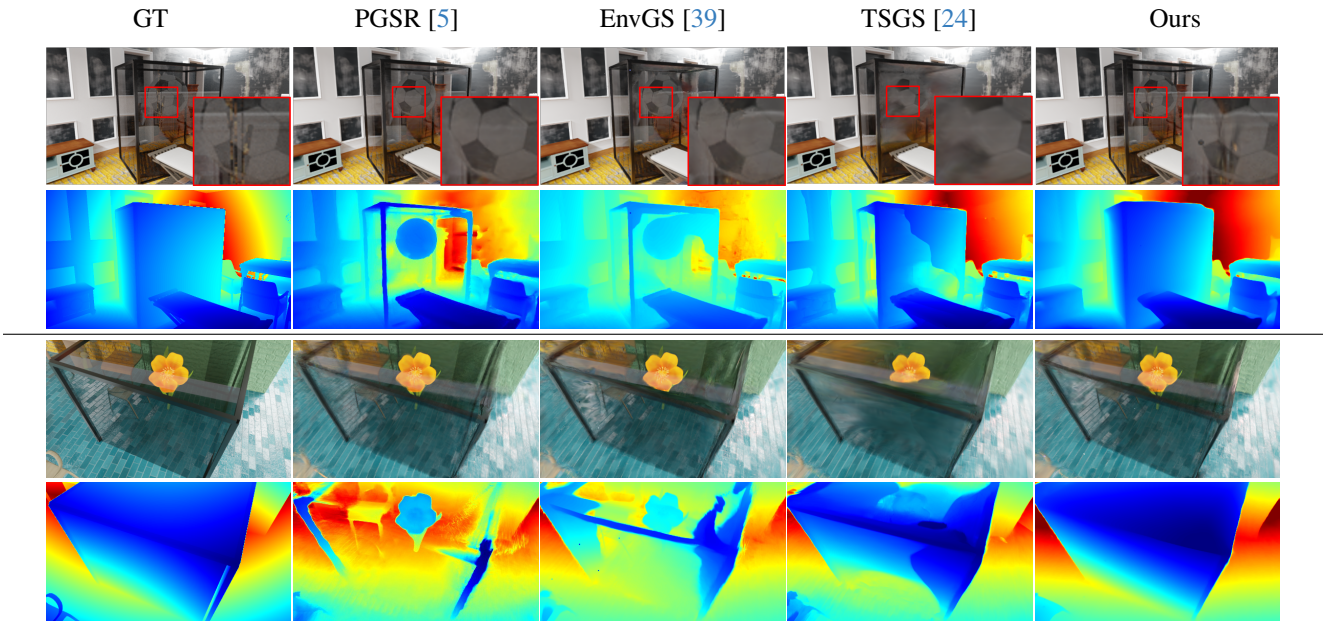


Figure 5. **Qualitative comparison on synthetic scenes.** Each column shows results from GT, PGSR [5], EnvGS [39], TSGS [24], and Ours. For each scene, rows correspond to RGB (top) and depth maps (bottom).

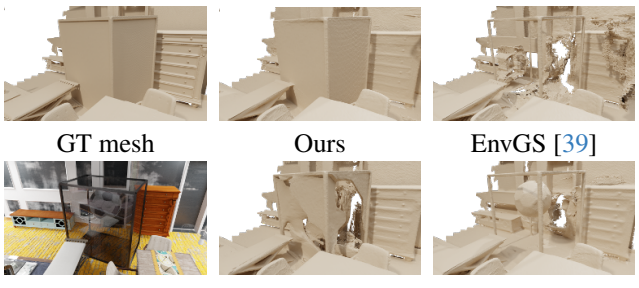


Figure 6. **Mesh visualization comparison.** The meshes are obtained from TSDF fusion following baselines.

4. Experimental Results

4.1. Implementation Details

We implement GLINT in PyTorch, integrating the 2DGS rasterizer [15] for primary interface rendering and a modified OptiX [30]-based ray tracer adapted from EnvGS [39] for secondary transmission and reflection queries. All experiments are conducted on a single NVIDIA RTX 4090 GPU. Following [19], we adopt adaptive densification and pruning, augmented with edge-aware normal smoothing and a normal consistency constraint between the rendered normal map and the depth-map gradients, as used in prior works [15, 24, 39]. For transparency bootstrapping, we set $\tau_d = 0.01$ and $\gamma_a = 0.05$ across all experiments. Additional implementation details are provided in the Appendix.

4.2. Datasets

We evaluate GLINT on both real-world and synthetic benchmarks to evaluate its performance in reconstructing geometry and appearance in scene-scale transparency.

Real-world. We leverage DL3DV-10K [27], a large-scale dataset containing diverse indoor and outdoor scenes with complex material properties, including strong reflection and

