
RefRef: A Synthetic Dataset and Benchmark for Reconstructing Refractive and Reflective Objects

Yue Yin

Australian National University
yue.yin1@anu.edu.au

Enze Tao

Australian National University
enze.tao@anu.edu.au

Weijian Deng

Australian National University
weijian.deng@anu.edu.au

Dylan Campbell

Australian National University
dylan.campbell@anu.edu.au

Abstract

Modern 3D reconstruction and novel view synthesis approaches have demonstrated strong performance on scenes with opaque Lambertian objects. However, most assume straight light paths and therefore cannot properly handle refractive and reflective materials. Moreover, datasets specialized for these effects are limited, stymieing efforts to evaluate performance and develop suitable techniques. In this work, we introduce a synthetic RefRef dataset and benchmark for reconstructing scenes with refractive and reflective objects from posed images. Our dataset has 50 such objects of varying complexity, from single-material convex shapes to multi-material non-convex shapes, each placed in three different background types, resulting in 150 scenes. We also propose an oracle method that, given the object geometry and refractive indices, calculates accurate light paths for neural rendering, and an approach based on this that avoids these assumptions. We benchmark these against several state-of-the-art methods and show that all methods lag significantly behind the oracle, highlighting the challenges of the task and dataset.

1 Introduction

Refractive and reflective objects, such as glass and water, pose significant challenges for 3D reconstruction and novel view synthesis due to the complex behavior of light as it passes through or reflects off these materials. Accurate modeling of these optical phenomena is essential for achieving precise 3D reconstructions and photorealistic renderings. However, most existing neural radiance approaches [28, 1, 6, 30], while excelling at handling scenes with opaque Lambertian objects, struggle with scenes containing refractive or reflective surfaces [12]. These methods typically assume that light travels in straight lines, which is valid for opaque Lambertian objects, as illustrated in Figure 1a, but fails in scenarios with complex optical behavior. This limitation is further compounded by the lack of suitable datasets focused on refractive and reflective properties, hindering the development and evaluation of more specialized methods.

To address these challenges, we introduce RefRef, a synthetic dataset and benchmark designed for the task of reconstructing scenes with complex refractive and reflective objects, as shown in Figure 1d. RefRef consists of 50 objects categorized based on their geometric and material complexity: single-material convex objects, single-material non-convex objects, and multi-material non-convex objects, where the materials have different colors, opacities, and refractive indices. Each object is rendered in three background settings: two bounded and one unbounded, resulting in 150 unique scenes with diverse geometries, material properties, and backgrounds. This provides a controlled environment for evaluating and developing 3D reconstruction methods that handle complex optical effects.

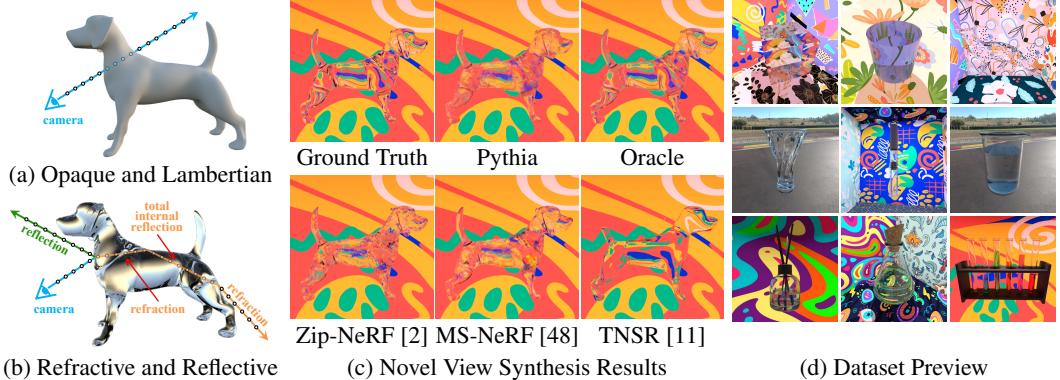


Figure 1: Comparison of light interactions in scenes with opaque Lambertian objects and refractive/reflective objects. (a) Opaque Lambertian objects allow modeling of light paths as linear. (b) Refractive and reflective objects lead to the curving and branching of light paths. (c) Novel view synthesis results. Our oracle achieves the best result, while our Pythia method loses some high-frequency detail and TNSR [11] sacrifices geometric quality for visual crispness. (d) Sample scenes from the RefRef dataset.

We also propose an oracle method that has access to the ground-truth geometry and refractive indices of refractive objects in the scene, allowing it to compute accurate light paths, as shown in Figure 1b. This approach provides a performance target for NeRF-based methods, showing how well they can perform if the light paths are properly modeled. We then propose a relaxation of the oracle method—Pythia—that circumvents its ground-truth requirements. Finally, we conduct an extensive evaluation of existing state-of-the-art methods, some of which are shown in Figure 1c, highlighting their shortcomings in handling scenes with complex refractive and reflective properties. Our contributions are:

1. a synthetic dataset for 3D reconstruction of scenes with refractive and reflective objects;
2. an oracle method that models light paths using ground-truth object geometry and refractive indices;
3. a method that relaxes these requirements by estimating and smoothing the object geometry; and
4. a benchmark evaluating state-of-the-art methods on this dataset, revealing the limits of existing approaches at handling complex optical phenomena.

2 Related Work

Neural 3D reconstruction. Neural Radiance Field (NeRF) [28] optimizes the parameters of a coordinate network to map from spatial position and viewpoint to color and density. To improve efficiency and accuracy, Zip-NeRF [2] combines a feature grid with anti-aliasing. Other approaches fit implicit geometry representations through volume rendering, including UNISURF [31], VolSDF [47], and NeuS [41], which model 3D surfaces using occupancy or signed distance fields. Another line of approaches rasterize 3D Gaussians [19, 34], allowing for real-time rendering with high visual quality. However, while some methods have explored single reflections [39, 27, 36, 32, 15], none explicitly handles the curved light paths necessary for modeling refraction.

Transparent object modeling. Modeling refractive and reflective materials is a significant challenge due to the complexity of the light paths [51, 23, 48, 12, 40, 14, 32, 17]. Some approaches focus on light transport in dynamic water surfaces and underwater scenes [46], while others reconstruct opaque objects in a single refractive medium [37, 50, 33, 4]. Moenne-Loccoz et al. [29] and Wu et al. [44] perform ray tracing on volumetric Gaussian particles, enabling efficient simulation of secondary rays required for rendering phenomena such as reflection, refraction, and shadows. Li et al. [23] assume known environment illumination and refractive indices, integrating rendering and cost volume layers to model reflection and refraction for precise point cloud reconstruction. Controlled experimental setups, such as using gray-coded patterns to determine ray–position correspondences, have improved surface reconstruction accuracy [26, 43, 24], but cannot be assumed in general. MS-NeRF [48] introduces multi-space feature fields to jointly model multiple subspaces, such as the real

world and the world reflected in a mirror, giving it some capacity to model refraction. NEMTO [40] directly predicts the exiting light direction by treating the internal refraction and reflection processes as a black box and assuming an infinitely-distant background. As a result, it cannot handle the general reconstruction of posed images. Yoon and Lee [49] use a visual hull method to approximate the object’s shape and then consider two-bounce light paths. While this improves view synthesis results, it suffers from voxelization artifacts and limited refraction modeling. Ray Deformation Networks [12] propose a deformation field to predict light bending in refractive objects, which works well for small levels of refraction and low-frequency shapes. TNSR [11] builds on NeuS by integrating ray tracing and sphere tracing, improving view synthesis and geometry refinement; a similar strategy is taken by Gao et al. [14]. Overall, most existing methods assume only two refractions and one reflection, limiting their ability to model complex light interactions such as total internal reflection and multiple successive refractions, which frequently occur in real-world transparent and reflective objects.

Datasets. As shown in Table 1, several datasets have been widely adopted for 3D reconstruction tasks, each offering unique strengths for general scene understanding. The DTU dataset [18] provides images across multiview setups. Tanks and Temples [20] provides a benchmark for 3D reconstruction with complex scenes captured using high-precision scanners, suitable for evaluating methods in real-world settings. Additionally, the Objaverse-XL [10], ShapeNet [5], and OmniObject3D [45] datasets extend the scale of the data but restrict the scope to object-centric reconstruction. These datasets contain too few refractive and reflective objects and so are not suitable for evaluating in this domain.

