NeuralTO: Neural Reconstruction and View Synthesis of Translucent Objects

YUXIANG CAI, VCIP, College of Computer Science, Nankai University, China
JIAXIONG QIU, VCIP, College of Computer Science, Nankai University, China
ZHONG LI, OPPO US Research, USA
BO REN*, VCIP, College of Computer Science, Nankai University, China

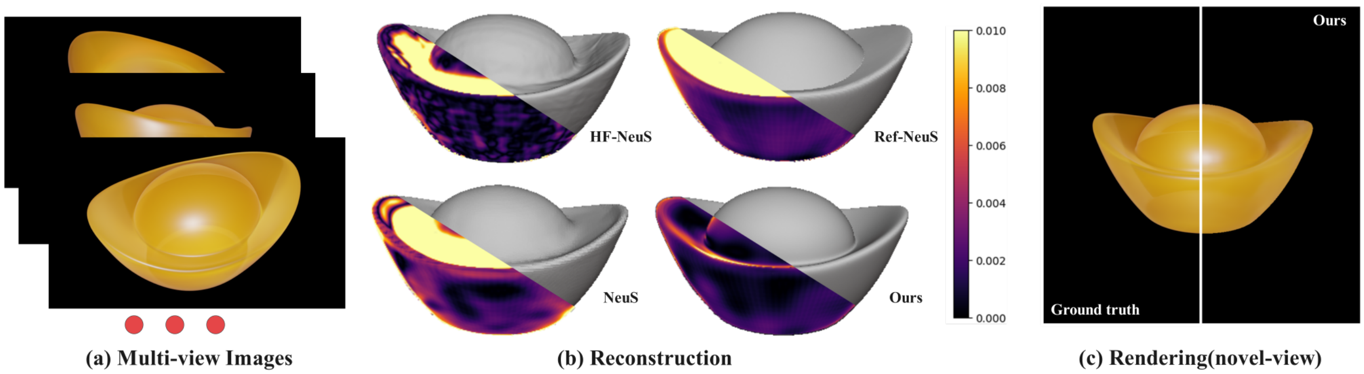


Fig. 1. In this paper, we propose a novel framework for high-fidelity surface reconstruction and novel-view synthesis of translucent objects. We derive an enhanced density function ensuring the constant extinction coefficient inside the translucent object. (a) Multi-view RGB image input. (b) The reconstruction error compared with the ground-truth geometry is far less than the baseline methods [Ge et al. 2023; Wang et al. 2021, 2022]. (c) Translucent appearance is faithfully rendered at the novel view using a learned neural participating medium with disentangled scattering properties.

Learning from multi-view images using neural implicit signed distance functions shows impressive performance on 3D Reconstruction of opaque objects. However, existing methods struggle to reconstruct accurate geometry when applied to translucent objects due to the non-negligible bias in their rendering function. To address the inaccuracies in the existing model, we have reparameterized the density function of the neural radiance field by incorporating an estimated constant extinction coefficient. This modification forms the basis of our innovative framework, which is geared towards high-fidelity surface reconstruction and the novel-view synthesis of translucent objects. Our framework contains two stages. In the reconstruction stage, we introduce a novel weight function to achieve accurate surface geometry reconstruction. Following the recovery of geometry, the second phase involves learning the distinct scattering properties of the participating media to enhance rendering. A comprehensive dataset, comprising both synthetic and real translucent objects, has been built for conducting extensive experiments. Experiments reveal that our method outperforms existing approaches in terms of reconstruction and novel-view synthesis.

CCS Concepts: • **Computing methodologies → Mesh geometry models**.

*Corresponding author.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

Additional Key Words and Phrases: Translucent Object, Neural Implicit Surface, Multi-view 3D Reconstruction

ACM Reference Format:
Yuxiang Cai, Jiaxiong Qiu, Zhong Li, and Bo Ren. 2018. NeuralTO: Neural Reconstruction and View Synthesis of Translucent Objects. In . ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Multi-view 3D reconstruction is a fundamental task in computer graphics and vision. Recently, inspired by the neural radiance field proposed in [Mildenhall et al. 2021], numerous follow-up works have focused on modeling the 3D scenes using density σ and view-dependent color c. This learned implicit representation of the object or scene performs impressive results in novel view synthesis. Pioneered by VolSDF [Yariv et al. 2021] and NeuS [Wang et al. 2021], one direction of improvement work uses the signed distance function (SDF) to optimize view consistency of the density field so that a meaningful surface can be extracted from it. They propose to learn implicit SDF using the neural network and combining the density of the scene with it. Trained using multi-view images, they optimize that implicit neural SDF network to obtain a solid surface.

Translucent objects are a kind of object with special optical properties. Unlike the opaque one, which obstructs light transferring from the outside to the inside, the translucent object lets light pass through its surface while simultaneously scattering it in different directions. For opaque objects, all light leaving the object is scattered from the surface. For translucent objects, some of the light leaving the object has entered the object and been scattered multiple times before emerging. Recent works [Ge et al. 2023; Wang et al. 2022] for reconstruction based on NeuS [Wang et al. 2021] using a neural radiance field and implicit SDF network achieve excellent results on opaque objects. Their method only considers the points on the reconstructed surface, while the points inside the object are overlooked. However, translucent appearance is strongly coupled with the total geometry, it is often the case that inverted shapes are observable in the rendering results. The absorption and scattering inside the translucent region play a key role in the rendered result, which can not be covered by the conventional NeuS-like model where the weight is only non-zero near the opaque surface.

To tackle the above issue, we propose a novel model for translucent object reconstruction and view synthesis. We propose a theoretical model for the neural radiance field of translucent objects and reparametrize the density field inside the object using an estimated extinction coefficient. The extinction coefficient (often informally referred to as "density") defines the net loss of radiance due to both absorption and scattering. For translucent objects with homogeneous material, their extinction coefficient is constant. We utilize this physical property to design our invariant density function related to the extinction coefficient. Based on the proposed model, we design a framework for high-fidelity surface reconstruction and novel-view synthesis. A simple pipeline of our method can be found in Fig. 1. In the first stage, we combine the transmission color and surface color to train our neural SDF network. In the second stage, we utilize the recovered geometry and density field to decompose scattering properties into single-scattering and multi-scattering. For novel-view synthesis, we learn their neural representations using participating media and multi-level conical sampling.

To evaluate the performance of our method, a dataset containing translucent objects is required for both reconstruction and rendering. The previous datasets like DTU [Jensen et al. 2014] and BlendedMVS [Yao et al. 2020] are available for reconstructing and rendering opaque objects. The Shiny Blender dataset in [Ge et al. 2023] and the Glossy dataset in [Liu et al. 2023] contain objects with highly specular appearance. However, none of them contain translucent objects. We propose a dataset of translucent objects under a co-located flashlight, which contains "Syn-Trans" consisting of synthetic images and "Real-Trans" captured using a smartphone. The details of our dataset are introduced in Sec. 4.2.

We summarize our key contributions as:

- We propose a theoretical model for the neural radiance field of translucent objects, which parametrizes the density field using a constant extinction coefficient.
- We propose a novel framework for high-fidelity surface reconstruction of translucent objects and refine the view synthesis result under the co-located flashlight using neural participating media.
- We construct a new translucent dataset under the co-located flashlight for evaluating reconstruction and rendering results.

2 RELATED WORKS

2.1 Neural radiance fields

NeRF [Mildenhall et al. 2021] utilizes the MLP (Multi-Layer Perceptron) network and multi-view images to learn an implicit representation of the scene. It proposes to predict the view-dependent radiance and view-independent volume density of points in 3D space. NeRF is a continuous implicit representation of 3D scenes, which has a significant improvement in expression ability compared to discrete display representations [Gao et al. 2022; Li et al. 2023a]. Through the volume rendering equation, NeRF can synthesize high-quality images from novel views. Many following works [Chen et al. 2022, 2023; Fridovich-Keil et al. 2022; Müller et al. 2022] improve the scene representations of NeRF. Mip-NeRF [Barron et al. 2021] essentially improves the sampling theory of NeRF to achieve anti-aliasing. NeuLF [Li et al. 2023d, 2021] represent scenes using a 4D light field, which is efficient for high-quality novel-view synthesis. Other works improve the radiance field to apply NeRF to complex scenes. Ref-NeRF [Verbin et al. 2022], Mirror-NeRF [Zeng et al. 2023] and NeRFReN [Guo et al. 2022] add specular reflections properties on the radiance field. These methods are designed for opaque objects and they can't model the appearance of translucent objects. For non-opaque objects, Bemana et al. [Bemana et al. 2022] propose to handle refraction radiance using simplified Eikonal rendering [Ihrke et al. 2007]. NeMF [Zhang et al. 2023a] combines Microflake theory [Heitz et al. 2015] with neural radiance field. OSF [Yu et al. 2023] proposes to use additional sampling between points and light sources. It requires objects with a known bounding box and the location of light. These methods focus on novel-view synthesis and surface reconstruction is not mentioned in their method.

