RT-GS: GAUSSIAN SPLATTING WITH REFLECTION AND TRANSMITTANCE PRIMITIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Gaussian Splatting is a powerful tool for reconstructing diffuse scenes, but it struggles to simultaneously model specular reflections and the appearance of objects behind semi-transparent surfaces. These specular reflections and transmittance are essential for realistic novel view synthesis, and existing methods do not properly incorporate the underlying physical processes to simulate them. To address this issue, we propose RT-GS, a unified framework that integrates a microfacet material model and ray tracing to jointly model specular reflection and transmittance in Gaussian Splatting. We accomplish this by using separate Gaussian primitives for reflections and transmittance, which allow modeling distant reflections and reconstructing objects behind transparent surfaces concurrently. We utilize a differentiable ray tracing framework to obtain the specular reflection and transmittance appearance. Our experiments demonstrate that our method successfully produces reflections and recovers objects behind transparent surfaces in complex environments, achieving significant qualitative improvements over prior methods where these specular light interactions are prominent.

1 INTRODUCTION

Reconstructing 3D scenes from multi-view input 2D images and rendering images for novel viewpoints is a fundamental problem in novel view synthesis. Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) achieve good reconstruction quality by leveraging the expressive power of neural networks. However, the high computational cost of NeRFs makes it difficult to achieve real-time performance. Recently, 3D Gaussian Splatting (3D-GS) (Kerbl et al., 2023) has demonstrated fast reconstruction and rendering. However, 3D-GS primarily relies on rasterization, which is well-suited for diffuse appearances but fundamentally limited in modeling multi-bounce light transport and view-dependent specular effects such as reflections.

Several recent works extend GS with additional mechanisms for non-diffuse phenomena. EnvGS (Xie et al., 2025) utilizes ray tracing on separate environment Gaussian primitives to model specular reflection. While it can synthesize near-field reflections, it still struggles to reconstruct complex reflections and dielectric surfaces due to its simplified rendering functions. Moreover, it does not consider transmittance of semi-transparent surfaces. TransparentGS (Huang et al., 2025) integrates transparent Gaussian primitives to model transparent objects. However, it bakes the environment around the transparent objects to generate refraction effects and fails to reconstruct objects behind semi-transparent surfaces.

In this paper, we present a novel framework to further improve the quality of specular reflection and transmittance appearance simultaneously in GS-based methods. We utilize the microfacet model to represent complex material properties on surfaces. We reconstruct the 3D scenes using three separate Gaussian primitives: diffuse Gaussian, reflection Gaussian, and transmittance Gaussian. We start with the rasterization process using diffuse Gaussians to obtain the per-pixel diffuse color, surface position, normal map, roughness map, and reflectance map. The microfacet model is integrated to calculate the bidirectional scattering distribution function (BSDF) on the surfaces of diffuse Gaussians. Then, we generate reflection and transmittance rays based on the normal map at surface positions, and utilize differentiable ray tracing on reflection and transmittance Gaussians, respectively. We assume that the outer layer of transparent objects is infinitely thin, so transmittance rays maintain the direction of incoming camera rays. The specular reflection color is obtained by

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

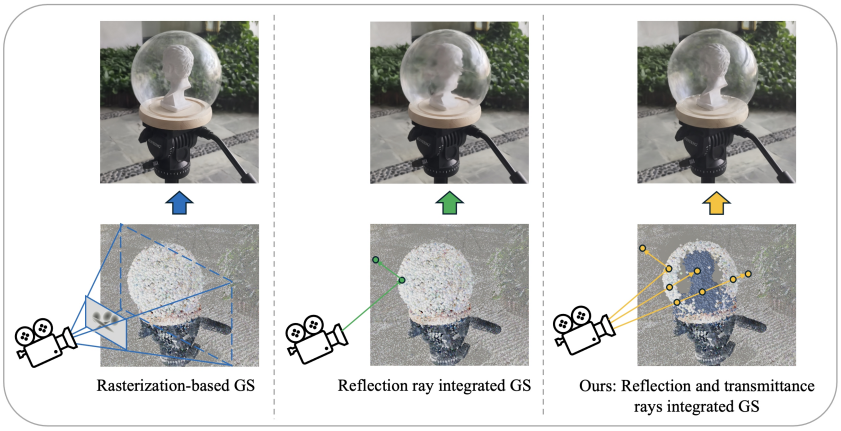


Figure 1: We present a method that optimizes the quality of specular reflection and transmittance concurrently in GS-based method. For the rasterization-based gaussian splatting method (Kerbl et al., 2023) in the left image, the specular reflection details are blurred on the surface of objects. For the reflection rays integrated method (Xie et al., 2025) in the middle image, the object behind the semi-transparent surface is blurred. We solve both the complex reflection and transmittance ray effects at the same time by integrating reflection and transmittance rays into gaussian splatting. Then, we can see the transmittance appearance and specular reflection details, as shown in the right image.

ray tracing on reflection Gaussian. The objects behind transparent surfaces are reconstructed by transmittance Gaussians. Transmittance rays exiting from the back of transparent objects continue to intersect with the diffuse Gaussians to reconstruct the light paths of seeing through transparent objects. Finally, we blend diffuse, reflection, and transmittance colors using the BSDF to obtain the final rendering results.

To train these components jointly from input 2D images, we first generate segmentation masks for transparent objects using SAM2 (Kirillov et al., 2023; Ravi et al., 2025) guided by GroundingDINO (Liu et al., 2024). Reflection and transmittance rays are traced simultaneously on the surface of transparent objects, while only reflection rays are traced elsewhere. We extract meshes of transparent objects to guide the transmittance rays and limit the transmittance Gaussians inside transparent objects. We add a regularization term on specular strength (Zhang et al., 2026) to force the reflection and transmittance colors to have a larger percentage on transparent objects. In summary, the technical contributions in this paper are: (i) We couple microfacet-weighted deferred shading with differentiable ray tracing over dedicated reflection and transmittance Gaussians in one full-scene GS-based pipeline. (ii) We introduce separate transmittance Gaussian primitives to model the interiors of transparent objects. (iii) We extract the mesh to guide the ray tracing on transmittance Gaussian and use masks of transparent objects to guide the training of material properties.

2 Related Work

The target of novel view synthesis is to generate images from new viewpoints of a scene based on a limited set of posed images. Neural radiance fields (NeRF) (Mildenhall et al., 2021) adopt implicit multilayer perceptrons (MLPs) to reconstruct 3D scenes and leverage differentiable volume rendering to obtain rendering results, achieving a high quality of appearance in novel views. Subsequent methods have further enhanced rendering quality (Verbin et al., 2024; Sun et al., 2024) and accelerated both training and rendering speed (Müller et al., 2022).