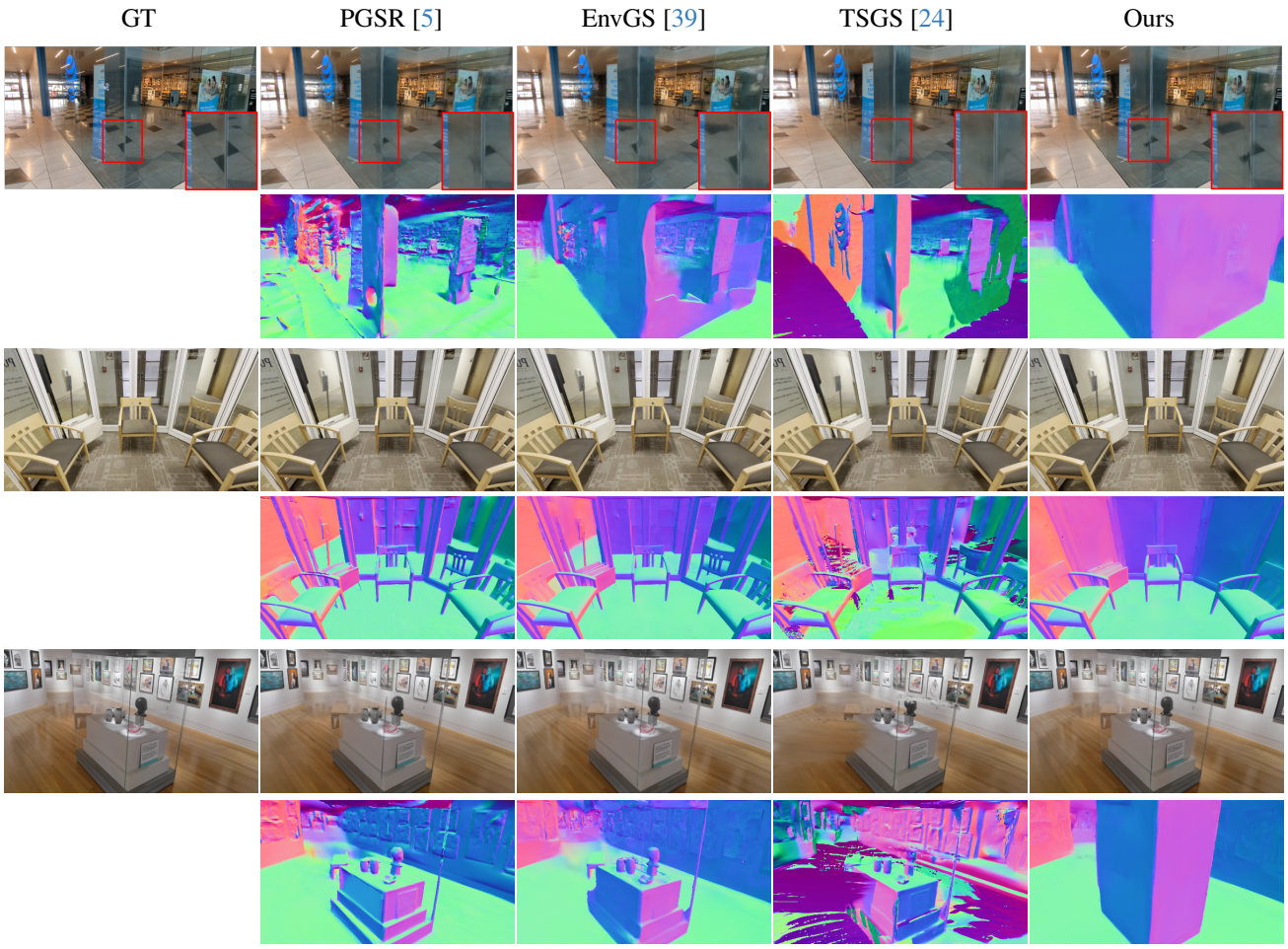


Figure 7. **Qualitative comparison on DL3DV-10K dataset.** Each column shows results from GT, PGSR [5], EnvGS [39], TSGS [24], and Ours. For each scene, rows correspond to RGB (top) and normal (bottom) maps.

transmission effects. We select a subset of 8 scenes featuring prominent transparent surfaces such as glass partitions, display cases, and windows. Since DL3DV-10K lacks ground-truth geometry annotations, the evaluation focuses on photometric quality and qualitative evaluation on geometric reconstruction.

Synthetic. To enable rigorous quantitative evaluation on transparent geometry, we introduce 3D-FRONT-T, a new synthetic benchmark specifically designed for scene-scale transparency. 3D-FRONT-T extends 3D-FRONT [8] by randomly placing thin transparent elements—such as glass panels and display cases—that enclose or interact with opaque objects, rendered with Blender [13]. The dataset contains 5 scenes with ground-truth depth and normal maps. Further details are provided in the Appendix.

4.3. Baselines and Metrics

Baselines. We compare against representative Gaussian splatting methods that enhance geometric fidelity or model complex non-Lambertian effects. Our baselines include planar-constrained variants (2DGS [15], PGSR [5]) and approaches specialized for strong specular reflection (Ref-GS [46], EnvGS [39]). We further include TSGS [24], which specifically targets the surface reconstruction of transparent surfaces. We follow the official implementation to reproduce their results. Detailed descriptions of all baselines are provided in the Appendix.

Evaluation metrics. For rendering quality, we report PSNR, SSIM [35], and LPIPS [45]. For geometry reconstruction, we follow Metric3Dv2 [14] to evaluate depth using absolute relative error (AbsRel), root mean squared error (RMSE), and threshold accuracies $\delta < 1.25$. For normals, we report the mean angular error (MAE) and angular accuracies under $11.25°$, and $22.5°$, representing the percentage of pixels whose estimated normals deviate from the ground truth by less than the specified angle. For mesh evaluation, we report Chamfer Distance (CD) and F1 score to assess surface reconstruction quality. The original metric for CD is in meters; we report CD in decimeters for readability.

Table 2. **Photometric evaluation on real and synthetic datasets.** GLINT achieves state-of-the-art rendering quality, quantitatively outperforming all baseline methods on both benchmarks.

Method	DL3DV-10K [27]			3D-FRONT-T		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
2DGS [15]	29.18	0.91	0.13	32.26	0.94	0.08
PGSR [5]	29.26	0.92	0.11	31.96	0.95	0.07
Ref-GS [46]	29.11	0.92	0.12	32.30	0.94	0.09
EnvGS [39]	29.65	0.91	0.12	33.71	0.94	0.07
TSGS [24]	25.94	0.85	0.19	28.80	0.87	0.14
GLINT (Ours)	30.21	0.92	0.11	34.50	0.96	0.05

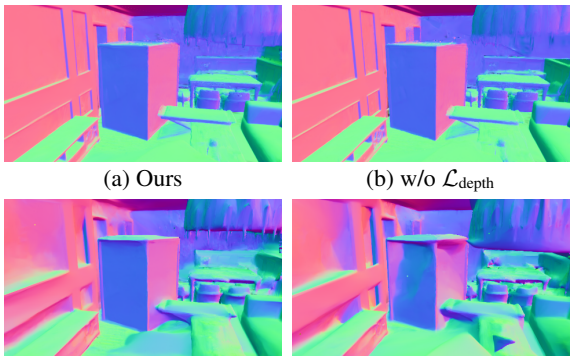
4.4. Baseline Comparisons

Quantitative evaluation. As shown in Tab. 1 and Tab. 2, GLINT achieves state-of-the-art performance on both real-world (DL3DV-10K [27]) and synthetic (3D-FRONT-T) benchmarks. While TSGS [24] attains reasonable geometric accuracy, its overall rendering quality remains low. In contrast, GLINT delivers both the highest photometric quality (Tab. 2) and the lowest geometric errors (Tab. 1), consistently outperforming all baselines. These results support our core hypothesis that explicitly decomposing radiance into interface, transmission, and reflection components effectively resolves the geometric ambiguities inherent to transparent scenes.

Qualitative evaluation. Figs. 5 and 7 present qualitative comparisons on synthetic and real-world datasets, demonstrating the superior capability of our approach in reconstructing scene-scale transparency. Baseline methods [5, 24, 39] exhibit missing or noisy geometry on glass regions, with corresponding normal and depth maps revealing either incomplete glass surfaces or noisy floating artifacts. While TSGS [24] partially alleviates this issue through first-surface transparency modeling with data-driven priors [31, 41], it still fails to recover transmitted radiance, resulting in blurred rendering of objects observed through glass. In contrast, GLINT successfully reconstructs well-defined transparent surfaces while simultaneously preserving accurate transmitted appearance and reflection details. We further visualize reconstructed meshes obtained via TSDF fusion from the rendered depth maps, as shown in Fig. 6. Compared to baseline methods, GLINT produces more coherent glass interfaces and reduces floating artifacts around transparent structures.

4.5. Ablation Studies

We systematically evaluate each component's contribution in Tab. 3. Ablating the transmission component ($\mathcal{G}_{\text{trans}}$) causes the most significant performance drop, as background content incorrectly blends with the interface Gaussians, creating geometric ambiguities that degrade both rendering quality and depth accuracy. Removing the reflection component ($\mathcal{G}_{\text{refl}}$) moderately reduces rendering fidelity, particularly in specular regions, though geometric metrics remain relatively stable. The transparency bootstrapping loss ($\mathcal{L}_{\text{trans}}$) serves as a stabilizing regularizer that guides the learning of transparency, yielding more consistent rendering and geometry. The geometric regularization losses, $\mathcal{L}_{\text{normal}}$ and $\mathcal{L}_{\text{depth}}$, respectively enhance surface orientation and depth accuracy. As shown in Fig. 8, our G-buffer guidance losses effectively regularize the interface geometry. Additional qualitative ablation results are provided in the Appendix.



