Learning Object-Centric Neural Scattering Functions for Free-viewpoint Relighting and Scene Composition

Anonymous authors
Paper under double-blind review

Abstract

Photorealistic object appearance modeling from 2D images is a constant topic in vision and graphics. While neural implicit methods (such as Neural Radiance Fields) have shown high-fidelity view synthesis results, they cannot relight the captured objects. More recent neural inverse rendering approaches have enabled object relighting, but they represent surface properties as simple BRDFs, and therefore cannot handle translucent objects. We propose Object-Centric Neural Scattering Functions (OSFs) for learning to reconstruct object appearance from only images. OSFs not only support free-viewpoint object relighting, but also can model both opaque and translucent objects. While accurately modeling subsurface light transport for translucent objects can be highly complex and even intractable for neural methods, OSFs learn to approximate the radiance transfer from a distant light to an outgoing direction at any spatial location. This approximation avoids explicitly modeling complex subsurface scattering, making learning a neural implicit model tractable. Experiments on real and synthetic data show that OSFs accurately reconstruct appearances for both opaque and translucent objects, allowing faithful free-viewpoint relighting as well as scene composition. In our supplementary material, we include a video for an overview.

1 Introduction

Modeling the geometry and appearance of 3D objects from captured 2D images is central to many applications in computer vision, graphics, and robotics, such as shape reconstruction (Mescheder et al., 2019; Park et al., 2019; Chang et al., 2015), view synthesis (Hedman et al., 2018; Mildenhall et al., 2020; Sitzmann et al., 2019a), relighting (Zhang et al., 2021c; Bi et al., 2020b;a), and object manipulation (Simeonov et al., 2021). Traditional inverse rendering approaches (Zhou et al., 2013; Nam et al., 2018) focus on directly estimating objects' shape and material properties; they then compose them using the estimated surface properties. These explicitly reconstructed object representations often lead to visible artifacts or limited fidelity when rendering images using the recovered object assets.

Recently, neural implicit methods have attracted much attention in image-based appearance modeling. They implicitly represent objects and scenes using deep neural networks (Lombardi et al., 2019; Sitzmann et al., 2019a;b). Neural Radiance Fields (NeRFs) (Mildenhall et al., 2020) are one of the most representative methods due to their high-fidelity view synthesis results. NeRFs implicitly model a scene's volumetric density and radiance via coordinate-based deep neural networks, which can be learned from images via direct volume rendering techniques (Max, 1995). However, NeRFs only reconstruct the outgoing radiance fields under a fixed lighting condition. Thus, they cannot relight the captured objects or compose multiple objects into new scenes.

To address this problem, a few neural inverse rendering methods (Zhang et al., 2021b;a; Boss et al., 2021a; Zhang et al., 2021c) have been proposed to jointly estimate lighting, material properties, and geometry from images. The disentanglement of illumination enables relightable object appearance modeling. However, these methods all assume simple Bidirectional Reflectance Distribution Functions (BRDFs) when modeling surface properties, without taking subsurface light transport into consideration*. Therefore, they can only model the appearance of opaque objects and cannot handle translucent objects with complex material properties.

*We use "simple BRDFs" to refer to BRDFs that do not consider subsurface scattering.

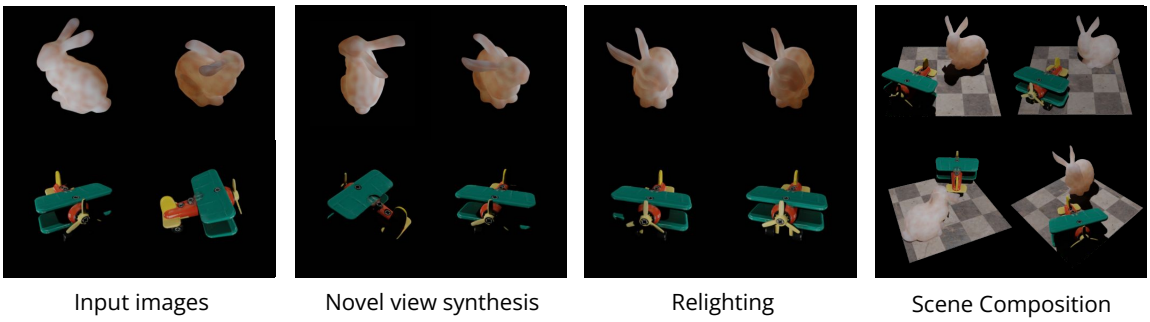


Figure 1: We propose a new neural object representation, Object-Centric Neural Scattering Functions (OSFs), to reconstruct object appearance from only images. OSFs can handle objects with complex materials or shapes (e.g., the translucent bunny), and support both free-viewpoint relighting and scene composition.

We observe that subsurface scattering effects are the key to relighting translucent objects. However, accurately modeling subsurface scattering can be complex and even intractable for neural implicit representations, because each sampling operation for numerical integration requires a complete forward pass of the network. This leads to a prohibitive computational cost in both time and memory.

To address this challenge, we propose Object-Centric Neural Scattering Functions (OSFs) for image-based relightable neural appearance reconstruction. An OSF learns to approximate the *cumulative radiance transfer function*, which models radiance transfer from a distant light to any outgoing direction at any spatial location for an object. This enables the use of a volumetric rendering formulation to learn from images, while being able to approximate both opaque and translucent object appearances. Thus, OSFs allow free-viewpoint relighting from only 2D images (see Figure 1).

Modeling object-level light transport allows neural scene composition with trained OSF models, as light transport effects such as object shadows and indirect lighting can be naturally incorporated. To demonstrate this, we present a simple and computationally tractable method for scene composition. To further accelerate neural scene rendering, we introduce two improvements on OSFs scene rendering that allow magnitudes of acceleration without compromising rendering quality.

Experiments on both real and synthetic objects demonstrate the effectiveness of our OSF formulation. We show that OSFs enable free-viewpoint relighting of both translucent and opaque objects by learning from only 2D images. We also show that in scene composition using the reconstructed objects, our approach significantly outperforms baseline methods.

In summary, our contributions are threefold. First, we introduce the concept of the cumulative radiance transfer function that models object-centric light transport. Second, we propose the Object-Centric Neural Scattering Functions (OSFs) that learn to approximate the cumulative radiance transfer function from only 2D images. OSFs allow free-viewpoint relighting and scene composition of *both opaque and translucent objects*. Third, we show experimental results on both real and synthetic objects, demonstrating the effectiveness of OSFs in free-viewpoint relighting and scene composition.