Several datasets address the unique challenges posed by refractive and reflective objects. The Shiny dataset [42] introduces complex view-dependent effects, such as rainbow reflections and refractions through glassware, designed to evaluate view synthesis under challenging conditions, however, it only includes 8 scenes, mostly with opaque objects. OpenMaterial [9] is a synthetic dataset that offers 295 distinct materials, but only half of its materials exhibit reflective or refractive properties, and it lacks multi-material objects with different refractive indices, limiting its realism for scenes involving complex object interactions. Unlike the above datasets, our RefRef dataset is designed to benchmark 3D reconstruction methods for handling refractive and reflective objects. It contains 50 objects with varying materials, including single-material and multi-material objects with different refractive indices and tints, providing a wide range of challenges.

3 A Refractive–Reflective Object Dataset

Dataset structure. The dataset contains 50 unique objects, each placed in three different backgrounds, resulting in 150 scenes. The objects are categorized into three categories based on their geometry (convex or non-convex) and refractive material count (single or multiple). This categorization facilitates evaluation of reconstruction methods at a range of difficulty levels.

1. **Single-material convex (27 scenes).** Objects with convex geometries, each composed of a single refractive material, such as transparent cubes, balls, cylinders, and pyramids.
2. **Single-material non-convex (60 scenes).** Objects with non-convex geometries, each composed of a single refractive material, such as animal sculptures, glass jars, light bulbs, and magnifiers.
3. **Multiple-materials non-convex (63 scenes).** Objects with non-convex geometries, each composed of multiple refractive materials or a combination of refractive and opaque materials, such as reed diffusers, a glass of wine, and flasks filled with liquid chemicals.

Scene backgrounds and cameras. For each object, we generate three background environments to enhance variability in the rendered scenes: a cube background, a sphere background, and an HDR environment map. Each cube background scene is constructed by randomly selecting 6 images from a pool of 24 highly textured images, while each sphere background is created by choosing 1 image from a set of 13, and the HDR environment map features an outdoor scene to provide realistic lighting

Table 1: Comparison of datasets by number of scenes $|\mathcal{S}|$, images $|\mathcal{I}|$, presence of refractive and reflective objects, ground-truth geometry, and multi-material composition.

Datasets	$ \mathcal{S} $	$ \mathcal{I} $	Refraction	Reflection	Geometry	Multi-mat.
DTU [18]	124	4.2k	✗	✗	✓	✓
T & T [20]	7	88k	✗	✗	✓	✓
ShapeNet [5]	5.1k	0	✗	✗	✓	✓
Omniobject3D [45]	6k	0	✗	✗	✓	✓
ObjaverseXL [10]	10M	0	✗	✗	✓	✓
Shiny [42]	8	879	✓	✓	✗	✗
OpenMaterial [9]	1k	90k	✓	✓	✓	✗
RefRef (Ours)	150	45k	✓	✓	✓	✓

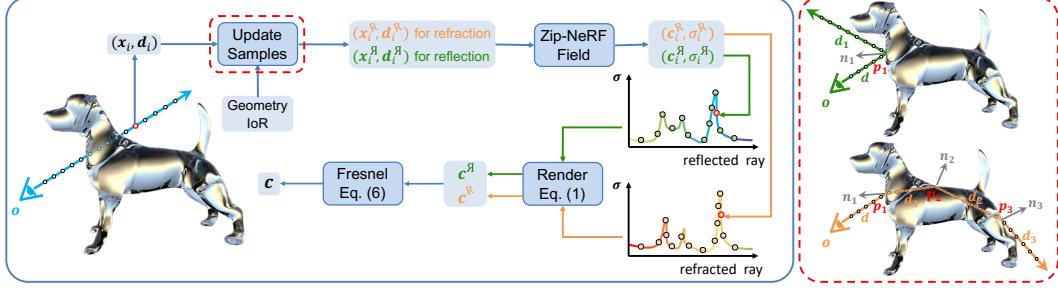


Figure 2: Overview of the oracle method. The process begins with the generation of a straight ray (blue arrow) and sample points along its path. The scene’s geometry and refractive index (IoR) are then used to update the ray trajectory, as detailed in the red dashed box on the right. Here, refracted and reflected rays are handled separately, resulting in two sets of updated sample point positions and directions, which are subsequently processed by the Zip-NeRF field that predicts the color c_i and density σ_i for each sample point. Using eq. (1), the final color along each ray is rendered and the refraction and reflection paths are combined via eq. (6). Note that Zip-NeRF samples points in a conical spiral; we only visualize the centerline of the cone for clarity.

conditions. This yields a total of 150 unique scenes. The images for each scene are subdivided into training, validation, and test sets. Each set consists of 100 images at a resolution of 800×800 pixels, accompanied by metadata, including camera positions, depth maps, 3D object models, and object masks. The camera viewpoints for the training and validation sets are randomly selected on a sphere centered around the object, ensuring diverse perspectives. For the test set, we employ a helical path, capturing 100 viewpoints by gradually ascending the camera position around the object.

Rendering details. The dataset was rendered using Blender [8]. To accurately capture light interactions with refractive and reflective materials, Blender simulates physics-based light transport, including refraction, reflection, total internal reflection, and absorption.

4 A Volume Rendering Oracle Method

Given a set of posed RGB images of a scene, NeRF-based [28] methods typically estimate the geometry and appearance by learning a volumetric representation that can render images from novel views. However, most NeRF-based methods struggle with scenes involving refractive and reflective objects due to their assumption of straight light paths [12]. To evaluate how well a NeRF-based model can perform in reconstructing scenes with refractive and reflective objects, we present an oracle method that has access to the ground-truth geometry and refractive indices, as shown in Figure 2. The method assumes piecewise linear light paths (i.e., piecewise constant refractive indices), a single explicit reflection occurring at the first surface intersection, and a maximum of 10 refractions or total internal reflections. Note that NeRF-based view-dependent color prediction can allow for additional implicit reflections. For real refractive objects, these assumptions are not very restrictive.

4.1 NeRF Preliminaries

The Neural Radiance Field (NeRF) method [28] parametrizes a 3D scene as a radiance field coordinate network ϕ_θ , which maps a 5 DoF position and view direction $(\mathbf{x}_i, \mathbf{d})$ to volume density σ_i and view-dependent color \mathbf{c}_i . The parameters θ of ϕ_θ are optimized with respect to a photometric loss that compares renders of the neural field to the ground-truth images. To render an image, NeRF integrates the predicted colors $\mathbf{c}_i(\mathbf{x}_i, \mathbf{d})$ and densities $\sigma_i(\mathbf{x}_i)$ along each camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where \mathbf{o} and \mathbf{d} denote the camera origin and direction, and t ranges from the near plane t_n to the far plane t_f . The rendered color $\mathbf{c}(\mathbf{r})$ of each pixel is approximated as the w_i -weighted sum of the colors \mathbf{c}_i at N sample points along the ray,

$$\mathbf{c}(\mathbf{r}) = \sum_{i=1}^N \underbrace{T_i(1 - \exp(-\sigma_i \Delta t_i))}_{w_i} \mathbf{c}_i, \quad (1)$$

where $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \Delta t_j\right)$ denotes the accumulated transmittance up to sample i , and Δt_i denotes the distance between consecutive samples.

4.2 Modeling Refractions and Reflections

This section outlines how the piecewise linear refracted and reflected light trajectories are computed, sampled, integrated along, and combined in our oracle method. It extends the robust Zip-NeRF [2] model to handle refraction and explicit reflection, as shown in Figure 2.

The oracle method represents light paths as piecewise linear functions parametrized by $K + 1$ points $\{\mathbf{p}_i\}_{i=0}^K$ and unit direction vectors $\{\mathbf{d}_i\}_{i=0}^K$,

$$\mathbf{r}(t) = \sum_{i=0}^K \llbracket t \in [\tau_i, \tau_{i+1}) \rrbracket (\mathbf{p}_i + (t - \tau_i)\mathbf{d}_i), \quad (2)$$

where $\llbracket \cdot \rrbracket$ is an Iverson bracket, $\mathbf{p}_0 = \mathbf{o}$, $\mathbf{d}_0 = \mathbf{d}$ and the cumulative distance is given by

$$\tau_i = \begin{cases} 0 & \text{for } i = 0 \\ \infty & \text{for } i = K + 1 \\ \sum_{j=1}^i \|\mathbf{p}_j - \mathbf{p}_{j-1}\| & \text{otherwise.} \end{cases} \quad (3)$$

The method considers two paths: a (multiple) refraction path \mathbf{r}^R and a (single) reflection path \mathbf{r}^A . We next show how to compute the parameters for each.