Recently, 3D Gaussian Splatting (3D-GS) (Kerbl et al., 2023) achieves real-time rendering and superior rendering quality over NeRF-based methods by rasterizing 3D Gaussian primitives. 2D Gaussian Splatting (2D-GS) (Huang et al., 2024) flattens 3D Gaussians into 2D Gaussian primitives to achieve better surface alignment. In our work, we adopt 2D Gaussian primitives as the scene representation to reconstruct better surface geometry and normals.

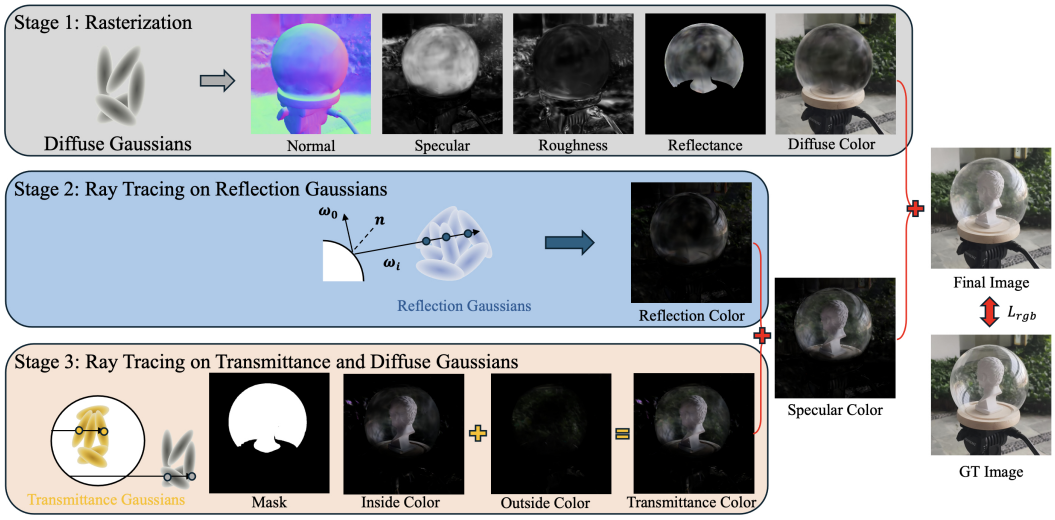


Figure 2: **Overview of our method.** In the first stage, we start with the rasterization process using diffuse Gaussians to obtain per-pixel material maps and diffuse color. In the second stage, we generate specular reflection rays from the surface of diffuse Gaussians and utilize ray tracing on reflection Gaussians. In the third stage, we trace first-bounce transmittance rays from the surface into the transparent objects on transmittance Gaussians to obtain the inside color and second-bounce transmittance rays from the back surface of the mesh on diffuse Gaussian again to obtain the outside color. Finally, we combine the diffuse, reflection and transmittance colors for the final image. We jointly optimize these three types of Gaussians using ground truth images for supervision.

3D-GS and 2D-GS struggle to accurately model specular reflection and transmittance concurrently due to their use of spherical harmonics (SH) to reconstruct appearance, which causes blurry results. Recent approaches (Jiang et al., 2024; Ye et al., 2024b) use the environment map to model specular reflection, but they fail to model near-field reflections and transmittance. Other methods (Yao et al., 2025; Zhang et al., 2026) integrate the PBR materials and deferred shading to achieve better physical properties of the surface. However, they only focus on specular reflections and fail to model the transmittance appearance. EnvGS (Xie et al., 2025) integrates ray tracing on Gaussians to achieve better specular reflection, but it also fails to model transmittance. TransparentGS (Huang et al., 2025) models transparent objects as separate Gaussian primitives and bakes the environment for deferred shading. However, it ignores objects behind the transparent surfaces. RT-Splatting (Shi et al., 2026) disentangles each Gaussian's geometric occupancy from its optical opacity to model semi-transparent surfaces, but it also fails to reconstruct specular details well. Our method integrates the microfacet model without the split-sum approximation to reconstruct surface properties. We adopt ray tracing to model specular effects and reconstruct objects behind transparent surfaces.

3 METHOD

3D-GS (Kerbl et al., 2023) represents a scene using differentiable 3D Gaussian primitives defined by a covariance matrix $\sum$ and a position μ in world space

$$G(x) = e^{-\frac{1}{2}(x)^\top \Sigma^{-1}(x)} \ . \tag{1}$$

During the rendering process, the 3D Gaussians are projected to 2D splats using the transformation matrix from world space to image space. The rendering color C is computed by α-blending the 2D splats in front-to-back order, such that

$$C = \sum_{i=1}^{N} C_i o_i \prod_{j=1}^{i-1}(1 - o_j) \ , \tag{2}$$

where o_i is the opacity and C_i is the view-dependent color represented by spherical harmonics of each Gaussian.

2D-GS (Huang et al., 2024) uses 2D Gaussian primitives to represent a scene and provide better surface reconstruction. The 2D Gaussian is defined in a local tangent plane in world space with two principal tangential vectors $\vec{t}_u$ and $\vec{t}_v$ and a scaling vector (s_u, s_v). Because the specular reflection and transmittance require accurate surface representation, we adopt the 2D Gaussian as our base primitive. However, 2D-GS integrates spherical harmonics to model scene appearance, which introduces a blurry appearance for highly view-dependent effects, including specular reflection and transmittance. To address this, we introduce separate Gaussian primitives to model specular effects.

3.1 Modeling Specular Light Transport

To reconstruct scenes with both specular reflection and transmittance, we separate the Gaussian primitives into diffuse, reflection, and transmittance Gaussian primitives. The diffuse and transmittance Gaussians are both initialized using the sparse point cloud obtained from Structure-from-Motion (SfM) (He et al., 2024; Snavely et al., 2006). The reflection Gaussian is randomly initialized within the scene's bounding box (Xie et al., 2025).

Each Gaussian is integrated into depth d, normal $\vec{n}$, roughness α, base reflectance f_0 and specular blending weight k_s properties. How we use these properties for deferred shading is demonstrated in subsection 3.2. We first use 2D Gaussian splatting to render the diffuse Gaussian. These properties of the diffuse Gaussians are α blended by rasterization to generate per-pixel material maps for deferred shading:

$$x = \sum_{i=1}^{N} x_i o_i \prod_{j=1}^{i-1}(1 - o_j),\ x \in \{d, \vec{n}, \alpha, f_0, k_s\}\ , \tag{3}$$

where o_i is the opacity of the i-th Gaussian primitive.

