REFRACSURFEL: NOVEL VIEW SYNTHESIS OF REFRACTIVE SCENES WITH ENCLOSED OBJECTS VIA GAUSSIAN SURFEL RAY TRACING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent novel view synthesis methods can reconstruct refractive scenes but still struggle with those containing enclosed objects: they either omit enclosed objects or rely on simplified optical models that limit rendering fidelity. To address this, our key idea is to use separate representations for the refractive interface and opaque surfaces: a neural signed distance field and ray-traceable Gaussian surfels, respectively. Our optimization proceeds in three stages: the first stage fits the surfels to the input images and masks while learning a per-surfel probability to separate the foreground from the background; the second stage initializes a signed distance field from the foreground surfels and refines it through differentiable ray tracing; the final stage optimizes surfels representing the enclosed objects using the recovered refracted ray paths. The effectiveness of our approach is demonstrated on various datasets of refractive scenes, comparing favorably against state-of-the-art techniques in terms of rendering quality.

1 INTRODUCTION

Refractive scenes are ubiquitous in the real world, and many of them enclose other objects, such as fish in a tank or specimens in a glass jar. Novel view synthesis of such scenes from posed images is particularly challenging for two reasons. First, refraction breaks multi-view photometric consistency: the same surface point can redirect rays from different viewpoints toward varying scene regions. Second, enclosed objects may be visible only through refraction, making their appearance difficult to recover without an accurate estimate of the refractive interface.

Existing methods either omit enclosed objects or rely on simplified optical models that limit faithful novel-view rendering. TNSR (Deng et al., 2024b) and TransparentGS (Huang et al., 2025) model the refractive object and its surroundings, but do not explicitly reconstruct enclosed opaque objects. Eikonal Field (Bemana et al., 2022) supports enclosed objects, but approximates sharp-interface refraction through continuous ray bending and omits partial reflections. NU-NeRF (Sun et al., 2024) reconstructs nested geometry, but predicts the color of refracted rays with a multilayer perceptron (MLP) instead of tracing them into the scene when recovering the refractive interface. In contrast, our method explicitly models the refractive interface, surrounding scene and enclosed objects, and physically traces reflection and refraction paths for faithful rendering of refractive scenes.

In this paper, we present RefracSurfel, a differentiable approach for novel view synthesis of refractive scenes with enclosed objects. Given posed images and corresponding foreground masks, we represent the refractive interface as a neural signed distance field (SDF), and the background and enclosed objects as ray-traceable Gaussian surfels. The SDF defines surface intersections and normals for refraction and reflection, while the ray-traceable surfels can be queried along arbitrary light paths to provide scene radiance. We optimize these representations progressively in three stages. First, we train Gaussian surfels with straight rays and learn a per-surfel probability to separate the foreground from the background; the resulting foreground surfels provide a geometric proxy for interface initialization, while the background surfels model the surrounding scene. Second, we initialize the SDF from this proxy and refine the refractive interface through differentiable tracing of reflected and refracted rays, while keeping the background surfels fixed. Finally, with the interface

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.

and background fixed, we optimize interior surfels along the recovered light paths to reconstruct the enclosed objects.

The effectiveness of our approach is demonstrated on various real and synthetic refractive scenes, including those with and without enclosed objects. Our approach consistently outperforms state-of-the-art techniques in rendering quality across the evaluated datasets. Moreover, by introducing a learnable absorption coefficient, our method readily extends to absorbing refractive objects. Our code and data will be made publicly available at `https://******`.

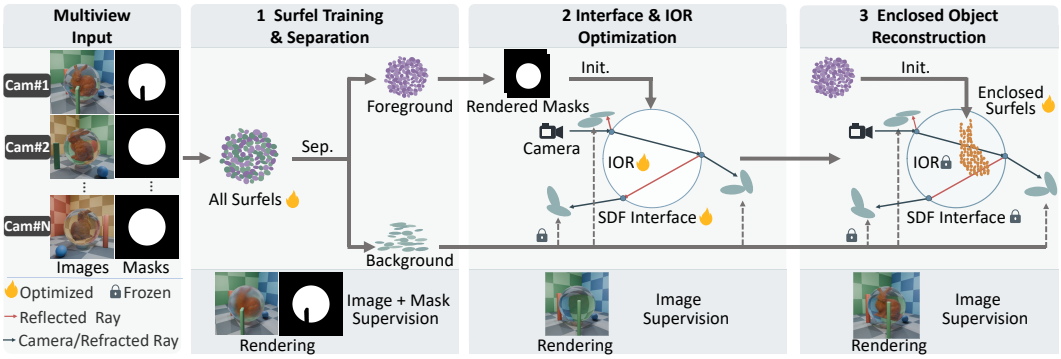

Figure 1: **Our pipeline.** Given multi-view images and foreground masks, we reconstruct the scene in three stages. First, we fit mask-aware ray-traceable Gaussian surfels with straight rays and separate them into foreground and background by their learned foreground probabilities. Second, we initialize an SDF from the foreground surfels and jointly refine it with the index of refraction (IOR) by tracing reflected and refracted rays, keeping the background surfels fixed. Finally, with the SDF, IOR, and background fixed, we optimize surfels for the enclosed objects along the recovered ray paths. The first stage uses both input images and masks as supervision. The second stage initializes the SDF using masks rendered from the foreground surfels, and subsequent refinement and the third stage use only the input images as supervision. Sep. = Separation and Init. = Initialization.

2 Related Work

This section focuses on research on novel view synthesis of refractive scenes from posed images. We divide existing work according to the underlying scene representation, neural fields (Sec. 2.1) and Gaussian primitives (Sec. 2.2). We do not cover several pioneering works on transparent object reconstruction (Zongker et al., 1999; Wexler et al., 2002; Wu et al., 2018) that rely on dedicated hardware setups or controlled backgrounds to provide additional information beyond image inputs. Interested readers are directed to excellent surveys (Ihrke et al., 2010; Karami et al., 2022; Cai et al., 2026) for a broader view of transparent object acquisition and reconstruction.

2.1 Neural Fields for Refractive Scenes

Physics-Based Light Transport. These methods explicitly model refractive effects by incorporating physical constraints into neural-field representations. Interface-based approaches represent the refractive boundary as an explicit surface and trace reflected or refracted rays from its intersections and normals. Most of them (Chen et al., 2023; Zhan et al., 2023; Deng et al., 2024b) reconstruct only the refractive object and its surroundings, not the objects it encloses. ReNeuS (Tong et al., 2023) reconstructs objects inside a transparent container, but assumes that the container geometry is known. NU-NeRF (Sun et al., 2024) handles nested transparent objects with multiple refractive interfaces, but predicts refracted colors with an MLP when recovering the refractive interface. Refractive-index-based approaches (Bemana et al., 2022; Pan et al., 2022) instead represent the scene with a spatially varying refractive-index field and compute bent light paths through the eikonal equation, which handles enclosed objects but blurs the sharp interface. To our knowledge, no existing method jointly recovers the unknown interface and the enclosed object while tracing refraction physically through a sharp interface. Our method achieves all three, yielding higher rendering quality.

