TransmissiveGS: Residual-Guided Disentangled Gaussian Splatting for Transmissive Scene Reconstruction and Rendering

Zhenyu Liang **Xiao Zhang** **Tianchao Li** **Jack C.P. Cheng** **Chi-Keung Tang**

The Hong Kong University of Science and Technology

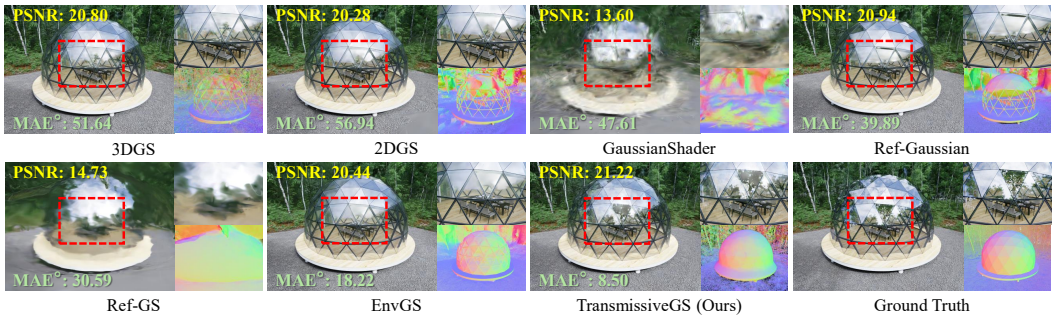

Figure 1: **Comparison on a challenging example consisting of a hemispherical, reflective, and transmissive dome.** We present **TransmissiveGS**, the *first* Gaussian Splatting framework that achieves both accurate transmissive surface reconstruction (MAE° ↓) and photorealistic inverse rendering (PSNR ↑) in non-planar transmissive scenes featuring near-field reflections and visible transmitted content behind the transmissive surface, without requiring any geometry prior.

Abstract

Transmissive scenes are ubiquitous in daily life, yet reconstructing and rendering them remains highly challenging due to the inherent entanglement between near-field reflections from the surrounding environment on the transmissive surface, and the transmitted content of the scene behind it. This coupling gives rise to dual surface geometries and dual radiance components within each observation, posing ambiguities for standard methods. We present TransmissiveGS, a novel framework for disentangled reconstruction and rendering of transmissive scenes. Specifically, we model the scene with a dual-Gaussian representation and introduce a deferred shading function to jointly render the two Gaussian components. To separate reflection and transmission, we exploit the inherent multi-view inconsistency of reflections and leverage the residuals from reconstructing multi-view consistent content as cues for disentangled geometry and appearance modeling. We further propose a reflection light field that enables high-fidelity estimation of near-field reflections. During training, we introduce a high-frequency regularization to preserve fine details. We also contribute a new synthetic dataset for evaluating transmissive surface reconstruction. Experiments on both synthetic and real-world scenes demonstrate that TransmissiveGS consistently outperforms prior Gaussian Splatting-based methods in both reconstruction and rendering quality for transmissive scenes.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Accurately reconstructing complex scenes from multi-view captures with photorealistic inverse rendering has long been a central problem in computer vision and computer graphics (Figure 1). Recently, neural radiance fields (NeRF [21]), based on implicit multi-layer perceptrons (MLP), and Gaussian Splatting (GS [12]), based on explicit Gaussian primitives, have brought substantial breakthroughs. Compared to NeRF, GS leverages rasterization to enable fast training and high-fidelity rendering. Although several recent works [37, 11, 47, 35, 39, 6, 45] have extended GS to handle reflective surfaces, the modeling of transmissive surfaces remains largely unexplored.

Objects with transmissive surfaces, such as glass, are challenging because their observed appearance is a mixture of two components: the reflected radiance from the transmissive surface itself, and the transmitted radiance from surfaces behind it. As a result, the observations in captured images entangle information from two distinct geometric surfaces. Most prior work represents the scene with a single set of Gaussians, which introduces ambiguity when reconstructing transmissive surfaces. Specifically, since the transmitted component is typically multi-view consistent, whereas the reflected component is high-frequency and strongly view-dependent, existing methods tend to explain the observations by fitting the transmitted surfaces, causing the transmissive surfaces to be poorly reconstructed. This, in turn, degrades the estimation and rendering of reflections.

Meanwhile, in real-world scenarios, the reflected radiance is often dominated by high-frequency near-field illumination. Vanilla GS methods [12, 9] represent color using spherical harmonics (SH), which can only capture low-frequency view-dependent appearance. Some works [37, 11, 39, 6, 45] instead use an environment map to model global illumination. While this approach can partially address high-frequency view-dependent effects, the environment map assumption that illumination originates from infinity introduces parallax errors under near-field lighting. EnvGS [35] addresses this limitation by explicitly reconstructing the surrounding environment geometry as environment Gaussians to represent near-field illumination, followed by querying them via ray tracing. However, when surface reflections are weak, as is typical in transmissive scenes, the reconstructed environment Gaussians lack sufficient supervisory signal, leading to degraded quality. Ref-GS [47] leverages surface light fields by augmenting directional encoding with spatial features and predicting reflected color via a lightweight MLP; nevertheless, its capacity for high-frequency details remains limited.

To overcome these challenges, we present **TransmissiveGS**, which (i) effectively disentangles and reconstructs the geometries of both transmissive surfaces and the transmitted surfaces visible behind them, and (ii) faithfully models high-frequency, near-field reflected radiance along with its composition with transmitted radiance (Figure 2). To this end, we introduce a **dual-Gaussian** scene representation that decomposes the scene into reflection Gaussians, capturing reflective surfaces including the transmissive surfaces themselves, and scattering Gaussians, capturing the remaining scene content visible through or around them. A novel deferred shading function composites the rendered outputs of both Gaussian sets to render the scenes. To resolve the geometric ambiguity of transmissive surfaces, we propose a **residual-guided disentangling** strategy: we first reconstruct the multi-view consistent content via the scattering Gaussian, then exploit the resulting **color residuals** to supervise the reflection Gaussian in recovering the transmissive surface geometry. To faithfully