Refraction and reflection parameters. For the refraction ray \mathbf{r}^R , given a position \mathbf{p}_i and direction \mathbf{d}_i , the next intersection \mathbf{p}_{i+1} with the refractive object is computed using ray tracing with the known ground-truth object geometry. Let $\alpha_i = \nu_i / \nu_{i+1}$, $\beta_i = -\mathbf{d}_i^\top \mathbf{n}(\mathbf{p}_{i+1})$, $\gamma_i^2 = 1 - \alpha_i^2(1 - \beta_i^2)$, ν_i be the refractive index of the i^{th} medium, and $\mathbf{n}(\mathbf{x})$ be the ground-truth unit normal vector at location \mathbf{x} . Then if $\gamma_i^2 \geq 0$, refraction will occur at the interface and the next direction \mathbf{d}_{i+1} is computed using Snell's Law [3],

$$\mathbf{d}_{i+1} = \alpha_i \mathbf{d}_i + (\alpha_i \beta_i - \gamma_i) \mathbf{n}(\mathbf{p}_{i+1}). \quad (4)$$

Otherwise, total internal reflection will occur and the direction is given by the Law of Reflection,

$$\mathbf{d}_{i+1} = \mathbf{d}_i - 2(\mathbf{d}_i^\top \mathbf{n}(\mathbf{p}_{i+1}))\mathbf{n}(\mathbf{p}_{i+1}). \quad (5)$$

For the reflection ray \mathbf{r}^A , only the reflection at the first surface intersection is considered. The first reflected direction, \mathbf{d}_1^A , is given by eq. (5). The proposed method does not explicitly model any other reflections (except for total internal reflections), because they are computationally expensive and have negligible impact in most situations.

Sampling and rendering. For neural rendering, our model samples points along the refraction and reflection paths using the proposal sampler from Mip-NeRF 360 [1], which first uniformly samples along the path and then uses the computed probability density function to concentrate samples in higher density regions. The neural field is queried at each sample location \mathbf{x}_i , using the corresponding direction vector \mathbf{d}_i at that location (unlike standard NeRF that uses a constant direction \mathbf{d}). That is, we obtain $(\mathbf{c}_i^R, \sigma_i^R) = \phi(\mathbf{x}_i^R, \mathbf{d}_i^R)$ and $(\mathbf{c}_i^A, \sigma_i^A) = \phi(\mathbf{x}_i^A, \mathbf{d}_i^A)$ for all sample points on both paths. We then apply eq. (1) to obtain the colors \mathbf{c}^R and \mathbf{c}^A . The refractive and reflective color contributions are combined using the Fresnel equations [16],

$$\mathbf{c}' = R(\mathbf{c}^A - \mathbf{c}^R) + \mathbf{c}^R, \quad \text{where} \quad R = \frac{1}{2}(R_p + R_s), \quad (6a)$$

$$R_p = \left(\frac{\nu_1 \beta_0 - \nu_0 \gamma_0}{\nu_1 \beta_0 + \nu_0 \gamma_0} \right)^2, \quad R_s = \left(\frac{\nu_0 \beta_0 - \nu_1 \gamma_0}{\nu_0 \beta_0 + \nu_1 \gamma_0} \right)^2, \quad (6b)$$

where R_p and R_s are the reflection coefficients for parallel and perpendicular polarized light. Finally, the color \mathbf{c}' is converted to a non-linear sRGB space to obtain the predicted color $\hat{\mathbf{c}}$ [38].

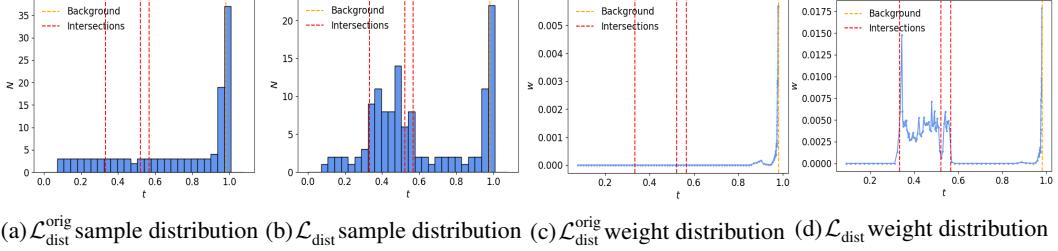


Figure 3: Comparison of the original [1] distortion loss $\mathcal{L}_{\text{dist}}^{\text{orig}}$ and the translucency-corrected distortion loss $\mathcal{L}_{\text{dist}}$. (a)–(b) Distribution of sample points. (c)–(d) Distribution of weights.

4.3 Optimization

The parameters θ of the coordinate network ϕ_θ are optimized with respect to a photometric loss \mathcal{L}_{rgb} , an anti-aliased interlevel loss \mathcal{L}_{int} , and a modified distortion loss $\mathcal{L}_{\text{dist}}$. The per-pixel photometric loss is given by the mean squared color error,

$$\mathcal{L}_{\text{rgb}}(\hat{\mathbf{c}}, \mathbf{c}) = \frac{1}{3} \|\hat{\mathbf{c}} - \mathbf{c}\|_2^2, \quad (7)$$

where \mathbf{c} is the ground-truth pixel color. The interlevel loss \mathcal{L}_{int} [2] encourages consistency between the proposal network, used for rapidly determining where best to sample points, and the main network.

The distortion loss $\mathcal{L}_{\text{dist}}^{\text{orig}}$ introduced in Mip-NeRF 360 [1] encourages the weight distribution along a ray to coalesce and otherwise tend to zero. This leads to a preference for a single high-weight peak, that is, a single opaque surface. This is desirable for standard NeRF settings, since it reduces ‘floaters’ and background collapse. However, it is not applicable for translucent objects, where color contributions along a ray arise from both the translucent and opaque media. As shown in Figure 3 (left), the unmodified loss tends to reduce the density of translucent objects to zero, which has the side-effect of reducing the number of samples within the object, due to proposal sampling. Thus, we propose a modified distortion loss that excludes samples within the refractive object, given by

$$\mathcal{L}_{\text{dist}}(\mathbf{s}, \mathbf{w}) = \sum_{i,j \in \mathcal{I}_{\text{out}}} w_i w_j \left| \frac{s_i + s_{i+1}}{2} - \frac{s_j + s_{j+1}}{2} \right| + \frac{1}{3} \sum_{i,j \in \mathcal{I}_{\text{out}}} w_i^2 (s_{i+1} - s_i), \quad (8)$$

where s_i denotes the normalized ray distance [1] to the i^{th} sample point, and \mathcal{I}_{out} is the set of indices outside the refractive object. As shown in Figure 3 (right), after applying the corrected distortion loss, the model allocates more samples with higher weights within the refractive object, leading to more accurate reconstruction. The overall loss is then

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_1 \mathcal{L}_{\text{int}} + \lambda_2 \mathcal{L}_{\text{dist}}, \quad (9)$$

where hyperparameters λ_1 and λ_2 weight the relative loss contributions and each loss term averages the per-pixel losses across the dataset.

5 Pythia: A Relaxed Oracle Method

Given the previously described oracle method, the requirements of ground-truth object geometry and refractive index can be relaxed in a relatively straightforward manner. We name the resulting 3D reconstruction method Pythia, as a relaxed version of the oracle method. The geometry of the refractive object may be estimated using a modern variant of the visual hull algorithm [22], where posed object masks are given as inputs to the UNISURF implicit surface model [31]. However, the resulting surfaces are insufficiently smooth for computing refracted light paths, since slight aberrations in the normal directions can cause large visual differences. We apply automatic post-processing to smooth and refine the surface, detailed in Appendix B. This provides the object geometry required by our oracle method. The refractive index, if unavailable, may be very precisely estimated using the approach outlined in TNSR [11]. While effective, Pythia is limited to reconstructing the visual hull of the object, and so is not suitable for the multi-material category and objects with holes in the RefRef dataset, where internal structures need to be considered.

Table 2: Quantitative results on the RefRef test set. We report view synthesis metrics (PSNR, masked PSNR, SSIM, LPIPS) and geometry accuracy using the distance mean absolute error (DMAE). For natural backgrounds, DMAE is masked since the background is infinitely distant.