Learned Light Transport. In contrast to physics-based formulations that explicitly model light transport, learned approaches directly learn refractive effects from data. HyperReel (Attal et al., 2023) learns ray-conditioned sampling, while Ray Deformation Networks (Deng et al., 2024a) and NEMTO (Wang et al., 2023) employ neural networks to predict ray deformations induced by refraction. MS-NeRF (Yin et al., 2023) takes a more implicit approach, using learned latent features to model appearance under reflection and refraction without explicitly modeling light transport. Unlike our method, these methods do not derive refraction from an interface geometry, so their modeled refractive effects, whether expressed as ray deformations or as appearance, are not guaranteed to be physically consistent.

2.2 Gaussian Primitives for Refractive Scenes

Existing Gaussian-based methods either ignore refraction or cannot reconstruct enclosed objects. A line of work extends Gaussian splatting (Kerbl et al., 2023; Huang et al., 2024) to glass-like surfaces by blending transmitted and reflected components along straight rays, including Glass-Gaussian (Cao et al., 2025), TR-Gaussians (Liu et al., 2026), and RT-Splatting (Shi et al., 2026). Another line of work models refraction. TransparentGS (Huang et al., 2025) renders refraction by querying the surrounding environment through Gaussian light-field probes, but assumes a purely transparent object with nothing inside. Building on Gaussian ray tracing (Moenne-Loccoz et al., 2024), RefracGS (Shao et al., 2026) reconstructs a water surface and the scene beneath it, but assumes a single open air–water interface. In contrast, we reconstruct the whole scene, including the background, the refractive interface, and the enclosed objects, and trace reflected and refracted rays through the closed interface.

3 Preliminaries

Our framework builds on the efficient Gaussian surfel representation (Sec. 3.1), with ray tracing accelerated by a bounding volume hierarchy (BVH) (Gu et al., 2025). We trace refracted and reflected light paths and combine their contributions using the Fresnel weights (Sec. 3.2). To optimize the refractive boundary through this ray-tracing process, ray–surface intersections are differentiated implicitly (Sec. 3.3).

Throughout this paper, *foreground* refers to the refractive interface and any enclosed nonrefractive objects, while *background* denotes the surrounding opaque scene outside the refractive object, including nearby external objects. Each refractive object is assigned a single spatially constant IOR, which is optimized during reconstruction. We trace all refracted rays and the reflected rays at the entry point and the first internal hit. Further reflections are traced only under total internal reflection (TIR). For efficiency, we trace TIR for up to three bounces.

3.1 Gaussian Surfel Representation

A Gaussian surfel (Huang et al., 2024) is parameterized by its center $\boldsymbol{\mu}_i$, two orthonormal tangent vectors $\mathbf{a}_i$ and $\mathbf{b}_i$, tangential scales $s_{u,i}$ and $s_{v,i}$, opacity o_i, and view-dependent colors $\mathbf{c}_i$ represented by spherical harmonics. Its surface position and Gaussian response at local coordinates (u, v) are

$$\mathbf{P}_i(u,v) = \boldsymbol{\mu}_i + s_{u,i}u\mathbf{a}_i + s_{v,i}v\mathbf{b}_i, \qquad G(u,v) = \exp\left(-\frac{u^2+v^2}{2}\right). \tag{1}$$

For an arbitrary ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, the Gaussian response is evaluated at the ray–surfel intersection. With surfel normal $\mathbf{n}_i = \mathbf{a}_i \times \mathbf{b}_i$ and assuming $\mathbf{n}_i^\top \mathbf{d} \neq 0$, the intersection and its local coordinates are computed analytically as

$$\begin{aligned} t_i &= \frac{\mathbf{n}_i^\top(\boldsymbol{\mu}_i - \mathbf{o})}{\mathbf{n}_i^\top \mathbf{d}}, & \mathbf{p}_i &= \mathbf{o} + t_i\mathbf{d}, \\ u_i &= \frac{\mathbf{a}_i^\top(\mathbf{p}_i - \boldsymbol{\mu}_i)}{s_{u,i}}, & v_i &= \frac{\mathbf{b}_i^\top(\mathbf{p}_i - \boldsymbol{\mu}_i)}{s_{v,i}}. \end{aligned} \tag{2}$$

The contributions from these intersections are then accumulated using front-to-back alpha blending. The N intersections with $t_i > 0$ are sorted by increasing t_i. With $\alpha_i = o_i G(u_i, v_i)$ denoting the

opacity at intersection i and T_i the accumulated transmittance before it, the ray color is

$$\mathbf{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i, \qquad T_i = \prod_{j<i} (1 - \alpha_j). \tag{3}$$

3.2 Fresnel Equations

The Fresnel equations determine the reflected and transmitted fractions of incident light, denoted by w_r and w_t, respectively. Let $\eta = \eta_i/\eta_t$ be the relative index of refraction, where η_i and η_t are the refractive indices of the incident and transmitted media, and let c_i and $c_t = \sqrt{1 - \eta^2(1 - c_i^2)}$ denote the cosines of the incident and transmitted angles. If $1 - \eta^2(1 - c_i^2) < 0$, total internal reflection occurs, giving $w_r = 1$ and $w_t = 0$. Otherwise, we use Schlick's approximation (Schlick, 1994):

$$w_r = R_0 + (1 - R_0)(1 - c)^5, \qquad R_0 = \left(\frac{\eta - 1}{\eta + 1}\right)^2, \qquad w_t = 1 - w_r, \tag{4}$$

where $c = c_i$ if $\eta \leq 1$ and $c = c_t$ otherwise, i.e., the cosine of the angle on the optically less dense side of the interface.