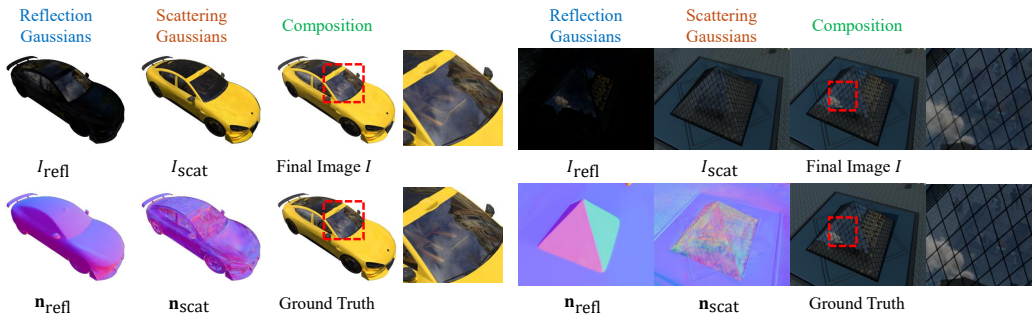

Figure 2: **Disentanglement and composite appearance.** *Left:* In the *car* case, the windshield exhibits both reflections of the surrounding environment and transmitted interior objects. *Right:* In the *pyramid* case, the glass surface reflects near-field surrounding structures while transmitting the people inside. Please see text for symbol definitions.

reproduce the high-frequency, near-field reflections observed on these surfaces, we further introduce a **reflection light field** conditioned on encoded surface position and reflection direction. Training additionally incorporates a high-frequency regularization to preserve fine details.

To summarize, our contributions are as follows:

- A dual-Gaussian scene representation with a deferred shading function that jointly models reflected and transmitted radiance and their composite appearance in transmissive scenes.
- A residual-guided reconstruction strategy that leverages multi-view fitting residuals to disentangle and recover transmissive surface geometry.
- A reflection light field that estimates high-frequency, near-field reflections, conditioned on surface position and reflection direction.

We additionally contribute a new synthetic dataset specifically designed for evaluating transmissive surface reconstruction, addressing the absence of such benchmarks in existing public datasets [20, 29]. Extensive experiments on both synthetic and real-world scenes demonstrate that TransmissiveGS consistently outperforms state-of-the-art GS-based methods.

2 Related Work

Inverse Rendering. Inverse rendering recovers physical scene properties (e.g., geometry, materials, and lighting) from observed images and re-renders the scene based on the recovered attributes [18, 1]. Early approaches build on NeRF [21]: NeRFactor [46] and PhySG [43] factorize shape, spatially-varying BRDF, and illumination; NeRO [20] and NeILF++ [42] introduce incident light fields to handle inter-reflections; and Ref-NeRF [29] improves specular reconstruction through structured view-dependent parameterizations. Despite their effectiveness, NeRF-based methods incur substantial computational overhead due to dense per-sample MLP evaluations. Gaussian Splatting (GS) [12] offers an efficient alternative through explicit primitives and differentiable rasterization. Recent works extend GS to inverse rendering along several axes: *forward shading* methods [11, 18] evaluate per-Gaussian shading before alpha-blending but suffer from specular noise; *deferred shading* approaches [39, 48] rasterize attributes into screen-space buffers for more stable optimization of glossy surfaces; *near-field illumination* methods [27, 6, 35] go beyond distant environment maps by reconstructing the surrounding light field and querying it via ray tracing; and *implicit* approaches [47, 28] predict reflected radiance via MLPs for greater flexibility. A few works [10, 47] further handle *refraction*, but target fully transparent objects without interior visible surfaces. More recent efforts on coupled *reflective* and *transmissive* scenes, such as TR-Gaussians [19] and RT-GS [41], are restricted to specific planar or semi-transparent surfaces and do not generalize to non-planar or fully transparent ones. TSGS [16] focuses only on transparent objects without reflections. Overall, inverse rendering through transmissive surfaces remains largely underexplored.

Disentanglement of Reflection and Transmission. Separating reflected and transmitted components through transmissive surfaces is a long-standing challenge in both 2D image processing [14, 17, 25, 3, 33, 15, 13, 36, 7] and 3D scene reconstruction. NeRFReN [8] pioneered this decomposition by modeling separate transmitted and reflected radiance fields, but treats reflections as a view-independent virtual image, an assumption valid only for planar reflectors. MS-NeRF [40] and NeuS-HSR [23] mitigate reflection-induced artifacts via parallel sub-space features and SDF-based auxiliary planes, respectively, yet neither imposes explicit decomposition constraints nor reconstructs the transmissive surface geometry. RA-NeRF [5] improves physical plausibility by casting reflected rays through a jointly optimized planar reflector, but remains restricted to planar geometry. Within the Gaussian Splatting paradigm, Flash-Splat [34] achieves high-quality disentanglement using unpaired flash/no-flash cues, though it requires controlled illumination that limits in-the-wild applicability. DC-Gaussian [31] uses image decomposition to separate windshield reflections and occlusions in driving scenes. GlassGaussian [2] models directional reflectance, while TR-Gaussians [19] combines mirrored Gaussians with Fresnel blending. Both methods assume planar glass geometry. In summary, existing methods either rely on planar surface assumptions or fail to accurately reconstruct the transmissive surface geometry itself. Our TransmissiveGS addresses these limitations, enabling the joint recovery of geometry, materials, and lighting for general transmissive scenes, including those with curved surfaces.

3 Preliminary

Our TransmissiveGS builds upon 2D Gaussian Splatting (2DGS [9]) as the underlying scene primitive for reconstruction and rendering. Unlike 3D Gaussian Splatting (3DGS [12]), which models a scene with volumetric Gaussian ellipsoids and is prone to multi-view inconsistency in surface normal estimation, 2DGS employs flat Gaussian elliptical disks that are intrinsically aligned with scene surfaces, thereby yielding substantially higher geometric reconstruction quality.