2 Related Work

Neural appearance reconstruction. Traditional methods use Structure-From-Motion (Hartley & Zisserman, 2003) and bundle adjustment (Triggs et al., 1999) to reconstruct colored point clouds. More recently, a number of learning-based novel view synthesis methods have been proposed, but they require 3D geometry as inputs (Hedman et al., 2018; Thies et al., 2019; Meshry et al., 2019; Aliev et al., 2020; Martin-Brualla et al., 2018). Recently, neural volume rendering approaches have been used to reconstruct scenes from only images (Lombardi et al., 2019; Sitzmann et al., 2019a). However, the rendering resolution of these methods is limited by the time and computational complexity of the discretely sampled volumes. To address this issue, Neural Radiance Fields (NeRFs) (Mildenhall et al., 2020) directly optimizes a *continuous* radiance

field representation using a multi-layer perceptron. This allows for synthesizing novel views of realistic images at an unprecedented level of fidelity. Various extensions of NeRF have also been proposed to improve efficiency (Liu et al., 2020; Yu et al., 2021; Garbin et al., 2021; Reiser et al., 2021), generalization (Niemeyer & Geiger, 2021; Yu et al., 2022; Yang et al., 2021), quality (Oechsle et al., 2021; Barron et al., 2021a;b; Verbin et al., 2021; Zhang et al., 2020), compositionality (Ost et al., 2021) etc. While these neural methods produce high-quality novel views of a scene, they do not support relighting. In contrast, our approach aims at relightable appearance reconstruction.

Learning-based relighting. Learning-based methods that relight images without explicit geometric reasoning have been proposed (Sun et al., 2019; Xu et al., 2018; Zhou et al., 2019), but they are not 3D-aware. Following the recent surge of neural 3D scene representations and neural differentiable rendering (Tewari et al., 2020; Mildenhall et al., 2020), neural relightable representations (Bi et al., 2020b;a; Boss et al., 2021a; Srinivasan et al., 2021; Zhang et al., 2021c; Baatz et al., 2021) and neural inverse rendering methods (Zhang et al., 2021b;a; Boss et al., 2021b) have been developed to address this limitation. However, most of them assume simple BRDFs which do not consider subsurface scattering in their formulations, and thus they cannot model translucent object appearances for relighting. In contrast, our method allows the reconstruction of both opaque and translucent object appearances. Some existing methods also approximate subsurface light transport for participating media (Kallweit et al., 2017; Zheng et al., 2021). However, they either need specific designs such as spherical harmonics and BRDF modeling (Zheng et al., 2021) or require 3D groundtruth data (Kallweit et al., 2017), while our method learns a general object representation from only images.

Precomputed light transport. Our OSFs learn to approximate a cumulative radiance transfer function, which is based on the classical idea from the precomputed light transport methods including the precomputed radiance transfer (Sloan et al., 2002) (also see (Lehtinen, 2007) and (Ramamoorthi, 2009) for overviews). These methods map incoming basis lighting to outgoing radiance, while we additionally learn to encapsulate the object-specific light transport at the visible surface. Related ideas include the classical reflectance field (Debevec et al., 2000; Garg et al., 2006). Our method is also related to classical techniques for modeling aggregate scattering behavior (Moon et al., 2007; Lee & O'Sullivan, 2007; Meng et al., 2015; Blumer et al., 2016). These techniques consider modeling the full asset-level internal scattering, a more general form than our cumulative radiance transfer. Our cumulative radiance transfer function borrows the general idea from these classical methods and adapts to neural implicit volumetric rendering framework for relightable and compositional appearance modeling from images. This adaptation allows using the power of deep networks to approximate complex light transport. Recently a few neural methods have been proposed to model light transport of complex materials (Kuznetsov, 2021; Baatz et al., 2021), e.g., Baatz et al. (2021) represent surface texture as neural reflectance fields which is technically relevant to ours. However, they need meshes as explicit object geometry, while ours learns object representations from only images.

3 Approach

We aim for relightable neural appearance reconstruction of both opaque and translucent objects from 2D images. To achieve this, we propose Object-centric neural Scattering Functions (OSFs) that learn to approximate subsurface scattering effects in a computationally tractable way. In the following, we first introduce our OSF formulation and discuss its relations to phase functions and BSSRDFs (Sec. 3.1). Then we show how we can model translucent objects (Sec. 3.2) and opaque objects (Sec. 3.3) with OSFs. Finally, we discuss how to learn OSFs from 2D images (Sec. 3.4) and compose multiple learned OSFs (Sec. 3.5).

3.1 OSF Formulation

Our OSFs are based on Neural Radiance Fields (NeRFs) (Mildenhall et al., 2020), which learn view-dependent emitted radiance at any point in space. NeRFs can be formulated by $\{L_{\text{out}}(\mathbf{x}, \boldsymbol{\omega}_{\text{out}}), \sigma(\mathbf{x})\} = \text{NN}(\mathbf{x}, \boldsymbol{\omega}_{\text{out}})$, where L_{out} denotes the outgoing radiance, $\mathbf{x}$ denotes a 3D spatial location, σ denotes the volumetric density, $\boldsymbol{\omega}_{\text{out}}$ denotes an outgoing radiance direction, and NN denotes a learnable deep neural network. Inspired from the principles of volume rendering (Kajiya & Von Herzen, 1984), the computational model for radiance $L(\mathbf{r})$

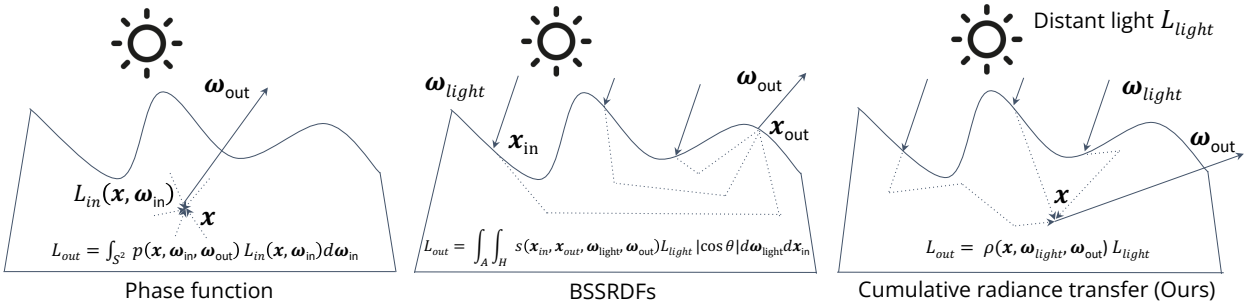


Figure 2: **Relations to classical graphics concepts for modeling subsurface scattering effects.** The process of light transport from an unobstructed distant light to a point $\mathbf{x}$ inside a translucent object is complex: 1) the light (radiance) hit a point on the visible surface area; 2) the surface transmits some amount of the light into the object interior; 3) the transmitted light scatters within the object; 4) some light finally arrives at $\mathbf{x}$ from incoming directions $\boldsymbol{\omega}_{\text{in}}$; 5) the arrived light scatters to directions $\boldsymbol{\omega}_{\text{out}}$; and 6) this process happens at all points on the visible surface, and the final outcome is the *accumulated* result. Phase functions only model 4) and 5), so one has to account for the other steps by multiple integrations. BSSRDFs encapsulate steps 2) - 5), so they require two integrations for step 1) and step 6). In contrast, our OSF formulation encapsulates the entire process that takes all steps into account. This abstraction of modeling the whole process as cumulative radiance transfer eliminates the need of any integration, making it computationally tractable to use in neural volume rendering.