2.2 Neural reconstruction and implicit surfaces

NeRF can not locate the precise surface position of an object. To represent the surfaces of the scene using a neural network, the occupancy functions and signed distance fields(SDF) are most commonly used. Early works like [Chen and Zhang 2019] take point clouds as input and output an implicit neural surface. More works are focused on reconstructing implicit surfaces from multi-view images and learning an SDF function consisting of the fully connected MLP network. DVR [Niemeyer et al. 2020] and IDR [Yariv et al. 2020] adopt surface rendering to reconstruct high-quality surfaces in relatively simple scenes. UNISURF [Oechsle et al. 2021], VolSDF [Yariv et al. 2021] and NeuS [Wang et al. 2021] propose to design weighting strategies on render equation of NeRF. UNISURF predicts the occupancy field to combine the color of the surface point of the object, as well as the points near the surface. It gradually removes ambiguities during training and finally obtains a solid surface. VolSDF and NeuS propose to design a weight function considering the SDF value of points in 3D space. Based on VolSDF and NeuS, works like [Mu et al. 2023; Wu et al. 2023; Zhang et al. 2021c] focus on reconstruction from sparse views. BakedSDF [Yariv et al. 2023] decomposes diffuse color and specular reflection components into the vertices of triangle meshes extracted from the SDF network. NeRO [Liu et al. 2023] and Ref-NeuS [Ge et al. 2023] extend NeuS to reflective surface reconstruction. They propose to separate the reflection radiance from the neural network. These methods assume that the radiance observed by the camera only relates to the irradiance at the surface while omitting the transmission and scattering light from the inside part of the translucent object. Methods like [Gao et al. 2023; Li et al. 2023b; Lyu et al. 2020] focus on transparent object reconstruction. Deng et al. [2022] utilize differentiable BSSRDF path-tracing to reconstruct real-world translucent objects, but their method is computationally costly. The method in [Lin et al. 2023] acquires the shape of translucent objects using sinusoidal and binary patterns of illumination, while our reconstructed method can handle arbitrary illumination.

2.3 Inverse rendering from multiple images

Given multiple images of an object, inverse rendering aims to recover the shape, material, and lighting through differentiable rendering. Recent inverse rendering work utilizes physics-based rendering equations with learned parameters from neural networks. Unlike methods in [Deschaintre et al. 2018; Li et al. 2020; Shi et al. 2023; Wang et al. 2023; Zhu et al. 2022], which learn the material from a single image, the shape and material learned from multiple images are more suitable for scene editing. Works like [Yao et al. 2022; Zhang et al. 2023b, 2021a,b] recover unknown environment light together with material appearance. Works like [Kaya et al. 2022; Yang et al. 2022] combine traditional photometric stereo with neural radiance field to make reconstruction or inverse rendering. IRON [Zhang et al. 2022] performs impressive material decomposition under the co-located flashlight. Our work addresses the same lighting conditions as IRON. For a robust novel view synthesis and rendering translucent appearance, we benefit from the physics-based rendering equation and learn neural representations of participating media with disentangled scattering properties.

2.4 Neural rendering for translucent object

Works like [Wang et al. 2008; Yang and Xiao 2016] learn material properties for the BSSRDF model to render translucent objects with scattering. Li et al. [2023c] predict parameters used in forward rendering and train a neural network to predict color using these parameters. It requires full supervision using ground truth parameters to train its network. RPNN [Kallweit et al. 2017] and MRPNN [Hu et al. 2023] use the neural network to render translucent objects like clouds with complex scattering properties. However, their method requires a ground-truth density field and supervision using ground-truth radiance. Zhu et al. [2023] propose to learn a neural radiance transfer field(NRTF) [Lyu et al. 2022] to render the scattering object but requires a pre-computed geometry in learning progress and a large number of images captured under varying lighting conditions. Zheng et al. [2021] propose to learn a relightable participating media for novel view synthesis on known light position. These methods above can not recover the geometry while our method exploits reconstructed geometry to render a more plausible result by contrast.

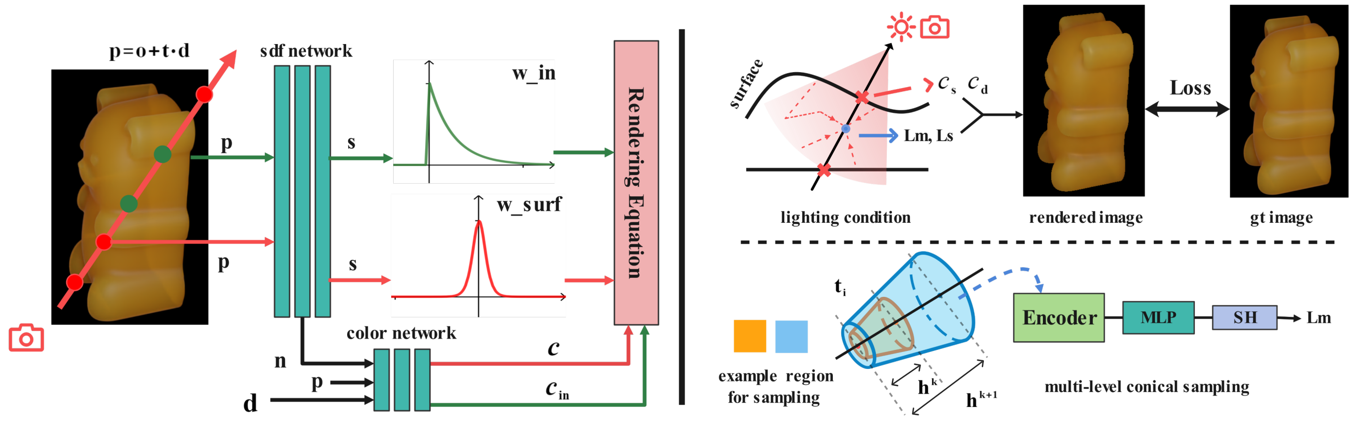
Table 1. **Symbols and its definitions.** For similar representations unlisted in this table, they represent similar meanings, such as the terms w_{in} and w_{surf}.

Symbol	Definition
o	Camera origin
$d, \omega, \omega', \omega_i, \omega_o$	Direction
$\mathbf{c}, \mathbf{c}_{\text{in}}, L(\mathbf{x}, \omega)$	Radiance of a point
σ	Volume density
σ_t, σ_s	Extinction and scattering coefficient
$T, T(t)$	Accumulated transmittance
$w(t), w_j, w_{\text{in}}$	Weight function computed using T and σ
α_j	Opacity value computed as $1 - T$
I	Intensity of the light
$L_{\text{rgb}}, L_{\text{eik}}$	Loss function
f_G	Implicit SDF function
n	surface normal
$x, x_2, p(t), p$	A 3D point
$t, t_i, \delta_t, t_n, t_n^*$	Distance along a certain direction
α	Roughness
$\text{Tr}, \text{Tr}(x, n)$	Transmittance value at surface
f_r	BRDF function
$c_l^m, Y_l^m(\omega_i)$	Weight and basis function for spherical harmonic
c_d, c_t, c_s	Diffuse color,scattering color and specular color
L_s, L_m	Single scattering and multi-scattering
$\mathcal{F}, F_1$	feature descriptor and extracted feature
F	Fresnel term in BRDF function

3 METHOD

3.1 Overview

Given a set of RGB images of translucent objects with known camera pose and camera intrinsic, our method adopts two steps to reconstruct the geometry and render arbitrary views with translucent appearance. We assume that all images are captured using the co-located flashlight. We first reconstruct the translucent object by optimizing a neural SDF network using the volume rendering equation. We analyze the limitation that exists in the baseline of NeuS [Wang et al. 2021]. To resolve such limitations in their model, we propose a theoretical model for the neural radiance field of translucent objects in reconstruction. We reparametrize the density inside the object using the extinction coefficient. Our density field and training process are introduced in Sec. 3.3. After that, we exploit the learned geometry of the object in the physically based rendering equation for novel-view synthesis. We learn spatial invariant color, represented as albedo in material, roughness, and transmission albedo for the surface rendering equation under direct co-located flashlight light. For the detailed translucent appearance under indirect light, we learn neural participating media with disentangled scattering properties. Inspired by [Kallweit et al. 2017] and [Zheng et al. 2021], we decompose scattering properties using single-scattering and multi-scattering. We propose a multi-level conical sampling module to learn the radiance of multi-scattering related to overall geometry. We introduce details of our rendering method in Sec. 3.4. The overall pipeline of our framework is shown in Fig. 2.



(a) Reconstruction pipeline (b) Rendering pipeline

Fig. 2. **Overview of our geometry reconstruction pipeline (left) and view synthesis pipeline (right).** For geometry reconstruction, we propose another weight function w_{in} based on the constant extinction coefficient at a homogeneous medium. For a better visual appearance of translucent objects, we further model the scattering property in our reconstructed volume. We model the specular color c_s and diffuse color c_d under direct light and decompose the scattering property under indirect light into single-scattering L_s and multi-scattering L_m. We learn the neural representation of L_m using multi-level conical sampling. A detailed explanation of our method can be found in Sec. 3.3 and Sec. 3.4.

3.2 Preliminaries

Neural radiance fields. NeRF [Mildenhall et al. 2021] proposes to use the volume rendering equation to render images under different view directions. The color C of the pixel corresponds to a given ray $p(t) = o + td$, where $o \in \mathbb{R}^3$ represents the camera origin and $d \in \mathbb{S}^2$ represents the view direction of the camera. NeRF involves an integral along the ray with boundaries t_n and t_f uses an MLP to predict unknown values in the rendering equation.

$$C = \int_{t_n}^{t_f} T(t)\sigma(p(t))\mathbf{c}(p(t), d)dt \tag{1}$$

The volume density σ and radiance $\mathbf{c}$ of a point is modeled by an MLP. The volume density is used to calculate the accumulated transmittance $T(t)$.