3.3 Differentiable Ray–SDF Intersection

Differentiable ray–SDF intersection enables optimization of the refractive interface by propagating rendering gradients through intersection positions and surface normals. For a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, sphere tracing (Hart, 1996) computes an intersection $\mathbf{x}^* = \mathbf{o} + t^*\mathbf{d}$ satisfying $f_{\boldsymbol{\theta}}(\mathbf{x}^*) = 0$. Implicit differentiation of this constraint provides intersection gradients without differentiating through the tracing iterations (Yariv et al., 2020). Let $\mathbf{g} = \nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x}^*)$ denote the spatial SDF gradient. For a locally stable, non-tangential intersection with $\mathbf{g}^\top \mathbf{d} \neq 0$, the derivatives of the intersection distance are

$$\frac{\partial t^*}{\partial \boldsymbol{\theta}} = -\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x}^*)/\partial \boldsymbol{\theta}}{\mathbf{g}^\top \mathbf{d}}, \qquad \frac{\partial t^*}{\partial \mathbf{o}} = -\frac{\mathbf{g}^\top}{\mathbf{g}^\top \mathbf{d}}, \qquad \frac{\partial t^*}{\partial \mathbf{d}} = -\frac{t^* \mathbf{g}^\top}{\mathbf{g}^\top \mathbf{d}}. \tag{5}$$

These partial derivatives treat $\boldsymbol{\theta}$, $\mathbf{o}$, and $\mathbf{d}$ as independent inputs, and the parameter derivative in the numerator is evaluated with $\mathbf{x}^*$ held fixed. The interface normal is obtained as $\mathbf{n} = \mathbf{g}/\|\mathbf{g}\|_2$.

4 Our Approach

We take posed images paired with foreground masks of a refractive scene as input, and output a neural SDF representing the refractive interface, an optimized IOR, and two separate sets of ray-traceable Gaussian surfels representing the enclosed objects and the background, respectively. Together, these components enable novel-view rendering through ray tracing.

Our method reconstructs the scene in three stages that optimize the surfels and the ray paths separately, reducing the ambiguity between appearance and light transport. We first fit mask-aware ray-traceable Gaussian surfels to the input images and corresponding foreground masks using straight rays. A learnable per-surfel probability enables foreground–background separation, providing a background representation and a foreground proxy for interface initialization (Sec. 4.1). We then initialize a neural SDF from the foreground surfels and jointly refine the refractive interface and the IOR through differentiable ray tracing, while keeping the background surfels fixed (Sec. 4.2). Finally, we reconstruct the enclosed objects by optimizing surfels along the recovered refracted paths, with the interface, the IOR, and the background surfels fixed (Sec. 4.3). Please refer to Fig. 1 for a graphical illustration.

4.1 Gaussian Surfel Training and Separation

We first fit ray-traceable Gaussian surfels to the input images and foreground masks using straight rays, without explicitly modeling refraction. We train them with ray tracing, even though this stage uses only straight rays, rather than the rasterization used in 2DGS (Huang et al., 2024). Rasterization composites surfels in order of their center depths, which can differ from the order in which a ray actually intersects them. Training with ray tracing from the start thus avoids the discrepancies that

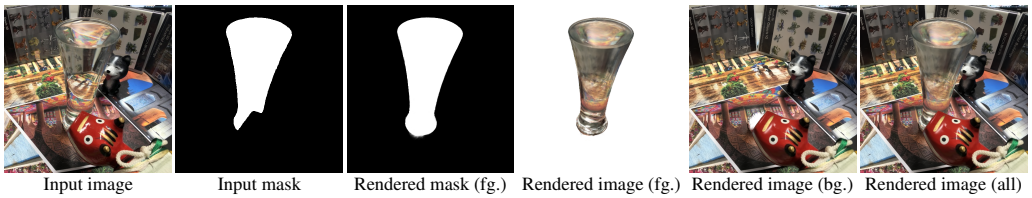

Figure 2: **Gaussian Surfel Training and Separation.** From left to right: the input image and foreground mask; the mask and RGB image rendered from foreground surfels; and the RGB images rendered from background surfels and all surfels, respectively.

would arise from switching rendering formulations in later stages. Following 3DGRT (Moenne-Loccoz et al., 2024), we accelerate ray tracing with a BVH built over the surfels. Details of BVH construction and surfel densification are provided in Appendix A.2.

To separate the foreground from the background consistently across multi-view masks, we assign each surfel a learnable foreground probability $m_i \in [0, 1]$. These probabilities are rendered as a foreground mask using the same alpha-compositing weights as color:

$$\hat{M}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i m_i. \tag{6}$$

We jointly optimize surfel geometry, appearance, and foreground probabilities under photometric and mask supervision:

$$\mathcal{L} = \mathcal{L}_{\text{photo}} + \lambda_{\text{m}} \mathcal{L}_{\text{mask}}. \tag{7}$$

Here, $\mathcal{L}_{\text{photo}}$ combines an L_1 loss and a loss term based on structural similarity (SSIM) between rendered and input images, following 3DGS (Kerbl et al., 2023). The mask loss $\mathcal{L}_{\text{mask}}$ is the L_1 difference between the predicted and input foreground masks, and λ_{m} controls its contribution.

After training, surfels with $m_i > \tau$ form the foreground proxy, while the remaining surfels represent the background and are fixed for the subsequent stages. We render masks from the foreground surfels alone to eliminate background occlusion and obtain consistent silhouette constraints for refractive interface initialization in the next stage, as illustrated in Fig. 2.

4.2 Refractive Interface and IOR Optimization

Given the input images and surfels reconstructed in the previous stage, we initialize the refractive interface from the foreground surfels and jointly refine the interface and the index of refraction (IOR) through differentiable ray tracing. A neural SDF $f_{\boldsymbol{\theta}}$, defined over the bounding box of the foreground surfels, represents only the refractive boundary as its zero level set, excluding the surfaces of enclosed objects. We use an SDF rather than discrete surfels to obtain a continuous refractive boundary with a well-defined interior and exterior. During refinement, we keep the background surfels fixed to prevent them from compensating for incorrect light paths, and exclude the foreground surfels after SDF initialization, deferring enclosed-object modeling to the next stage.

Initialization. We initialize the IOR to a constant η_0 and fit the SDF to masks re-rendered from the foreground surfels obtained in the previous stage. Inspired by the sigmoid SDF-to-density mapping in StyleSDF (Or-El et al., 2022), we convert signed distances to densities and compute the foreground mask value $\hat{M}_{\text{SDF}}(\mathbf{r})$ for each ray through volume rendering:

$$\rho_j = s_\rho \sigma(-\beta f_{\boldsymbol{\theta}}(\mathbf{x}_j)), \qquad \hat{M}_{\text{SDF}}(\mathbf{r}) = 1 - \exp\left(-\sum_{j=1}^{K} \rho_j \Delta t_j\right). \tag{8}$$

Here, $\mathbf{x}_j$ are K points uniformly sampled along the portion of ray $\mathbf{r}$ inside the bounding box, and Δt_j is the sampling interval. The SDF is negative inside the object, σ is the sigmoid function, and s_ρ is a density scale. We gradually increase the sharpness β during fitting, allowing a soft transition initially and a sharper boundary as optimization proceeds.