of a camera ray $\mathbf{r}(t) = \mathbf{o} - t\boldsymbol{\omega}_{\text{out}}$ with near and far bounds t_n and t_f is given by:

$$L(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))L_{\text{out}}(\mathbf{r}(t), \boldsymbol{\omega}_{\text{out}})dt, \tag{1}$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)$ denotes the transmittance along the ray from t_n to t.

While a NeRF can render high-fidelity novel views of a captured object, it does not support relighting because it is based on the absorption-plus-emission model (Max, 1995), which itself is light source–agnostic. Namely, NeRFs model L_{out} as the *emitted* radiance after all light transport has occurred. To address this limitation, we propose OSFs that model L_{out} as the *scattered* radiance due to external light sources instead. More specifically, OSFs learn to approximate the cumulative radiance transfer from an unobstructed distant light, in addition to the volume density:

$$\{\rho(\mathbf{x}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}), \sigma(\mathbf{x})\} = \text{NN}(\mathbf{x}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}), \tag{2}$$

where $\rho(\mathbf{x}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}})$ denotes the *cumulative radiance transfer function*, which is defined as the ratio of two quantities. The numerator is the outgoing radiance $L_{\text{out}}(\mathbf{x}, \boldsymbol{\omega}_{\text{out}})$ in the direction of $\boldsymbol{\omega}_{\text{out}}$ at location $\mathbf{x}$ that is caused by an unobstructed distant light $L_{\text{light}}(\boldsymbol{\omega}_{\text{light}})$ from direction $\boldsymbol{\omega}_{\text{light}}$. The denominator is $L_{\text{light}}(\boldsymbol{\omega}_{\text{light}})$. It is *cumulative* about all paths that $L_{\text{light}}(\boldsymbol{\omega}_{\text{light}})$ scatters to $\mathbf{x}$. See Figure 2 for an illustration.

Given the cumulative radiance transfer function ρ, the scattered outgoing radiance L_{out} is given by

$$L_{\text{out}}(\mathbf{x}, \boldsymbol{\omega}_{\text{out}}) = \int_{S^2} \rho(\mathbf{x}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}})L_{\text{light}}(\boldsymbol{\omega}_{\text{light}})d\boldsymbol{\omega}_{\text{light}}. \tag{3}$$

We assume the object itself does not emit light. We argue that this distant light assumption in ρ is acceptable and can allow relighting with environment maps that are also distant light.

Phase Functions vs. BSSRDFs. vs. Cumulative Radiance Transfer. The cumulative radiance transfer function ρ approximates a complex radiance transfer process from an unobstructed distant light to an outgoing direction at a location, including those inside the object. It is conceptually akin to a phase function that describes the scattered radiance distribution at a specific location in a scattering process. It is

also related to the Bidirectional Scattering Surface Reflectance Distribution Functions (BSSRDFs) (Jensen et al., 2001). See Figure 2 for a detailed comparison of these concepts.

Below we list the major distinctions against phase functions:

- Generally, a phase function is local, i.e., it depends only on the material/media (and in some cases also geometry) at the location. In contrast, the cumulative radiance transfer function ρ is object-specific and spatially varying even if the object is made from a uniform homogeneous material. This is because ρ potentially needs to account for a complex process, including the light transport for all surface points that are visible to the distant light, the subsurface scattering effects, and the radiance transfer at a specific location. Thus, this process depends on both the material and the object geometry at $\mathbf{x}$, as well as the light direction.
- Phase functions for some naturally occurring isotropic media only need 1 Degree of Freedom (DoF) for angular distribution, i.e., the cosine between incoming and outgoing radiance directions (see Chapter 11 in Pharr et al. (2016)). However, ρ generally needs 4 DoF for angular distribution: 2 DoF of light direction because the light direction can interact with object geometry to affect the incoming radiance distribution at any given $\mathbf{x}$, and 2 DoF of outgoing direction as we want to model view-dependent effects as in NeRFs (Mildenhall et al., 2020).

Compared to BSSRDFs, which are surface models that describe how light is transported between any two rays that hit a surface, ρ is used in a volumetric rendering process. OSFs do not explicitly model surface properties.

3.2 Modeling Translucent Objects with OSFs

Our OSF formulation can model the appearance of translucent objects in a computationally tractable way for volumetric neural implicit methods, while directly modeling the radiance transfer in a volumetric scattering process is intractable. In direct modeling, $L_{\text{out}}(\mathbf{x}, \boldsymbol{\omega}_{\text{out}}) = \int_{S^2} p(\mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\omega}_{\text{out}}) L_{\text{in}}(\boldsymbol{\omega}) d\boldsymbol{\omega}$, where p denotes a phase function, L_{in} denotes the incoming radiance at location $\mathbf{x}$ from the direction $\boldsymbol{\omega}$. The recursive dependency of L_{in} over the unit sphere in the media makes the equation very difficult to solve (Novák et al., 2018; Max & Chen, 2005). From the computational perspective, to numerically compute this integral to train the neural network, one needs to account for the whole sphere (due to subsurface scattering) through dense sampling. However, for neural implicit methods, each sample requires a forward pass of the deep network, leading to a high computation cost. Moreover, training deep networks often requires saving intermediate variables for auto-differentiation (Paszke et al., 2017), which demands a massive amount of GPU memory for dense sampling. As a result, it can become computationally intractable to train the neural network in this way. In contrast, during training, we use lighting setups that match well the assumption of a single directional light source; the spherical integration in Eq. (3) thus reduces to a single evaluation. The key feature of OSFs is that they approximate the complex cumulative radiance transfer process using the expressive power of deep neural networks.