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma\left(p(s)\right) ds\right) \tag{2}$$

Then we can compute a weight function $w(t) = T(t)\sigma(p(t))$ to the c of sampled points along the ray $p(t)$ to compute the pixel color C. NeRF solves this equation of weight with the numerical integration method:

$$\begin{aligned} w_j &= \alpha_j \prod_{i=1}^{j-1}(1-\alpha_i) \\ \alpha_j &= 1 - \exp(-\sigma_j \cdot (t_{j+1} - t_j)) \end{aligned} \tag{3}$$

NeuS for surface reconstruction. NeuS [Wang et al. 2021] represents the surface of an object using implicit neural SDF. It combines volume rendering and surface rendering by setting up a connection between $w(t)$ of sampled points and the SDF value of these points. NeuS defines S-density as $\phi_s(x) = se^{-sx}/(1+e^{-sx})^2$ to replace the original one where x represent SDF value. The weight function completed using ϕ_s attains local maximal value at surface intersect points in their rendering function, which conforms to the optical properties of opaque objects.

Physics-based surface rendering. In the surface rendering equation, the observed radiance from the view direction is modeled as an integral of the bidirectional reflectance distribution function (BRDF) and irradiance at the surface point x with normal n.

$$L_o\left(x, \omega_o\right) = \int_{\Omega} L_i\left(x, \omega_i\right) f_r\left(x, \omega_i, \omega_o\right)\left(\omega_i \cdot n\right) \mathrm{d}\omega_i \tag{4}$$

Where $L_i\left(x, \omega_i\right)$ is the incoming light on x from direction ω_i. The f_r is the BRDF function which defines an energy distribution of incoming light with respect to view direction ω_o.

Participating medium. A participating medium [Cerezo et al. 2005] affects light that passes through it, rather than leaving light unchanged as when it passes through the clear air. A participating medium absorbs, scatters, and emits light at each point along a light ray as the ray passes through it. The radiative transfer equation in the non-emissive participating medium is defined as:

$$\begin{aligned} (\omega \cdot \nabla)L(\mathbf{x}, \omega) &= -\sigma_t(\mathbf{x})L(\mathbf{x}, \omega) + \sigma_s(\mathbf{x})L_{\text{st}} \\ L_{\text{st}} &= \int_{S^2} \text{phase}\left(\mathbf{x}, \omega, \omega'\right) L\left(\mathbf{x}, \omega'\right) d\omega' \end{aligned} \tag{5}$$

where S^2 denotes the spherical region around the position x, L represents the radiance and ω represents the direction. σ_t and σ_s represent extinction coefficient and scattering coefficient. The derivative of the radiance in the direction of ω is expressed as $\omega \cdot \nabla$. The phase function represents bi-directional energy distribution. For objects with isotropic scattering, $\text{phase}(x, \omega', \omega) = \frac{1}{4\pi}$.

3.3 Surface reconstruction from translucent appearance

NeuS optimizes non-zero weights only near the opaque object surface and assumes zero weight inside the object, which is improper for translucent objects. A direct result is that the extinction coefficient inside the objects is not optimized and can vary largely even if a homogeneous material is considered. To resolve the limitation in NeuS, we model the density inside the object with correct physical properties using the extinction coefficient. Detailed explanations are provided in the following paragraphs.

Modeling density with correct physical properties. Within homogeneous translucent objects, the extinction coefficient is a non-zero constant while outside of the object, the extinction coefficient is zero. That is to say, we need to find a differentiable form for a square-wave function that keeps invariant inside the object and equal to zero outside the object. Inspired by VolSDF[Yariv et al. 2021], we use a Laplace distribution function associated with SDF value to represent our density field:

$$\sigma(x) = \begin{cases} \frac{\sigma_t}{2} \exp\left(\frac{-f_G(x)}{\beta}\right) & \text{if } f_G(x) \geq 0 \\ \sigma_t - \frac{\sigma_t}{2} \exp\left(\frac{f_G(x)}{\beta}\right) & \text{if } f_G(x) < 0 \end{cases} \tag{6}$$

where f_G is our neural implicit SDF and $f_G(x)$ is the SDF value of point x. σ_t is the constant value of the extinction coefficient. A larger σ_t tends to lead to a lower translucency as the attenuation of light is higher when traveling inside the volume. When β approaches zero, our density σ of points inside the object converges to σ_t, while the density of points outside is set to 0. Note that Eq. 6 is not mathematically smooth inside the translucent object. However, with the discrete calculation nature of neural rendering equation, such errors are small and only vary slightly near the center, especially for near-zero β values. For generalization, we set the σ_t and β learnable and optimize them in the training process. We report our learned density field and β in Fig. 4.

Training process. Shown in the left of Fig. 2, we sample N_1 points alongside the ray from t_n to t_f. Conventionally, n represents "near" and f represents "far". For each point $p(t) = o + t * d$, where o is the original position of the camera and d is the view direction. These points are sent to the neural SDF network to get the predicted SDF value $f_G(p(t))$. Our goal is to reconstruct the correct geometry from the input image sequence. However, most translucent object appearances are not purely transmitting and contain surface highlights. It has been proven by [Fan et al. 2023; Qiu et al. 2023] that separating the specular and diffuse components is beneficial for reconstruction and rendering. Inspired by them, our framework contains two branches each for the on-surface color c and the transmitted color c_{in} inside the object, respectively. The former can be estimated further by separating into spatial-invariance color c_d and reflection color c_r, where we learn c_d using network proposed by [Yariv et al. 2020] and c_r using the method in [Verbin et al. 2022]. The latter needs to be calculated using our scheme.

$$C = (1-\gamma)\sum_{t_n}^{t_f} \mathrm{w}_{\text{surf}}\mathbf{c}(p(t), d) + \gamma \sum_{t_{n^*}}^{t_{f^*}} \mathrm{w}_{\text{in}}\mathbf{c}_{\text{in}}(p(t), d) \tag{7}$$

The rendered color is given by Eq. 7. w_{surf} is the weight function proposed in NeuS. We add a parameter γ to balance these terms. For the uniformly sampled t from "near" t_n^* to "far" t_f^*, the w_{in} is derived from our density model Eq. 6 using Eq. 3. Theoretically, the weight function of points inside the object with a constant density relies exponentially on the extinction coefficient.

$$\mathrm{w}_{\text{in}}(t_i) = (1 - \exp(-\sigma_t \cdot \delta_t))\exp(-\sigma_t \cdot (t_i - t_n^*)) \tag{8}$$

where $\delta_t = t_{i+1} - t_i$. The detailed derivation is provided in the Appendix. For physical plausibility, in the calculation of w_{in}, we confine the sampling region to points between the intersection positions t_n^* and t_f^*, which can be calculated following [Fu et al. 2022]. We define the set of intersection points Ω as:

$$\begin{aligned} R &= \{t_i \mid f(t_i) \cdot f(t_{i+1}) < 0\} \\ \Omega &= \left\{t^* \mid t^* = \frac{f(t_i)t_{i+1} - f(t_{i+1})t_i}{f(t_i) - f(t_{i+1})}, t_i \in R\right\} \end{aligned} \tag{9}$$

where $f(t_i)$ is a simplicity format of $f_G(p(t_i))$. We take the minimum value and maximum value in Ω as t_n^* and t_f^*.

We restrict the integral of the weight of points alongside the camera ray to 1 in the training process, which indicates whether this ray hits the surface or not. We use the L1 RGB loss L_{rgb}, Eikonal loss L_{eik}[Gropp et al. 2020] and normal penalty loss L_n[Verbin et al. 2022] in the training process. k_1, k_2 are hyper-parameters to adjust the penalty weight.

$$\text{Loss} = L_{\text{rgb}} + k_1 \cdot L_{\text{eik}} + k_2 \cdot L_n \tag{10}$$

It is to be noted that, after the above optimization, we are able to obtain a satisfiable geometry of the translucent object. However, the direct result of the rendered colors using Eq. 7 is still not perfect. The reason is that we do not fully capture the scattering effects inside the participating media in this function.

3.4 Neural rendering using recovered geometry

For a better visual appearance of translucent objects, we further model the scattering property in our reconstructed volume using spherical harmonic with learned coefficients and jointly optimize neural materials from photometric images. For conforming to physically based rendering, we separate the rendered color into surface color under direct light and translucent appearance under indirect light. We render surface color using the physics-based surface rendering equation and the translucent appearance using neural participating media with disentangled scattering properties. We introduce these parts in the following paragraphs respectively.