After we obtain these material maps, we compute surface positions based on the depth map. Then, the reflection direction can be calculated based on the surface position and normal map. The transmittance direction is taken as the direction of the incoming camera rays, assuming infinitely thin semi-transparent surfaces. The reflection and transmittance Gaussians are rendered by a differentiable ray tracer (Xie et al., 2025) with the reflection and transmittance rays to obtain the reflection color C_r and transmittance color C_t, respectively. Based on the diffuse color C_d obtained through rasterization on the diffuse Gaussian, we can derive the final color using

$$C = (1 - k_s)C_d + k_s(f_r C_r + f_t C_t)\ . \tag{4}$$

where k_s is the blending weight for the specular term, f_r is the bidirectional reflectance distribution function (BRDF) and f_t is the bidirectional transmittance distribution function (BTDF).

3.2 Microfacet Material Model

To model dielectric surfaces, we incorporate a microfacet material. We divide the rendering color C at the surface position $\vec{x}$ in the viewing direction $\vec{\omega}_o$ into two parts: the diffuse term and the specular term (Burley & Studios, 2012):

$$C(\vec{x}, \vec{\omega}_o) = (1 - k_s)C_d(\vec{x}) + k_s L_s(x, \omega_o) \tag{5}$$

where L_s is the outgoing specular radiance.

We adopt the rendering equation (Kajiya, 1986) to calculate outgoing specular radiance L_s at the surface position $\vec{x}$ in the direction $\vec{\omega}_o$:

$$L_s(\vec{x}, \vec{\omega}_o) = \int_{\Omega} L_i(\vec{x}, \vec{\omega}_i) f_s(\vec{x}, \vec{\omega}_i, \vec{\omega}_o)(\omega_i \cdot n)\mathrm{d}\vec{\omega}_i \tag{6}$$

where L_i is the incident radiance from the upper hemisphere, $\vec{\omega}_i$ is the incidence direction and $\vec{n}$ is the surface normal. f_s is the bidirectional scattering distribution function (BSDF), which can be composed by the BRDF f_r and the BTDF f_t, such that

$$f_s(\vec{x}, \vec{\omega}_i, \vec{\omega}_o) = f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_o) + f_t(\vec{x}, \vec{\omega}_i, \vec{\omega}_o) \tag{7}$$

For the BRDF f_r, we adopt the Torrance-Sparrow BRDF (Cook & Torrance, 1982)

$$f_r(x, \omega_i, \omega_o) = \frac{DGF}{4(\vec{\omega}_o \cdot \vec{n})(\vec{\omega}_i \cdot \vec{n})} \tag{8}$$

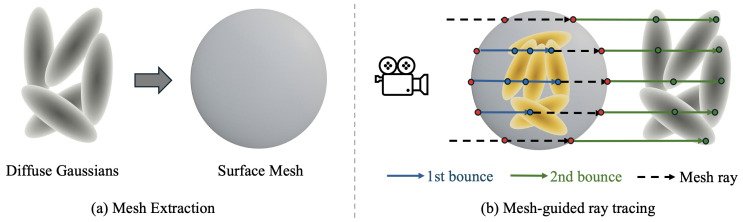


Figure 3: **Mesh-guided ray tracing for transparent objects.** The surface mesh is extracted from the diffuse Gaussian. The first bounces of transmittance rays are traced from the depth positions of the diffuse Gaussian on the transmittance Gaussian. The second bounces of transmittance rays are traced from the back surface positions of the extracted mesh on the diffuse Gaussian. To obtain the back surface positions, we trace rays on the mesh to get the second nearest depth.

where D, G, F denote the normal distribution function, the shadowing-masking term and the Fresnel term, respectively.

For the normal distribution function, we use the isotropic Trowbridge–Reitz model (Trowbridge & Reitz, 1975)

$$D(\vec{n}, \vec{\omega}_h, \alpha) = \frac{\alpha^2}{\pi\Big((\vec{n} \cdot \vec{\omega}_h)^2(\alpha^2 - 1) + 1\Big)^2} \tag{9}$$

where $\vec{\omega}_h$ is the half vector between the incidence direction $\vec{\omega}_i$ and the outgoing direction $\vec{\omega}_o$ such that $\vec{\omega}_h = (\vec{\omega}_i + \vec{\omega}_o)/\|\vec{\omega}_i + \vec{\omega}_o\|$, and α is the roughness.

When α approaches zero, it indicates near-perfect specular reflection. During training, the roughness approaching zero causes the normal distribution term to become excessively large, thereby adversely affecting the quality of training. Therefore, we apply a mapping from such roughness values to input α values (Pharr et al., 2016):

$$g(\alpha) = 1.62142 + 0.819955 \ln \alpha + 0.1734(\ln \alpha)^2 + 0.0171201(\ln \alpha)^3 + 0.000640711(\ln \alpha)^4 \tag{10}$$

For the BTDF f_t, we consider specular transmission:

$$f_t(\vec{x}, \vec{\omega}_i, \vec{\omega}_o) = (1 - F)\frac{\delta(\vec{\omega}_i - \vec{\omega}_t)}{(\vec{\omega}_i \cdot \vec{n})} \tag{11}$$

where $\vec{\omega}_t$ is the specular transmission direction and δ is the Dirac delta function.

3.3 Optimizations

We employ a multi-stage reconstruction process for these three different Gaussian primitives. Figure 2 illustrates the overview of our rendering pipeline.

In the first stage, we obtain the material maps $\{d, \vec{n}, \alpha, f_0, k_s\}$ and the diffuse color C_d based on the diffuse Gaussian.

In the second stage, we add specular reflection rays starting from the surface positions of the diffuse Gaussian to obtain the reflection color C_r from the reflection Gaussian. We adopt the joint optimization of the diffuse Gaussian and the reflection Gaussian to obtain better material quality and extract high-quality surface meshes.

In the final stage, after obtaining high-quality surface positions and normals, we first extract the surface mesh from the diffuse Gaussian. Then, we trace the transmittance rays guided by the extracted mesh to obtain the transmittance color C_t and jointly optimize these three Gaussian primitives. Figure 3 illustrates the process of mesh-guided ray tracing for transparent objects. Specifically, the first bounce of transmittance rays starts from the surface positions of the diffuse Gaussian and is traced on the transmittance Gaussian. To obtain the color of the external environment as seen through transparent objects, we need to trace rays from the back surface of transparent objects on the diffuse

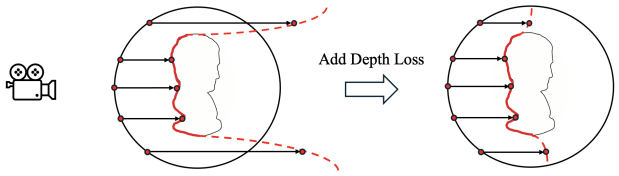


Figure 4: **Illustration of the regularization to the depth.** When reconstructing the internal object, the depth of first-bounce transmittance rays of the part of the object that exists is reconstructed well, as demonstrated by the solid red curve. But in the parts where no object is existing, the depth is trained to the outside the transparent object, as shown by the red dashes curve, because it tries to reconstruct the outside color, which should be rendered by second-bounce transmittance rays. Therefore, we add regularization to the depth to push it back into the transparent object.