3.3 Modeling Opaque Objects with OSFs

Apart from modeling translucent objects, our OSF formulation is also able to model opaque objects. Theoretically, the formulation may subsume surface light transport when modeling opaque convex objects. To see this, we consider a single distant light source from $\boldsymbol{\omega}_{\text{light}}$, the light transport at a visible surface point can be described by

$$L(\mathbf{r}) = f(\mathbf{x}_{\text{surf}}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}) L_{\text{light}} \left| \mathbf{n}(\mathbf{x}_{\text{surf}}) \cdot \boldsymbol{\omega}_{\text{light}} \right|, \tag{4}$$

where f denotes a BRDF, $\mathbf{x}_{\text{surf}}$ denotes the first surface point hit by the camera ray $\mathbf{r}$, and $\mathbf{n}(\mathbf{x})$ denotes the unit normal vector at $\mathbf{x}$. We have used the single distant light assumption to solve the integration. Now we consider an ideal case, where an OSF learns that $\sigma(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_{\text{surf}})$ for all $\mathbf{x}_{\text{surf}}$. This solves the integration in Eq. (1) and gives

$$L(\mathbf{r}) = L_{\text{out}}(\mathbf{x}_{\text{surf}}, \boldsymbol{\omega}_{\text{out}}) = \rho(\mathbf{x}_{\text{surf}}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}) L_{\text{light}}. \tag{5}$$

We can see that if the learned OSF satisfies $\rho(\mathbf{x}_{\text{surf}}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}) = f(\mathbf{x}_{\text{surf}}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}) \left|\mathbf{n}(\mathbf{x}_{\text{surf}}) \cdot \boldsymbol{\omega}_{\text{light}}\right|$ for all surface points, then it subsumes the surface light transport with the BRDF. Although a neural network cannot perfectly represent a delta function for $\sigma(\mathbf{x})$, in our experiments, we observe that empirically OSFs learn to approximate the surface light transport for opaque objects of both convex and concave shapes.

3.4 Learning OSFs from Images

We aim to learn OSFs from only 2D images, allowing easy capture of real objects. As discussed in Section 3.1, we prefer a tractable training setup. Thus, we propose capturing object images using only a single distant light source, which leads to a light source radiance function $L_{\text{light}}(\omega) = L_0\delta(\omega - \omega_0)$, where L_0 and ω_0 are obtained from the capture setup. We can define the sampling Probabilistic Density Function (PDF) for lighting as $\text{PDF}_{\text{light}}(\omega) = \delta(\omega - \omega_0)$ and this gives an analytic solution of the integration in Eq. (3):

$$L_{\text{out}}(x, \boldsymbol{\omega}_{\text{out}}) = \mathop{\mathbb{E}}_{\omega \sim \text{PDF}_{\text{light}}(\omega)} \left[\rho(x, \omega, \boldsymbol{\omega}_{\text{out}}) \frac{L_{\text{light}}(\omega)}{\text{PDF}_{\text{light}}(\omega)} \right] \tag{6}$$

$$= \rho(x, \omega_0, \boldsymbol{\omega}_{\text{out}}) L_0. \tag{7}$$

We capture images of the object-to-reconstruct under different ω_0 in a dark room. We describe our accessible real data capture setup using two iPhones in Section 4.1.

Given Eq. (7), the camera ray radiance $L(r)$ defined in Eq. (1) is trivially differentiable with respect to OSFs. We follow the learning strategy from NeRFs (Mildenhall et al., 2020), which models the pixel color of a ray in an analog to radiance instead of the raw radiometric quantity to simplify learning. This may be interpreted as allowing the neural network to implicitly deal with the camera response that maps camera ray radiance to pixel color (also see a recent discussion in Mildenhall et al. (2021)). We also follow NeRFs to use quadrature to compute Eq. (1), and use positional encoding and hierarchical volume sampling to facilitate training (i.e., we use a coarse network for coarse sampling a ray to give a rough estimation of transmittance distribution, and then use a fine network to do additional informed sampling to further reduce variance). The loss function for learning an OSF is thus defined as

$$\mathcal{L} = \mathbb{E}_r \left[\|\hat{C}_{\text{coarse}}(r) - C(r)\|^2 + \|\hat{C}_{\text{fine}}(r) - C(r)\|^2 \right], \tag{8}$$

where $C(r)$ denotes the groundtruth pixel color for ray r, $\hat{C}_{\text{coarse}}(r)$ denotes the predicted color by the coarse network, and $\hat{C}_{\text{fine}}(r)$ denotes the predicted color by the fine network. During testing, we only use the predicted colors of the fine network.

3.5 Composing Multiple Learned OSFs

We show how we can use the learned OSFs for visually plausible scene composition. Note that due to the complexity of the problem and the approximations made by OSFs, physical correctness may not be guaranteed in the following discussion. Below we present a simple and computationally tractable approximation for composing multiple learned OSFs. We leave the design of physically correct scene composition algorithms as future work.

We restrict our discussion to a scene with multiple objects represented by OSFs, and we assume a single distant light source to illuminate the scene. In this case, the light distribution in Eq. (3) can potentially contain two parts: one for direct lighting, and the other for indirect lighting. We discuss them separately below, and then provide an algorithm for rendering a composed scene.

Direct lighting. Direct lighting is assumed to be a distant light, so the major point to consider is solving visibility. Notice that although OSFs allow reconstructing translucent object appearances, we do not model advanced light transport effects including transmitted lights or caustics in this work, but only consider shadows caused by light visibility. While solving for hard visibility can be tricky for OSFs, because they do not have the concept of surface just like NeRFs, the transmittance T in Eq. (1) can be seen as a soft measure for visibility. Therefore, we use a shadow ray $r_{\text{shadow}}(t) = x - t\omega_{\text{light}}$ to solve for visibility at x. We volume-render this shadow ray for all other OSFs in the scene with compositional rendering (Niemeyer &

Geiger, 2021), and use the obtained transmittance $T(t) = \exp(-\int_{t_n}^{t} \sigma(r_{\text{shadow}}(s))ds)$ to serve as visibility: $L_{\text{direct}} = T(t_f)L_0$, where we abuse t_f to denote the far bound of the shadow ray and L_0 to denote the light radiance. Given L_{direct}, we compute outgoing radiance using Eq. (7).

Notice that this is a biased estimation for translucent objects, because the visibility is solved by the ray cast from a single point x, while the cumulative transfer function ρ assumes the full light visibility for the object surface (Figure 2). Therefore, if a translucent object is partially occluded from the distant light, the outgoing radiance is underestimated for those points casting occluded shadow rays, and overestimated for those points casting unobstructed shadow rays. This also happens for concave opaque objects due to global transport. But for ideally learned OSFs of convex opaque objects (i.e., they degenerate to BRDFs) as discussed in section 3.3, the estimation is unbiased.