We minimize the L_1 difference between $\hat{M}_{\text{SDF}}$ (Eq. 8) and $\hat{M}_{\text{fg}}$, obtained using Eq. 6 with compositing restricted to the foreground surfels. We additionally apply Eikonal regularization $\mathcal{L}_{\text{eik}}$ (Gropp

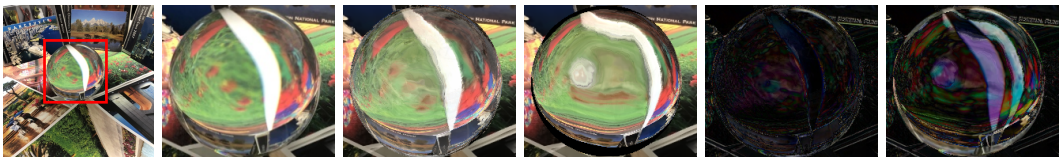

Figure 3: **Refractive interface and IOR refinement.** Left to right: ground truth with a red-boxed region and its crop; renderings after and before refinement; and their differences from ground truth in the same order. Differences are per-pixel absolute RGB errors without normalization or amplification.

et al., 2020) to encourage unit-norm SDF gradients, and curvature regularization $\mathcal{L}_{\text{curv}}$ (Li et al., 2023) to promote surface smoothness by penalizing the absolute finite-difference Laplacian. With λ_{eik} and λ_{curv} controlling the regularization strengths, the initialization objective is

$$\mathcal{L}_{\text{init}} = \left\| \hat{M}_{\text{SDF}} - \hat{M}_{\text{fg}} \right\|_1 + \lambda_{\text{eik}} \mathcal{L}_{\text{eik}} + \lambda_{\text{curv}} \mathcal{L}_{\text{curv}}. \tag{9}$$

Refinement. We jointly refine the SDF and IOR under photometric supervision, deferring enclosed-object modeling to the next stage. We directly compare rendered pixels with their ground-truth counterparts without identifying or excluding those affected by enclosed objects, relying on rays that reach the background to provide sufficient reliable gradients for refinement.

Rays intersect the SDF zero level set via sphere tracing (Hart, 1996), with normalized SDF gradients providing surface normals for computing reflected and refracted directions. At each intersection, we trace the refracted branch unless total internal reflection occurs, and trace the reflected branch following the policy in Sec. 3, i.e., at the entry point, at the first internal hit, and under total internal reflection. For each branch traveling outside the refractive interface, we render the background surfels along its ray to obtain its color contribution. The branch contributions are then recursively combined using the Fresnel weights at each intersection (Eq. 4). Camera rays that miss the interface query the background directly. Fig. 3 illustrates the effect of refinement.

During refinement, gradients flow from the photometric loss through the traced light paths to the SDF and IOR. The surfel renderer provides color derivatives with respect to ray origins and directions, which are propagated to the SDF parameters and IOR through the chain rule and implicit differentiation of ray–SDF intersections (Eq. 5). The refinement objective combines the photometric loss $\mathcal{L}_{\text{photo}}$ (Eq. 7) with the Eikonal and curvature regularizers:

$$\mathcal{L}_{\text{refine}} = \mathcal{L}_{\text{photo}} + \lambda_{\text{eik}} \mathcal{L}_{\text{eik}} + \lambda_{\text{curv}} \mathcal{L}_{\text{curv}}. \tag{10}$$

4.3 Enclosed Object Reconstruction

Given the input images and the SDF, IOR, and background surfels obtained from the previous stages, we optimize ray-traceable Gaussian surfels to represent enclosed objects. These surfels are initialized from the foreground surfels learned in the first stage, with their opacities reset.

Rendering follows Sec. 4.2, with the enclosed-object surfels added to the scene. Both surfel sets share a single BVH, so every ray segment along a light path, including the camera-to-interface segment and the segments inside and outside the interface, is intersected with all surfels. We alpha-composite the intersected surfels within each segment, and the remaining transmittance scales the contribution of the subsequent segments along the path. A per-surfel label restricts gradients to the enclosed-object surfels: we keep the SDF, IOR, and background surfels fixed and optimize only the enclosed-object surfels, using the same photometric loss $\mathcal{L}_{\text{photo}}$ as in the first two stages, which combines L_1 and SSIM-based loss terms between the rendered and input images.

4.4 Implementation Details

Stage 1 trains the Gaussian surfels for 30,000 iterations. Stage 2 uses 3,000 iterations for SDF initialization, followed by 10,000 iterations for joint refinement of the SDF and IOR. Stage 3 runs for

5,000 iterations and is generally omitted for scenes without enclosed objects. Since the foreground surfels can be rendered from arbitrary viewpoints, we supplement the training views with 100 sampled views during SDF initialization to provide additional silhouette constraints. During Stage 2 refinement and Stage 3, training rays are sampled only from pixels whose camera rays intersect the bounding box of the SDF.

Following TNSR (Deng et al., 2024b), we represent the SDF with a multiresolution hash grid followed by an MLP, using compact configuration since the SDF models only the refractive interface rather than the entire scene. Since Fresnel weights combine radiance linearly, we composite path contributions in a linear space obtained by modeling the camera response function (Appendix A.4). Foreground masks, which cover the refractive object together with its enclosed objects, are rendered by Blender for synthetic scenes and obtained with SAM 2 (Ravi et al., 2024) for the remaining scenes by prompting the first training image and propagating the segmentation to the other views. The network architecture, view sampling, and hyperparameters are detailed in Appendix A.3.

5 Results and Discussions

We conduct most experiments on a workstation with dual AMD EPYC 7763 CPUs, 768 GB RAM, and an RTX 4090 GPU. We evaluate our method on 11 scenes from 3 datasets, including four real-world scenes from Eikonal Field (Bemana et al., 2022), three synthetic scenes from TNSR (Deng et al., 2024b) and four additional synthetic scenes rendered with Blender. Although the TNSR paper reports four scenes, only three are publicly available. We do not include additional datasets such as 3DReflecNet (Liang et al., 2026) and RefRef (Yin et al., 2025), as they contain refractive scenes without enclosed objects, a setting already covered by the Eikonal Field and TNSR datasets.

For quantitative assessments of reconstruction quality, we compute peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and learned perceptual image patch similarity (LPIPS). We first average each metric over the test images of each scene and then average the scene-level scores equally across the scenes in each evaluated dataset.