Gaussian again, because the diffuse Gaussians have already been well trained. However, the back surface positions cannot be obtained by the rasterization of the diffuse Gaussian, as we can only get the positions of the nearest depth. Therefore, we trace rays from cameras on the extracted surface mesh to obtain the nearest depth and second nearest depth. The second bounce of transmittance rays starts from the second-nearest surface positions and is traced on the diffuse Gaussians again.

To add transmittance rays only to the parts of transparent objects, we generate segmentation masks for transparent objects using SAM2 (Kirillov et al., 2023; Ravi et al., 2025) guided by GroundingDINO (Liu et al., 2024). Based on the observation that surfaces on transparent objects have a high specular strength, we add regularization to the specular map to encourage the specular colors C_r and C_t to dominate the final color for transparent objects. Inspired by MaterialRefGS (Zhang et al., 2026), the masks are used to add the constraint to the specular map k_s at a high target value, such that

$$\mathcal{L}_{\text{spec}} = \frac{1}{N_p}\sum_{i=1}^{N_p} \Gamma_i \cdot \text{ReLU}(K_0 - k_{s,i}) \ , \tag{12}$$

where N_p is the number of pixels in the image, Γ_i is the binary mask for the transparent object at pixel i, K_0 is the target value, and ReLU is the rectified linear unit (Nair & Hinton, 2010).

The transmittance Gaussian is used to reconstruct the objects inside transparent objects, so it needs to be constrained within transparent objects. As illustrated in Figure 4, we add regularization to the depth map d obtained by the first bounce of transmittance rays, such that

$$\mathcal{L}_{d} = \frac{1}{N_p}\sum_{i=1}^{N_p} \Gamma_i \cdot \text{ReLU}(d_i - D_i) \ , \tag{13}$$

where D_i is the second-nearest depth of the diffuse Gaussian at the i-th pixel, specifically on the back surface of transparent objects.

We adopt the constraint for aligning the normal map values $\vec{n}_i$ with the gradients of the depth maps $\vec{N}_d$ (Huang et al., 2024)

$$\mathcal{L}_{\text{norm}} = \frac{1}{N_p}\sum_{i=1}^{N_p} (1 - \vec{n}_i \cdot \vec{N}_d) \ , \tag{14}$$

where $\vec{N}_d$ is computed with finite differences from nearby pixel positions $\vec{p}_d$ in the depth map, using

$$\vec{N}_d = \frac{\nabla_x \vec{p}_d \times \nabla_y \vec{p}_d}{\|\nabla_x \vec{p}_d \times \nabla_y \vec{p}_d\|} \ . \tag{15}$$

We follow EnvGS (Xie et al., 2025) to use monocular normal estimates $\vec{N}_m$ by StableNormal (Ye et al., 2024a) to supervise the normal map values $\vec{n}_i$ and perceptual loss (Xie et al., 2025) to enhance the perceived quality of the rendered image:

$$\mathcal{L}_{\text{mono}} = \frac{1}{N_p}\sum_{i=1}^{N_p} (1 - \vec{n}_i \cdot \vec{N}_m) \tag{16}$$

$$\mathcal{L}_{\text{perc}} = \|\Phi(\mathbf{I}) - \Phi(\mathbf{I}_{gt})\|_1 \ , \tag{17}$$

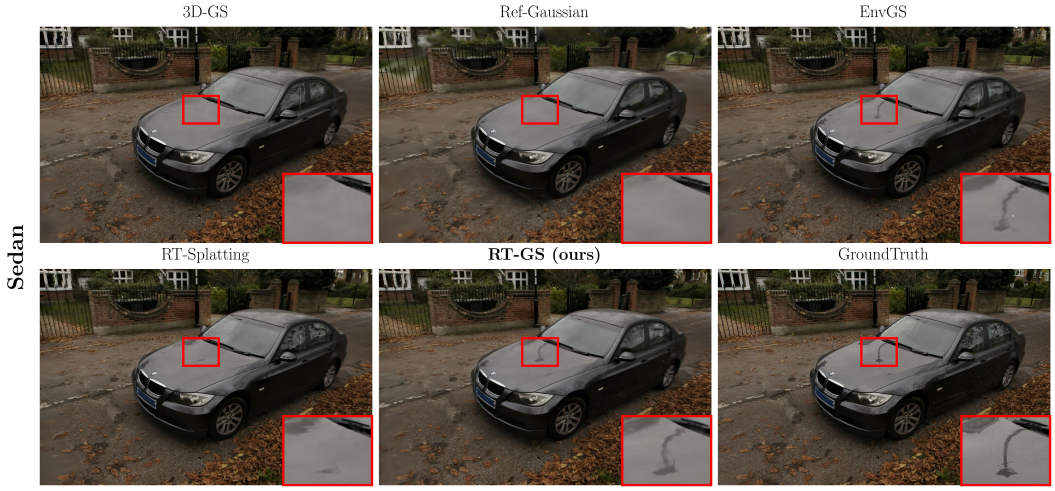


Figure 5: **Qualitative comparison on Ref-Real (Verbin et al., 2024) Scenes.**

Table 1: **Quantitative comparison on the Ref-Real (Verbin et al., 2024) dataset.** Best results are highlighted as 1st , 2nd , 3rd .

Method	Sedan			Spheres			Toycar		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS (Kerbl et al., 2023)	25.556	0.729	0.294	22.386	0.590	0.239	24.394	0.661	0.229
GaussianShader (Jiang et al., 2024)	22.759	0.648	0.395	22.080	0.599	0.268	23.641	0.661	0.283
Ref-GS (Zhang et al., 2025)	25.522	0.724	0.338	22.236	0.593	0.274	24.237	0.646	0.270
Ref-Gaussian (Yao et al., 2025)	24.324	0.675	0.380	22.788	0.614	0.278	23.991	0.644	0.279
EnvGS (Xie et al., 2025)	26.142	0.727	0.300	23.031	0.620	0.196	24.759	0.669	0.206
RT-Splatting (Shi et al., 2026)	25.885	0.727	0.285	22.648	0.613	0.221	24.425	0.666	0.222
RT-GS (ours)	26.203	0.730	0.282	23.041	0.622	0.192	24.827	0.674	0.205

where Φ is the pre-trained VGG-16 network (Simonyan & Zisserman, 2014), $\mathbf{I}$ is the rendered image and $\mathbf{I}_{gt}$ is the ground truth image.