Each 2D Gaussian disk is parameterized by its center position $\mathbf{p}_k$, two principal tangent vectors $\mathbf{t}_u$ and $\mathbf{t}_v$ with associated scaling factors s_u and s_v, an opacity value α, and a set of spherical harmonics (SH) coefficients $\mathbf{c}$ encoding view-dependent color. A point on the disk is expressed in world space through a local tangent-plane parameterization:

$$P(u,v) = \mathbf{p}_k + s_u\mathbf{t}_u\,u + s_v\mathbf{t}_v\,v = H\,(u,\,v,\,1,\,1)^\top \tag{1}$$

where $H = \begin{bmatrix} s_u\mathbf{t}_u & s_v\mathbf{t}_v & \mathbf{0} & \mathbf{p}_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is a homogeneous transformation matrix that encodes the geometry of the disk. Given a local coordinate $\mathbf{u} = (u, v)$ on the tangent plane, the Gaussian evaluation is defined as:

$$\mathcal{G}(\mathbf{u}) = \exp\!\left(-\frac{u^2+v^2}{2}\right) \tag{2}$$

During rendering, each image pixel $\mathbf{x} = (x, y)$ is mapped to the local coordinate of a given disk via a ray–splat intersection. Letting $W \in \mathbb{R}^{4\times4}$ denote the world-to-screen transformation matrix and z the depth at the intersection point, this mapping satisfies:

$$\mathbf{x} = (xz,\ yz,\ z,\ 1)^\top = W\,H\,(u,\,v,\,1,\,1)^\top \tag{3}$$

The final pixel color is then composited via front-to-back alpha blending over the sorted splats:

$$\mathbf{c}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{c}_i\,\alpha_i\,\hat{\mathcal{G}}_i\big(\mathbf{u}(\mathbf{x})\big)\prod_{j=1}^{i-1}\Big(1-\alpha_j\,\hat{\mathcal{G}}_j\big(\mathbf{u}(\mathbf{x})\big)\Big) \tag{4}$$

where $\hat{\mathcal{G}}$ is a low-pass filtered variant of the Gaussian kernel, introduced to suppress numerical instability under highly oblique viewing angles [9].

4 TransmissiveGS

We propose TransmissiveGS to render photorealistic transmissive scenes while faithfully reconstructing their geometry. Figure 3 provides an overview of the pipeline.

4.1 Dual-GS Representation and Deferred Shading Function

When light encounters a transmissive surface, part of the incident energy is reflected while the remainder is transmitted and subsequently scattered by the interior structure. Motivated by this physical observation, we employ a *dual* Gaussian representation that disentangles the two transport modes: (i) a set of *reflection Gaussians* $\mathcal{G}_{\text{refl}}$ that model the reflective surfaces, including the transmissive surfaces themselves, and capture the high-frequency, view-dependent reflections; and (ii) a set of *scattering Gaussians* $\mathcal{G}_{\text{scat}}$ that represent the remaining scene content visible through or around the transmissive surfaces, capturing the multi-view consistent diffuse radiance. The two sets are rasterized independently, and the resulting images are merged in screen space via a physically motivated deferred shading function.

The scattering Gaussians follow the standard 2DGS parameterization described in Section 3: Rasterizing them produces a per-pixel scattering image I_{scat}. The reflection Gaussians augment this base parameterization with two additional learnable attributes: a Fresnel vector $\mathbf{f}_0 \in \mathbb{R}^3$ that controls the wavelength-dependent base reflectance, and a scalar reflectivity coefficient $r \in \mathbb{R}$ that encodes the overall proportion of reflected energy at each Gaussian.

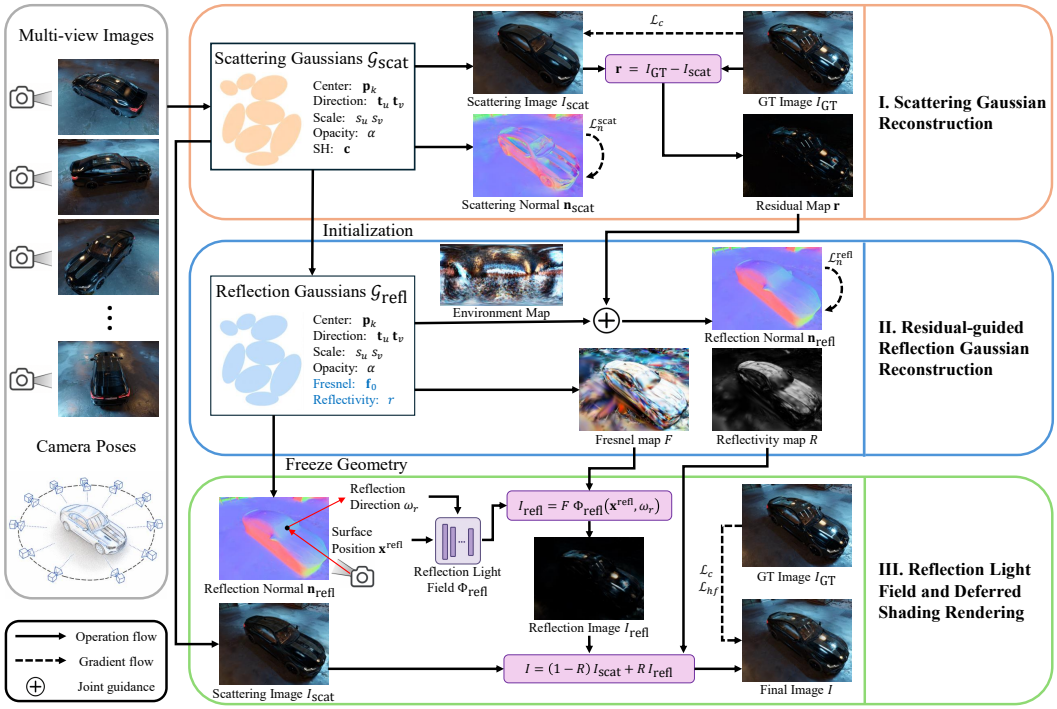

Figure 3: **Pipeline of our proposed TransmissiveGS.** Our framework consists of three stages. In **Stage I**, the scattering Gaussians are trained to reconstruct multi-view consistent scene content. In **Stage II**, the residual signal, together with an environment map, jointly supervises the reflection Gaussians to recover the geometry of reflective and transmissive surfaces. In **Stage III**, a reflection light field is trained to capture near-field, view-dependent reflections, and the final image is produced via a deferred shading composition of the scattering image and the reflection image.

After rasterization, the two components are composited on a per-pixel basis. Let R denote the reflectivity map obtained by rasterizing r. The final pixel color is given by:

$$I = \big(1 - R\big)\, I_{\text{scat}} + R\, I_{\text{refl}} \tag{5}$$

where the first term accounts for the transmitted and scattered light and the second term captures the surface reflection.