Co-located light assumption. Our scheme starts from the physics equation Eq. 5. It is challenging for the general task of learning translucent appearance from captured images. One reason is that the geometry and lighting complexities are strongly coupled for translucent objects. Thanks to the method in Sec. 3.3, we can recover good geometry from arbitrary lighting environments, and we can consider that the geometry is known in this section. However, recovering scattering properties under unknown arbitrary environmental lighting is still challenging because the path of light passing through the interior of an object is very complex. Previous works simplify this issue using a known lighting condition. For example, works like [Zheng et al. 2021; Zhu et al. 2023] learn scattering properties under known light position, [Hu et al. 2023; Kallweit et al. 2017] focus on parallel light. To limit the input complexity, in this section, we take photometric images under a co-located flashlight as our input, which assumes that the captured object is exposed to only one light and the aligned with the view direction. The simplified surface rendering equation is defined as:

$$L_o(x, \omega_o) = \frac{I}{\|x - o\|_2^2} f_r(x, \omega_o, \omega_o)(\omega_o \cdot n) \tag{11}$$

where L_o, x, n, ω_o, f_r are observed light, surface intersection, surface normal, view direction of the camera, and BRDF function. The incident light is represented as the attenuation of a white point light with I intensity.

Appearance under direct light. Shown in the right top of Fig. 2, we render appearance under direct light using neural materials. We represent roughness α as a neural network f_α and we represent diffuse albedo as f_a. Moreover, we estimate light transmission under unknown material using the neural network f_{Tr}, which determines how much energy of light transfers to the inner of the object. For appearance under direct light, we use the same GGX microfacet BRDF function f_r as IRON [Zhang et al. 2022] to compute the specular c_s and diffuse color c_d. We revise the L in Eq. 4 to $(1 - \text{Tr})L$ considering the transmission of the light.

Modeling translucent appearance using neural participating media. Similar to [Zheng et al. 2021], we decompose scattering properties into single-scattering L_s and multi-scattering L_m and learn their neural representations. The single-scattering represents the scattering radiance of in-coming light. For homogeneous material that exhibits isotropic scattering, the single scattering can be calculated as Eq. 12.

$$L_s(p) = \frac{1}{4\pi}\sigma_s(p(t)) \cdot L_t(n, \omega_i, x)T(x, p) \tag{12}$$

where σ_s is the scattering coefficient, T is the transmittance from surface point x to points p inside the object. We compute T using the reconstructed density field from stage one. Following [Hu et al. 2023], we represent the ratio of the extinction coefficient and the scattering coefficient using a learnable constant parameter ξ. Inherited from Eq. 6, we represent the real extinction coefficient of the points related to its position. The learned β from the first stage is small enough to ensure the invariance of the extinction coefficient so that the ratio can be regarded as a constant value. L_t represents the indirect light transferring from the surface.

$$L_t(n, \omega_i, x) = (1 - F)\text{Tr}(x, n)\frac{I}{||x - o||_2^2}(n \cdot \omega_i) \tag{13}$$

Where F is the Fresnel term in BRDF function f_r. The multi-scattering represents the in-coming light scattering more than once inside the media, which can be conducted by recursively substituting the solved L into the right part in RTE equation 5. Unlike single scattering, multiple scattering is not only related to a single point but also to the overall shape. The estimated value of multi-scattering usually requires path-tracing and integral computation over all traced paths. Works in [Kallweit et al. 2017] propose that the multiple scattering can be learned using a neural network that takes features from multi-level sampling as input. Inspired by them, we propose to estimate multi-scattering using extracted features from different levels of sampling. Shown in the right bottom of Fig. 2, we compress the area affected by the point light into cones with different heights and widths, which is more efficient for representing the spatial region. When light scatters more than one time, the reachable region is enlarged and the sampling level should increase at the same time. Our sampling region grows from the yellow one to the blue one, which contains more points. We use Integrated Positional Encoding(IPE) proposed in [Barron et al. 2021] as the feature descriptor z of each cone region. We feed encoded features under different sampling levels to the neural network to predict the weight of spherical harmonic coefficients c_l^m. The detailed algorithm of our conical sampling is listed in the Algorithm. I.

Algorithm I: Multi-level conical sampling

Input: Sampled positions $\{t_1, t_2, ...\}$; view direction d; surface intersection point x; MLP module $\text{MLP}_1, \text{MLP}_2$ level k, initial value for radius r_0; factor λ.

Output: Coefficient of spherical harmonics for sampled points $\{\{c_l^m\}^1, \{c_l^m\}^2 ...\}$

```
i = 0; h=0.5 * (t_2 − t_1); r = r_0;
features {F_1, F_2, ...} ⟵ None
repeat
    for each t in sampled positions do
        μ_t = (3(t+h)^4 − (t−h)^4) / (4(t+h)^3 − (t−h)^3)
        σ_t = (3(t+h)^5 − (t−h)^5) / (5(t+h)^3 − (t−h)^3) − μ_t^2, σ_r = r^2 (3(t+h)^5 − (t−h)^5) / (20(t+h)^3 − (t−h)^3)
        z ⟵ IPE(Gau(μ_t, v_t, v_r), d, x)
        F_t ⟵ MLP_1(z, F_t)
    end
    r = r * λ, h = h * λ
until i >= k;
for each t in sampled positions do
    {c_l^m}^t ⟵ MLP_2(F_t)
end
```

After that, we resolve the integral in the multi-scattering function using Monte Carlo integration with M sampled direction.

$$L_m = \int_{S^2} \frac{1}{4\pi}\sigma_s(p(t)) \cdot \sum_{l=0}^{l_{\max}} \sum_{m=-l}^{l} c_l^m Y_l^m(\omega_i) d\omega_i \tag{14}$$

Where Y_l^m are spherical harmonic basis functions and l_{max} is the maximum band. Note that the We follow the method in [Zheng et al. 2021] to solve the RTE rendering equation using the numerical integration method as Eq. 15.

$$c_t = \Sigma_{j=1}^{N_2} T(x_1, p(t)) \left(1 - \exp\left(-\sigma_t(p(t))\delta t\right)\right)(L_s + L_m) \tag{15}$$

Where $p(t) = x + t * d$, x is defined as the first intersection of the surface. Instead of computing intersection points using Eq. 9, we use the sphere tracing algorithm to improve the accuracy. We can find at least two intersection points and x_2 is the farthest one. We compute t using the distance of x and x_2: $\delta t = \frac{||x_2 - x||_2}{N_2}$, $t_j = j * \delta_t$.

Training process. We combine the appearance under direct light and the translucent appearance under indirect light as our rendered result: $C = c_s + c_d + c_t$. We optimize σ_s, L, α, Tr, diffuse albedo, and coefficients of spherical harmonic using L1 RGB loss L_{rgb}. To reduce the complexity, we set ξ equals to diffuse albedo. Moreover, we add eikonal loss for x and x_2 to fine-tune the learned neural SDF network, which ensures the accurate normal for surface intersection points. We add Bilateral Smoothness Loss in [Yao et al. 2022] to encourage α not to change rapidly. k_3, k_4 are hyper-parameter.

$$\text{Loss} = L_{\text{rgb}} + k_3 \cdot L_{\text{eik}} + k_4 \cdot L_{\text{smoothness}} \tag{16}$$

4 EXPERIMENTS

4.1 Implementation Detail

We represent the geometry of the object following NeuS [Wang et al. 2021]: $f_G : x \rightarrow (\mathcal{F}, f_G(x))$. The output of our neural SDF network consists of a 256D geometric feature descriptor $\mathcal{F}$ and an SDF value $f_G(x)$. The geometric features $\mathcal{F}$, gradients of the SDF network $\nabla f_G(x)$, and points p are fed to color networks to predict c. We obtain the normal n using: $n = \nabla f_G(x)/||\nabla f_G(x)||$.

The on-surface color c in our framework consists of the spatially invariant color c_d and reflection color c_r. Following [Verbin et al. 2022] we compute c using: $c = c_d + \text{tint} * c_r$. c_d is predicted using the same MLP structure in [Yariv et al. 2020], which takes position, and geometric feature descriptor as the input. $tint$ is a parameter between 0 and 1, which determines the intensity of reflection light. c_r is predicted using method in [Verbin et al. 2022], which takes the computed reflection direction $\hat{d}$, position, and geometric feature descriptor as the input.

For the surface reconstruction stage, we train our model with 100k iteration and sample 1024 camera rays on every step. We uniformly sample 64 points to compute w_{surf}, 32 points to compute w_{in}. We adopt the Adam optimizer [Kingma and Ba 2014] with $\beta_1 = 0.9, \beta_2 = 0.999$ and we set the initial learning rate to 0.0005. The parameter γ is set to 0.5 initially. k_1, k_2 are set to 0.1 and 0.005. The overall training time is about 8 hours using a single NVIDIA GeForce RTX 3090 GPU.

For the rendering stage, we represent roughness α as a neural network: $f_\alpha : (x, n, \mathcal{F}) \rightarrow \alpha \in R$. The diffuse albedo is predicted using neural network $f_a : (x, n, \mathcal{F}) \rightarrow \text{albedo} \in R^3$. transmission albedo is predicted using the neural network $f_{\text{Tr}} : (x, n) \rightarrow \text{Tr} \in R$. We train our model with 80k iteration and sample 128x128 camera rays per step. The scale factor λ is set to 0.5, r_0 is set to the width of the pixel in world coordinates and h_0 is set to the interval of neighboring sampled points, where M is equal to 64 and N is equal to 32. $k3, k4$ are set to 0.1 and 0.05 separately.

4.2 Dataset

For "Syn-Trans" dataset. We choose 6 different objects to create our synthetic scenes with different translucent materials, including "Gummybear", "Stanford Dragon", "Yuanbao", "Ancient Dragon", "Nail", and "Juice". We use the PrincipleBSDF shader in Blender to simulate real-world materials such as jade, gummies, juice, and plastics. The detail of each scene and material is shown in the Appendix. We set 90-120 views that uniform sampling on a sphere or semi-sphere to render training images of resolution 800x800.