5.1 Comparisons

We compare our approach with 3DGS (Kerbl et al., 2023), which does not explicitly model refraction, and four refraction-aware reconstruction methods: TNSR (Deng et al., 2024b) and TransparentGS (Huang et al., 2025), which model the refractive object but not enclosed objects, and NU-NeRF (Sun et al., 2024) and Eikonal Field (Bemana et al., 2022), which can handle enclosed objects. Averaged over the scenes of each dataset, our method outperforms all baselines on the three datasets.

As shown in Tab. 1, our method achieves the best PSNR, SSIM, and LPIPS on all three datasets, improving PSNR over the strongest baseline on each dataset by 4.08, 0.62, and 2.75 dB on the TNSR, Eikonal Field, and our synthetic datasets, respectively. Fig. 4 and the accompanying video present qualitative comparisons on representative real and synthetic scenes, where the full ground-truth images indicate the selected regions and the enlarged crops show local rendering details. Our renderings preserve sharp refracted details and the appearance of enclosed objects.

Table 1: Quantitative comparisons between our approach and state-of-the-art techniques on three datasets: TNSR (Deng et al., 2024b), Eikonal Field (Bemana et al., 2022), and our synthetic dataset. We report PSNR, SSIM, and LPIPS on held-out test views, averaged over all scenes in each dataset. **Bold** marks the best result in each column.

Method / Dataset	TNSR			Eikonal Field			Ours		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	**27.78**	**0.941**	**0.059**	**23.90**	**0.855**	**0.097**	**29.84**	**0.927**	**0.079**
3DGS	22.10	0.881	0.152	23.28	0.849	0.098	27.09	0.925	0.101
NU-NeRF	19.59	0.695	0.336	20.33	0.676	0.291	24.30	0.800	0.246
Eikonal Field	18.91	0.705	0.385	21.25	0.693	0.249	23.66	0.821	0.234
TNSR	23.70	0.861	0.121	21.60	0.787	0.154	22.05	0.838	0.283
TransparentGS	21.46	0.864	0.131	19.32	0.802	0.155	22.74	0.883	0.162



Figure 4: **Qualitative comparisons with state-of-the-art methods.** We compare our method with 3DGS (Kerbl et al., 2023), TransparentGS (Huang et al., 2025), NU-NeRF (Sun et al., 2024), TNSR (Deng et al., 2024b), and Eikonal Field (Bemana et al., 2022). The numbers below each crop report PSNR↑, SSIM↑, and LPIPS↓, respectively, computed over the displayed region.

Table 2: **Ablation studies.** We evaluate both ablations with PSNR, SSIM, and LPIPS, averaged over the scenes of each dataset, with the best results in bold. **Left:** We ablate the optimization strategy on the four scenes of our synthetic dataset. *w/o Refine* and *w/o Enclosed* remove interface refinement and enclosed-object reconstruction, respectively, while *Joint-I* additionally optimizes the SDF and IOR, and *Joint-IB* further optimizes the background surfels, together with the enclosed-object surfels in the last stage. **Right:** We ablate the light path modeling on the three scenes of the TNSR dataset.

	Optimization strategy					Light path modeling			
	w/o Refine	w/o Enclosed	Joint-I	Joint-IB	Ours	Eikonal	No refl.	Single refl.	Dual refl. (ours)
PSNR ↑	28.99	23.86	29.59	29.67	**29.84**	19.70	23.92	26.69	**27.78**
SSIM ↑	0.922	0.915	0.922	0.923	**0.927**	0.842	0.918	0.936	**0.941**
LPIPS ↓	0.082	0.115	0.080	0.080	**0.079**	0.149	0.084	0.063	**0.059**

5.2 Ablation Studies

We ablate two main design choices of our method: the optimization strategy, by removing interface refinement or enclosed-object reconstruction, and the light path modeling, by replacing the sharp interface with an eikonal field or reducing the reflections traced at the interface. Tab. 2 summarizes the results of both ablations. We further evaluate the robustness of our method to corrupted foreground masks (Appendix A.5).

Optimization Strategy. We evaluate our optimization strategy on our synthetic dataset, since all four of its scenes contain enclosed objects, whereas both the TNSR and Eikonal Field datasets include scenes with purely transparent objects. As shown in Tab. 2 (left), removing enclosed-object reconstruction causes the largest drop, lowering PSNR by 5.98 dB and highlighting the importance of explicitly modeling enclosed objects. Removing interface refinement lowers PSNR by 0.85 dB

and degrades all three metrics, indicating that refinement improves rendering quality over the initialized interface. We further examine whether the interface should remain fixed in the last stage by optimizing the SDF and IOR jointly with the enclosed-object surfels, with or without the background surfels. Neither variant improves over our strategy, with PSNR decreasing by 0.25 dB and 0.17 dB, respectively. We therefore keep the SDF, IOR, and background surfels fixed in the last stage.

Light Path Modeling. We evaluate light path modeling on the three TNSR scenes, which contain no enclosed objects. All variants share the same fixed background surfels from Stage 1 and differ only in the light paths. First, we render our background along the paths of a pretrained Eikonal Field (Bemana et al., 2022), which bends rays continuously instead of refracting them at a sharp interface. Second, we re-run Stage 2 from the same SDF initialization with three reflection settings: *No reflection*, as in Eikonal Field (Bemana et al., 2022); *Single reflection* at the entry point, as in TNSR (Deng et al., 2024b); and *Dual reflections*, which adds reflection at the first internal hit and total internal reflection of up to three bounces. As shown in Tab. 2 (right), the eikonal paths reduce PSNR by 8.08 dB relative to *Dual reflections*, while *Single reflection* and *No reflection* reduce it by 1.09 dB and 3.86 dB, respectively. This suggests that, for the same background, sharp-interface refraction better reproduces the observations on these scenes, and that modeling additional reflections further improves rendering quality.

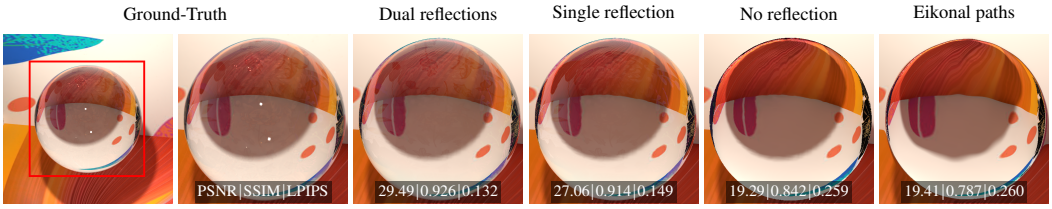

Figure 5: **Ablation of light path modeling on the Ball scene of the TNSR dataset.** PSNR, SSIM, and LPIPS are computed over the displayed cropped regions.