(a) Ours (b) w/o $\mathcal{L}_{\text{depth}}$ (c) w/o $\mathcal{L}_{\text{normal}}$ (d) w/o $\mathcal{L}_{\text{geo}}$

Figure 8. **Qualitative Ablation study of G-buffer guidance.**

Table 3. **Ablation studies on our method.** We report PSNR, SSIM, LPIPS, Normal MAE, and Depth AbsRel.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	MAE ↓	AbsRel ↓
w/o $\mathcal{G}_{\text{trans}}$	32.26	0.93	0.085	8.11	0.038
w/o $\mathcal{G}_{\text{refl}}$	32.70	0.94	0.067	8.78	0.038
w/o $\mathcal{L}_{\text{trans}}$	33.57	0.94	0.066	8.07	0.037
w/o $\mathcal{L}_{\text{normal}}$	33.92	0.95	0.060	12.21	0.043
w/o $\mathcal{L}_{\text{depth}}$	34.05	0.95	0.058	8.54	0.061
w/o $\mathcal{L}_{\text{geo}}$	33.62	0.95	0.060	24.69	0.126
Full model	34.50	0.96	0.048	7.96	0.035

5. Conclusion and Discussion

In this study, we introduced GLINT, the first framework to reconstruct scene-scale transparency through an explicit decomposition of interface, transmission, and reflection components. This formulation enables coherent geometry and appearance modeling, effectively disentangling transparent regions. Extensive experiments show that GLINT achieves state-of-the-art performance across both appearance and geometry metrics, enabling faithful reconstruction of complex transparent structures.

Limitations. While our rendering pipeline focuses on first-order radiance interactions, extending it to handle multi-bounce phenomena including nested transparency remains an important direction for future work.

6. Acknowledgements

We sincerely appreciate the reviewers for their thoughtful comments. We also thank Kyu Beom Han and Taeyeon Kim for their careful proofreading. This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-25443318, Physically-grounded Intelligence: A Dual Competency Approach to Embodied AGI through Constructing and Reasoning in the Real World), in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00208506), and in part by the Korea Planning & Evaluation Institute of Industrial Technology (KEIT) and the Ministry of Trade, Industry & Resources (MOTIR) of the Republic of Korea (No. RS-2024-00417108).

References

[1] Frederick O Bartell, Eustace L Dereniak, and William L Wolfe. The theory and measurement of bidirectional reflectance distribution function (brdf) and bidirectional transmittance distribution function (btdf). In *Radiation scattering in optical systems*, pages 154–160. SPIE, 1981. 4

[2] Mojtaba Bemana, Karol Myszkowski, Jeppe Revall Frisvad, Hans-Peter Seidel, and Tobias Ritschel. Eikonal fields for refractive novel-view synthesis. In *ACM SIGGRAPH*, pages 1–9, 2022. 3

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 5

[4] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *ICLR*, 2025. 5

[5] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv preprint arXiv:2406.06521*, 2024. 2, 6, 7, 8

[6] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH*, pages 1–11, 2024. 2

[7] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 2

[8] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *ICCV*, pages 10933–10942, 2021. 7, 1

[9] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *CVPR*, pages 20796–20805, 2024. 2

[10] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv:2311.16043*, 2023. 2

[11] Jinwei Gu, Ravi Ramamoorthi, Peter N Belhumeur, and Shree K Nayar. Dirty glass: Rendering contamination on transparent surfaces. *Rendering Techniques*, 159:170, 2007. 3

[12] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *CVPR*, 2024. 2

[13] Roland Hess. *Blender Foundations: The Essential Guide to Learning Blender 2.6*. Focal Press, 2010. 7, 1

[14] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *PAMI*, 2024. 7, 5

[15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH*, pages 1–11, 2024. 2, 3, 6, 7, 8

[16] Letian Huang, Dongwei Ye, Jialin Dan, Chengzhi Tao, Huiwen Liu, Kun Zhou, Bo Ren, Yuanqi Li, Yanwen Guo, and Jie Guo. Transparentgs: Fast inverse rendering of transparent objects with gaussians. *ACM TOG*, 44(4):1–17, 2025. 3

[17] 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 *CVPR*, pages 5322–5332, 2024. 2

[18] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024. 5

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139–1, 2023. 1, 2, 3, 6

[20] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *NeurIPS*, 37:80965–80986, 2024. 2

[21] Jeongyun Kim, Jeongho Noh, Dong-Guw Lee, and Ayoung Kim. Transplat: Surface embedding-guided 3d gaussian splatting for transparent object manipulation. *arXiv preprint arXiv:2502.07840*, 2025. 2

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 5

[23] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d

gaussian radiance fields with global-local depth normalization. In *CVPR*, pages 20775–20785, 2024. 5

[24] Mingwei Li, Pu Pang, Hehe Fan, Hua Huang, and Yi Yang. Tsgs: Improving gaussian splatting for transparent surface reconstruction via normal and de-lighting priors. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 7220–7229, 2025. 2, 3, 5, 6, 7, 8

[25] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *CVPR*, pages 1262–1271, 2020. 3

[26] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Chih-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusion renderer: Neural inverse and forward rendering with video diffusion models. In *CVPR*, pages 26069–26080, 2025. 2, 4, 5, 3, 6

[27] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, pages 22160–22169, 2024. 2, 6, 8, 1

[28] 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 TOG*, 43(6):1–19, 2024. 3

[29] Yinyu Nie, Angela Dai, Xiaoguang Han, and Matthias Nießner. Learning 3d scene priors with 2d supervision. In *CVPR*, pages 792–802, 2023. 1

[30] Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al. Optix: a general purpose ray tracing engine. *ACM TOG*, 29 (4):1–13, 2010. 6

[31] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 5, 8, 2, 7

[32] Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer Graphics Forum*, pages 233–246. Wiley Online Library, 1994. 4

[33] Jia-Mu Sun, Tong Wu, Ling-Qi Yan, and Lin Gao. Nunerf: neural reconstruction of nested transparent objects with uncontrolled capture environment. *ACM TOG*, 43(6):1–14, 2024. 3

[34] Jinguang Tong, Xuesong Li, Fahira Afzal Maken, Sundaram Muthu, Lars Petersson, Chuong Nguyen, and Hongdong Li. Gs-2dgs: Geometrically supervised 2dgs for reflective object reconstruction. In *CVPR*, pages 21547–21557, 2025. 5

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5, 7

[36] Patrick Wieschollek, Orazio Gallo, Jinwei Gu, and Jan Kautz. Separating reflection and transmission images in the wild. In *ECCV*, pages 89–104, 2018. 2

[37] Tong Wu, Jia-Mu Sun, Yu-Kun Lai, Yuewen Ma, Leif Kobbelt, and Lin Gao. Deferredgs: Decoupled and editable gaussian splatting with deferred shading. *arXiv preprint arXiv:2404.09412*, 2024. 2, 4

[38] Tianhao Wu, Hanxue Liang, Fangcheng Zhong, Gernot Riegler, Shimon Vainer, Jiankang Deng, and Cengiz Oztireli. αsurf: Implicit surface reconstruction for semi-transparent and thin objects with decoupled geometry and opacity. In *3DV*, pages 44–63. IEEE, 2025. 3

[39] Tao Xie, Xi Chen, Zhen Xu, Yiman Xie, Yudong Jin, Yujun Shen, Sida Peng, Hujun Bao, and Xiaowei Zhou. Envgs: Modeling view-dependent appearance with environment gaussian. In *CVPR*, pages 5742–5751, 2025. 2, 3, 4, 5, 6, 7, 8

[40] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 5

[41] 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 TOG*, 2024. 5, 8, 2

[42] Keyang Ye, Qiming Hou, and Kun Zhou. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH*, pages 1–10, 2024. 2, 4

[43] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *NeurIPS*, 35:25018–25032, 2022. 5, 2, 3

[44] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *CVPR*, pages 19447–19456, 2024. 2

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5, 7

[46] Youjia Zhang, Anpei Chen, Yumin Wan, Zikai Song, Junqing Yu, Yawei Luo, and Wei Yang. Ref-gs: Directional factorization for 2d gaussian splatting. In *CVPR*, pages 26483–26492, 2025. 4, 6, 7, 8, 2

GLINT: Modeling Scene-Scale Transparency via Gaussian Radiance Transport

Supplementary Material

This supplementary material provides additional details on (i) the proposed synthetic 3D-Front-T dataset (Sec. A), (ii) the baseline methods used in our experiments (Sec. B), and (iii) implementation details (Sec. C). In addition, we include qualitative results on ablation studies (Sec. D), comparative analysis with baselines (Sec. E), discussion on foundation models (Sec. F), detailed discussion on limitations and future work (Sec. G), and additional qualitative results on both real and synthetic datasets (Sec. H).