Indirect lighting. Besides direct lighting, we also want to account for indirect lighting in evaluating Eq. (3). To do this, we use Monte Carlo integration with uniform sampling of solid angles. Specifically, we cast a ray from x to the sampled direction, and volume-render the ray using Eq. (1) to obtain the incoming radiance. Due to high computational cost, we only consider one-bounce indirect lighting. Similar to direct lighting, the estimation is biased for translucent objects, because ρ assumes distant light, while inter-reflection from objects do not meet this assumption. Nonetheless, our experiments show that our approximation can create reasonably faithful scene composition results in Sec. 4.3.

3.6 Accelerating OSF Rendering

The standard rendering procedure must be repeated per pixel to render an image. The total cost of rendering a single image with N_{pixel} pixels and N_{object} objects is thus

$$\mathcal{O}((N_{\text{pixel}}N_{\text{sample}}N_{\text{object}})(N_{\text{light}}N_{\text{sample}}N_{\text{object}})^{d-1}), \tag{9}$$

where N_{sample} denotes the number of samples per ray-object pair, N_{light} denotes the number of light sources, and d denotes the number of light bounces we model. There are many samples for which we must query OSFs, which can result in significant rendering time. We introduce two approaches to improve rendering efficiency: (i) reducing sample query time by KiloOSF and (ii) excluding rays that do not intersect with objects.

KiloOSF. To accelerate OSFs, in the same spirit as KiloNeRF (Reiser et al., 2021), we introduce a variant of our model called KiloOSF that represents the static scene with a large number of independent and small MLPs. KiloOSF represents each object with thousands of small MLPs, each responsible for a small portion of the object, making each individual MLP sufficient for photo-realistic rendering.

Specifically, we subdivide each *object* into a uniform grid of resolution $\mathbf{r} = (r_x, r_y, r_z)$. Each grid cell is of 3D index $\mathbf{i} = (i_x, i_y, i_z)$. We define a mapping m from position $\mathbf{x}$ to index $\mathbf{i}$ through spatial binning as follows:

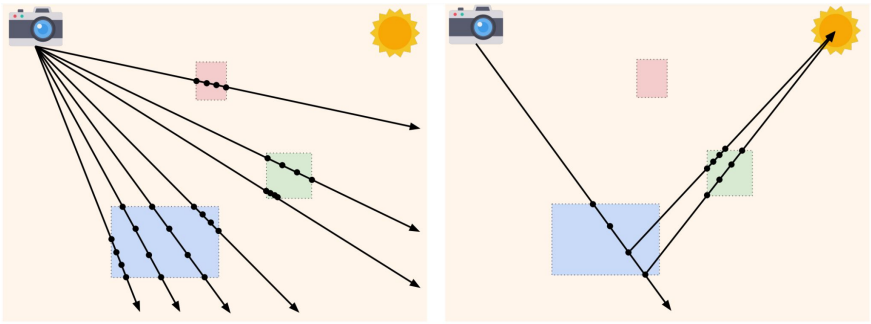
$$m(\mathbf{x}) = \lfloor (\mathbf{x} - \mathbf{b}_{\min})/(\mathbf{b}_{\max} - \mathbf{b}_{\min}) \rfloor, \tag{10}$$

where $\mathbf{b}_{\min}$ and $\mathbf{b}_{\max}$ are the respective minimum and maximum bounds of the axis-aligned bounding box (AABB) enclosing the object. For each grid cell, a small MLP neural network with parameters $\Theta(\mathbf{i})$ is used to represent the corresponding portion of the object. Then, the color and density values at a point $\mathbf{x}$ can be obtained by first determining the index $m(\mathbf{x})$ responsible for the grid cell that contains this point, then querying the respective small MLP:

$$\{\rho(\mathbf{x}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}), \sigma(\mathbf{x})\} = \text{NN}_{\Theta(m(\mathbf{x}))}(\mathbf{x}, \boldsymbol{\omega}_{\text{light}}, \boldsymbol{\omega}_{\text{out}}), \tag{11}$$

Following KiloNeRF, we use a "training with distillation" strategy. We first train an ordinary OSF model for each object and then distill the knowledge of the teacher model into the KiloOSF model. We also use empty space skipping and early ray termination to increase rendering efficiency. See Reiser et al. (2021) for details.

Excluding non-intersecting rays. In addition to using KiloOSF, we also implement a sampling procedure that precludes the evaluation of rays that do not intersect with objects. Our sampling procedure assumes



(a) Primary ray sampling (b) Shadow ray sampling

Figure 3: Sampling procedure. (a) Scene with a camera, light source, and object bounding boxes. Primary rays are sent from the camera into the scene. Rays that do not intersect with objects are pruned. Of the intersecting rays, we sample points within intersecting regions. (b) Shadow rays from each sample are sent to the light source, and samples within intersecting regions are evaluated.

access to (rough) bounding box dimensions for each object. In practice, such a bounding box can be automatically computed from a trained OSF by extracting the object's bounding volume (using the predicted alpha values). As shown in Fig. 3, for each ray and each object of interest, we intersect the ray with the object's bounding box. If the ray does not intersect with the object's bounding box, then no computation is required for the ray-object pair. If the ray does intersect with the object's bounding box, we adopt the intersection points as the near and far sampling bounds for the ray-object pair. Thus the $N_{\text{pixel}}N_{\text{object}}$ (number of primary rays) and $N_{\text{light}}N_{\text{object}}$ (number of secondary rays) are upper bounds on the number of rays that need to be evaluated. In practice, a single ray often only intersects with at most one object in the scene, which means that the proposed rendering procedure is not significantly more expensive than the single object setting.

Compared with traditional volumetric path tracing methods, our OSF model does not require running path tracing *within* each object (by evaluating itself during secondary ray computations) to simulate intra-object light bounces, because it learns the object-level scattering function that directly predicts the effects after all light bounces (reflections) and occlusions (shadows) within an object have occurred. Therefore, OSFs are faster than alternative methods such as NRFs (Bi et al., 2020a), which relies on simulating intra-object light bounces while querying its learned BRDF model.

4 Experiments

We validate our approach by free-viewpoint relighting on both synthetic and real-world objects, including translucent and opaque ones. Furthermore, we also demonstrate that OSFs allow visually plausible scene composition by showcasing a scene with both synthetic and real objects. We leave implementation details in our supplementary material.

4.1 Data and baselines

In our experiments, we use two real and six synthetic objects.