For "Real-Trans" dataset. We place the translucent object at the center of an automatic rotating platform and shoot a video for about 40 seconds in a black room. We extract 1 frame every 10 frames from the video for training and estimate camera poses using COLMAP [Schonberger and Frahm 2016]. We extract 1 frame every 20 frames from the video to test the result of view synthesis. For some translucent objects with complex optical properties, we add an opaque object for camera pose estimation. Each real scene uses about 100 images from a circular trajectory with a resolution of 960x540 pixels.

Table 2. **Reconstruction evaluation result on Syn-Trans dataset in Chamfer Distance(CD ↓).** We compare the state-of-the-art reconstruction method using neural implicit SDF network: NeuS[Wang et al. 2021], HF-NeuS[Wang et al. 2022], Ref-NeuS[Ge et al. 2023]. The text with **Bold** represents the best evaluation result while underline text represents the second best result.

Scene	NeuS	HF-NeuS	Ref-NeuS	Ours
GummyBear	0.0047	0.0024	0.0023	**0.0011**
Stanford Dragon	0.6457	0.0070	0.0045	**0.0037**
Nail	0.1371	0.0380	0.0327	**0.0022**
Juice	0.0150	0.0170	0.0216	**0.0132**
Yuanbao	0.0085	0.0093	0.0089	**0.0012**
Ancient Dragon	N/A	0.0059	0.0037	**0.0022**

4.3 Geometry evaluation

To export the mesh from the learned neural SDF network, we take grid sampling within a fixed square space(from -1 to 1) to predict every sampled point using the SDF network and obtain the reconstructed mesh using the Marching-Cubes algorithm. We evaluate geometry reconstruction results under the Syn-Trans dataset using the Chamfer Distance(CD) between the ground-truth mesh and the reconstructed one. The quantitative comparison result is shown in Tab. 2 and the qualitative comparison is shown in Fig. 3. NeuS fails to reconstruct the accurate surface due to nonnegligible color bias from points away from the forward-face surface. HF-NeuS takes the translucent appearance as the high-frequency details of geometry, which leads to a noisy and unsmooth surface. Ref-NeuS takes advantage of the reflective highlights in captured images but predicts inaccurate shape, especially on concave surfaces. By contrast, our method performs best due to the proper method to model the density of points inside and the revised rendering function in the training process. To evaluate our density field and weight function learned from the multi-view image, we display the learned density value, SDF value, and the weight of sampled points in Fig. 4. The learned β at this scene is $2.53e{-}5$, which is small enough to ensure the invariant of density inside. The density of points inside, shown in (b) is suitable for the constant extinction coefficient of homogeneous objects. As a result, the weight of points inside, shown in the red line in (c) gradually declines when far away from the surface points. The overall curve is close to an exponential function relative to the distance to the surface, which is theoretically analyzed in the Appendix. NeuS omits the weight of these points and the weight of the second intersection plane is close to zero in their figure, which is wrong as we can see the color of the second intersection plane in the marker point at the left top figure.

Result on natural scene. The method in Sec. 3.3 is able to reconstruct from arbitrary lighting and is not limited to the co-located flashlight. We show our reconstruction result on natural scenes with environment light in Fig. 6.

Result on Real-Trans. We show the reconstruction result for the real scene in Fig. 5. There is no ground-truth geometry data in our "Real-Trans" dataset so we skip the metric comparison. For opaque or nearly opaque parts of the object, there is a significant difference in the reconstruction performance of Ref-NeuS compared to synthetic

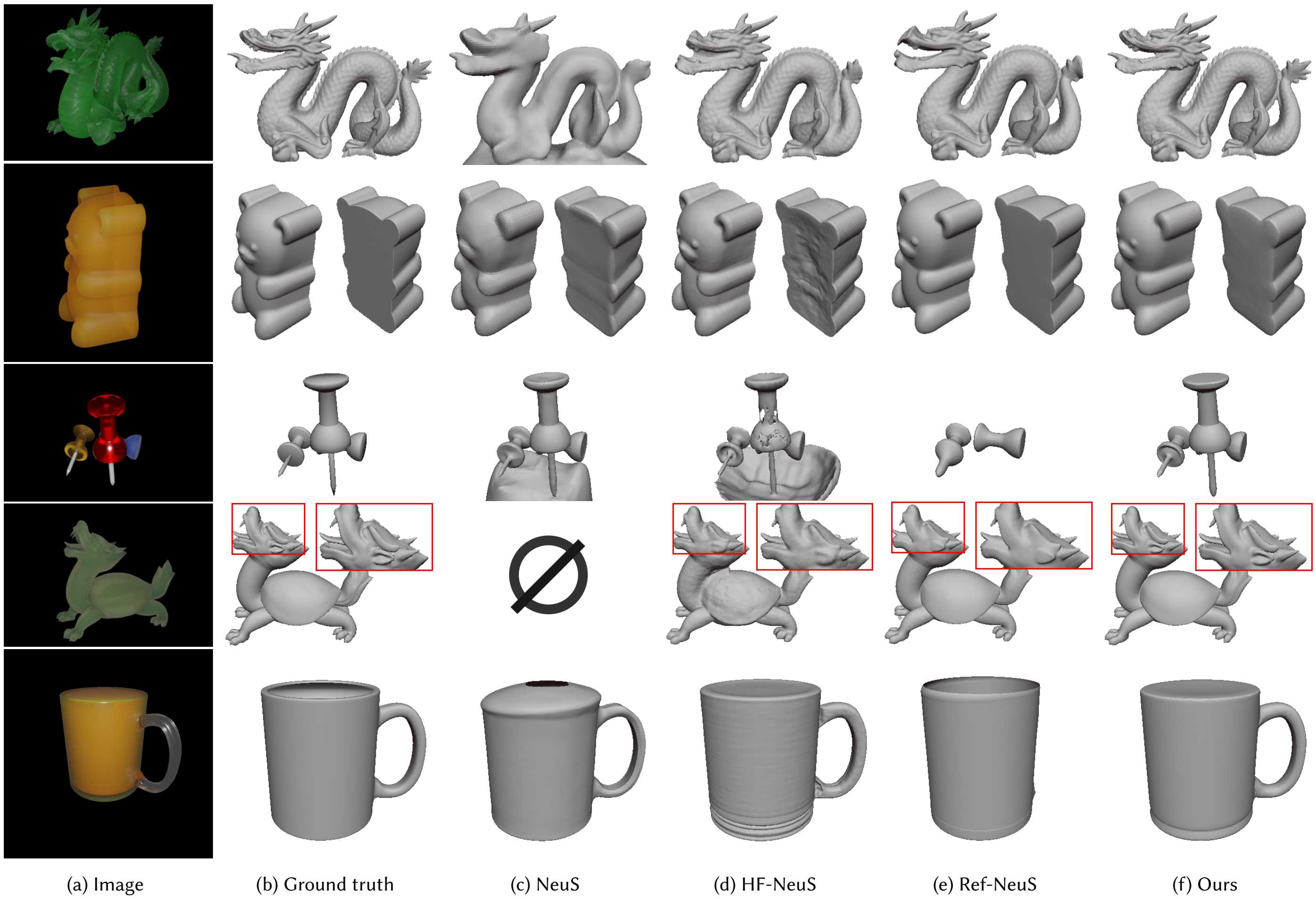


Fig. 3. **Reconstruction result.** We compare our result with the reconstruction method using implicit neural SDF: NeuS[Wang et al. 2021], HF-NeuS[Wang et al. 2022], Ref-NeuS[Ge et al. 2023] on our proposed "Syn-Trans" dataset.

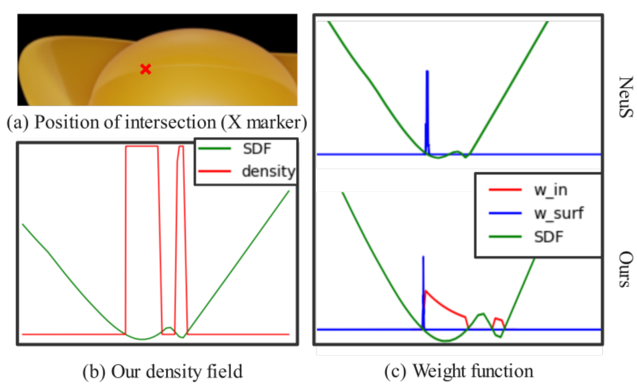
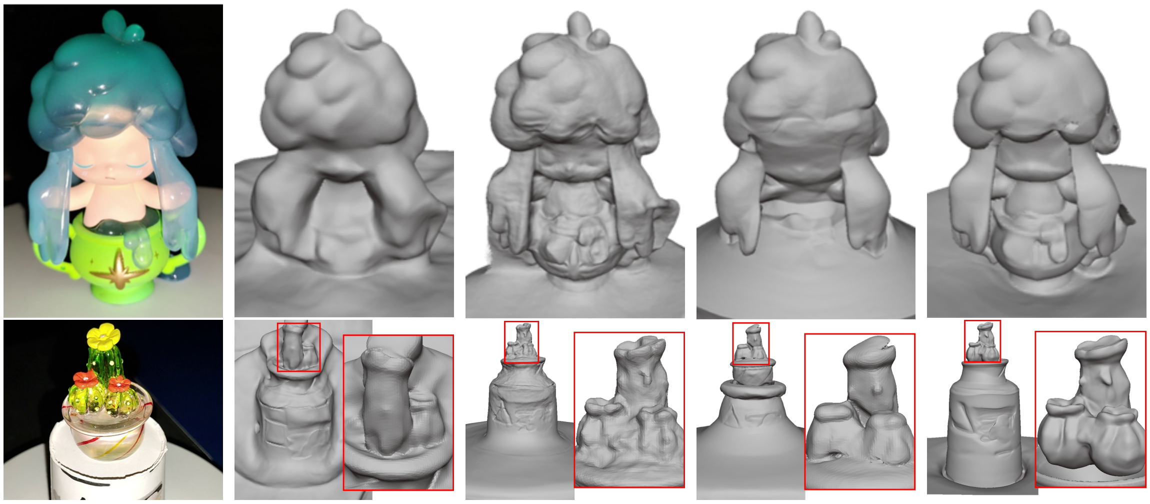