A. 3D-FRONT-T Dataset

To enable rigorous quantitative evaluation of scene-scale transparency reconstruction, we introduce 3D-FRONT-T, a new synthetic benchmark for scene-scale transparency reconstruction. We construct our dataset built upon the 3D-FRONT (3D Furnished rooms with layouts and semantics) dataset [8]. While existing real-world datasets contain transparent scenes [27], they lack ground-truth geometry annotations for quantitative evaluation on geometry reconstruction. Our 3D-FRONT-T addresses this by providing depth and normal ground truth alongside RGB renderings of transparent scenes.

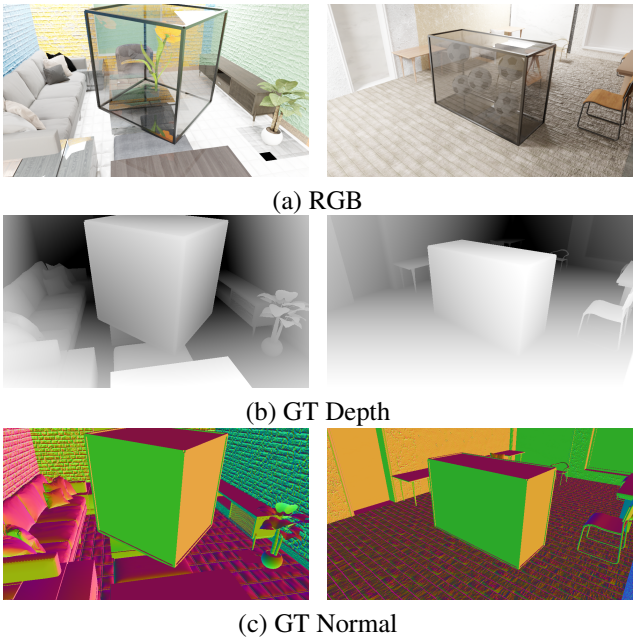
A.1. Dataset Construction

We collect 5 indoor scenes from the 3D-FRONT [8] datasets with diverse configurations. For each scene, we first identify the largest room by floor area and retain only the objects within that room, including walls, ceilings, doors, windows, and furniture placed on the floor. Then, we texture the floor, wall and ceiling in the scene with the diverse random materials following [29].

On top of this setup, we create and put a glass display container with a black metal frame, enclosing opaque objects. The container consists of six thin glass panels with a supporting frame structure. The container size is randomly determined to create different sizes of transparency regions.

To achieve realistic placement, we utilize Blender's physics-based simulation [13] to settle objects into physically plausible configurations. The glass container is initialized at a random position above the floor and released using rigid-body dynamics, allowing it to naturally come to rest on the floor or on top of existing furniture. This process ensures physically consistent placement while generating diverse occlusion patterns with the surrounding scene geometry.

All scenes are rendered using the Blender Cycles path tracer with 4096 samples per pixel and a maximum of 200 light bounces across all channels (diffuse, glossy, and transmission) to accurately capture complex light interactions through transparent surfaces. For our experiments, we render all images at a resolution of 960×540. For each scene, we generate camera trajectories which samples continuous viewpoints that ensure visibility of key objects including the transparent container. The example images of the dataset are shown in Fig. 9.



(a) RGB

(b) GT Depth

(c) GT Normal

Figure 9. **3D-FRONT-T dataset.** RGB, depth, and normal ground truth examples. As can be seen in the transparent surface (glass), the outgoing radiance for each pixel is entanglement of radiance from interface surface and the transmitted radiance.

A.2. Ground Truth Annotations

For each rendered viewpoint, we provide a complete set of physically accurate ground-truth annotations tailored for evaluating scene-scale transparency reconstruction:

RGB Images. High-fidelity renderings produced via multi-bounce path tracing in Blender Cycles. These images capture intricate light behavior including interreflections, refractions, and transmission through transparent materials, serving as a challenging benchmark for appearance modeling.

Depth Maps. Metric depth is extracted directly from Blender's rendering pipeline, ensuring physically consistent geometry even in regions partially occluded or viewed through transparent media. This enables rigorous evaluation of depth recovery in transparent scenes, a regime un-

derexplored in existing benchmarks.

Normal Maps. Per-pixel surface normals are rendered for all visible surfaces. These provide robust supervision signals for assessing fine-grained geometric accuracy, independent of texture or lighting cues.

Taken together, these annotations establish a comprehensive benchmark for transparent-scene reconstruction. To encourage reproducibility and extensibility, we will release our full dataset-generation pipeline, implemented on top of BlenderProc [7], enabling automatic synthesis of large-scale, diverse scenes with configurable and physically plausible object placement.

B. Overview of Baseline Methods

We compare GLINT with representative Gaussian-splatting–based approaches that cover planar-constrained geometry modeling, reflective radiance modeling, and transparent-surface reconstruction. Below, we briefly summarize each baseline to clarify their modeling assumptions and limitations in the context of scene-scale transparency.

2DGS [15] adopts 2D Gaussians to improve geometric accuracy over volumetric 3DGS. While effective for opaque surfaces, its monolithic α-compositing cannot separate interface geometry from secondary effects, causing transparent surfaces to be ignored or entangled into a single depth layer.

PGSR [5] extends planar-aligned Gaussian primitives with additional geometric constraints. Despite achieving sharper surface reconstruction, it shares the same opacity-based rendering pipeline and therefore struggles with multi-depth radiance, often collapsing glass regions into the background.

Ref-GS [46] models view-dependent appearance through directional factorization, enabling more expressive specular materials. However, its formulation implicitly assumes that all non-Lambertian behavior arises from surface reflection, without accounting for light transmission. As a result, transparent materials whose appearance is governed by both interface reflectance and background transmission are incorrectly treated as purely reflective surfaces, producing biased material estimates and degraded geometric cues in regions where transmitted radiance is dominant.

EnvGS [39] models environment radiance using a dedicated set of Gaussians and performs ray-traced reflection queries, supported by a monocular normal prior to stabilize geometry estimation from limited viewpoints. While highly effective for opaque materials, it lacks any mechanism to account for light interactions through transparent surfaces, leading background content seen through glass to be incorrectly attributed to reflections.

TSGS [24] targets transparent objects using first-surface rasterization combined with monocular normal [41], delighting [41], and segmentation priors [31]. While effective for thin, object-centric transparency, its formulation assumes a single transparent interface and does not explicitly model transmitted radiance. Consequently, scenes containing multiple depth layers (e.g., glass–background–interior structures) often exhibit blurred transmission and incomplete geometry reconstruction. The segmentation module also produces noisy or missing transparency masks, an issue we further analyze in the next section.

Overall, existing baselines either (i) emphasize geometric fidelity, (ii) specialize in reflective appearance modeling, or (iii) operate under object-centric transparency assumptions. None provide a unified framework capable of addressing the inherently ill-posed nature of scene-scale transparency, where accurate reconstruction requires jointly modeling interface geometry, background transmission, and reflection. Our study bridges this gap by introducing a decomposed Gaussian representation and transparency-aware radiance transport, which together provide a more physically consistent formulation for scene-scale transparency.