The final loss function is a weighted combination of these loss terms, such that

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \lambda_d\mathcal{L}_d + \lambda_{\text{norm}}\mathcal{L}_{\text{norm}} + \lambda_{\text{mono}}\mathcal{L}_{\text{mono}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} , \tag{18}$$

where $\mathcal{L}_{\text{rgb}}$ is the photometric reconstruction loss combining the $\mathcal{L}_1$ with the D-SSIM term(Kerbl et al., 2023), and λ values are the weights.

4 Implementation and Results

Our training process is divided into 3 stages. We adopt rasterization of diffuse Gaussians for 3,000 steps to obtain all material maps. Then, we generate specular reflection rays and utilize the differentiable ray tracing framework from EnvGS (Xie et al., 2025) to reconstruct reflection colors for 17,000 steps. We extract the surface mesh of diffuse Gaussians at step 20,000 using the truncated signed distance fusion (TSDF) from 2DGS (Huang et al., 2024). After that, the mesh is fixed and we generate transmittance rays guided by the extracted mesh to obtain transmittance colors and jointly optimize these three types of Gaussians for 40,000 steps. The regularization $\mathcal{L}_d$ is added at step 40,000 to push the depth back into the transparent surfaces. The target value K_0 of the constraint to the specular map is set to 0.9.

4.1 Datasets and Comparisons

We evaluate our method on datasets that contain complex specular reflections and transparent objects in real-world scenes to demonstrate the ability to reconstruct complex specular reflections and

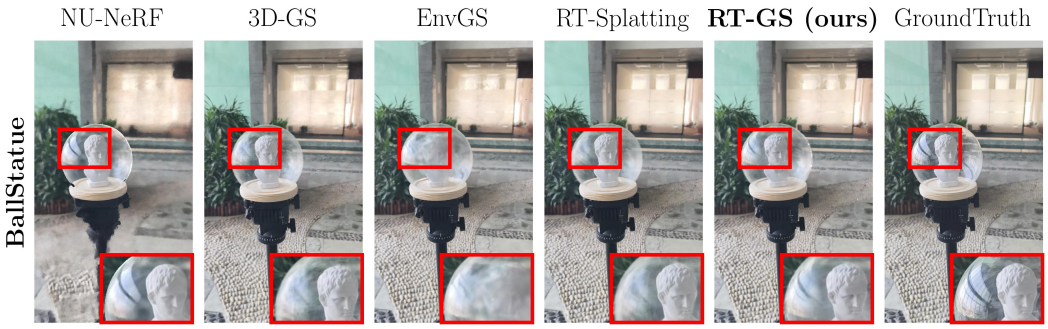


Figure 6: **Qualitative comparison on NU-NeRF (Sun et al., 2024) Scenes.**

Table 2: **Quantitative comparison on NU-NeRF (Sun et al., 2024) scenes.** Best results are highlighted as 1st , 2nd , 3rd .

Method	BallStatue			PopcornCup			RealBottle			TallBottle		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NU-NeRF (Sun et al., 2024)	23.453	0.706	0.345	17.031	0.727	0.333	25.818	0.722	0.336	18.923	0.707	0.344
3DGS (Kerbl et al., 2023)	27.645	0.904	0.135	25.679	0.869	0.202	30.740	0.936	0.162	25.292	0.881	0.178
GaussianShader (Jiang et al., 2024)	25.621	0.874	0.202	16.618	0.701	0.417	22.610	0.824	0.326	18.137	0.734	0.389
Ref-GS (Zhang et al., 2025)	23.889	0.855	0.263	22.919	0.817	0.342	29.527	0.926	0.249	24.818	0.860	0.267
Ref-Gaussian (Yao et al., 2025)	15.192	0.629	0.512	22.837	0.829	0.279	28.019	0.896	0.236	23.766	0.845	0.261
EnvGS (Xie et al., 2025)	26.124	0.885	0.171	23.898	0.847	0.232	30.244	0.915	0.178	25.393	0.886	0.183
RT-Splatting (Shi et al., 2026)	27.744	0.909	0.127	22.961	0.843	0.230	28.011	0.932	0.165	25.553	0.885	0.169
RT-GS (ours)	28.423	0.912	0.122	26.070	0.883	0.186	31.232	0.939	0.143	26.305	0.894	0.167

transmittance appearances concurrently. Specifically, for reflection scenes, we evaluate three scenes from Ref-Real (Verbin et al., 2024): Sedan, Spheres and Toycar. For transparent scenes, we evaluate four scenes from NU-NeRF (Sun et al., 2024): BallStatue, PopcornCup, RealBottle and TallBottle. For the BallStatue scene, we initialize the base Gaussians using only COLMAP points observed in at least three images to reduce unreliable sparse-point initialization. We use the standard PSNR, SSIM (Wang et al., 2004) and LPIPS (Zhang et al., 2018) metrics to evaluate quality.

We compare our method with the following baselines: 3D-GS (Kerbl et al., 2023), GaussianShader (Jiang et al., 2024), Ref-GS (Zhang et al., 2025), Ref-Gaussian (Yao et al., 2025), EnvGS (Xie et al., 2025) and RT-Splatting (Shi et al., 2026). 3D-GS is designed for general diffuse appearance, while the others are the state-of-the-art (SOTA) methods for specular ray effects. For transparent scenes, we add NU-NeRF (Sun et al., 2024) for comparison because it is the SOTA method for transparent objects based on neural signed distance field. We also provide additional comparisons with TransparentGS (Huang et al., 2025) in Appendix subsection A.2.

Figure 5 shows the qualitative comparison on the Ref-Real (Verbin et al., 2024) dataset. More results for other scenes are presented in Appendix subsection A.4. Our method provides better reflection details on the specular surfaces, such as the streetlight reflected on the hood. Table 1 shows the quantitative results, demonstrating that our method achieves the best performance on all other baselines.

Figure 6 shows the qualitative comparison on the transparent dataset. More results for other scenes are presented in Appendix subsection A.4. Our method accurately reconstructs transmittance appearance and models specular reflections of the transparent object, while other methods cannot achieve both at the same time. Like 3D-GS, it can handle the transmittance appearance but is not effective at specular reflections. Table 2 shows the quantitative results, where our method achieves the best results on all other baselines across all transparent scenes.