Real image capture setup. We use an easily accessible image-capturing setup, where we take photos with two iPhone 12 in a dark room. We use the flash of a cellphone P_{light} as a distant light source. We position P_{light} such that it is far away from the object (the distance to the object is about 10 times compared to the object diameter). We use the other cellphone P_{image} as a camera without using its flash. We capture 20 images from random viewpoints for a single random light direction, and we repeat this for 10 different light directions. This gives us a dataset of 200 images from different viewpoints. We split our dataset such that all 20 images of a randomly chosen light direction are held out for testing and all other images for training. We use a standard Structure from Motion (SfM) method, COLMAP (Schonberger & Frahm, 2016), to solve

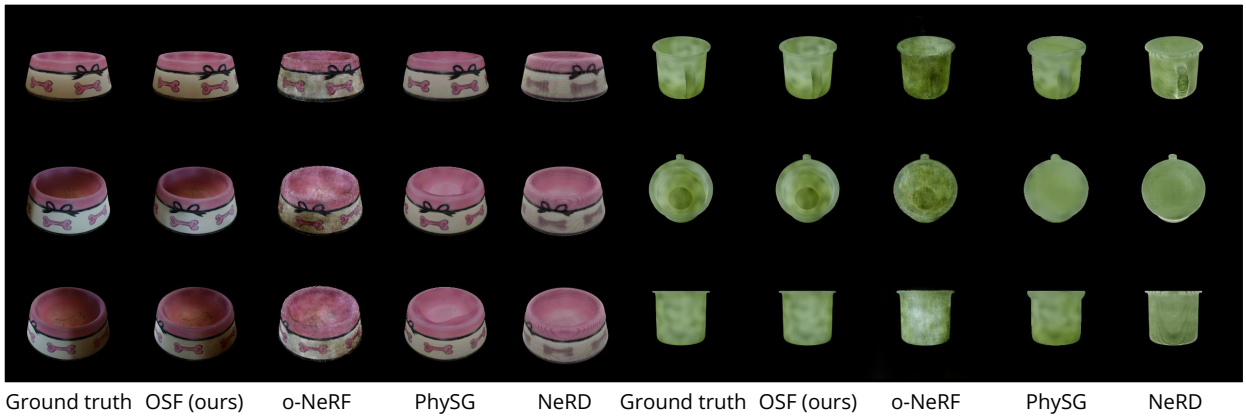


Figure 4: Free-viewpoint relighting for translucent objects from ObjectFolder.

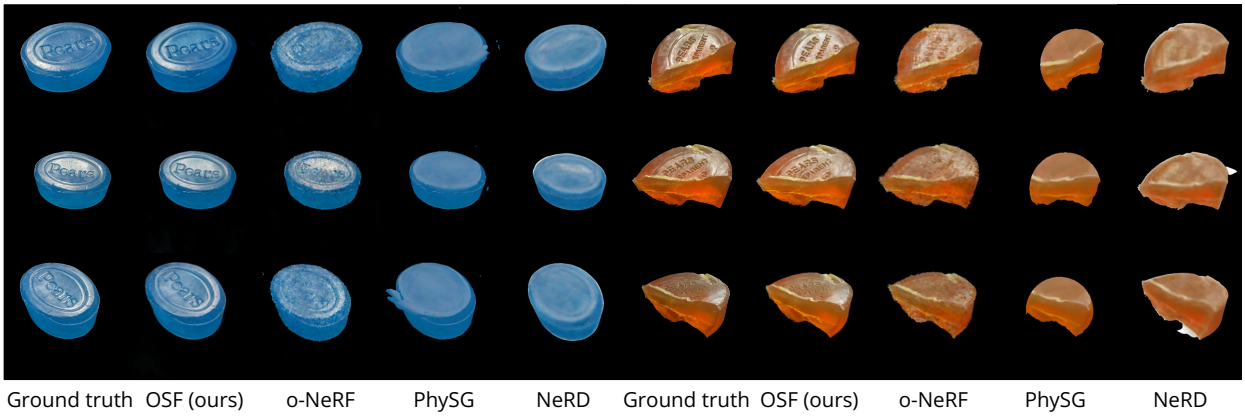


Figure 5: Results on free-viewpoint relighting for real translucent objects captured by a cellphone.

for camera poses for all images. To calibrate light position, we additionally take a photo using $\mathrm{P}_{\mathrm{light}}$ for every light position, and solve the camera pose together with other captured images. Photos taken by $\mathrm{P}_{\mathrm{light}}$ are only used for light direction calibration but not for training. In order for SfM to work well in the dark room, we place highly-textured newspaper pieces surrounding the captured object to provide robust feature points, and use a foreground segmentation network (Qin et al., 2020) to extract clean object images after SfM.

Synthetic images. We use 5 synthetic objects from the ObjectFolder (Gao et al., 2021) dataset, as well as a Stanford bunny. We assign translucent materials to half of them and opaque materials to the other half. For each synthetic object, we render 1,000 images, each of which is under a random light direction and a random viewpoint. We use 500 images for training and 500 for testing.

Baselines. We consider the following baselines:

o-NeRF (Mildenhall et al., 2020): We train a NeRF for each object and refer to it as o-NeRF. For scene composition, each object is rendered separately. We also add shadows like our OSFs, denoting them as *o-NeRF + Shadow*.

PhySG (Zhang et al., 2021b): A recent representative neural inverse rendering method that jointly estimates lighting, materials, and geometry from multi-view images and masks of shiny objects.

NeRD (Boss et al., 2021a): A recent representative neural relightable representation that decomposes the appearance into a neural BRDF field and environment lighting.

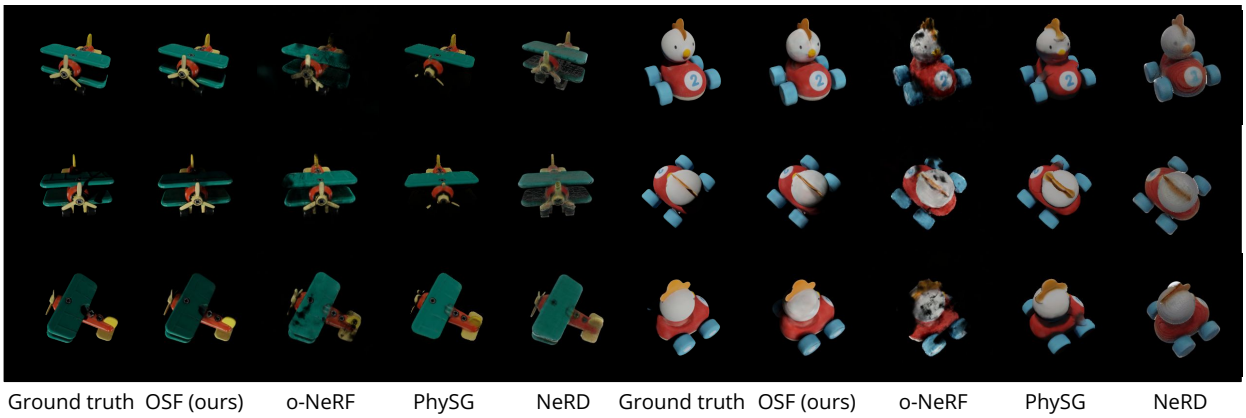


Figure 6: Free-viewpoint relighting for opaque objects from ObjectFolder.