6 Limitations and Future Work

Our work is subject to a number of limitations. First, we model each refractive object as a homogeneous dielectric medium with a smooth interface and a single, spatially constant, and wavelength-independent IOR. This does not handle rough or frosted surfaces, subsurface scattering, dispersion, or nested refractive objects with distinct IORs, such as a glass marble immersed in water. Second, the refractive interface is refined through refracted rays that reach the surrounding scene. When the enclosed objects occupy most of the interior, or the surroundings are sparsely observed, fewer rays provide constraints on the interface.

In the future, it is promising to incorporate complementary measurements, such as polarization imaging or computed tomography scans (Kang et al., 2023), as additional inputs to provide stronger constraints on refracted light paths. We also hope that large models with generative priors could be leveraged to capture refractive effects within a unified scene representation, without explicitly decomposing the scene into separate representations for different components. This may ultimately enable more flexible reconstruction of complex refractive scenes with fewer hand-designed assumptions about scene structure and light transport.

AI Use Statement

We used generative AI tools to provide feedback on research methodology and experimental design, assist with method implementation and debugging, and discuss experimental observations. We also used these tools to assist with literature search, scientific figure design, LaTeX formatting, and drafting and editing the manuscript for clarity and readability. Generative AI was not used to generate experimental data, results or references.

All AI-assisted suggestions, code, and scientific claims were reviewed and verified by the authors. The authors made the final decisions regarding the method, experiments, analysis, and presentation, and take full responsibility for the content of this work.

Reproducibility Statement

We have made every effort to ensure that our results are reproducible. Our method is described in Sec. 4, and implementation details and hyperparameters are given in Sec. 4 and the appendix. Our code is included in the supplementary material for review and will be publicly released. All evaluations use the publicly available Eikonal Field and TNSR datasets and our synthetic dataset, which will also be released.

Ethics Statement

This work adheres to the ICLR Code of Ethics. No human subjects or animal experimentation were involved. All datasets used, including the public Eikonal Field and TNSR datasets and our synthetic scenes rendered in Blender, depict only inanimate objects and contain no personally identifiable information. We are not aware of any privacy, security, or fairness concerns arising from this work.

References

Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16610–16620, 2023.

Mojtaba Bemana, Karol Myszkowski, Jeppe Revall Frisvad, Hans-Peter Seidel, and Tobias Ritschel. Eikonal fields for refractive novel-view synthesis. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–9, 2022.

Xinyang Cai, Hedan Liu, Yiwen Qi, and Hao Chen. Non-invasive 3d reconstruction techniques for transparent objects: A systematic review. *Applied Intelligence*, 56(13):387, 2026.

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

Xiaoxue Chen, Junchen Liu, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Nerrf: 3d reconstruction and view synthesis for transparent and specular objects with neural refractive-reflective fields. *arXiv preprint arXiv:2309.13039*, 2023.

Weijian Deng, Dylan Campbell, Chunyi Sun, Shubham Kanitkar, Matthew Shaffer, and Stephen Gould. Ray deformation networks for novel view synthesis of refractive objects. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3106–3116. IEEE, 2024a.

Weijian Deng, Dylan Campbell, Chunyi Sun, Shubham Kanitkar, Matthew E Shaffer, and Stephen Gould. Differentiable neural surface refinement for modeling transparent objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20268–20277. IEEE, 2024b.

Isaac Deutsch, Nicolas Moënne-Loccoz, Zan Gojcic, et al. Ppisp: Physically-plausible compensation and control of photometric variations in radiance field reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7289–7298, 2026.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10943–10952. IEEE, 2025.

John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996.

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.

Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. Transparent and specular object reconstruction. In *Computer graphics forum*, volume 29, pp. 2400–2426. Wiley Online Library, 2010.

Kaizhang Kang, Zoubin Bi, Xiang Feng, Yican Dong, Kun Zhou, and Hongzhi Wu. Differentiable dynamic visible-light tomography. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–12, 2023.

Ali Karami, Roberto Battisti, Fabio Menna, and Fabio Remondino. 3d digitization of transparent and glass surfaces: State of the art and analysis of some methods. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:695–702, 2022.

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.

Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. In *European conference on computer vision*, pp. 441–457. Springer, 2024.

Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8456–8465. IEEE, 2023.

Zhicheng Liang, Haoyi Yu, Boyan Li, Dayou Zhang, Zijian Cao, Tianyi Gong, Junhua Liu, Shuguang Cui, and Fangxin Wang. 3dreflecnet: A large-scale dataset for 3d reconstruction of reflective, transparent, and low-texture objects. *arXiv preprint arXiv:2605.10204*, 2026.

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*, 2026.

Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo De Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics (TOG)*, 43(6):1–19, 2024.

Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13503–13513, 2022.

Jen-I Pan, Jheng-Wei Su, Kai-Wen Hsiao, Ting-Yu Yen, and Hung-Kuo Chu. Sampling neural radiance fields for refractive objects. In *SIGGRAPH Asia 2022 Technical Communications*, pp. 1–4. Association for Computing Machinery, 2022.

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. *arXiv preprint arXiv:2408.00714*, 2024.

Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pp. 233–246. Wiley Online Library, 1994.

Yiming Shao, Qiyu Dai, Chong Gao, Guanbin Li, Yeqiang Wang, He Sun, Qiong Zeng, Baoquan Chen, and Wenzheng Chen. Refracgs: Novel view synthesis through refractive water surfaces with 3d gaussian ray tracing. *arXiv preprint arXiv:2603.21695*, 2026.

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.

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.

Jinguang Tong, Sundaram Muthu, Fahira Afzal Maken, Chuong Nguyen, and Hongdong Li. Seeing through the glass: Neural 3d reconstruction of object inside a transparent container. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12555–12564. IEEE, 2023.

Dongqing Wang, Tong Zhang, and Sabine Süsstrunk. Nemto: Neural environment matting for novel view and relighting synthesis of transparent objects. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 317–327. IEEE, 2023.

Yonatan Wexler, Andrew W. Fitzgibbon, and Andrew Zisserman. Image-based environment matting. In *Proceedings of the 13th Eurographics Workshop on Rendering*, 2002.

Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3d reconstruction of transparent objects. *arXiv preprint arXiv:1805.03482*, 2018.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

Yue Yin, Enze Tao, Weijian Deng, and Dylan Campbell. Refref: A synthetic dataset and benchmark for reconstructing refractive and reflective objects. *arXiv preprint arXiv:2505.05848*, 2025.

Ze-Xin Yin, Jiaxiong Qiu, Ming-Ming Cheng, and Bo Ren. Multi-space neural radiance fields. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12407–12416. IEEE, 2023.

Yifan Zhan, Shohei Nobuhara, Ko Nishino, and Yinqiang Zheng. Nerfrac: Neural radiance fields through refractive surface. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18356–18366. IEEE, 2023.