4.2 Ablation Studies

We present ablation studies on our method to validate the effect of key algorithmic choices. The quantitative and qualitative comparison results are shown in Table 3 and Figure 7 respectively. The



Figure 7: **Ablation study on the BallStatue scene.**

Table 3: **Ablation study on the BallStatue scene.**

	PSNR↑	SSIM↑	LPIPS↓
w/o transmittance Gaussians	27.054	0.901	0.152
w/o specular loss	27.445	0.902	0.149
w/o depth loss	27.700	0.901	0.141
w/o mesh guided ray tracing	27.735	0.902	0.140
Full model	**28.423**	**0.912**	**0.122**

"w/o transmittance Gaussians" option removes the separate transmittance Gaussian and directly utilizes the diffuse Gaussian to reconstruct the object inside the transparent object. As shown in Figure 7(a), it fails to reconstruct the internal object. The "w/o specular loss" option removes the constraint $\mathcal{L}_{\text{spec}}$ on the specular map. Without this constraint, the diffuse Gaussian attempts to reconstruct the internal object, which degrades the training process of the transmittance Gaussian. As shown in Figure 7(b), the internal object also fails to be reconstructed. The "w/o depth loss" option removes the constraint $\mathcal{L}_d$ on the depth map obtained by the first bounce of transmittance rays. Without this constraint, the transmittance Gaussian attempts to model the outside color in areas where there are no internal objects, which fails to be reconstructed in multiple views. A low quality first bounce of transmittance rays affects the quality of transmittance colors. As shown in Figure 7(c), there are artifacts within the transparent object. The "w/o mesh guiding" option removes the guide of the extracted mesh for tracing transmittance rays. The second bounce of transmittance rays starts from the depth of the first bounce of transmittance rays. Then the depth of the first bounce is free of constraints, like the situation in the "w/o depth loss". Therefore, artifacts show up in Figure 7(d). We provide additional ablation studies about the split-sum approximation and reflection Gaussians in Appendix subsection A.1.

5 Conclusion and Discussion

We have introduced RT-GS, a unified framework for reconstructing high-quality specular reflections and the appearance of objects observed through transparent surfaces. Our method integrates a microfacet model for complex material properties and represents the scene using separate Gaussian primitives for diffuse, reflection, and transmittance components. We have also introduced mesh-guided ray tracing and regularization terms for better reconstruction of complex transmittance effects. Extensive experiments on multiple datasets and ablation studies demonstrate the improvements of our method over existing methods in quantitative metrics and visual quality. Our method assumes that the thickness of the outer layer of transparent objects is nearly zero, which makes it difficult to handle the case where the light is strongly curved. Future work could be to extend our method to reconstruct thicker transparent objects.

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing and suggesting experimental parameters. We have not used generative AI tools for other tasks with required disclosure. We have reviewed all AI-assisted work. We checked the accuracy of the AI-polished text and the reasonableness of the suggested training parameters. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will publicly release our source code, including training and evaluation scripts, experiment configurations, and data preparation instructions, to facilitate reproducibility. Our implementation builds on EasyVolcap (Xu et al., 2023), with CUDA-based differentiable Gaussian rasterization and OptiX-based (Parker et al., 2010) ray tracing. Dataset settings, optimization hyperparameters, and training schedules are specified in YAML configuration files.

REFERENCES

Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm siggraph*, volume 2012, pp. 1–7. vol. 2012, 2012.

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

Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21594–21603. IEEE, 2024.

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

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.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5322–5332. IEEE, 2024.

James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 143–150, 1986.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

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 *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8280–8290, 2022.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

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 transactions on graphics (tog)*, 29(4):1–13, 2010.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, third edition, 2016. ISBN 978-0-12-800645-0. doi: 10.1016/C2013-0-15557-2.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Ji Shi, Xianghua Ying, Bowei Xing, Ruohao Guo, and Wenzhen Yue. Rt-splatting: Joint reflection-transmission modeling with gaussian splatting. *arXiv preprint arXiv:2605.18263*, 2026.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pp. 835–846. 2006.

Jia-Mu Sun, Tong Wu, Ling-Qi Yan, and Lin Gao. Nu-nerf: Neural reconstruction of nested transparent objects with uncontrolled capture environment. *ACM Transactions on Graphics (TOG)*, 43 (6):1–14, 2024.

TS Trowbridge and Karl P Reitz. Average irregularity representation of a rough surface for ray reflection. *Journal of the optical society of America*, 65(5):531–536, 1975.

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, 2024.

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.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5742–5751. IEEE, 2025.

Zhen Xu, Tao Xie, Sida Peng, Haotong Lin, Qing Shuai, Zhiyuan Yu, Guangzhao He, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Easyvolcap: Accelerating neural volumetric video research. In *SIGGRAPH Asia 2023 Technical Communications*, pp. 1–4. 2023.

Yuxuan Yao, Zixuan Zeng, Chun Gu, Xiatian Zhu, and Li Zhang. Reflective gaussian splatting. In *International Conference on Learning Representations*, volume 2025, pp. 68695–68711, 2025.

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, 2024a.

Keyang Ye, Qiming Hou, and Kun Zhou. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–10, 2024b.

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

Wenyuan Zhang, Jimin Tang, Weiqi Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Material-refgs: Reflective gaussian splatting with multi-view consistent material inference. *Advances in Neural Information Processing Systems*, 38:89896–89917, 2026.

Youjia Zhang, Anpei Chen, Yumin Wan, Zikai Song, Junqing Yu, Yawei Luo, and Wei Yang. Ref-gs: Directional factorization for 2d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26483–26492. IEEE, 2025.



Figure 8: **Ablation study on the separate reflection Gaussian.** (a) Reconstruction without specular reflection rays. (b) Reconstructing reflections using the environment map. (c) Tracing reflection rays on diffuse Gaussians. (d) Tracing reflection rays on transmittance Gaussians. (e) Tracing reflection rays on separate reflection Gaussians.

Table 4: **Ablation study on the separate reflection Gaussian.**

	PSNR↑	SSIM↑	LPIPS↓
no reflection	25.476	0.722	0.308
reflection on environment map	25.788	0.719	0.309
reflection on diffuse Gaussians	25.196	0.712	0.317
reflection on transmittance Gaussians	25.511	0.721	0.307
reflection on reflection Gaussians	**26.203**	**0.730**	**0.282**

A Appendix

A.1 Ablation studies

A.1.1 Reflection Gaussians.

We present ablation studies on the separate reflection Gaussian. The quantitative and qualitative comparison results are shown in Table 4 and Figure 8 respectively.