C. Additional Implementation Details

In this section, we provide comprehensive implementation details for reproducibility, including initialization process, multi-stage optimization, and the specific loss formulations designed to ensure physical plausibility and geometric consistency. Our code will be made publicly available.

Initialization. To establish a reliable geometric foundation, we initialize the interface Gaussian ($\mathcal{G}_{\text{intr}}$) and transmission Gaussian ($\mathcal{G}_{\text{trans}}$) primitives using the sparse point cloud derived from Structure-from-Motion (SfM) [9]. Meanwhile, we follow EnvGS [39] for initializing reflectance component ($\mathcal{G}_{\text{refl}}$), where we partition the scene into N^3 sub-grids by partitioning the bounding box. Then we randomly sample K primitives within the grid where we set $N = 32$ and $K = 5$.

Optimization Schedule. To ensure stable training, we adopt a multi-stage optimization strategy that progressively recovers scene geometry before refining complex radiance effects. We begin with a 5k-iteration warm-up stage in which only the interface component is optimized, allowing the primary surface geometry to converge. Afterward, the transmission and reflection components are introduced for joint optimization. Finally, between 40k and 60k iterations, we freeze the interface Gaussians and update only the transmission and reflection Gaussians to refine their radiance behavior without destabilizing the established geometry.

Regularization. For geometric regularization, we adopt a scale-and-shift–invariant depth loss together with a normal consistency loss, following the formulation of MonoSDF [43]. Let z denote the rendered depth from the

Figure 10. **Qualitative ablation study on representation components.** Visual comparison between the full model and ablated variants.

interface component and $\hat{z}$ the monocular depth prior predicted by the encoder [26]. We align z and $\hat{z}$ by solving for the optimal scale w and shift q in closed form, and define the depth loss as:

$$\mathcal{L}_{\text{depth}} = \frac{1}{N}\sum_{i=1}^{N}\left(w\, z_i + q - \hat{z}_i\right)^2, \tag{15}$$

where N denotes the number of pixels in the image. This formulation removes the global scale ambiguity of monocular predictions while preserving their relative depth structure. The scale–shift parameters (w, q) are recomputed for each image using the closed-form least-squares solution [43].

For the normal loss, we follow the thresholded normal supervision strategy introduced in TSGS [24] to mitigate the influence of noisy monocular priors. Given the rendered normal $\mathbf{n}$ and the normal prior from [26] $\hat{\mathbf{n}}$, we apply a cosine-similarity mask from 10k iterations:

$$\mathbf{M}_{\text{prior}}(u) = \left[\langle \mathbf{n}(u), \hat{\mathbf{n}}(u)\rangle \geq \tau_{\text{n}}\right],$$

and compute the masked normal loss as:

$$\mathcal{L}_{\text{normal}} = \sum_{u} \mathbf{M}(u)\big(1 - \langle \mathbf{n}(u), \hat{\mathbf{n}}(u)\rangle\big).$$

We set $\tau_{\text{n}} = 0.3$ for all experiments.

D. Qualitative Ablation Studies

In this section, we present additional qualitative ablation results to further elucidate the distinct roles of each component within the GLINT framework.

First, we examine the impact of our decomposed representation in Fig. 10. Removing the transmission branch ($\mathcal{G}_{\text{trans}}$) leads to significant entanglement between the interface and background content, resulting in geometric inconsistencies and a washed-out appearance in regions observed behind glass. Excluding the reflection branch ($\mathcal{G}_{\text{refl}}$) diminishes specular cues, leading to a loss of realistic surface gloss, particularly on glass or highly reflective surfaces.

Next, we visualize the ablation of geometry regularization losses in Fig. 11. While our framework remains relatively robust, removing specific losses introduces characteristic degradations that highlight their individual contributions. Excluding the depth loss ($\mathcal{L}_{\text{depth}}$) causes inaccuracies in interface geometry placement, manifesting as deviations in absolute depth, especially for transparent surfaces. Removing the normal loss ($\mathcal{L}_{\text{normal}}$) results in less consistent surface orientation, which appears as locally unstable shading and noisy normal transitions. When all geometric priors are removed ($\mathcal{L}_{\text{geo}}$), these artifacts accumulate, leading to noticeable distortions in depth and normal maps. This indicates that the geometric constraints operate complementarily to maintain structural fidelity.

Finally, we analyze the effect of the transparency bootstrapping loss ($\mathcal{L}_{\text{trans}}$) in Fig. 12. Omitting this loss yields noisier and spatially inconsistent transparency estimates. This is because the loss utilizes 3D geometric cues derived from depth separation to guide the optimization toward sharp and physically consistent transparency boundaries.

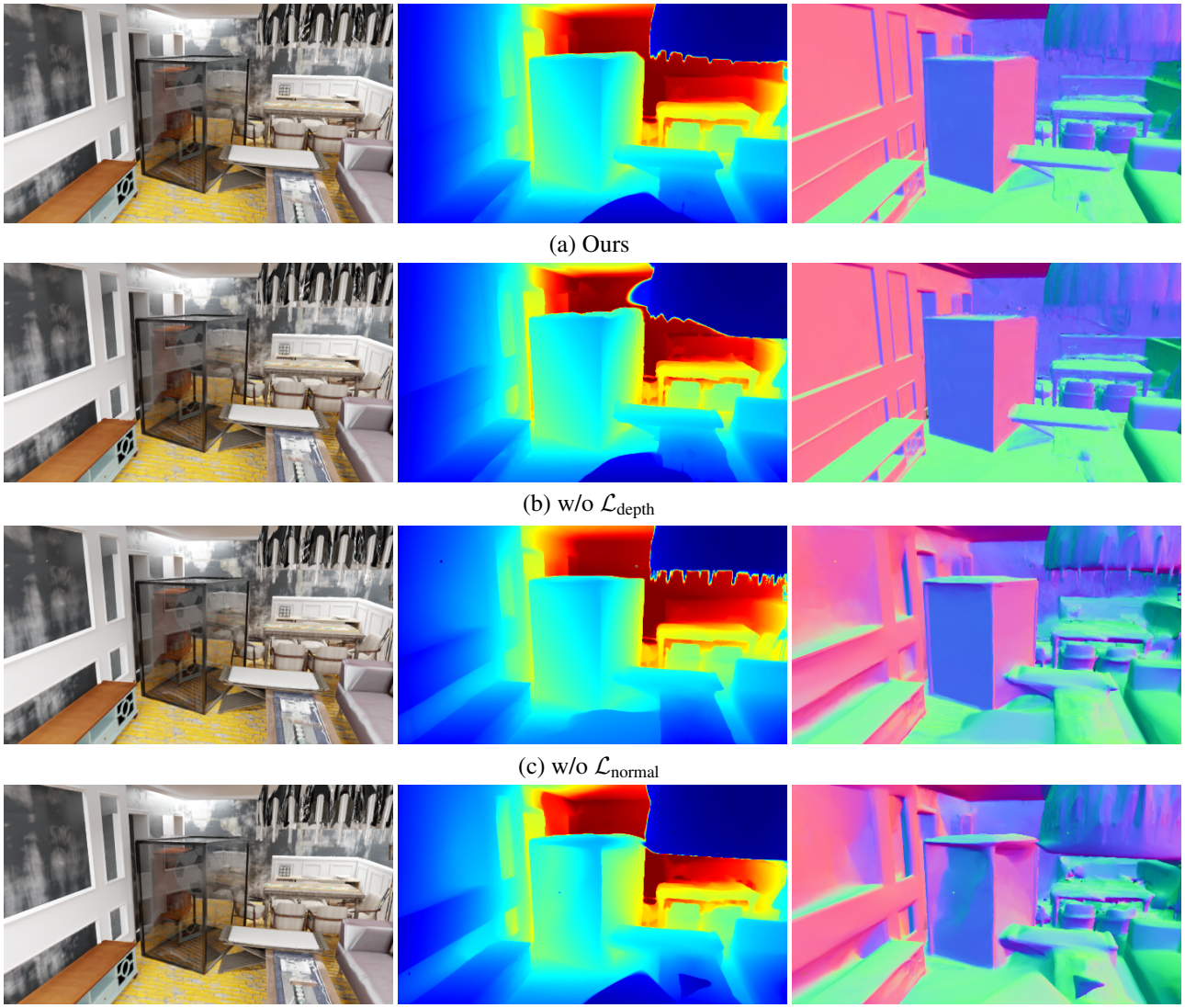
E. Comparative Analysis with Baselines

In this section, we provide a comprehensive comparative analysis against baseline methods. We focus on three key aspects: the interpretability of radiance decomposition, the accuracy of transparency localization, and a quantitative analysis of computational costs versus reconstruction quality.