Method	Cube Background				Sphere Background				Natural Background				All Backgrounds								
	PSNR	PSNR _M	SSIM	LPIPS	PSNR	PSNR _M	SSIM	LPIPS	PSNR	PSNR _M	SSIM	LPIPS	DMAE _M	PSNR	PSNR _M	SSIM	LPIPS	DMAE			
	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓	↓	↑	↑	↑	↓	↓			
Convex 1-mat	NeuS [41]	18.48	14.94	0.66	0.19	1.20	20.77	14.88	0.60	0.16	1.81	—	—	—	19.62	14.91	0.63	0.18	1.50		
	Splatfacto [19]	22.32	14.90	0.87	0.11	14.56	21.67	15.00	0.88	0.16	33.41	11.88	13.05	0.62	0.71	1.17	18.88	14.36	0.80	0.31	16.96
	Zip-NeRF [2]	22.16	14.88	0.86	0.13	0.10	22.95	15.70	0.91	0.13	0.22	33.04	26.52	0.94	0.10	0.08	25.78	18.75	0.90	0.12	0.17
	TNSR [11]	19.30	11.92	0.84	0.12	0.90	18.62	11.66	0.85	0.17	1.62	—	—	—	18.96	11.79	0.85	0.14	1.26		
	MS-NeRF [48]	21.60	14.00	0.85	0.12	0.07	21.42	14.58	0.85	0.18	1.44	26.56	21.03	0.81	0.37	1.93	23.06	16.36	0.84	0.22	1.12
	RayDef [12]	21.85	14.37	0.84	0.15	0.15	21.16	14.68	0.84	0.22	1.08	26.96	24.72	0.77	0.37	0.08	23.19	17.66	0.82	0.24	0.45
	RoseNeRF [25]	20.18	14.02	0.82	0.32	—	22.33	15.22	0.88	0.14	—	23.54	26.35	0.77	0.51	—	22.67	17.94	0.81	0.31	—
Non-convex 1-mat	Pythia	23.55	16.49	0.86	0.12	0.08	25.08	18.10	0.90	0.12	0.18	30.91	24.17	0.93	0.13	0.01	26.51	19.58	0.90	0.12	0.09
	Oracle	31.64	25.37	0.96	0.03	0.04	32.87	26.14	0.96	0.03	0.19	33.48	26.85	0.96	0.08	0.00	32.67	26.12	0.96	0.05	0.08
	NeuS [41]	19.11	14.86	0.67	0.13	1.17	20.72	13.21	0.62	0.13	1.97	—	—	—	19.92	14.03	0.64	0.13	1.57		
	Splatfacto [19]	24.40	16.02	0.88	0.07	0.94	22.07	15.30	0.85	0.17	26.30	10.71	11.64	0.51	0.76	0.97	19.17	14.36	0.75	0.33	12.73
	Zip-NeRF [2]	24.15	15.87	0.88	0.08	0.09	24.37	15.91	0.89	0.09	0.20	27.17	19.10	0.89	0.14	0.15	25.23	16.96	0.89	0.10	0.18
	TNSR [11]	18.92	11.25	0.83	0.14	1.31	19.38	11.85	0.83	0.16	1.62	—	—	—	19.22	11.57	0.83	0.15	1.47		
	MS-NeRF [48]	23.83	15.74	0.87	0.10	0.26	22.68	15.27	0.85	0.15	1.28	24.06	16.82	0.75	0.42	1.92	23.56	15.96	0.82	0.22	0.94
Non-convex	RayDef [12]	22.69	14.98	0.83	0.18	0.48	21.77	14.82	0.82	0.20	1.23	24.29	19.43	0.70	0.43	0.08	22.94	16.46	0.78	0.26	0.59
	RoseNeRF [25]	22.83	14.97	0.81	0.15	—	22.49	15.12	0.85	0.13	—	23.76	17.15	0.72	0.58	—	23.02	16.21	0.79	0.24	—
	Pythia	23.17	15.12	0.87	0.10	0.11	23.27	15.03	0.88	0.11	0.25	26.54	18.66	0.87	0.20	0.03	24.33	16.27	0.87	0.14	0.13
	Oracle	27.81	19.86	0.92	0.06	0.04	28.69	20.39	0.93	0.05	0.15	29.46	21.24	0.92	0.11	0.00	28.66	20.49	0.93	0.07	0.06
	NeuS [41]	19.19	15.97	0.63	0.20	1.31	19.49	14.16	0.61	0.20	1.99	—	—	—	19.35	15.11	0.62	0.19	1.62		
	Splatfacto [19]	24.70	17.72	0.86	0.10	11.03	24.53	17.95	0.89	0.11	19.93	10.60	10.55	0.54	0.76	1.13	20.10	15.53	0.76	0.32	10.61
	Zip-NeRF [2]	25.61	18.05	0.88	0.09	0.11	26.10	18.36	0.90	0.09	0.24	29.55	22.58	0.91	0.14	0.19	27.09	19.66	0.89	0.11	0.21
Non-convex	TNSR [11]	17.66	10.06	0.81	0.16	1.19	17.69	10.61	0.80	0.21	2.20	—	—	—	17.94	10.44	0.82	0.17	1.62		
	MS-NeRF [48]	24.93	17.89	0.85	0.12	0.39	20.94	15.27	0.78	0.27	2.56	25.99	20.36	0.78	0.40	0.19	23.64	16.84	0.82	0.24	1.02
	RayDef [12]	23.94	16.66	0.84	0.14	0.34	21.10	15.56	0.80	0.27	1.60	24.85	22.09	0.72	0.43	0.12	23.29	18.12	0.78	0.28	0.69
	RoseNeRF [25]	23.16	15.94	0.80	0.16	—	20.20	16.56	0.78	0.29	—	22.72	21.91	0.73	0.55	—	22.78	18.27	0.75	0.30	—
	Pythia	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	Oracle	27.45	20.02	0.91	0.08	0.04	27.76	20.05	0.92	0.08	0.14	30.82	23.61	0.92	0.13	0.00	28.67	21.23	0.92	0.10	0.06
	NeuS [41]	19.15	15.33	0.65	0.17	1.23	20.15	13.97	0.61	0.16	1.93	—	—	—	19.62	14.64	0.63	0.19	1.62		
Entire dataset	Splatfacto [19]	24.26	16.69	0.87	0.09	10.96	23.03	16.36	0.87	0.14	24.90	10.86	11.39	0.54	0.75	1.08	19.53	14.87	0.77	0.32	12.55
	Zip-NeRF [2]	24.41	16.60	0.88	0.09	0.10	24.84	16.90	0.90	0.10	0.22	29.15	21.80	0.91	0.13	0.16	26.11	18.41	0.89	0.11	0.20
	TNSR [11]	18.78	11.02	0.83	0.18	1.10	18.51	11.28	0.82	0.18	1.87	—	—	—	18.64	11.14	0.83	0.16	1.49		
	MS-NeRF [48]	23.98	16.43	0.86	0.11	0.22	21.72	15.15	0.82	0.21	1.84	25.32	19.07	0.77	0.40	1.93	23.64	16.84	0.82	0.24	1.02
	RayDef [12]	23.07	15.61	0.83	0.15	0.37	21.37	15.11	0.81	0.23	1.36	24.96	21.43	0.72	0.42	0.10	23.13	17.38	0.79	0.27	0.61
	RoseNeRF [25]	22.49	15.07	0.83	0.18	—	21.50	15.74	0.83	0.20	—	23.28	20.81	0.73	0.55	—	22.86	17.39	0.78	0.28	—
	Pythia	23.28	15.51	0.86	0.11	0.10	23.79	15.90	0.89	0.11	0.23	27.79	20.24	0.88	0.18	0.02	24.95	17.22	0.88	0.13	0.12
	Oracle	28.28	20.82	0.92	0.06	0.04	29.05	21.28	0.93	0.06	0.15	30.69	23.17	0.93	0.11	0.00	29.34	21.75	0.93	0.08	0.07

6 Experiments

6.1 Experimental Setup

Baselines and prior work. Alongside the oracle and Pythia methods, we evaluate several state-of-the-art approaches: Neural Surface Reconstruction (NeuS) [41], Splatfacto [19], Zip-NeRF [2], Multi-Space Neural Radiance Fields (MS-NeRF) [48], Ray Deformation Networks (RayDef) [12], Transparent Neural Surface Refinement (TNSR) [11], and RoseNeRF [25]. The first three use the standard straight line light path assumption, while the others model refraction and reflection.