Fig. 4. **Visualization of weight function and density value at one camera ray.** Figure (b) shows our density inside the object. The learned $\beta = 2.53e{-}5$ in Eq. 6. Figure (c) shows the weight function of ours and NeuS. Note that the learned SDF value of NeuS is wrong as the interval between two surfaces is too small.

scenes. The method in Ref-NeuS is highly related to the accurate input view direction to compute reflection. In real scenes, it is hard to predict accurate camera poses. The inaccurate view direction led to the wrong surface in their result.

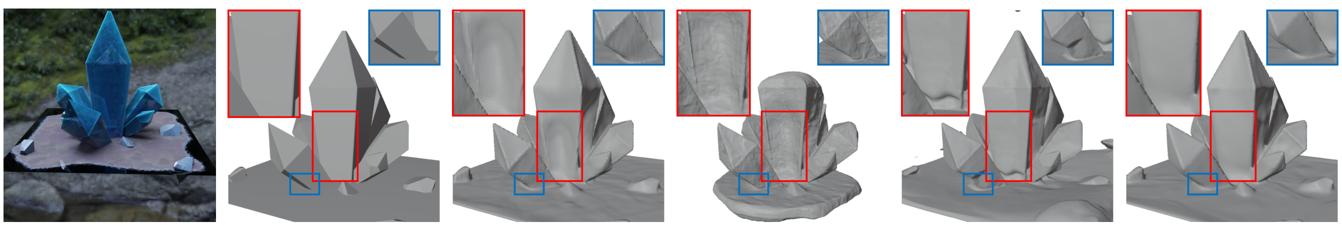
4.4 Evaluation of view synthesis

For evaluation of our neural rendering stage, we report the qualitative result of the novel view in Fig. 7. For quantitative comparison in Tab. 3, we compare the PSNR, LPIPS [Zhang et al. 2018], and SSIM [Schonberger and Frahm 2016] with the ground-truth result under the same lighting condition. IRON [Zhang et al. 2022] relies on the pipeline of NeuS to reconstruct geometry at stage one and decompose material at stage two. The rendering results in IRON show no transparency because of the omitted translucent appearance in their render equation and inaccurate geometry. The image metric is high in our method owing to the correct reconstructed geometry and learned scattering property. Note that IRON fails to recover the geometry of "Ancient Dragon", so their rendering result



(a) Image (b) NeuS (c) HF-NeuS (d) Ref-NeuS (e) Ours

Fig. 5. **Reconstruction result in real scene.** We evaluate the reconstruction result in the "Real-Trans" dataset. Note that, for the result in the last row, NeuS fails to recover a complete shape so we display their result in another view.



(a) Image (b) Ground truth (c) NeuS (d) HF-NeuS (e) Ref-NeuS (f) Ours

Fig. 6. **Reconstruction result on the natural scene.** This scene has environment lighting, which is different from the co-located flashlight setting.

Table 3. **Quantitative comparisons of rendering result under novel co-located flashlight views.** We compare the inverse rendering method in IRON [Zhang et al. 2022] with ours from photometric images using PSNR ↑, LPIPS ↓, and SSIM ↑.

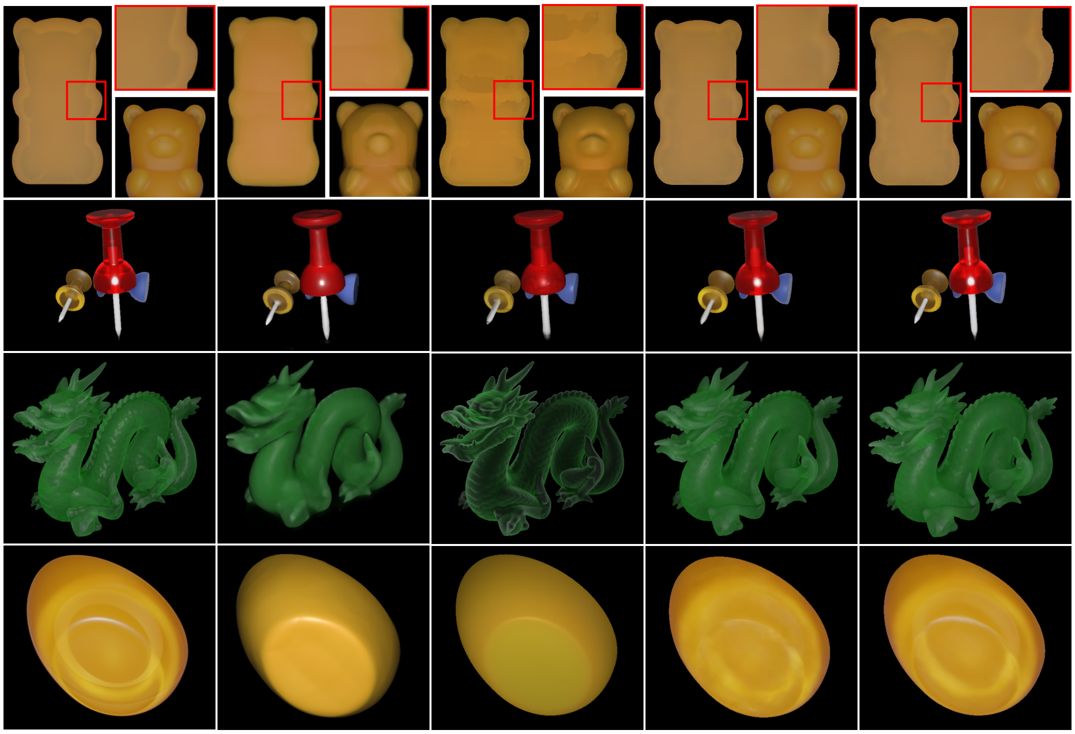
	IRON			Ours-w/o Lm			Ours-w/o cone			Ours		
	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
GummyBear	35.05	0.0322	0.976	34.47	0.0312	0.977	35.11	0.0275	0.979	39.17	0.0152	0.987
Stanford Dragon	28.43	0.1834	0.866	29.96	0.0739	0.917	41.90	0.0352	0.972	41.98	0.0356	0.972
Nail	25.99	0.0759	0.939	34.50	0.0318	0.976	36.76	0.0250	0.982	38.27	0.0217	0.986
Juice	27.71	0.0457	0.905	34.39	0.0623	0.970	38.95	0.0341	0.983	43.54	0.0172	0.988
Yuanbao	29.24	0.0648	0.957	32.16	0.0435	0.978	37.23	0.0245	0.983	40.18	0.0299	0.988
Ancient Dragon	14.57	0.1663	0.001	42.23	0.0235	0.988	43.72	0.0267	0.989	44.87	0.0194	0.991

only contains the black background. In Fig. 8, we show our view synthesis on real scenes compared with IRON.

4.5 Ablation studies

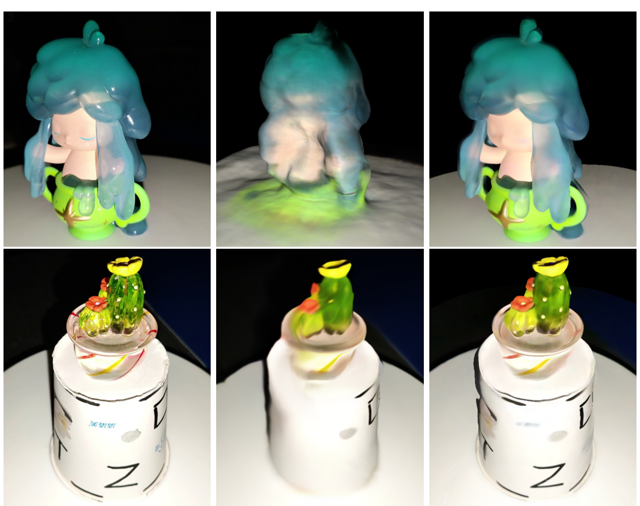
Model scattering property for rendering. We obtain a rendered result using Eq. 7, which can also used for novel-view synthesis. However, the scattering property is omitted in this equation. As a result, the learned transmission color contains noise and the quality of images rendered at the novel view is low. We report the average result of quantitative comparison in the second line of Tab. 4. The term "stage two" represents the neural rendering method in Sec. 3.4. The LPIPS score is almost the same in our experiments except for the scene "nail". The LPIPS value of rendered images using Eq. 7 is 0.0068 compared with 0.0217. It is a limitation of our method in optimizing parameters related to surface intersection points at thin regions in our method.