Radiance decomposition. Fig. 13 presents an extended comparison of radiance decomposition between our method and EnvGS [39].

For our method (Fig. 13(a)), the decomposed Gaussian representation enables a clear and physically grounded partitioning of radiance. The interface component isolates the surface base color, while the reflection component captures specular highlights. Crucially, the transmission component reconstructs the background scene visible specifically through transparent surfaces, revealing the correct spatial structure behind the glass. These components form a coherent explanation of the observed radiance, demonstrating that our explicit decomposition naturally disentangles overlapping radiance sources in transparent regions.

In contrast, EnvGS (Fig. 13(b)) focuses solely on modeling view-dependent reflections via environment Gaussians and lacks a dedicated transmission component. Consequently, radiance originating from behind transparent sur-



(a) Ours

(b) w/o $\mathcal{L}_{\text{depth}}$

(c) w/o $\mathcal{L}_{\text{normal}}$

(d) w/o $\mathcal{L}_{\text{geo}}$

Figure 11. **Effect of geometric losses.** Ablating geometric supervision leads to degraded geometry reconstruction. Removing $\mathcal{L}_{\text{depth}}$ produces inaccurate interface depth, removing $\mathcal{L}_{\text{normal}}$ results in unstable surface orientation, and removing all geometric losses ($\mathcal{L}_{\text{geo}}$) severely distorts both depth and normals.

faces is incorrectly entangled within its diffuse or reflection modeling, leading to mixed or incomplete visual explanations. Furthermore, its reflection strength is modulated by a single scalar weight s, lacking the physically accurate Fresnel-driven angular dependence inherent to our formulation.

To further validate the necessity of our decomposition, we conducted an experiment training EnvGS [39] with the geometric normal priors from DiffusionRenderer [26], identical to our setup. As illustrated in Fig. 14, enforcing geometric consistency on EnvGS paradoxically leads to corrupted appearance rendering. This degradation occurs because EnvGS lacks a dedicated transmission component and fundamentally conflates transmitted and reflected radiance into a single surface interaction. Typically, EnvGS implicitly minimizes photometric error by distorting the geometry to effectively baking background textures onto incorrect depths. However, when the surface geometry is enforced via stronger priors, this compensatory mechanism is blocked. Consequently, the model is forced to approximate the superposition of multiple radiance layers onto a single interface, leading to an averaging effect that manifests as severe blurring.

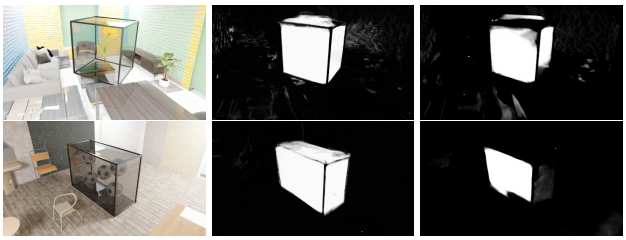


Figure 12. **Effect of the transparency loss $\mathcal{L}_{\text{trans}}$.** With the proposed transparency loss $\mathcal{L}_{\text{trans}}$, the learned transparency maps align well with the true transmitted content, while removing this loss leads to noisy or inconsistent transparency estimation.

Transparency Map Comparison. We compare our learned transparency maps with those generated by Grounded-SAM-2 (G-SAM2) [31], a state-of-the-art open-world segmentation model. As illustrated in Fig. 15, although G-SAM2 can roughly localize transparent objects, it frequently yields spatially inconsistent or fragmented masks. Common failure cases include missing sections of glass cabinets or exhibiting severe flickering across viewpoints, even when utilizing the tracking mode. These limitations arise because G-SAM2 relies on 2D image features, which are inherently ambiguous for transparent surfaces that mix reflection and transmission, lacking a unified understanding of 3D geometry.

In contrast, GLINT bootstraps transparency localization by explicitly leveraging 3D geometric cues—specifically, the depth discrepancy (Δz) between the interface and transmission components that emerges during optimization and the diffuse-albedo prior from [26]. This allows our method to generate spatially coherent and boundary-sharp transparency maps that accurately align with the physical extent of the glass.

Computational Costs. We report a runtime comparison with EnvGS and TSGS on the synthetic 3D-FRONT-T dataset (downscale $\times 2$), with results summarized in Tab. B. TSGS achieves the highest speed due to rasterization-only rendering, while EnvGS adopts a hybrid formulation limited to reflection. Our method introduces additional overhead from explicit multi-component optimization and transmission handling, resulting in lower throughput. However, this design choice directly supports transparent geometry modeling and leads to improved reconstruction quality, reflecting a deliberate and practical trade-off between efficiency and capability.

F. Discussion on Foundation Models

In this section, we discuss the motivation behind integrating foundation models into the GLINT framework and analyze the advantages of video-based priors compared to conventional image-based approaches in the context of transparent scene reconstruction.

Table 4. Comparison of computational costs and reconstruction quality on the 3D-FRONT-T dataset.

Method	FPS (↑)	Training Time (↓)	PSNR (↑)	MAE (↓)	AbsRel (↓)
TSGS [24]	**159**	**∼40 mins**	28.80	9.89	0.08
EnvGS [39]	80	∼1 h	33.71	14.37	0.13
Ours	51	∼2.5 h	**34.50**	**7.96**	**0.04**

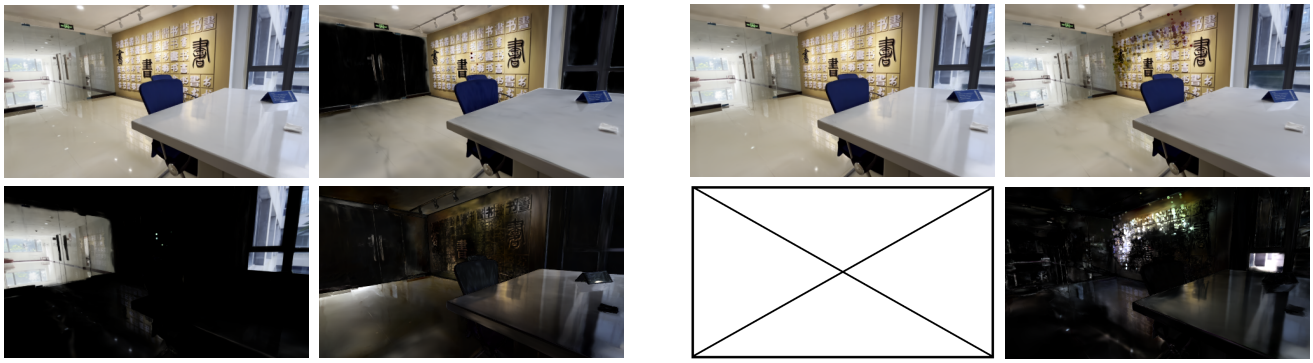
In our implementation, we utilize the inverse-rendering encoder of the video relighting model, DiffusionRenderer [26], to obtain geometric (depth, normal) and material (diffuse-albedo) priors. A key motivation for choosing a video-based foundation model over monocular estimators is its multi-view consistency. Since the model leverages the architecture of Stable Video Diffusion [3], it processes multiple frames in a single feed-forward pass, producing geometric cues that are coherent across viewpoints. This is particularly crucial for transparent surfaces, where per-frame estimation often flickers or yields inconsistent depth due to view-dependent reflections. To the best of our knowledge, GLINT is the first approach to adopt video relighting priors for 3D Gaussian splatting reconstruction, demonstrating that material-aware priors effectively aid in distinguishing between interface and transmitted radiance.