Metrics. To assess the quality of the novel views, we report the peak signal-to-noise ratio (PSNR) between the rendered and ground-truth images, the masked PSNR (PSNR_M) that masks out background pixels to focus on the refractive objects, the structural similarity index measure (SSIM), and the learned perceptual image patch similarity (LPIPS), which is more indicative of differences to the human eye than PSNR, especially with respect to blur. To assess the geometric fidelity, we report the distance mean absolute error (DMAE) between the estimated and ground-truth distance maps for the novel views. Importantly, standard weighted-sum distance rendering techniques [28] cannot be used for translucent objects, since the weights are distributed across the translucent medium and opaque background. Instead, we take the median distance of the weight samples along each ray, as a robust estimate of the distance to the nearest (potentially translucent) surface.

Implementation details. Oracle and Pythia both extend Zip-NeRF [2] to allow piecewise linear light paths and explicit reflection. For bounded scenes, we set near plane $t_n=0.05$, far plane $t_f=15$, and distortion loss weight 0.01. For unbounded scenes, we use $t_f=1000$ and a contraction warp function. Models are trained for 25k iterations on a single A6000 GPU with batch size 4096. Code and dataset will be released. Further implementation details are provided in Appendix B.

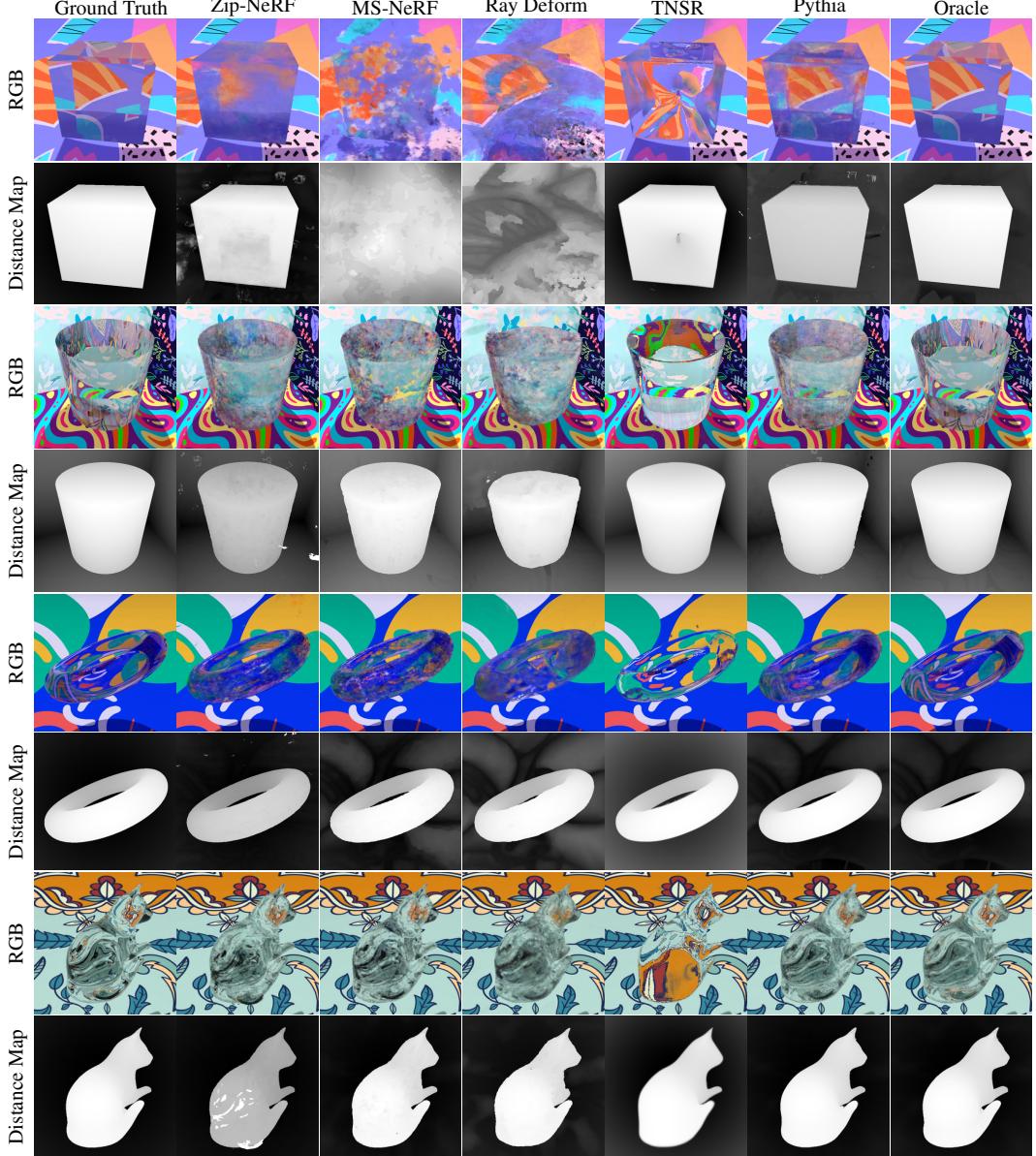


Figure 4: Qualitative comparison of novel view synthesis and distance maps across four scenes using Zip-NeRF [2], MS-NeRF [48], Ray Deformation Network [12], TNSR [11], Pythia, and Oracle. Pythia and Oracle render more accurate results, especially in scenes with multiple refractions and total internal reflection, where other methods often fail.

6.2 Results

The quantitative results are reported in Table 2, for the three data subsets in section 3. Pythia performs strongly in the single-material convex object category, outperforming all other methods except the oracle (which receives privileged information). However, it struggles with handling objects with holes—a consequence of its reliance on a variant of the visual hull. The other methods perform reasonably well on simple scenes (e.g., convex objects, natural environment map backgrounds), but perform significantly worse for harder objects and background types. The object geometry is particularly poorly estimated, showing that the models are taking shortcuts either by deforming the geometry to fit the refracted appearance or by placing floaters around objects. This reveals the difficulty of refractive reconstruction and the usefulness of the oracle method as a benchmark.

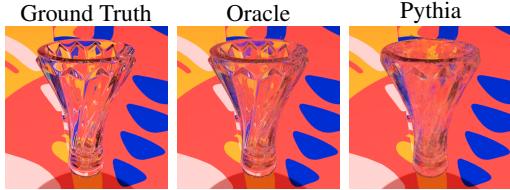


Figure 5: Failure case of the oracle and Pythia methods. The oracle method, despite access to ground-truth geometry and refractive indices, struggles to model the vase’s uneven surface. Meanwhile, Pythia treats the vase as solid, causing appearance distortions near the top.

Table 3: Ablation study of the oracle method on the single-material convex object category (cube background). The ablated components are the distortion loss correction, first surface reflection, total internal reflection, and Zip-NeRF [2] backbone (substituting Nerfacto [35]).

Method	PSNR \uparrow	PSNR _M \uparrow	SSIM \uparrow	LPIPS \downarrow $\times 10^{-2}$	DMAE \downarrow $\times 10^{-2}$
Oracle	31.64	25.37	96.13	2.76	4.38
w/o corrected $\mathcal{L}_{\text{dist}}$	31.59	25.31	96.12	2.78	4.07
w/o \mathfrak{I}	29.65	22.92	94.69	3.67	2.97
w/o TIR	25.85	19.27	90.84	8.38	15.86
w/o Zip-NeRF	24.68	21.45	87.79	11.63	27.42

Qualitative results are presented in Figure 4. A clear performance gap is observed between the oracle method and existing state-of-the-art approaches. In scenes involving multiple refractions and total internal reflections (TIR), most methods produce blurry or incorrect outputs. In contrast, the oracle and Pythia methods are better able to capture these complex light interactions, although Pythia may exhibit artifacts due to inaccuracies in geometry estimation. Additional qualitative results are provided in Appendix E.