The reflection image I_{refl} is obtained by querying the reflection light field Φ_{refl} (described in Section 4.3) at the corresponding surface point $\mathbf{x}^{\text{refl}} \in \mathbb{R}^3$ on the reflection Gaussians, and modulating the returned environment radiance with a Fresnel term F to account for view-dependent reflectance:

$$I_{\text{refl}} = F\ \Phi_{\text{refl}}\big(\mathbf{x}^{\text{refl}}, \omega_r\big) \tag{6}$$

Here, the reflection direction ω_r is computed from the viewing direction ω_o (from camera to surface) and the surface normal $\mathbf{n}_{\text{refl}}$ as:

$$\omega_r = \omega_o - 2(\omega_o^\top \mathbf{n}_{\text{refl}})\, \mathbf{n}_{\text{refl}} \tag{7}$$

Assuming ideal specular reflection, the Fresnel term F is evaluated with the Schlick approximation:

$$F = F_0 + \big(1 - F_0\big)\big(1 - \mathbf{n}_{\text{refl}}^\top(-\omega_o)\big)^5 \tag{8}$$

where F_0 is a per-pixel Fresnel coefficient map obtained by rasterizing the learnable Fresnel vector $\mathbf{f}_0$ of the reflection Gaussians.

4.2 Residual-Guided Disentangling Reconstruction

As shown in Figure 3, our framework consists of three stages. In Stage I, we optimize only the scattering Gaussians $\mathcal{G}_{\text{scat}}$ from multi-view images. Because diffuse surfaces exhibit strong multi-view photometric consistency, $\mathcal{G}_{\text{scat}}$ can reconstruct them faithfully. Reflective and transmissive

surfaces, however, introduce view-dependent appearance, causing the optimizer to leave these regions unoccupied. A per-pixel photometric residual map $\mathbf{r}$ is then computed as:

$$\mathbf{r} \;=\; I_{\mathrm{GT}} - I_{\mathrm{scat}} \tag{9}$$

In Stage II, we initialize $\mathcal{G}_{\mathrm{refl}}$ from the converged $\mathcal{G}_{\mathrm{scat}}$, freeze $\mathcal{G}_{\mathrm{scat}}$, and optimize only $\mathcal{G}_{\mathrm{refl}}$. For each Gaussian disk $g_i \in \mathcal{G}_{\mathrm{refl}}$, we query a learnable environment map E along its local reflection $\omega_{r,i}$ and modulate the result by the Fresnel vector $\mathbf{f}_{0,i}$:

$$\mathbf{c}_i^{\mathrm{refl}} \;=\; \mathbf{f}_{0,i} \;\cdot\; E\big(\omega_{r,i}\big) \tag{10}$$

These per-Gaussian colors serve as the color prior for rasterizing the reflection image I_{refl}. Substituting eq. (9) into the rendering equation of eq. (5) and setting the photometric error to zero yields the optimality condition:

$$R\left(I_{\mathrm{refl}} - I_{\mathrm{scat}}\right) \;=\; \mathbf{r} \tag{11}$$

In diffuse regions where $\mathbf{r} \approx \mathbf{0}$, eq. (11) is trivially satisfied by $R \to 0$, reducing the final image to I_{scat}. In reflective regions where $\|\mathbf{r}\|$ is large, R must be positive and I_{refl} must produce correct view-dependent radiance to compensate. Notably, satisfying eq. (11) implicitly forces the reflection Gaussians to recover accurate surface geometry: because E is a radiance dictionary indexed by direction, only a correct $\omega_{r,i}$ retrieves the correct radiance; since $\omega_{o,i}$ is fixed for a given viewpoint, the only free variable controlling $\omega_{r,i}$ is the surface normal $\mathbf{n}_i$. The photometric residuals from Stage I therefore act as implicit geometric supervision, driving $\mathcal{G}_{\mathrm{refl}}$ to recover the shape of reflective and transmissive surfaces without explicit geometry annotations or priors.

4.3 Reflection Light Field

After reconstructing the geometry of the reflective surfaces, we seek to recover the reflected near-field environment illumination. To this end, we design a reflection light field, illustrated in Figure 7 in the Appendix. Unlike distant illumination that depends only on direction, near-field reflections vary with both direction and surface position. The reflection light field therefore takes the reflection Gaussian's surface position $\mathbf{x}^{\mathrm{refl}}$ and the reflection direction ω_r as input, enabling the network to implicitly triangulate the near-field environment across multiple viewpoints. The network consists of eight fully connected layers with ReLU activations and 256 hidden units per layer, followed by a sigmoid output layer. Because standard MLPs exhibit a spectral bias toward low-frequency functions [24], while specular reflections often contain high-frequency details, we first lift the inputs $\mathbf{x}^{\mathrm{refl}}$ and ω_r into a higher-dimensional space via Fourier feature encoding [26]:

$$\gamma(p) = \big(\sin(2^0\pi p),\ \cos(2^0\pi p),\ \ldots,\ \sin(2^{L-1}\pi p),\ \cos(2^{L-1}\pi p)\big) \tag{12}$$

Before encoding, we normalize $\mathbf{x}^{\mathrm{refl}}$ to $[-1, 1]$ according to the spatial extent of the reflection Gaussian geometry, and normalize ω_r to a unit vector. We set $L = 12$ for both $\gamma(\mathbf{x}^{\mathrm{refl}})$ and $\gamma(\omega_r)$; the relatively high frequency band for the directional input is chosen to provide sufficient capacity for modeling the sharp angular variation of specular reflections. In the first four layers, only the encoded surface position $\gamma(\mathbf{x}^{\mathrm{refl}})$ is processed, rather than being concatenated with the encoded reflected direction from the outset. This design allows the network to first extract position-dependent features such as spatially varying visibility and local geometric context before fusing them with directional information, thereby helping disentangle spatially varying properties from direction-dependent reflectance. At the fifth layer, the learned position features are concatenated with the encoded reflected direction $\gamma(\omega_r)$, and the encoded surface position $\gamma(\mathbf{x}^{\mathrm{refl}})$ is re-injected via a skip connection. Because reflected radiance is highly sensitive to surface location, this skip connection facilitates gradient flow and preserves fine-grained positional cues, mitigating the over-smoothing commonly observed in deep networks [22]. Finally, a sigmoid activation produces the reflected color.

During the training of the reflection light field, we freeze the geometry of the reflection Gaussians while allowing the geometry and appearance of the scattering Gaussians to remain learnable. This strategy enables the scattering Gaussians and the reflection light field to be jointly optimized, thereby achieving accurate scene reconstruction and inverse rendering of the transmissive scene.