The "no reflection" option removes the reflection rays to obtain reflection colors. The reflection color is obtained by diffuse Gaussians. As shown in Figure 8(a), it fails to reconstruct the specular reflection color on the car's surface.

The "reflection on environment map" option reconstructs the specular reflection color by reconstructing and sampling an environment map. As shown in Figure 8(b), it can reconstruct part of the reflection color on the car's surface compared to Figure 8(a), but it fails to reconstruct some details, such as the streetlight.

The "reflection on diffuse Gaussians" and "reflection on transmittance Gaussians" options reconstruct the reflection color by tracing specular reflection rays on diffuse Gaussians and transmittance Gaussians, respectively without separate reflection Gaussians. As shown in Figure 8(c) and (d), they also fail to reconstruct the specular reflection color on the car's surface. The result suggests that using the same set of Gaussians to represent two colors affects the quality of the reconstruction.

The "reflection on reflection Gaussians" option is our full model. As shown in Figure 8(e), it can reconstruct the specular reflections on the car's surface very well.



Figure 9: **Ablation study on the split-sum approximation.** (a) Approximating the BSDF with separate components (Yao et al., 2025; Munkberg et al., 2022) leads to substantial errors in the reconstructions of reflections. (b) Our solution of using the full microfacet model produces better reflections.

A.1.2 Split-sum approximation.

We present ablation studies on the split-sum approximation in the microfacet model. The Figure 9 shows that the split-sum approximation degrades the joint optimization of diffuse and reflection Gaussians. The "with approximation" and "w/o approximation" indicate whether to use or not to use the split-sum approximation in calculating the BRDF term Equation 8 as shown in Figure 9(a) and (b), respectively. The "Final Color" is the blended rendering result combining the "Diffuse Color" with "Reflection Color". In Figure 9(a), while the car window can produce specular reflections, the diffuse surface is not well trained, like the grass in the background. This is because the blending weight k_s is not trained well during the joint optimization, and the specular term of the ground takes a more dominant role, resulting in a poor effect on the diffuse ground. In Figure 9(b), the diffuse and specular reflection terms can be separated well to achieve both good reflections on the glass and diffuse colors on the ground.

A.2 Additional Baselines

We also compare our method with baselines including TransparentGS (Huang et al., 2025). Figure 10 shows the qualitative comparison on the BallStatue scene. In Figure 10 (a), TransparentGS fails to reconstruct the object behind the semi-transparent surface: the statue inside the ball is missing and the ball is treated as a solid glass sphere. This is consistent with its formulation: TransparentGS separates transparent objects and opaque environments, reconstructs the environment excluding transparent objects with 3DGS, and bakes GaussProbes around transparent objects. Rays refract through the transparent object, but the interior/behind-surface content is not reconstructed as a separate transmittance representation. Our method successfully reconstructs the statue inside the ball and the specular reflection color on the ball's surface, as shown in Figure 10 (b).

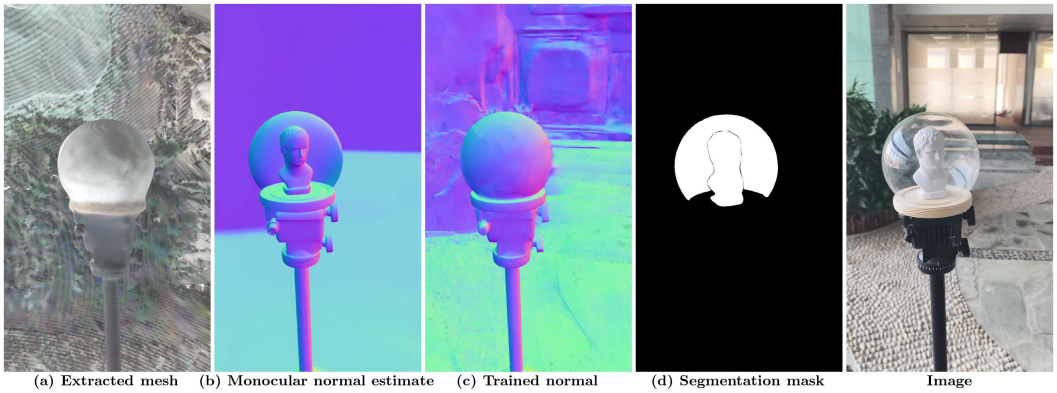
A.3 Robustness of our method

During the optimization, our method utilizes several techniques to guide the training process, including the extracted mesh to guide ray tracing, monocular normal estimates and segmentation masks. The extracted surface mesh is bumpy and not perfect, as shown in Figure 11 (a). The monocular normals estimated by StableNormal (Ye et al., 2024a) are imperfect on transparent surfaces; for example, it fails to recognize the glass sphere and instead treats the statue inside as the surface normal, as shown in Figure 11 (b). However, the normals trained by our method successfully reconstruct the glass sphere, as shown in Figure 11 (c). The segmentation masks using SAM2 (Kirillov et al., 2023; Ravi et al., 2025) guided by GroundingDINO (Liu et al., 2024) for transparent objects are



(a) TransparentGS (b) RT-GS (ours) Ground Truth

Figure 10: **Comparison with TransparentGS (Huang et al., 2025).**



(a) Extracted mesh (b) Monocular normal estimate (c) Trained normal (d) Segmentation mask Image

Figure 11: **Robustness of our method.**

also imperfect; for example, they have boundary errors or missing views, as shown in Figure 11 (d). Despite these imperfections, RT-GS still successfully reconstructs stable reflection and transmission from multiple views, indicating robustness.

A.4 Additional results

We present additional qualitative comparisons for other scenes on the Ref-Real Verbin et al. (2024) and NU-NeRF Sun et al. (2024) datasets. As shown in Figure 12, our method significantly provides more specular reflection details on specular surfaces, such as near-field reflections on the sphere. As shown in Figure 13, our method provides better transmittance appearance results compared to methods that focus on solving reflection problems while also providing better reflection results compared to other methods.