Douglas E Zongker, Dawn M Werner, Brian Curless, and David H Salesin. Environment matting and compositing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 205–214, 1999.

A Appendix

This appendix provides an extension of our method, additional implementation details, and further evaluations. We first extend our method to refractive media with absorption and evaluate it on two scenes from our synthetic dataset rendered with absorbing glass materials (Appendix A.1). We then describe the details of Gaussian surfel ray tracing, including BVH construction, initialization, and adaptive densification (Appendix A.2), followed by the training settings and hyperparameters (Appendix A.3). Next, we detail how reflected and refracted contributions are composited in an approximately linear space obtained by modeling the camera response function (Appendix A.4). We then evaluate the robustness of our method to corrupted foreground masks (Appendix A.5). Finally, we ablate the Eikonal and curvature regularization terms of the SDF on the Eikonal Field dataset (Appendix A.6).

A.1 Extension to Absorbing Refractive Media

Our method can be extended to colored refractive objects that absorb light as it propagates through the medium, by incorporating a homogeneous absorption model into the rendering of ray segments inside the refractive interface. The scenes used in our main experiments do not exhibit absorption. To evaluate this extension, we additionally render the synthetic *Bunny* and *Hornet* scenes in Blender, each with a different absorbing glass material.

We assume that the refractive medium is homogeneous, so that its absorption can be described by a single set of RGB absorption coefficients $\boldsymbol{\sigma}_a = (\sigma_R, \sigma_G, \sigma_B)$ shared by the entire object. We parameterize $\boldsymbol{\sigma}_a$ through a softplus function to keep it non-negative. Following the Beer–Lambert law, a path of length L inside the medium has the per-channel transmittance

$$\mathbf{T}_{\text{abs}}(L) = \exp(-\boldsymbol{\sigma}_a L)\,, \tag{11}$$

where the exponential is applied channel-wise, so that different attenuation rates across channels produce the color of the glass. Absorption applies only to ray segments traveling inside the refractive interface, whose entry and exit points are given by sphere tracing on the SDF, and is computed in linear space (Appendix A.4).

For paths through enclosed objects, the surfels intersected along the internal segment are alpha-composited into a color $\mathbf{C}_G$ with accumulated opacity A. Since $\mathbf{C}_G$ aggregates multiple surfels, we approximate its attenuation with a single representative depth, where the accumulated opacity reaches 0.5, and denote by L_f the path length from the entry point to this depth, or to the end of the segment if the threshold is not reached. With L_b the remaining internal path length and $\mathbf{C}_{\text{bg}}$ the background color along the path, the transmitted color is

$$\mathbf{C}_{\text{trans}} = \mathbf{T}_{\text{abs}}(L_f)\,[\mathbf{C}_G + (1 - A)\,\mathbf{T}_{\text{abs}}(L_b)\,\mathbf{C}_{\text{bg}}]\,, \tag{12}$$

so that the background is attenuated by the full internal path length $L_f + L_b$. The absorption coefficients $\boldsymbol{\sigma}_a$ are optimized through the photometric loss in the second stage using Eq. 11 and in the final stage using Eq. 12. Fig. 6 shows qualitative comparisons with 3DGS on both scenes.



Figure 6: **Qualitative comparison with 3DGS on absorbing scenes.** For *Hornet* (left) and *Bunny* (right), each group shows the full ground-truth image, a ground-truth crop, and the corresponding crops from our method and 3DGS. PSNR, SSIM, and LPIPS are computed over the displayed cropped regions.

A.2 Details of Gaussian Surfel Ray Tracing

BVH Construction. We accelerate ray–surfel intersection with a BVH over proxy geometry, in which each surfel's truncated support is conservatively enclosed by a circumscribed diamond composed of two triangles. Using the parameterization in Sec. 3.1, we truncate the Gaussian response of each surfel to the disk $u^2 + v^2 \le r_i^2$ in normalized local coordinates, where the radius r_i is determined by the kernel-response cutoff. In world space, this support is an ellipse, which we enclose with a planar diamond whose vertices are

$$\begin{aligned} \mathbf{v}_{0,1} &= \boldsymbol{\mu}_i \pm \sqrt{2}\, r_i s_{u,i} \mathbf{a}_i, \\ \mathbf{v}_{2,3} &= \boldsymbol{\mu}_i \pm \sqrt{2}\, r_i s_{v,i} \mathbf{b}_i. \end{aligned} \tag{13}$$

The factor $\sqrt{2}$ makes the diamond tangent to the ellipse. We split the diamond into triangles $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$ and $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_3)$ and build the BVH with OptiX. Following 3DGRT (Moenne-Loccoz et al., 2024), we rebuild the BVH at every training iteration. Since the proxies lie in the surfel plane, a proxy hit yields the ray–surfel intersection of Eq. 2. We accept the hit only if its local coordinates satisfy $u_i^2 + v_i^2 \le r_i^2$, and then evaluate the Gaussian response and opacity there.

Initialization and Adaptive Densification. In the first stage, we follow 2DGS (Huang et al., 2024) for initialization and 3DGRT for densification, while additionally orienting each surfel to face its nearest training camera at initialization and adapting the score for selecting surfels to densify. We compute this score from the 3D positional gradient of the photometric loss, which we project onto the surfel's tangent plane (Li et al., 2024) and empirically further project to remove its component along the camera's forward axis:

$$\tilde{\mathbf{g}}_{iv} = \left(\mathbf{I} - \mathbf{z}_v \mathbf{z}_v^\top\right)\left(\mathbf{I} - \mathbf{n}_i \mathbf{n}_i^\top\right) \nabla_{\boldsymbol{\mu}_i} \mathcal{L}_{\text{photo},v}, \tag{14}$$

where $\mathbf{n}_i$ is the unit surfel normal and $\mathbf{z}_v$ is the unit forward axis of camera v, both in world coordinates. Following 3DGRT (Moenne-Loccoz et al., 2024), we scale $\|\tilde{\mathbf{g}}_{iv}\|$ by half the camera–surfel distance and average it over all views observing the surfel to obtain the score. These projections are used only for densification and do not affect the gradients used for optimization. In the final stage, we initialize the enclosed-object surfels from the foreground surfels of the first stage with their opacities reset. Since these surfels are observed along refracted rays, the camera's forward axis and the camera–surfel distance no longer describe the viewing geometry, so we select surfels for splitting directly by their accumulated positional gradients: at each split step, we select the top 5% of surfels by averaged positional gradient, up to 2,048 parent surfels.