4.4 Training

We supervise the rendered image against the ground truth with an RGB reconstruction loss that combines an $\mathcal{L}_1$ term and a D-SSIM term:

$$\mathcal{L}_c = (1-\lambda)\left\|I - I_{\mathrm{GT}}\right\|_1 + \lambda\,\mathcal{L}_{\text{D-SSIM}}\big(I,\ I_{\mathrm{GT}}\big) \tag{13}$$



Figure 4: Qualitative comparison of photorealistic rendering quality.

where $\lambda = 0.2$. To regularize the surface geometry, we adopt the normal consistency loss $\mathcal{L}_n$ from 2DGS [9] and apply it independently to both the reflection Gaussians and the scattering Gaussians:

$$\mathcal{L}_n^{(\cdot)} = \sum \left(1 - \mathbf{n}_{(\cdot)} \cdot \hat{\mathbf{n}}_{(\cdot)}\right) \tag{14}$$

where $(\cdot) \in \{\text{refl}, \text{scat}\}$, $\mathbf{n}_{(\cdot)}$ is the rendered normal map obtained by alpha-blending the normals of the individual 2D Gaussian primitives, and $\hat{\mathbf{n}}_{(\cdot)}$ is the corresponding normal map derived from the gradient of the rendered depth map via finite differences. While $\mathcal{L}_c$ is computed over the entire image, such an image-wide objective tends to match the average appearance and can smooth out high-frequency details. To preserve these details, we introduce a high-frequency loss $\mathcal{L}_{hf}$ that focuses the optimization on pixels with the largest rendering errors. Specifically, let $e(\mathbf{x})$ denote the per-pixel error computed in the same manner as $\mathcal{L}_c$, and let τ_p be the p-th percentile of $\{e(\mathbf{x})\}_{\mathbf{x}\in\Omega}$. The high-frequency loss is defined as the mean error over all pixels whose error exceeds τ_p:

$$\mathcal{L}_{hf} = \sum_{\mathbf{x}\in\Omega} \mathbb{I}(e(\mathbf{x}) > \tau_p)\, e(\mathbf{x}) \;\Big/\; \left(\sum_{\mathbf{x}\in\Omega} \mathbb{I}(e(\mathbf{x}) > \tau_p) + \epsilon\right) \tag{15}$$

where Ω is the set of valid pixels, $\mathbb{I}(\cdot)$ is the indicator function, and ϵ is a small constant for numerical stability. The overall training objective is:

$$\mathcal{L} = \mathcal{L}_c + \lambda_n^{\text{refl}}\, \mathcal{L}_n^{\text{refl}} + \lambda_n^{\text{scat}}\, \mathcal{L}_n^{\text{scat}} + \lambda_{hf}\, \mathcal{L}_{hf} \tag{16}$$

Table 1: Quantitative comparison of photorealistic rendering quality.

Scenes	Blender Synthesis Scenes							Real World Scenes			
	bookshelf	car	computer	house	garage	pyramid	dome	compact	facade	glazing	hatchback
PSNR↑											
3DGS [12]	36.21	32.84	35.48	33.55	23.56	32.81	20.80	29.52	23.88	27.70	26.50
2DGS [9]	35.76	32.56	35.25	33.08	23.23	31.95	20.28	29.43	24.44	26.60	26.43
GaussianShader [11]	28.51	16.88	22.88	17.12	17.09	19.72	13.60	28.37	22.60	27.68	25.72
Ref-Gaussian [37]	35.64	34.68	35.55	31.73	23.58	33.09	20.94	28.83	22.76	27.73	26.24
Ref-GS [47]	27.51	17.60	22.33	16.84	17.79	24.08	14.73	29.42	24.51	27.26	26.54
EnvGS [35]	34.79	32.92	35.83	33.62	23.64	31.82	20.44	27.64	24.22	28.92	23.74
TransmissiveGS	37.19	34.74	36.93	34.31	24.35	33.49	21.22	29.65	24.69	29.16	26.79
SSIM↑											
3DGS [12]	0.983	0.963	0.976	0.958	0.730	0.940	0.721	0.889	0.896	0.910	0.845
2DGS [9]	0.981	0.966	0.980	0.961	0.729	0.932	0.678	0.887	0.901	0.902	0.843
GaussianShader [11]	0.946	0.854	0.917	0.767	0.517	0.531	0.214	0.882	0.841	0.909	0.837
Ref-Gaussian [37]	0.980	0.961	0.978	0.966	0.735	0.943	0.720	0.869	0.863	0.913	0.827
Ref-GS [47]	0.944	0.865	0.916	0.787	0.562	0.588	0.245	0.887	0.899	0.916	0.845
EnvGS [35]	0.982	0.958	0.981	0.953	0.739	0.937	0.697	0.825	0.878	0.919	0.764
TransmissiveGS	0.985	0.970	0.983	0.971	0.746	0.946	0.731	0.891	0.904	0.917	0.852
LPIPS↓											
3DGS [12]	0.031	0.035	0.025	0.035	0.439	0.076	0.244	0.164	0.089	0.089	0.201
2DGS [9]	0.033	0.041	0.022	0.039	0.447	0.092	0.281	0.179	0.087	0.108	0.210
GaussianShader [11]	0.092	0.182	0.113	0.257	0.687	0.584	0.700	0.168	0.131	0.099	0.197
Ref-Gaussian [37]	0.036	0.038	0.027	0.043	0.431	0.082	0.246	0.206	0.111	0.098	0.231
Ref-GS [47]	0.094	0.182	0.123	0.256	0.602	0.551	0.707	0.169	0.088	0.091	0.196
EnvGS [35]	0.048	0.051	0.031	0.053	0.449	0.078	0.263	0.240	0.088	0.084	0.289
TransmissiveGS	0.029	0.033	0.022	0.033	0.418	0.072	0.218	0.158	0.085	0.081	0.197

5 Experiments

5.1 Datasets, Evaluation Metrics and Implementation Details

We evaluate the TransmissiveGS on both synthetic and real-world data. For synthetic experiments, we construct a new Blender dataset comprising seven scenes specifically designed for transmissive scenarios. Among them, *bookshelf*, *car*, *computer*, and *house* each contain a single transmissive object whose reflected illumination is driven by an HDRI environment map. The remaining three scenes, *garage*, *pyramid*, and *dome*, include both a transmissive object and surrounding scene geometry, where reflections arise from near-field illumination produced by nearby objects. For real-world experiments, we adopt *compact* and *hatchback* from the NeRF-Casting dataset [30], and additionally capture two new scenes (*facade* and *glazing*) that feature transmissive glass curtain walls.

We use consistent training/testing splits across all datasets. Photorealistic rendering quality is evaluated on all seven synthetic and four real-world scenes using three standard metrics: PSNR, SSIM [32], and LPIPS [44]. To further assess the geometric accuracy of the reconstructed transmissive surfaces, we additionally compute the Mean Angular Error in degrees (MAE°) [4] over the transmissive surface regions in the seven synthetic scenes.