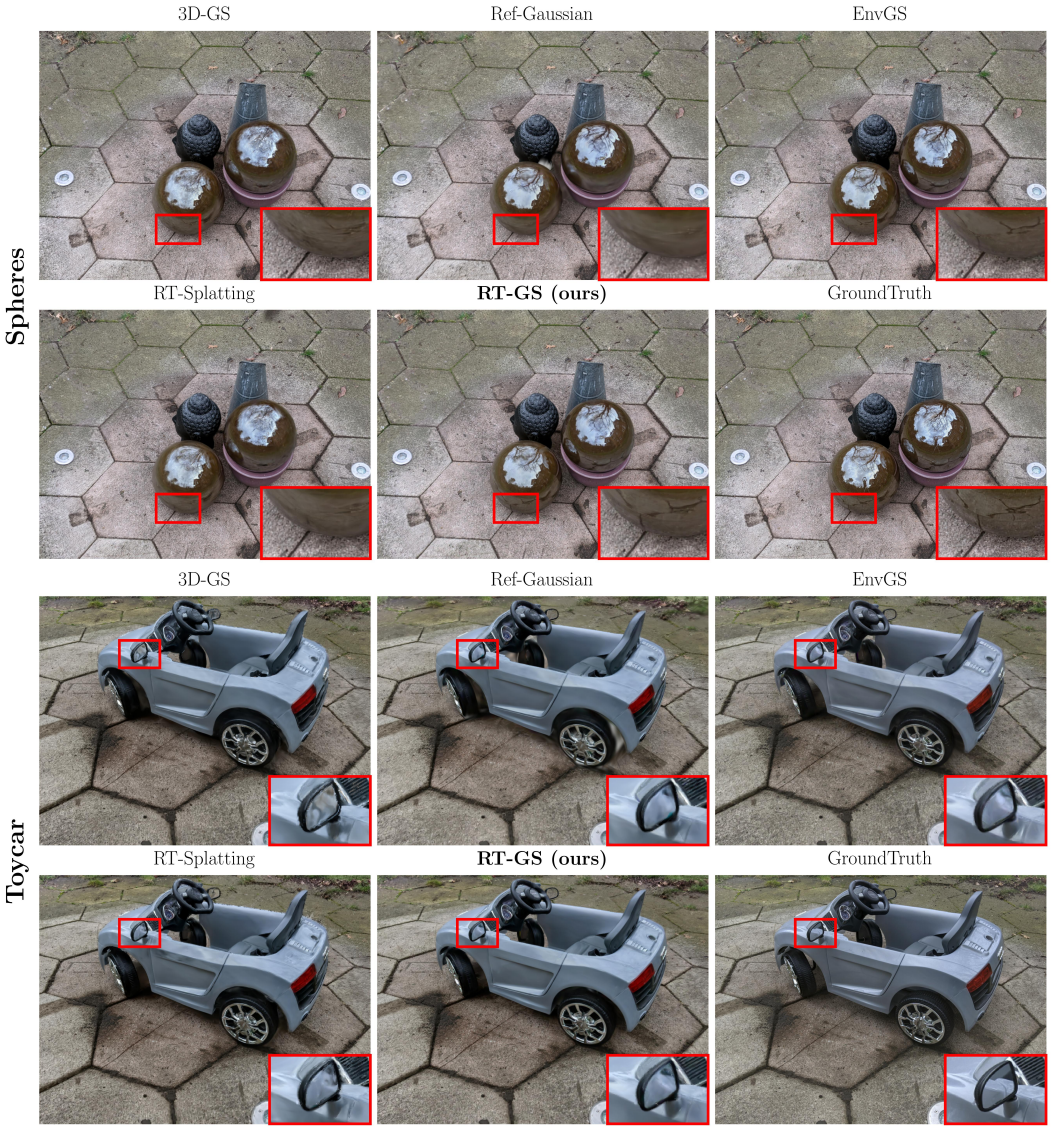


Figure 12: **Qualitative comparison on Ref-Real (Verbin et al., 2024) Scenes.**

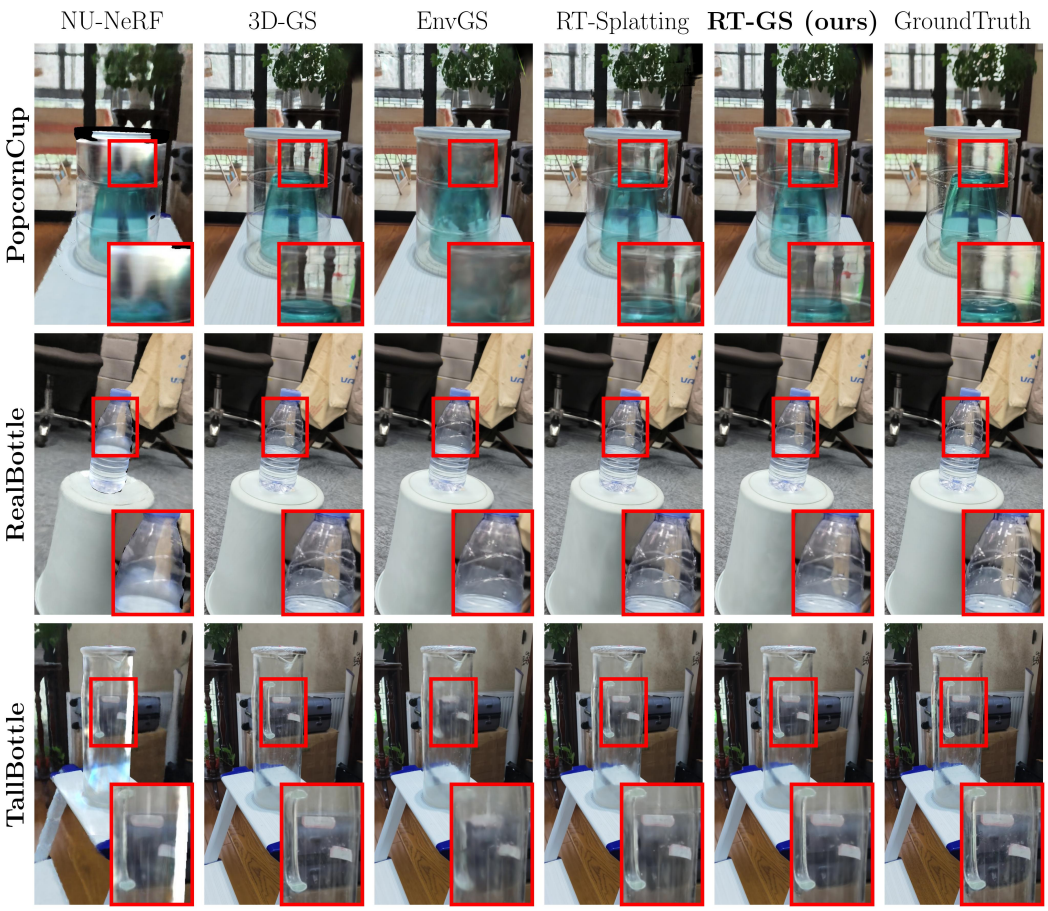


Figure 13: **Qualitative comparison on NU-NeRF (Sun et al., 2024) Scenes.**