Comparison with Image-based Priors. Recent approaches [24, 34, 39] typically combine various image-based foundation models, such as monocular depth [4, 14], normal estimators [18, 41], and segmentation modules (e.g., Grounded-SAM [22, 31]). For instance, TSGS [24] relies on Grounded-SAM to mask transparent objects and uses StableDelight and StableNormal [41] to make the texture of the transaprent surface dinstinctive. While empirically beneficial for object-centric scenes with clear boundaries, this mask strategy often struggles with scene-scale transparency—such as large glass facades or windows—where semantic boundaries are ambiguous and binary segmentation fails to capture the continuous transition of radiance as in Fig. 15.

G. Limitations and Future Work

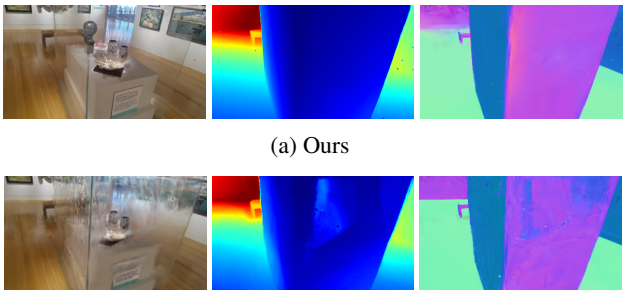
While GLINT achieves state-of-the-art performance in reconstructing scene-scale transparent geometry and appearance, there are still room for improvement in various perspectives. In this section, we detail the discussion on the current boundary conditions of our framework and suggest potential directions to address them.

Decomposition ambiguity under sparse observations. Our method implicitly relies on multi-view consistency to



(a) Ours (b) EnvGS [39]

Figure 13. **Radiance decomposition comparison.** Each 2×2 grid is arranged in clockwise order as rendered RGB, base color, reflection and transmission. EnvGS does not provide a transmission component, so the corresponding slot is left blank.



(a) Ours

(b) EnvGS [39] w/ DiffusionRenderer [26]

Figure 14. **Comparison with EnvGS [39] trained with a DiffusionRenderer [26] prior.** With the DiffusionRenderer normal priors, EnvGS still fails to produce consistent depth and normals for transparent regions. In particular, the transmission surface remains incorrectly reconstructed, showing blurry rendering quality.

disentangle the intertwined radiance contributions from interface reflection and background transmission. Consequently, in scenarios with sparse viewpoints or limited parallax, where a surface is observed from a stationary angle, the problem becomes inherently ill-posed. In such cases, the optimization may struggle to uniquely assign radiance to either the reflection ($\mathcal{G}_{\text{refl}}$) or transmission ($\mathcal{G}_{\text{trans}}$) component. We believe that incorporating high-level semantic understanding could resolve this ambiguity. Future work could leverage vision-language models (VLMs) or unprojecting semantic features to enforce physically and semantically plausible decomposition, ensuring that regions identified as glass (e.g., windows vs. mirrors) adhere to their expected optical behaviors even under constrained observation.

Recursive light transport. To maintain rendering efficiency and optimization stability, our current radiance transport formulation focuses on primary transmission and reflection events (i.e., up to first-order interactions at the interface). While this approximation is sufficient for most architectural glass and display cases, it may not fully capture complex multi-bounce phenomena found in nested transparent structures, such as a glass vase inside a glass cabinet or a mirror room. Extending our hybrid rendering pipeline to support recursive ray-tracing or multi-pass rendering with physically-based rendering formulation would allow for the simulation of higher-order approximation of light transport, albeit at the cost of increased computational overhead.

H. Additional Qualitative Results

We present additional qualitative examples to complement the results in the main paper. Figures 16 and 17 provide further visualizations on representative scenes from both the 3D-FRONT-T benchmark and the real-world DL3DV-10K dataset [27].

We also include the video results for the visualization of the continuous frames. We kindly refer the readers to the attached supplementary videos.

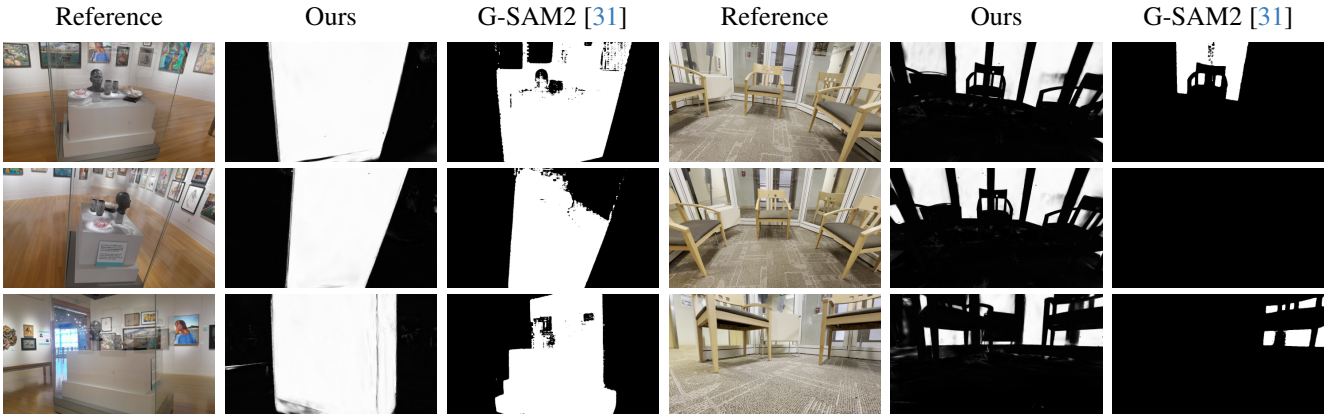


Figure 15. Comparison of transparency masks across two scenes. Each scene is shown with three viewpoints (rows), including ground-truth reference RGB (left), our predicted transparency maps (middle), and G-SAM2 [31] masks (right). GT images provide visual context, highlighting that our method consistently isolates transparent surfaces, while Grounded-SAM2 often produces noisy, incomplete or completely missing masks.