All experiments are conducted on a single NVIDIA RTX 4090 GPU with 24 GB of VRAM. The optimizable parameters comprise those of the reflection light field, the reflection Gaussians, and the scattering Gaussians. All parameters are optimized with the Adam optimizer through differentiable splatting and gradient-based backpropagation. Before being fed into the reflection light field, surface positions are normalized according to the spatial extent of the reflection Gaussians.

5.2 Baseline Comparisons

We compare our method against current state-of-the-art Gaussian Splatting-based approaches, including GaussianShader [11], Ref-Gaussian [37], Ref-GS [47], and EnvGS [35], as well as vanilla 3DGS [12] and 2DGS [9]. We train all baselines using their publicly available code and configurations.

The quantitative and qualitative results on photorealistic rendering quality are presented in Table 1 and Figure 4. As shown in Table 1, our TransmissiveGS achieves a clear improvement over all baselines. The qualitative comparisons in Figure 4 further corroborate this advance. On the one hand, our method faithfully reconstructs high-frequency near-field reflections with sharp appearance details, such as the reflected surrounding structures in *pyramid*, the reflected trees in *glazing*, and the reflected neighboring building on the car hood in *hatchback*, as well as local inter-reflections (e.g.,

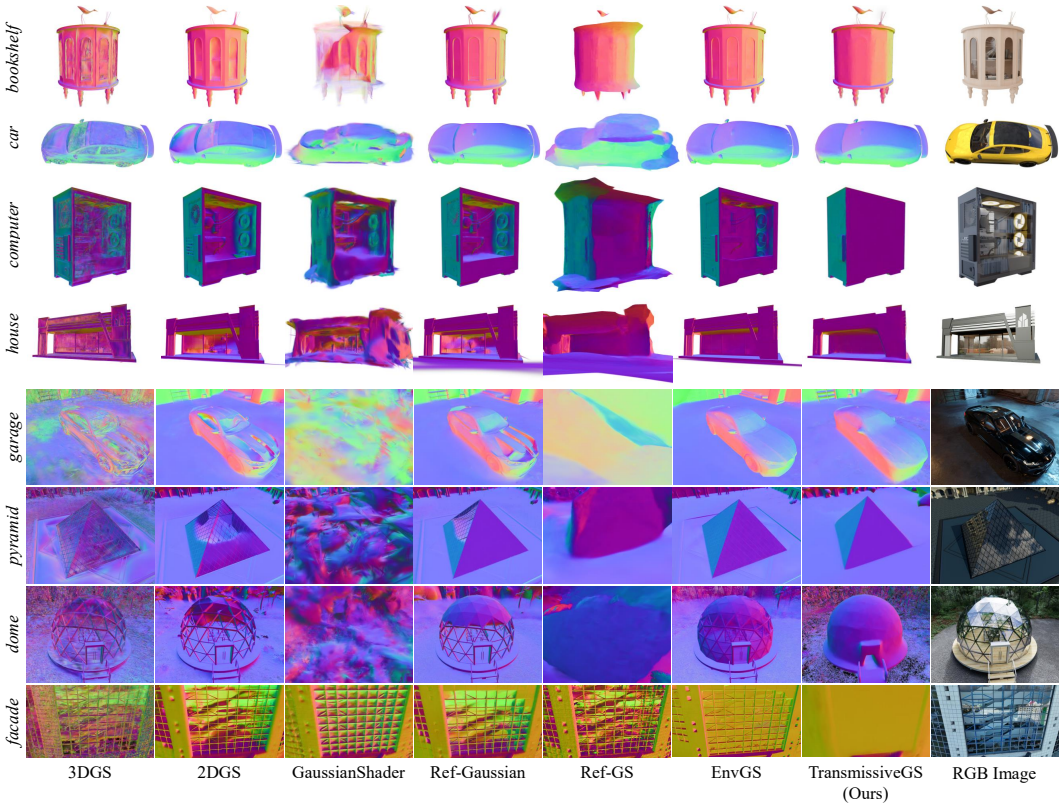

Figure 5: Qualitative comparison of transmissive surface reconstruction.

Table 2: Quantitative comparison of transmissive surface reconstruction (MAE°).

Scenes	bookshelf	car	computer	house	garage	pyramid	dome	Average
3DGS [12]	56.41	41.45	48.80	46.46	42.15	47.35	51.64	47.75
2DGS [9]	37.59	19.98	43.89	59.61	33.93	33.41	56.94	40.76
GaussianShader [11]	44.43	30.96	42.51	55.08	45.29	45.03	47.61	44.42
Ref-Gaussian [37]	13.39	5.91	42.09	44.05	28.76	17.57	39.89	27.38
Ref-GS [47]	8.10	23.95	22.83	20.52	49.63	42.42	30.59	28.29
TransmissiveGS	7.38	5.50	4.57	12.72	7.01	6.53	8.50	7.46

the columns of the house itself reflected on the glass in *house*). On the other hand, TransmissiveGS simultaneously preserves the transmitted content behind the transparent surface with high fidelity: in *compact*, both the window reflections and the interior details (e.g., the seat and steering wheel) remain clearly visible; in *hatchback*, the windshield reflections and the interior objects coexist without mutual degradation.

In addition, the quantitative and qualitative results of transmissive surface reconstruction are shown in Table 2 and Figure 5. Since EnvGS [35] leverages StableNormal [38] as a geometric prior, we exclude it from the quantitative comparison for fairness. As shown in Table 2, TransmissiveGS substantially outperforms all methods across every scene. The qualitative results in Figure 5 further demonstrate that our method can faithfully recover the geometry of fully transparent transmissive surfaces. In contrast, although EnvGS incorporates an external geometric prior, it still fails to reconstruct the transmissive surface geometry in several challenging cases such as *computer*, *dome*, and *facade*.

5.3 Ablation Studies

We conduct ablation studies on the *facade* scene to validate the effectiveness and necessity of each design component. The results are summarized in Table 3 and Figure 6.