Furthermore, both the oracle method and Pythia exhibit failure cases, particularly when reconstructing objects with complex structures, as shown in Figure 5. The oracle method, with access to ground-truth geometry, can capture fine details such as holes. However, it still struggles to model highly uneven surfaces, as demonstrated with the vase example. Pythia relies on estimated geometry generated by a UniSurf model [31] using posed object masks, a variant of the visual hull algorithm [22]. This prevents it from modeling internal cavities, resulting in solid geometries and rendering artifacts, as also shown in Figure 5. These examples highlight the difficulty of handling intricate objects with hidden or internal structures, where accurate geometry estimation is critical for realistic reconstruction. They underscore the broader challenges of modeling refractive objects, even when ground-truth geometry and precise refractive indices are available, and point to the need for further advances in reconstruction methods capable of handling complex light transport.

6.3 Ablation Study

An ablation study is presented in Table 3, comparing the full oracle method with four ablated versions: without the corrected distortion loss, first surface reflection, total internal reflection (TIR), and replacing the Zip-NeRF [2] backbone with Nerfacto [35]). The full oracle method accurately models both reflections and TIR, producing highly detailed and realistic renderings. In contrast, removing the corrected distortion loss reduces the number of sample points within transparent objects, as illustrated in Figure 3. Omitting the first surface reflection leads to the loss of subtle reflective details, while disabling TIR results in missing critical light interactions within refractive objects. Additionally, replacing the Zip-NeRF backbone with Nerfacto results in lower performance. The quantitative results in Table 3 further support these observations, showing a noticeable performance drop when any of these components is ablated.

7 Discussion and Conclusion

We have presented the RefRef dataset for 3D reconstruction and novel view synthesis of scenes containing refractive and reflective objects. To establish performance bounds, we introduce an oracle method based on ground-truth geometry and refractive indices, as well as a more practical alternative, Pythia, that relaxes these assumptions. Benchmarking state-of-the-art methods revealed significant performance gaps, even for models explicitly designed to handle nonlinear light paths. More surprisingly, the oracle method exhibits several limitations despite its fairly mild assumptions (a maximum of ten bounces, a single explicit reflection). This highlights the high sensitivity of light transport to geometric inaccuracies—small errors in surface normals can cause large deviations in ray paths. These results point to the need for new reconstruction methods that can more reliably account for complex light interactions in transparent and reflective scenes.

References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022. 1, 5, 6, 13
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, pages 19697–19705, 2023. 2, 5, 6, 7, 8, 9, 13, 15, 16
- [3] Max Born and Emil Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Elsevier, 2013. 5
- [4] Matthew Cassidy, Jean Mélou, Yvain Quéau, François Lauze, and Jean-Denis Durou. Refractive multi-view stereo. In *International Conference on 3D Vision*, pages 384–393, 2020. 2
- [5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015. 3
- [6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields. In *ECCV*, 2022. 1
- [7] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. 13
- [8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2024. 4
- [9] Zheng Dang, Jialu Huang, Fei Wang, and Mathieu Salzmann. OpenMaterial: A Comprehensive Dataset of Complex Materials for 3D Reconstruction. *arXiv preprint arXiv:2406.08894*, 2024. 3
- [10] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-XL: A Universe of 10M+ 3D Objects. *NeurIPS*, 36, 2024. 3
- [11] Weijian Deng, Dylan Campbell, Chunyi Sun, Shubham Kanitkar, Matthew Shaffer, and Stephen Gould. Differentiable neural surface refinement for transparent objects. In *CVPR*, 2024. 2, 3, 6, 7, 8, 14, 15
- [12] Weijian Deng, Dylan Campbell, Chunyi Sun, Shubham Kanitkar, Matthew Shaffer, and Stephen Gould. Ray Deformation Networks for Novel View Synthesis of Refractive Objects. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. 1, 2, 3, 4, 7, 8, 14, 15, 16
- [13] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999. 13
- [14] Fangzhou Gao, Lianghao Zhang, Li Wang, Jiamin Cheng, and Jiawan Zhang. Transparent object reconstruction via implicit differentiable refraction rendering. In *SIGGRAPH Asia*, pages 57:1–57:11, 2023. 2, 3
- [15] Yufei Han, Heng Guo, Koki Fukai, Hiroaki Santo, Boxin Shi, Fumio Okura, Zhanyu Ma, and Yunpeng Jia. Nersp: Neural 3d reconstruction for reflective objects with sparse polarized images. In *CVPR*, pages 11821–11830, 2024. 2
- [16] Eugene Hecht. *Optics*. Pearson Education India, 2012. 5
- [17] 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*, 2025. 2
- [18] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large Scale Multi-view Stereopsis Evaluation. In *CVPR*, pages 406–413, 2014. 3
- [19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4), 2023. 2, 7, 14
- [20] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM TOG*, 36(4), 2017. 3

- [21] Leif P Kobbelt, Mario Botsch, Ulrich Schwanecke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 57–66, 2001. 13
- [22] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE TPAMI*, 16(2):150–162, 1994. 6, 9
- [23] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *CVPR*, pages 1259–1268, 2020. 2
- [24] Zongcheng Li, Xiaoxiao Long, Yusen Wang, Tuo Cao, Wenping Wang, Fei Luo, and Chunxia Xiao. Neto: Neural reconstruction of transparent objects with self-occlusion aware refraction-tracing. In *ICCV*, pages 18501–18511, 2023. 2
- [25] Xiaoqian Liang, Jianji Wang, Yuanliang Lu, Xubin Duan, Xichun Liu, and Nanning Zheng. Refracting once is enough: Neural radiance fields for novel-view synthesis of real refractive objects. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, page 694–703, New York, NY, USA, 2024. Association for Computing Machinery. 7, 13
- [26] Jiahui Lyu, Bojian Wu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Differentiable refraction-tracing for mesh reconstruction of transparent objects. *ACM TOG*, 39(6):195:1–195:13, 2020. 2
- [27] Li Ma, Vasu Agrawal, Haithem Turki, Changil Kim, Chen Gao, Pedro V. Sander, Michael Zollhöfer, and Christian Richardt. Specnerf: Gaussian directional encoding for specular reflections. In *CVPR*, pages 21188–21198, 2024. 2
- [28] 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. 1, 2, 4, 7
- [29] 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 and SIGGRAPH Asia*, 2024. 2
- [30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM TOG*, 41(4):102:1–102:15, 2022. 1
- [31] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. In *ICCV*, pages 5589–5599, 2021. 2, 6, 9
- [32] Jiaxiong Qiu, Peng-Tao Jiang, Yifan Zhu, Ze-Xin Yin, Ming-Ming Cheng, and Bo Ren. Looking through the glass: Neural surface reconstruction against high specular reflections. In *CVPR*, pages 20823–20833, 2023. 2
- [33] Jia-Mu Sun, Tong Wu, Ling-Qi Yan, and Lin Gao. Nu-nerf: Neural reconstruction of nested transparent objects with uncontrolled capture environment. *ACM TOG*, 43(6):262:1–262:14, 2024. 2
- [34] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3D: Feed-forward generalisable 3D scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 2
- [35] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 9, 15, 16
- [36] Kushagra Tiwary, Akshat Dave, Nikhil Behari, Tzofi Klinghoffer, Ashok Veeraraghavan, and Ramesh Raskar. Orca: Glossy objects as radiance-field cameras. In *CVPR*, pages 20773–20782, 2023. 2
- [37] 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 *CVPR*, pages 12555–12564, 2023. 2
- [38] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields. In *CVPR*, pages 5481–5490. IEEE, 2022. 5
- [39] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *CVPR*, pages 5481–5490. IEEE, 2022. 2

- [40] Dongqing Wang, Tong Zhang, and Sabine Süsstrunk. Nemto: Neural environment matting for novel view and relighting synthesis of transparent objects. In *ICCV*, pages 317–327, 2023. 2, 3
- [41] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *NeurIPS*, 2021. 2, 7, 14, 15
- [42] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. NeX: Real-time View Synthesis with Neural Basis Expansion. In *CVPR*, pages 8534–8543, 2021. 3
- [43] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3d reconstruction of transparent objects. *ACM TOG*, 37(4):103, 2018. 2
- [44] Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2
- [45] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. OmniObject3D: Large-Vocabulary 3D Object Dataset for Realistic Perception, Reconstruction and Generation. In *CVPR*, 2023. 3
- [46] Jinhui Xiong and Wolfgang Heidrich. In-the-wild single camera 3d reconstruction through moving water surfaces. In *ICCV*, pages 12538–12547, 2021. 2
- [47] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 34:4805–4815, 2021. 2
- [48] Ze-Xin Yin, Jiaxiong Qiu, Ming-Ming Cheng, and Bo Ren. Multi-space neural radiance fields. In *CVPR*, pages 12407–12416, 2023. 2, 7, 8, 13, 15, 16
- [49] Heechan Yoon and Seungkyu Lee. Neural radiance fields for transparent object using visual hull. In *IEEE International Conference on Big Data and Smart Computing*, pages 375–376. 3
- [50] Yifan Zhan, Shohei Nobuhara, Ko Nishino, and Yinqiang Zheng. Nerfrac: Neural radiance fields through refractive surface. In *ICCV*, pages 18356–18366, 2023. 2
- [51] Haoran Zhang, Junkai Deng, Xuhui Chen, Fei Hou, Wencheng Wang, Hong Qin, Chen Qian, and Ying He. From transparent to opaque: Rethinking neural implicit surfaces with α -neus. In *NeurIPS*, 2024. 2
- [52] Denis Zorin, Peter Schröder, and Wim Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 189–192, 1996. 13

A Additional Methodological Details

In this section, we provide additional details on our sampling strategy and how updated sample points are processed for density and color prediction in our oracle method. These extensions complement the core concepts outlined in the main paper and focus on implementation-specific details.