	PSNR↑	SSIM↑	LPIPS↓
o-NeRF	23.69	0.811	0.137
PhySG	26.63	**0.941**	0.046
NeRD	24.39	0.892	0.125
OSF (ours)	**34.06**	0.923	**0.035**

Table 1: Results for free-viewpoint relighting on translucent objects from ObjectFolder.

	PSNR↑	SSIM↑	LPIPS↓
o-NeRF	17.63	0.725	0.331
PhySG	16.81	0.833	0.185
NeRD	21.94	0.815	0.182
OSF (ours)	**27.06**	**0.865**	**0.051**

Table 2: Results for free-viewpoint relighting on real translucent objects captured by a cellphone.

We use the following standard metrics in all our comparisons: peak signal-to-noise ratio (PSNR), structural similarity (SSIM) (Wang et al., 2003), and a perceptual metric LPIPS (Zhang et al., 2018).

4.2 Free-Viewpoint Relighting

OSF takes light directions as input in addition to the camera viewpoint; thus, it inherently supports simultaneous relighting and novel view synthesis. We use a variety of objects for evaluation, including three opaque synthetic objects from the ObjectFolder dataset (Gao et al., 2021), three translucent synthetic objects, and two real-world translucent objects.

We train one model per object. For each synthetic object, we train on 400 images with randomized viewpoint and lighting, and test on 100 images of novel viewpoint and lighting. For each real object, we train on 200 captured images with 10 different light directions, and test on 20 images with a novel light direction.

Relighting synthetic translucent objects. We show qualitative (Figure 4) and quantitative (Table 1) comparisons on synthetic translucent objects. Our approach significantly outperforms the baseline methods. It successfully models the material and shape of the objects, and accurately relights them from varied viewpoints compared to the ground truth. In contrast, o-NeRF and methods that parameterize materials by BRDFs, such as PhySG and NeRD, fail to produce complex shading effects, and instead relight the entire body of the object.

Relighting real translucent objects. To demonstrate that our approach works well on challenging real-world translucent objects, we capture images of two objects with a cell phone in a dark room from different camera viewpoints under varied light directions with a point light source. We show quantitative (Table 2) and qualitative (Figure 5) results of two translucent soaps. Our method consistently outperforms the baseline methods. While both o-NeRF, PhySG, and NeRD cannot recover the correct shading effects on the objects, our approach well captures the detailed textures on the surface of the objects.

	PSNR↑	SSIM↑	LPIPS↓
o-NeRF	20.93	0.817	0.181
PhySG	22.30	0.884	0.125
NeRD	21.12	0.866	0.106
OSF (ours)	**31.60**	**0.905**	**0.065**

Table 3: Results for free-viewpoint relighting on opaque objects.

	PSNR↑	SSIM↑	LPIPS↓
o-NeRF+Shadow	17.21	0.678	0.244
OSF (ours)	**35.57**	**0.927**	**0.031**

Table 4: Free-viewpoint relighting results for scene composition.

	FPS↑	PSNR↑
OSF	0.27	**0.035**
KiloOSF	**16.13**	0.037

Table 5: Evaluation of KiloOSF on relighting translucent objects.

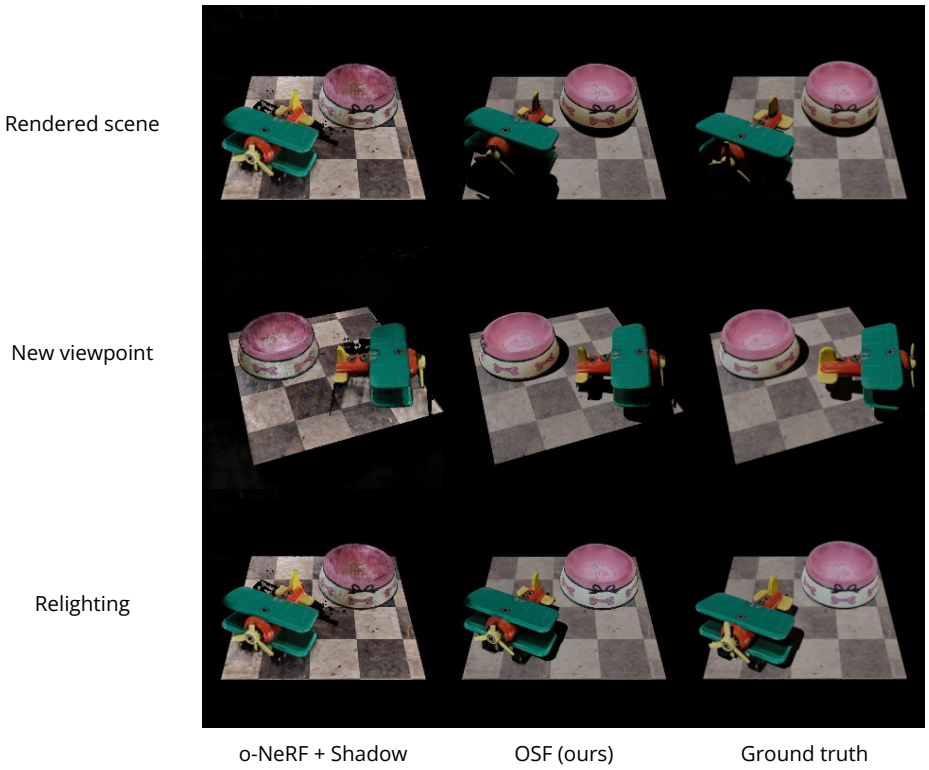


Figure 7: Scene composition results for a translucent bowl, an opaque airplane, and an opaque floor.

Relighting synthetic opaque objects. In Figure 6, we show the free-viewpoint relighting results for synthetic opaque objects. o-NeRFs predict radiance fields instead of modeling light transport; therefore, they do not perform well in novel lighting conditions. OSFs also outperform PhySG and NeRD. While baselines fail to predict self-shadows, our method produces more accurate appearance of the objects under novel lighting conditions. We show quantitative results in Table 3, where OSFs also significantly outperform all baselines.