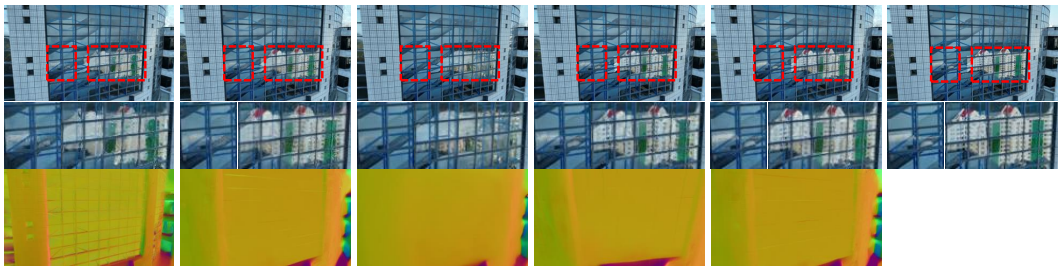

Figure 6: Qualitative results of ablation studies.

The **"w/o Dual-Gaussian"** variant removes the scattering Gaussians and performs training using only the reflection Gaussians. As shown in Figure 6, this variant fails to reconstruct the transmissive surface geometry, which in turn degrades the estimation of the reflection light field.

The **"w/o Reflection MLP"** variant replaces the reflection light field in Section 4.3 with an environment map-based reflection formulation. This variant struggles to reconstruct sharp, high-frequency near-field reflections and introduces parallax errors.

The **"w/o Decoupling"** variant removes the attributes freezing of scattering Gaussians during Stage II and geometry freezing of reflection Gaussians during Stage III. Without this decoupled optimization strategy, the two Gaussian sets compete with each other during joint optimization, leading to degraded transmissive surface geometry and inferior reflection estimation.

The **"w/o High-frequency Loss"** variant removes the high-frequency loss term in Section 4.4 during optimization. As a result, the fine details of the reconstructed reflections are degraded; e.g., the small reflected structures on the left side of the glass become blurred.

Table 3: Quantitative results of ablation studies.

	PSNR↑	**SSIM↑**	**LPIPS↓**
w/o Dual-Gaussian	22.98	0.827	0.152
w/o Reflection MLP	24.20	0.858	0.124
w/o Decoupling	23.23	0.878	0.100
w/o High-frequency Loss	24.52	0.891	0.089
Ours	24.69	0.904	0.085

6 Conclusion

We have presented **TransmissiveGS**, a novel Gaussian Splatting framework for accurate reconstruction and photorealistic rendering of transmissive scenes. Our key contributions include: (i) a dual-Gaussian representation that disentangles reflected and transmitted radiance; (ii) a residual-guided disentanglement strategy that enables effective reconstruction of transmissive surface geometry; and (iii) a reflection light field formulation that faithfully captures high-frequency, near-field reflections. Extensive experiments demonstrate that TransmissiveGS significantly outperforms existing methods in both inverse rendering quality and transmissive surface reconstruction accuracy. A current limitation is that our method assumes both the reflected and transmitted scene content to be static. In future work, we plan to extend TransmissiveGS to handle dynamic reflections and transmissions.

Acknowledgments and Disclosure of Funding

The authors would like to acknowledge the support by RGC Theme-based Research Schemes (2023/24 T22-606/23-R), RGC Theme-based Research Schemes (2024/25 T22-607/24-N) and Strategic Public Policy Research Funding Scheme (S2024.A7.022). The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

A Architecture of Reflection Light Field.

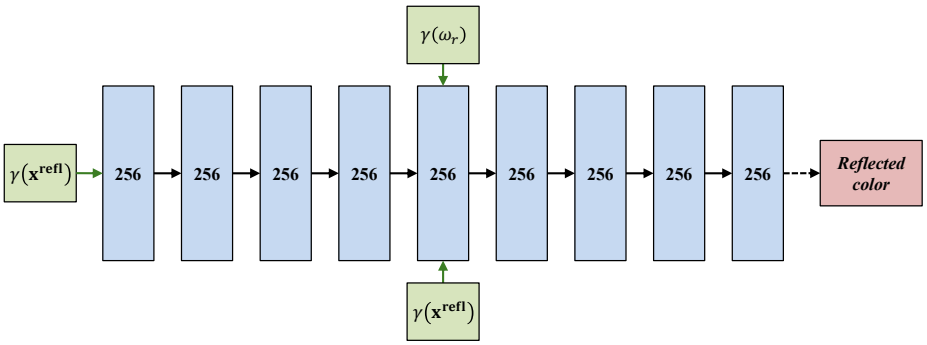

Figure 7: **Architecture of reflection light field.** Green blocks denote input features, blue blocks denote hidden layers, and the red block denotes the output. Solid green arrows indicate data flow. Solid black arrows indicate ReLU activations, and dashed black arrows indicate a sigmoid activation.

B Additional Ablation Studies on Reflection Light Field.

We further conduct ablation studies on the reflection light field to validate the design choices of its architecture. We investigate the following factors that affect its performance: (1) the network width *hidden dim*; (2) the number of layers *depth*; (3) the order of Fourier feature encoding L; (4) whether to re-inject the encoded surface position at intermediate layers via a *skip connection*; and (5) whether to input the encoded surface position and encoded reflected direction separately using *staged fusion*. We perform these architecture ablations on the *garage* case, and present the results in Table 4.

The results show that removing staged fusion causes a noticeable drop in performance, suggesting that it is beneficial to first let the reflection light field learn position-dependent characteristics and then infer direction-dependent reflections. Incorporating a skip connection consistently improves reflection estimation. Reducing either the network depth or width degrades performance. For the Fourier feature encoding, a sufficiently large order is required for the MLP to model high-frequency reflections; however, overly large orders tend to introduce high-frequency noise and can hurt performance. Overall, these ablation studies confirm that our reflection light field design is effective and reasonable.

Table 4: Ablation study on the reflection light field.

	PSNR↑	SSIM↑	LPIPS↓
hidden dim 128, depth 8, L 12, w/ skip connection, w/ staged fusion	24.26	0.744	0.424
hidden dim 256, depth 8, L 10, w/ skip connection, w/ staged fusion	24.24	0.742	0.425
hidden dim 256, depth 8, L 14, w/ skip connection, w/ staged fusion	24.26	0.743	0.423
hidden dim 256, depth 6, L 12, w/ skip connection, w/ staged fusion	24.05	0.741	0.420
hidden dim 256, depth 8, L 12, w/o skip connection, w/ staged fusion	24.17	0.742	0.421
hidden dim 256, depth 8, L 12, w/ skip connection, w/o staged fusion	23.96	0.739	0.424
(Ours) hidden dim 256, depth 8, L 12, w/ skip connection, w/ staged fusion	24.35	0.746	0.418

C Failure Mode

Figure 8 illustrates a failure case of our method. When the transmissive surface is dominated by transmission with only faint reflections at certain viewpoints, the reflection component at these regions can be difficult to estimate accurately and may degenerate.