Calculation of w_{in} The scattering property only exists at points inside, so we restrict the sampling region to the interior of the object.



(a) GT Image (b) IRON (c) w/o Lm (d) w/o cone (e) Ours

Fig. 7. **Rendering result of novel view in "Syn-Trans"**. Columns (c) and (d) are the result of ablation methods.



(a) GT Image (b) IRON (c) Ours

Fig. 8. **Visualization of rendering at novel-view.**

We use our sample method as a kind of importance sampling, on account of preserving a clear boundary of inside points and outside points. Besides, we restrict the sum of the w_{in} for all sampled points to one, so that the integration of the w_{surf} and w_{in} after balanced by γ is equal to one, which is useful for deciding whether a ray hits the surface or not. We apply a normalization for our weight $\frac{w_{\text{in}}}{\sum w_{\text{in}}}$. We show the ablation results on whether to use our sampling method and normalization in Tab. 4.

Multi-scattering and multi-level conical sampling. The method in [Zheng et al. 2021] first proposes to resolve multi-scattering using SH. In their method, c_l^m is predicted using MLP, which takes positional encoding of points and view direction as input, without considering spatial information with multi-level sampling. Because their method can not reconstruct geometry, we leave the comparison with theirs as an ablation experiment. We implement their method as "w/o cone", which removes the conical sampling module in our method. Besides, to evaluate the effect of learned multi-scattering properties, we design an ablation experiment "w/o Lm", which represents that we omit the L_m in the final rendering equation. The rendering result and metric comparison are shown in Fig. 7 and Tab. 3. When the translucent appearance is complex and highly corresponds to the geometric shape of the unseen region, our method greatly improves the rendering result.

5 CONCLUSION

In this paper, we propose a novel framework for high-fidelity surface reconstruction and novel-view synthesis for translucent objects. Our framework contains two stages. We first reconstruct the surface of the translucent object using neural implicit SDF. We reparametrize

Table 4. **Results of experiments in ablation study.**The first table is the average result of Chamfer Distance. The second table is the average result of image metrics.

	Ours	w/o norm	w/o sampler
CD(Avg)	**0.00393**	0.00488	0.00549

	PSNR	LPIPS	SSIM
Ours	**41.33**	0.0233	**0.985**
w/o stage two	36.54	**0.0184**	0.886

the density field inside the object using an estimated constant extinction coefficient. Unlike the proposed "S-density" in NeuS, our density field maintains uniformity inside the object which is suitable for the physical property of the homogeneous object. Moreover, to perform a better rendering result at novel views, we exploit the learned geometry and learn translucent appearance using a neural representation of participating media. We propose a multi-level conical sampling method to learn complex translucent appearances related to the overall geometry shape. To evaluate our method, we create a dataset containing real-world translucent objects and synthetic ones.

Limitation. The refraction phenomenon is overlooked in both our reconstruction and rendering methods, which is crucial for highly transparent objects like glass. As shown in Fig. 9, our method fails to obtain a solid geometry and plausible rendering result. We leave the improvement on such scenes as our future work. The method for optimizing parameters in Sec. 3.4 leads to a poorer rendering result at thin regions of an object. A constraint on the invariance property of ξ is required explicitly in more general scenes. Besides, the scattering properties learned through training are limited by the viewpoints used and co-located lighting conditions. Our learned scattering property under insufficient training views performs badly on novel-view synthesis, especially in the real scene. We leave the modeling of more general scattering properties to future work.

(a) Test View (b) Reconstruction (c) Rendering(Test View)

Fig. 9. **Failure case on real scene: "cactus2"**

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China (62132012), the Fundamental Research Funds for the Central Universities (Nankai University, No. 63233080), and the Supercomputing Center of Nankai University(NKSC).

REFERENCES

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5855–5864.

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

Brent Burley. 2015. Extending the Disney BRDF to a BSDF with integrated subsurface scattering. *SIGGRAPH Course: Physically Based Shading in Theory and Practice. ACM, New York, NY* 19, 7 (2015), 9.

Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J Seron, and François X Sillion. 2005. A survey on participating media rendering techniques. *The Visual Computer* 21 (2005), 303–328.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*. Springer, 333–350.

Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. 2023. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4182–4194.

Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5939–5948.

Xi Deng, Fujun Luan, Bruce Walter, Kavita Bala, and Steve Marschner. 2022. Reconstructing translucent objects using differentiable rendering. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–10.

Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. 2018. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (ToG)* 37, 4 (2018), 1–15.

Yue Fan, Ivan Skorokhodov, Oleg Voynov, Savva Ignatyev, Evgeny Burnaev, Peter Wonka, and Yiqun Wang. 2023. Factored-NeuS: Reconstructing Surfaces, Illumination, and Materials of Possibly Glossy Objects. *arXiv preprint arXiv:2305.17929* (2023).

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5501–5510.

Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. 2022. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems* 35 (2022), 3403–3416.

Fangzhou Gao, Lianghao Zhang, Li Wang, Jiamin Cheng, and Jiawan Zhang. 2023. Transparent Object Reconstruction via Implicit Differentiable Refraction Rendering. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. 2022. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379* (2022).

Wenhang Ge, Tao Hu, Haoyu Zhao, Shu Liu, and Ying-Cong Chen. 2023. Ref-NeuS: Ambiguity-Reduced Neural Implicit Surface Learning for Multi-View Reconstruction with Reflection. *arXiv preprint arXiv:2303.10840* (2023).

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit Geometric Regularization for Learning Shapes. In *International Conference on Machine Learning*. PMLR, 3789–3799.

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

Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. 2015. The SGGX microflake distribution. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–11.

Jinkai Hu, Chengzhong Yu, Hongli Liu, Lingqi Yan, Yiqian Wu, and Xiaogang Jin. 2023. Deep real-time volumetric rendering using multi-feature fusion. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–10.

Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. 2007. Eikonal rendering: Efficient light transport in refractive objects. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 59–es.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. 2014. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 406–413.

Simon Kallweit, Thomas Müller, Brian Mcwilliams, Markus Gross, and Jan Novák. 2017. Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.

Berk Kaya, Suryansh Kumar, Francesco Sarno, Vittorio Ferrari, and Luc Van Gool. 2022. Neural radiance fields approach to deep multi-view photometric stereo. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1965–1977.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Chenhao Li, Trung Thanh Ngo, and Hajime Nagahara. 2023c. Inverse Rendering of Translucent Objects using Physical and Neural Renderers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12510–12520.

Jiyang Li, Lechao Cheng, Jingxuan He, and Zhangye Wang. 2023a. Current Status and Prospects of Research on Neural Radiance Fields. *Journal of Computer-Aided Design and Computer Graphics* (2023). https://doi.org/10.3724/SP.J.1089.2023-00376

Zongcheng Li, Xiaoxiao Long, Yusen Wang, Tuo Cao, Wenping Wang, Fei Luo, and Chunxia Xiao. 2023b. NeTO: Neural Reconstruction of Transparent Objects with Self-Occlusion Aware Refraction-Tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 18547–18557.

Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2020. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2475–2484.

Zhong Li, Liangchen Song, Zhang Chen, Xiangyu Du, Lele Chen, Junsong Yuan, and Yi Xu. 2023d. Relit-NeuLF: Efficient Relighting and Novel View Synthesis via Neural 4D Light Field. In *Proceedings of the 31st ACM International Conference on Multimedia*. 7007–7016.

Zhong Li, Liangchen Song, Celong Liu, Junsong Yuan, and Yi Xu. 2021. Neulf: Efficient novel view synthesis with neural 4d light field. *arXiv preprint arXiv:2105.07112* (2021).

Arvin Lin, Yiming Lin, and Abhijeet Ghosh. 2023. Practical Acquisition of Shape and Plausible Appearance of Reflective and Translucent Objects. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, e14889.

Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. 2023. NeRO: Neural Geometry and BRDF Reconstruction of Reflective Objects from Multiview Images. *arXiv preprint arXiv:2305.17398* (2023).

Jiahui Lyu, Bojian Wu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. 2020. Differentiable refraction-tracing for mesh reconstruction of transparent objects. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–13.

Linjie Lyu, Ayush Tewari, Thomas Leimkühler, Marc Habermann, and Christian Theobalt. 2022. Neural radiance transfer fields for relightable novel-view synthesis with global illumination. In *European Conference on Computer Vision*. Springer, 153–169.

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

Tai-Jiang Mu, Hao-Xiang Chen, Jun-Xiong Cai, and Ning Guo. 2023. Neural 3D reconstruction from sparse views using geometric priors. *Computational Visual Media* 9, 4 (2023), 687–697.

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

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3504–3515.

Michael Oechsle, Songyou Peng, and Andreas Geiger. 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5589–5599.

Jiaxiong Qiu, Peng-Tao Jiang, Yifan Zhu, Ze-Xin Yin, Ming-Ming Cheng, and Bo Ren. 2023. Looking Through the Glass: Neural Surface Reconstruction Against High Specular Reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20823–20833.

Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4104–4113.