Sampling Strategy. We begin by generating an initial straight ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, which is then split into two paths: a refraction path \mathbf{r}^R and a reflection path \mathbf{r}^R . To sample along these paths, we adopt the proposal sampling strategy introduced in ZipNeRF [2], which builds upon the hierarchical sampling approach of Mip-NeRF 360 [1]. ZipNeRF samples inside a cone following a spiral path, we update the center line of the cone while preserving the sampling pattern. It starts with a uniform sampler that generates N sample points \mathbf{x}_i along the initial straight ray. These points are then updated, resulting in new positions \mathbf{x}_i^R and \mathbf{x}_i^R with corresponding direction vectors \mathbf{d}_i^R and \mathbf{d}_i^R , which follow curved paths for refractions and reflections, respectively. Next, the density σ_i and weight w_i of each sample point along these curved paths are computed. These weights, which represent the contribution of each sample, are further passed through a probability density function sampler to concentrate samples in higher density regions to enhance the accuracy of rendering.

Processing Updated Sample Points. After each sampling stage, the updated sample point positions \mathbf{x}_i^R and \mathbf{x}_i^R are processed through a spatial encoding function ξ for efficient representation. These encoded spatial points are then fed into a multi-layer perceptron (MLP) f_θ , which predicts the scene density σ_i . To obtain the predicted color at each 3D position, another MLP g_θ processes the refined viewing directions \mathbf{d}_i^R and \mathbf{d}_i^R , encoded using spherical harmonics ψ , alongside the computed density σ_i . This yields the emitted color $\mathbf{c}_i = (R, G, B)$ for each sample point.

B Further Implementation Details

For optimizing the oracle method, we use an Adam optimizer with an initial learning rate of 8×10^{-3} , $\epsilon = 1 \times 10^{-15}$, and an exponential decay to 1×10^{-3} over 2.5×10^4 steps, with a 1000-step warm-up. Scene contraction is applied for HDR environment map backgrounds.

For Pythia, geometry post-processing begins with applying a convex hull to all meshes. We identify and remove floaters by computing the maximum convex-to-original vertex distance $\max_i(d_i)$. If this distance exceeds a threshold, we apply the Remove Isolated Pieces filter (90% diameter) in MeshLab [7] to eliminate disconnected or spurious components; otherwise, we retain the convex hull as the final mesh, assuming it sufficiently represents the original shape. In Blender, we further refine the meshes to improve smoothness and geometric fidelity. Specifically, we apply a bevel operation [52] ($0.015m$, 3 segments) to round sharp edges, perform smoothing [13] (factor = 1.0, repeat = 100), and then utilize remeshing [21] (smooth mode, octree depth = 8, scale = 0.9, threshold = 1.0).

C Licenses

Detailed information on the sources and licensing of objects in our dataset is provided in Table 4, covering both objects created by our team and those sourced from online repositories with specific licensing terms. For sourced objects, we made material modifications where necessary to better align with the dataset’s requirements.

D Hyperparameters

In this section, we detail the hyperparameter settings used for training the models evaluated in our experiments. The configurations, including learning rates, optimizers, and scheduler settings, were carefully chosen to ensure stable convergence and performance across different methods. Specific hyperparameter choices for each evaluated approach are described below.

For MS-NeRF [48] and RoseNeRF [25], both the proposal networks and the field optimizer utilize an Adam optimizer with an initial learning rate of 4×10^{-3} , $\epsilon = 1 \times 10^{-15}$, and an exponential decay scheduler that reduces the learning rate to 1×10^{-4} over 2×10^5 steps. For Zip-NeRF [2], the

Table 4: Source and license information for objects used in our dataset. Objects are obtained from various sources, including BlenderKit (<https://www.blenderkit.com/>), CGTrader (<https://www.cgtrader.com/>), Free3D (<https://free3d.com/>), Keenan’s 3D Model Repository (<https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/>), and the rest were created by our team. Additionally, objects sourced from these websites underwent modifications to their materials to better suit our dataset requirements.

Source	License	Objects
BlenderKit	Royalty Free	cat, diamond, fox, man sculpture, sleeping dragon, candle holder, cola bottle, crystal vase, demijohn vase, flower vase, Korken jar, Vardagen jar, wisolt kettle, magnifier, plastic bottle, reed diffuser, skull bottle, teacup, teapot, water pitcher, wine glass, light bulb, perfume red, perfume yellow, star-shaped bottle, household item set, ampoule, beaker, conical flask, vial, lab equipment set
	CC0	generic sculpture, woman sculpture
Blender	GPL	monkey
CGTrader	Royalty Free	graduated cylinder, test tube, round bottom flask
Free3D	Royalty Free	dog
Keenan’s	CC0	cow
Authors	CC0	ball, coloured ball, cube, coloured cube, cylinder, coloured cylinder, pyramid, coloured pyramid, torus, coloured torus, syringe

model optimizer utilizes a default configuration with an Adam optimizer set to an initial learning rate of 8×10^{-3} , $\epsilon = 1 \times 10^{-15}$, and an exponential decay scheduler reducing the learning rate to 1×10^{-3} over 2.5×10^4 steps, with 1000 warm-up steps. For Ray Deformation Network [12], both the proposal networks and the field optimizer utilize an Adam optimizer with an initial learning rate of 2×10^{-3} , $\epsilon = 1 \times 10^{-15}$, and an exponential decay scheduler reducing the learning rate to 1×10^{-4} over 1×10^5 steps. For NeuS [41] and TNSR [11], the proposal networks utilize a default configuration with an Adam optimizer set to an initial learning rate of 1×10^{-2} , $\epsilon = 1 \times 10^{-15}$, and a multi-step decay scheduler reducing the learning rate by a factor $\gamma = 0.33$ every milestone over 2×10^4 steps. The milestones are at 10000, 15000, and 18000 steps, respectively. For the field optimizer, we use an initial learning rate of 5×10^{-4} , and a cosine decay scheduler with 500 warm-up steps and learning rate peak value set to 5×10^{-2} . We have also included a 3D Gaussian splatting approach, Splatfacto [19], which is implemented in nerfstudio. For Splatfacto, we used the default configurations in nerfstudio.

E Extended Experimental Results

In this section, we compare more results of the oracle and Pythia methods with existing state-of-the-art methods on our RefRef test set. Figure 6 and Figure 7 present comparative qualitative results on objects placed in patterned cube, patterned sphere, and HDR environment map backgrounds, respectively. The patterned backgrounds are more challenging due to the presence of complex textures, where most methods struggle, especially in handling multiple refractions and total internal reflection. By contrast, the HDR environment map background is relatively easier, and most methods perform well on simple geometries. However, their performance fluctuates significantly on more complex shapes. In both settings, our approaches consistently produce more accurate and stable results, especially in scenes dominated by refractive components.

F Author Statement

The authors confirm that they bear full responsibility for any violations of rights related to the objects and data used in this work. All objects utilized in the dataset were either sourced from publicly available repositories with appropriate licensing or created by the authors. The data licenses are documented in appendix C, and any modifications to the sourced objects were performed in compliance with the respective licenses. The authors ensure that all data used adhere to the specified licenses and terms of use.