Ground Truth Rendering

Figure 8: Failure mode of our method.

References

[1] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021.

[2] Junming Cao, Jiadi Cui, and Sören Schwertfeger. Glassgaussian: extending 3d gaussian splatting for realistic imperfections and glass materials. *IEEE Access*, 13:31517–31531, 2025.

[3] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE international conference on computer vision*, pages 3238–3247, 2017.

[4] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3392–3399, 2013.

[5] Chen Gao, Yipeng Wang, Changil Kim, Jia-Bin Huang, and Johannes Kopf. Planar reflection-aware neural radiance fields. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024.

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

[7] Xiaojie Guo, Xiaochun Cao, and Yi Ma. Robust separation of reflection from multiple images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2187–2194, 2014.

[8] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022.

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

[10] 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 Transactions on Graphics (TOG)*, 44(4):1–17, 2025.

[11] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5322–5332, 2024.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[13] Chenyang Lei, Xuhua Huang, Mengdi Zhang, Qiong Yan, Wenxiu Sun, and Qifeng Chen. Polarized reflection removal with perfect alignment in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1750–1758, 2020.

[14] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1647–1654, 2007.

[15] Chao Li, Yixiao Yang, Kun He, Stephen Lin, and John E Hopcroft. Single image reflection removal through cascaded refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3565–3574, 2020.

[16] 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.

[17] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2752–2759, 2014.

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

[19] Yong Liu, Keyang Ye, Tianjia Shao, and Kun Zhou. Tr-gaussians: High-fidelity real-time rendering of planar transmission and reflection with 3d gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics*, 32(7):5882–5894, 2026.

[20] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Transactions on Graphics (ToG)*, 42(4):1–22, 2023.

[21] 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*, pages 405–421. Springer, 2020.

[22] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[23] Jiaxiong Qiu, Peng-Tao Jiang, Yifan Zhu, Ze-Xin Yin, Ming-Ming Cheng, and Bo Ren. Looking through the glass: Neural surface reconstruction against high specular reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20823–20833, 2023.

[24] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

[25] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T Freeman. Reflection removal using ghosting cues. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3193–3201, 2015.

[26] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[27] Jiajun Tang, Fan Fei, Zhihao Li, Xiao Tang, Shiyong Liu, Youyu Chen, Binxiao Huang, Zhenyu Chen, Xiaofei Wu, and Boxin Shi. Spectre-gs: Modeling highly specular surfaces with reflected nearby objects by tracing rays in 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16133–16142, 2025.

[28] Zhe Jun Tang and Tat-Jen Cham. 3igs: Factorised tensorial illumination for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 143–159. Springer, 2024.

[29] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):9426–9437, 2025.

[30] Dor Verbin, Pratul P Srinivasan, Peter Hedman, Ben Mildenhall, Benjamin Attal, Richard Szeliski, and Jonathan T Barron. Nerf-casting: Improved view-dependent appearance with consistent reflections. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024.

[31] Linhan Wang, Kai Cheng, Shuo Lei, Shengkun Wang, Wei Yin, Chenyang Lei, Xiaoxiao Long, and Chang-Tien Lu. Dc-gaussian: Improving 3d gaussian splatting for reflective dash cam videos. *Advances in Neural Information Processing Systems*, 37:99898–99920, 2024.

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

[33] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8178–8187, 2019.

[34] Mingyang Xie, Haoming Cai, Sachin Shah, Yiran Xu, Brandon Y Feng, Jia-Bin Huang, and Christopher A Metzler. Flash-splat: 3d reflection removal with flash cues and gaussian splats. In *European Conference on Computer Vision*, pages 122–139. Springer, 2024.

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

[36] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015.

[37] Yuxuan Yao, Zixuan Zeng, Chun Gu, Xiatian Zhu, and Li Zhang. Reflective gaussian splatting. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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

[40] Ze-Xin Yin, Jiaxiong Qiu, Ming-Ming Cheng, and Bo Ren. Multi-space neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12407–12416, 2023.

[41] Kunnong Zeng, Chensheng Peng, Yichen Xie, Masayoshi Tomizuka, and Cem Yuksel. Rt-gs: Gaussian splatting with reflection and transmittance primitives. *arXiv preprint arXiv:2604.00509*, 2026.

[42] Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf++: Inter-reflectable light fields for geometry and material estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3601–3610, 2023.

[43] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5453–5462, 2021.

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[45] Wenyuan Zhang, Jimin Tang, Weiqi Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Materialrefgs: Reflective gaussian splatting with multi-view consistent material inference. *Advances in Neural Information Processing Systems*, 38:89896–89917, 2025.

[46] 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 (ToG)*, 40(6):1–18, 2021.

[47] Youjia Zhang, Anpei Chen, Yumin Wan, Zikai Song, Junqing Yu, Yawei Luo, and Wei Yang. Ref-gs: Directional factorization for 2d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26483–26492, 2025.

[48] Yongyang Zhou, Fanglue Zhang, Zichen Wang, and Lei Zhang. Rtr-gs: 3d gaussian splatting for inverse rendering with radiance transfer and reflection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 6888–6897, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction (section 1), we state that the key contribution of this paper is the proposal of TransmissiveGS, which achieves high-fidelity reconstruction and rendering of transmissive scenes.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the conclusion (Section 6), we state that the limitation and basic assumption of this paper are based on static reflections and transmissions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: In the methodology sections (Sections 3 and 4), we provide all the theorems, formulas, and proofs underlying the results presented in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: In Section 4, we describe the framework of TransmissiveGS in detail, and in Section 5, we provide detailed information on the datasets, evaluation metrics, and implementation details.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code and the datasets used in this work in the supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 5, we provide detailed information on the datasets, evaluation metrics, and implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes],

   Justification: In Section 5, we report the quantitative results and the statistical significance of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 5, we state that all experiments are conducted on a single NVIDIA RTX 4090 GPU with 24 GB of VRAM.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully read and strictly adhered to the NeurIPS Code of Ethics throughout this research.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Section 1, we emphasize that transmissive scenes are ubiquitous in everyday life, yet existing methods fail to effectively reconstruct and render such scenes. Our research has the potential to deliver significant societal impact and practical value.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The methods and data presented in this paper pose no high risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: In Section 3, we cite the 2DGS model employed in our method, and in Section 5, we cite the public dataset of NeRF-Casting used in our experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new model and the newly provided dataset introduced in this paper have been well documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.