A.3 Details of Training

SDF Network. The hash grid has eight levels with two features each, geometrically spaced resolutions from 16 to 64, at most 2^{17} entries per level, and smoothstep interpolation. The 16 encoded features and the 3 normalized coordinates form a 19-dimensional input to an MLP with hidden widths $(128, 128, 128, 109)$ and Softplus activations ($\beta = 100$). Before the scalar output layer, we concatenate the 19-dimensional input with the last hidden layer, giving a 128-dimensional feature.

Sampled Views for SDF Initialization. The 100 additional cameras are placed uniformly on a sphere centered at the least-squares intersection of the training cameras' optical axes, with a radius equal to the mean distance of the training cameras to this center, and are oriented toward the center. Target masks $\hat{M}_{\text{fg}}$ are rendered from the foreground surfels at both the training and sampled views. The mask term in Eq. 9 is computed separately over the two view pools and combined with weights 0.9 and 0.1 for the training and sampled views, respectively.

Hyperparameters and Optimization. The per-surfel foreground probabilities m_i are initialized to 0.01, and the separation threshold is set to $\tau = 0.01$. The IOR of the refractive object is initialized to $\eta_0 = 1.5$, and that of the surrounding air is set to 1.0003. The mask-loss weight in Stage 1 is $\lambda_{\text{m}} = 1.0$, and the Eikonal and curvature regularization weights are $\lambda_{\text{eik}} = 0.1$ and $\lambda_{\text{curv}} = 4 \times 10^{-4}$ during both SDF initialization and refinement. We use Adam for all optimization stages. The learning rate of the SDF network follows a cosine decay from 10^{-3} to 10^{-5} during initialization and remains fixed at 10^{-5} during refinement, while that of the IOR follows a cosine decay from 10^{-3} to 10^{-5} during refinement. Surfels use the same learning-rate settings in Stages 1 and 3, following 3DGRT (Moenne-Loccoz et al., 2024), and the foreground probabilities are optimized with a learning rate of 0.05 in Stage 1. We run all three stages for every scene in our synthetic dataset, and only the first two stages for all scenes in the TNSR dataset. For the Eikonal Field dataset, we run the first two stages for *Ball* and all three stages for *Glass*, *Pen*, and *WineGlass*. For *Glass*, Stage 3 approximates residual appearance not fully captured by the refractive model.

A.4 Details of Reflection–Refraction Compositing

We combine the reflected and refracted contributions at each interface intersection using the Fresnel weights. Since these weights split radiance, which adds linearly, the combination must be performed in linear radiance space. Observed image colors, however, are related to radiance through an unknown nonlinear camera response function (CRF). To composite in linear space, we additionally model the CRF with a parametric function $\mathcal{C}_{\boldsymbol{\phi}}$ implemented with PPISP (Deutsch et al., 2026).

Since the background surfels are fitted to observed image colors, we map the rendered color $\mathbf{c}_p$ of each light path p to linear space with $\mathcal{C}_{\boldsymbol{\phi}}^{-1}$, combine the paths using their transport weights, and map the result back to image space:

$$\hat{\mathbf{c}} = \mathcal{C}_{\boldsymbol{\phi}} \left(\sum_p w_p \mathcal{C}_{\boldsymbol{\phi}}^{-1}(\mathbf{c}_p) \right), \tag{15}$$

where w_p is the transport weight of path p, given by the product of Fresnel reflection and transmission factors along the path. For absorbing media (Appendix A.1), absorption is likewise applied in linear space before the forward mapping. The CRF is optimized with a learning rate of 10^{-4} during the last 5,000 iterations of the second stage, and fixed in the final stage.

A.5 Robustness to Mask Quality

By applying multi-view mask supervision to per-surfel foreground probabilities, our method reduces the impact of erroneous masks. To evaluate robustness to corrupted masks, we randomly remove foreground blocks and sample background regions of equal area for every training mask, covering $p\%$ of the original foreground area, where $p \in \{10, 25\}$. Fig. 7 shows the corrupted input masks and the final image renderings. Even when 25% of the foreground area is removed, our method still maintains high visual quality.

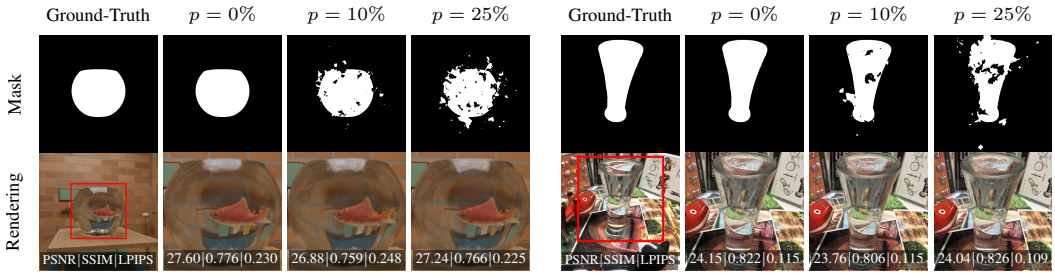

Figure 7: **Robustness to corrupted foreground masks.** Each group corresponds to one scene. The top row illustrates mask corruption on held-out test views, and the bottom row shows the corresponding novel-view renderings from models trained under the same corruption settings. The numbers at the bottom of the renderings report PSNR, SSIM, and LPIPS, respectively, computed over the cropped regions indicated by the red boxes.

A.6 Ablation of SDF Regularization

We evaluate the effects of SDF regularization on the four scenes of the Eikonal Field dataset by removing either the Eikonal or curvature term, or scaling the curvature weight by 10 or 0.1. For each variant, we reuse the same Stage 1 surfels and rerun SDF initialization and refinement with the modified weights applied throughout both steps. We then follow the same Stage 3 training procedure as the default configuration, except for *Ball*, which is evaluated after Stage 2 in all configurations. As shown in Tab. 3, the default weights achieve the best results across all three metrics, supporting the use of both regularizers. Changing the curvature weight by an order of magnitude in either direction reduces average PSNR by less than 0.36 dB, indicating that rendering quality remains relatively stable under substantial changes in curvature regularization strength.

Table 3: **Ablation of SDF regularization.** Results are averaged equally over the four scenes of the Eikonal Field dataset. Bold indicates the best result.

Setting	λ_{eik}	λ_{curv}	PSNR ↑	SSIM ↑	LPIPS ↓
w/o curvature	0.1	0	23.05	0.840	0.120
w/o Eikonal	0	4×10^{-4}	23.11	0.841	0.112
Higher curvature weight	0.1	4×10^{-3}	23.55	0.847	0.105
Lower curvature weight	0.1	4×10^{-5}	23.60	0.849	0.107
Default	0.1	4×10^{-4}	**23.90**	**0.855**	**0.097**