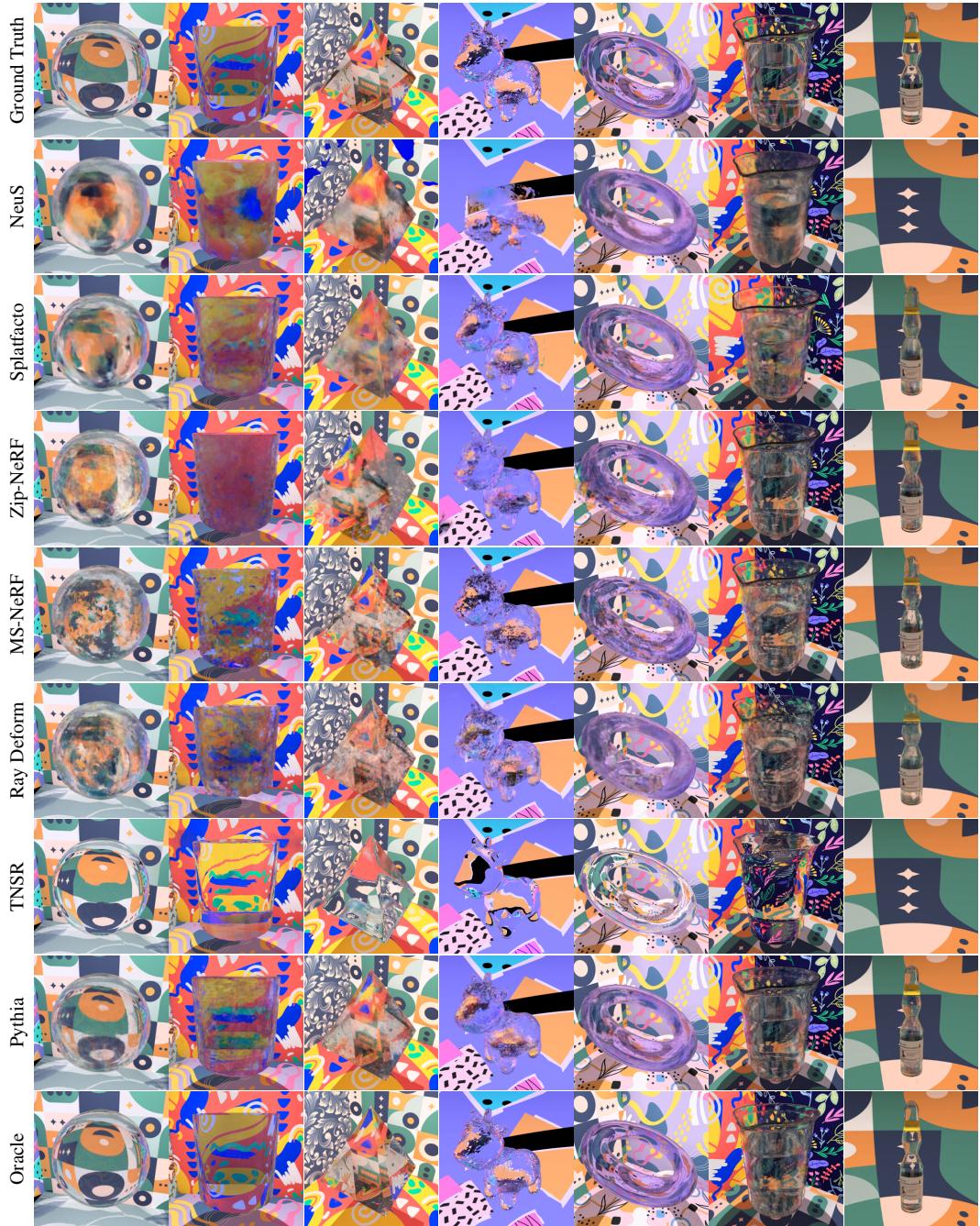


Figure 6: Qualitative comparison of novel view synthesis results on scenes with patterned cube and patterned sphere backgrounds using NeuS [41], Splatfacto [35], Zip-NeRF [2], MS-NeRF [48], Ray Deformation Network [12], TNSR [11], Pythia, and Oracle. Pythia and Oracle produce more accurate renderings, especially in scenes involving multiple refractions and total internal reflection, where other methods often fail.

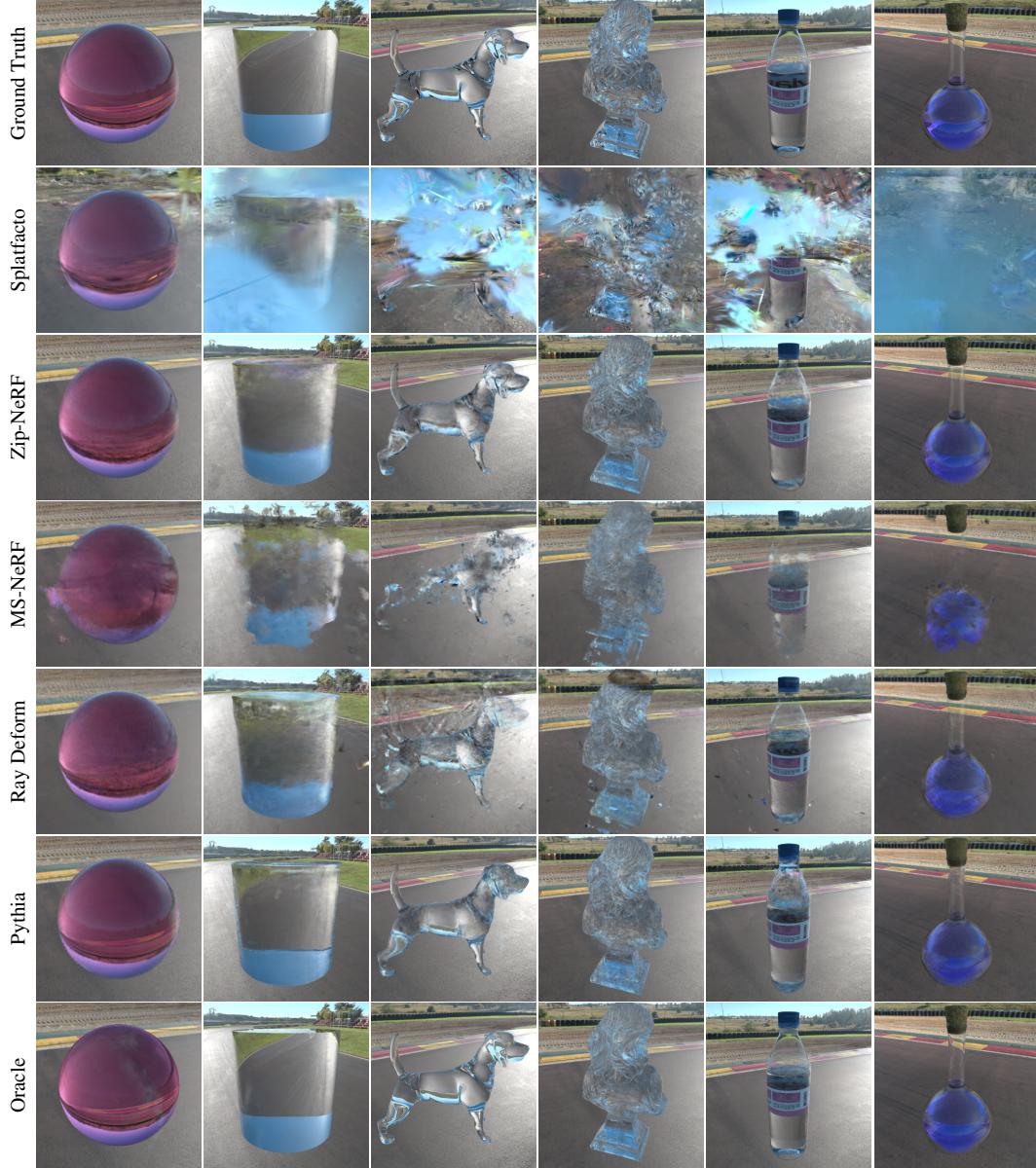


Figure 7: Qualitative comparison of novel view synthesis on scenes with HDR environment map backgrounds, using Splatfacto [35], Zip-NeRF [2], MS-NeRF [48], Ray Deformation Network [12], Pythia, and Oracle. This background type is generally less challenging than the patterned ones. Most methods perform well on simple geometries, but their results vary significantly on complex shapes, where Pythia and Oracle remain relatively robust.