Zeqi Shi, Xiangyu Lin, and Ying Song. 2023. An attention-embedded GAN for SVBRDF recovery from a single image. *Computational Visual Media* 9, 3 (2023), 551–561.

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. 2022. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5481–5490.

Jiaping Wang, Shuang Zhao, Xin Tong, Stephen Lin, Zhouchen Lin, Yue Dong, Baining Guo, and Heung-Yeung Shum. 2008. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Transactions on Graphics (TOG)* 27, 1 (2008), 1–18.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *Advances in Neural Information Processing Systems* 34 (2021), 27171–27183.

Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. 2022. Hf-neus: Improved surface reconstruction using high-frequency details. *Advances in Neural Information Processing Systems* 35 (2022), 1966–1978.

Zongji Wang, Yunfei Liu, and Feng Lu. 2023. Discriminative feature encoding for intrinsic image decomposition. *Computational Visual Media* 9, 3 (2023), 597–618.

Haoyu Wu, Alexandros Graikos, and Dimitris Samaras. 2023. S-VolSDF: Sparse Multi-View Stereo Regularization of Neural Implicit Surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3556–3568.

Jingjie Yang and Shuangjiu Xiao. 2016. An inverse rendering approach for heterogeneous translucent materials. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*. 79–88.

Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K Wong. 2022. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In *European Conference on Computer Vision*. Springer, 266–284.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. 2020. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1790–1799.

Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2022. Neilf: Neural incident light field for physically-based material estimation. In *European Conference on Computer Vision*. Springer, 700–716.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems* 34 (2021), 4805–4815.

Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. 2023. BakedSDF: Meshing Neural SDFs for Real-Time View Synthesis. *arXiv preprint arXiv:2302.14859* (2023).

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. 2020. Multiview neural surface reconstruction with implicit lighting and material. *Adv. Neural Inform. Process. Syst* 1, 2 (2020), 3.

Hong-Xing Yu, Michelle Guo, Alireza Fathi, Yen-Yu Chang, Eric Ryan Chan, Ruohan Gao, Thomas Funkhouser, and Jiajun Wu. 2023. Learning object-centric neural scattering functions for free-viewpoint relighting and scene composition. *arXiv preprint arXiv:2303.06138* (2023).

Junyi Zeng, Chong Bao, Rui Chen, Zilong Dong, Guofeng Zhang, Hujun Bao, and Zhaopeng Cui. 2023. Mirror-NeRF: Learning Neural Radiance Fields for Mirrors with Whitted-Style Ray Tracing. In *Proceedings of the 31st ACM International Conference on Multimedia*. 4606–4615.

Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. 2021c. Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. *Advances in Neural Information Processing Systems* 34 (2021), 29835–29847.

Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2023b. NeILF++: Inter-Reflectable Light Fields for Geometry and Material Estimation. *arXiv preprint arXiv:2303.17147* (2023).

Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. 2022. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5565–5574.

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021a. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5453–5462.

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

Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. 2021b. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)* 40, 6 (2021), 1–18.

Youjia Zhang, Teng Xu, Junqing Yu, Yuteng Ye, Yanqing Jing, Junle Wang, Jingyi Yu, and Wei Yang. 2023a. Nemf: Inverse volume rendering with neural microflake field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22919–22929.

Quan Zheng, Gurprit Singh, and Hans-Peter Seidel. 2021. Neural relightable participating media rendering. *Advances in Neural Information Processing Systems* 34 (2021), 15203–15215.

Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker. 2022. Irisformer: Dense vision transformers for single-image inverse rendering in indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2822–2831.

Shizhan Zhu, Shunsuke Saito, Aljaz Bozic, Carlos Aliaga, Trevor Darrell, and Christop Lassner. 2023. Neural Relighting with Subsurface Scattering by Learning the Radiance Transfer Gradient. *arXiv preprint arXiv:2306.09322* (2023).

A APPENDIX

A.1 Formula Derivation

We analyze the theoretical weight function on the constant density field in the section. Recall that we solve the volume rendering

Table 5. **Detailed camera setting, images, and material in our synthesis dataset.**

Name	Number of Images	Camera Setting	Material Parameters				
			Color	Subsurface Color	Specular	Roughness	Transmission
GummyBear	80	sphere	0.80,0.36,0.05	0.80,0.23,0.00	0.5	0.5	0.8
Stanford Dragon	100	semi-sphere	0.0,1.0,0.1	0.09,0.42,0.05	0.1	0.35	0.7
Yuanbao	80	sphere	1.0,0.7,0.05	1.0,0.33,0.0	0.1	0.4	0.8
Ancient Dragon	120	semi-sphere	0.79,0.67,0.44	0.50,0.80,0.33	0.5	0.75	0.8
Nail	80	semi-sphere	(1,0,0), (0.1,0.23,1), (1,0.66,0,09)		0.63,0.63,0.1	0.1,0.8,0.4	0.95,0.95,0.9
Juice	100	semi-sphere	1.0,0.6,0.0	1.0,0.28,0.0	0.0	0.5	0.8
Doll	125	circle	\	\	\	\	\
Cactus1	147	circle	\	\	\	\	\
Cactus2	121	circle	\	\	\	\	\

equation using Eq. 3. The density is assumed constant for points inside the object, so we use σ_t to represent the constant value: $\alpha_j = 1 - \exp(-\sigma_t \cdot \delta_j)$. Substituting this formula into the weight function, we get the weight function for points inside the object.

$$w_j = \alpha_j \prod_{i=1}^{j-1}(1-\alpha_i) = (1-\exp(-\sigma_t \cdot \delta_j)) \cdot \exp(-\sigma_t \cdot (t_j - t_1)) \quad (17)$$

For uniformly sampled points, the δ_j remains equal for all j, so we can simplify this equation to:

$$w_j = (1 - \exp(-\sigma_t \cdot \delta_t)) \cdot \exp(-\sigma_t \cdot \delta_t \cdot (j-1)) \quad (18)$$

where $\delta_t = (t_f - t_n)/N$. t_n, t_f represents the near and far in the NeRF. N is the number of sampled points. For a camera ray, δ_t is fixed. We assume σ_t is constant so $\sigma_t \cdot \delta_t$ can be regarded as a constant value.

A.2 Additional Experiments

Comparsion with VolSDF. The comparison result of geometry reconstruction with VolSDF [Yariv et al. 2021] is shown in Fig. 10 and Tab. 6. The unlisted scenes are those VolSDF failed to reconstruct.

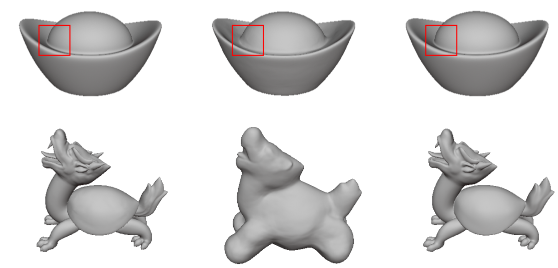
Comparsion with the method specifically designed for the translucent objects. For the method in [Deng et al. 2022], they utilize differentiable BSSRDF path-tracing to reconstruct translucent objects. However, their reconstruction largely requires a suitable geometry initialization and the geometric topology of the examples in their paper is relatively simple. They assume that the rendered model is BSSRDF, which does not comply with our dataset. Besides, the GPU memory used in their method is too large, so we reduced the image resolution in our dataset to 256x256 (512x512 used in the original paper) in this experiment. Their reconstruction result on the "gummybear" scene is shown in Fig. 11.

A.3 Details of Dataset

The detailed information in our dataset can be found in Tab. 5. The camera setting represents the sample region of the camera. "Sphere" means sampling camera at a unit sphere while "semi-sphere" represents sampling only on the upper surface of the sphere. For the materials, we use the PrincipleBSDF shader in Blender, which is an implementation of Disney BSDF [Burley 2015]. The values of "Color" and "Subsurface Color" are RGB values in floating format. The omitted parameters like metallic, sheen, and clearcoat are zero. The IOR is set to 1.3 in all scenes except for the scene "Juice", which contains a glass cup. The parameters in scene "Nail" correspond to three objects separately and no "Subsurface Color" is set for this scene.

Table 6. **Reconstruction evaluation result.**

Scene	VolSDF	Ours
GummyBear	0.0036	**0.0011**
Juice	0.0152	**0.0132**
Yuanbao	0.0065	**0.0012**
Ancient Dragon	0.0550	**0.0022**



(a) Ground truth (b) VolSDF (c) Ours

Fig. 10. **Reconstruction result compared with VolSDF.**

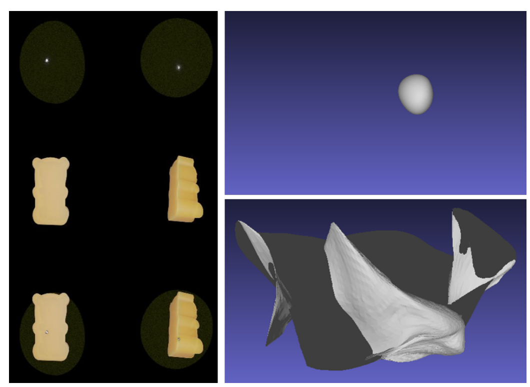


Fig. 11. **Method "InvTranslucent" in [Deng et al. 2022] failed to recover the geometry**.