4.3 Scene Composition

Since OSFs are object-centric representations that model light transport for an object instead of appearances under specific lighting conditions, they allow trained models to be readily composed into new scenes. To evaluate neural scene composition, we use a translucent and an opaque object with randomized poses, lighting directions, and viewpoints. We show examples in Figure 7. We compare with the *o-NeRF+Shadow* baseline. Qualitatively, while *o-NeRF+Shadow* produces incorrect shadows and artifacts, our method generates more realistic compositions that match the ground truth. We also show quantitative results in Table 4, where we observe that OSFs significantly outperform *o-NeRF+Shadow*.

Figure 8: Scene composition results on real objects. We compare to ablated versions of our model. "Ours w/o shadow or indirect" considers only direct illumination. "Ours w/o indirect" includes both direct illumination and shadows. "Indirect only" considers only indirect illumination.

Composing real objects. In the rightmost column of Figure 8, we further show qualitative examples of scene composition using real objects from Bi et al. (2020a). In the first row, we show the composition results of real objects on the synthetic floor. In the second row, we compose the objects into a real background scene from Mildenhall et al. (2020) where the light sources are randomized to approximate the unknown environment lighting. In both examples, our approach demonstrates good composition results with realistic shadows.

Ablation study. In the left three columns of Figure 8, we show an ablation study comparing our full model to two variants: (1) ours without shadow or indirect lighting, and (2) ours without indirect lighting. Without accounting for shadow or indirect lighting, the composition results appear to be unrealistic, similar to a simple cut-and-paste (the first column of Figure 8). While adding shadows improves the visual quality, the lack of indirect lighting leads to hard shadows in the first example, which degrades the photorealism (the second column of Figure 8). In comparison, our full model incorporates both shadow and indirect lighting (visualization of indirect lighting effects is shown in the third column of Figure 8), leading to visually plausible scene compositions.

4.4 Evaluation on KiloOSF

We introduce KiloOSF for accelerating OSF rendering. Thus, in this section, we evaluate KiloOSF in terms of rendering speed and relighting performance. To this end, we test KiloOSF on all synthetic translucent objects. We show the rendering speed (frames per second) and the rendering quality in Table 5. KiloOSF achieves 60× (16.13 vs. 0.27 FPS) speed up while maintaining comparable relighting performances, suggesting the benefits brought by the KiloOSF design.

5 Conclusion

We presented a novel, learning-based object representation—Object-Centric Neural Scattering Functions (OSFs). An OSF models a cumulative radiance transfer function rather than a radiance field modeled by NeRF, allowing both free-viewpoint relighting and scene composition. Moreover, OSFs can model objects of complex shape and materials, and can relight both opaque and translucent objects. Our work offers a new promising graphics pipeline for modeling real-world scenes that combines the flexibility of object-centric neural modeling with the photorealism of graphics rendering algorithms.

References

Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *European Conference on Computer Vision*, 2020.

Hendrik Baatz, Jonathan Granskog, Marios Papas, Fabrice Rousselle, and Jan Novák. Nerf-tex: Neural reflectance field textures. In *Computer Graphics Forum*. Wiley Online Library, 2021.

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

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *arXiv preprint arXiv:2111.12077*, 2021b.

Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020a.

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. *European Conference on Computer Vision*, 2020b.

Adrian Blumer, Jan Novák, Ralf Habel, Derek Nowrouzezahrai, and Wojciech Jarosz. Reduced aggregate scattering operators for path tracing. In *Computer Graphics Forum*. Wiley Online Library, 2016.

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

Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34, 2021b.

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.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 145–156, 2000.

Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *CoRL*, 2021.

Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *arXiv:2103.10380*, 2021.

Gaurav Garg, Eino-Ville Talvala, Marc Levoy, and Hendrik PA Lensch. Symmetric photography: Exploiting data-sparseness in reflectance fields. In *Rendering Techniques*, pp. 251–262, 2006.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6):1–15, 2018.

Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *SIGGRAPH*, 1984.

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

Alexandr Kuznetsov. Neumip: Multi-resolution neural materials. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.

Richard Lee and Carol O'Sullivan. Accelerated light propagation through participating media. In *VG@ Eurographics*, pp. 17–23, 2007.

Jaakko Lehtinen. A framework for precomputed and captured light transport. *ACM Transactions on Graphics (TOG)*, 26(4):13–es, 2007.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 2020.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM Transactions on Graphics*, 38 (4):1–14, 2019.

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, et al. Lookingood: Enhancing performance capture with real-time neural re-rendering. *ACM Transactions on Graphics*, 2018.

Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

Nelson Max and Min Chen. Local and global illumination in the volume rendering integral. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2005.

Johannes Meng, Marios Papas, Ralf Habel, Carsten Dachsbacher, Steve Marschner, Markus H Gross, and Wojciech Jarosz. Multi-scale modeling and rendering of granular materials. *ACM Trans. Graph.*, 34(4): 49–1, 2015.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Conference on Computer Vision and Pattern Recognition*, pp. 6878–6887, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. *arXiv preprint arXiv:2111.13679*, 2021.

Jonathan T Moon, Bruce Walter, and Stephen R Marschner. Rendering discrete random media using precomputed scattering solutions. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp. 231–242, 2007.

Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Transactions on Graphics (TOG)*, 2018.

Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021.

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. Monte carlo methods for volumetric light transport simulation. In *Computer Graphics Forum*. Wiley Online Library, 2018.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction, 2021.

Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2856–2865, 2021.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106: 107404, 2020.

Ravi Ramamoorthi. *Precomputation-based rendering*. NOW Publishers Inc, 2009.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14335–14345, October 2021.

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

Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *CoRL*, 2021.

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Conference on Computer Vision and Pattern Recognition*, pp. 2437–2446, 2019a.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pp. 1121–1132, 2019b.

Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Conference on Computer Graphics and Interactive Techniques*, pp. 527–536, 2002.

Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7495–7504, 2021.

Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on Graphics*, 38(4):79–1, 2019.

Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*. Wiley Online Library, 2020.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics*, 38(4):1–12, 2019.

Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International Workshop on Vision Algorithms*, pp. 298–372. Springer, 1999.

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. *arXiv preprint arXiv:2112.03907*, 2021.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pp. 1398–1402. IEEE, 2003.

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics*, 37(4):1–13, 2018.

Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *International Conference on Computer Vision (ICCV)*, October 2021.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.

Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *ICLR*, 2022.

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

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NERF++: Analyzing and improving neural radiance fields. *https://arxiv.org/abs/2010.07492*, 2020.

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

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern recognition*, pp. 586–595, 2018.

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

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

Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *International Conference on Computer Vision*, pp. 7194–7202, 2019.

Zhenglong Zhou, Zhe Wu, and Ping Tan. Multi-view photometric stereo with spatially varying isotropic materials. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.