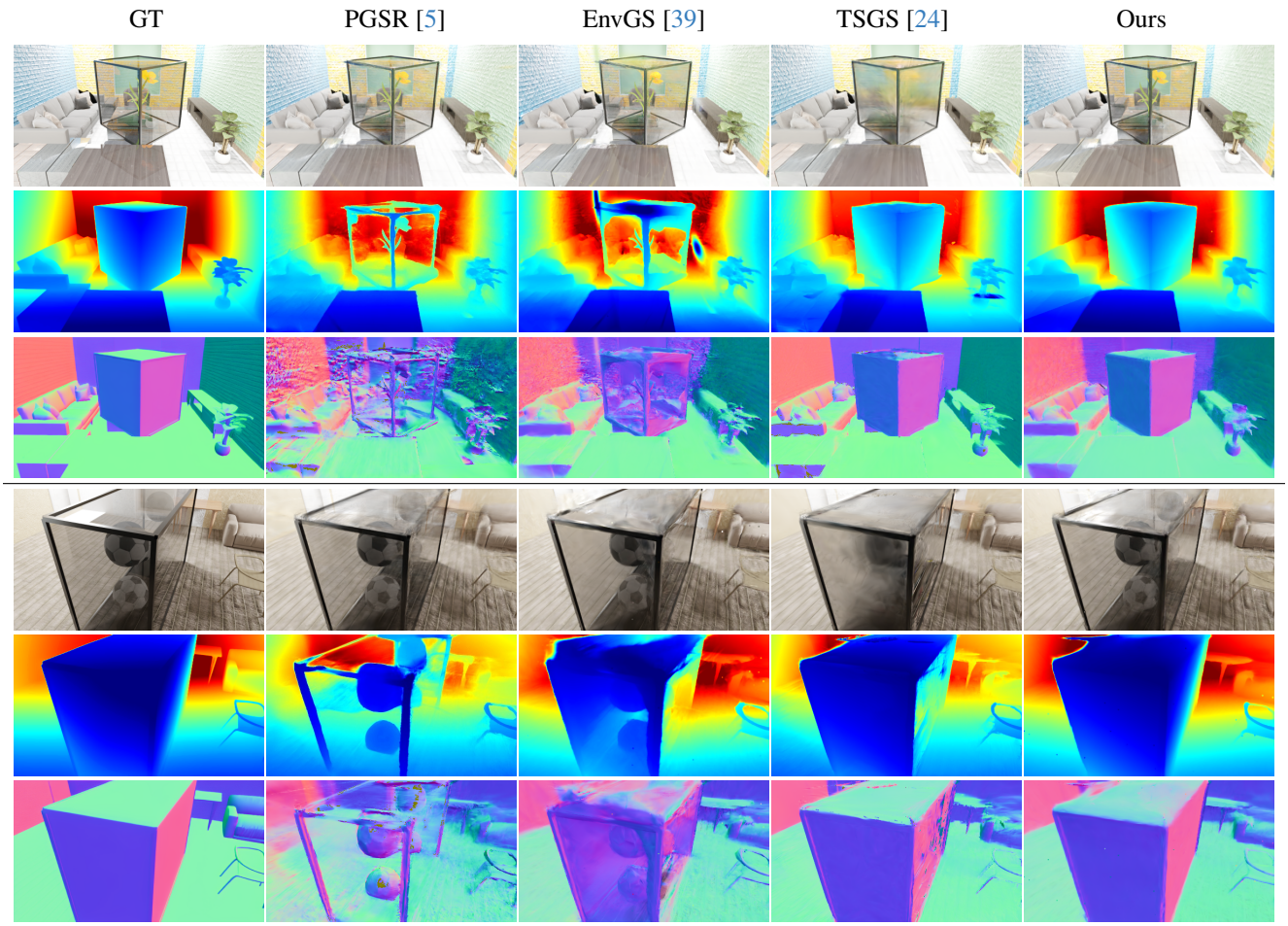


Figure 16. **Qualitative comparison on synthetic scenes.** Each column shows results from GT, PGSR [5], EnvGS [39], TSGS [24], and Ours. For each scene, rows correspond to RGB (top), depth (middle), and normal (bottom) maps.

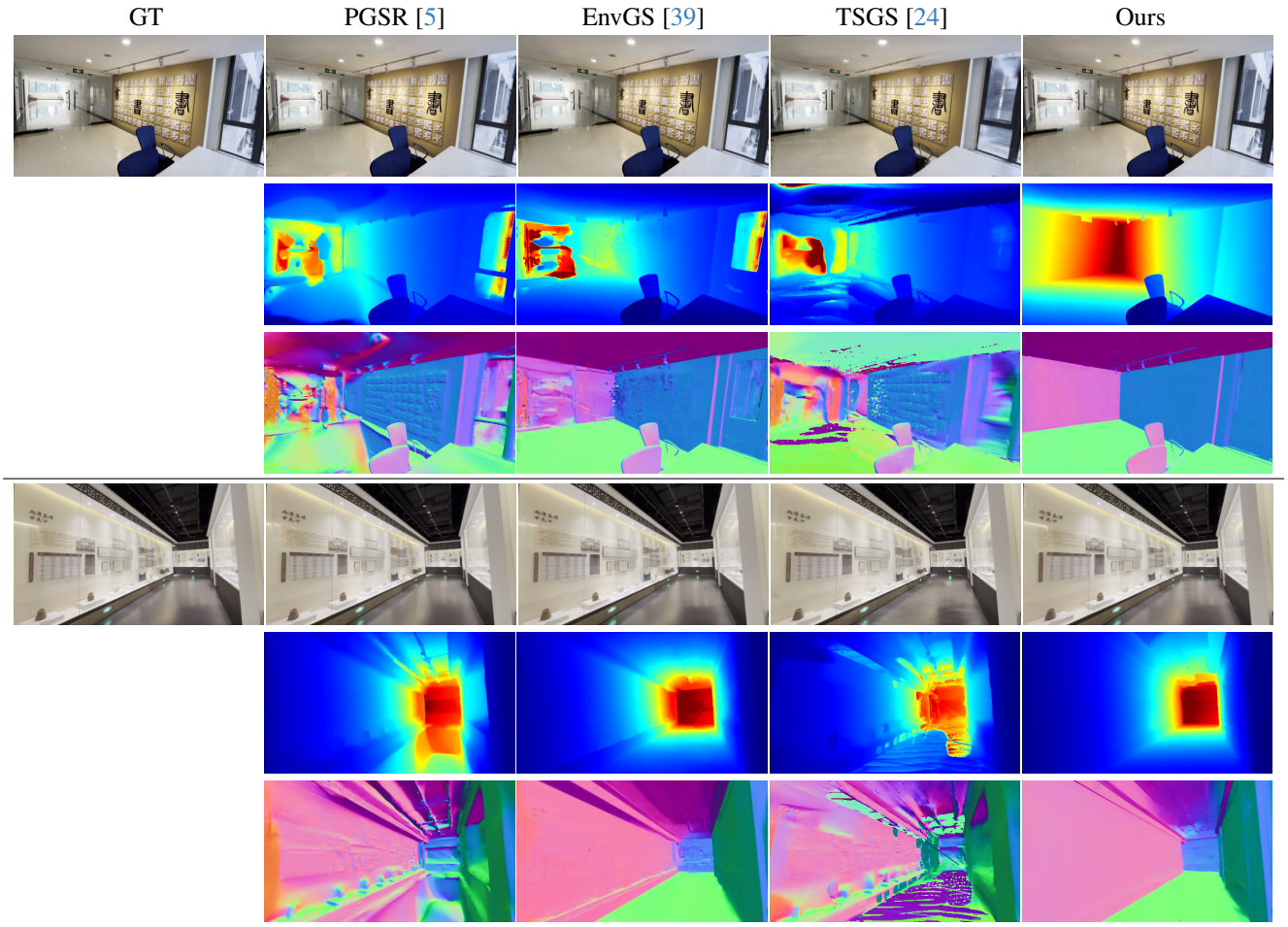


Figure 17. **Additional qualitative comparisons on the DL3DV-10K dataset.** Each column shows results from GT, PGSR [5], EnvGS [39], TSGS [24], and Ours. For each scene, rows correspond to RGB (top), depth (middle), and normal (bottom) predictions.